01LIP – Linear Programming

Concise Lecture Notes

Jan Bureš

Version: August 2026

PREFACE

These lecture notes accompany the linear programming course 01LIP at the Faculty of Nuclear Sciences and Physical Engineering of the Czech Technical University in Prague and follow its structure exactly.

The content draws extensively on existing lecture notes and monographs, especially *Lineární programování: Úvod pro informatiky* (Matoušek, 2006), *Lineární a nelineární programování* (Rohn, 1997), *Understanding and Using Linear Programming* (Matoušek and Gärtner, 2007), *Linear Programming: Foundations and Extensions* (Vanderbei, 2020), *Introduction to Linear Optimization* (Bertsimas and Tsitsiklis, 1997), and *Theory of Linear and Integer Programming* (Schrijver, 1986).

The text also covers topics beyond the scope of the exam for this course.

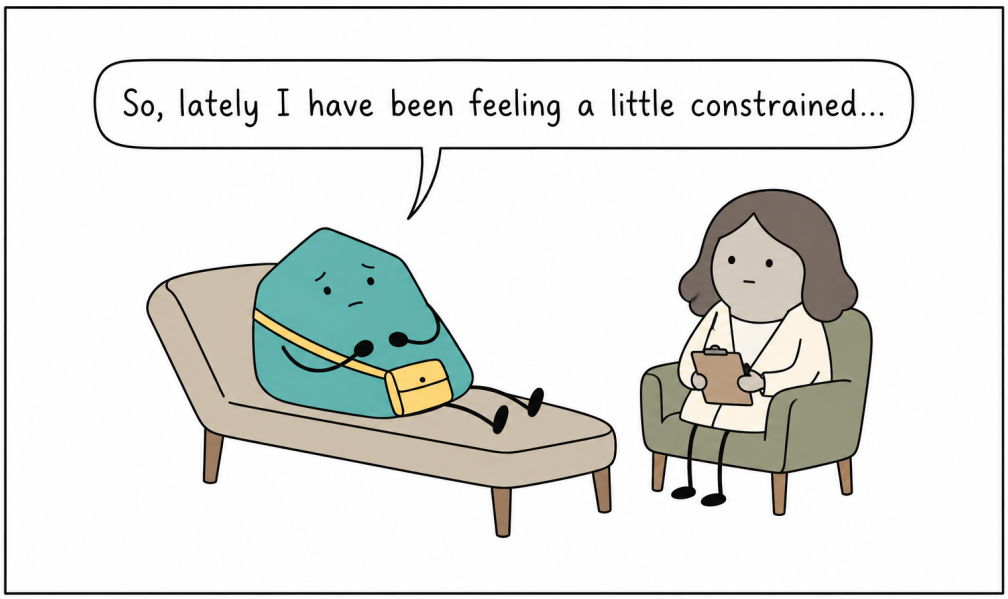
Jan Bureš
Dallas, August 2026

CONTENTS

CHAPTER 1

FORMULATING A LINEAR PROGRAMMING PROBLEM

We introduce the basic concepts of linear programming and show how to turn verbal descriptions and different formulations into a common form. We define feasible and optimal solutions, distinguish integer from mixed-integer programming, and outline the connections to linear algebra and the history of the field. We finish by constructing six models that illustrate the range of applications of LP.



1 The basics of linear programming

We begin with a fact that may come as a surprise: the word *programming* here does not mean writing computer code. Historically, it referred to preparing a plan or schedule. The field emerged before computers were widely available. We will use *linear program* (LP) to mean an optimization problem with a linear objective function and finitely many linear equations or inequalities. The field is therefore also called *linear optimization*. For basic theory and terminology, see, for example, Matoušek (2006) and Rohn (1997). A systematic modern treatment is given by Matoušek and Gärtner (2007).

The formal statement of an LP is surprisingly simple, yet it provides an exceptionally useful tool. A linear model is appropriate whenever the essential relationships can be expressed as sums: machine and resource capacities, material or energy balances, budget constraints, or minimum quality requirements. Decision variables describe our choices, constraints specify which choices are allowed, and the objective function determines which choice we regard as best. We now translate this verbal description into matrix notation.

1.1 Standard and equality forms of a linear program

We first establish the notation and two problem forms that we will move between in later chapters. Their names are not used consistently in the literature, so we will always give both a name and a precise formulation.

Notation 1.1: Notation and conventions

Let $m, n \in \mathbb{N}$ be the numbers of constraints and variables. Vectors are column vectors. The matrix $A \in \mathbb{R}^{m\times n}$ has ith row $\mathbf{a}_i^{\mathsf{T}}$ and entries a_{ij}. Also, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{c} \in \mathbb{R}^n$. Vector inequalities, such as $\mathbf{x} \geq \mathbf{0}$, are always interpreted componentwise. The superscript T denotes transposition.

Standard maximization form

In this text, standard maximization form means the problem

$$\begin{aligned} \text{maximize} \quad & \mathbf{c}^{\mathsf{T}}\mathbf{x}, \\ \text{subject to} \quad & A\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned} \tag{1.1}$$

Definition 1.2: Objective function

The linear function

$$\mathbf{c}^{\mathsf{T}}\mathbf{x} = \sum_{j=1}^{n} c_j x_j$$

is called the *objective function*. The variables x_j are *decision variables*, and the components of $\mathbf{c}$ are their coefficients in the objective function.

In general optimization, the objective function can be almost arbitrary. In this course, however, we will stay true to its name and work with linear functions. Their simplicity will later allow us to connect computation with particularly intuitive geometry.

In component form, (1.1) becomes

$$\max \sum_{j=1}^{n} c_j x_j \quad \text{subject to} \quad \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad (i = 1, \dots, m), \qquad x_j \ge 0 \quad (j = 1, \dots, n).$$

The nonnegativity condition must be stated explicitly. Without it, we would have a different problem.

Similarly, standard minimization form will mean

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^{\mathsf{T}}\mathbf{x}, \\ \text{subject to} \quad & A\mathbf{x} \ge \mathbf{b}, \\ & \mathbf{x} \ge \mathbf{0}. \end{aligned}$$

Choosing the opposite inequality direction is useful when constructing the dual problem. Any minimization can also be converted to a maximization by changing the sign of the objective function.

Remark 1.3: Free variables

A variable with no sign restriction, $x_j \in \mathbb{R}$, is called *free*. We replace it by the difference $x_j = x_j^+ - x_j^-$, where $x_j^+, x_j^- \ge 0$. The expanded model has one additional variable. The map $(x_j^+, x_j^-) \mapsto x_j$ is not one-to-one, but its image covers every original value. Returning to the original variables therefore gives exactly the original feasible solutions. In a pure integer linear program (ILP), x_j^+ and x_j^- must also be integers. In contrast, shifting an integer variable by a noninteger lower bound generally does not preserve integrality.

Equality form will be convenient for the simplex method. For a $\le$ inequality, we introduce a *slack variable* $s_i \ge 0$,

$$\mathbf{a}_i^{\mathsf{T}}\mathbf{x} \le b_i \quad \Longleftrightarrow \quad \mathbf{a}_i^{\mathsf{T}}\mathbf{x} + s_i = b_i,$$

and for a $\ge$ inequality, a *surplus variable* $s_i \ge 0$,

$$\mathbf{a}_i^{\mathsf{T}}\mathbf{x} \ge b_i \quad \Longleftrightarrow \quad \mathbf{a}_i^{\mathsf{T}}\mathbf{x} - s_i = b_i.$$

Equality form of an LP

After splitting free variables and introducing slack and surplus variables, a linear program can be written as

$$\max \ \mathbf{c}^{\mathsf{T}}\mathbf{x} \quad \text{subject to} \quad A\mathbf{x} = \mathbf{b}, \qquad \mathbf{x} \ge \mathbf{0}. \tag{1.2}$$

Here $A, \mathbf{x}, \mathbf{c}$ include the newly introduced columns, variables, and zero coefficients in the objective function.

If the original system is divided into $\le$, $=$, and $\ge$ blocks, the same conversion can be written in more detail as

$$\begin{aligned} A_{\le}\mathbf{x} + \mathbf{s}_{\le} &= \mathbf{b}_{\le}, \\ A_{=}\mathbf{x} &= \mathbf{b}_{=}, \qquad \mathbf{x}, \mathbf{s}_{\le}, \mathbf{s}_{\ge} \ge \mathbf{0}. \\ A_{\ge}\mathbf{x} - \mathbf{s}_{\ge} &= \mathbf{b}_{\ge}, \end{aligned}$$

The original and expanded models are equivalent in the following precise sense: each original feasible point determines unique slack and surplus variables. If free variables have been split, the original point is recovered by a linear map from the expanded point. The objective value does not change.

Why go to the trouble of using equality form? Equalities together with nonnegativity allow us to select a few columns of the matrix, compute the corresponding variables, and set all the others to zero. This seemingly purely algebraic construction will correspond exactly to the corners of the feasible set. It leads directly to basic feasible solutions and the simplex method.

Remark 1.4: Different names for problem forms

Some authors call equality form *standard* or *canonical*. Others use the same words for inequality form. In this book, *standard form* means (1.1), and *equality form* means (1.2). The precise formula always takes precedence over its name.

1.2 The feasible set and optimal solutions

In any optimization problem, we must first specify which points we may choose from. In standard form, we can read this set directly from the constraints.

Definition 1.5: Feasible set and optimal solution

For problem (1.1), the set

$$P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \leq \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$$

is the *feasible set*. A point $\mathbf{x} \in P$ is a *feasible solution*. A feasible point $\mathbf{x}^\star$ is an *optimal solution* to the maximization problem if

$$\mathbf{c}^\mathsf{T}\mathbf{x} \leq \mathbf{c}^\mathsf{T}\mathbf{x}^\star \qquad \text{for all } \mathbf{x} \in P.$$

The number $z^\star = \mathbf{c}^\mathsf{T}\mathbf{x}^\star$ is the *optimal value*. For minimization, the direction of the last inequality is reversed.

Remark 1.6: The three possible outcomes of a linear program

A maximization linear program is in exactly one of the following situations:

a) it is *infeasible*, meaning $P = \varnothing$,

b) it is *unbounded above*, meaning $P \neq \varnothing$ and $\sup_{\mathbf{x}\in P} \mathbf{c}^\mathsf{T}\mathbf{x} = +\infty$,

c) it has a finite optimal value and at least one optimal solution.

For minimization, the second case is unboundedness below. The fact that a finite supremum of a linear function on a nonempty polyhedron is attained follows from the fundamental theorem of linear programming, which we will prove later. When we first analyze a particular problem, we therefore always ask which of these three outcomes occurs.

2 Converting between forms

Verbal models come in many different forms. The following practical recipes let us convert them into a common form that is convenient for both geometry and algorithms. Each transformation preserves feasible solutions and objective values after the stated reverse conversion.

1. Minimization and maximization.

$$\min\ \mathbf{c}^{\mathsf{T}}\mathbf{x} \quad \Longleftrightarrow \quad \max\ (-\mathbf{c})^{\mathsf{T}}\mathbf{x}.$$

The optimal points do not change, and the optimal value changes sign.

2. Inequality direction and equality.

$$\mathbf{a}_i^{\mathsf{T}}\mathbf{x} \geq b_i \quad \Longleftrightarrow \quad (-\mathbf{a}_i)^{\mathsf{T}}\mathbf{x} \leq -b_i.$$

An equality can be replaced by a pair of opposite inequalities. In a model, however, we usually leave it as an equality, since doubling the row is unnecessary.

3. Equality form. For a $\leq$ inequality, add a nonnegative slack variable. For a $\geq$ inequality, subtract a nonnegative surplus variable, as described above. Introducing a surplus variable alone does not generally produce an initial feasible basis. Chapter 5 addresses this issue.

A slack variable measures unused capacity, while a surplus variable measures the excess above a required bound. These variables are more than an algebraic trick: in many models, they have a direct practical interpretation.

4. Free variables and bounds. Replace a free variable by the difference of two nonnegative variables. If a variable has bounds $L_j \leq x_j \leq U_j$, introduce

$$x_j = L_j + y_j, \qquad 0 \leq y_j \leq U_j - L_j.$$

This removes the lower bound, while the upper bound remains as a single linear constraint.

5. Absolute values and maxima of linear functions. For a linear function ℓ,

$$|\ell(\mathbf{x})| \leq t \quad \Longleftrightarrow \quad -t \leq \ell(\mathbf{x}) \leq t,$$

where $t \geq 0$ already follows from these two inequalities if the system is feasible. The minimization problem

$$\min \sum_i w_i |\ell_i(\mathbf{x})|, \qquad w_i \geq 0,$$

can be linearized using $t_i \geq 0$ and the constraints $-t_i \leq \ell_i(\mathbf{x}) \leq t_i$, with objective $\sum_i w_i t_i$. For $w_i > 0$, every optimum must satisfy $t_i = |\ell_i(\mathbf{x})|$. If $w_i = 0$, this equality need not hold at every optimal solution, but t_i can be reduced to the absolute value without changing the objective value.

Similarly,

$$\min_{\mathbf{x}} \max_{k=1,\ldots,q} \{\mathbf{a}_k^{\mathsf{T}}\mathbf{x} + b_k\}$$

can be converted to minimizing t subject to $\mathbf{a}_k^{\mathsf{T}}\mathbf{x} + b_k \leq t$ for every k.

Remark 1.7: A conversion checklist

There is no single required order of transformations. Always check, however:

1. that the sign of the objective function has been changed correctly,
2. that all inequality directions have been preserved,
3. that the domain of every variable is stated explicitly,
4. that slack and surplus variables have zero objective coefficients,
5. how to recover the original variables from a solution to the expanded model.

For the simplex method, we will usually aim for maximization in equality form, $\max\{\mathbf{c}^{\mathsf{T}}\mathbf{x} \mid A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$.

Example 1.8: Converting a problem with a free variable

Consider

$$\min\ 2x_1 - x_2 \quad \text{subject to} \quad -x_1 + 2x_2 \geq 1, \qquad x_1 + x_2 = 3,$$

where $x_1 \geq 0$ and $x_2 \in \mathbb{R}$ is free. We convert the problem step by step:

1. change the sign of the objective function to obtain $\max(-2x_1 + x_2)$,
2. split the free variable as $x_2 = x_2^+ - x_2^-$, where $x_2^+, x_2^- \geq 0$,
3. multiply the inequality by -1 to obtain $x_1 - 2x_2 \leq -1$,
4. introduce a slack variable $s \geq 0$ and replace the inequality by an equality.

Substituting the expression for x_2 gives the following equality form:

$$\begin{aligned} \max \quad & -2x_1 + x_2^+ - x_2^-, \\ \text{subject to} \quad & x_1 - 2x_2^+ + 2x_2^- + s = -1, \\ & x_1 + x_2^+ - x_2^- = 3, \\ & x_1, x_2^+, x_2^-, s \geq 0. \end{aligned}$$

The negative right-hand side of the first equation does not mean that the conversion has lost equivalence. It indicates that the slack variable alone will not provide an initial feasible basis.

Example 1.9: Converting interval bounds

Interval bounds arise, for example, when a variable represents a capacity with both a prescribed minimum and a maximum. They do not require us to introduce two new variables.
For $1 \leq x \leq 5$, set $x = 1 + y$. Then $0 \leq y \leq 4$. Replace x by $1 + y$ in every constraint and in the objective function. For equality form, the upper bound can

be written as $y + s = 4$, where $s \geq 0$.

3 Integer and mixed-integer programming

In many practical problems, variables cannot take arbitrary real values. The number of vehicles or shifts must be an integer, and a decision whether to open a facility can naturally be represented by a binary variable. Such conditions fundamentally change both the geometry and the computational difficulty of the problem. For now, we will learn only to recognize and formulate ILPs and mixed-integer linear programs (MILPs) correctly. We return to methods for solving them in the final chapter.

Integer linear program (ILP)

In a pure integer linear program, all decision variables are integers. For example,

$$\max \ \mathbf{c}^{\mathsf{T}}\mathbf{x} \quad \text{subject to} \quad A\mathbf{x} \leq \mathbf{b}, \qquad \mathbf{x} \in \mathbb{Z}_{+}^{n}.$$

A binary variable additionally satisfies $x_j \in \{0, 1\}$.

Mixed-integer linear program (MILP)

In a mixed-integer model, only some of the variables are required to be integers:

$$\begin{aligned} \max \quad & \mathbf{c}_y^{\mathsf{T}}\mathbf{y} + \mathbf{c}_z^{\mathsf{T}}\mathbf{z}, \\ \text{subject to} \quad & A_y\mathbf{y} + A_z\mathbf{z} \leq \mathbf{b}, \\ & \mathbf{y} \in \mathbb{Z}_{+}^{p}, \qquad \mathbf{z} \in \mathbb{R}_{+}^{n-p}. \end{aligned}$$

The feasible set of a pure ILP consists of the integer points of a polyhedron and is therefore discrete. The feasible set of an MILP may contain entire continuous polyhedral pieces, but it is generally not convex. We therefore cannot transfer geometric conclusions from LP to ILP or MILP without further justification.

3.1 Computational complexity

Statements about complexity assume rational input data encoded in binary. The input size includes the number of bits needed to represent the coefficients. Linear programming can be solved in time polynomial in this input size. In contrast, the decision version of general integer linear programming is NP-complete, and the optimization version is NP-hard. MILP contains ILP as a special case (Schrijver, 1986, Papadimitriou, 1981). This does not rule out solving many practical instances quickly. If P $\neq$ NP, however, there is no polynomial-time algorithm for all instances of general ILP.

Remark 1.10: What do NP-hard and NP-complete mean here?

A decision problem asks only for a yes-or-no answer, such as whether there is a feasible integer solution with objective value at least K. A problem is *NP-hard* if every problem in the class NP can be reduced to it in polynomial time. If it also

belongs to NP, it is called *NP-complete*. This does not mean that every particular problem must be slow to solve in practice. It warns us that, in general, we should not expect a single algorithm that is always fast for every instance.

3.2 The LP relaxation

The *LP relaxation* is obtained by dropping the integrality conditions while keeping all other constraints. This enlarges the feasible set. For maximization, $z^\star_{\mathrm{LP}} \geq z^\star_{\mathrm{ILP}}$, while for minimization the inequality is reversed, provided both optimal values are finite.

What good is the solution of the relaxed problem if we wanted integers in the first place? Precisely because we optimize over a larger set, we obtain an optimistic bound on the outcome of the original model. For maximization, it is an upper bound. For minimization, it is a lower bound. Later, we will use this bound to decide which parts of a branch-and-bound tree can no longer contain a better integer solution.

The nonnegative additive gap for a particular instance is $z^\star_{\mathrm{LP}} - z^\star_{\mathrm{ILP}}$ for maximization and $z^\star_{\mathrm{ILP}} - z^\star_{\mathrm{LP}}$ for minimization. The multiplicative *integrality ratio* is usually defined only for positive optimal values: as $z^\star_{\mathrm{ILP}}/z^\star_{\mathrm{LP}} \geq 1$ for minimization and $z^\star_{\mathrm{LP}}/z^\star_{\mathrm{ILP}} \geq 1$ for maximization. Without positivity, the ratio may be undefined or misleading. The term *integrality gap* is also often used for the supremum of these ratios over an entire specified class of instances. We will therefore always state whether we mean a single instance or a class of problems.

4 A brief comparison with linear algebra

You probably spent your first semester of linear algebra studying matrices, systems of equations, and Gaussian elimination. The same objects return in linear programming, but we ask a more demanding question: we seek the best solution, rather than just any solution. Four parallels are worth keeping in mind.

The central problem. Linear algebra describes all solutions of $A\mathbf{x} = \mathbf{b}$, usually an affine subspace. Linear programming adds inequalities and an objective function $\mathbf{c}^{\mathsf{T}}\mathbf{x}$. The feasible region is a polyhedron, and we are interested in its best point.

The computational workhorse. Gaussian elimination dominates linear algebra. In LP, the same row operations appear inside the simplex method, which moves from one vertex to another. We will later also discuss interior-point methods, which instead proceed through the interior of the polyhedron.

A certificate of infeasibility. A rank test detects inconsistent equations. Farkas' lemma will play a similar role for systems of inequalities.

Geometry. An affine subspace is flat and has no distinguished points. In the feasible polyhedron of our equality form, vertices will have a special role. We will show that if this polyhedron is nonempty and the problem has a finite optimal value, at least one optimum is attained at a vertex.

Linear algebra is therefore more than a formal prerequisite for this course. It is the language in which we will actually compute the geometry and algorithms of linear programming.

5 How linear programming developed

Linear programming did not emerge from a single discovery at a single moment. Its mathematical components arose separately: the theory of linear inequalities led to Farkas' lemma, von Neumann's minimax theorem connected optimization with matrix games, and Hitchcock formulated the transportation problem before a general algorithm existed (Farkas, 1902, von Neumann, 1928, Hitchcock, 1941). These strands came together as a modern field only when mathematics met urgent questions of production, economics, and wartime planning. Calling any one person its sole inventor therefore misses part of the story (Cottle et al., 2007).

5.1 Leningrad, now St. Petersburg: from plywood to a general method

In 1938, engineers from the laboratory of a plywood trust approached the young professor Leonid V. Kantorovich in Leningrad. They needed to schedule eight veneer lathes with different productivities for five types of wood. The enterprise had to maintain a prescribed product mix and make the best use of machine time. Kantorovich recognized a general problem of optimally allocating scarce resources (Kantorovich, 1960, Boldyrev and Düppe, 2020).

He formulated a whole class of problems using linear constraints and an optimization criterion. His constraint multipliers anticipated dual variables and shadow prices. Under suitable conditions, these describe the marginal change in the optimal value resulting from a small change in the amount of a resource available. We will explain the precise conditions in the chapters on duality (Kantorovich, 1960, Boldyrev and Düppe, 2020).

The booklet *Mathematical Methods of Organizing and Planning Production* appeared in Leningrad in 1939 in an edition of a thousand copies. An English translation followed only in 1960. The work thus remained outside normal international circulation for many years and did not immediately influence developments in the United States (Kantorovich, 1960, Boldyrev and Düppe, 2020).

During the siege of Leningrad, Kantorovich's work acquired a different urgency. Trucks carried supplies across frozen Lake Ladoga. Safe passage along this "Road of Life" depended on the strength of the ice. Kantorovich belonged to a team that calculated its load-bearing capacity, taking ice properties, weather, and loading into account. Here, mathematical calculations helped guide the safe supply of the city (Nazarov and Sinkevich, 2019, p. 58).

Remark 1.11: Two unusual homework problems

In 1939, George B. Dantzig began doctoral studies with the statistician Jerzy Neyman at Berkeley. According to his later recollection, he once arrived late to a lecture, copied two problems from the board, and assumed they were homework. He submitted solutions a few days later. Only then did he learn that they were two unsolved problems in mathematical statistics. The results became the basis of his dissertation (Albers and Reid, 1986). The story is often retold as a legend about a problem solved overnight. Dantzig's own account is more restrained: he worked for several days and did not know the solutions were supposed to be difficult. Although the problems were not formulated as ordinary finite linear programs, Dantzig later

recalled that the geometric work with columns in his dissertation helped him develop the simplex method (Albers and Reid, 1986, Dantzig, 2002). The story is a fitting introduction to someone willing to search for a general method where others saw only too many possibilities.

5.2 Washington: when a program meant a plan

World War II interrupted Dantzig's doctoral studies. From 1941, he worked as a civilian statistician for the U.S. Army Air Forces and headed a unit that analyzed combat operations. After completing his doctorate in 1946, he returned to the Pentagon as a mathematical adviser. His colleagues posed a question that sounded more organizational than mathematical: could the planning of Air Force deployment, training, and supply over time be made faster? The computing equipment of the day consisted mainly of punched cards, accounting machines, and people with desk calculators (Albers and Reid, 1986, Cottle et al., 2007).

Here, a *program* meant a plan of activities. The research group later named Project SCOOP (*Scientific Computation of Optimum Programs*) was therefore not established to write computer code. Its task was to compare possible plans that had to respect available aircraft, materials, personnel, and the timing of successive activities. Dantzig started from Leontief's input-output models, allowed multiple alternative activities, and gradually replaced numerous individual planning rules with a system of linear constraints. The crucial step was to add a single objective function: the goal was no longer to find a plan that merely followed the rules, but the best of all feasible plans (Dantzig, 2002, Gass, 2002).

In the summer of 1947, Dantzig sought an algorithm for the resulting general problem. His first idea was geometrically natural: move along the edges of a polyhedron from one vertex to an adjacent vertex, improving the objective function at each step. He initially rejected it because he feared that the path would be hopelessly long in many dimensions. Once he began to view the problem in terms of matrix columns and bases, however, the same procedure looked much more promising. This was the origin of the simplex method. Dantzig later openly acknowledged that his initial geometric intuition for high dimensions had been too pessimistic (Albers and Reid, 1986, Dantzig, 2002).

5.3 Meeting von Neumann: duality enters the American story

On October 3, 1947, Dantzig visited John von Neumann at the Institute for Advanced Study in Princeton. The simplex method already existed. Dantzig wanted to learn about other possible solution methods against which to compare it. In his recollection, von Neumann quickly interrupted the lengthy introduction and asked for the mathematical formulation itself. Once Dantzig wrote the geometric and algebraic forms of the problem on the board, von Neumann gave a roughly ninety-minute presentation on Farkas' lemma and duality (Albers and Reid, 1986, Cottle et al., 2007).

Von Neumann recognized an analogue of the theory of two-person zero-sum games on which he had worked earlier. Dantzig subsequently wrote a proof of duality in an internal report dated January 1948. Albert Tucker, Harold Kuhn, and David Gale further developed the theory in published form (Albers and Reid, 1986, Cottle et al., 2007). The meeting was therefore not the origin of the simplex method. It does explain why duality,

Farkas' lemma, and matrix games became inseparable parts of linear programming. We will return to all three topics later in this book.

5.4 The first computations: 120 person-days for one problem

One of the first large tests of the new method was Stigler's diet problem: combine affordable foods to meet prescribed minimum nutrient requirements at the lowest cost (Stigler, 1945). The model had 77 variables representing individual foods and nine nutritional constraints. In the fall of 1947, a team led by Jack Laderman at the U.S. National Bureau of Standards solved it using manually operated desk calculators. The computation took approximately 120 person-days (Dantzig, 1963, Gass, 2002).

The result was instructive for another reason. Stigler had previously constructed a carefully reasoned estimate with an annual cost of 39.93 USD at 1939 prices. The proven optimum at the same price level was 39.69 USD. The difference of just 24 cents showed how good Stigler's intuition had been. The simplex method added something new, however: a guarantee that no cheaper feasible combination existed (Dantzig, 1990).

Remark 1.12: An optimizer has no common sense

After joining the RAND Corporation in 1952 (Cottle et al., 2007), Dantzig also tried to use the IBM 701 computer for his own weight-loss diet. The model faithfully optimized the numbers supplied but did not understand what made a meal acceptable: the first solution called for 500 gallons of vinegar, and after vinegar was excluded, the next recommended 200 bouillon cubes a day. Dantzig and his wife gradually added upper bounds and removed unsuitable foods until Anne Dantzig ended the experiment and prescribed the menu herself (Dantzig, 1990). The anecdote captures a basic lesson of modeling: an algorithm optimizes exactly what we have written down, not what we had in mind.

5.5 From a practical workhorse to complexity theory

A conference organized by Koopmans in Chicago in 1949, later called the zeroth symposium on mathematical programming, brought together mathematicians, economists, and statisticians who had often worked independently until then. Dantzig had not known Kantorovich's work when he developed the simplex method. Only gradually did it become clear that production planning in Leningrad, American military logistics, game theory, and transportation models spoke different dialects of the same mathematical language (Dantzig, 2002, Cottle et al., 2007).

The simplex method became an exceptionally successful practical tool. In 1972, however, Klee and Minty constructed a family of problems on which the simplex method, using Dantzig's rule to choose its next step and starting at the origin, visits all 2^d vertices of a d-dimensional polyhedron (Klee and Minty, 1972). This does not mean that every simplex computation or every rule for choosing a step is slow. It shows that excellent practical performance alone does not prove good worst-case complexity.

In 1975, Kantorovich and Tjalling Koopmans jointly received the Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel for their contributions to the theory of optimal resource allocation (Royal Swedish Academy of Sciences, 1975).

The first algorithm to solve LP in time polynomial in the length of the binary encoding of rational input data was published by Leonid Khachiyan (Khachiyan, 1979). It used the ellipsoid method. Karmarkar (1984) opened a new practical direction with interior-point methods. The simplex method did not disappear. Instead, the history came full circle with another contrast: one algorithm had long excelled in practice without a polynomial guarantee, another first transformed the theory, and further methods then transformed computational practice as well.

6 What can be modeled by a linear program?

We now have the basic concepts. Let us recognize the same mathematical language in several seemingly unrelated situations. The following six problems are inspired by Chapter 2 of Matoušek (2006).

6.1 The least-cost mixture with a prescribed composition

A manufacturer prepares one kilogram of a mixture from three bulk raw materials. The mixture must contain at least 120 grams of component P and at least 100 grams of component Q. The raw materials can be divided and mixed freely, their component contents add, and processing causes no losses. What is the lowest cost of the mixture? This is the same type of model as the diet problem, in which we meet individual nutrient requirements at minimum cost (Matoušek, 2006, Section 2.1).

Data per 1 kg of raw material	A	B	C
Component P content [g]	200	100	0
Component Q content [g]	0	100	300
Price [CZK]	8	6	5

Table 1.1: Illustrative data for preparing a one-kilogram mixture.

Let x_A, x_B, x_C be the amounts of the raw materials in kilograms. The model is

$$\begin{aligned} \min \quad & 8x_A + 6x_B + 5x_C, \\ \text{subject to} \quad & x_A + x_B + x_C = 1, \\ & 200x_A + 100x_B \geq 120, \\ & 100x_B + 300x_C \geq 100, \\ & x_A, x_B, x_C \geq 0. \end{aligned}$$

The first equation enforces the total mass, and the next two constraints enforce the required composition. Notice the units: multiplying content in grams per kilogram by an amount in kilograms gives the number of grams of the component in the final mixture.

The mixture with $x_A = 0.4$, $x_B = 0.4$, and $x_C = 0.2$ meets both minimum requirements exactly and costs 6.60 CZK. No cheaper feasible mixture exists, because its cost satisfies

$$\begin{aligned} 8x_A + 6x_B + 5x_C &= 2(x_A + x_B + x_C) \\ &\quad + 0.03(200x_A + 100x_B) + 0.01(100x_B + 300x_C) \\ &\geq 2 + 0.03 \cdot 120 + 0.01 \cdot 100 = 6.6. \end{aligned}$$

Figure 1.1 shows the composition of this optimal solution.

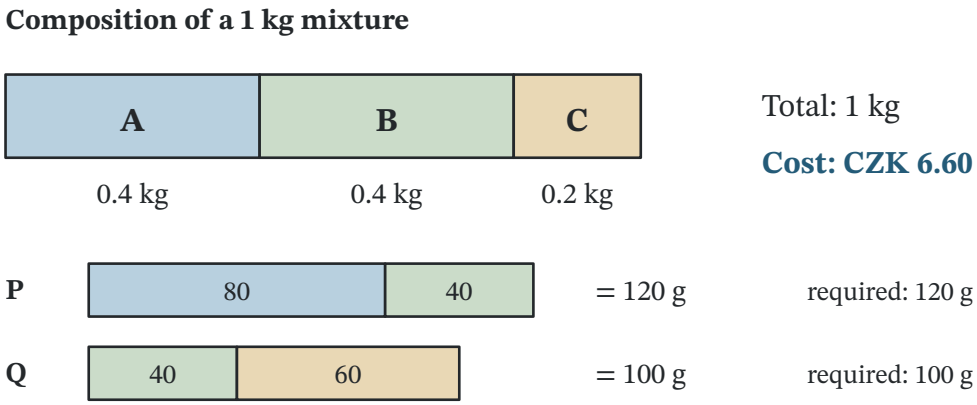


Figure 1.1: The optimal mixture and the contributions of the individual raw materials to its content of components P and Q. Segment lengths in the top bar represent the masses of the raw materials, while those in the lower bars represent the amounts of the components in grams.

This simple lower bound foreshadows duality. For a first lecture, however, another idea matters more: the model finds the cheapest mixture that meets the *stated* requirements. If the mixture must also have a particular taste or color, or a maximum proportion of raw material C, we must add the corresponding constraints.

6.2 How much data can pass through a network?

Data travel from a source s through two intermediate nodes a, b to a destination t. The five directed links have the capacities, in MB/s, shown on the left of Figure 1.2. No data are lost or accumulated at the intermediate nodes. We seek the largest steady-state transmission rate. This is a version of a network flow problem (Matoušek, 2006, Section 2.2).

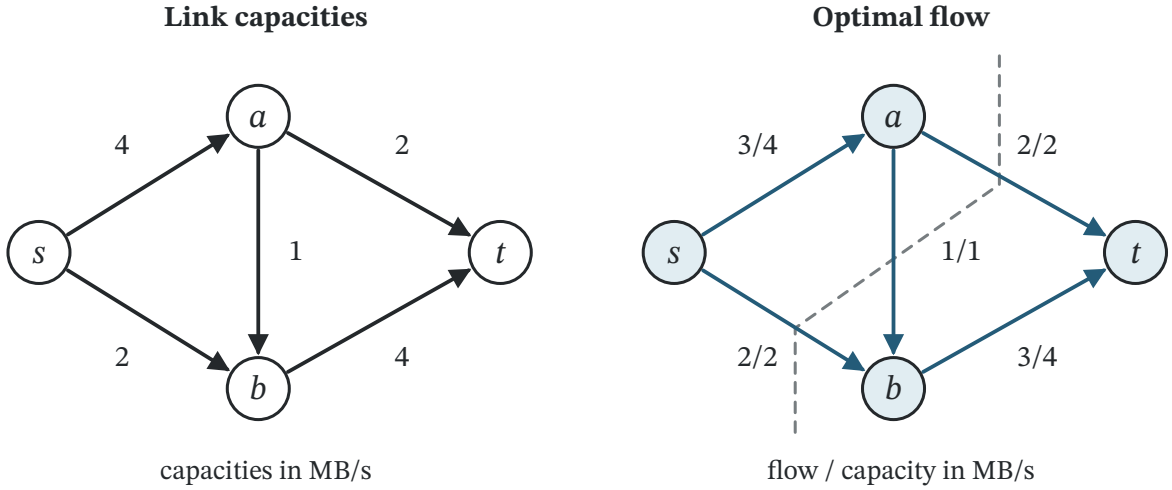


Figure 1.2: Left: capacities of the directed links. Right: an optimal flow of value 5 MB/s. The label 3/4 means a flow of 3 MB/s on a link with capacity 4 MB/s. The dashed line separates nodes s and a from nodes b and t.

The variable x_{uv} denotes the transmission rate from u to v. The model is

$$\begin{aligned} \max \quad & x_{sa} + x_{sb}, \\ \text{subject to} \quad & x_{sa} = x_{ab} + x_{at}, \\ & x_{sb} + x_{ab} = x_{bt}, \\ & 0 \le x_{sa} \le 4, \quad 0 \le x_{sb} \le 2, \quad 0 \le x_{ab} \le 1, \\ & 0 \le x_{at} \le 2, \quad 0 \le x_{bt} \le 4. \end{aligned}$$

The equalities express flow balance at nodes a and b. Adding them gives $x_{sa} + x_{sb} = x_{at} + x_{bt}$, so the rates sent and received agree.

The choice

$$(x_{sa}, x_{sb}, x_{ab}, x_{at}, x_{bt}) = (3, 2, 1, 2, 3)$$

transmits 5 MB/s. No higher rate is possible, because every feasible flow satisfies

$$x_{sa} + x_{sb} = x_{ab} + x_{at} + x_{sb} \le 1 + 2 + 2 = 5.$$

Simply adding the capacities of the links leaving the source would give 6 MB/s, an overly optimistic estimate. There are limits within the network as well. Balance equations are not specific to data transmission: the same approach describes the transport of divisible materials.

6.3 Produce to meet demand or build inventory?

A company plans production of a divisible product over $T \ge 1$ months. Demand $d_i \ge 0$ in month i is known in advance and must be met on time. Production may fluctuate, but changes cost money. Storage also has a cost. We seek the least-cost balance between steady production and low inventory, as in the ice cream example of Matoušek (2006, Section 2.3).

Let x_i be production in month i and s_i the end-of-month inventory, both in metric tons. Previous production $x_0 \ge 0$ is known. We begin and end with no inventory, so $s_0 = s_T = 0$. There are no storage losses. Carrying one metric ton into the next month costs $h > 0$, and changing monthly production by one metric ton costs $k > 0$, whether production increases or decreases. The unit production cost itself is the same in every month. Since total production must equal $\sum_i d_i$, this cost does not affect the choice of plan, so we omit it from the objective.

Using auxiliary variables u_i for the magnitude of each production change, we formulate the LP

$$\begin{aligned} \min \quad & h \sum_{i=1}^{T-1} s_i + k \sum_{i=1}^{T} u_i, \\ \text{subject to} \quad & s_{i-1} + x_i - s_i = d_i \qquad (i = 1, \dots, T), \\ & -u_i \le x_i - x_{i-1} \le u_i \quad (i = 1, \dots, T), \\ & x_i, s_i, u_i \ge 0 \qquad (i = 1, \dots, T), \\ & s_0 = s_T = 0. \end{aligned}$$

Nonnegativity of s_i prevents deliveries from being postponed. Since k is positive, every optimum must satisfy $u_i = |x_i - x_{i-1}|$. Otherwise, we could reduce u_i alone and lower the cost of the plan. In an arbitrary feasible solution, u_i is only an upper bound on this absolute value. Production or storage capacities can be added as further upper bounds.

For a small example, take $T = 2$, $(d_1, d_2) = (2, 6)$, $x_0 = 2$, $h = 1$, and $k = 2$. Measure costs in thousands of CZK. Producing exactly to meet demand, $(x_1, x_2) = (2, 6)$, incurs a production-change cost of 8. The constant plan $(4, 4)$ has inventory $s_1 = 2$, changes $(u_1, u_2) = (2, 0)$, and cost $2 + 2 \cdot 2 = 6$. It is also optimal. Setting $x = x_1$, the balance equations give $s_1 = x - 2$, $x_2 = 8 - x$, and $2 \le x \le 8$. The lowest cost for this x is

$$(x-2) + 2\big(|x-2| + |8-2x|\big) = \begin{cases} 10 - x, & 2 \le x \le 4, \\ 7x - 22, & 4 \le x \le 8. \end{cases}$$

Both branches attain their common minimum of 6 at $x = 4$. An absolute value therefore need not put a problem beyond linear programming. Figure 1.3 compares the two production plans.

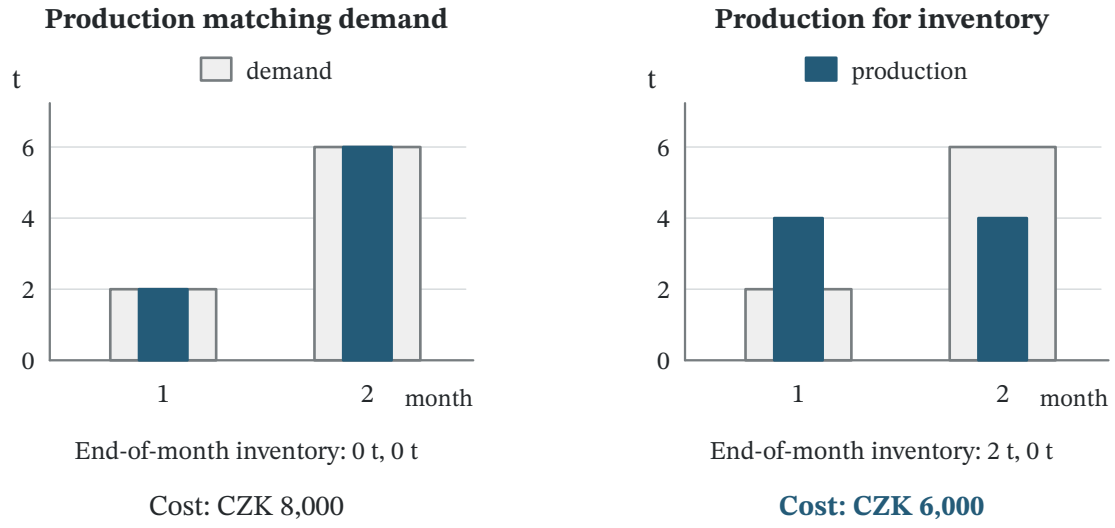


Figure 1.3: Demand and production over two months, with previous production $x_0 = 2$ t. The wide light bars show demand, and the narrow dark bars show production. The constant production plan carries two metric tons of inventory from the first month into the second.

6.4 Can two groups of points be separated by a line?

We have two nonempty finite sets of measurements in the plane. Points $\mathbf{p}_1, \ldots, \mathbf{p}_m$ represent products in one class, and points $\mathbf{q}_1, \ldots, \mathbf{q}_n$ represent products in the other. Their coordinates might record two measured properties. Is there a line that places all points of the first class on one side and all points of the second class on the other, without passing through any of them? This problem illustrates the connection between LP and data classification (Matoušek, 2006, Section 2.4).

Write the line as $\mathbf{w}^\mathsf{T}\mathbf{z} + b = 0$, where we seek the vector $\mathbf{w} = (w_1, w_2)^\mathsf{T}$ and the number b. Instead of strict inequalities, require a functional margin of at least one:

$$\begin{aligned} \min \quad & 0, \\ \text{subject to} \quad & \mathbf{w}^\mathsf{T}\mathbf{p}_i + b \ge 1 \qquad (i = 1, \ldots, m), \\ & \mathbf{w}^\mathsf{T}\mathbf{q}_j + b \le -1 \qquad (j = 1, \ldots, n), \\ & \mathbf{w} \in \mathbb{R}^2, \quad b \in \mathbb{R}. \end{aligned}$$

The objective function is constant because only feasibility matters. The measurement coordinates are data, so all constraints are linear in the unknowns w_1, w_2, b. The formulation also includes vertical lines.

Why does fixing a margin preserve every strict separation? If some $(\mathbf{w}, b)$ strictly separates all points, the minimum of the finitely many positive numbers $\mathbf{w}^\mathsf{T}\mathbf{p}_i + b$ and $-(\mathbf{w}^\mathsf{T}\mathbf{q}_j + b)$ is a number $\delta > 0$. Dividing all coefficients by δ gives the stated constraints. The reverse implication is immediate. Nonemptiness of both classes also rules out $\mathbf{w} = \mathbf{0}$. The one specifies a margin in the value of the linear expression, not a fixed geometric distance from the line.

For example, we can separate $(2, 0)$ and $(0, 2)$ from $(0, 0)$ and $(1, 0)$ by choosing $\mathbf{w} = (2, 2)^\mathsf{T}$ and $b = -3$. The separating line is $z_1 + z_2 = 3/2$. In contrast, two classes containing $(0, 0), (1, 1)$ and $(1, 0), (0, 1)$, respectively, cannot be separated in this way. Adding the first two inequalities would require $w_1 + w_2 + 2b \geq 2$, while adding the other two would require $w_1 + w_2 + 2b \leq -2$. Infeasibility of the LP has a clear meaning here: the chosen type of classification rule is insufficient for these measurements. Figure 1.4 shows both situations.

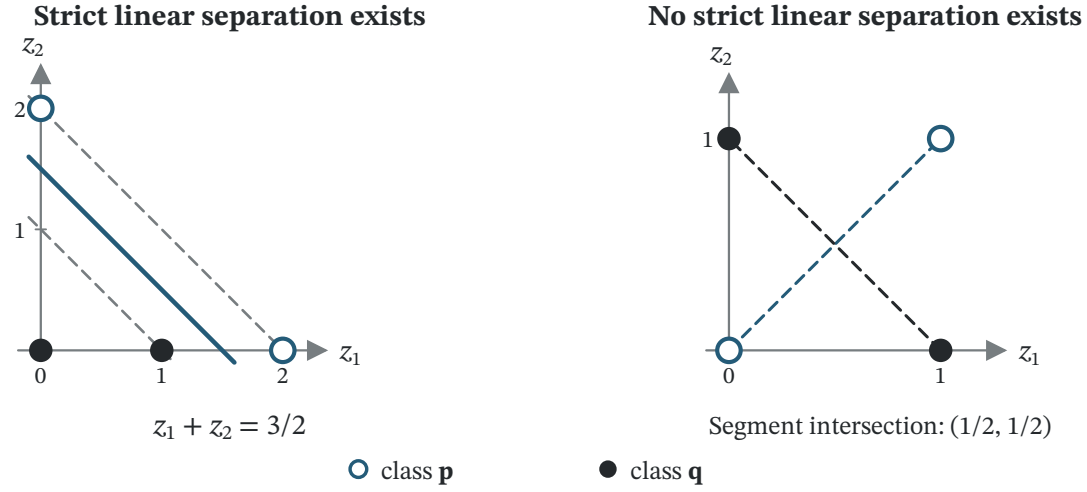


Figure 1.4: Left: the classes admit strict linear separation. The dashed lines $z_1 + z_2 = 1$ and $z_1 + z_2 = 2$ mark the functional margin for the expression $2z_1 + 2z_2 - 3$. Right: the segments joining points in the same class intersect, so strict linear separation is impossible.

6.5 A line that best fits the measurements

Given points (t_i, y_i), $i = 1, \ldots, N$, we seek a line $y = at + b$. What does *best* mean? Minimizing the sum of squared deviations gives a quadratic objective function. The sum of absolute deviations, however, can be expressed using an LP (Matoušek, 2006, Section 2.5):

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} e_i, \\ \text{subject to} \quad & -e_i \leq at_i + b - y_i \leq e_i \quad (i = 1, \ldots, N), \\ & e_i \geq 0 \qquad\qquad\qquad\quad (i = 1, \ldots, N), \\ & a, b \in \mathbb{R}. \end{aligned}$$

The values t_i are known coefficients, not variables. At an optimum, $e_i = |at_i + b - y_i|$. These are deviations in the y direction, not perpendicular distances from the points to the line.

To make the *largest* deviation as small as possible, we use a single auxiliary variable e and a different LP:

$$\min\ e \quad \text{subject to} \quad -e \le at_i + b - y_i \le e \quad (i = 1, \dots, N), \qquad a, b \in \mathbb{R},\ e \ge 0.$$

Just three measurements, $(0,0), (1,1), (2,0)$, show the difference between the two objectives. For the sum of absolute deviations, the line $y = 0$ is optimal with value 1. For the largest deviation, the line $y = 1/2$ is optimal with value $1/2$. We can verify these claims without an algorithm. The signed deviations

$$r_1 = b, \qquad r_2 = a + b - 1, \qquad r_3 = 2a + b$$

satisfy $r_1 - 2r_2 + r_3 = 2$, so

$$2 \le |r_1| + 2|r_2| + |r_3| \le 2\sum_{i=1}^{3} |r_i|, \qquad 2 \le 4 \max_{i=1,2,3} |r_i|.$$

The stated lines attain the corresponding lower bounds. The same measurements thus lead to different answers depending on how we choose to measure error (Figure 1.5).

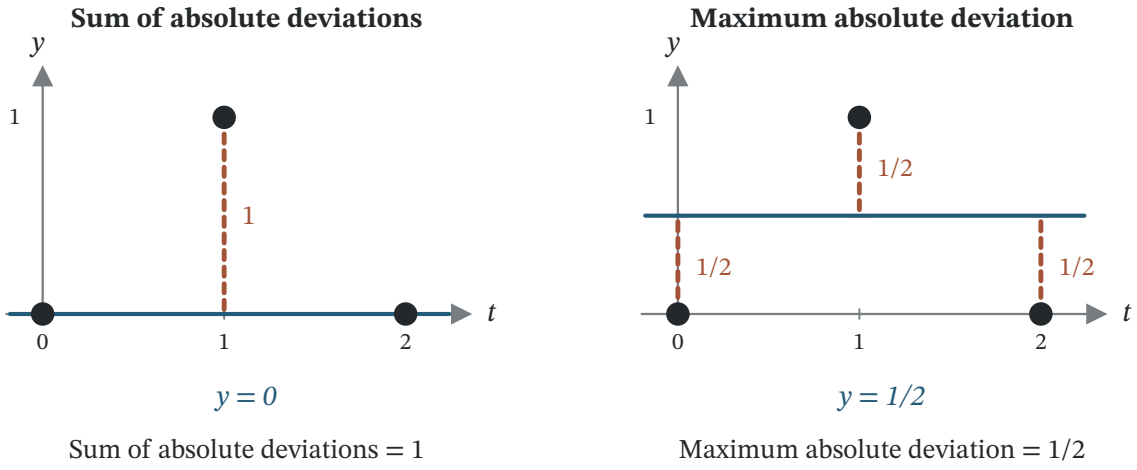


Figure 1.5: Two optimal lines for the same three measurements. The dashed segments show the nonzero absolute vertical deviations. On the left, their sum is minimized. On the right, their maximum is minimized.

6.6 The largest disk inside a polygon

We want to place the largest possible disk inside a convex polygon, for example, a circular area on a plot of land with straight boundaries. A disk is not a polygon, and its defining inequality contains squares. Even so, its center and radius can be found by solving a linear program (Matoušek, 2006, Section 2.6).

Assume that the plot is a nonempty bounded polygon with interior points, given as an intersection of half-planes:

$$P = \{\mathbf{z} \in \mathbb{R}^2 : \mathbf{a}_i^{\mathsf{T}}\mathbf{z} \le b_i,\ i = 1, \dots, m\}, \qquad \mathbf{a}_i \ne \mathbf{0}.$$

We seek a center $\mathbf{s} \in \mathbb{R}^2$ and a radius $r \geq 0$. The closed disk lies entirely in the ith half-plane if and only if

$$\mathbf{a}_i^{\mathsf{T}}\mathbf{s} + r\|\mathbf{a}_i\|_2 \leq b_i, \qquad \|\mathbf{a}_i\|_2 = \sqrt{a_{i1}^2 + a_{i2}^2}.$$

Indeed, the maximum of $\mathbf{a}_i^{\mathsf{T}}\mathbf{z}$ over the disk with center $\mathbf{s}$ and radius r is attained in the direction $\mathbf{a}_i$ and equals the left-hand side of this inequality. For $r = 0$, the statement reduces to the condition on the center alone. It therefore suffices to solve

$$\max\ r \quad \text{subject to} \quad \mathbf{a}_i^{\mathsf{T}}\mathbf{s} + r\|\mathbf{a}_i\|_2 \leq b_i \quad (i = 1, \dots, m), \qquad \mathbf{s} \in \mathbb{R}^2,\ r \geq 0.$$

The square roots are computed from the given data. The model is linear in the three decision variables s_1, s_2, r.

For the right triangle $z_1 \geq 0$, $z_2 \geq 0$, $3z_1 + 4z_2 \leq 12$, the LP has a particularly simple form:

$$\max\ r \quad \text{subject to} \quad s_1 \geq r, \quad s_2 \geq r, \quad 3s_1 + 4s_2 + 5r \leq 12, \quad r \geq 0.$$

Combining the first two constraints with the third gives $12r \leq 12$. The center $\mathbf{s} = (1, 1)^{\mathsf{T}}$ and radius $r = 1$ are feasible, so we have found the largest disk (Figure 1.6). The example shows that what matters is not the shape of the geometric object, but how the constraints depend on the quantities we actually choose.

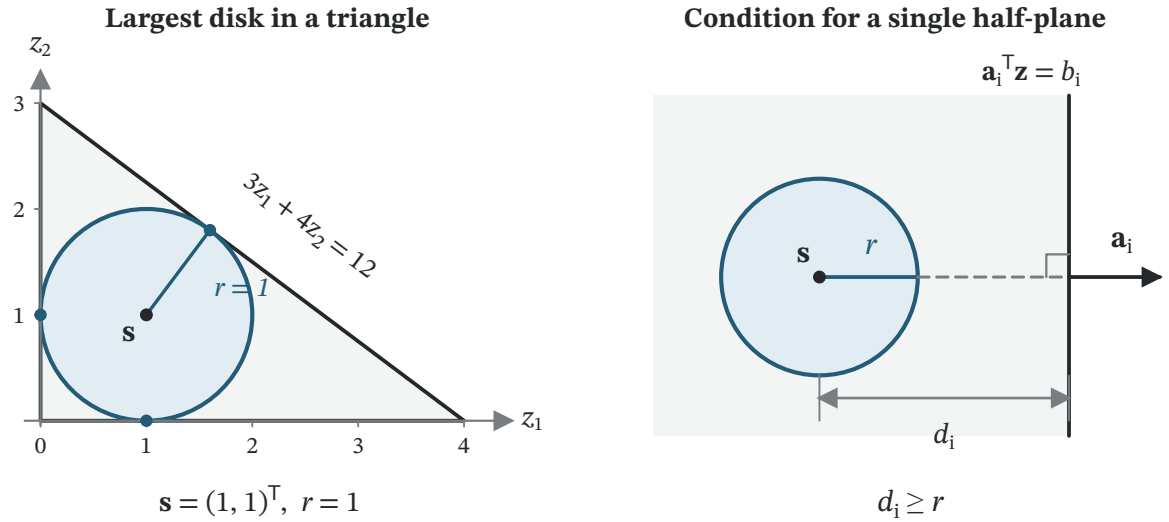
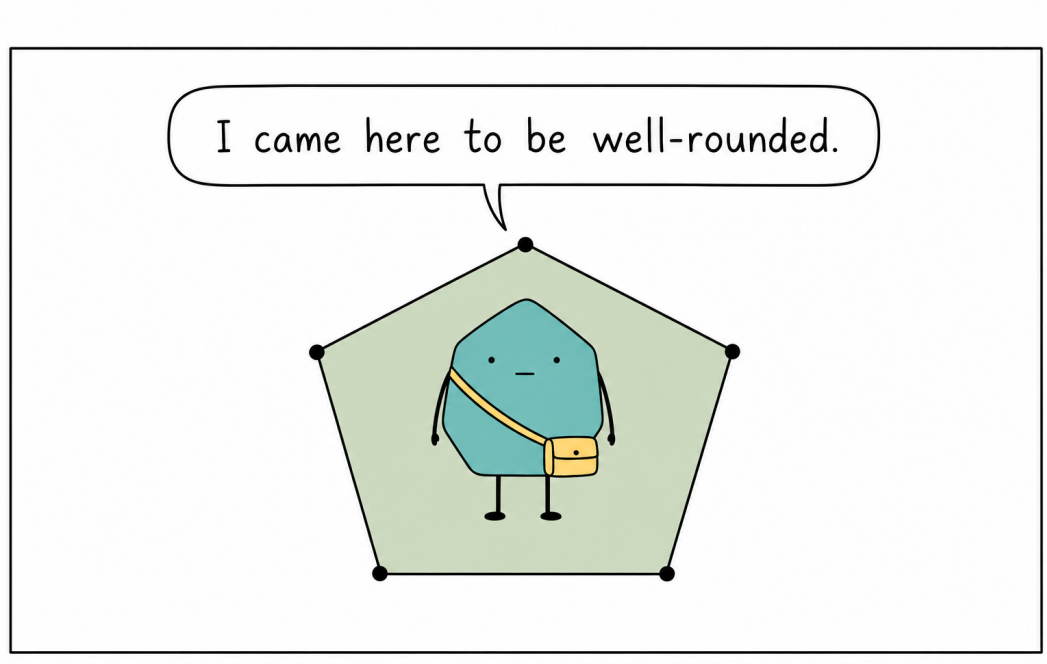


Figure 1.6: Left: the largest disk in the given triangle. Right: the condition for a single half-plane. The perpendicular distance $d_i = (b_i - \mathbf{a}_i^{\mathsf{T}}\mathbf{s})/\|\mathbf{a}_i\|_2$ must be at least r. The normal vector $\mathbf{a}_i$ points out of the half-plane.

CHAPTER 2

THE GEOMETRY OF LP: CONVEXITY AND BASIC SOLUTIONS

This chapter introduces the geometric foundations of linear programming. We define convex combinations, affine and convex hulls, halfspaces, polyhedra, and their vertices. These are more than new terms: geometry explains why it makes sense to focus on the "corners" of the feasible region when looking for an optimum. In the second part, we turn to LPs in equality form and give precise descriptions of basic feasible solutions, their support, and degeneracy.



1 An introduction to convexity

Convexity is one of the foundations of the geometry of linear programming. We review several key concepts and results that will later help us not only compute solutions to LPs but also visualize them.

1.1 Convex combinations and convex and affine hulls

Start with two points. If a set contains the entire line segment joining any two of its points, it has no "indentations." The following definition makes this property precise.

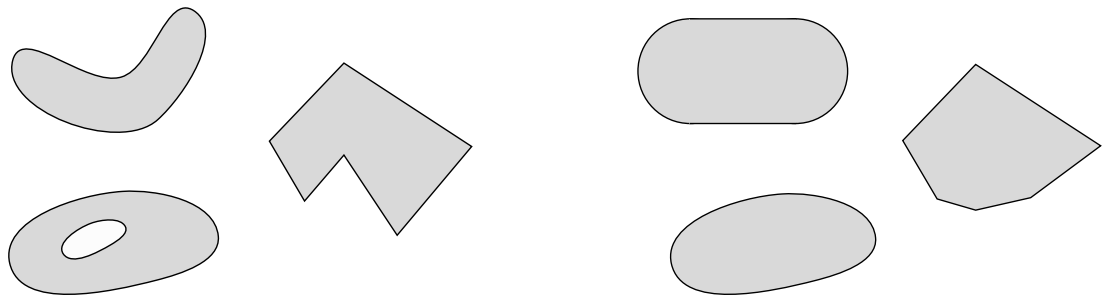
Definition 2.1: Convex combination and convex set

Let $S \subseteq \mathbb{R}^n$. A point

$$\mathbf{x} = \sum_{i=1}^{k} \lambda_i \mathbf{x}^{(i)}, \qquad \lambda_i \geq 0, \quad \sum_{i=1}^{k} \lambda_i = 1,$$

is a *convex combination* of the points $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)} \in S$. A set $C \subseteq \mathbb{R}^n$ is *convex* if $\theta\mathbf{x} + (1-\theta)\mathbf{y} \in C$ for every $\mathbf{x}, \mathbf{y} \in C$ and $\theta \in [0, 1]$.

A convex combination of two points thus traces the line segment between them. For several points, the nonnegative coefficients represent their "weights," and requiring these weights to sum to one keeps the resulting point within their convex hull.



(a) A nonconvex set. **(b)** A convex set.

Figure 2.1: The line segment between two points of a convex set stays within the set.

Definition 2.2: Affine combination and affine and convex hulls

An *affine combination* of points $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)}$ has the form $\sum_i \alpha_i \mathbf{x}^{(i)}$ with $\sum_i \alpha_i = 1$. The coefficients need not be nonnegative. The *affine hull* $\operatorname{aff}(S)$ is the set of all finite affine combinations of points in S.
The *convex hull* $\operatorname{conv}(S)$ is the intersection of all convex sets containing S. It is therefore the smallest convex set containing S.

Here, a *hull* is the smallest object of the given type that contains the original set. The convex hull fills in all the required line segments between points. The affine hull extends them to entire lines, planes, and their higher-dimensional counterparts.

Example 2.3: The convex hull of three points

If three points $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^2$ are affinely independent, their convex hull is the filled triangle with these vertices. Their affine hull is the entire plane $\mathbb{R}^2$.

A triangle is easy to visualize, but the definition applies to any set in any dimension. The following lemma states that all finite convex combinations suffice to construct the convex hull.

Lemma 2.4: Characterization of the convex hull

For every set $X \subseteq \mathbb{R}^n$,

$$\operatorname{conv}(X) = \left\{ \sum_{i=1}^{k} \lambda_i \mathbf{x}^{(i)} \;\middle|\; k \geq 1,\ \mathbf{x}^{(i)} \in X,\ \lambda_i \geq 0,\ \sum_{i=1}^{k} \lambda_i = 1 \right\}.$$

Proof of the lemma.

Denote the set on the right by D. Every convex set containing X also contains every finite convex combination of points in X. We show this by induction on the number of points. For one point the claim is immediate, and for two it is the definition of convexity. If $\mathbf{x} = \sum_{i=1}^{k} \lambda_i \mathbf{x}^{(i)}$ and $\lambda_k < 1$, set

$$\mathbf{y} = \sum_{i=1}^{k-1} \frac{\lambda_i}{1 - \lambda_k} \mathbf{x}^{(i)}.$$

The coefficients in $\mathbf{y}$ are again nonnegative and sum to one, so the induction hypothesis places $\mathbf{y}$ in the given convex set. Then $\mathbf{x} = (1 - \lambda_k)\mathbf{y} + \lambda_k \mathbf{x}^{(k)}$ belongs to the same set. The case $\lambda_k = 1$ is immediate. Hence $D \subseteq \operatorname{conv}(X)$.

It remains to check that D is convex. Let $\mathbf{x} = \sum_{i=1}^{k} \lambda_i \mathbf{x}^{(i)}$ and $\mathbf{y} = \sum_{j=1}^{\ell} \mu_j \mathbf{y}^{(j)}$ belong to D. For $\theta \in [0, 1]$,

$$\theta \mathbf{x} + (1 - \theta)\mathbf{y} = \sum_{i=1}^{k} (\theta \lambda_i) \mathbf{x}^{(i)} + \sum_{j=1}^{\ell} ((1 - \theta)\mu_j) \mathbf{y}^{(j)}.$$

All the new coefficients are nonnegative and sum to one, so this point belongs to D. The set D is convex and contains X. By the minimality of the convex hull, $\operatorname{conv}(X) \subseteq D$. ■

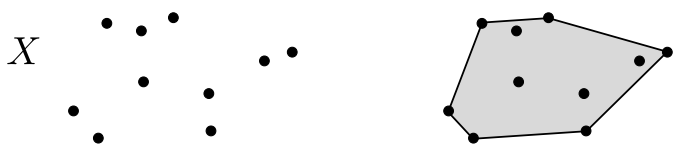


Figure 2.2: The convex hull of a finite set of points in $\mathbb{R}^2$.

1.2 Hyperplanes, halfspaces, and polyhedra

Flat objects in higher dimensions are the basic building blocks of LP. A single linear inequality defines a halfspace. Together, all the constraints carve out the feasible region.

Definition 2.5: Hyperplane and halfspace

Let $\mathbf{a} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ and $\beta \in \mathbb{R}$. The set

$$H(\mathbf{a}, \beta) = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{a}^\mathsf{T}\mathbf{x} = \beta\}$$

is a *hyperplane*. It determines two closed halfspaces, such as

$$H^-(\mathbf{a}, \beta) = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{a}^\mathsf{T}\mathbf{x} \leq \beta\}.$$

The requirement $\mathbf{a} \neq \mathbf{0}$ excludes the empty set and the entire space, which are not hyperplanes.

Every hyperplane divides space into two sides and forms the common boundary of the two corresponding closed halfspaces. This is how we will interpret the individual constraints of a linear program.

Definition 2.6: Affine set

A nonempty set $M \subseteq \mathbb{R}^n$ is *affine* if it is closed under all affine combinations. Equivalently, it has the form $M = \mathbf{x}_0 + L$, where L is a linear subspace.

Convex combinations and intersections work well together. We will use the following simple result repeatedly in later chapters.

Theorem 2.7: An intersection of convex sets is convex

The intersection of any family of convex sets in $\mathbb{R}^n$ is convex.

Proof of the theorem.

Write the intersection as $C = \bigcap_{i \in I} C_i$, and take $\mathbf{x}, \mathbf{y} \in C$ and $\theta \in [0, 1]$. For every $i \in I$, we have $\mathbf{x}, \mathbf{y} \in C_i$. The convexity of C_i gives $\theta\mathbf{x} + (1 - \theta)\mathbf{y} \in C_i$. This point belongs to every set in the family and therefore also to their intersection C. ■

This simple fact has a far-reaching consequence: an intersection of finitely many halfspaces is always convex. It gives us a geometric object that we will encounter throughout linear programming.

Definition 2.8: Convex polyhedron

A *convex polyhedron* is the solution set of a finite system of linear inequalities,

$$P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \leq \mathbf{b}\}.$$

It may be empty, bounded, or unbounded. A bounded polyhedron is called a *polytope*.

An equality can be written as two opposite inequalities, so the definition also includes sets described by both equations and inequalities. Every polyhedron is closed and convex. In particular, the term *polyhedron* does not require boundedness. The same convention is used by Matoušek (2006).

Notice that even a single halfspace is a polyhedron. A polyhedron need not resemble a closed box and may extend to infinity. When it is bounded, we call it a *polytope*.

Definition 2.9: Dimension of a polyhedron

For a nonempty polyhedron P, we define

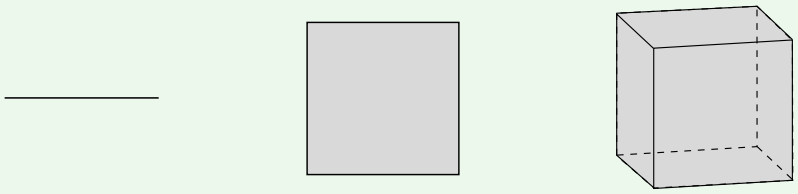
$$\dim P := \dim \operatorname{aff}(P).$$

Equivalently, $\dim P$ is the largest d for which P contains $d+1$ affinely independent points. The dimension refers to the affine hull, not necessarily to the ambient space $\mathbb{R}^n$.

The following familiar objects show how many different shapes this one definition covers. They also illustrate how to describe a geometric object by a finite system of linear equations and inequalities.

Example 2.10: Examples of convex polyhedra

- The cube $[-1,1]^n$ is described by the $2n$ inequalities $-1 \le x_i \le 1$.

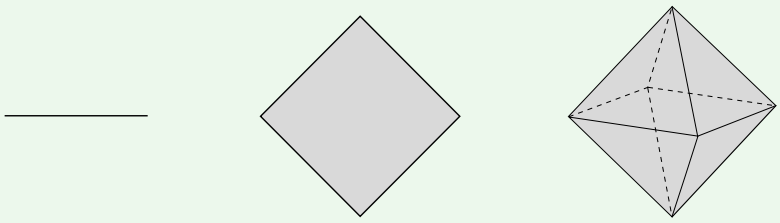


$n=1$ $n=2$ $n=3$

- The cross-polytope $\{\mathbf{x} \mid \sum_{i=1}^n |x_i| \le 1\}$ is indeed a polyhedron, since it can equivalently be described by the finite system

$$\boldsymbol{\sigma}^\mathsf{T}\mathbf{x} \le 1 \qquad \text{for all } \boldsymbol{\sigma} \in \{-1,1\}^n.$$

  For $n=3$, it is a regular octahedron.

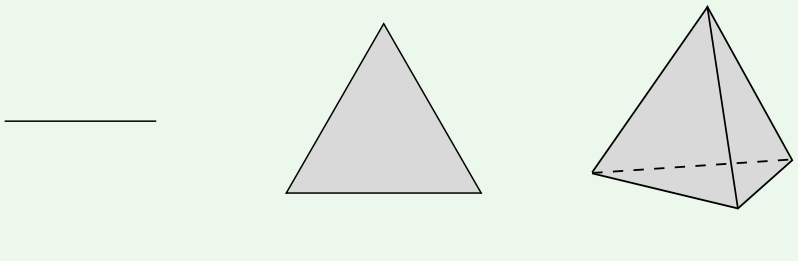


$n=1$ $n=2$ $n=3$

- The probability simplex

$$\Delta_n = \left\{ \mathbf{x} \in \mathbb{R}^{n+1} \;\middle|\; \sum_{i=1}^{n+1} x_i = 1,\ x_i \ge 0 \right\}$$

  has dimension n, and its vertices are the standard basis vectors $\mathbf{e}_1, \ldots, \mathbf{e}_{n+1}$. Notice that it is exactly the feasible set of a linear program with a single equality and nonnegativity constraints.



Vertices have a special role among the points of a polyhedron: they are its "corners" or "tips." We cannot rely on a picture alone, however, since we cannot draw one in higher dimensions. We need a definition that identifies a vertex entirely through properties of the points in the set.

Definition 2.11: Vertex of a convex polyhedron

A point $\mathbf{v} \in P$ is a *vertex* (or *extreme point*) of a convex polyhedron P if

$$\mathbf{v} = \theta\mathbf{x} + (1-\theta)\mathbf{y}, \qquad \mathbf{x}, \mathbf{y} \in P, \quad 0 < \theta < 1,$$

always implies $\mathbf{x} = \mathbf{y} = \mathbf{v}$.

A vertex thus lies in the interior of no nontrivial line segment contained in P. For polyhedra, there is an equivalent characterization: $\mathbf{v}$ is the unique maximizer of some linear function on P. This equivalence is a special property of polyhedral sets. For a general closed convex set, an extreme point need not be exposed in this way (Ziegler, 1995, Schrijver, 1986, Bertsimas and Tsitsiklis, 1997, Rockafellar, 1970).

Geometrically, we can picture the second characterization by moving a hyperplane through the polyhedron in a direction perpendicular to it. If its last contact consists of a single point, the resulting supporting hyperplane exposes a vertex. In the next chapter, this picture becomes the graphical method for solving LPs.

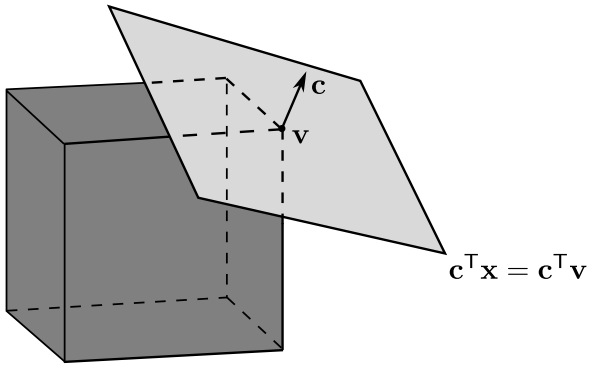


Figure 2.3: A supporting hyperplane exposing a vertex of a polyhedron.

2 Basic feasible solutions

From now on, we work with equality form. Linear algebra tells us that if the system $A\mathbf{x} = \mathbf{b}$ is consistent, its solutions form an affine set. The condition $\mathbf{x} \geq \mathbf{0}$ intersects it

with the nonnegative orthant[1]. Their intersection is the set

$$P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\},$$

where $A \in \mathbb{R}^{m \times n}$, $m \leq n$, and

$$\operatorname{rank} A = m. \tag{2.1}$$

Linearly dependent rows can be removed after checking consistency. This check matters: a dependent left-hand side with an incompatible right-hand side is evidence of infeasibility, not a redundant equation. Full row rank then guarantees that we can select m linearly independent columns of A. The set P may be empty, in which case it has no basic feasible solution.

The feasible set of an LP is a polyhedron

If the system $A\mathbf{x} = \mathbf{b}$ is consistent, its equalities define an affine set, while $\mathbf{x} \geq \mathbf{0}$ defines the nonnegative orthant. Their intersection is a convex polyhedron. If the system is inconsistent, the feasible set is empty and is therefore also a polyhedron under our definition. The same conclusion applies directly to LPs in inequality form.

Certain feasible points, called basic feasible solutions, will play a special role. Their construction follows a simple recipe: set the nonbasic components to zero and compute the basic components from a square system. We first state this recipe precisely.

Definition 2.12: Column selection and a basis

For an index set $B \subseteq \{1, \ldots, n\}$, let A_B denote the matrix formed by the corresponding columns of A, in increasing order of their indices, and let $\mathbf{x}_B$ denote the corresponding subvector. We call B a *basis* if $|B| = m$ and the square matrix A_B is nonsingular. The complement $N = \{1, \ldots, n\} \setminus B$ consists of the nonbasic indices.

We use the word *basis* as a convenient shorthand. More precisely, the columns of A_B form a basis for the column space of A.

Definition 2.13: Basic feasible solution (BFS)

A basis B determines a *basic solution*

$$\mathbf{x}_N = \mathbf{0}, \qquad \mathbf{x}_B = A_B^{-1}\mathbf{b}.$$

If $\mathbf{x}_B \geq \mathbf{0}$, this is a *basic feasible solution* (BFS), and B is a *feasible basis*. Variables with indices in B are called basic, and the remaining variables are nonbasic.

Example 2.14: A basic feasible solution

For

$$A = \begin{pmatrix} 1 & 5 & 3 & 4 & 6 \\ 0 & 1 & 3 & 5 & 6 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 14 \\ 7 \end{pmatrix}, \qquad B = \{2, 4\}$$

[1] An orthant is the higher-dimensional generalization of a quadrant in $\mathbb{R}^2$ and an octant in $\mathbb{R}^3$.

the matrix $A_B = \begin{pmatrix} 5 & 4 \\ 1 & 5 \end{pmatrix}$ is nonsingular, and $A_B^{-1}\mathbf{b} = (2,1)^\mathsf{T}$. The nonbasic indices are $N = \{1,3,5\}$, so we fill these positions with zeros to obtain

$$\mathbf{x} = (0,2,0,1,0)^\mathsf{T}.$$

All components are nonnegative, and direct substitution verifies $A\mathbf{x} = A_B\mathbf{x}_B = \mathbf{b}$. Thus we have obtained an actual BFS, rather than merely a basic candidate.

Lemma 2.15: A support criterion

For a feasible point $\mathbf{x} \in P$, let $K = \{j \mid x_j > 0\}$. The point $\mathbf{x}$ is basic if and only if the columns of A_K are linearly independent.

Proof of the lemma.

If $\mathbf{x}$ is determined by a feasible basis B, then $K \subseteq B$, so the columns of A_K are linearly independent.

Conversely, suppose that A_K has linearly independent columns. Then $|K| \leq m$. If $|K| = m$, take $B = K$. If $|K| < m$, the condition $\operatorname{rank} A = m$ allows us to extend this independent selection with further columns of A to an m-element basis B. This is the familiar linear algebra procedure for extending a linearly independent set to a basis. Since $K \subseteq B$, all components outside B are zero. The equality $A\mathbf{x} = \mathbf{b}$ then gives $A_B\mathbf{x}_B = \mathbf{b}$, so $\mathbf{x}_B = A_B^{-1}\mathbf{b}$ and $\mathbf{x}$ is a BFS. ■

Theorem 2.16: A basis determines a unique candidate

Every basis B determines exactly one basic solution. This solution is a BFS if and only if $A_B^{-1}\mathbf{b} \geq \mathbf{0}$.

Proof of the theorem.

Expand the left-hand side as $A\mathbf{x} = A_B\mathbf{x}_B + A_N\mathbf{x}_N$. Setting the nonbasic variables $\mathbf{x}_N = \mathbf{0}$ reduces the equations to $A_B\mathbf{x}_B = \mathbf{b}$. This is where nonsingularity of A_B matters: the system has the unique solution $\mathbf{x}_B = A_B^{-1}\mathbf{b}$. If this vector is nonnegative, filling the remaining positions with zeros gives exactly one BFS. If it contains a negative component, the basis yields no feasible basic solution. ■

Remark 2.17: BFSs and the objective function

Notice that the vector $\mathbf{c}$ does not appear in the definition of a BFS. Basic feasible solutions are a purely geometric property of the set P, independent of any particular objective function. Each individual basis determines at most one feasible point. Does the converse hold, so that a point can correspond to only one basis? It does not. Degeneracy allows the same point to have several different basic representations.

Definition 2.18: Degeneracy of a basic feasible solution

A BFS $\mathbf{x}$ is *nondegenerate* if it has exactly m positive components, that is, $|\{j \mid x_j > 0\}| = m$. It is *degenerate* if it has fewer than m positive components. Equivalently, at least one basic variable is zero for every corresponding basis. A problem is degenerate if it has at least one degenerate BFS.

Example 2.19: Different bases in a degenerate problem

Consider

$$A = \begin{pmatrix} 1 & 2 & 1 & 2 & 1 \\ 3 & 0 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 & 2 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 5 \\ 8 \\ 7 \end{pmatrix}, \qquad P = \{\mathbf{x} \geq \mathbf{0} \mid A\mathbf{x} = \mathbf{b}\}.$$

Let us examine four choices of basis. A single example will show the difference between a nondegenerate BFS, an infeasible basic solution, and a degenerate point with more than one basis.

Basis $B_1 = \{1, 2, 3\}$. The system $A_{B_1}\mathbf{x}_{B_1} = \mathbf{b}$ is

$$a + 2b + c = 5, \qquad 3a + 2c = 8, \qquad 2a + 2b + c = 7.$$

Its solution is $(a, b, c) = (2, 1, 1)$. Filling the nonbasic positions with zeros gives $\mathbf{x}^{(1)} = (2, 1, 1, 0, 0)^{\mathsf{T}}$. All three basic components are positive, so this is a nondegenerate BFS.

Basis $B_2 = \{2, 3, 4\}$. We now solve

$$2a + b + 2c = 5, \qquad 2b + c = 8, \qquad 2a + b + c = 7.$$

This gives $(a, b, c) = (2, 5, -2)$ and hence the basic solution $\mathbf{x}^{(2)} = (0, 2, 5, -2, 0)^{\mathsf{T}}$. The negative fourth component means that this candidate is infeasible.

Basis $B_3 = \{1, 3, 5\}$. The result is $\mathbf{x}^{(3)} = (0, 0, 3, 0, 2)^{\mathsf{T}}$. This point is feasible, but its support is only $\{3, 5\}$, of size $2 < m = 3$. The first basic variable is zero, revealing degeneracy.

Basis $B_4 = \{2, 3, 5\}$. The calculation again gives the same point:

$$\mathbf{x}^{(4)} = (0, 0, 3, 0, 2)^{\mathsf{T}} = \mathbf{x}^{(3)}.$$

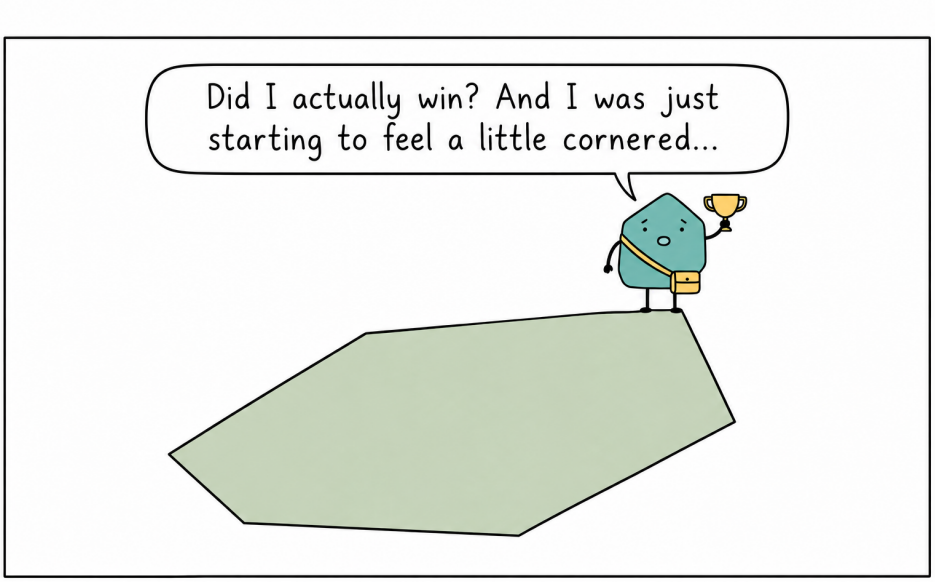
This time the variable with index two is basic and zero. We have described the same degenerate point using two different bases. The example also shows that a repeated point is a typical manifestation of degeneracy, not its definition.

For a detailed treatment of polyhedral theory, see Schrijver (1986). For its connection to the simplex method, see Bertsimas and Tsitsiklis (1997).

CHAPTER 3

THE FUNDAMENTAL THEOREM OF LP AND THE GRAPHICAL METHOD

In this chapter, we connect the algebra of basic feasible solutions with the geometry of vertices. Finding a maximum among infinitely many feasible points may seem hopeless at first. The fundamental theorem of linear programming shows, however, that when the optimum is finite, we can focus on the corners of the feasible region. We prove that at least one basic feasible solution is optimal and then illustrate this principle graphically. Along the way, we distinguish carefully between a unique optimum, an optimal face, infeasibility, and unboundedness.



1 The fundamental theorem of linear programming

The preceding chapters have supplied all the building blocks: equality form, convex polyhedra, and basic feasible solutions. We now put them together. The result turns the geometric observation that we should look for an optimum at a corner into a precise mathematical tool.

Consider the feasible set

$$P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\},$$

where $A \in \mathbb{R}^{m \times n}$ has full row rank, $\operatorname{rank} A = m$. This is the same assumption used in the definition of a BFS in Chapter 2.

We will repeatedly use the support criterion from the preceding chapter: a feasible point $\mathbf{x} \in P$ is a BFS if and only if the columns A_K indexed by

$$K = \{j \mid x_j > 0\}$$

are linearly independent.

Theorem 3.1: The fundamental theorem of linear programming

If $P \neq \varnothing$ and the linear function $\mathbf{c}^\mathsf{T}\mathbf{x}$ is bounded above on P, then

(i) the maximum is attained on P,

(ii) at least one basic feasible solution is optimal.

Proof of the theorem.

The proof uses an idea that will later reappear in a more practical form in the simplex method: move within the feasible set until a positive component decreases to zero.

We first prove an auxiliary claim: for every $\mathbf{x}^{(0)} \in P$, there is a BFS $\widehat{\mathbf{x}} \in P$ satisfying $\mathbf{c}^\mathsf{T}\widehat{\mathbf{x}} \geq \mathbf{c}^\mathsf{T}\mathbf{x}^{(0)}$.

Why does this imply the entire theorem? There are only finitely many bases, and therefore only finitely many BFSs. If every feasible point can be replaced by a BFS that is at least as good, we need only choose the best of these finitely many candidates. It remains to prove the auxiliary claim.

Among the feasible points with objective value at least $\mathbf{c}^\mathsf{T}\mathbf{x}^{(0)}$, choose a point $\widehat{\mathbf{x}}$ with the fewest positive components. Such a minimum exists because the number of positive components can take only the values $0, 1, \ldots, n$. Denote the support by

$$K = \{j \mid \widehat{x}_j > 0\}.$$

If the columns of A_K are linearly independent, the support criterion shows that $\widehat{\mathbf{x}}$ is a BFS.

Suppose, then, that the columns of A_K are dependent. We can now move away from the point without violating the equations. Indeed, there is a nonzero vector $\mathbf{w}$ whose support is contained in K such that $A\mathbf{w} = \mathbf{0}$. By changing its sign if needed, we can ensure that $\mathbf{c}^\mathsf{T}\mathbf{w} \geq 0$.

We can choose $\mathbf{w}$ to have a negative component. If $\mathbf{w} \geq \mathbf{0}$ and $\mathbf{c}^\mathsf{T}\mathbf{w} > 0$, the points $\widehat{\mathbf{x}} + t\mathbf{w}$, $t \geq 0$, would be feasible and their objective values would grow without bound, contradicting boundedness. If $\mathbf{w} \geq \mathbf{0}$ and $\mathbf{c}^\mathsf{T}\mathbf{w} = 0$, replace $\mathbf{w}$ by $-\mathbf{w}$. The objective value of the direction remains zero, and a negative component appears.

Now set

$$t^\star = \min_{j:w_j<0} \frac{\widehat{x}_j}{-w_j} > 0.$$

The point $\mathbf{x}' = \widehat{\mathbf{x}} + t^\star \mathbf{w}$ satisfies $A\mathbf{x}' = \mathbf{b}$ and $\mathbf{x}' \geq \mathbf{0}$. At least one previously positive component has become zero, no new positive component has appeared outside K, and

$$\mathbf{c}^\mathsf{T}\mathbf{x}' = \mathbf{c}^\mathsf{T}\widehat{\mathbf{x}} + t^\star \mathbf{c}^\mathsf{T}\mathbf{w} \geq \mathbf{c}^\mathsf{T}\widehat{\mathbf{x}}.$$

The ratio test thus stops the motion exactly when the first component reaches zero. We have obtained a feasible point that is at least as good and has smaller support, contradicting our choice of $\widehat{\mathbf{x}}$. The columns of A_K must therefore be independent, and $\widehat{\mathbf{x}}$ is a BFS.

There are at most $\binom{n}{m}$ bases, so there are only finitely many distinct BFSs. If we choose one with the largest objective value, the preceding argument guarantees that it is at least as good as every feasible point. This point is optimal, proving both parts of the theorem. ■

The proof did not require the feasible set to be compact. The polyhedron P may be unbounded. It is enough that the particular objective function be bounded above on it. The theorem is a standard result of polyhedral theory (Matoušek and Gärtner, 2007, Bertsimas and Tsitsiklis, 1997).

So far, the fundamental theorem speaks the algebraic language of bases. The geometric picture, however, promised corners of the polyhedron. The following result confirms that these are the same idea: vertices and basic feasible solutions are the same points described in two different languages.

Theorem 3.2: Vertices are exactly the basic feasible solutions

For $\mathbf{v} \in P$, the following conditions are equivalent:

(i) $\mathbf{v}$ is a vertex of the polyhedron P,

(ii) $\mathbf{v}$ is a basic feasible solution.

Proof of the theorem.

First, every BFS is a vertex. Let $\mathbf{v}$ be a BFS determined by a basis B, and write $N = \{1, \dots, n\} \setminus B$. Suppose that

$$\mathbf{v} = \theta\mathbf{x} + (1-\theta)\mathbf{y}, \qquad \mathbf{x}, \mathbf{y} \in P, \quad 0 < \theta < 1.$$

For $j \notin B$, we have $v_j = 0$. Since x_j, y_j are nonnegative, it follows that $x_j = y_j = 0$, and hence $\mathbf{x}_N = \mathbf{y}_N = \mathbf{0}$. The feasibility equations give $A_B\mathbf{x}_B = \mathbf{b} = A_B\mathbf{y}_B$. Nonsingularity of A_B therefore implies $\mathbf{x}_B = \mathbf{y}_B = \mathbf{v}_B$. Thus $\mathbf{v}$ is a vertex.

Next, a nonbasic point is not a vertex. Conversely, suppose that $\mathbf{v}$ is not a BFS, and let $K = \{j \mid v_j > 0\}$. By the support criterion, the columns of A_K are linearly dependent. There is therefore a nonzero $\mathbf{w}$ supported on K with $A\mathbf{w} = \mathbf{0}$. Since all v_j for $j \in K$ are positive, we can choose $\varepsilon > 0$ small enough that $\mathbf{v} \pm \varepsilon\mathbf{w} \geq \mathbf{0}$. These two points are feasible and distinct, and

$$\mathbf{v} = \tfrac{1}{2}(\mathbf{v} + \varepsilon\mathbf{w}) + \tfrac{1}{2}(\mathbf{v} - \varepsilon\mathbf{w}).$$

Thus $\mathbf{v}$ is not a vertex. ■

The proof also covers the case $n = m$. With full rank, the only possible basis is then the entire set of columns, and the system $A\mathbf{x} = \mathbf{b}$ has at most one solution. If that solution is nonnegative, it is both a BFS and a vertex.

Remark 3.3: The optimal face, multiple optima, and degeneracy

If $\mathbf{c} \neq \mathbf{0}$ and the problem has a finite optimum, imagine moving a hyperplane in the direction $\mathbf{c}$ until its last contact with the feasible set, where it becomes a supporting hyperplane. This last contact need not be a single corner. It may be an entire edge, a two-dimensional face, or even an unbounded face.
If the problem has a finite optimal value $z^\star$, the set

$$F^\star = P \cap \{\mathbf{x} \mid \mathbf{c}^\mathsf{T}\mathbf{x} = z^\star\}$$

is the *optimal face*. If $\mathbf{c} \neq \mathbf{0}$, the hyperplane in this expression is supporting: the entire polyhedron lies in the halfspace $\mathbf{c}^\mathsf{T}\mathbf{x} \leq z^\star$. If $\mathbf{c} = \mathbf{0}$, every feasible point is optimal and $F^\star = P$.
Two distinct optimal BFSs always determine a whole line segment of optimal solutions. The converse does not hold for an unbounded polyhedron. For example, the set $P = \{(x_1, x_2) \geq 0 \mid x_1 = 1\}$ has infinitely many optimal points for the objective function x_1, but only one vertex, $(1, 0)$. If the optimal face is bounded and has positive dimension, it contains at least two vertices.
Degeneracy is different from nonuniqueness of the optimum: it describes the number of positive components of a BFS relative to the rank of the constraints. A degenerate BFS may be the unique optimum, and an optimal edge may have nondegenerate endpoints.

Enumerating bases. The fundamental theorem immediately suggests a simple algorithm. It is not clever, but it clearly shows what the theorem has achieved:

1. go through all m-element sets $B \subseteq \{1, \ldots, n\}$,
2. for each, test whether A_B is nonsingular, compute $\mathbf{x}_B = A_B^{-1}\mathbf{b}$, and fill the positions outside B with zeros,
3. keep the nonnegative candidates and compare their objective values.

If we obtain no candidate, the problem is infeasible, since the preceding proof shows that a nonempty P would contain a BFS. The difficulty is the number of possibilities: $\binom{n}{m}$ can be enormous. For $n = 2m$, it grows approximately as $4^m/\sqrt{\pi m}$. A finite list of candidates does not yet give us an efficient algorithm. This is where the simplex method will later enter the picture.

Enumeration alone also fails to distinguish a finite optimum from unboundedness. An unbounded problem may have many BFSs, while its objective function grows to infinity along a ray. We must therefore also be able to detect a direction $\mathbf{d} \geq \mathbf{0}$ with $A\mathbf{d} = \mathbf{0}$ and $\mathbf{c}^\mathsf{T}\mathbf{d} > 0$.

Theorem 3.4: The fundamental theorem of LP: a complete statement

Under the assumption $\operatorname{rank} A = m$, exactly one of the following holds:

a) $P = \varnothing$,

b) $P \neq \varnothing$, and there is a $\mathbf{d} \geq \mathbf{0}$ with $A\mathbf{d} = \mathbf{0}$ and $\mathbf{c}^{\mathsf{T}}\mathbf{d} > 0$, so the problem is unbounded,

c) there is a finite optimum and at least one optimal BFS, which is also a vertex of P.

Proof of the theorem.

Case a) excludes both remaining cases. If $P \neq \varnothing$ and a direction $\mathbf{d}$ as in b) exists, then for any $\mathbf{x}^0 \in P$ and every $t \geq 0$,

$$A(\mathbf{x}^0 + t\mathbf{d}) = \mathbf{b}, \qquad \mathbf{x}^0 + t\mathbf{d} \geq \mathbf{0},$$

while $\mathbf{c}^{\mathsf{T}}(\mathbf{x}^0 + t\mathbf{d}) \longrightarrow +\infty$. Thus b) indeed means unboundedness and cannot hold together with c).

It remains to show that no other situation can occur when P is nonempty. We use the support reduction from the proof of the fundamental theorem. Start with any $\mathbf{x} \in P$. If the columns of A indexed by the positive support of $\mathbf{x}$ are dependent, there is a nonzero vector $\mathbf{w}$ supported there with $A\mathbf{w} = \mathbf{0}$. Choose its sign so that $\mathbf{c}^{\mathsf{T}}\mathbf{w} \geq 0$. If $\mathbf{w}$ has a negative component, set

$$t = \min_{j : w_j < 0} \frac{x_j}{-w_j} > 0 \qquad \text{and} \qquad \mathbf{x} \leftarrow \mathbf{x} + t\mathbf{w}.$$

The objective value does not decrease, feasibility is preserved, and the support shrinks. If $\mathbf{w}$ has no negative component and $\mathbf{c}^{\mathsf{T}}\mathbf{w} > 0$, we have found a direction as in b). If this inner product is zero, replace $\mathbf{w}$ by $-\mathbf{w}$ and again reduce the support. If, on the other hand, the columns indexed by the positive support are independent, the current point is already a BFS by the support criterion.

If b) does not occur, finitely many repetitions take us from any feasible point to a BFS with at least as large an objective value. There are finitely many bases, so the largest objective value among the BFSs bounds the objective above on all of P. The fundamental theorem then gives a finite optimum and an optimal BFS, yielding c). This proves that the alternatives are exhaustive (Bertsimas and Tsitsiklis, 1997, Schrijver, 1986). ■

2 The graphical method

In the plane, we can literally see the fundamental theorem. We draw the intersection of the halfspaces and then move a level line of the objective function in the direction of improvement as long as it still touches the feasible region. The graphical method is, of course, not a practical algorithm in higher dimensions. For building intuition, however, it is hard to find a better picture of what a linear program does.

The graphical method in $\mathbb{R}^2$

If $\mathbf{c} = \mathbf{0}$, every feasible point is optimal. Otherwise, use the following construction.

1. For each constraint, draw the boundary line and determine the correct half-plane. Their intersection is the feasible region P.
2. Draw a level line $\mathbf{c}^\mathsf{T}\mathbf{x} = \alpha$. For maximization, the value increases in the direction of the normal vector $\mathbf{c}$.
3. Move the line parallel to itself in the direction of increase. When the optimum is finite, its last contact is the optimal face. This may be a single vertex or an entire edge.
4. Verify the coordinates of the candidates by solving the corresponding equations and substituting into every constraint and the objective function.

Example 3.5: Four possible outcomes of the graphical method

We start with a very simple problem and modify it to obtain all four basic situations. Maximize $x_1 + x_2$ subject to

$$x_1, x_2 \geq 0, \qquad x_2 - x_1 \leq 1, \qquad x_1 + 6x_2 \leq 15, \qquad 4x_1 - x_2 \leq 10.$$

The intersection of the five half-planes is a convex polygon. Consider the line $x_1 + x_2 = \alpha$, perpendicular to the vector $\mathbf{c} = (1, 1)^\mathsf{T}$. As we move it in the direction of this vector, α increases. Its last contact with the feasible region occurs at the intersection of the last two boundary lines, the point $(3, 2)$. This point satisfies every constraint and has objective value 5.

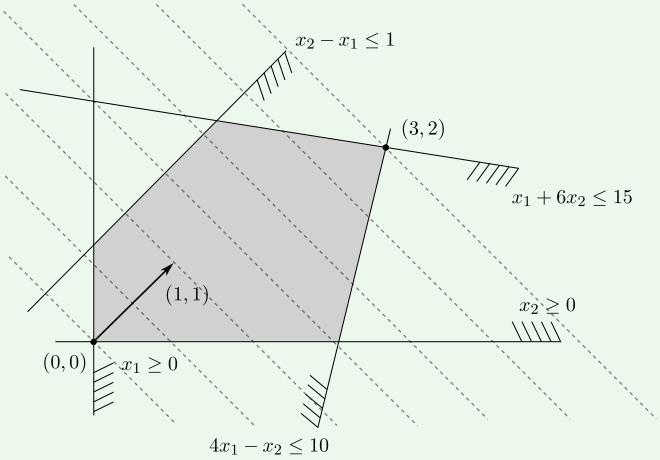


Figure 3.1: The unique optimum at $(3, 2)$.

If we change the objective function to $\frac{1}{6}x_1 + x_2$, its level lines are parallel to the edge $x_1 + 6x_2 = 15$. The entire corresponding edge is optimal because the objective function is constant on this face. Degeneracy is not the reason.

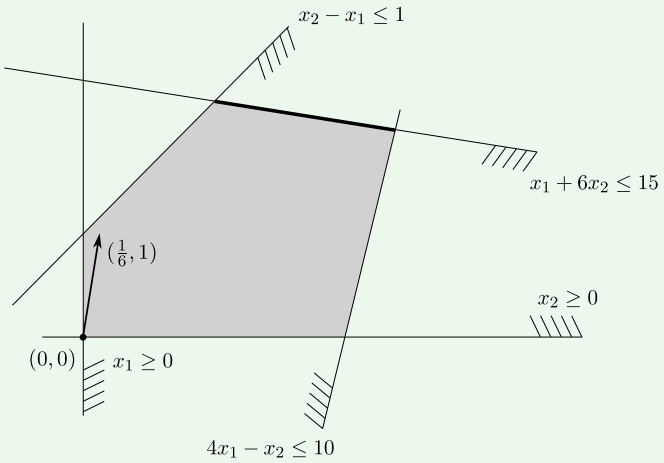


Figure 3.2: An optimal edge and infinitely many optimal points.

If we replace two of the slanted constraints by $x_2 - x_1 \geq 1$ and $4x_1 - x_2 \geq 10$ while keeping the other constraints, the intersection becomes empty. The problem is infeasible.

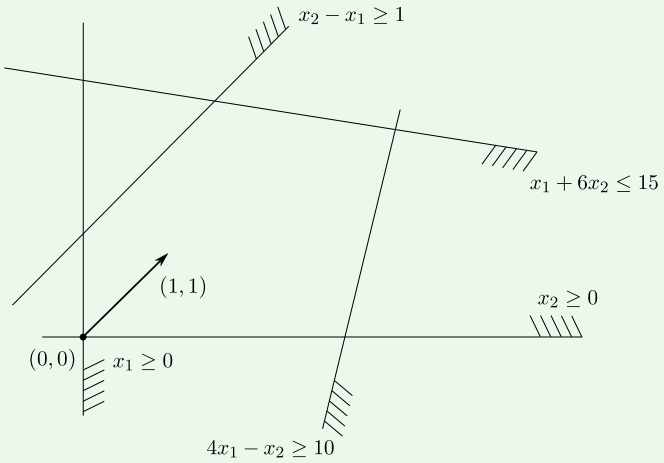


Figure 3.3: Incompatible constraints and an empty feasible set.

If instead we remove the constraints $x_1 + 6x_2 \leq 15$ and $4x_1 - x_2 \leq 10$, the ray $(x_1, x_2) = (t, t)$ for $t \geq 0$ remains feasible. The value $x_1 + x_2 = 2t$ grows without bound.

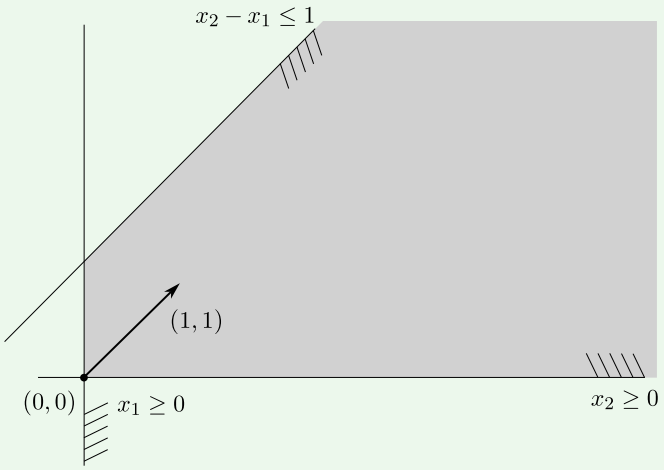


Figure 3.4: Both the feasible region and the objective function are unbounded.

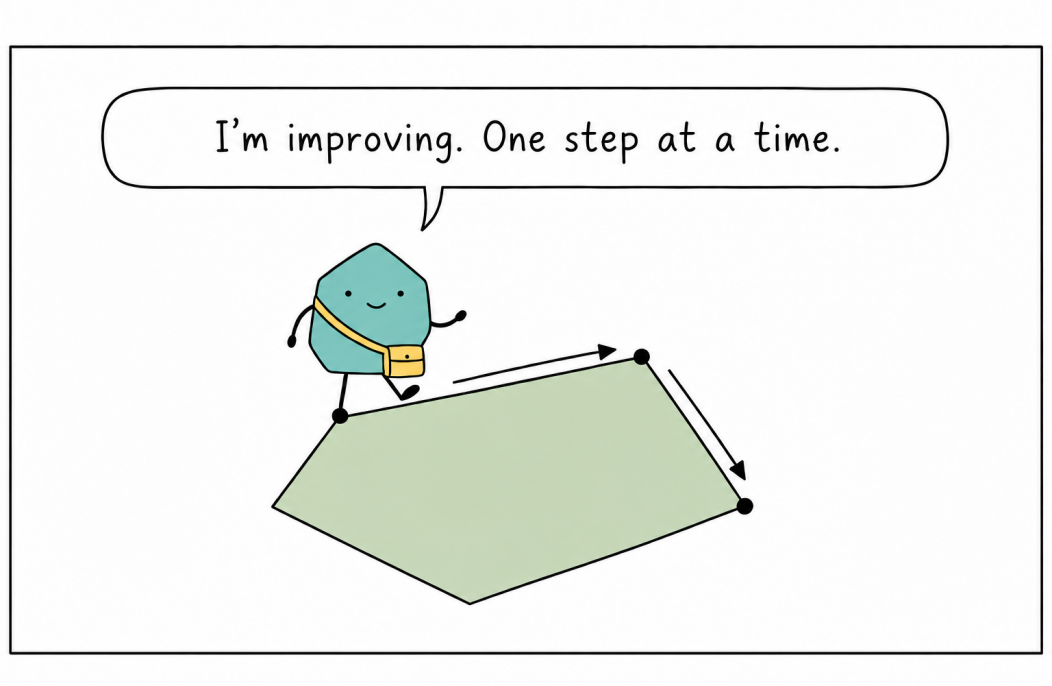
Remark 3.6: What the picture actually tells us

A bounded feasible region guarantees that every linear objective function is bounded on it. The converse is false: an unbounded polyhedron may have a finite optimum for a particular objective function. In the graphical method, we therefore consider both the shape of P and the direction $\mathbf{c}$.

CHAPTER 4

THE SIMPLEX ALGORITHM – FOUNDATIONS OF THE METHOD

This chapter introduces the computational core of linear programming: the simplex method for a maximization problem in equality form. We begin with a concrete calculation and then extract the general procedure. We derive the simplex tableau, the optimality criterion, the ratio test, and a valid pivot. Examples also show how the tableau itself reveals unboundedness or an entire line segment of optimal solutions. Degeneracy, cycling, and rules that guarantee termination are covered in Chapter 6.



1 A straightforward introductory example

George B. Dantzig introduced the simplex method in 1947 (Dantzig, 1963, 2002). Geometrically, we can view it as a path along the edges of the feasible polyhedron: at each step, we look at neighboring vertices, choose an improving direction, and move along an edge. Algebraically, we replace a single column of the basis and transform an equivalent system of equations, much as in Gaussian elimination.

This picture is useful, but it needs one important qualification. In a nondegenerate step, we really do move to a neighboring vertex and strictly improve the objective value. Under degeneracy, the basis may change without changing either the point or its objective value. The simplex method can therefore sometimes "turn in place at a vertex" before moving away from it.

We will use a problem in *maximization equality form*

$$\max\ \mathbf{c}^{\mathsf{T}}\mathbf{x} \quad \text{subject to} \quad A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0},$$

where $A \in \mathbb{R}^{m\times n}$ has linearly independent rows.

To start the method, we assume that an initial feasible basis and its associated BFS $\mathbf{x}^{[0]}$ are available. Chapter 5 explains how to find such a basis. Successive pivots generate a sequence of feasible bases and their associated points $\mathbf{x}^{[0]}, \mathbf{x}^{[1]}, \ldots$, where consecutive bases share $m-1$ indices. The corresponding vertices are adjacent when the step length is positive. For a zero step, both representations may describe the same degenerate vertex. At each step, one of the following occurs:

1. $\mathbf{x}^{[p]}$ is optimal.
2. The problem is unbounded.
3. A pivot to a new feasible basis is possible without decreasing the objective value. The value increases strictly if and only if the step length is positive.

In the following example, we first carry out the entire calculation without general formulas. Once we have seen what each step does, we will describe the simplex method in general.

Example 4.1: An introductory simplex example

We begin by solving an elementary problem in inequality form

$$\textbf{maximize} \quad x_1 + x_2 \qquad \textbf{subject to} \quad \begin{aligned} -x_1 + x_2 &\leq 1,\\ x_1 &\leq 3,\\ x_2 &\leq 2,\\ x_1, x_2 &\geq 0. \end{aligned}$$

Introducing slack variables x_3, x_4, x_5 gives the equality form

$$-x_1 + x_2 + x_3 = 1, \qquad x_1 + x_4 = 3, \qquad x_2 + x_5 = 2, \qquad x_j \geq 0 \qquad (j = 1, \ldots, 5),$$

with the objective equation $z = x_1 + x_2$. The initial *simplex tableau*, written as a

dictionary, is

$$\begin{aligned} x_3 &= 1 + x_1 - x_2, \\ x_4 &= 3 - x_1, \\ x_5 &= 2 - x_2, \\ z &= x_1 + x_2. \end{aligned}$$

For $x_1 = x_2 = 0$, the basic feasible solution is $\mathbf{x} = (0, 0, 1, 3, 2)^\mathsf{T}$ and $z = 0$.

Step 1. x_2 enters and x_3 leaves. To increase z, we choose a nonbasic variable with a positive coefficient in the last row. Both x_1 and x_2 qualify. We choose x_2. Nonnegativity of the basic variables limits the increase in x_2 through

$$x_3 = 1 - x_2 \geq 0, \quad x_4 = 3 \geq 0, \quad x_5 = 2 - x_2 \geq 0,$$

so the tightest bound is $x_2 \leq 1$. Set $x_2 = 1$ and $x_3 = 0$. Solving the first equation for x_2 gives

$$x_2 = 1 + x_1 - x_3$$

and substitution produces the new tableau

$$\begin{aligned} x_2 &= 1 + x_1 - x_3, \\ x_4 &= 3 - x_1, \\ x_5 &= 1 - x_1 + x_3, \\ z &= 1 + 2x_1 - x_3, \end{aligned}$$

with basis $\{2, 4, 5\}$ and value $z = 1$ (for $x_1 = x_3 = 0$).

Step 2. x_1 enters and x_5 leaves. The z-row has a positive coefficient for x_1, so increasing x_1 can increase z. The constraints are

$$\begin{aligned} x_2 &= 1 + x_1 && \text{(no upper bound)}, \\ x_4 &= 3 - x_1 \geq 0 && \Rightarrow x_1 \leq 3, \\ x_5 &= 1 - x_1 \geq 0 && \Rightarrow x_1 \leq 1. \end{aligned}$$

The tightest bound is $x_1 \leq 1$, so we set $x_1 = 1$ and $x_5 = 0$. Solving the third equation gives

$$x_1 = 1 + x_3 - x_5$$

and, after substitution,

$$\begin{aligned} x_1 &= 1 + x_3 - x_5, \\ x_2 &= 2 - x_5, \\ x_4 &= 2 - x_3 + x_5, \\ z &= 3 + x_3 - 2x_5. \end{aligned}$$

The basis is now $\{1, 2, 4\}$, and $x_3 = x_5 = 0$ gives $z = 3$.

Step 3. x_3 enters and x_4 leaves. The z-row allows us to increase x_3. Nonnegativity imposes a bound only through the equation

$$x_4 = 2 - x_3 + x_5 \geq 0 \Rightarrow x_3 \leq 2 + x_5,$$

as the other basic variables remain nonnegative. Set $x_5 = 0$ and $x_3 = 2$, which gives $x_4 = 0$. The third equation yields

$$x_3 = 2 + x_5 - x_4,$$

and the final tableau is

$$\begin{aligned} x_1 &= 3 - x_4, \\ x_2 &= 2 - x_5, \\ x_3 &= 2 - x_4 + x_5, \\ z &= 5 - x_4 - x_5. \end{aligned}$$

The coefficients of the nonbasic variables in the last row are nonpositive, so z cannot increase without violating feasibility. The optimal solution is

$$\mathbf{x} = (3, 2, 2, 0, 0)^{\mathsf{T}}, \qquad z^\star = 5.$$

The final tableau also provides a certificate of optimality. Every feasible solution $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_5)^{\mathsf{T}}$, with objective value $\tilde{z}$, satisfies all its equations, since these were obtained from the original problem by equivalent transformations. Thus

$$\tilde{z} = 5 - \tilde{x}_4 - \tilde{x}_5.$$

Nonnegativity of $\tilde{x}_4, \tilde{x}_5$ implies $\tilde{z} \leq 5$. The tableau also proves uniqueness: equality $z = 5$ requires $x_4 = x_5 = 0$, which uniquely determines all the other variables. Hence $\mathbf{x} = (3, 2, 2, 0, 0)^{\mathsf{T}}$ is the unique optimal solution.

Remark 4.2: Geometric illustration

The method moves from vertex to vertex along edges of the feasible polygon in the direction of increasing $x_1 + x_2$. The figure shows its path:

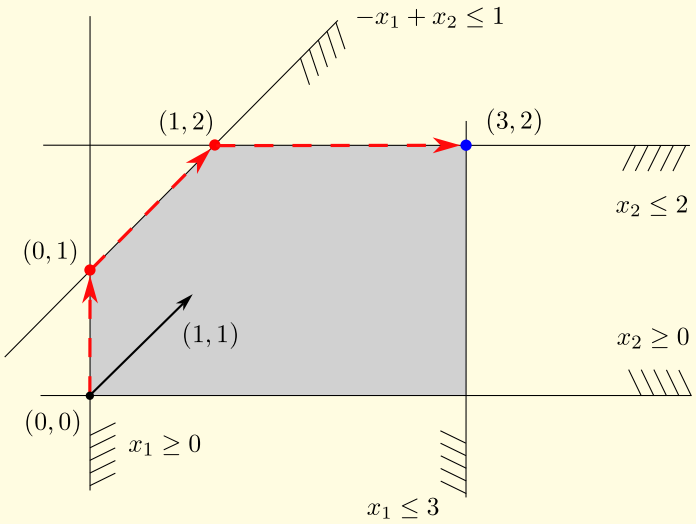


Figure 4.1: Three nondegenerate steps of the simplex method.

2 Unboundedness and multiple optimal solutions

The algorithm ran smoothly in the preceding simple example. In practice, complications can arise, and we will discuss them in detail later. First, we examine how the method behaves for an unbounded problem and how it detects multiple optimal solutions.

Example 4.3: An unbounded problem

The following example shows how the simplex method detects unboundedness:

$$\textbf{maximize} \quad x_1 \qquad \textbf{subject to} \quad \begin{aligned} x_1 - x_2 &\le 1, \\ -x_1 + x_2 &\le 2, \\ x_1, x_2 &\ge 0. \end{aligned}$$

Introducing x_3, x_4 and including the equation $z = x_1$ gives

$$\begin{aligned} x_3 &= 1 - x_1 + x_2, \\ x_4 &= 2 + x_1 - x_2, \\ z &= x_1. \end{aligned}$$

The coefficient of x_1 in the z-row is positive, so x_1 enters the basis. The first equation gives

$$x_1 = 1 + x_2 - x_3,$$

and substitution gives

$$x_4 = 3 - x_3, \qquad z = 1 + x_2 - x_3.$$

If we now try to bring x_2 into the basis, we find that no equation limits its increase: x_2 has no negative coefficient on any right-hand side. We may therefore set $x_2 = t$, $x_3 = 0$ with t arbitrarily large, obtaining

$$(x_1, x_2, x_3, x_4, z) = (1 + t,\ t,\ 0,\ 3,\ 1 + t), \qquad t \ge 0,$$

which proves unboundedness. Geometrically, this is a ray along an edge of the feasible region:

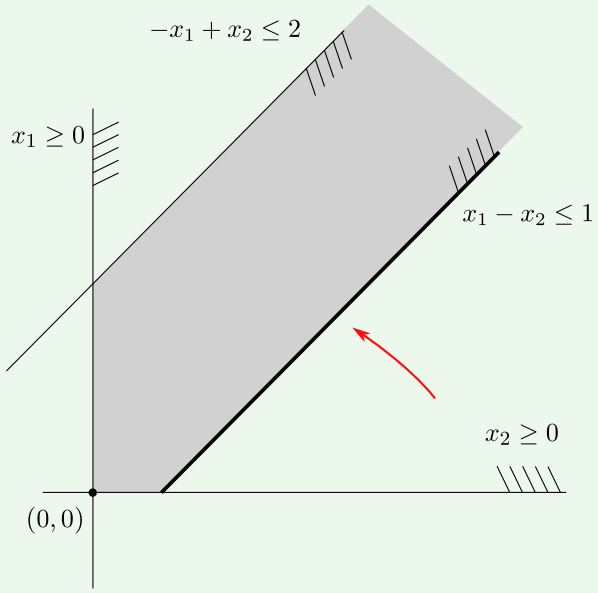


Figure 4.2: A feasible ray certifying unboundedness.

Example 4.4: The simplex method – a line segment of optimal solutions

Let us solve the following maximization problem using a simplex tableau:

$$\textbf{maximize} \quad x_1 + x_2 \qquad \textbf{subject to} \qquad \begin{aligned} x_1 + x_2 &\le 5, \\ x_1 &\le 4, \\ x_2 &\le 3, \\ x_1, x_2 &\ge 0. \end{aligned}$$

Adding slack variables x_3, x_4, x_5 gives the system

$$x_1 + x_2 + x_3 = 5, \qquad x_1 + x_4 = 4, \qquad x_2 + x_5 = 3, \qquad x_j \ge 0 \qquad (j = 1, \dots, 5),$$

and the objective function

$$z = x_1 + x_2.$$

The initial simplex tableau is

$$\begin{aligned} x_3 &= 5 - x_1 - x_2, \\ x_4 &= 4 - x_1, \\ x_5 &= 3 - x_2, \\ z &= x_1 + x_2. \end{aligned}$$

Step 1. Choose x_2 as the entering variable. The constraints are

$$x_3 = 5 - x_1 - x_2, \quad x_4 = 4 - x_1, \quad x_5 = 3 - x_2.$$

The tightest bound is $x_2 \leq 3$ (the third equation), so x_5 leaves and

$$x_2 = 3 - x_5.$$

Substituting gives

$$\begin{aligned} x_3 &= 2 - x_1 + x_5, \\ x_4 &= 4 - x_1, \\ x_2 &= 3 - x_5, \\ z &= 3 + x_1 - x_5. \end{aligned}$$

The basis is $\{2, 3, 4\}$, and $x_1 = x_5 = 0$ gives $z = 3$.

Step 2. Increase x_1. The constraints are

$$x_3 = 2 - x_1 + x_5 \geq 0, \qquad x_4 = 4 - x_1 \geq 0.$$

The tightest bound is $x_1 \leq 2$ (the first equation). x_1 enters and x_3 leaves:

$$x_1 = 2 + x_5 - x_3.$$

After substitution,

$$\begin{aligned} x_1 &= 2 + x_5 - x_3, \\ x_2 &= 3 - x_5, \\ x_4 &= 2 - x_5 + x_3, \\ z &= 5 - x_3. \end{aligned}$$

The basis is $\{1, 2, 4\}$. For $x_3 = x_5 = 0$, we have $z = 5$.

Optimal solutions. There are no positive coefficients in the z-row, and the coefficient of x_5 is zero, so we have reached an optimum. Examining the solution more closely, we see that for $x_3 = 0$ and $x_5 \in [0, 2]$,

$$(x_1, x_2) = (2 + x_5,\ 3 - x_5), \qquad z = 5,$$

so every point on the line segment from $(2, 3)$ to $(4, 1)$ attains the optimal value.

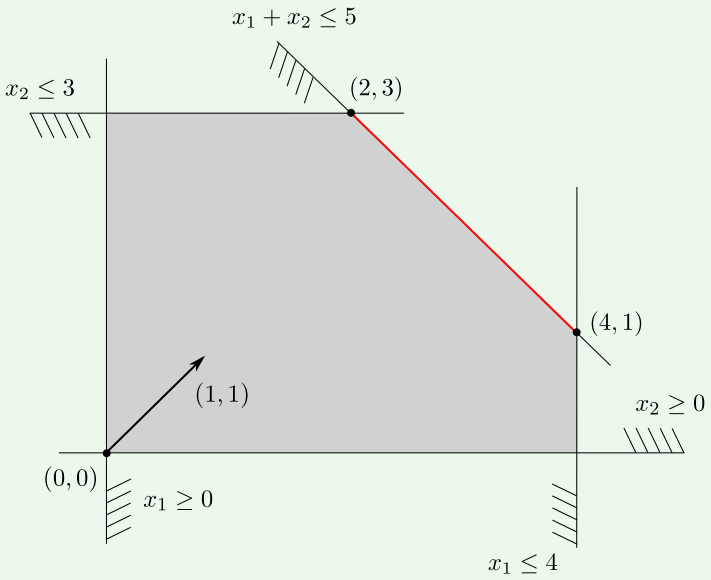


Figure 4.3: A line segment of optimal solutions.

3 Simplex tableaux

We now formulate what we have observed in the examples in general terms, using the same sign convention as in the subsequent chapters.

Consider a general maximization problem in equality form

$$\textbf{maximize} \quad \mathbf{c}^{\mathsf{T}}\mathbf{x} \qquad \textbf{subject to} \quad A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0},$$

where $A \in \mathbb{R}^{m\times n}$ has linearly independent rows and the feasible set is nonempty. For a feasible basis $B \subseteq \{1,\dots,n\}$, $|B| = m$, write $B = \{j_1 < \cdots < j_m\}$ and set $N = \{1,\dots,n\} \setminus B$.

Simplex tableau

The *simplex tableau* associated with a basis B is the system

$$\begin{aligned} \mathbf{x}_B &= \mathbf{p} + Q\,\mathbf{x}_N, \\ z &= z_0 + \mathbf{r}^{\mathsf{T}}\mathbf{x}_N, \end{aligned} \tag{4.1}$$

whose solution set in the variables $(\mathbf{x}, z)$ is identical to the solution set of $A\mathbf{x} = \mathbf{b}$ and $z = \mathbf{c}^{\mathsf{T}}\mathbf{x}$. The vector $\mathbf{x}_B$ contains the basic variables. For $N = \{1,\dots,n\} \setminus B$, $\mathbf{x}_N$ contains the nonbasic variables. The coefficient dimensions are $\mathbf{p} \in \mathbb{R}^m$, $Q \in \mathbb{R}^{m\times(n-m)}$, $\mathbf{r} \in \mathbb{R}^{n-m}$, and $z_0 \in \mathbb{R}$. For convenience, we index the coordinates of $\mathbf{x}_N$ and $\mathbf{r}$, and the columns of Q, directly by the original indices $j \in N$. Thus r_j, q_{ij}, and $Q_{\cdot j}$ denote the coordinate or column associated with the variable x_j, even though the indices in N need not be the consecutive integers $1,\dots,n-m$.

We can read the associated basic feasible solution directly from the tableau: set $\mathbf{x}_N = \mathbf{0}$, giving $\mathbf{x}_B = \mathbf{p}$ and objective value $z = z_0$. Since B is a feasible basis, $\mathbf{p} \geq \mathbf{0}$ always holds. With this sign convention, the components of $\mathbf{r}$ are called *reduced costs*.

The coefficients $\mathbf{p}, Q, \mathbf{r}$, and z_0 are easily expressed in terms of B and $A, \mathbf{b}, \mathbf{c}$. There is no need to memorize the following formulas, as they can readily be derived when needed.

Lemma 4.5: Uniqueness and form of the tableau coefficients

For every feasible basis B, there is exactly one tableau (4.1), and its coefficients are

$$Q = -A_B^{-1}A_N, \qquad \mathbf{p} = A_B^{-1}\mathbf{b}, \qquad z_0 = \mathbf{c}_B^{\mathsf{T}}A_B^{-1}\mathbf{b}, \qquad \mathbf{r} = \mathbf{c}_N - \left(\mathbf{c}_B^{\mathsf{T}}A_B^{-1}A_N\right)^{\mathsf{T}}.$$

Proof of the lemma.

The decomposition $A\mathbf{x} = A_B\mathbf{x}_B + A_N\mathbf{x}_N = \mathbf{b}$ and invertibility of A_B give

$$\mathbf{x}_B = A_B^{-1}\mathbf{b} - A_B^{-1}A_N\mathbf{x}_N.$$

This uniquely determines $\mathbf{p}$ and Q. Substituting into $z = \mathbf{c}_B^{\mathsf{T}}\mathbf{x}_B + \mathbf{c}_N^{\mathsf{T}}\mathbf{x}_N$ yields

$$z = \mathbf{c}_B^{\mathsf{T}}A_B^{-1}\mathbf{b} + \left(\mathbf{c}_N^{\mathsf{T}} - \mathbf{c}_B^{\mathsf{T}}A_B^{-1}A_N\right)\mathbf{x}_N,$$

which uniquely determines z_0 and $\mathbf{r}$. All transformations are equivalent, so the tableau describes exactly the original equations. ■

4 The simplex method – general principles

4.1 The optimality criterion

If all reduced costs in tableau (4.1) are nonpositive, that is,

$$\mathbf{r} \leq \mathbf{0},$$

then its basic feasible solution is optimal. Indeed, that solution has objective value z_0, and any other feasible solution $\tilde{\mathbf{x}}$ satisfies $\tilde{\mathbf{x}}_N \geq \mathbf{0}$ and

$$\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}} = z_0 + \mathbf{r}^\mathsf{T}\tilde{\mathbf{x}}_N \leq z_0.$$

This is a sufficient certificate of optimality. In a degenerate tableau, $\mathbf{r} \leq \mathbf{0}$ need not be necessary for the point itself to be optimal, because a positive reduced cost may allow only a zero feasible step.

4.2 A pivot – choosing the entering and leaving variables

A pivot involves two decisions. First, we choose which nonbasic variable to increase, thus choosing the direction in which to move. Next, we find which current basic variable reaches zero first, thus identifying where the boundary of the feasible region stops us.

The entering variable: choosing a direction. Choose a nonbasic variable x_v with $r_v > 0$. Its positive reduced cost tells us that any feasible positive increase in x_v improves the objective value. If all other nonbasic variables remain zero and $x_v = t \geq 0$, the tableau becomes

$$x_{j_i} = p_i + q_{iv}t \qquad (i = 1, \ldots, m), \qquad z = z_0 + r_v t.$$

The leaving variable: reaching the boundary. Since $\mathbf{p} \geq \mathbf{0}$, a row with $q_{iv} \geq 0$ imposes no bound on the increase in t. A row with $q_{iv} < 0$ requires $t \leq -p_i/q_{iv}$. If at least one such row exists, the maximum feasible step length is

$$\theta = \min_{i:q_{iv}<0} \left(-\frac{p_i}{q_{iv}} \right) \geq 0. \tag{4.2}$$

Choose as the leaving variable any basic variable x_{j_s} whose row attains this minimum. If several ratios tie, several basic variables reach zero simultaneously. Choosing one of them is part of the pivot rule. In other words, consider only negative coefficients in the entering variable's column, compute the corresponding ratios, and take the smallest. This value is the longest step that preserves nonnegativity of all basic variables. The new basis is

$$B' = (B \setminus \{j_s\}) \cup \{v\}.$$

Solve the equation for x_{j_s} for x_v and substitute it into all other rows. This is the pivot operation itself.

What if the column contains no negative coefficient? Then no basic variable stops us from increasing x_v. The improving direction extends arbitrarily far, and the tableau directly supplies a certificate of unboundedness.

Lemma 4.6: Correctness of a pivot

Let B be a feasible basis, let $r_v > 0$, and suppose that some row has $q_{iv} < 0$. If s is chosen by the ratio test (4.2), then B' is a feasible basis. The new BFS has objective value $z_0 + \theta r_v$, which is greater than z_0 for $\theta > 0$ and equal to z_0 for $\theta = 0$. If $q_{iv} \geq 0$ in every row for the chosen v, the problem is unbounded.

Proof of the lemma.

For $0 \leq t \leq \theta$, all values $p_i + q_{iv}t$ are nonnegative. At $t = \theta$, the leaving variable x_{j_s} is zero, so this point is feasible.

It remains to verify that B' is indeed a basis. Lemma 4.5 gives $Q = -A_B^{-1}A_N$, and the chosen pivot satisfies $q_{sv} \neq 0$. If the columns indexed by B' were linearly dependent, there would be a nontrivial relation

$$\alpha_v A_v + \sum_{i \neq s} \alpha_i A_{j_i} = \mathbf{0}.$$

After multiplication by A_B^{-1}, its sth component is $-\alpha_v q_{sv} = 0$, hence $\alpha_v = 0$. The remaining standard unit vectors then give $\alpha_i = 0$ for every $i \neq s$, a contradiction. Thus $A_{B'}$ is invertible. The pivoted system is equivalent to the original one, and setting the new nonbasic variables to zero gives exactly the point corresponding to $t = \theta$. Therefore B' is feasible. The objective row gives the value $z_0 + \theta r_v$.

If there is no negative q_{iv}, then for every $t \geq 0$, $\mathbf{x}_B = \mathbf{p} + tQ_{\cdot v} \geq \mathbf{0}$. These points are feasible, and $z = z_0 + r_v t \to +\infty$ because $r_v > 0$. ■

Remark 4.7: A zero in the objective row is not the whole story

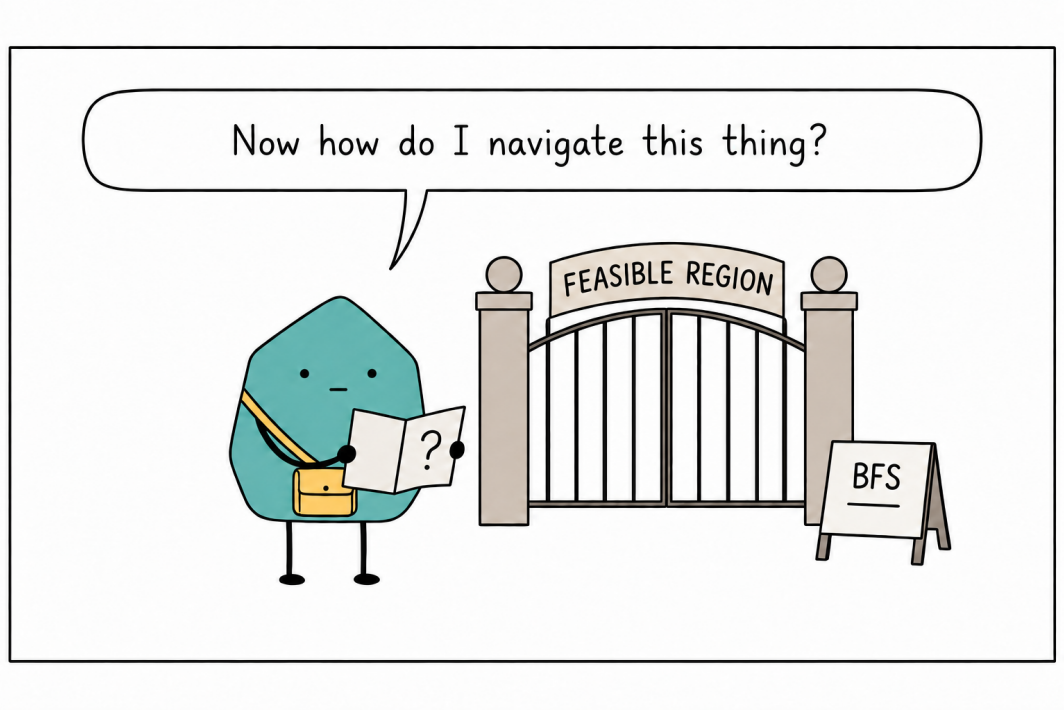
In an optimal tableau, a nonbasic variable may have $r_v = 0$. Increasing it does not change the objective value, but further conclusions depend on the feasible step length. If $\theta > 0$, we obtain distinct optimal points and, at $t = \theta$, another optimal BFS. If $\theta = 0$, the pivot may merely change the basis of the same degenerate point. If the column contains no negative q_{iv}, it gives an optimal ray. Thus $r_v = 0$ alone does not prove the existence of another optimal vertex without examining the column $Q_{\cdot v}$.

We have now justified an individual pivot. We can read an optimum, an unbounded improving direction, or another feasible basis from the tableau. One difficulty remains: zero steps might cause the method to cycle among bases without making progress. To guarantee termination even under degeneracy, we need an anti-cycling rule. This is the subject of Chapter 6. For systematic derivations of the simplex method, see Vanderbei (2020) and Bertsimas and Tsitsiklis (1997).

CHAPTER 5

THE SIMPLEX ALGORITHM II: THE TWO-PHASE AND BIG-M METHODS

Before the simplex algorithm can start, we must supply a basic feasible solution. In our earlier examples, one was available almost for free, but it may not be apparent in a general equality-form problem. This chapter presents two standard techniques: the big-M method and the two-phase method. Both use an augmented auxiliary problem that either reveals a feasible basis for the original problem or proves that none exists.



1 An initial basic feasible solution

We have seen how the simplex method works once a feasible basis is available. A careful reader may still have one troubling question: where do we get the very first basis? It is not part of the general problem statement, so we must find it ourselves.

In our examples so far, a feasible basis was available almost for free. This is the case for every inequality-form problem

$$\text{maximize } \mathbf{c}^{\mathsf{T}}\mathbf{x} \text{ subject to } A\mathbf{x} \le \mathbf{b} \text{ and } \mathbf{x} \ge \mathbf{0}$$

with $\mathbf{b} \ge \mathbf{0}$. The slack variables introduced when converting to equality form then provide a feasible basis.

In general, however, neither deciding feasibility nor finding an initial basic feasible solution is straightforward in itself. Knowing how to move from one tableau to another is therefore not enough. We must first find a starting point. We will see how to do this systematically, or how to prove along the way that the original problem has no feasible solution.

We use two standard methods (Dantzig, 1963): the big-M method and the two-phase method. Both add *artificial variables* to create an easily available initial basis. These variables are not part of the original model and must be removed before we begin optimizing the original problem itself.

First, we prove the result on which both methods rely.

Proposition 5.1: The auxiliary problem and feasibility of the original system

Let $A \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$ with $\mathbf{b} \ge \mathbf{0}$. For the system $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \ge \mathbf{0}$, consider the auxiliary problem

$$A\mathbf{x} + I\,\mathbf{w} = \mathbf{b}, \qquad \mathbf{x} \ge \mathbf{0},\ \mathbf{w} \ge \mathbf{0},$$

with objective $\min \mathbf{1}^{\mathsf{T}}\mathbf{w}$. The auxiliary problem has the feasible solution $(\mathbf{0}, \mathbf{b})$, attains its optimum, and satisfies

$$\min\bigl\{\mathbf{1}^{\mathsf{T}}\mathbf{w} : A\mathbf{x} + \mathbf{w} = \mathbf{b},\ \mathbf{x}, \mathbf{w} \ge \mathbf{0}\bigr\} = 0 \quad\Longleftrightarrow\quad \exists\, \mathbf{x} \ge \mathbf{0} :\ A\mathbf{x} = \mathbf{b}.$$

Proof of the proposition.

The objective function is bounded below by zero. If there is an $\mathbf{x} \ge \mathbf{0}$ with $A\mathbf{x} = \mathbf{b}$, then $(\mathbf{x}, \mathbf{0})$ is feasible for the auxiliary problem and attains the value zero. Conversely, if the auxiliary problem attains zero at $(\mathbf{x}^\star, \mathbf{w}^\star)$, then $\mathbf{w}^\star \ge \mathbf{0}$ and $\mathbf{1}^{\mathsf{T}}\mathbf{w}^\star = 0$ imply $\mathbf{w}^\star = \mathbf{0}$. Its equations therefore give $A\mathbf{x}^\star = \mathbf{b}$. Existence of an optimum follows from the fundamental theorem of linear programming, since the auxiliary problem is feasible and its objective is bounded below. ∎

2 The big-M method

We begin with the *big-M method*. The idea is straightforward: where the original columns do not provide a suitable initial basis, we temporarily use artificial variables. At the same time, we penalize them so heavily in the objective function that the algorithm eliminates

them whenever possible. For the original maximization problem in equality form, we solve the augmented problem

$$\max \ \mathbf{c}^{\mathsf{T}}\mathbf{x} - \mathbf{m}^{\mathsf{T}}\mathbf{w}$$

subject to

$$A\mathbf{x} + \mathbf{w} = \mathbf{b}, \qquad \mathbf{x} \geq \mathbf{0}, \ \mathbf{w} \geq \mathbf{0},$$

where $\mathbf{m} = M\mathbf{1}$ and $\mathbf{w} \in \mathbb{R}^m$ is the vector of artificial variables. This notation may suggest choosing a huge number. The mathematically safe interpretation of M, however, is *lexicographic*: first minimize $\mathbf{1}^{\mathsf{T}}\mathbf{w}$, then maximize $\mathbf{c}^{\mathsf{T}}\mathbf{x}$ among the solutions attaining that minimum. Thus M is not a large real number chosen in advance. It is a concise notation for giving the first objective absolute priority over the second.

Remark 5.2: Why not choose an enormous finite M?

The phrase "M is larger than every number in the problem" is a useful mnemonic, not a recipe for choosing a specific number. For a general problem, no rule such as "a few orders of magnitude above the largest coefficient" guarantees both the correct priority and good numerical stability without further analysis. If M is too small, a positive artificial variable may be accepted in exchange for an improvement in the original objective. If M is too large, it increases the dynamic range of the objective-row coefficients and may cause loss of accuracy through subtraction and reduced-cost calculations. The value of M does not, however, change the conditioning of the basis matrix A_B itself. Therefore:

- in hand calculations and theoretical work, we treat M symbolically and compare its coefficients and the constant parts separately,
- in numerical implementations, we prefer the two-phase method, which requires no large constant.

There is no need to add an artificial variable to every row. If an original or slack column already supplies the required unit column of the initial basis, we use it. We add w_i only to a row i that lacks a suitable basic column. The index of an artificial variable thus identifies a *constraint row*, not a column of the original matrix.

Once all artificial variables are zero and nonbasic, we have a basic feasible solution of the original system. We delete their columns, reconstruct the original objective row $z = \mathbf{c}^{\mathsf{T}}\mathbf{x}$ for the current basis, and continue with the ordinary simplex method. The reduced-cost signs of the deleted artificial variables impose no additional condition on this transition.

If lexicographic optimization terminates with some artificial variable positive, the original problem is infeasible. If an artificial variable remains basic at zero, we proceed as at the end of Phase I of the two-phase method: either pivot it out using an original column with a nonzero coefficient in its row, or delete the redundant row.

Remark 5.3: Why the big-M method works

Consider the augmented problem with objective

$$\tilde{z} = \mathbf{c}^{\mathsf{T}}\mathbf{x} - \mathbf{m}^{\mathsf{T}}\mathbf{w} \quad \text{subject to} \quad A\mathbf{x} + \mathbf{w} = \mathbf{b}, \quad \mathbf{x}, \mathbf{w} \geq \mathbf{0},$$

where $\mathbf{m}$ has the meaning described above. The part of the objective containing M dominates. Maximizing $\tilde{z}$ is therefore lexicographically[a] equivalent to these two steps:

1. minimize $\sum_j w_j$,
2. maximize $\mathbf{c}^{\mathsf{T}}\mathbf{x}$ among solutions attaining the minimum of $\sum_j w_j$.

Proposition 5.1 shows that $\min \sum_j w_j = 0$ if and only if some $\mathbf{x} \geq \mathbf{0}$ satisfies $A\mathbf{x} = \mathbf{b}$. If any $w_j > 0$ remains at the lexicographic optimum, the original problem is infeasible.

[a] *Lexicographic order*: for vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^k$, we have $\mathbf{u} <_{\text{lex}} \mathbf{v}$ if there is a first index i with $u_i \neq v_i$ and $u_i < v_i$. Maximizing a vector-valued function $(f_1, \ldots, f_k)$ lexicographically means first maximizing f_1, then maximizing f_2 among solutions with the same optimal f_1, and so on.

2.1 Examples

The first example follows the usual course of the big-M method and actually removes the artificial variables from the basis. The second is deliberately simple, so that we can clearly see how a positive artificial variable reveals infeasibility of the original problem.

Example 5.4: The big-M method

In the introductory example, we wish to

$$\textbf{maximize} \quad z = x_1 + 4x_2 + 2x_3 \qquad \textbf{subject to} \quad \begin{aligned} x_1 - x_2 + x_3 &= 1, \\ x_1 + x_2 + 2x_3 &= 4, \\ x_1, x_2, x_3 &\geq 0. \end{aligned}$$

Adding artificial variables w_1, w_2 gives the equality form

$$x_1 - x_2 + x_3 + w_1 = 1, \qquad x_1 + x_2 + 2x_3 + w_2 = 4, \qquad x_1, x_2, x_3, w_1, w_2 \geq 0,$$

and objective equation

$$\tilde{z} = x_1 + 4x_2 + 2x_3 - M(w_1 + w_2),$$

where M is a lexicographically dominant symbol.

The initial simplex tableau. Choose the artificial variables as the initial basic variables:

$$\begin{aligned} w_1 &= 1 - x_1 + x_2 - x_3, \\ w_2 &= 4 - x_1 - x_2 - 2x_3, \\ \tilde{z} &= -5M + (2M+1)x_1 + 4x_2 + (3M+2)x_3. \end{aligned}$$

We now pivot to make the artificial variables nonbasic and reduce their sum to zero.

Step 1. x_3 **enters and** w_1 **leaves.**

$$\begin{aligned} x_3 &= 1 - x_1 + x_2 - w_1, \\ w_2 &= 2 + x_1 - 3x_2 + 2w_1, \\ \tilde{z} &= 2 - 2M + (-M-1)x_1 + (3M+6)x_2 - (3M+2)w_1. \end{aligned}$$

Step 2. x_2 **enters and** w_2 **leaves.**

$$\begin{aligned} x_2 &= \tfrac{2}{3} + \tfrac{1}{3}x_1 + \tfrac{2}{3}w_1 - \tfrac{1}{3}w_2, \\ x_3 &= \tfrac{5}{3} - \tfrac{2}{3}x_1 - \tfrac{1}{3}w_1 - \tfrac{1}{3}w_2, \\ \tilde{z} &= 6 + x_1 + (2-M)w_1 - (M+2)w_2. \end{aligned}$$

Both artificial variables are now nonbasic and zero. We therefore delete their columns and reconstruct the original objective row for the current basis.

A basic feasible solution of the original problem.

$$\mathbf{x} = (0, \tfrac{2}{3}, \tfrac{5}{3})^{\mathsf{T}}, \qquad z = 6.$$

Continuing with the simplex algorithm. After removing $\mathbf{w}$, the starting tableau is

$$x_2 = \tfrac{2}{3} + \tfrac{1}{3}x_1, \quad x_3 = \tfrac{5}{3} - \tfrac{2}{3}x_1, \quad z = 6 + x_1.$$

x_1 enters and x_3 leaves. After the pivot,

$$x_1 = \tfrac{5}{2} - \tfrac{3}{2}x_3, \quad x_2 = \tfrac{3}{2} - \tfrac{1}{2}x_3, \quad z = \tfrac{17}{2} - \tfrac{3}{2}x_3.$$

The objective row certifies optimality. Setting $x_3 = 0$ gives

$$\mathbf{x}^\star = (\tfrac{5}{2}, \tfrac{3}{2}, 0)^{\mathsf{T}}, \qquad z^\star = \tfrac{17}{2}.$$

Example 5.5: The big-M method – an infeasible problem

In this example, we wish to

$$\textbf{maximize} \quad z = x_1 + x_2 \qquad \textbf{subject to} \quad \begin{aligned} x_1 + x_2 &= 1, \\ x_1 + x_2 &= 2, \\ x_1, x_2 &\geq 0. \end{aligned}$$

Adding w_1, w_2 gives

$$x_1 + x_2 + w_1 = 1, \qquad x_1 + x_2 + w_2 = 2, \qquad x_1, x_2, w_1, w_2 \geq 0,$$

with objective

$$\tilde{z} = x_1 + x_2 - M(w_1 + w_2),$$

where M is again lexicographically dominant.

The initial simplex tableau.

$$w_1 = 1 - x_1 - x_2, \qquad w_2 = 2 - x_1 - x_2, \qquad \tilde{z} = -3M + (2M+1)(x_1 + x_2).$$

Step 1. x_1 **enters and** w_1 **leaves.**

$$x_1 = 1 - x_2 - w_1, \qquad w_2 = 1 + w_1, \qquad \tilde{z} = -M + 1 - (2M+1)w_1 + 0 \cdot x_2.$$

There are no positive coefficients in the $\tilde{z}$-row, so we have found an optimum of the auxiliary problem at $w_1 = 0$, $x_2 = 0$:

$$x_1 = 1, \quad x_2 = 0, \quad w_1 = 0, \quad w_2 = 1, \quad \tilde{z} = -M + 1.$$

Since $w_2 > 0$ at the optimum, the original problem is **infeasible**.

A quick diagnostic.

$$(x_1 + x_2 + w_2) - (x_1 + x_2 + w_1) = 1 \Rightarrow w_2 - w_1 = 1,$$

so $\min(w_1 + w_2) = 1$, and $w = 0$ cannot be reached.

3 The two-phase method

The second way to find an initial basis is the *two-phase method*. It follows the same idea as the big-M method but keeps the two priorities in separate objectives. We augment the original equality-form problem to obtain an initial BFS of the auxiliary problem easily. Its optimum tells us whether the original system is feasible and, if so, supplies a basis for optimizing the original objective.

As its name suggests, the procedure has two phases. In Phase I, we set aside the original profit: our only goal is to minimize the sum of the artificial variables, reducing it to zero if possible. In Phase II, we remove the artificial variables, restore the original objective function, and continue the simplex method from the feasible basis we have found.

Phase I is a direct application of Proposition 5.1.

3.1 Phase I

First, convert the problem to equality form. Multiply any row with a negative right-hand side by -1, so that $\mathbf{b} \geq \mathbf{0}$. Add artificial variables to rows that lack a suitable basic column. For simplicity of notation, suppose for now that every row needs one:

$$A\mathbf{x} + I\,\mathbf{w} = \mathbf{b}, \qquad \mathbf{x} \geq \mathbf{0},\ \mathbf{w} \geq \mathbf{0}.$$

Then solve the *auxiliary maximization problem* by the simplex method:

$$\max\ \left(-w_1 - \cdots - w_m\right) \quad \text{subject to} \quad A\mathbf{x} + I\,\mathbf{w} = \mathbf{b}, \quad \mathbf{x}, \mathbf{w} \geq \mathbf{0}.$$

An initial BFS for Phase I is immediate: $\mathbf{x} = \mathbf{0}$, $\mathbf{w} = \mathbf{b}$.

Depending on the Phase I optimum, three situations can occur:

1. $\sum_{i=1}^{m} w_i^\star > 0$, or equivalently the auxiliary maximization problem has a negative optimal value. By Proposition 5.1, the original problem is **infeasible**.

2. $\sum_{i=1}^{m} w_i^\star = 0$ and all artificial variables are nonbasic. Then $\mathbf{w}^\star = \mathbf{0}$, and the basic point in the original variables is an initial BFS for Phase II.

3. $\sum_{i=1}^{m} w_i^\star = 0$, but some artificial variable w_i remains basic at zero. In its row, find a nonbasic nonartificial variable with a nonzero coefficient and perform a zero-step pivot to remove w_i from the basis. If no such coefficient exists, the corresponding equation is linearly dependent on the others. Delete that row and the column of w_i. Repeat until no artificial variable remains basic.

3.2 Phase II

If the Phase I optimum is zero, adjust the basis as described above, delete all artificial-variable columns, restore the original objective $\mathbf{c}^{\mathsf{T}}\mathbf{x}$, and recompute its row for the current basis. Then continue with the ordinary simplex method. Phase I guarantees only feasibility: in Phase II, the original problem may have a finite optimum or turn out to be unbounded.

3.3 Examples

The following three examples illustrate all three possible outcomes at the end of Phase I.

Example 5.6: The two-phase method

We illustrate the two-phase method with the simple problem

$$\textbf{maximize} \quad z = x_1 + 4x_2 + 2x_3 \qquad \textbf{subject to} \quad \begin{aligned} x_1 - x_2 + x_3 &= 1,\\ x_1 + x_2 + 2x_3 &= 4,\\ x_1, x_2, x_3 &\geq 0. \end{aligned}$$

Phase I. Add artificial variables w_1, w_2 and solve

$\max(-w_1 - w_2)$ subject to $x_1 - x_2 + x_3 + w_1 = 1$, $x_1 + x_2 + 2x_3 + w_2 = 4$, $\mathbf{x}, \mathbf{w} \geq \mathbf{0}$.

Write $\tilde{z} = -w_1 - w_2$.

The initial tableau.

$$\begin{aligned} w_1 &= 1 - x_1 + x_2 - x_3,\\ w_2 &= 4 - x_1 - x_2 - 2x_3,\\ \tilde{z} &= -5 + 2x_1 + 3x_3. \end{aligned}$$

Step 1. x_3 **enters and** w_1 **leaves.**

$$\begin{aligned} x_3 &= 1 - x_1 + x_2 - w_1,\\ w_2 &= 2 + x_1 - 3x_2 + 2w_1,\\ \tilde{z} &= -2 - x_1 + 3x_2 - 3w_1. \end{aligned}$$

Step 2. x_2 **enters and** w_2 **leaves.**

$$\begin{aligned} x_2 &= \tfrac{2}{3} + \tfrac{1}{3}x_1 + \tfrac{2}{3}w_1 - \tfrac{1}{3}w_2,\\ x_3 &= \tfrac{5}{3} - \tfrac{2}{3}x_1 - \tfrac{1}{3}w_1 - \tfrac{1}{3}w_2,\\ \tilde{z} &= -w_1 - w_2. \end{aligned}$$

The Phase I optimum is $\tilde{z}^\star = 0$ at $w_1 = w_2 = 0$. This gives the BFS

$$\mathbf{x}^{(0)} = (0, \tfrac{2}{3}, \tfrac{5}{3})^\mathsf{T}.$$

Phase II. Delete the columns of $\mathbf{w}$ and solve the original problem

$$z = x_1 + 4x_2 + 2x_3 = 6 + x_1 \quad \text{subject to} \quad \begin{aligned} x_2 &= \tfrac{2}{3} + \tfrac{1}{3}x_1, \\ x_3 &= \tfrac{5}{3} - \tfrac{2}{3}x_1. \end{aligned}$$

x_1 enters and x_3 leaves. Pivoting and setting $x_3 = 0$ gives

$$\mathbf{x}^\star = (\tfrac{5}{2}, \tfrac{3}{2}, 0)^\mathsf{T}, \qquad z^\star = \tfrac{17}{2}.$$

Example 5.7: Detecting infeasibility with the two-phase method

To recall the general procedure, we start with an inequality-form problem:

$$\textbf{maximize} \quad z = 3x_1 + 5x_2 \qquad \textbf{subject to} \quad \begin{aligned} x_1 - 2x_2 &\le 6, \\ 2x_1 + \tfrac{1}{2}x_2 &\le 7, \\ x_1 + x_2 &\ge 16, \\ x_1, x_2 &\ge 0. \end{aligned}$$

Conversion to equalities and Phase I.

$$\begin{aligned} x_1 - 2x_2 + x_3 &= 6, \\ 2x_1 + \tfrac{1}{2}x_2 + x_4 &= 7, \\ x_1 + x_2 - x_5 + w &= 16, \\ x_1, x_2, x_3, x_4, x_5, w &\ge 0. \end{aligned} \qquad \max \tilde{z} = \max(-w).$$

The initial basic variables are x_3, x_4, w:

$$x_3 = 6 - x_1 + 2x_2, \quad x_4 = 7 - 2x_1 - \tfrac{1}{2}x_2, \quad w = 16 - x_1 - x_2 + x_5.$$

The smallest possible w is obtained with $x_5 = 0$. The second equation gives

$$2x_1 + \tfrac{1}{2}x_2 \le 7 \;\Rightarrow\; \tfrac{1}{2}(x_1 + x_2) \le 2x_1 + \tfrac{1}{2}x_2 \le 7 \;\Rightarrow\; x_1 + x_2 \le 14.$$

Hence

$$w \ge 16 - (x_1 + x_2) \ge 2, \quad \Rightarrow \quad \tilde{z} \le -2.$$

The bound is attained, for example, at $x_1 = 0$, $x_2 = 14$, $x_5 = 0$, where the remaining basic variables are nonnegative and $w = 2$. Thus $\tilde{z}^\star = -2 < 0$.
The Phase I optimal value is negative, so $w^\star > 0$. The original problem is **infeasible**.

Example 5.8: The two-phase method – a redundant constraint

Consider the problem

$$\textbf{maximize} \quad z = x_1 + 2x_2 + x_3 \qquad \textbf{subject to} \quad \begin{aligned} 3x_1 + x_2 - x_3 &= 15,\\ 8x_1 + 4x_2 - x_3 &= 50,\\ 2x_1 + 2x_2 + x_3 &= 20,\\ x_1, x_2, x_3 &\geq 0. \end{aligned}$$

Phase I. Add w_1, w_2, w_3 and solve $\max -(w_1 + w_2 + w_3)$:

$$\begin{aligned} 3x_1 + x_2 - x_3 + w_1 &= 15,\\ 8x_1 + 4x_2 - x_3 + w_2 &= 50,\\ 2x_1 + 2x_2 + x_3 + w_3 &= 20. \end{aligned}$$

We have

$$\tilde{z} = -w_1 - w_2 - w_3 = -85 + 13x_1 + 7x_2 - x_3.$$

x_1 enters and w_1 leaves. After elimination, the coefficient of x_2 in the $\tilde{z}$-row is positive. The second ratio test has a tie between the rows of w_2 and w_3. Choose w_2 to leave. After this pivot, $\tilde{z}^\star = 0$, while w_3 remains basic at zero and its row contains no original variable with a nonzero coefficient. This is a **redundant constraint**. Delete the row and the column of w_3, then enter Phase II with two equations.

Checking redundancy directly. The third equation is a linear combination of the first two:

$$(2) \;-\; 2\cdot(1): \qquad \begin{aligned} (8x_1 + 4x_2 - x_3) - 2(3x_1 + x_2 - x_3) &= 2x_1 + 2x_2 + x_3,\\ 50 - 2\cdot 15 &= 20. \end{aligned}$$

The constraint is indeed redundant.

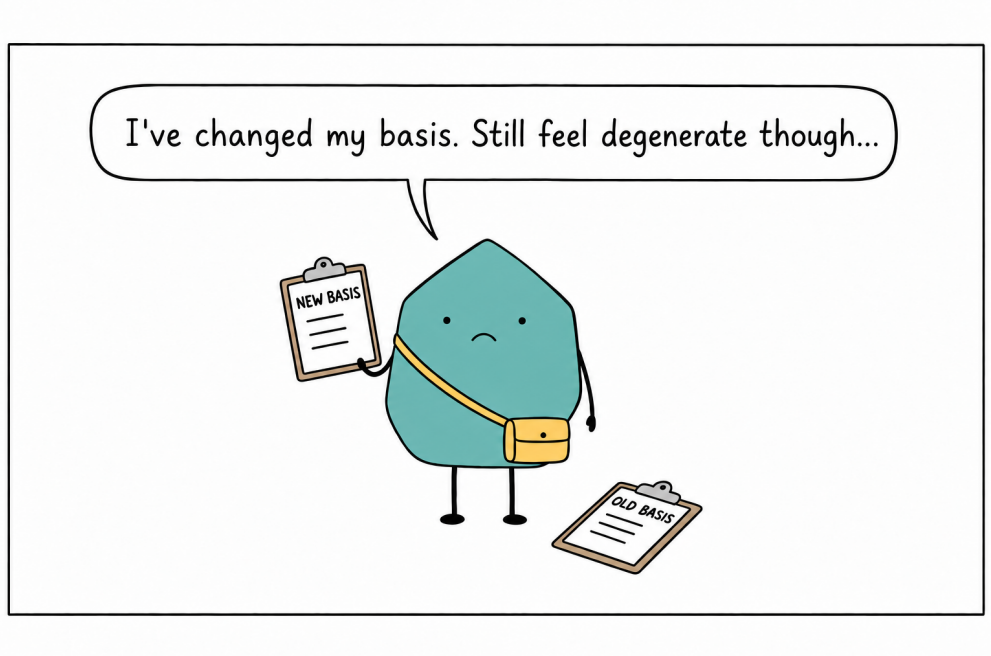
Phase II. After deleting the redundant row, the simplex method proceeds from the initial basis and gives

$$\mathbf{x}^\star = \left(\tfrac{5}{2}, \tfrac{15}{2}, 0\right)^{\mathsf{T}}, \qquad z^\star = \tfrac{35}{2}.$$

CHAPTER 6

DEGENERACY, PIVOT RULES, AND EFFICIENCY

In the preceding chapters, the simplex method moved from one basic feasible solution to another while increasing the objective value. We now encounter a special situation: the basis changes, but neither the point nor its value moves. We explain degeneracy, show how a simplex computation can literally "turn in place" at a single vertex, and introduce rules that prevent cycling. Finally, we compare the excellent practical performance of the simplex method with its surprisingly poor theoretical examples.



1 Degeneracy in the simplex method

Degeneracy is easiest to recognize in a dictionary, but let us first define it precisely. Consider the equality form

$$A\mathbf{x} = \mathbf{b}, \qquad \mathbf{x} \geq \mathbf{0}, \qquad \operatorname{rank} A = m,$$

and a feasible basis B. A basic feasible solution is *degenerate* if at least one of its basic components is zero. Equivalently, such a point has fewer than m positive components. Degeneracy is therefore a property of the basic point. A zero step in the ratio test is not its definition, but one of its manifestations during a computation.

In the dictionary

$$\mathbf{x}_B = \mathbf{p} + Q\mathbf{x}_N, \qquad z = z_0 + \mathbf{r}^{\mathsf{T}}\mathbf{x}_N$$

choose an entering variable x_j with $r_j > 0$. When it increases by $\theta \geq 0$, the basic variables change according to

$$x_{B_i}(\theta) = p_i + q_{ij}\theta.$$

Feasibility therefore requires

$$0 \leq \theta \leq \min_{i:q_{ij}<0} \frac{p_i}{-q_{ij}}.$$

If the minimum is zero, we take a *zero step*: the basis changes, but the point it represents and the objective value remain the same.

Geometrically, we have not left the vertex. We have only changed its basis representation. If such zero steps recur, the simplex computation can "turn in place" at a single vertex. We speak of *cycling* only when a basis repeats, bringing back the entire dictionary with it.

Remark 6.1: Zero steps and cycling

One or several zero steps do not indicate an error. The number of bases is finite, however, and is at most $\binom{n}{m}$. If a simplex computation continued indefinitely, it would eventually revisit a basis. Cycling can occur only under degeneracy. A suitable anti-cycling rule prevents it entirely.

1.1 An example with a degenerate starting point

Example 6.2: A degenerate starting point and a zero step

Consider a small problem that exhibits the entire phenomenon without lengthy calculations:

$$\begin{aligned} \textbf{maximize} \quad & x_2, \\ \textbf{subject to} \quad & -x_1 + x_2 \leq 0, \\ & x_1 \leq 2, \\ & x_1, x_2 \geq 0. \end{aligned}$$

The figure shows the feasible region. At the origin, the inequalities $-x_1 + x_2 \leq 0$, $x_1 \geq 0$, and $x_2 \geq 0$ are active: more constraints meet there than are needed to determine a vertex in the plane. Thus the starting point $(0, 0)$ is degenerate, whereas the optimal point $(2, 2)$ is not.

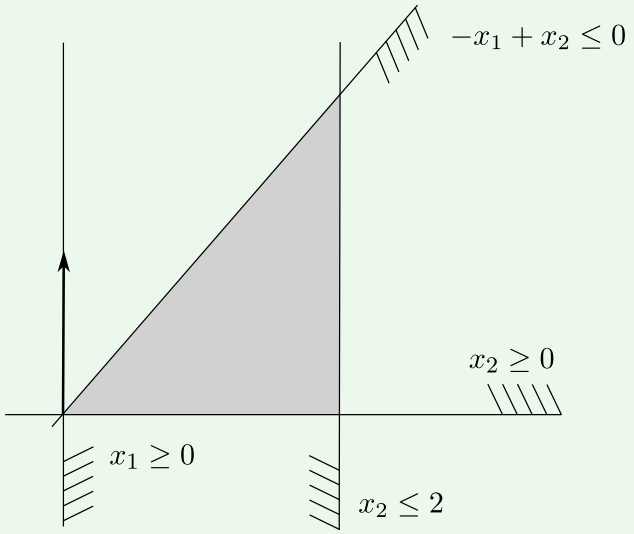


Figure 6.1: The feasible region. Three inequalities are active simultaneously at $(0, 0)$.

Adding slack variables gives

$$-x_1 + x_2 + x_3 = 0, \qquad x_1 + x_4 = 2, \qquad x_1, x_2, x_3, x_4 \geq 0.$$

For the basis $B = \{3, 4\}$, the initial dictionary is

$$x_3 = x_1 - x_2, \qquad x_4 = 2 - x_1, \qquad z = x_2.$$

It represents the point $(0, 0, 0, 2)$. The basic variable x_3 is zero, so this is a degenerate BFS. Since x_2 has a positive reduced cost, we would like to increase it. The first row immediately stops us: with $x_1 = 0$, it requires $x_3 = x_1 - x_2 \geq 0$, hence $x_2 \leq 0$. The ratio test allows only $\theta = 0$. We nevertheless pivot: x_2 enters and x_3 leaves,

$$x_2 = x_1 - x_3, \qquad x_4 = 2 - x_1, \qquad z = x_1 - x_3.$$

The basis has changed, but geometrically we are still at the origin and the value $z = 0$ is unchanged. The new dictionary now permits actual movement: x_1 can enter, the ratio test gives a step of 2, and x_4 leaves. We obtain

$$x_1 = 2 - x_4, \qquad x_2 = 2 - x_3 - x_4, \qquad z = 2 - x_3 - x_4.$$

The optimal BFS is therefore $(x_1, x_2, x_3, x_4) = (2, 2, 0, 0)$. Both basic variables in the basis $\{x_1, x_2\}$ are positive, so this optimum is nondegenerate.

2 Pivot rules and cycling

We have seen that degeneracy can lead to zero steps and, in an extreme case, to cycling. For an algorithm, "choose an improving variable" is therefore too vague. If several nonbasic variables have positive reduced costs, the algorithm must specify which one enters the basis. It must also choose a leaving variable when the ratio test ties. Together, these

choices constitute a *pivot rule*. The rule does not change what counts as an optimum, but it can dramatically change the path the simplex method takes to reach it.

2.1 Rules for choosing the entering variable

Use the notation of the general simplex dictionary. When several candidates x_j have $r_j > 0$, several natural strategies are available. Each answers a slightly different question: which variable looks best per unit increase, which yields the greatest improvement over the full step, or which follows the geometrically steepest edge?

Dantzig's rule. Choose an entering index

$$j \in \arg\max_{k \in N : r_k > 0} r_k.$$

This is the most direct choice: select the variable whose objective-row coefficient promises the greatest immediate gain per unit increase. The rule does not yet consider how far we may travel along that edge, and by itself it does not prevent cycling (Dantzig, 1963).

The greatest-improvement rule. For each candidate j, first compute the feasible step

$$\theta_j = \min_{i : q_{ij} < 0} \frac{p_i}{-q_{ij}}.$$

If the set of limiting rows is empty, the direction certifies unboundedness. Otherwise, choose a candidate maximizing $r_j \theta_j$. We now consider both the rate of improvement and the length of the feasible step. The price of this extra information is a larger number of ratio tests.

The steepest-edge rule. When the nonbasic variable x_j increases by one unit, the corresponding direction in the space of all variables has components

$$\mathbf{d}_N = \mathbf{e}_j, \qquad \mathbf{d}_B = -A_B^{-1} \mathbf{a}_j.$$

Its Euclidean length is

$$\|\mathbf{d}\|_2 = \sqrt{1 + \|A_B^{-1} \mathbf{a}_j\|_2^2}.$$

The basic version of the rule therefore chooses

$$j \in \arg\max_{k \in N : r_k > 0} \frac{r_k}{\sqrt{1 + \|A_B^{-1} \mathbf{a}_k\|_2^2}}.$$

Geometrically, we select the edge with the greatest "slope" relative to its length. The denominator includes the unit component of the entering variable, which accounts for the 1 under the square root. The norm $\|A_B^{-1} \mathbf{a}_j\|$ alone would not measure the length of the full edge direction. Practical implementations update these weights efficiently (Goldfarb and Reid, 1977).

The simplest option is, of course, to choose a candidate at random. This can be a useful heuristic, but without an additional safeguard it does not guarantee termination of a particular run.

Remark 6.3: Why the literature sometimes speaks of a maximum decrement

For a minimization problem, the greatest-improvement rule is often called the *maximum decrement rule*: it chooses the variable giving the largest decrease in the objective value. Since we use a maximization dictionary, the same idea calls for the largest increase.

2.2 Anti-cycling rules

Degeneracy itself does not invalidate the simplex method. Trouble arises only when ambiguous choices allow a return to a previously visited basis. The following rules impose a consistent order on ties and thereby prevent such a return.

Bland's rule. For a maximization dictionary with our sign convention, proceed as follows:

1. Among all nonbasic indices with $r_j > 0$, choose the smallest index j.
2. Apply the ratio test to this column. Among all rows attaining the minimum

$$\frac{p_i}{-q_{ij}}, \qquad q_{ij} < 0,$$

choose the row whose *basic variable* has the smallest index.

Bland proved that this rule does not cycle (Bland, 1977). The proof does not assert that the visited bases form an ordinary lexicographically increasing sequence. It considers the largest-index variable whose basic or nonbasic status would change in a hypothetical cycle and derives a contradiction from the least-index choices.

The rule is disarmingly simple: whenever a choice is ambiguous, favor the smallest index. This consistency is essential. Applying the rule only to the entering variable and breaking ratio-test ties arbitrarily is not enough.

Lexicographic pivoting. Instead of using indices alone, we can break ties by comparing entire, suitably normalized rows. The first differing component decides, just as the first differing letter determines dictionary order. More precisely, fix an ordering of the components and let B_0 be the chosen initial feasible basis. For a chosen entering column j, compare the vectors

$$\boldsymbol{\ell}_i = \frac{1}{-q_{ij}} \left(p_i,\, (A_B^{-1}A_{B_0})_{i1}, \ldots, (A_B^{-1}A_{B_0})_{im}\right)$$

among rows with $q_{ij} < 0$, using the displayed lexicographic order, and choose the row with the smallest $\boldsymbol{\ell}_i$. The first component performs the ordinary ratio test. The remaining components uniquely break any tie. Thus the leaving row is unique for a chosen entering column. To make the entire pivot step unique, a rule for choosing the entering variable must also be specified. Lexicographic feasibility guarantees that no basis repeats.

Symbolic perturbation of the right-hand side. We can also interpret the lexicographic rule by symbolically separating equal values by tiny amounts. Use a hierarchy of small perturbations

$$\mathbf{b}(\varepsilon) = \mathbf{b} + A_{B_0}(\varepsilon, \varepsilon^2, \dots, \varepsilon^m)^{\mathsf{T}}, \qquad 0 < \varepsilon \ll 1,$$

so that the initial basic vector is $A_{B_0}^{-1}\mathbf{b} + (\varepsilon, \dots, \varepsilon^m)^{\mathsf{T}}$, and its formerly zero components become symbolically positive. Expressions are not evaluated in floating-point arithmetic. They are compared using the first nonzero power of ε. Such a generic hierarchical perturbation removes the equalities responsible for degeneracy of the visited bases. After the calculation, let $\varepsilon \to 0^+$. If the original problem has a finite optimum, this gives an optimal limit point. Arbitrary small, pairwise distinct decimal perturbations do not generally guarantee this property and can change the numerical problem.

Remark 6.4: Perturbation as a conceptual device

We should not blindly substitute very small decimal numbers for $\varepsilon, \varepsilon^2, \dots$. We treat them symbolically and let $\varepsilon \to 0^+$ at the end. The aim is not to change the original problem secretly, but to resolve choices that are indistinguishable in the unperturbed problem.

3 Efficiency of the simplex method

In practice, the simplex method is remarkably successful even for problems with many variables and constraints. This has made it a working tool of optimization, rather than merely an elegant textbook algorithm. Theory nevertheless reminds us that good practical performance is not a guarantee for every possible input. The number of pivots depends on the problem's geometry, the initial basis, and the pivot rule. Without precisely defining the class of input data, we cannot promise a fixed multiple of the number of constraints.

3.1 The Klee–Minty example

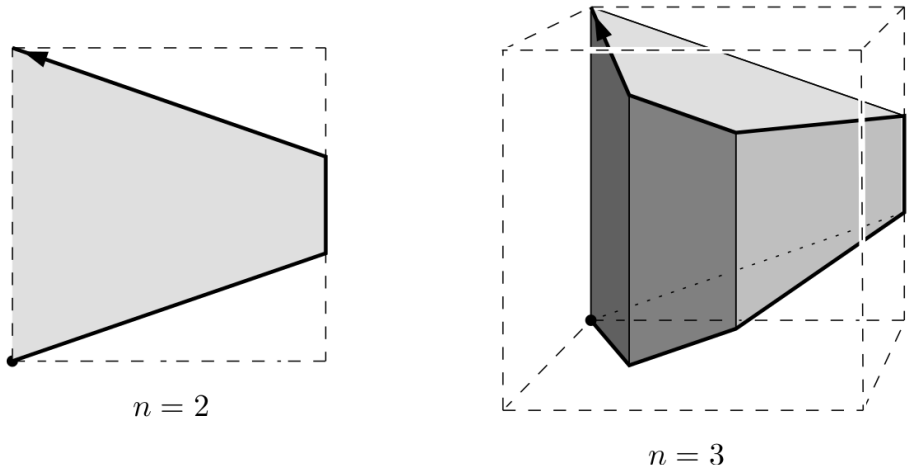
Theoretical expectations received a sobering correction in 1972. Victor Klee and George Minty constructed a family of linear programs for which Dantzig's rule, starting from the slack-variable basis, visits all 2^n vertices of a deformed n-dimensional cube (Klee and Minty, 1972). One common form of this family is

$$\begin{aligned} \textbf{maximize} \quad & \sum_{j=1}^{n} 2^{n-j} x_j, \\ \textbf{subject to} \quad & \sum_{j=1}^{i-1} 2^{i-j+1} x_j + x_i \le 5^i, \quad i = 1, \dots, n, \\ & x_1, \dots, x_n \ge 0. \end{aligned}$$

For $n = 3$, this becomes the concrete problem

$$\begin{aligned} \textbf{maximize} \quad & 4x_1 + 2x_2 + x_3, \\ & x_1 \le 5, \\ & 4x_1 + x_2 \le 25, \\ & 8x_1 + 4x_2 + x_3 \le 125, \\ & x_1, x_2, x_3 \ge 0. \end{aligned}$$

The cube is skewed so that the locally most attractive choice sends the algorithm along a very long route. Starting at the origin with Dantzig's rule, the simplex method requires $2^n - 1$ pivots. This is an exponential lower bound for this particular combination of a problem family, starting basis, and pivot rule. It does not say that every rule follows the same route through the problem.



A schematic deformed cube. The exact simplex path depends on the coefficients and the pivot rule.

The figure conveys only the idea of a deformed cube. The actual coefficients that force Dantzig's rule to visit all vertices in the required order are considerably more peculiar. The author therefore takes the liberty of sparing both the reader and his own drawing skills, leaving the figure as a geometric sketch.

3.2 Theoretical limits

The Klee–Minty example does not imply that linear programming itself is exponentially hard. Linear programming can be solved in time polynomial in the bit length of rational input data, for example by the ellipsoid method or a suitable interior-point method (Khachiyan, 1979, Karmarkar, 1984). This fact alone does not bound the number of simplex pivots by a polynomial in the numbers of variables and constraints.

It is easy to confuse two different open questions here. No efficiently implementable pivot rule is known that takes only polynomially many pivots on every problem. New lower bounds rule out further broad classes of natural rules, but not every possible rule (Disser et al., 2026). The existence of any strongly polynomial algorithm for general linear programming is a separate open question (Allamigeon et al., 2021). An efficient rule with a polynomial number of pivots would yield a strongly polynomial simplex algorithm. A

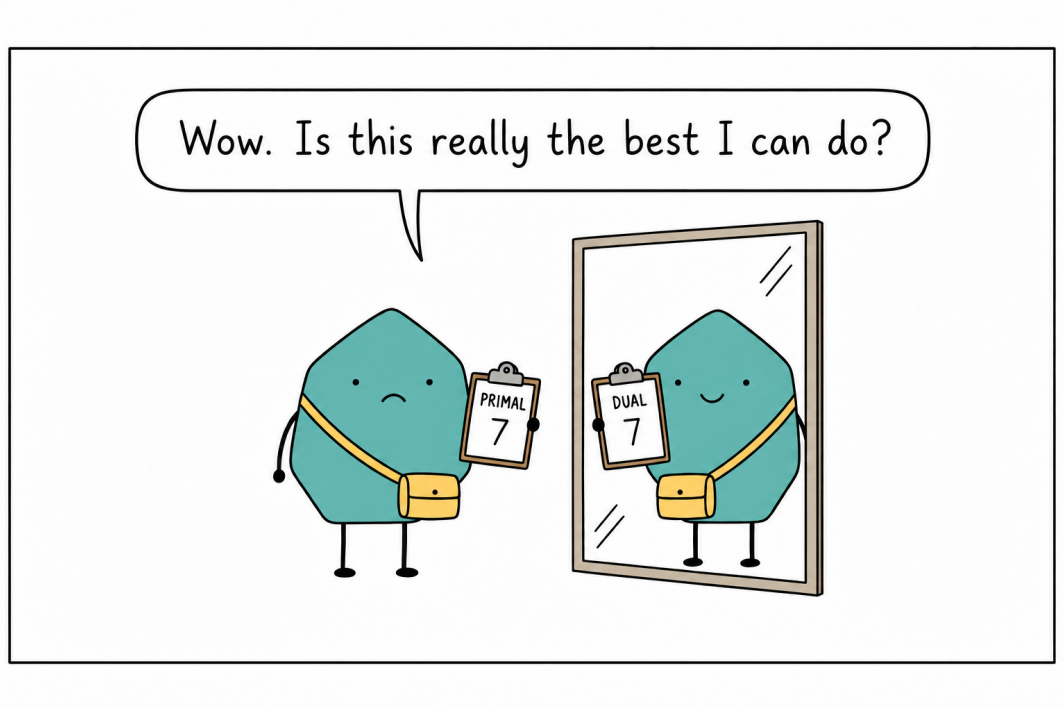
different strongly polynomial algorithm, however, would not necessarily define any simplex path.

Likewise, a short path that exists in the graph must not be confused with the path an algorithm actually chooses. The diameter of a polyhedron's graph and the number of pivots along a particular simplex path are not the same quantity. The diameter concerns shortest distances between pairs of vertices, whereas a pivot rule may choose a much longer route. Moreover, the original Hirsch conjecture of a linear diameter bound for general polyhedra has been disproved (Santos, 2012). Even if it had been true, it would not by itself have guaranteed polynomial performance for an arbitrary pivot rule.

CHAPTER 7

LINEAR PROGRAMMING DUALITY I

Duality lets us view a single linear program from two sides. On one side, we seek the best production plan, flow, or schedule. On the other, we price resources so that their total cost bounds the value of every feasible plan. We first derive this perhaps surprising idea through an example, then present general rules for forming the dual, the weak and strong duality theorems, and complementary slackness conditions.



1 Motivation and construction of the dual problem

Linear programs have an important and perhaps surprising property: every problem admits two interpretations. We already know the first. For example, we seek a production plan that maximizes profit without exceeding the available resources. The second shifts our attention from products to resources: what prices should we assign to the resources so that no feasible plan can be worth more than the cost of all the resources it uses?

We call the first formulation the *primal problem* and the second the *dual problem*. These are not unrelated problems that happen to look alike. They are two mathematical descriptions of the same situation: one seeks the best decision, and the other seeks the best proof that no decision can do better.

The algebraic route to this second view is surprisingly simple. Consider a maximization problem. A nonnegative linear combination of its constraints can sometimes yield an inequality whose left-hand side dominates the objective function. The right-hand side of this combination then bounds the value of every feasible solution from above. The dual problem seeks the best such bound.

Example 7.1: Upper bounds from combinations of constraints

Consider the problem

$$\begin{aligned} \textbf{maximize} \quad & 2x_1 + 3x_2, \\ \textbf{subject to} \quad & 4x_1 + 8x_2 \le 12, \\ & 2x_1 + x_2 \le 3, \\ & 3x_1 + 2x_2 \le 4, \\ & x_1, x_2 \ge 0. \end{aligned}$$

Without finding the optimum, we can immediately write, for example,

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

Thus 12 is a valid, though clearly not very clever, upper bound. We need not use constraints one at a time. Adding one third of the first constraint and one third of the second gives

$$2x_1 + 3x_2 \le \tfrac{1}{3}(4x_1 + 8x_2) + \tfrac{1}{3}(2x_1 + x_2) \le \tfrac{1}{3}\,12 + \tfrac{1}{3}\,3 = 5.$$

The value 5 is a better upper bound on the optimum. We can continue this idea: which nonnegative combination of constraints gives the lowest valid upper bound of all?

Let y_1, y_2, y_3 be nonnegative weights for the three constraints. Their combination has left-hand side

$$(4y_1 + 2y_2 + 3y_3)x_1 + (8y_1 + y_2 + 2y_3)x_2.$$

To dominate the objective function for every $\mathbf{x} \ge \mathbf{0}$, it must satisfy

$$4y_1 + 2y_2 + 3y_3 \ge 2, \qquad 8y_1 + y_2 + 2y_3 \ge 3.$$

The coefficients y_i play two roles. Algebraically, they are the weights used to combine the original inequalities. In the production story, they can also be interpreted

as unit prices for the three resources. The dual constraints then say that the resources consumed by one unit of a product must cost at least as much as its profit. The best upper bound is therefore found by the dual problem

$$\begin{aligned} \textbf{minimize} \quad & 12y_1 + 3y_2 + 4y_3, \\ \textbf{subject to} \quad & 4y_1 + 2y_2 + 3y_3 \geq 2, \\ & 8y_1 + y_2 + 2y_3 \geq 3, \\ & y_1, y_2, y_3 \geq 0. \end{aligned}$$

The primal solution $\mathbf{x}^\star = (\frac{1}{2}, \frac{5}{4})^\mathsf{T}$ and the dual solution $\mathbf{y}^\star = (\frac{5}{16}, 0, \frac{1}{4})^\mathsf{T}$ are both feasible and both have value $19/4$. The first says "I can produce this much value," while the second says "no production plan can do better." Weak duality, proved below, will show that this equality certifies optimality of both solutions.

1.1 The standard primal–dual pair

For $A \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{c} \in \mathbb{R}^n$, the standard pair consists of

$$\max \ \mathbf{c}^\mathsf{T}\mathbf{x} \quad \text{subject to} \quad A\mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0} \tag{P}$$

and

$$\min \ \mathbf{b}^\mathsf{T}\mathbf{y} \quad \text{subject to} \quad A^\mathsf{T}\mathbf{y} \geq \mathbf{c}, \quad \mathbf{y} \geq \mathbf{0}. \tag{D}$$

We call (P) the *primal* and (D) the *dual*. Their feasible sets are generally different. Calling a problem "dual" does not mean that it is the same optimization problem. Their close relationship concerns, above all, bounds and optimal values.

In the production interpretation, a primal variable x_j is the quantity of product j, while a dual variable y_i prices resource i. Each column of A can thus be read as a recipe for one product, and each row as the budget for one resource. The transpose in the dual simply exchanges these two perspectives. The data remain the same, read once by rows and once by columns.

Every *dual feasible* $\mathbf{y}$, meaning a vector satisfying both $\mathbf{y} \geq \mathbf{0}$ and $A^\mathsf{T}\mathbf{y} \geq \mathbf{c}$, gives an upper bound for every primal feasible $\mathbf{x}$:

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

The condition $\mathbf{y} \geq \mathbf{0}$ alone is not enough.

1.2 General rules for forming the dual

The standard pair is easy to remember, but real formulations also contain equalities, reversed inequalities, and free variables. The following recipe is not a new theory, merely careful bookkeeping of signs. Start with a primal maximization problem whose constraints may have directions $\leq$, $\geq$, or $=$, and whose variables may be nonnegative, nonpositive, or free. Each primal constraint gives one dual variable, and each primal variable gives one dual constraint. The rules are:

Element of a maximization primal	Corresponding dual element
constraint i with $\le$	$y_i \ge 0$
constraint i with $\ge$	$y_i \le 0$
constraint i with $=$	$y_i \in \mathbb{R}$
variable $x_j \ge 0$	$(A^{\mathsf{T}}\mathbf{y})_j \ge c_j$
variable $x_j \le 0$	$(A^{\mathsf{T}}\mathbf{y})_j \le c_j$
free variable $x_j \in \mathbb{R}$	$(A^{\mathsf{T}}\mathbf{y})_j = c_j$

The dual objective is $\min \mathbf{b}^{\mathsf{T}}\mathbf{y}$. Notice the simple symmetry: primal constraints create dual variables, and primal variables create dual constraints. If the primal is a minimization problem, we can first convert it to maximization by changing the sign of its objective, or use the symmetric rules. Taking the dual once more returns the original problem after the usual identification of equivalent notation.

Example 7.2: Forming the dual with mixed directions and signs

For the problem

$$\begin{aligned} &\textbf{maximize} \quad 3x_1 + x_2 - 2x_3, \\ &x_1 + 2x_2 - x_3 \le 4, \\ &2x_1 - x_2 + x_3 \ge 1, \\ &x_1 + x_2 + x_3 = 2, \\ &\qquad x_1 \ge 0, \quad x_2 \le 0, \quad x_3 \in \mathbb{R} \end{aligned}$$

introduce $y_1 \ge 0$, $y_2 \le 0$, and a free variable y_3. The dual problem is

$$\begin{aligned} &\textbf{minimize} \quad 4y_1 + y_2 + 2y_3, \\ &y_1 + 2y_2 + y_3 \ge 3, \\ &2y_1 - y_2 + y_3 \le 1, \\ &-y_1 + y_2 + y_3 = -2, \\ &\qquad y_1 \ge 0, \quad y_2 \le 0, \quad y_3 \in \mathbb{R}. \end{aligned}$$

The direction of each dual constraint is determined by the sign restriction on the corresponding primal variable, not by the direction of any single primal constraint.

1.3 Another perspective: a Lagrangian derivation

The preceding construction combined inequalities and is the most natural introduction to duality. The same idea can be expressed using Lagrange multipliers. This perspective is not needed for forming dual problems in practice, but it shows why LP duality later extends naturally to convex optimization.

For the standard problem (P) and a multiplier $\mathbf{y} \ge \mathbf{0}$, define the Lagrangian

$$L(\mathbf{x}, \mathbf{y}) = \mathbf{c}^{\mathsf{T}}\mathbf{x} + \mathbf{y}^{\mathsf{T}}(\mathbf{b} - A\mathbf{x}) = \mathbf{b}^{\mathsf{T}}\mathbf{y} + \mathbf{x}^{\mathsf{T}}(\mathbf{c} - A^{\mathsf{T}}\mathbf{y}).$$

If $\mathbf{x}$ is primal feasible, then $\mathbf{b} - A\mathbf{x} \ge \mathbf{0}$, and hence $L(\mathbf{x}, \mathbf{y}) \ge \mathbf{c}^{\mathsf{T}}\mathbf{x}$. Therefore

$$g(\mathbf{y}) = \sup_{\mathbf{x} \ge \mathbf{0}} L(\mathbf{x}, \mathbf{y})$$

is an upper bound on the primal optimal value. If $\mathbf{c} - A^{\mathsf{T}}\mathbf{y}$ has a positive component, then $g(\mathbf{y}) = +\infty$. If instead $A^{\mathsf{T}}\mathbf{y} \geq \mathbf{c}$, the supremum is attained at $\mathbf{x} = \mathbf{0}$, and

$$g(\mathbf{y}) = \mathbf{b}^{\mathsf{T}}\mathbf{y}.$$

Minimizing this finite upper bound gives exactly (D). The multiplier y_i is not a measure of how "active" a constraint is. Its precise relationship to activity is described by complementary slackness.

2 Weak and strong duality

The dual problem is meant to bound the primal optimum from above. Weak duality confirms that every dual feasible solution does so. Strong duality says something much more surprising: when a finite optimum is attained, the best upper bound meets it exactly.

Theorem 7.3: Weak duality

If $\mathbf{x}$ is feasible for (P) and $\mathbf{y}$ is feasible for (D), then

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

Consequently, if (P) is unbounded above, (D) is infeasible. If (D) is unbounded below, (P) is infeasible.

Proof of the theorem.

We prove the same idea in two ways. The first is a short chain of inequalities. The second reveals the components of the difference between the dual and primal values.

First perspective: a chain of inequalities. The conditions $\mathbf{x} \geq \mathbf{0}$ and $A^{\mathsf{T}}\mathbf{y} \geq \mathbf{c}$ give the first inequality, while $\mathbf{y} \geq \mathbf{0}$ and $A\mathbf{x} \leq \mathbf{b}$ give the second:

$$\mathbf{c}^{\mathsf{T}}\mathbf{x} \leq (A^{\mathsf{T}}\mathbf{y})^{\mathsf{T}}\mathbf{x} = \mathbf{y}^{\mathsf{T}}A\mathbf{x} \leq \mathbf{y}^{\mathsf{T}}\mathbf{b} = \mathbf{b}^{\mathsf{T}}\mathbf{y}.$$

Second perspective: the duality gap. The difference between the values can be written as

$$\begin{aligned}\mathbf{b}^{\mathsf{T}}\mathbf{y} - \mathbf{c}^{\mathsf{T}}\mathbf{x} &= \mathbf{y}^{\mathsf{T}}(\mathbf{b} - A\mathbf{x}) + \mathbf{x}^{\mathsf{T}}(A^{\mathsf{T}}\mathbf{y} - \mathbf{c}) \\ &\geq 0.\end{aligned}$$

Both inner products are nonnegative, since each pairs two nonnegative vectors. We call the difference the *duality gap*. This second proof both confirms its nonnegativity and will later let us determine exactly when the gap is zero.

If the primal were unbounded above and a dual feasible $\mathbf{y}$ existed, the finite number $\mathbf{b}^{\mathsf{T}}\mathbf{y}$ would bound all primal values from above by the inequality just proved, a contradiction. The second consequence is symmetric. ■

Theorem 7.4: Strong duality

For the standard pair (P), (D), if either problem is feasible and its objective is bounded in the relevant direction, then both problems have optimal solutions and their optimal values coincide. Equivalently, exactly one of the following occurs:

1. both problems are infeasible,
2. (P) is unbounded above and (D) is infeasible,
3. (P) is infeasible and (D) is unbounded below,
4. both problems have optimal solutions $\mathbf{x}^\star, \mathbf{y}^\star$ and

$$\mathbf{c}^\mathsf{T}\mathbf{x}^\star = \mathbf{b}^\mathsf{T}\mathbf{y}^\star.$$

Weak duality allowed a possible gap between the best plan and the best upper bound. Strong duality closes it: when a finite optimum exists, there is a resource pricing that matches the value of the optimal plan exactly. We will prove this in the next chapter, first from the simplex tableau and then using Farkas' lemma. It is one of the fundamental results of linear programming (Dantzig, 1963, Matoušek and Gärtner, 2007).

2.1 Complementary slackness conditions

Equality of optimal values has a finer consequence. Each summand in the decomposition of the duality gap is nonnegative, so their sum can be zero only when every summand vanishes individually. This is precisely what complementary slackness expresses.

Theorem 7.5: Complementary slackness

Let $\mathbf{x}$ be feasible for (P) and $\mathbf{y}$ feasible for (D). Then $\mathbf{x}$ and $\mathbf{y}$ are optimal if and only if

$$y_i\big(b_i - \mathbf{a}_i^\mathsf{T}\mathbf{x}\big) = 0 \quad (i = 1, \dots, m)$$

and simultaneously

$$x_j\big((A^\mathsf{T}\mathbf{y})_j - c_j\big) = 0 \quad (j = 1, \dots, n).$$

Proof of the theorem.

For a feasible pair, all terms in the duality-gap decomposition

$$\sum_{i=1}^{m} y_i\big(b_i - \mathbf{a}_i^\mathsf{T}\mathbf{x}\big) + \sum_{j=1}^{n} x_j\big((A^\mathsf{T}\mathbf{y})_j - c_j\big)$$

are nonnegative. Thus the gap is zero if and only if every displayed product is zero. By weak duality, a zero gap means that no primal feasible solution has a higher value than $\mathbf{x}$, and no dual feasible solution has a lower value than $\mathbf{y}$. Both solutions are optimal. Conversely, strong duality gives a zero gap for any optimal pair. ■

The conditions have two important consequences, but the implications are only one-way:

$$b_i - \mathbf{a}_i^\mathsf{T}\mathbf{x}^\star > 0 \implies y_i^\star = 0, \qquad y_i^\star > 0 \implies \mathbf{a}_i^\mathsf{T}\mathbf{x}^\star = b_i,$$

and similarly

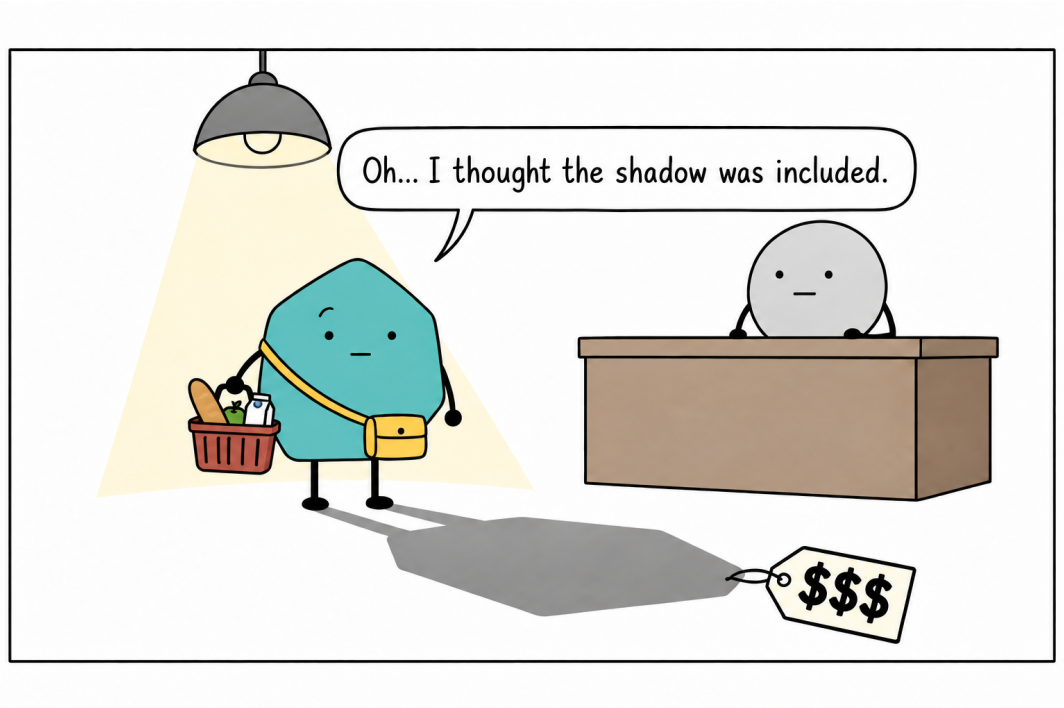
$$(A^{\mathsf{T}}\mathbf{y}^\star)_j > c_j \implies x_j^\star = 0, \qquad x_j^\star > 0 \implies (A^{\mathsf{T}}\mathbf{y}^\star)_j = c_j.$$

In other words, a resource with unused capacity at the optimum must have zero price. A resource with positive price must be fully used. Beware of the converse implications, however. An active primal constraint may have a zero dual variable, and a tight dual constraint may correspond to a zero primal variable, especially under degeneracy or with multiple optimal solutions.

CHAPTER 8

LINEAR PROGRAMMING DUALITY II

In the previous chapter, we had to take strong duality on trust. We now repay that debt twice: first by reading a dual solution from an optimal simplex tableau, and then by using Farkas' lemma and the geometry of a convex cone. We then turn from proofs to interpreting dual variables as shadow prices. Finally, we reverse the logic of the simplex method and introduce its dual variant.



1 Proving strong duality from an optimal basis

The first proof looks directly at the final simplex tableau. It therefore relies on correctness and termination of the simplex method with an anti-cycling rule. Readers not yet convinced by this algorithmic starting point need not worry: the second proof in this chapter takes a different route.

Recall the standard primal–dual pair

$$\max\ \mathbf{c}^{\mathsf{T}}\mathbf{x} \quad \text{subject to} \quad A\mathbf{x} \le \mathbf{b}, \quad \mathbf{x} \ge \mathbf{0} \tag{P}$$

and

$$\min\ \mathbf{b}^{\mathsf{T}}\mathbf{y} \quad \text{subject to} \quad A^{\mathsf{T}}\mathbf{y} \ge \mathbf{c}, \quad \mathbf{y} \ge \mathbf{0}. \tag{D}$$

Suppose (P) is feasible and its objective is bounded above. Adding slack variables $\mathbf{s} \ge \mathbf{0}$ gives

$$\widetilde{A}\widetilde{\mathbf{x}} = \mathbf{b}, \qquad \widetilde{A} = (A\ I_m), \qquad \widetilde{\mathbf{x}} = \begin{pmatrix}\mathbf{x}\\ \mathbf{s}\end{pmatrix}, \qquad \widetilde{\mathbf{c}} = \begin{pmatrix}\mathbf{c}\\ \mathbf{0}\end{pmatrix}.$$

The matrix $\widetilde{A}$ has full row rank. The fundamental theorem of linear programming and a terminating simplex procedure with an anti-cycling rule therefore provide a final basis B satisfying the optimality criterion: $\mathbf{p} \ge \mathbf{0}$ and $\mathbf{r} \le \mathbf{0}$. Here we call these two conditions together optimality of the basis. Under degeneracy, optimality of the associated BFS alone is not enough. The nonpositive reduced costs turn out to encode a dual certificate. Write

$$\mathbf{p} = \widetilde{A}_B^{-1}\mathbf{b} \ge \mathbf{0}, \qquad \mathbf{y}^\star = \widetilde{A}_B^{-\mathsf{T}}\widetilde{\mathbf{c}}_B.$$

Lemma 8.1: A dual solution from an optimal basis

The vector $\mathbf{y}^\star$ is feasible for (D), and the value of the associated optimal BFS $\mathbf{x}^\star$ satisfies

$$\mathbf{c}^{\mathsf{T}}\mathbf{x}^\star = \mathbf{b}^{\mathsf{T}}\mathbf{y}^\star.$$

Proof of the lemma.

We have already constructed a candidate dual solution from the final basis. We must check two things: that it is dual feasible and that its value agrees with that of the primal solution we found.

For each column j of the augmented matrix, define the reduced cost

$$\widetilde{r}_j = \widetilde{c}_j - \widetilde{\mathbf{a}}_j^{\mathsf{T}}\mathbf{y}^\star.$$

For basic columns, $\widetilde{r}_j = 0$, because

$$\widetilde{A}_B^{\mathsf{T}}\mathbf{y}^\star = \widetilde{\mathbf{c}}_B.$$

For nonbasic columns, the reduced costs in the optimal tableau are nonpositive. Thus, for all columns,

$$\widetilde{A}^{\mathsf{T}}\mathbf{y}^\star \ge \widetilde{\mathbf{c}}.$$

The first n components give $A^{\mathsf{T}}\mathbf{y}^\star \ge \mathbf{c}$. The last m components correspond to the identity columns and give $\mathbf{y}^\star \ge \mathbf{0}$. This step is essential: the dual inequalities alone do not suffice, as the sign restriction must also hold. Hence $\mathbf{y}^\star$ is indeed dual feasible.

The value of the basic solution is

$$\begin{aligned}\mathbf{c}^{\mathsf{T}}\mathbf{x}^\star &= \widetilde{\mathbf{c}}_B^{\mathsf{T}}\widetilde{A}_B^{-1}\mathbf{b} = \left(\widetilde{A}_B^{-\mathsf{T}}\widetilde{\mathbf{c}}_B\right)^{\mathsf{T}}\mathbf{b} \\ &= (\mathbf{y}^\star)^{\mathsf{T}}\mathbf{b} = \mathbf{b}^{\mathsf{T}}\mathbf{y}^\star.\end{aligned}$$

■

We have found a primal feasible $\mathbf{x}^\star$ and a dual feasible $\mathbf{y}^\star$ with equal values. Weak duality leaves no room for a better solution on either side, so both are optimal. This proves the main part of strong duality. If the dual, rather than the primal, is feasible and bounded, apply the same argument to the dual and use the fact that the dual of the dual is the original problem. Weak duality supplies the unboundedness and infeasibility cases. The proof also explains why the simplex multipliers $\mathbf{y}^\star = \widetilde{A}_B^{-\mathsf{T}}\widetilde{\mathbf{c}}_B$ form a dual solution (Dantzig, 1963).

2 Farkas' lemma and a second proof of strong duality

Farkas' lemma is a *theorem of the alternative*: exactly one of two systems has a solution. One system describes the desired nonnegative representation, and the other provides a certificate that no such representation exists (Farkas, 1902, Matoušek and Gärtner, 2007). Before stating the lemma algebraically, let us build its geometric picture.

2.1 A point, a cone, and a separating hyperplane

Definition 8.2: A convex cone

Let $\mathbf{a}_1, \dots, \mathbf{a}_n \in \mathbb{R}^m$. The *convex cone* generated by these vectors is the set of all their nonnegative linear combinations,

$$\operatorname{cone}\{\mathbf{a}_1, \dots, \mathbf{a}_n\} = \left\{ \sum_{j=1}^{n} x_j \mathbf{a}_j : \; x_j \geq 0 \right\}.$$

Imagine building the vector $\mathbf{b}$ from the columns of $A = (\mathbf{a}_1 \; \cdots \; \mathbf{a}_n)$ using nonnegative coefficients. All vectors we can build this way fill the cone

$$C = \operatorname{cone}\{\mathbf{a}_1, \dots, \mathbf{a}_n\} = \{A\mathbf{x} : \mathbf{x} \geq \mathbf{0}\}.$$

The system $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$ has a solution if and only if $\mathbf{b}$ belongs to this cone. If it does not, we want a single hyperplane to certify that fact: the whole cone lies on its nonnegative side, while $\mathbf{b}$ lies strictly on its negative side. The normal vector of this hyperplane will be the required certificate.

2.2 Algebraic statement and proof

Lemma 8.3: Farkas' lemma

Let $A \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$. Exactly one of the following systems has a solution:

(F1) $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$,

(F2) there exists $\mathbf{y} \in \mathbb{R}^m$ such that $A^\mathsf{T}\mathbf{y} \geq \mathbf{0}$ and $\mathbf{b}^\mathsf{T}\mathbf{y} < 0$.

Proof of the lemma.

Both alternatives cannot hold at once. If $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$, and $A^\mathsf{T}\mathbf{y} \geq \mathbf{0}$, then

$$\mathbf{b}^\mathsf{T}\mathbf{y} = \mathbf{x}^\mathsf{T}A^\mathsf{T}\mathbf{y} \geq 0,$$

contradicting the condition in (F2).

It remains to prove that a certificate exists when (F1) has no solution. Denote the columns of A by $\mathbf{a}_1, \ldots, \mathbf{a}_n$ and set

$$C = \operatorname{cone}\{\mathbf{a}_1, \ldots, \mathbf{a}_n\} = \{A\mathbf{x} : \mathbf{x} \geq \mathbf{0}\}.$$

Convexity of C follows directly from the definition. Let us briefly justify closedness. Every point of C can be expressed as a nonnegative combination of a linearly independent subset of the generators. Indeed, if the generators with positive coefficients are dependent, there is a nonzero vector $\mathbf{d}$, supported on their indices, with $A\mathbf{d} = \mathbf{0}$. After changing its sign if necessary, it has some component $d_j > 0$. For

$$\lambda = \min_{j:d_j>0} \frac{x_j}{d_j}$$

we have $\mathbf{x} - \lambda\mathbf{d} \geq \mathbf{0}$. This represents the same point of the cone, and at least one positive component vanishes. After finitely many repetitions, only independent generators remain.

Thus C is the union of finitely many cones $\{A_I\mathbf{x}_I : \mathbf{x}_I \geq \mathbf{0}\}$ with linearly independent columns in A_I. The empty subset gives only $\mathbf{0}$. Every other such cone is closed: if $A_I\mathbf{x}_I^{(k)} \to \mathbf{v}$, then

$$\mathbf{x}_I^{(k)} = (A_I^\mathsf{T}A_I)^{-1}A_I^\mathsf{T}(A_I\mathbf{x}_I^{(k)})$$

converges to a nonnegative vector $\mathbf{x}_I$, and $\mathbf{v} = A_I\mathbf{x}_I$. A finite union of these closed cones is closed, so C is closed as well.

Since $\mathbf{b} \notin C$, a point $\mathbf{z} \in C$ closest to $\mathbf{b}$ exists. For every $\mathbf{w} \in C$ and $t \in [0, 1]$, the point $\mathbf{z} + t(\mathbf{w} - \mathbf{z})$ lies in C. The distance is minimized at $t = 0$, so its right derivative gives

$$(\mathbf{b} - \mathbf{z})^\mathsf{T}(\mathbf{w} - \mathbf{z}) \leq 0.$$

The geometric meaning is simple: if any direction from $\mathbf{z}$ into the cone formed an acute angle with the vector toward $\mathbf{b}$, moving in that direction would bring us closer to $\mathbf{b}$. This would contradict the choice of a closest point. Since C is a cone, we may substitute $\mathbf{w} = \mathbf{0}$ and $\mathbf{w} = 2\mathbf{z}$, obtaining

$$(\mathbf{b} - \mathbf{z})^\mathsf{T}\mathbf{z} = 0.$$

Set $\mathbf{y} = \mathbf{z} - \mathbf{b}$. Then $\mathbf{y}^{\mathsf{T}}\mathbf{w} \geq 0$ for all $\mathbf{w} \in C$, and in particular

$$A^{\mathsf{T}}\mathbf{y} \geq \mathbf{0}.$$

Moreover,

$$\mathbf{b}^{\mathsf{T}}\mathbf{y} = \mathbf{b}^{\mathsf{T}}(\mathbf{z} - \mathbf{b}) = -\|\mathbf{b} - \mathbf{z}\|_2^2 < 0,$$

where we used $(\mathbf{b} - \mathbf{z})^{\mathsf{T}}\mathbf{z} = 0$ and $\mathbf{b} \neq \mathbf{z}$. Hence $\mathbf{y}$ satisfies (F2). ■

Geometrically, the statement says that either $\mathbf{b}$ belongs to the cone generated by the columns of A, or a hyperplane with normal $\mathbf{y}$ separates it from that cone. The inequalities $\mathbf{y}^{\mathsf{T}}\mathbf{a}_j \geq 0$ place the entire cone in one closed halfspace, while $\mathbf{y}^{\mathsf{T}}\mathbf{b} < 0$ places $\mathbf{b}$ in the opposite open halfspace.

For duality, we need the following consequence.

Lemma 8.4: Farkas' lemma for inequalities

The system

$$H\mathbf{x} \leq \mathbf{h}, \qquad \mathbf{x} \geq \mathbf{0}$$

is infeasible if and only if some $\boldsymbol{\lambda} \geq \mathbf{0}$ satisfies

$$H^{\mathsf{T}}\boldsymbol{\lambda} \geq \mathbf{0}, \qquad \mathbf{h}^{\mathsf{T}}\boldsymbol{\lambda} < 0.$$

Proof of the lemma.

After adding slack variables, the system is equivalent to

$$(H \; I)\begin{pmatrix}\mathbf{x}\\ \mathbf{s}\end{pmatrix} = \mathbf{h}, \qquad \mathbf{x}, \mathbf{s} \geq \mathbf{0}.$$

Applying Farkas' lemma to $(H \; I)$ gives $(H \; I)^{\mathsf{T}}\boldsymbol{\lambda} \geq \mathbf{0}$, which is exactly the pair of conditions $H^{\mathsf{T}}\boldsymbol{\lambda} \geq \mathbf{0}$ and $\boldsymbol{\lambda} \geq \mathbf{0}$. ■

2.3 Strong duality from Farkas' lemma

The proof works as follows. Place a target an arbitrarily small $\varepsilon > 0$ above the optimal value γ. No primal feasible solution can reach it, so Farkas' lemma supplies a certificate of infeasibility. After suitable normalization, this certificate becomes a nearly optimal dual solution. Letting ε decrease to zero closes the gap.

A target above the primal optimum. Suppose (P) has an optimal solution $\mathbf{x}^\star$, and let $\gamma = \mathbf{c}^{\mathsf{T}}\mathbf{x}^\star$. For every $\varepsilon > 0$, the system

$$A\mathbf{x} \leq \mathbf{b}, \qquad -\mathbf{c}^{\mathsf{T}}\mathbf{x} \leq -(\gamma + \varepsilon), \qquad \mathbf{x} \geq \mathbf{0}$$

is infeasible. Apply the preceding lemma with

$$H = \begin{pmatrix} A \\ -\mathbf{c}^{\mathsf{T}} \end{pmatrix}, \qquad \mathbf{h} = \begin{pmatrix} \mathbf{b} \\ -(\gamma + \varepsilon) \end{pmatrix}.$$

The Farkas certificate. There are therefore $\mathbf{u} \geq \mathbf{0}$ and $t \geq 0$ such that

$$A^{\mathsf{T}}\mathbf{u} - t\mathbf{c} \geq \mathbf{0}, \qquad \mathbf{b}^{\mathsf{T}}\mathbf{u} - t(\gamma + \varepsilon) < 0.$$

Multiplying the first inequality by $\mathbf{x}^\star \geq \mathbf{0}$ and using $A\mathbf{x}^\star \leq \mathbf{b}$ gives

$$\mathbf{b}^\mathsf{T}\mathbf{u} \geq \mathbf{u}^\mathsf{T} A\mathbf{x}^\star \geq t\,\mathbf{c}^\mathsf{T}\mathbf{x}^\star = t\gamma.$$

Thus $t > 0$. If $t = 0$, the last two inequalities would give both $\mathbf{b}^\mathsf{T}\mathbf{u} \geq 0$ and $\mathbf{b}^\mathsf{T}\mathbf{u} < 0$.

Normalizing to a dual solution. Set $\mathbf{y}_\varepsilon = \mathbf{u}/t$. Then

$$\mathbf{y}_\varepsilon \geq \mathbf{0}, \qquad A^\mathsf{T}\mathbf{y}_\varepsilon \geq \mathbf{c}, \qquad \gamma \leq \mathbf{b}^\mathsf{T}\mathbf{y}_\varepsilon < \gamma + \varepsilon.$$

The dual is therefore feasible, and weak duality together with the last bound shows that its infimum equals γ. Its objective is bounded below on the feasible polyhedron, so the fundamental theorem of linear programming guarantees that the minimum is attained. Hence there is an optimal dual solution $\mathbf{y}^\star$ with

$$\mathbf{b}^\mathsf{T}\mathbf{y}^\star = \gamma = \mathbf{c}^\mathsf{T}\mathbf{x}^\star.$$

The reverse direction and the remaining alternatives. We have proved that an optimal solution of (P) gives an optimal solution of (D) with the same value. Conversely, if (D) is feasible and its objective is bounded below, it has an optimal solution. Write it as a standard maximization problem

$$\max\ (-\mathbf{b})^\mathsf{T}\mathbf{y} \quad \text{subject to} \quad (-A^\mathsf{T})\mathbf{y} \leq -\mathbf{c}, \quad \mathbf{y} \geq \mathbf{0}.$$

Applying the result just proved gives an optimum of its dual, which becomes (P) after reversing the sign of the objective. This is the fact that the dual of the dual is the original problem. Weak duality rules out a dual feasible solution when (P) is unbounded above and, symmetrically, rules out a primal feasible solution when (D) is unbounded below. If one problem is infeasible and the other is feasible, the latter cannot be bounded in the relevant direction, since the result just proved would then imply feasibility of the former. This establishes all four alternatives of strong duality without using a simplex tableau.

3 Shadow prices and right-hand-side sensitivity

Dual variables are an important and highly intuitive output of linear programming. Primal variables describe concrete decisions, such as how much to produce, where to route material, or which activities to run. Dual variables price the constraints: the resources, capacities, and limits in the model.

For a capacity constraint, an optimal dual variable $y_i^\star$ is often called a *shadow price*. Suppose, for example, that b_i is the number of hours available on a machine. The shadow price answers the question: what is one additional hour worth at the current optimum? The word "current" matters. The answer is local and remains valid only while the optimal basis stays the same.

Let us state this precisely. After conversion to equality form, denote its matrix by $\widetilde{A}$, the objective coefficient vector by $\widetilde{\mathbf{c}}$, and an optimal basis satisfying $\mathbf{p} \geq \mathbf{0}$, $\mathbf{r} \leq \mathbf{0}$ by B. Set

$$\mathbf{y}^\star = \widetilde{A}_B^{-\mathsf{T}}\widetilde{\mathbf{c}}_B.$$

If only the right-hand side changes from $\mathbf{b}$ to $\mathbf{b}+\Delta\mathbf{b}$, the reduced costs remain unchanged. If

$$\widetilde{A}_B^{-1}(\mathbf{b} + \Delta\mathbf{b}) \geq \mathbf{0},$$

the same basis remains feasible and therefore optimal. Within this region, the following relation holds *exactly*:

$$\begin{aligned} v(\mathbf{b}+\Delta\mathbf{b}) &= \widetilde{\mathbf{c}}_B^{\mathsf{T}}\widetilde{A}_B^{-1}(\mathbf{b}+\Delta\mathbf{b}) \\ &= v(\mathbf{b}) + (\mathbf{y}^\star)^{\mathsf{T}}\Delta\mathbf{b}. \end{aligned}$$

This is not merely a rough estimate. The dependence is exactly linear as long as the same basis remains feasible. At the boundary of this region, the optimal basis may change, and the shadow price may change with it.

Complementary slackness from the preceding chapter gives

$$b_i - \mathbf{a}_i^{\mathsf{T}}\mathbf{x}^\star > 0 \implies y_i^\star = 0, \qquad y_i^\star > 0 \implies \mathbf{a}_i^{\mathsf{T}}\mathbf{x}^\star = b_i.$$

This yields a practical rule: a resource with unused capacity has zero shadow price. If its shadow price is positive, the constraint is active and the resource is fully used. The converses do not hold in general. An active constraint can have zero shadow price, for example because of redundancy or degeneracy. Likewise, $y_i^\star = 0$ alone does not mean that the constraint has slack or can be deleted.

In the example from the preceding chapter,

$$\mathbf{y}^\star = \left(\tfrac{5}{16}, 0, \tfrac{1}{4}\right)^{\mathsf{T}}.$$

As long as the same basis remains feasible, changes in the right-hand sides give

$$\Delta v = \tfrac{5}{16}\Delta b_1 + \tfrac{1}{4}\Delta b_3.$$

The second constraint has zero local price for this basis. This short relation is a direct "what-if" model: while the basis remains feasible, we need not solve the entire problem again after every small change.

Remark 8.5: Interpreting shadow prices

A positive shadow price identifies a resource whose small expansion can improve the optimum. It is not automatically an instruction to invest. Comparisons of $y_i^\star$ across resources must account for their physical and economic units, the cost of obtaining another unit, and the interval in which the current basis remains feasible. The numerically largest shadow price alone does not identify the best investment.

Remark 8.6: When optimal prices are not unique

The optimal dual solution need not be unique. Different optimal vectors $\mathbf{y}^\star$ have the same total value for the current right-hand side, but may assign different prices to individual resources. With multiple dual optima, we therefore cannot speak of a single "correct" shadow price or mechanically predict the change in an arbitrary direction without further analysis. For our purposes, the formula associated with a particular optimal basis and its range of feasibility is sufficient. A more general description belongs to sensitivity analysis and convex optimization (Boyd and Vandenberghe, 2004, Rockafellar, 1970).

The dual vector can always be recovered safely from the tableau using

$$(\mathbf{y}^\star)^{\mathsf{T}} = \widetilde{\mathbf{c}}_B^{\mathsf{T}}\widetilde{A}_B^{-1}.$$

In standard form with slack columns I_m, the reduced cost of slack variable i equals $-y_i^\star$. Its position in a displayed tableau depends on the format. Without specifying the convention, it cannot generally be identified as the "second-to-last row."

4 The dual simplex method

The classical primal simplex method starts in the "right place," at a primal feasible basis. It preserves feasibility throughout and progressively eliminates positive reduced costs. The dual simplex method reverses these roles: its starting point may violate some primal constraints, but its reduced costs already have the required signs. Each pivot preserves dual feasibility while correcting negative basic variables, until both conditions hold at an optimum (Lemke, 1954, Dantzig, 1963).

We consistently use the dictionary

$$\boxed{\mathbf{x}_B = \mathbf{p} + Q\mathbf{x}_N, \qquad z = z_0 + \mathbf{r}^\mathsf{T}\mathbf{x}_N,} \qquad Q = -A_B^{-1}A_N.$$

For a maximization problem,

$$\mathbf{p} \geq \mathbf{0} \quad \text{means primal feasibility,} \qquad \mathbf{r} \leq \mathbf{0} \quad \text{means dual feasibility.}$$

If both hold, the basic solution is optimal, since every feasible $\mathbf{x}_N \geq \mathbf{0}$ satisfies $z = z_0 + \mathbf{r}^\mathsf{T}\mathbf{x}_N \leq z_0$.

4.1 Deriving the pivot rule

In primal simplex, we first chose an improving column and then used the ratio test to find the row that stopped us. Here we reverse the order. First, choose a violated row, that is, a negative basic variable. Then find a column that can correct it without destroying dual feasibility.

Let $p_s < 0$. The row of the leaving basic variable is

$$x_{B_s} = p_s + \sum_{j \in N} q_{sj} x_j.$$

To correct the negative right-hand side by increasing a nonbasic variable, the entering index must satisfy $q_{sj} > 0$. If $q_{sj} \leq 0$ for every $j \in N$, then for every $\mathbf{x}_N \geq \mathbf{0}$,

$$x_{B_s} \leq p_s < 0,$$

and the original problem is infeasible.

For a candidate v with $q_{sv} > 0$, solve the pivot row for x_v:

$$x_v = -\frac{p_s}{q_{sv}} + \frac{1}{q_{sv}} x_{B_s} - \sum_{j \in N \setminus \{v\}} \frac{q_{sj}}{q_{sv}} x_j.$$

Substitution into the objective function gives the new reduced costs

$$r'_{B_s} = \frac{r_v}{q_{sv}}, \qquad r'_j = r_j - \frac{r_v}{q_{sv}} q_{sj} \quad (j \in N \setminus \{v\}).$$

Since initially $\mathbf{r} \leq \mathbf{0}$, components with $q_{sj} \leq 0$ automatically remain nonpositive. For all columns with $q_{sj} > 0$, we need

$$\frac{r_v}{q_{sv}} \geq \frac{r_j}{q_{sj}}.$$

The dual ratio test therefore chooses the *largest*, not the smallest, ratio r_j/q_{sj}. This is a common source of sign errors: always derive the rule from the actual dictionary $\mathbf{x}_B = \mathbf{p} + Q\mathbf{x}_N$, not from a tableau using a different sign convention.

One step of the dual simplex method

Assume $\mathbf{r} \leq \mathbf{0}$.

1. If $\mathbf{p} \geq \mathbf{0}$, the current basic solution is optimal.
2. Otherwise, choose a row s with $p_s < 0$, for example the most negative right-hand side.
3. If $q_{sj} \leq 0$ for every $j \in N$, the problem is infeasible.
4. Otherwise, choose
$$v \in \arg\max_{j \in N : q_{sj} > 0} \frac{r_j}{q_{sj}}$$
and pivot: x_v enters the basis and x_{B_s} leaves.
5. Repeat from the first step.

With ties and degeneracy, an anti-cycling rule is needed, such as an appropriate dual version of Bland's rule or lexicographic pivoting. Choosing the most negative right-hand side alone does not guarantee termination.

4.2 A complete numerical example

A short example checks both parts of the rule: a candidate column must have a positive coefficient in the violated row, and among the candidates we choose the largest ratio r_j/q_{sj}.

Example 8.7: One step of the dual simplex method

Consider the equality-form problem

$$\begin{aligned} \textbf{maximize} \quad & z = -x_3 - 3x_4, \\ & x_1 - x_3 - 2x_4 = -1, \\ & x_2 + x_3 - x_4 = 3, \\ & x_1, x_2, x_3, x_4 \geq 0. \end{aligned}$$

For the basis $B = \{1, 2\}$, the dictionary is

$$x_1 = -1 + x_3 + 2x_4, \qquad x_2 = 3 - x_3 + x_4, \qquad z = -x_3 - 3x_4.$$

The reduced-cost vector $(-1, -3)$ is nonpositive, so the basis is dual feasible. It is not primal feasible, since $p_1 = -1$.
Both candidate coefficients in the first row are positive. The ratios are

$$\frac{r_3}{q_{13}} = -1, \qquad \frac{r_4}{q_{14}} = -\frac{3}{2}.$$

Choose the larger ratio: x_3 enters and x_1 leaves. The first row gives

$$x_3 = 1 + x_1 - 2x_4.$$

Substitution into the other rows yields

$$x_3 = 1 + x_1 - 2x_4, \qquad x_2 = 2 - x_1 + 3x_4, \qquad z = -1 - x_1 - x_4.$$

The new basis $\{x_3, x_2\}$ is primal feasible because its right-hand sides are $(1, 2)$, and remains dual feasible because its new reduced costs are $(-1, -1)$. The dictionary is therefore optimal:

$$(x_1, x_2, x_3, x_4) = (0, 2, 1, 0), \qquad z^\star = -1.$$

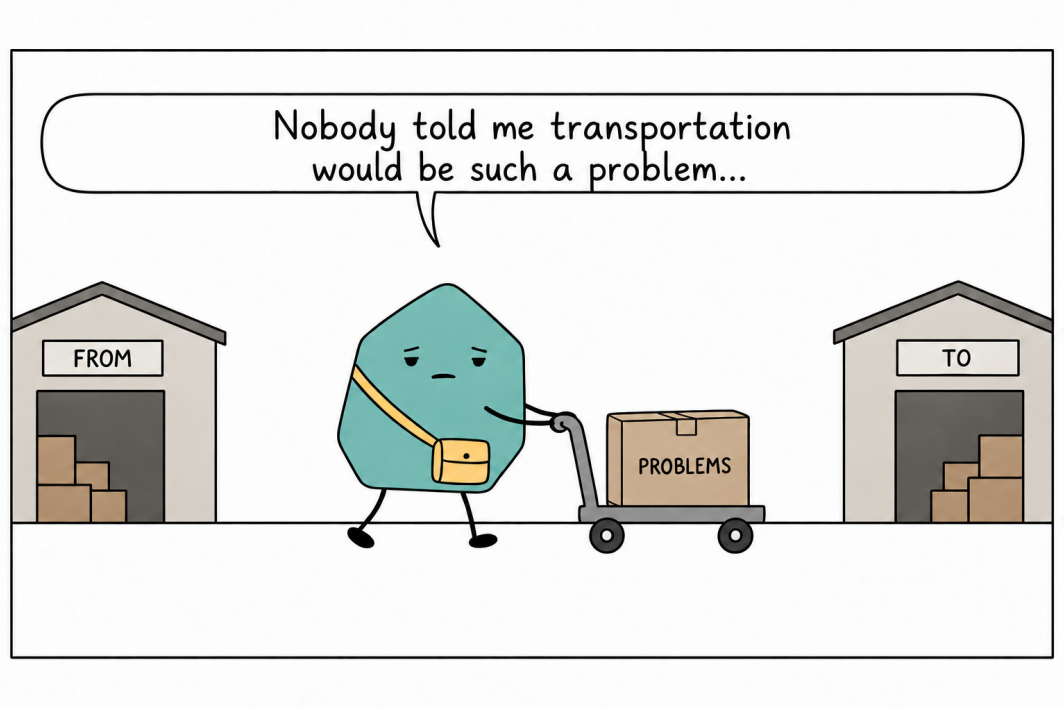
The example also shows why the signs matter: the candidates are columns with $q_{1j} > 0$. The opposite condition would incorrectly declare this feasible problem infeasible.

A dual feasible basis is often naturally available during reoptimization. Suppose a production model has already been solved optimally and we add a new capacity constraint. The reduced costs may remain nonpositive while the new right-hand side violates primal feasibility. Dual simplex can then reuse the previous work, restoring feasibility from the original optimal basis. If the starting basis is not dual feasible, it must first be obtained by another valid procedure. Dual simplex alone does not solve this initialization problem.

CHAPTER 9

THE TRANSPORTATION PROBLEM

The transportation problem seeks the least expensive way to ship goods from suppliers to customers, given their supplies and demands. We formulate it as a linear program and examine its constraint structure, basic feasible solutions, and degeneracy. We then introduce the *modified distribution method* (MODI) for finding an optimal solution.



1 Problem description and notation

The transportation problem is both a classical model and one of the most useful applications of linear programming. It arises naturally whenever resources must be allocated efficiently to customers while minimizing the associated costs, from logistics and distribution to production planning and energy allocation. We begin by formulating the problem.

Consider m suppliers (sources) $S_1, \ldots, S_m$ and n customers $D_1, \ldots, D_n$, where $m, n \geq 1$. Supplier S_i has an available quantity $s_i \geq 0$ (capacity or supply), and customer D_j requires a quantity $d_j \geq 0$ (demand). For each pair (i, j), the unit transportation cost $c_{ij} \geq 0$ is given.

We denote the decision variables of the transportation problem by

$$x_{ij} \ \geq \ 0, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n,$$

and interpret x_{ij} as the quantity shipped from source S_i to customer D_j.

Remark 9.1: Description of the transportation problem

Find allocations $\{x_{ij}\}$ such that

- the total shipment from source S_i does not exceed its available supply s_i,
- the total quantity received by customer D_j covers its demand d_j,
- the total transportation cost is minimized.

In the usual pure transportation model, we assume that *all supply must be shipped* and *all demand must be met*. Formally, this means

$$\sum_{j=1}^{n} x_{ij} = s_i, \qquad i = 1, \ldots, m,$$

$$\sum_{i=1}^{m} x_{ij} = d_j, \qquad j = 1, \ldots, n.$$

2 Formulating the transportation problem as an LP

2.1 The canonical (balanced) formulation

We begin with the standard case in which total supply equals total demand:

$$\sum_{i=1}^{m} s_i \ = \ \sum_{j=1}^{n} d_j.$$

Such a problem is called a *balanced transportation problem*.

LP formulation of the transportation problem

For a balanced transportation problem, we solve the minimization problem

$$\begin{aligned} \textbf{minimize} \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \\ \textbf{subject to} \quad & \sum_{j=1}^{n} x_{ij} = s_i, \qquad i = 1,\dots,m, \\ & \sum_{i=1}^{m} x_{ij} = d_j, \qquad j = 1,\dots,n, \\ & x_{ij} \geq 0 \qquad \text{for all } i,j. \end{aligned} \tag{9.1}$$

This is a linear program with mn variables and $m+n$ equations. As we will see below, these equations are not all independent.

2.2 The unbalanced transportation problem

The balanced transportation problem has many useful properties. In practice, however, it is common to have

$$\sum_{i=1}^{m} s_i \neq \sum_{j=1}^{n} d_j.$$

Total supply may exceed total demand, leaving part of the capacity unused, or the available supply may be insufficient to meet all requirements. Such a problem is called an *unbalanced transportation problem*. By adding a dummy row or column, we can obtain a balanced table for computational purposes. We must first specify exactly what the dummy allocations represent. In particular, when supply is insufficient, this changes the original model rather than merely rewriting it.

Remark 9.2: Balancing a transportation problem

- If $\sum_i s_i > \sum_j d_j$, add a *dummy customer* D_{n+1} with demand

$$d_{n+1} := \sum_{i=1}^{m} s_i - \sum_{j=1}^{n} d_j$$

  and unit costs $c_{i,n+1} = 0$. The variable $x_{i,n+1}$ then represents unused supply at supplier i. This formulation is equivalent to a model in which supplies are upper bounds and all actual demands must be met. If disposing of or storing the surplus incurs a cost, use that cost instead of zero.

- If $\sum_i s_i < \sum_j d_j$, add a *dummy supplier* S_{m+1} with supply

$$s_{m+1} := \sum_{j=1}^{n} d_j - \sum_{i=1}^{m} s_i$$

  and costs $c_{m+1,j}$. The allocation $x_{m+1,j}$ represents the unmet part of customer

j's demand, so $c_{m+1,j}$ is usually chosen as the corresponding penalty per unit of shortage.

In the second case, this is not an equivalent reformulation of the original model requiring all demands to be met. That model is infeasible when supply is insufficient. The dummy supplier changes the model by explicitly allowing shortages and assigning them a chosen penalty. Zero costs would mean that unmet demand carries no penalty.

In what follows, we analyze the balanced problem (9.1). These methods apply to an unbalanced model only after the additions described above, with the meaning and costs of the dummy allocations clearly specified.

3 Structure and properties of the transportation problem

The constraint matrix in (9.1) has a very special structure. Its first m rows describe the row sums of the transportation table, and its next n rows describe the column sums. This has several consequences, which we examine in this section.

3.1 Independent constraints and basic solutions

First, we show that some of the constraints must be linearly dependent. Adding all row (supply) constraints gives

$$\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} = \sum_{i=1}^{m} s_i,$$

whereas adding all column (demand) constraints gives

$$\sum_{j=1}^{n}\sum_{i=1}^{m} x_{ij} = \sum_{j=1}^{n} d_j.$$

The two left-hand sides are identical, with the same double sum written in a different order. In a balanced transportation problem, the right-hand sides also agree: $\sum_i s_i = \sum_j d_j$.

To establish the rank, let $\mathbf{r}_i$ denote the coefficient row of the ith supply constraint and $\mathbf{c}_j$ the coefficient row of the jth demand constraint. Both types of rows belong to $\mathbb{R}^{mn}$. The right-hand sides s_i, d_j are not part of the matrix A. The preceding observation gives

$$\sum_{i=1}^{m} \mathbf{r}_i - \sum_{j=1}^{n} \mathbf{c}_j = \mathbf{0}.$$

This is a nontrivial linear combination of rows of A that equals zero, and hence

$$\operatorname{rank}(A) \;\le\; m + n - 1.$$

We now verify that there is no further linear dependence among the constraints. Consider a linear combination

$$\sum_{i=1}^{m} \alpha_i \mathbf{r}_i + \sum_{j=1}^{n-1} \beta_j \mathbf{c}_j = \mathbf{0}.$$

Each variable x_{ij} has a nonzero coefficient precisely in row $\mathbf{r}_i$ and, if $j < n$, in row $\mathbf{c}_j$. Its coefficient in this linear combination is therefore

$$\alpha_i + \beta_j \qquad (\text{for } \; j < n).$$

For the variables x_{in}, whose row $\mathbf{c}_n$ we omitted, the coefficient is just α_i. For the entire linear combination to be zero, every one of these coefficients must vanish. The variables x_{in} thus give

$$\alpha_1 = \alpha_2 = \cdots = \alpha_m = 0,$$

and substituting into the remaining coefficients gives

$$\beta_1 = \beta_2 = \cdots = \beta_{n-1} = 0.$$

The only linear combination of these $m+n-1$ equations that equals zero is therefore the trivial one. Thus these $m + n - 1$ equations are linearly independent.

We have established that $\operatorname{rank}(A) = m + n - 1$. To make the notion of a basis agree exactly with the definition in earlier chapters, we remove, for example, the last demand equation. Write the resulting matrix as $\bar{A} \in \mathbb{R}^{(m+n-1)\times mn}$ and its right-hand side as $\bar{\mathbf{b}}$. In a balanced problem, the omitted equation follows from all supply equations and the first $n - 1$ demand equations. Hence

$$A\mathbf{x} = \mathbf{b} \quad \Longleftrightarrow \quad \bar{A}\mathbf{x} = \bar{\mathbf{b}}.$$

The matrix $\bar{A}$ has full row rank $m + n - 1$. By a basis of the transportation problem, we therefore mean a selection of $m + n - 1$ linearly independent columns of this reduced matrix. The precise feasibility criterion is now particularly simple.

Theorem 9.3: Feasibility of the transportation problem

Assume $s_i, d_j \geq 0$. For the transportation problem with equality constraints (9.1), the following conditions are equivalent:

1. The problem has a basic feasible solution.
2. The problem has an optimal solution.
3. The transportation problem is balanced, that is,

$$\sum_{i=1}^{m} s_i \; = \; \sum_{j=1}^{n} d_j.$$

Proof of the theorem.

Adding the row and column equations shows that every feasible solution necessarily satisfies

$$\sum_i s_i = \sum_j d_j.$$

Conversely, if the problem is balanced, the northwest corner rule described below constructs a basic feasible solution of the equivalent reduced system. The feasible set is closed and bounded, since $0 \leq x_{ij} \leq \min\{s_i, d_j\}$. The linear objective function therefore attains its minimum on this set. Finally, if an optimal solution exists, the

fundamental theorem of linear programming guarantees the existence of an optimal basic feasible solution. ■

For a more detailed treatment of the network structure of the transportation problem and its bases, see (Bazaraa et al., 2010, Ahuja et al., 1993).

Let us examine the constraint matrix in more detail. It has dimensions $(m+n) \times mn$ and a natural block structure

$$A = \begin{pmatrix} A^{(S)} \\ A^{(D)} \end{pmatrix}, \qquad A^{(S)} \in \mathbb{R}^{m \times mn},\ A^{(D)} \in \mathbb{R}^{n \times mn},$$

where the block $A^{(S)}$ contains the supply constraints

$$\sum_{j=1}^{n} x_{ij} = s_i, \quad i = 1, \ldots, m,$$

and the block $A^{(D)}$ contains the demand constraints

$$\sum_{i=1}^{m} x_{ij} = d_j, \quad j = 1, \ldots, n.$$

Each cell of the table, corresponding to a variable x_{ij}, represents one column of A. This column contains exactly two ones, one in the row of supplier S_i and one in the row of customer D_j. All other entries are zero.

For example, with three suppliers and three customers, we obtain

$A =$

	x_{11}	x_{12}	x_{13}	x_{21}	x_{22}	x_{23}	x_{31}	x_{32}	x_{33}
S_1	1	1	1	0	0	0	0	0	0
S_2	0	0	0	1	1	1	0	0	0
S_3	0	0	0	0	0	0	1	1	1
D_1	1	0	0	1	0	0	1	0	0
D_2	0	1	0	0	1	0	0	1	0
D_3	0	0	1	0	0	1	0	0	1

The first three rows state that the row sums of the table equal the supplies s_1, s_2, s_3. The next three rows state that the column sums equal the demands d_1, d_2, d_3.

It is useful to view the transportation problem as a network model on a bipartite graph. The vertices represent the sources S_i and customers D_j, and the edge joining S_i to D_j represents the variable x_{ij}.

Remark 9.4: Bipartite graphs

A bipartite graph is a graph whose vertex set can be partitioned into two disjoint sets U and V such that every edge joins a vertex in U to a vertex in V. Thus there are no edges between vertices within the same part.
In a transportation problem, the two natural parts are the set of sources $\{S_1, \ldots, S_m\}$ and the set of customers $\{D_1, \ldots, D_n\}$. The structure of the transportation table is therefore precisely that of a bipartite graph.

Remark 9.5: The tree structure of a basis

Here the abstract notion of a basis has a particularly concrete interpretation. Let F be a set of cells in the transportation table, viewed as a set of edges of the bipartite graph with vertices $\{S_1, \ldots, S_m\} \cup \{D_1, \ldots, D_n\}$. The columns of the reduced matrix $\bar{A}$ indexed by F are linearly independent if and only if F contains no cycle. Indeed, multiplying all demand rows by -1 gives a reduced vertex-edge incidence matrix. The columns of a cycle have a nontrivial alternating linear dependence. Conversely, independence for a forest follows by successively removing leaves. If F has exactly $m + n - 1$ edges, such a forest is connected and is therefore a spanning tree of the bipartite graph. The bases of the transportation problem thus correspond exactly to spanning trees (Bazaraa et al., 2010, Ahuja et al., 1993).

It is therefore natural to organize both the solution process and the search for an initial basic feasible solution in a table. Columns represent customers and rows represent suppliers. In a nondegenerate solution, we can enter the positive basic allocations in their cells and leave the nonbasic cells blank. In a degenerate solution, however, a cell with zero allocation may also be basic. Such cells must be marked explicitly in the table so that the basis is unambiguously specified.

	D_1	D_2	D_3	supply
S_1	c_{11} x_{11}	c_{12} x_{12}	c_{13} x_{13}	s_1
S_2	c_{21} x_{21}	c_{22} x_{22}	c_{23} x_{23}	s_2
S_3	c_{31} x_{31}	c_{32} x_{32}	c_{33} x_{33}	s_3
demand	d_1	d_2	d_3	$d_1 + d_2 + d_3$

3.2 Degeneracy

As in a general LP, a basic feasible solution of a transportation problem may be *degenerate*.

Definition 9.6: Degeneracy in the transportation problem

A basic feasible solution of the transportation problem is *degenerate* if

$$\#\{(i,j) \mid x_{ij} > 0\} < m + n - 1.$$

Otherwise, it is *nondegenerate*.

Degeneracy means that at least one of the $m + n - 1$ basic cells has zero allocation. Such a cell can easily disappear from a drawing of the table, but we must continue to

keep track of the basis as a tree with exactly $m + n - 1$ edges. In a hand calculation, a zero basic cell can be marked 0_B. A symbolic ε is only a bookkeeping device and must not be confused with an actual positive shipment. A degenerate pivot may have step length $\theta = 0$. The basis changes, but the shipping plan and objective value do not. To guarantee finite termination, we then need an anti-cycling rule for selecting entering and leaving cells, such as the lexicographic rule.

4 The MODI method (method of potentials)

We could, of course, solve the transportation problem with the general simplex method. Its regular table structure, however, allows specialized algorithms to work directly with the data much more economically. One such algorithm is the *modified distribution method* (MODI), also called the *method of potentials* or the *u–v method*. It is a specialized network simplex method (Bazaraa et al., 2010, Ahuja et al., 1993).

4.1 An initial basic feasible solution

The MODI method starts from a BFS of the transportation problem. In practice, an initial BFS can be obtained using one of several simple constructions:

- the *northwest corner rule*,
- the *least-cost method*,
- *Vogel's approximation method*.

The simplest is the northwest corner rule. Its name describes almost the entire procedure: start in the upper left corner of the table and move right and down. Zero supplies and demands can be marked as exhausted immediately. In the first cell whose row and column have not yet been exhausted, proceeding from left to right and from top to bottom, assign

$$x_{ij} = \min\{\text{remaining supply in row } i, \text{remaining demand in column } j\}.$$

If this exhausts the row's supply, move down one row. If it satisfies the column's demand, move right one column. If both occur at the same time, move in both directions and add a zero basic cell later. In the new cell, again allocate the minimum of the remaining supply and demand. Continue until all sums are satisfied. The cells with positive allocations form a forest. If there are fewer than $m + n - 1$ of them, for example because a supply and a demand were exhausted simultaneously, or because some supplies or demands are zero, add zero basic cells without creating a cycle until a spanning tree with $m + n - 1$ edges is obtained. In a complete bipartite graph, the components of the forest can always be connected in this way. The result is a basis and a BFS of the balanced problem. The only weakness of this very simple construction is that it completely ignores costs when choosing cells, so the initial plan need not be a good one.

4.2 Potentials and reduced costs

For a transportation problem, it is convenient to write a basis as a set of index pairs, or equivalently, as a set of cells in the table.

Suppose we have a basic feasible solution with basic cells $B \subseteq \{1,\dots,m\} \times \{1,\dots,n\}$, where $|B| = m + n - 1$. In the degenerate case, some basic allocations x_{ij} may equal 0.

The MODI method introduces two sets of auxiliary quantities:

$$u_1,\dots,u_m \quad \text{(row potentials)}, \qquad v_1,\dots,v_n \quad \text{(column potentials)}.$$

Choose the potentials u_i and v_j so that for every cell $(i,j) \in B$,

$$u_i + v_j = c_{ij}. \tag{9.2}$$

The basic cells form a tree in a bipartite graph with $m + n$ vertices, so this tree has exactly $m + n - 1$ edges. The same conclusion follows from the linear independence argument above. Thus the system (9.2) has $m + n - 1$ equations in $m + n$ unknowns $u_1,\dots,u_m,v_1,\dots,v_n$. The potentials are determined up to the transformation

$$u_i \longmapsto u_i + \lambda, \qquad v_j \longmapsto v_j - \lambda, \qquad \lambda \in \mathbb{R},$$

which preserves every sum $u_i + v_j$. We usually fix one potential, for example $u_1 := 0$, and then determine all remaining potentials uniquely from (9.2) by following the edges of the tree.

Remark 9.7: Why the decomposition $c_{ij} = u_i + v_j$?

In the dual of the transportation problem, u_i and v_j are the dual variables for the supply equations at S_i and the demand equations at D_j. Each dual inequality has the form $u_i + v_j \le c_{ij}$, because the column of the constraint matrix corresponding to x_{ij} has exactly two ones, one in row S_i and one in row D_j. The equalities $u_i + v_j = c_{ij}$ on the chosen basis define the simplex multipliers. If the computed potentials satisfy all dual inequalities, they are dual feasible, and duality proves that the current plan is optimal. At optimality, the equalities on basic cells also agree with the complementary slackness conditions.

Once the potentials u_i, v_j are known, define the *reduced cost* of every cell, including nonbasic cells, by

$$\Delta_{ij} \;:=\; c_{ij} - (u_i + v_j). \tag{9.3}$$

The MODI optimality criterion

Consider the transportation minimization problem (9.1) and a basic feasible solution with potentials u_i, v_j and reduced costs Δ_{ij} defined by (9.2)–(9.3). Then:

- if $\Delta_{ij} \ge 0$ for all cells, in particular all nonbasic cells, the given basic solution is *optimal*,
- if a nonbasic cell has $\Delta_{ij} < 0$, the corresponding direction does not increase cost. If the feasible step length is $\theta > 0$, total cost strictly decreases. A degenerate step may have $\theta = 0$, in which case only the basis changes.

Intuitively, Δ_{ij} is the change in cost per unit if we add a small shipment to cell (i,j) and adjust the other allocations so that all row and column sums remain unchanged. Nonnegative reduced costs therefore say that no local adjustment can lower cost, which characterizes an optimum.

Remark 9.8: Connection with the simplex method

In a general LP, reduced costs describe how the objective value changes when a nonbasic variable enters the basis (see the chapter on the simplex method). The MODI method is essentially a simplex method tailored to the special structure of transportation tables. The potentials u_i, v_j act as dual variables and Δ_{ij} are the reduced costs.

4.3 An improving step and a cycle in the table

Suppose that, for the current basic solution, at least one cell (p, q) has $\Delta_{pq} < 0$. We want to bring cell (p, q) into the basis, corresponding to increasing x_{pq} from zero to a positive value. To preserve all row and column sums, we must adjust the other allocations.

In the bipartite graph, adding the edge (p, q) creates a *unique cycle*. In the table, this appears as a closed polygonal path through an even number of cells, alternating between horizontal and vertical steps. Every cell on the cycle except the entering cell is basic. In a degenerate solution, some of these cells may have zero allocation.

The procedure is as follows:

1. Choose a nonbasic cell (p, q) with negative reduced cost $\Delta_{pq} < 0$ (for example, one with the smallest value) and temporarily add it to the tree of basic cells.

2. Find the closed cycle through (p, q) in the table. Its steps alternate between horizontal and vertical, with every turn other than the entering cell at a basic cell.

3. Mark cells on the cycle alternately with $+$ and $-$, starting with $+$ at (p, q).

4. Compute
$$\theta \;:=\; \min\{x_{ij} \mid (i, j) \text{ is a minus cell on the cycle}\},$$
that is, the smallest allocation among basic cells marked $-$. This may give $\theta = 0$.

5. Increase the allocation in every $+$ cell by θ and decrease the allocation in every $-$ cell by θ. At least one minus cell now has zero allocation and leaves the basis. If several cells attain the minimum, choose exactly one to leave, keeping the other zero cells basic. Cell (p, q) becomes basic with allocation θ.

The objective value changes by exactly $\theta\Delta_{pq}$ along this cycle. Thus it strictly decreases when $\Delta_{pq} < 0$ and $\theta > 0$. When $\theta = 0$, it stays unchanged.

Example 9.9: Applying the MODI method

Consider a transportation problem with three sources S_1, S_2, S_3 and three customers D_1, D_2, D_3. The supplies, demands, and unit costs c_{ij} are

$$(c_{ij}) = \begin{pmatrix} 6 & 8 & 10 \\ 7 & 11 & 11 \\ 4 & 5 & 12 \end{pmatrix}, \qquad (s_1, s_2, s_3) = (15, 17, 18), \qquad (d_1, d_2, d_3) = (20, 10, 20).$$

We seek a shipping plan that minimizes total cost.
Suppose a construction heuristic, such as the least-cost method, has already pro-

vided an initial basic feasible solution with allocations

$$x_{11} = 2, \quad x_{12} = 10, \quad x_{13} = 3, \quad x_{23} = 17, \quad x_{31} = 18,$$

and all other x_{ij} equal to zero. This solution is shown in the following table. The cost c_{ij} appears at the top of each cell, and the allocated quantity x_{ij} appears at the lower left.

	D_1	D_2	D_3	supply	u_i
S_1	6 2	8 10	10 3	15	$u_1 = 0$
S_2	7	11	11 17	17	$u_2 = 1$
S_3	4 18	5	12	18	$u_3 = -2$
demand	20	10	20	50	
v_j	$v_1 = 6$	$v_2 = 8$	$v_3 = 10$		$z = 381$

The basic cells are

$$B = \{(1,1), (1,2), (1,3), (2,3), (3,1)\}.$$

For every basic cell, the potentials satisfy

$$u_i + v_j = c_{ij}, \qquad (i,j) \in B.$$

This gives a system of five equations:

$$\begin{aligned} u_1 + v_1 &= 6, \\ u_1 + v_2 &= 8, \\ u_1 + v_3 &= 10, \\ u_2 + v_3 &= 11, \\ u_3 + v_1 &= 4. \end{aligned}$$

Since the potentials are determined only up to the transformation $u_i \mapsto u_i + \lambda$, $v_j \mapsto v_j - \lambda$, we can fix $u_1 := 0$. Solving successively gives

$$v_1 = 6, \quad v_2 = 8, \quad v_3 = 10, \quad u_2 = 1, \quad u_3 = -2,$$

as recorded in the table above.

For the nonbasic cells, we compute the reduced costs $\Delta_{ij} = c_{ij} - (u_i + v_j)$:

$$\begin{aligned}
\Delta_{21} &= c_{21} - (u_2 + v_1) = 7 - (1 + 6) = 0,\\
\Delta_{22} &= c_{22} - (u_2 + v_2) = 11 - (1 + 8) = 2,\\
\Delta_{32} &= c_{32} - (u_3 + v_2) = 5 - ((-2) + 8) = -1,\\
\Delta_{33} &= c_{33} - (u_3 + v_3) = 12 - ((-2) + 10) = 4.
\end{aligned}$$

The only negative reduced cost is $\Delta_{32} = -1$, so we try to bring cell $(3, 2)$ into the basis.

The improving cycle. We find the cycle through cell $(3, 2)$ that alternates between basic cells in rows and columns. It is

$$(3,2)\ (+)\ \rightarrow\ (3,1)\ (-)\ \rightarrow\ (1,1)\ (+)\ \rightarrow\ (1,2)\ (-)\ \rightarrow\ (3,2),$$

where the sign in parentheses is the sign assigned to the cell. Let $\theta > 0$ be the amount by which we increase the allocation in $(3, 2)$. We must then decrease allocations in minus cells by θ and increase allocations in plus cells by θ. The resulting allocations are shown as functions of θ in the following table.

	D_1	D_2	D_3	supply	u_i
S_1	6 $2 + \theta$	8 $10 - \theta$	10 3	15	$u_1 = 0$
S_2	7	11	11 17	17	$u_2 = 1$
S_3	4 $18 - \theta$	5 θ	12	18	$u_3 = -2$
demand	20	10	20	50	
v_j	$v_1 = 6$	$v_2 = 8$	$v_3 = 10$		$z = 381 - \theta$

For all allocations to remain nonnegative, we need

$$10 - \theta \geq 0, \qquad 18 - \theta \geq 0,$$

that is, $\theta \leq 10$. Since $\Delta_{32} = -1 < 0$, we choose the largest possible step, $\theta^\star = 10$, to obtain the greatest decrease in the objective. Substituting $\theta = 10$ gives the new basic plan

	D_1	D_2	D_3	supply	u_i
S_1	6 12	8	10 3	15	$u_1 = 0$
S_2	7	11	11 17	17	$u_2 = 1$
S_3	4 8	5 10	12	18	$u_3 = -2$
demand	20	10	20	50	
v_j	$v_1 = 6$	$v_2 = 7$	$v_3 = 10$		$z = 371$

Using the new basic cells

$$B' = \{(1,1), (1,3), (2,3), (3,1), (3,2)\}$$

we recompute the potentials. The equations for u_i, v_j are now

$$\begin{aligned} u_1 + v_1 &= 6, \\ u_1 + v_3 &= 10, \\ u_2 + v_3 &= 11, \\ u_3 + v_1 &= 4, \\ u_3 + v_2 &= 5. \end{aligned}$$

Setting $u_1 := 0$ gives

$$v_1 = 6, \quad v_3 = 10, \quad u_2 = 1, \quad u_3 = -2, \quad v_2 = 7,$$

as shown in the table. The reduced costs of the nonbasic cells are

$$\begin{aligned} \Delta_{12} &= c_{12} - (u_1 + v_2) = 8 - (0 + 7) = 1, \\ \Delta_{21} &= c_{21} - (u_2 + v_1) = 7 - (1 + 6) = 0, \\ \Delta_{22} &= c_{22} - (u_2 + v_2) = 11 - (1 + 7) = 3, \\ \Delta_{33} &= c_{33} - (u_3 + v_3) = 12 - ((-2) + 10) = 4. \end{aligned}$$

All reduced costs are nonnegative, so this basic plan is optimal. Moreover, $\Delta_{21} = 0$, and there is at least one other optimal basic solution.

An alternative optimal solution. Let us bring cell $(2,1)$ into the basis and construct the new cycle

$$(2,1)\ (+)\ \to\ (2,3)\ (-)\ \to\ (1,3)\ (+)\ \to\ (1,1)\ (-)\ \to\ (2,1).$$

Again denoting the change along the cycle by θ, the allocations become

$$x_{11} = 12 - \theta, \quad x_{13} = 3 + \theta, \quad x_{21} = \theta, \quad x_{23} = 17 - \theta.$$

The remaining basic cells $(3,1)$ and $(3,2)$ are unchanged. This gives the table

	D_1	D_2	D_3	supply	u_i
S_1	6 $12-\theta$	8	10 $3+\theta$	15	$u_1 = 0$
S_2	7 θ	11	11 $17-\theta$	17	$u_2 = 1$
S_3	4 8	5 10	12	18	$u_3 = -2$
demand	20	10	20	50	
v_j	$v_1 = 6$	$v_2 = 7$	$v_3 = 10$		$z = 371$

Nonnegativity of the allocations requires

$$12 - \theta \geq 0, \qquad 17 - \theta \geq 0,$$

so $0 \leq \theta \leq 12$. Since $\Delta_{21} = 0$, the objective value remains $z = 371$ for every θ in this interval. For example, $\theta = 12$ gives the alternative optimal plan

$$x_{11} = 0, \quad x_{13} = 15, \quad x_{21} = 12, \quad x_{23} = 5, \quad x_{31} = 8, \quad x_{32} = 10,$$

which satisfies every constraint and has the same minimum cost as the previous solution.

4.4 Summary of the MODI method

The MODI method can be summarized as follows:

The MODI method – an overview

1. Find a basic feasible solution, for example by the northwest corner rule or the least-cost method.
2. Compute potentials u_i, v_j such that $u_i + v_j = c_{ij}$ for every basic cell (i,j), fixing one potential first, for example $u_1 = 0$.
3. Compute the reduced cost $\Delta_{ij} = c_{ij} - (u_i + v_j)$ for each cell.

4. If $\Delta_{ij} \geq 0$ for every (i, j), the current basic solution is optimal. Stop.

5. Otherwise, choose a nonbasic cell (p, q) with $\Delta_{pq} < 0$, usually one with the smallest value. Construct its cycle in the table and take the improving step described above. If several cells attain the minimum, choose one leaving cell according to an anti-cycling rule, such as the lexicographic rule. This produces a new basis and a basic feasible solution. When $\theta = 0$, the shipping plan may stay unchanged.

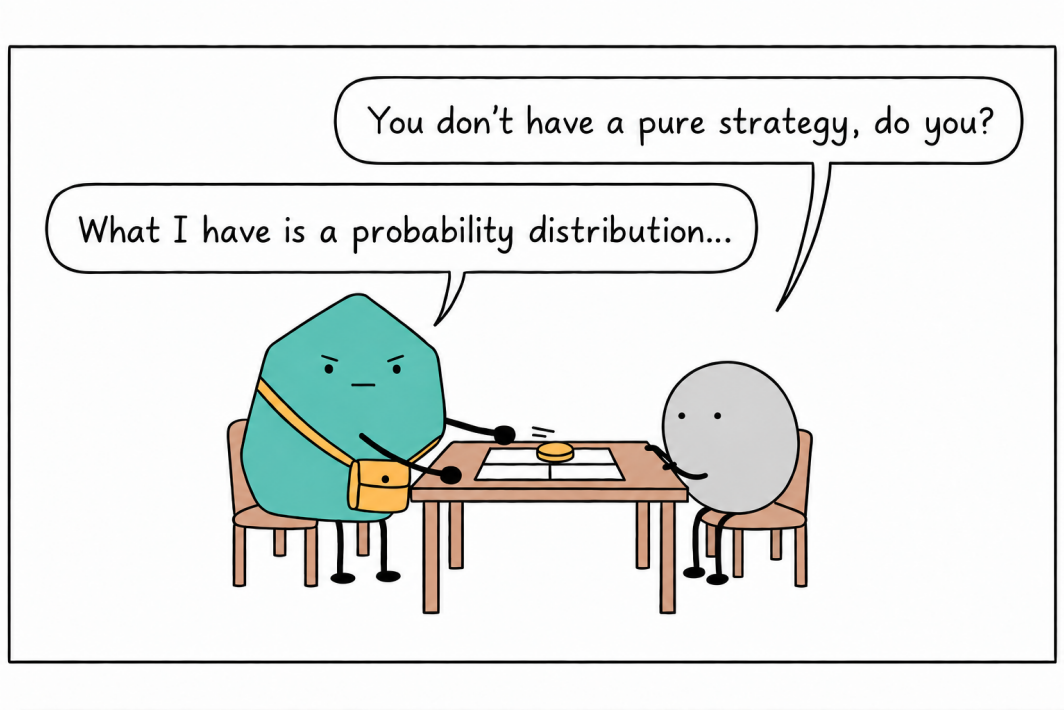
6. Return to step 2.

The MODI method is widely used in practice. It exploits the special structure of the transportation problem, is relatively easy to implement, and is often very efficient for problems with tens or hundreds of rows and columns. Theoretically, however, it is not a fundamentally new algorithm. It is a specialized version of the simplex method that works directly with the allocation table and with potentials corresponding to dual variables.

CHAPTER 10

ZERO-SUM MATRIX GAMES AND THE MINIMAX THEOREM

This chapter focuses on a special but important class of two-player games: *zero-sum games* that can be represented by a single real matrix. We show that finding their optimal mixed strategies reduces to solving a pair of mutually dual linear programs. This provides both an efficient algorithm and a particularly clear connection between game theory and linear programming.



1 Zero-sum matrix games

In this chapter, we show how linear programming arises naturally in a simple model of strategic interaction between two players. Imagine a head-to-head contest: two players choose their actions simultaneously, and the combination of their actions determines how many points, or how much money, one gains and the other loses.

- If we know all available actions for both players and can specify the gains or losses for every pair of actions, we have a *game in normal (strategic) form*.
- If one player's gain is exactly the other player's loss, the game is *zero-sum*.
- If both action sets are finite, the entire game can be represented by one *matrix*. We therefore often call it a *matrix game*.

The following example illustrates a zero-sum matrix game.

Example 10.1: A smuggler and customs patrols

A smuggler is trying to transport illegal goods from Austria to the Czech Republic. There are two possible routes:

- **Route 1** – a secondary road through a lightly traveled border crossing,
- **Route 2** – a highway, where a successful trip would allow much more cargo to be transported.

The customs authority has only one mobile patrol. Each day, it sends the patrol either to the secondary road or to the highway. The patrol does not know which route the smuggler will choose that day, and the smuggler does not know where the patrol will be. They therefore decide *simultaneously*, without knowing the other player's choice.

We use the following notation:

- row player = customs authority,
- column player = smuggler,
- rows: R_1 = patrol on the secondary road, R_2 = patrol on the highway,
- columns: C_1 = smuggler takes the secondary road, C_2 = smuggler takes the highway.

The following table gives the *smuggler's gain*, measured, for example, in units of CZK 10,000:

$A =$

	C_1 (secondary road)	C_2 (highway)
R_1 (patrol on secondary road)	0	6
R_2 (patrol on highway)	2	0

The entry a_{ij} is the smuggler's gain when customs chooses row R_i and the smuggler chooses column C_j. The game is *zero-sum*: the smuggler's gain is the customs authority's loss, so the latter's payoff is $-a_{ij}$.

How should the two players act?

- If customs *always* patrolled the same route, the smuggler would immediately adapt and use the other one.
- If the smuggler *always* used the highway, customs would know to send the patrol there.

It therefore makes sense for both players to *randomize* their choices, using *mixed strategies* to make life harder for the opponent. For this particular game, we will find an equilibrium in which

- the smuggler chooses the secondary road C_1 with probability $\frac{3}{4}$ and the highway C_2 with probability $\frac{1}{4}$,
- customs sends the patrol to the secondary road R_1 with probability $\frac{1}{4}$ and to the highway R_2 with probability $\frac{3}{4}$.

Against this mixed strategy of customs, either pure route gives the smuggler an expected payoff of 3/2. Likewise, against the smuggler's stated strategy, either pure patrol assignment leads to the same expected payment of 3/2. Neither player can therefore improve by changing strategy unilaterally. This balanced outcome follows from the minimax theorem. Later in the chapter, we will see how finding such optimal probabilities can be elegantly formulated as a linear programming problem.

From the perspective of linear programming, these games are interesting because:

1. the players' best (optimal) behavior can be described as the solution of certain optimization problems,
2. those problems can be written as linear programs and their duals, corresponding to the minimax theorem.

Before showing that finding a sensible strategy in this special class of games amounts to solving an LP, we give a precise definition of a zero-sum matrix game.

Definition 10.2: A game in normal form

A *(finite) game in normal form* is a triple

$$G = (P, \mathcal{A}, \mathbf{u}),$$

where $P = \{1, \ldots, N\}$ for some $N \geq 1$, with:

- P the finite set of players,
- $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$ the set of all *action profiles*, where $\mathcal{A}_i$ is a nonempty finite set of actions available to player i,
- $\mathbf{u} = (u_1, \ldots, u_N)$ an N-tuple of *utility (payoff) functions*, $u_i : \mathcal{A} \to \mathbb{R}$, where $u_i(\mathbf{a})$ is player i's utility at the action profile $\mathbf{a} \in \mathcal{A}$.

We now restrict attention to two-player games. For clarity, we call the players the *row player* and the *column player*, and number their actions as

$$R = \{1, \dots, m\}, \qquad C = \{1, \dots, n\}.$$

An action profile is thus a pair (i, j), where i is the chosen row and j the chosen column.

Definition 10.3: A zero-sum game in matrix form

A two-player game in normal form

$$G = (\{\text{row player}, \text{column player}\}, R \times C, \mathbf{u})$$

is a *zero-sum game* if every action profile $(i, j) \in R \times C$ satisfies

$$u_{\text{row}}(i, j) + u_{\text{column}}(i, j) = 0.$$

Since only the transfer between the players matters, the entire game can be described by a single *payoff matrix* $A \in \mathbb{R}^{m \times n}$. Each entry

$$a_{ij} = u_{\text{column}}(i, j)$$

is interpreted as a *payment from the row player to the column player*. A positive a_{ij} means that the column player gains and the row player pays. A negative value means the reverse.

The zero-sum condition immediately implies that the row player's payoff is always $-a_{ij}$. Thus a single matrix A represents the entire game, specifying how much the row player pays the column player for every pair of choices.

Remark 10.4: Matrix games and notation

We use the following convention:

- the *rows* $1, \dots, m$ are the row player's actions,
- the *columns* $1, \dots, n$ are the column player's actions,
- A is the *column player's payoff matrix*.

We deliberately use A for the payoff of the column player, who maximizes it. Some references instead use A for the row player's payoff matrix. Keeping the same row and column actions, one changes between the conventions by $A_{\text{here}} = -A_{\text{row}}$. The directions of the corresponding optimization inequalities then reverse as well. This is the standard matrix model of a two-player zero-sum game.

We can now describe what happens when each player chooses a particular row or column, called a *pure strategy*, meaning a choice of one action from a finite set. Strategically, however, randomizing often makes sense. In rock–paper–scissors, for example, a player who repeats the same move is easy to predict. This leads to the notion of a *mixed strategy*.

2 Mixed strategies and expected payoff

The idea of a mixed strategy is simple. Instead of prescribing "always play row 3," a player prescribes "play row 1 with probability 40 %, row 2 with probability 30 %, and row 3 with probability 30 %." This introduces randomness and prevents the opponent from predicting the choice.

Formally, a mixed strategy is a *probability distribution* on the set of pure strategies.

Definition 10.5: Mixed strategies and the probability simplex

A vector $\mathbf{x} \in \mathbb{R}^n$ is a *stochastic vector*, or *probability vector*, if

$$x_j \geq 0 \quad \text{for all } j \qquad \text{and} \qquad \sum_{j=1}^{n} x_j = 1.$$

We denote the set of all such vectors of length n by

$$S_n = \Big\{ \mathbf{x} \in \mathbb{R}^n \;\Big|\; \mathbf{x} \geq \mathbf{0},\ \sum_{j=1}^{n} x_j = 1 \Big\}.$$

- A *mixed strategy of the column player* is a vector $\mathbf{x} \in S_n$, where x_j is the probability of choosing column j.
- A *mixed strategy of the row player* is a vector $\mathbf{y} \in S_m$, where y_i is the probability of choosing row i.

Remark 10.6: The probability simplex is indeed a simplex

The set S_n is the convex hull of the standard basis vectors $\mathbf{e}_1, \ldots, \mathbf{e}_n \in \mathbb{R}^n$. Indeed, every stochastic vector $\mathbf{x} \in S_n$ satisfies

$$\mathbf{x} = x_1\mathbf{e}_1 + x_2\mathbf{e}_2 + \cdots + x_n\mathbf{e}_n,$$

where the coefficients x_j are nonnegative and sum to 1. Thus S_n consists exactly of the convex combinations of the vertices $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ and is an $(n-1)$-dimensional *simplex*.

A pure strategy is a special case of a mixed strategy: its vector has one entry equal to 1 and all other entries equal to 0. Mixed strategies extend the space of possible strategies by allowing convex combinations of pure strategies.

2.1 Expected payoff

Notation 10.7: Probability and expectation

We use elementary probability notation.

- The symbol $\mathbb{P}[\cdot]$ denotes the *probability* of an event.

- The symbol $\mathbb{E}[\cdot]$ denotes the *expectation* (expected value) of a random variable.

We assume that the players use independent sources of randomness and do not observe the outcome of the opponent's random choice before choosing their own action. If the row player uses mixed strategy $\mathbf{y} \in S_m$ and the column player uses $\mathbf{x} \in S_n$, the probability of action profile (i, j) is

$$\mathbb{P}[(i,j)] = \mathbb{P}[\text{row } i] \cdot \mathbb{P}[\text{column } j] = y_i x_j.$$

The column player's payoff at profile (i, j) is a_{ij}. The *expected payoff* under repeated use of the same strategies $\mathbf{x}, \mathbf{y}$ is the sum of the payoffs for all possible profiles, weighted by their probabilities:

$$\mathbb{E}[\text{column player's payoff}] = \sum_{i=1}^{m}\sum_{j=1}^{n} \underbrace{\mathbb{P}[(i,j)]}_{y_i x_j}\, a_{ij} = \sum_{i=1}^{m}\sum_{j=1}^{n} y_i\, a_{ij}\, x_j.$$

This double sum has a compact matrix representation. Treat $\mathbf{y}$ as a column vector of length m and $\mathbf{x}$ as a column vector of length n. Then

$$\mathbb{E}[\text{column player's payoff}] = \mathbf{y}^{\mathsf{T}} A \mathbf{x}.$$

Since the game is zero-sum, the row player's expected payoff is

$$\mathbb{E}[\text{row player's payoff}] = -\,\mathbf{y}^{\mathsf{T}} A \mathbf{x}.$$

Remark 10.8: Gains and losses

For the column player, $\mathbf{y}^{\mathsf{T}} A \mathbf{x}$ is an *expected gain*. For the row player, it is an *expected loss*. It is therefore natural that:

- the column player seeks to *maximize* $\mathbf{y}^{\mathsf{T}} A \mathbf{x}$,
- the row player seeks to *minimize* $\mathbf{y}^{\mathsf{T}} A \mathbf{x}$.

This conflict of objectives is the essence of a zero-sum matrix game. Both players use the same quantity as their criterion, but one wants it as large as possible and the other as small as possible.

3 Maximin, minimax, and the value of a game

For a fixed strategy of one player, the other chooses a response that optimizes the expected outcome, maximizing a gain or minimizing a loss. This leads to the notions of *maximin*, *minimax*, and the *value of a game*.

3.1 The worst case for a fixed strategy

Fix a mixed strategy $\mathbf{x} \in S_n$ of the column player. For every possible strategy $\mathbf{y} \in S_m$ of the row player, the column player's expected payoff is $\mathbf{y}^{\mathsf{T}} A \mathbf{x}$.

Consider the column player's perspective. To evaluate the *worst possible outcome* for strategy $\mathbf{x}$, assuming the row player acts as unfavorably as possible, we minimize this quantity over all $\mathbf{y} \in S_m$:

$$\min_{\mathbf{y} \in S_m} \mathbf{y}^{\mathsf{T}} A \mathbf{x}.$$

This motivates the following definition.

Definition 10.9: The functions β and α

For a strategy $\mathbf{x} \in S_n$ of the column player, define

$$\beta(\mathbf{x}) :- \min_{\mathbf{y} \in S_m} \mathbf{y}^{\mathsf{T}} A \mathbf{x}.$$

This is the worst possible (smallest) expected payoff of the column player when playing $\mathbf{x}$ against an optimal response by the row player.
Similarly, for a strategy $\mathbf{y} \in S_m$ of the row player, define

$$\alpha(\mathbf{y}) :- \max_{\mathbf{x} \in S_n} \mathbf{y}^{\mathsf{T}} A \mathbf{x}.$$

This is the worst possible (largest) expected loss of the row player when playing $\mathbf{y}$ against an optimal response by the column player.

Neither optimum can escape us here. The sets S_m and S_n are nonempty compact simplices and the expected payoff is continuous, so the minima and maxima in these definitions are always attained.

The definitions immediately give

$$\beta(\mathbf{x}) \le \mathbf{y}^{\mathsf{T}} A \mathbf{x} \le \alpha(\mathbf{y}) \quad \text{for all } \mathbf{x} \in S_n,\ \mathbf{y} \in S_m.$$

For a fixed pair $(\mathbf{x}, \mathbf{y})$, the value $\mathbf{y}^{\mathsf{T}} A \mathbf{x}$ cannot be below the minimum over all $\mathbf{y}$, or above the maximum over all $\mathbf{x}$.

3.2 Maximin and minimax strategies and the value of a game

Using $\beta(\mathbf{x})$ and $\alpha(\mathbf{y})$, we can now state precisely what it means for a mixed strategy to be optimal in the worst case.

Definition 10.10: Maximin strategies, minimax strategies, and game value

A mixed strategy $\mathbf{x}^\star \in S_n$ is a *maximin strategy of the column player* if it maximizes the player's worst possible payoff:

$$\beta(\mathbf{x}^\star) = \min_{\mathbf{y} \in S_m} \mathbf{y}^{\mathsf{T}} A \mathbf{x}^\star = \max_{\mathbf{x} \in S_n} \beta(\mathbf{x}).$$

Similarly, a mixed strategy $\mathbf{y}^\star \in S_m$ is a *minimax strategy of the row player* if it minimizes the player's worst possible loss:

$$\alpha(\mathbf{y}^\star) = \max_{\mathbf{x} \in S_n} (\mathbf{y}^\star)^{\mathsf{T}} A \mathbf{x} = \min_{\mathbf{y} \in S_m} \alpha(\mathbf{y}).$$

A number $v \in \mathbb{R}$ is the *value of the game* if there are strategies $\mathbf{x}^\star, \mathbf{y}^\star$ such that

$$\beta(\mathbf{x}^\star) = v = \alpha(\mathbf{y}^\star).$$

Intuitively:

- by choosing $\mathbf{x}^\star$, the column player guarantees an expected payoff of at least v, even against the opponent's worst possible response,
- by choosing $\mathbf{y}^\star$, the row player guarantees a loss of at most v, even against the opponent's worst possible choice.

The definitions imply that every such game satisfies

$$\max_{\mathbf{x}\in S_n} \beta(\mathbf{x}) \;\le\; \min_{\mathbf{y}\in S_m} \alpha(\mathbf{y}).$$

Zero-sum matrix games have a surprisingly strong property: *these two values are equal*. This is the content of the following minimax theorem, whose classical proof is due to John von Neumann (von Neumann, 1928). For a modern treatment, see also (Osborne and Rubinstein, 1994).

Theorem 10.11: The minimax theorem for zero-sum matrix games

Let $A \in \mathbb{R}^{m\times n}$ be the payoff matrix of a two-player zero-sum game. Then

$$\max_{\mathbf{x}\in S_n} \beta(\mathbf{x}) \;=\; \min_{\mathbf{y}\in S_m} \alpha(\mathbf{y}).$$

In other words, there is a number $v \in \mathbb{R}$ such that

$$\max_{\mathbf{x}\in S_n} \min_{\mathbf{y}\in S_m} \mathbf{y}^{\mathsf{T}} A\mathbf{x} \;=\; v \;=\; \min_{\mathbf{y}\in S_m} \max_{\mathbf{x}\in S_n} \mathbf{y}^{\mathsf{T}} A\mathbf{x}.$$

The number v is the *value of the game*. Moreover, there are mixed strategies $\mathbf{x}^\star \in S_n$ and $\mathbf{y}^\star \in S_m$ satisfying

$$\beta(\mathbf{x}^\star) = v = \alpha(\mathbf{y}^\star).$$

Thus $\mathbf{x}^\star$ is a maximin strategy of the column player and $\mathbf{y}^\star$ is a minimax strategy of the row player.

Thus v is the value of the game and $(\mathbf{x}^\star, \mathbf{y}^\star)$ gives optimal mixed strategies for the two players. Each player uses their strategy to guarantee the same value, one from below and the other from above.

Remark 10.12: What exactly is an equilibrium?

The meaning of optimal strategies can be expressed by the saddle-point inequalities

$$(\mathbf{y}^\star)^{\mathsf{T}} A\mathbf{x} \;\le\; (\mathbf{y}^\star)^{\mathsf{T}} A\mathbf{x}^\star \;\le\; \mathbf{y}^{\mathsf{T}} A\mathbf{x}^\star \qquad \text{for all } \mathbf{x} \in S_n,\ \mathbf{y} \in S_m. \tag{10.1}$$

The left inequality says that the column player cannot improve by changing strategy unilaterally against $\mathbf{y}^\star$. The right inequality says that the row player cannot reduce the loss by changing strategy unilaterally against $\mathbf{x}^\star$. In game-theoretic terms, $(\mathbf{x}^\star, \mathbf{y}^\star)$ is a Nash equilibrium. In optimization terms, it is a saddle point of the

expected payoff. If $\mathbf{x}^\star$ and $\mathbf{y}^\star$ are pure strategies, this saddle point appears directly in the payoff matrix.

This property of zero-sum games follows directly from linear programming duality, as we now show.

4 Formulating a matrix game as a linear program

We now show that the column player's problem of choosing a strategy $\mathbf{x}$ that maximizes the worst possible payoff can be written as a linear program. Similarly, the row player's problem leads to the dual linear program.

4.1 The column player's LP (maximin)

Recall that the column player wants to maximize $\beta(\mathbf{x})$:

$$\max_{\mathbf{x}\in S_n} \beta(\mathbf{x}) = \max_{\mathbf{x}\in S_n} \min_{\mathbf{y}\in S_m} \mathbf{y}^\mathsf{T} A\mathbf{x}.$$

For a fixed strategy $\mathbf{x}$, the expression $\mathbf{y}^\mathsf{T} A\mathbf{x}$ is linear in $\mathbf{y}$. Its minimum over the simplex S_m is therefore attained at a vertex, that is, at a pure strategy of the row player. Let $\mathbf{e}_i$ denote the ith standard basis vector in $\mathbb{R}^m$. Recall that these vectors correspond exactly to pure strategies. We obtain

$$\min_{\mathbf{y}\in S_m} \mathbf{y}^\mathsf{T} A\mathbf{x} = \min_{i=1,\dots,m} \mathbf{e}_i^\mathsf{T} A\mathbf{x}.$$

The column player therefore solves the maximin problem

$$\max_{\mathbf{x}\in S_n} \min_{i=1,\dots,m} \mathbf{e}_i^\mathsf{T} A\mathbf{x}.$$

Introduce a variable $t \in \mathbb{R}$ that serves as a lower bound on all values $\mathbf{e}_i^\mathsf{T} A\mathbf{x}$. The constraints

$$t \le \mathbf{e}_i^\mathsf{T} A\mathbf{x} \qquad \text{for } i = 1, \dots, m$$

guarantee that $t \le \min_i \mathbf{e}_i^\mathsf{T} A\mathbf{x}$. Maximizing t therefore does maximize the column player's worst possible payoff.

For $\mathbf{x}$ to be a feasible mixed strategy, it must satisfy

$$x_j \ge 0 \quad \text{for all } j, \qquad \sum_{j=1}^{n} x_j = 1.$$

This gives the column player's linear program:

The column player's LP

$$\begin{aligned} \text{maximize} \quad & t \\ \text{subject to} \quad & t \le (A\mathbf{x})_i = \mathbf{e}_i^{\mathsf{T}} A\mathbf{x}, \quad i = 1, \dots, m, \\ & \textstyle\sum_{j=1}^{n} x_j = 1, \\ & x_j \ge 0 \quad \text{for all } j. \end{aligned} \tag{LP_C}$$

Each constraint $t \le (A\mathbf{x})_i$ says that if the row player chooses pure strategy i, the column player's expected payoff is at least t. Maximizing t makes the payoff as large as possible even against the row player's worst response.

4.2 The row player's LP (minimax)

The row player's problem can be formulated symmetrically:

$$\min_{\mathbf{y} \in S_m} \max_{\mathbf{x} \in S_n} \mathbf{y}^{\mathsf{T}} A\mathbf{x}.$$

As before, for fixed $\mathbf{y}$,

$$\max_{\mathbf{x} \in S_n} \mathbf{y}^{\mathsf{T}} A\mathbf{x} = \max_{j=1,\dots,n} (A^{\mathsf{T}}\mathbf{y})_j,$$

because a linear function on a simplex attains an optimum at a vertex.

Introduce an auxiliary variable $u \in \mathbb{R}$ as an upper bound on the values $(A^{\mathsf{T}}\mathbf{y})_j$. The constraints

$$(A^{\mathsf{T}}\mathbf{y})_j \le u \qquad \text{for } j = 1, \dots, n$$

ensure that $u \ge \max_j (A^{\mathsf{T}}\mathbf{y})_j$. Minimizing u therefore minimizes the largest possible payoff to the column player, which is the row player's worst possible loss.

The row player's mixed strategy must again satisfy

$$y_i \ge 0 \quad \text{for all } i, \qquad \sum_{i=1}^{m} y_i = 1.$$

The row player's linear program is therefore:

The row player's LP

$$\begin{aligned} \text{minimize} \quad & u \\ \text{subject to} \quad & (A^{\mathsf{T}}\mathbf{y})_j \le u, \quad j = 1, \dots, n, \\ & \textstyle\sum_{i=1}^{m} y_i = 1, \\ & y_i \ge 0 \quad \text{for all } i. \end{aligned} \tag{LP_R}$$

4.3 Duality: one problem for each player

The primal problem in matrix form

We take the column player's LP (LP_C) as the primal problem. We first write it more compactly and then derive its dual.

Introduce the vectors

$$\mathbf{z} :- \begin{pmatrix} \mathbf{x} \\ t \end{pmatrix} \in \mathbb{R}^{n+1}, \qquad \mathbf{c} :- \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \in \mathbb{R}^{n+1},$$

so that the primal objective can be written as $\max \mathbf{c}^\mathsf{T}\mathbf{z}$.

Writing the ith row of A as $\mathbf{a}_i^\mathsf{T}$, the inequalities $t \le \mathbf{e}_i^\mathsf{T} A\mathbf{x} = \mathbf{a}_i^\mathsf{T}\mathbf{x}$ become

$$(-\mathbf{a}_i^\mathsf{T} \;\; 1)\,\mathbf{z} \le 0, \quad i = 1, \ldots, m.$$

Define

$$B :- \begin{pmatrix} -\mathbf{a}_1^\mathsf{T} & 1 \\ \vdots & \vdots \\ -\mathbf{a}_m^\mathsf{T} & 1 \end{pmatrix}, \qquad \mathbf{d} :- \begin{pmatrix} 1 \\ \vdots \\ 1 \\ 0 \end{pmatrix} \in \mathbb{R}^{n+1},$$

and rewrite the problem as

$$\begin{aligned} &\text{maximize} && \mathbf{c}^\mathsf{T}\mathbf{z} \\ &\text{subject to} && B\,\mathbf{z} \le \mathbf{0}, \\ &&& \mathbf{d}^\mathsf{T}\mathbf{z} = 1, \\ &&& z_1, \ldots, z_n \ge 0,\ z_{n+1} = t \text{ free}. \end{aligned} \tag{P}$$

The dual problem in matrix form

We briefly recall the duality rules from the preceding chapters that we need here:

- a primal $\le$ constraint $\longleftrightarrow$ a dual variable ≥ 0,
- a primal equality constraint $\longleftrightarrow$ an unrestricted dual variable,
- a primal variable ≥ 0 $\longleftrightarrow$ a dual $\ge$ constraint,
- a free primal variable $\longleftrightarrow$ a dual equality constraint,
- a primal maximization objective $\longleftrightarrow$ a dual minimization objective.

Applying these rules to (P) gives:

- a vector of dual variables $\mathbf{y} \in \mathbb{R}^m$, with $y_i \ge 0$, corresponding to the inequalities $B\mathbf{z} \le \mathbf{0}$,
- one unrestricted dual variable $u \in \mathbb{R}$ corresponding to $\mathbf{d}^\mathsf{T}\mathbf{z} = 1$,
- a dual objective function to be minimized.

To write the dual as symmetrically as possible with the primal matrix formulation, introduce the vectors

$$\mathbf{w} :- \begin{pmatrix} \mathbf{y} \\ u \end{pmatrix} \in \mathbb{R}^{m+1}, \qquad \mathbf{r} :- \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \in \mathbb{R}^{m+1}.$$

The dual objective is then the inner product

$$\min \mathbf{r}^{\mathsf{T}}\mathbf{w}.$$

The coefficients associated with the primal variables in the dual are linear combinations of the rows of B and the vector $\mathbf{d}$, with weights $\mathbf{y}$ and u. They form the vector

$$B^{\mathsf{T}}\mathbf{y} + u\,\mathbf{d} \in \mathbb{R}^{n+1},$$

which must be compared with $\mathbf{c}$ according to the types of primal variables:

- since the primal variables $z_1, \dots, z_n$ are nonnegative, the first n components must satisfy $\geq$ inequalities,
- the last variable $z_{n+1} = t$ is free, so its dual constraint is an equality.

We obtain the compact matrix form of the dual problem:

$$\begin{aligned} &\text{minimize} && \mathbf{r}^{\mathsf{T}}\mathbf{w} \\ &\text{subject to} && B^{\mathsf{T}}\mathbf{y} + u\,\mathbf{d} \;\geq\; \mathbf{c} \quad \text{(in the first } n \text{ components)}, \\ &&& (B^{\mathsf{T}}\mathbf{y} + u\,\mathbf{d})_{n+1} = c_{n+1}, \\ &&& \mathbf{y} \geq \mathbf{0}. \end{aligned} \tag{D}$$

In this case, $B^{\mathsf{T}}\mathbf{y} + u\,\mathbf{d}$ has the explicit form

$$B^{\mathsf{T}}\mathbf{y} + u\,\mathbf{d} = \begin{pmatrix} -A^{\mathsf{T}}\mathbf{y} + u\,\mathbf{1} \\ \sum_{i=1}^{m} y_i \end{pmatrix}.$$

The dual constraints therefore read

$$-A^{\mathsf{T}}\mathbf{y} + u\,\mathbf{1} \;\geq\; \mathbf{0}, \qquad \sum_{i=1}^{m} y_i = 1, \qquad \mathbf{y} \geq \mathbf{0}.$$

This is the exact matrix form of the dual problem. Writing it componentwise gives the row player's linear program derived earlier from the minimax formulation.

We now reach the main point of the chapter. Both problems are feasible: for example, use a uniform mixed strategy and choose t sufficiently small or u sufficiently large. Their optimal values are finite because every expected payoff lies between the smallest and largest entries of A. Strong duality therefore gives $t^\star = u^\star$. The primal optimum yields a maximin strategy of the column player, the dual optimum yields a minimax strategy of the row player, and their common optimal value is v. Linear programming duality thus proves the minimax theorem for finite matrix games.

5 A short example

We illustrate the two linear programs with a simple two-strategy game with a symmetric payoff matrix.

Example 10.13: A game with two opposing strategies

Consider a matrix game with two rows, two columns, and payoff matrix

$$A = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

Here a_{ij} is the column player's payoff. A positive entry represents a gain for the column player, and a negative entry a gain for the row player.

The column player's LP. Let $\mathbf{x} = (x_1, x_2)^\mathsf{T}$ be the column player's mixed strategy. By (LP_C), the player solves

$$\begin{aligned} \text{maximize} \quad & v \\ \text{subject to} \quad & v \le (1,-1)\,\mathbf{x} = x_1 - x_2, \\ & v \le (-1,1)\,\mathbf{x} = -x_1 + x_2, \\ & x_1 + x_2 = 1, \\ & x_1, x_2 \ge 0. \end{aligned}$$

The row player's LP. Let $\mathbf{y} = (y_1, y_2)^\mathsf{T}$ be the row player's strategy. From (LP_R), we obtain

$$\begin{aligned} \text{minimize} \quad & u \\ \text{subject to} \quad & \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}^\mathsf{T} \mathbf{y} \le u \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \\ & y_1 + y_2 = 1, \\ & y_1, y_2 \ge 0. \end{aligned}$$

Expanding the first constraint gives

$$\begin{aligned} \text{minimize} \quad & u \\ \text{subject to} \quad & y_1 - y_2 \le u, \\ & -y_1 + y_2 \le u, \\ & y_1 + y_2 = 1, \\ & y_1, y_2 \ge 0. \end{aligned}$$

In this example, the optimal strategies must balance the two opposing linear payoffs. Both players therefore choose their pure strategies with equal probabilities:

$$\mathbf{x}^\star = \mathbf{y}^\star = \left(\tfrac{1}{2}, \tfrac{1}{2}\right)^\mathsf{T}.$$

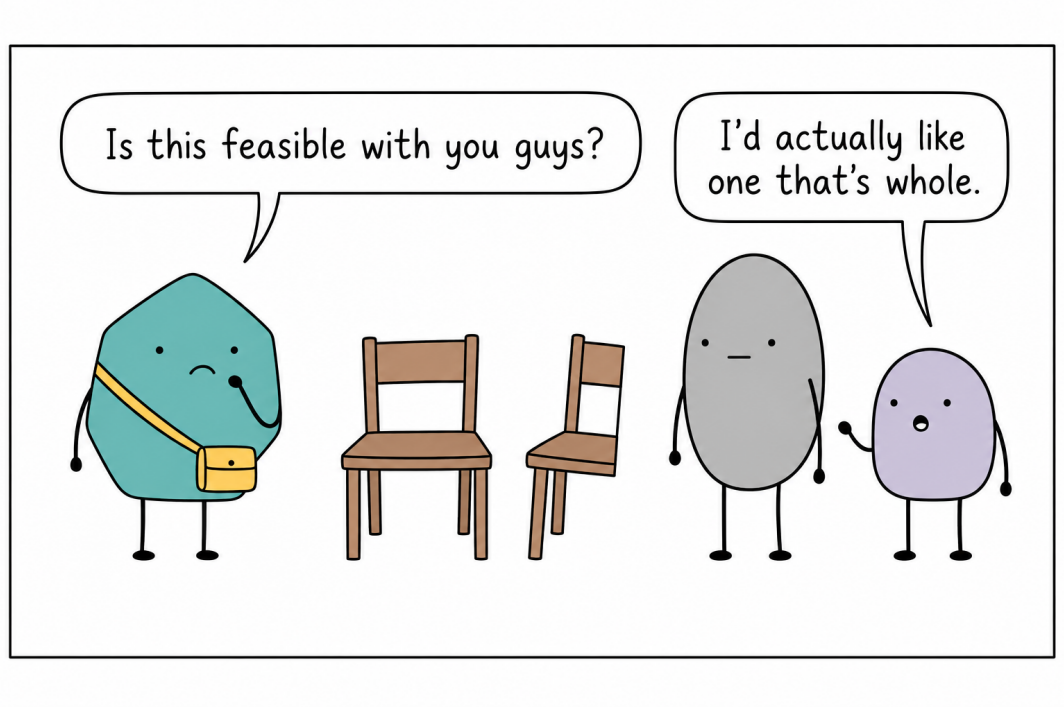
The value of the game is $v = 0$, so at equilibrium neither player has an expected advantage.

There is a tempting shortcut to avoid, however. Symmetry of the matrix alone does not generally imply a uniform optimal strategy. For example, the symmetric payoff matrix $\left(\begin{smallmatrix} 2 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ has optimal strategies $(1/3, 2/3)^\mathsf{T}$, not $(1/2, 1/2)^\mathsf{T}$. What matters is solving the maximin and minimax conditions, not merely observing symmetry in the matrix.

CHAPTER 11

INTEGER PROGRAMMING: SELECTED SOLUTION METHODS

In this final chapter, we return to integer and mixed-integer linear programming (ILP/MILP), which we briefly introduced earlier. We first review what it means to impose integrality constraints and the role of the LP relaxation. We then discuss two methods for solving integer programming problems: branch and bound and Gomory's cutting-plane method. Both build on the LP relaxation, but in different ways.



1 A brief review of ILP and MILP

In the introductory chapter, we defined integer linear programs (ILPs) and mixed-integer linear programs (MILPs) as linear programs with additional integrality constraints on selected variables. Here we recall a few key observations needed in this chapter:

- In a pure ILP, every decision variable must be integer, so $\mathbf{x} \in \mathbb{Z}^n$, often with $\mathbf{x} \geq \mathbf{0}$ as well. In an MILP, integrality is required only for indices in a nonempty set $I \subsetneq \{1, \ldots, n\}$. The remaining variables are continuous.
- The LP relaxation is obtained by dropping the conditions $x_i \in \mathbb{Z}$ for $i \in I$. All linear constraints, including variable bounds, are retained. If the corresponding finite optimal values exist, the optimum of the LP relaxation is an upper bound on the ILP/MILP optimum for a maximization problem and a lower bound for a minimization problem.
- The feasible set of a pure ILP consists of the integer points of a polyhedron and is generally discrete and nonconvex. The feasible set of an MILP may contain continuous pieces, because some variables are real-valued. It too is generally nonconvex. In both cases, the feasible set of the LP relaxation is a convex polyhedron.

Geometrically, we select the integer points in the polyhedron P_{LP}. This polyhedron is the feasible set of the LP relaxation, which we can solve using the techniques developed in earlier chapters.

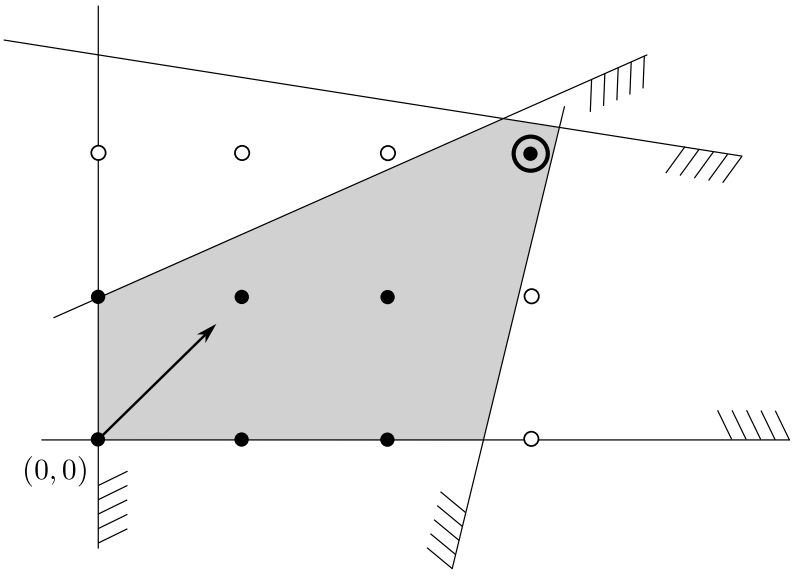


Figure 11.1: The feasible set of the LP relaxation (the gray convex polygon) and the feasible points of the pure ILP (the black integer points).

The basic strategy remains the same: first solve the relatively inexpensive LP relaxation to obtain a bound on the optimum. Then cut away unwanted fractional parts or branch into subproblems until we find an optimal solution satisfying the required integrality conditions, or prove that no such solution exists.

2 Two basic modeling examples

Integer variables allow us to describe yes-or-no decisions and indivisible numbers of objects. The following two examples also show that the strength of an LP relaxation depends on the structure of the particular model.

Example 11.1: The binary knapsack problem

There are n items. Item j has profit $p_j > 0$ and weight $w_j > 0$. The knapsack has capacity $W > 0$. A binary variable x_j indicates whether item j is selected. The model is

$$\begin{aligned} \textbf{maximize} \quad & \sum_{j=1}^{n} p_j x_j, \\ \textbf{subject to} \quad & \sum_{j=1}^{n} w_j x_j \le W, \\ & x_j \in \{0,1\}, \qquad j = 1, \dots, n. \end{aligned}$$

The LP relaxation replaces $x_j \in \{0,1\}$ by $0 \le x_j \le 1$. It thus allows fractional parts of items to be selected, and its optimal value may be strictly larger than that of the original problem.

Example 11.2: Assigning workers to tasks

There are m workers, m tasks, and a cost c_{ij} for assigning worker i to task j. The variable x_{ij} equals one if and only if worker i performs task j. The model is

$$\begin{aligned} \textbf{minimize} \quad & \sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij} x_{ij}, \\ \textbf{subject to} \quad & \sum_{j=1}^{m} x_{ij} = 1, \qquad i = 1, \dots, m, \\ & \sum_{i=1}^{m} x_{ij} = 1, \qquad j = 1, \dots, m, \\ & x_{ij} \in \{0,1\}. \end{aligned}$$

Unlike a general ILP, every vertex of this model's LP relaxation is a permutation matrix. Thus, if the problem is feasible, an integer optimum already exists among the optimal solutions of the LP relaxation. This result, known as the Birkhoff–von Neumann theorem, is a special property of the assignment polytope, not a general rule of integer programming (Schrijver, 1986).

3 LP relaxations, branching, and cuts: a general framework

Most methods for general ILP/MILP problems are based on two steps:

1. **Relaxation:** temporarily drop the integrality constraints and solve the LP relaxation to obtain a bound on the optimum.

2. **Refinement:** if the relaxation's optimal solution does not satisfy the required integrality conditions, tighten the problem either *locally* by branching or *globally* by adding a valid cut.

A general framework for ILP/MILP methods

Consider the maximization problem

$$\max\{\mathbf{c}^{\mathsf{T}}\mathbf{x} \mid \mathbf{x} \in X_I\}, \qquad X_I := \{\mathbf{x} \in P \mid x_i \in \mathbb{Z} \text{ for all } i \in I\},$$

where $P \subseteq \mathbb{R}^n$ is a polyhedron and I indexes the integer variables. For a pure ILP, $I = \{1, \ldots, n\}$. A typical method maintains:

- a *relaxation polyhedron* $P^{(k)}$ containing all solutions in X_I that satisfy the branching constraints of the current subproblem,
- the best solution found so far that satisfies the required integrality conditions, $\mathbf{x}^{\text{best}}$, or an indication that none has yet been found.

At iteration k, the method:

1. solves an LP over $P^{(k)}$, obtaining an optimum $\mathbf{x}^{(k)}$ and an upper bound $z_{\text{LP}}^{(k)}$,
2. depending on the method, either:
   - divides the problem into subproblems by adding constraints on integer variables (branching), or
   - adds new inequalities (cuts) to $P^{(k)}$ that exclude $\mathbf{x}^{(k)}$ but preserve all solutions in X_I.

If a branch's LP upper bound is no greater than the value of the best known solution, the branch can safely be closed. Once no open branches remain, the best solution found is globally optimal. If no solution has been found at that point, the original problem is infeasible.

The next two sections show both approaches in action. Branch and bound divides the problem into subproblems arranged in a tree. Gomory cuts instead maintain a single global relaxation and gradually cut away its unwanted parts.

4 Branch and bound

4.1 The basic idea

At first sight, we might try every combination of integer values. Such a brute-force approach quickly becomes overwhelming: a binary vector of length n has 2^n possibilities. *Branch and bound* replaces exhaustive enumeration with a more selective tree search.

Upper bounds from LP relaxations allow many subtrees to be skipped entirely. The classical general formulation of the method is due to Land and Doig (1960).

Each node of the tree represents an ILP/MILP problem with additional constraints, typically $x_j \le k$ or $x_j \ge k+1$ for $j \in I$. After solving the node's LP relaxation, one of the following situations occurs. We discuss unbounded relaxations separately below.

a) The LP relaxation is infeasible $\Rightarrow$ neither the node nor its subtree can contain a solution satisfying the required integrality conditions.

b) The LP relaxation's optimal value is no better than the value of the best known solution satisfying integrality $\Rightarrow$ close the node.

c) The LP optimum satisfies $x_i \in \mathbb{Z}$ for every $i \in I$ $\Rightarrow$ it is also optimal for the ILP/MILP subproblem. Update the best known solution if appropriate.

d) Some x_j, $j \in I$, is fractional at the LP optimum $\Rightarrow$ create two child nodes using branching constraints.

4.2 An algorithmic outline

For simplicity, consider a maximization problem. Let z_{best} be the value of the best solution found so far satisfying the required integrality conditions, initially $-\infty$.

Branch and bound – a basic outline

1. Create the *root node* containing the original ILP/MILP problem, with no additional branching constraints.
2. While unresolved nodes remain:
   i) Select a node according to a chosen rule and solve its LP relaxation.
   ii) If the LP relaxation is infeasible, close the node.
   iii) If $z_{\text{LP}} \le z_{\text{best}}$, close the node, since it cannot contain a better solution.
   iv) If the LP optimum satisfies $x_i \in \mathbb{Z}$ for every $i \in I$, update the best solution if appropriate and close the node.
   v) Otherwise, choose a fractional variable x_j with $j \in I$ and create two children by adding the constraints
   $$x_j \le \lfloor \hat{x}_j \rfloor, \qquad x_j \ge \lceil \hat{x}_j \rceil,$$
   where $\hat{x}_j$ is the value of x_j at the LP optimum. Add the new nodes to the list awaiting processing.
3. When no nodes remain, $\mathbf{x}^{\text{best}}$ is globally optimal. If no solution satisfying integrality was found, the problem is infeasible.

Remark 11.3: Selecting nodes and branching variables

Many heuristics are used in practice to choose a node, including depth-first, breadth-first, and best-bound search, and to choose a branching variable. These rules can dramatically affect the size of the search tree without changing the method's correctness: each node is either visited or its subtree is correctly pruned using LP bounds.

Remark 11.4: Finite termination and unbounded relaxations

The outline above assumes that the LP relaxations being solved have finite optima. An unbounded relaxation gives no finite bound and requires separate analysis of the integer subproblem's feasibility and the unbounded direction. The tree is immediately guaranteed to be finite, for example, when every integer variable has finite lower and upper bounds. Without such assumptions, the outline is a valid general principle, but not by itself a proof of finite termination for every possible implementation.

4.3 A simple example

Example 11.5: Branch and bound in two variables

Consider the problem

$$\begin{aligned} \textbf{max} \quad & 3x_1 + 2x_2 \\ \textbf{subject to} \quad & 4x_1 + 2x_2 \le 9, \\ & 2x_1 + 4x_2 \le 9, \\ & x_1, x_2 \in \mathbb{Z}_{\ge 0}. \end{aligned}$$

Its LP relaxation, with real variables $x_1, x_2 \ge 0$, attains its optimum at the intersection

$$4x_1 + 2x_2 = 9, \qquad 2x_1 + 4x_2 = 9,$$

namely, at

$$(\hat{x}_1, \hat{x}_2) = \left(\tfrac{3}{2}, \tfrac{3}{2}\right), \qquad z_{\mathrm{LP}} = 3 \cdot \tfrac{3}{2} + 2 \cdot \tfrac{3}{2} = \tfrac{15}{2} = 7.5.$$

This point is fractional, so we branch, for example, on x_1:

$$x_1 \le 1 \quad \text{and} \quad x_1 \ge 2.$$

Using best-bound node selection, for example, gives the following sequence:

- At the node $x_1 \ge 2$, the LP optimum is $(2, \frac{1}{2})$ with bound $z_{\mathrm{LP}} = 7$. Branch on x_2 into $x_2 \le 0$ and $x_2 \ge 1$. The node $x_2 \ge 1$ is infeasible, since $4x_1 + 2x_2 \le 9$ and $x_1 \ge 2$ imply $x_2 \le \frac{1}{2}$. At $x_2 \le 0$, the LP optimum is $(\frac{9}{4}, 0)$ with bound $z_{\mathrm{LP}} = \frac{27}{4} = 6.75$, so we branch on x_1. The branch $x_1 \le 2$ yields the integer solution $(2, 0)$ with value $z_{\mathrm{best}} = 6$. Its sibling $x_1 \ge 3$ is infeasible, since $x_2 \ge 0$ would imply $4x_1 + 2x_2 \ge 12 > 9$.

- At the node $x_1 \le 1$, the LP optimum is $(1, \frac{7}{4})$ with bound $z_{\mathrm{LP}} = \frac{13}{2} = 6.5$. This node cannot yet be pruned because $6.5 > 6$. Branch on x_2 into $x_2 \le 1$ and $x_2 \ge 2$. The node $x_2 \le 1$ has integer optimum $(1, 1)$ with value 5, which does not improve the best known solution. The node $x_2 \ge 2$ has upper bound $z_{\mathrm{LP}} = \frac{11}{2} = 5.5 \le 6$, so it is pruned by bound.

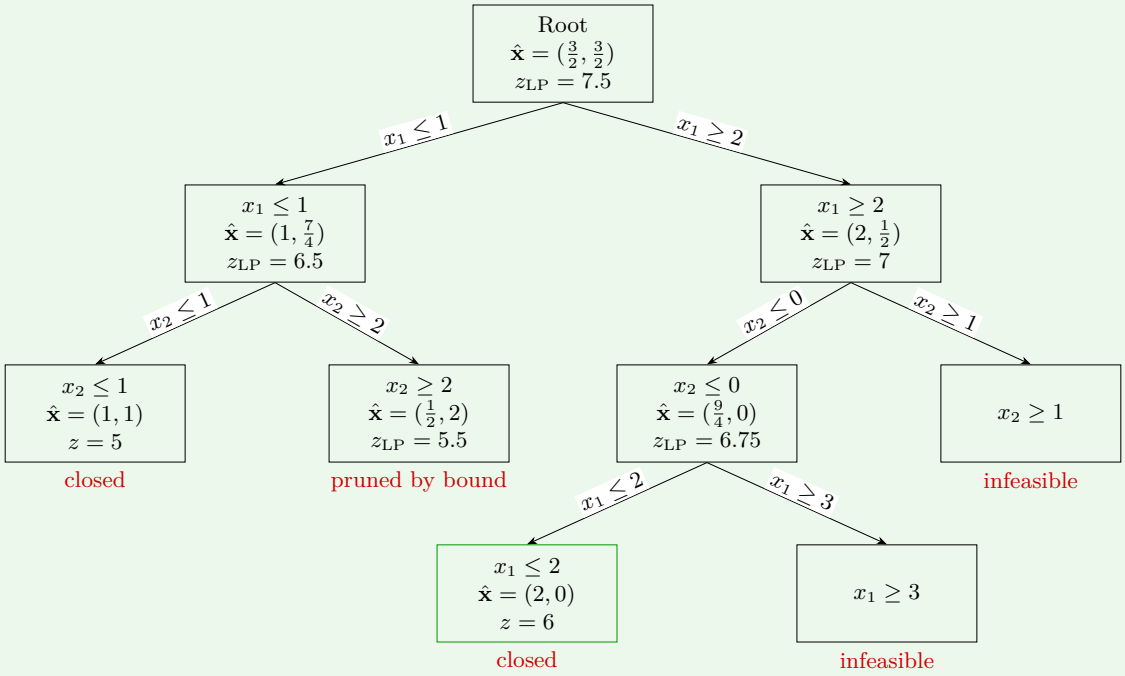


Figure 11.2: The branch-and-bound tree for Example 11.5. Each solved node shows its LP optimum and upper bound z_{LP}. Leaves are closed because of an integer solution, infeasibility, or a comparison of their bound with the best known solution.

The resulting global optimum is

$$z^\star = 6 \quad \text{at} \quad (x_1, x_2) = (2, 0).$$

5 Gomory cuts

5.1 The basic idea

Cutting-plane methods work with a single global LP relaxation, which is tightened by adding constraints. Instead of a branching tree, we follow one polyhedron, successively removing parts that contain no integer solutions. Starting with the relaxation of the original integer problem, at each step we seek an inequality that

- is *valid* for all integer solutions, so it excludes no point of $P \cap \mathbb{Z}^n$,
- is *violated* by the optimal solution of the current LP relaxation.

Such an inequality is called a *cut*: it cuts off the current optimum while preserving all integer feasible points. After adding a cut, we solve the LP relaxation again, for

example with the simplex method, and repeat until an integer optimum is obtained. In practice, adding cuts can lead to numerical difficulties such as degeneracy, ill-conditioning, and an accumulation of constraints. Modern solvers therefore usually combine cuts with branching.

There are many kinds of cuts, including Chvátal–Gomory cuts, cover cuts, and subtour elimination cuts for the traveling salesperson problem. Here we restrict attention to classical *Gomory fractional cuts* for pure ILPs (Gomory, 1958). MILPs use generalizations, notably Gomory mixed-integer (GMI) cuts. Their derivation is beyond the scope of this text.

5.2 The Gomory cut

Consider an integer program with rational data in equality form:

$$\begin{aligned} \textbf{max} \quad & \mathbf{c}^{\mathsf{T}}\mathbf{x} \\ \textbf{subject to} \quad & A\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0},\ \mathbf{x} \in \mathbb{Z}^n. \end{aligned}$$

Here $A \in \mathbb{Q}^{m\times n}$ has rank m, and $\mathbf{b} \in \mathbb{Q}^m$, $\mathbf{c} \in \mathbb{Q}^n$. In this formulation, all variables are integer, not just the original decision variables. If equality form was obtained from inequalities with integer coefficients and right-hand sides by adding slack or surplus variables, these variables are integer at every integer solution as well. If any variable in the row were continuous, the fractional cut derived below would generally not be valid.

Remark 11.6: Rational data

Gomory cuts are constructed by separating numbers into their integer and fractional parts. We formally assume rational problem data, which ensure that all coefficients in the simplex tableau are rational.

After solving the LP relaxation, we have an optimal simplex tableau in the form

$$\mathbf{x}_B = \mathbf{p} + Q\mathbf{x}_N, \qquad z = z_0 + \mathbf{r}^{\mathsf{T}}\mathbf{x}_N,$$

or, componentwise, for $B = \{j_1 < \cdots < j_m\}$ and $N = \{\ell_1 < \cdots < \ell_{n-m}\}$,

$$x_{j_i} = p_i + \sum_{k=1}^{n-m} q_{ik}\, x_{\ell_k}.$$

The associated basic feasible solution is obtained by setting $\mathbf{x}_N = \mathbf{0}$, giving $\mathbf{x}_B = \mathbf{p}$.

Suppose $p_i \notin \mathbb{Z}$ for some row i. Write the fractional part of a number t as $\{t\} = t - \lfloor t \rfloor \in [0,1)$. Set

$$f_i = \{p_i\} \in (0,1), \qquad f_{ik} = \{-q_{ik}\} \in [0,1).$$

Then the inequality

$$\sum_{k=1}^{n-m} f_{ik}\, x_{\ell_k} \;\geq\; f_i$$

is valid for every integer solution and violated by the current fractional basic optimum. At that optimum, $\mathbf{x}_N = \mathbf{0}$, so the left-hand side is $0 < f_i$. This inequality defines the Gomory cut, as stated below.

Definition 11.7: A Gomory fractional cut

Suppose an optimal simplex tableau of the LP relaxation of a pure ILP has basis $B = \{j_1, \ldots, j_m\}$ and nonbasic index set $N = \{\ell_1, \ldots, \ell_{n-m}\}$, and its ith row is

$$x_{j_i} = p_i + \sum_{k=1}^{n-m} q_{ik}\, x_{\ell_k}, \qquad p_i \notin \mathbb{Z}.$$

All variables in this row must be integer in the equality-form model, including any introduced slack or surplus variables. Write $f_i = \{p_i\}$ and $f_{ik} = \{-q_{ik}\}$. The inequality

$$\sum_{k=1}^{n-m} f_{ik}\, x_{\ell_k} \;\geq\; f_i$$

is called a *Gomory cut*.

Remark 11.8: Why the Gomory cut is valid and violated

Consider the chosen row of the simplex tableau,

$$x_{j_i} = p_i + \sum_{k=1}^{n-m} q_{ik}\, x_{\ell_k}, \qquad p_i \notin \mathbb{Z},$$

and assume a pure ILP, so that at any integer solution, $x_{j_i} \in \mathbb{Z}$ and all nonbasic variables satisfy $x_{\ell_k} \in \mathbb{Z}$. Recall that $\mathbf{x} \geq \mathbf{0}$ is still required.
Introduce auxiliary coefficients $a_{ik} = -q_{ik}$ to rewrite the row as

$$x_{j_i} = p_i - \sum_{k=1}^{n-m} a_{ik}\, x_{\ell_k}.$$

Now separate the coefficients on the right into their integer and fractional parts:

$$p_i = \lfloor p_i \rfloor + f_i, \qquad a_{ik} = \lfloor a_{ik} \rfloor + f_{ik},$$

where $f_i = \{p_i\} \in (0, 1)$ and $f_{ik} = \{a_{ik}\} = \{-q_{ik}\} \in [0, 1)$. Substitution gives

$$x_{j_i} = \underbrace{\Big(\lfloor p_i \rfloor - \sum_{k=1}^{n-m} \lfloor a_{ik} \rfloor\, x_{\ell_k}\Big)}_{\in \mathbb{Z}} + \Big(f_i - \sum_{k=1}^{n-m} f_{ik}\, x_{\ell_k}\Big).$$

At an integer solution, the left-hand side x_{j_i} and the first parenthesized expression are integers. The second parenthesized expression must therefore also be an integer. At the same time,

$$f_i - \sum_{k=1}^{n-m} f_{ik}\, x_{\ell_k} \leq f_i < 1,$$

so this expression cannot be a positive integer. It must be ≤ 0, which gives

$$\sum_{k=1}^{n-m} f_{ik}\, x_{\ell_k} \geq f_i.$$

Thus the inequality, the Gomory cut, is *valid* for all integer solutions.
On the other hand, the current basic solution has $\mathbf{x}_N = \mathbf{0}$, so $x_{\ell_k} = 0$ for all k, and the left-hand side of the cut is 0. Since $f_i \in (0, 1)$, we have $0 < f_i$, and the cut is *violated* at this basic optimum.

Remark 11.9: Cuts and branches

Gomory cuts and branch and bound are complementary approaches:

- branch and bound divides the solution space into many subproblems (branches), which are pruned using LP bounds,
- cutting-plane methods maintain one global polyhedron and refine it by adding valid inequalities.

Modern solvers usually combine both ideas: cuts first strengthen the root relaxation substantially, and branch and bound then works with the strengthened relaxation. This combination is called *branch and cut*.

5.3 An algorithmic outline

Again, for simplicity, consider a pure maximization ILP in equality form, and assume that every LP relaxation solved is either infeasible or has a finite optimum. The cutting-plane method maintains a sequence of relaxations $P^{(k)}$ such that

$$P^{(k)} \supseteq P \cap \mathbb{Z}^n.$$

Thus no cut ever excludes an integer solution.

Gomory's cutting-plane method – a basic outline

1. Set $k \leftarrow 0$ and $P^{(0)} \leftarrow P_{\text{LP}}$, the initial LP relaxation.
2. Repeat:
   i) Solve the LP over $P^{(k)}$ by the simplex method. If it is infeasible, there is no integer solution either, because every added cut is valid. Stop.
   ii) Otherwise, take an optimal simplex tableau in the form
   $$\mathbf{x}_B = \mathbf{p} + Q\mathbf{x}_N, \qquad z = z_0 + \mathbf{r}^{\mathsf{T}}\mathbf{x}_N.$$
   The associated basic feasible solution has $\mathbf{x}_N = \mathbf{0}$ and hence $\mathbf{x}_B = \mathbf{p}$. If $\mathbf{p} \in \mathbb{Z}^m$, we have an integer optimum. Stop.
   iii) Otherwise, choose a row i with fractional p_i and construct its Gomory cut (Definition 11.7).
   iv) Multiply the cut by a common denominator so that its coefficients and right-hand side are integers, and add it to the relaxation. Any new surplus variable is integer. Set $P^{(k+1)} \leftarrow P^{(k)} \cap \{\text{new cut}\}$ and $k \leftarrow k+1$, and continue.

Remark 11.10: Termination of the cutting-plane method

The procedure above captures the main idea, but the instruction to choose any fractional row does not by itself guarantee finite termination. Classical finite variants of Gomory's method assume a pure ILP with rational data, exact arithmetic, a prescribed choice of fractional row, and the lexicographic dual simplex method to restore feasibility. Under these conditions, the method finds an integer optimum or proves infeasibility after finitely many iterations (Gomory, 1958, Schrijver, 1986, Gade and Küçükyavuz, 2013). The technical details of the proof are beyond our scope. Floating-point arithmetic also introduces difficulties with rounding and nearly integer values. Practical solvers therefore usually combine cuts with branching. For a detailed modern treatment, see Conforti et al. (2014).

5.4 An example of Gomory cuts

Example 11.11: Two iterations of Gomory cuts

Consider the pure integer linear program

$$\begin{aligned} \textbf{max} \quad & z = x_1 + x_2, \\ \textbf{subject to} \quad & 2x_1 + x_2 \le 4, \\ & x_1 + 2x_2 \le 4, \\ & x_1, x_2 \in \mathbb{Z}_{\ge 0}. \end{aligned}$$

Introducing slack variables $s_1, s_2 \ge 0$ gives the equality form

$$2x_1 + x_2 + s_1 = 4, \qquad x_1 + 2x_2 + s_2 = 4.$$

Since the coefficients and right-hand sides are integers, every integer pair (x_1, x_2) also gives integer values of s_1 and s_2. We may therefore treat them as integer variables when deriving a Gomory fractional cut.

The LP relaxation. The LP relaxation attains its optimum at the intersection of the two constraint boundaries,

$$(\hat{x}_1, \hat{x}_2) = \left(\tfrac{4}{3}, \tfrac{4}{3}\right), \qquad z_{\mathrm{LP}} = \tfrac{8}{3},$$

which is fractional.
One optimal simplex tableau, with basis $B = \{x_1, x_2\}$ and nonbasic set $N = \{s_1, s_2\}$, is

$$\begin{aligned} x_1 &= \tfrac{4}{3} - \tfrac{2}{3}s_1 + \tfrac{1}{3}s_2, \\ x_2 &= \tfrac{4}{3} + \tfrac{1}{3}s_1 - \tfrac{2}{3}s_2, \\ z &= \tfrac{8}{3} - \tfrac{1}{3}s_1 - \tfrac{1}{3}s_2. \end{aligned}$$

Its basic feasible solution is obtained by setting $s_1 = s_2 = 0$.

The first Gomory cut. Choose the row for x_1, where

$$p_1 = \tfrac{4}{3}, \qquad f_1 = \{p_1\} = \tfrac{1}{3}.$$

The coefficients of the nonbasic variables are

$$q_{1,s_1} = -\tfrac{2}{3}, \qquad q_{1,s_2} = \tfrac{1}{3},$$

so

$$-q_{1,s_1} = \tfrac{2}{3}, \qquad -q_{1,s_2} = -\tfrac{1}{3},$$

and hence

$$f_{1,s_1} = \{-q_{1,s_1}\} = \tfrac{2}{3}, \qquad f_{1,s_2} = \{-q_{1,s_2}\} = -\tfrac{1}{3} - (-1) = \tfrac{2}{3}.$$

The Gomory cut is therefore

$$\tfrac{2}{3}s_1 + \tfrac{2}{3}s_2 \geq \tfrac{1}{3},$$

or, multiplying by three,

$$2s_1 + 2s_2 \geq 1.$$

Substituting $s_1 = 4 - 2x_1 - x_2$ and $s_2 = 4 - x_1 - 2x_2$ gives the equivalent constraint

$$2x_1 + 2x_2 \leq 5.$$

The second iteration. After adding the cut and solving the LP relaxation again, we obtain a new optimum

$$(x_1, x_2) = (1, \tfrac{3}{2}), \qquad z_{\mathrm{LP}} = \tfrac{5}{2}.$$

One corresponding optimal simplex tableau has basis $B = \{x_1, x_2, s_1\}$ and nonbasic set $N = \{s_2, s_3\}$, where the integer slack variable s_3 satisfies $2x_1 + 2x_2 + s_3 = 5$:

$$\begin{aligned} x_1 &= 1 + s_2 - s_3, \\ x_2 &= \tfrac{3}{2} - s_2 + \tfrac{1}{2}s_3, \\ s_1 &= \tfrac{1}{2} - s_2 + \tfrac{3}{2}s_3, \\ z &= \tfrac{5}{2} - \tfrac{1}{2}s_3. \end{aligned}$$

The row for x_2 gives

$$q_{2,s_2} = -1, \quad q_{2,s_3} = \tfrac{1}{2},$$

so

$$f_{2,s_2} = \{-q_{2,s_2}\} = 0, \qquad f_{2,s_3} = \{-q_{2,s_3}\} = \tfrac{1}{2}.$$

The Gomory cut is

$$\tfrac{1}{2}s_3 \geq \tfrac{1}{2}, \quad \text{that is} \quad s_3 \geq 1.$$

Since $s_3 = 5 - 2x_1 - 2x_2$, we obtain the new constraint

$$x_1 + x_2 \leq 2.$$

After adding this cut, the optimal set of the LP relaxation is the entire line segment

$$\{(x_1, x_2) \in \mathbb{R}^2_{\geq 0} \mid x_1 + x_2 = 2\}, \qquad z_{\mathrm{LP}} = 2.$$

The integer optimal solutions are $(2, 0)$, $(1, 1)$, and $(0, 2)$. The simplex method may return, for example, $(2, 0)$ or $(0, 2)$ as an optimal basic feasible solution. Such a solution is integer, so Gomory's method terminates with optimal value $z^\star = 2$. The presence of fractional optimal points in the interior of the segment causes no difficulty: finding one integer optimum is enough.

BIBLIOGRAPHY

Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1993. ISBN 978-0-13-617549-0.

Donald J. Albers and Constance Reid. An interview with George B. Dantzig: The father of linear programming. *The College Mathematics Journal*, 17(4):292–314, 1986. doi: 10.1080/07468342.1986.11972971.

Xavier Allamigeon, Pascal Benchimol, Stéphane Gaubert, and Michael Joswig. What tropical geometry tells us about the complexity of linear programming. *SIAM Review*, 63(1):123–164, 2021. doi: 10.1137/20M1380211.

Mokhtar S. Bazaraa, John J. Jarvis, and Hanif D. Sherali. *Linear Programming and Network Flows*. John Wiley & Sons, Hoboken, NJ, 4 edition, 2010. ISBN 978-0-470-46272-0.

Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Belmont, MA, 1997. ISBN 978-1-886529-19-9.

Robert G. Bland. New finite pivoting rules for the simplex method. *Mathematics of Operations Research*, 2(2):103–107, 1977. doi: 10.1287/moor.2.2.103.

Ivan Boldyrev and Till Düppe. Programming the USSR: Leonid V. Kantorovich in context. *The British Journal for the History of Science*, 53(2):255–278, 2020. doi: 10.1017/S0007087420000059.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004. URL https://web.stanford.edu/~boyd/cvxbook/.

Abraham Charnes and William W. Cooper. The stepping stone method of explaining linear programming calculations in transportation problems. *Management Science*, 1 (1):49–69, 1954. doi: 10.1287/mnsc.1.1.49.

Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. *Integer Programming*, volume 271 of *Graduate Texts in Mathematics*. Springer, Cham, 2014. ISBN 978-3-319-11007-3. doi: 10.1007/978-3-319-11008-0.

Richard W. Cottle, Ellis L. Johnson, and Roger J.-B. Wets. George B. Dantzig (1914–2005). *Notices of the American Mathematical Society*, 54(3):344–362, 2007. URL https://www.ams.org/notices/200703/fea-cottle.pdf.

George B. Dantzig. Application of the simplex method to a transportation problem. In Tjalling C. Koopmans, editor, *Activity Analysis of Production and Allocation*, number 13 in Cowles Commission Monograph, pages 359–373. John Wiley & Sons, New York, 1951. URL https://cowles.yale.edu/sites/default/files/2022-09/m13-all.pdf.

George B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ, 1963. doi: 10.1515/9781400884179. The DOI refers to a later digital edition.

George B. Dantzig. The diet problem. *Interfaces*, 20(4):43–47, 1990. doi: 10.1287/inte.20.4.43.

George B. Dantzig. Linear programming. *Operations Research*, 50(1):42–47, 2002. doi: 10.1287/opre.50.1.42.17798.

Yann Disser, Georg Loho, Matthew Maat, and Nils Mosis. Lower bounds for ranking-based pivot rules. In *43rd International Symposium on Theoretical Aspects of Computer Science (STACS 2026)*, volume 364 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026. doi: 10.4230/LIPIcs.STACS.2026.31. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.STACS.2026.31.

Julius Farkas. Theorie der einfachen ungleichungen. *Journal für die reine und angewandte Mathematik*, 124:1–27, 1902. doi: 10.1515/crll.1902.124.1.

Jana Friebelová. *Operační analýza*. Jihočeská univerzita v Českých Budějovicích, České Budějovice, 2009. ISBN 978-80-7394-193-2. URL https://omp.ef.jcu.cz/index.php/EF/catalog/view/44/43/104-1.

Dinakar Gade and Simge Küçükyavuz. Pure cutting-plane algorithms and their convergence. In James J. Cochran, Louis Anthony Cox, Pinar Keskinocak, Jeffrey P. Kharoufeh, and J. Cole Smith, editors, *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, 2013. doi: 10.1002/9780470400531.eorms1084.

Saul I. Gass. The first linear-programming shoppe. *Operations Research*, 50(1):61–68, 2002. doi: 10.1287/opre.50.1.61.17781.

Donald Goldfarb and John K. Reid. A practicable steepest-edge simplex algorithm. *Mathematical Programming*, 12:361–371, 1977. doi: 10.1007/BF01593804.

Ralph E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64(5):275–278, 1958. doi: 10.1090/S0002-9904-1958-10224-4.

Frank L. Hitchcock. The distribution of a product from several sources to numerous localities. *Journal of Mathematics and Physics*, 20:224–230, 1941. doi: 10.1002/sapm1941201224.

Josef Jablonský and Milada Lagová. *Lineární modely*. Oeconomica, Praha, 3 edition, 2014. ISBN 978-80-245-2020-9. URL https://oeconomica.vse.cz/publikace/linearni-modely/.

Leonid V. Kantorovich. Mathematical methods of organizing and planning production. *Management Science*, 6(4):366–422, 1960. doi: 10.1287/mnsc.6.4.366. English translation of the 1939 original.

Narendra Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984. doi: 10.1007/BF02579150.

Leonid G. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20(1):191–194, 1979. URL https://www.mathnet.ru/eng/dan42319. English translation. The link leads to the record of the Russian original.

Victor Klee and George J. Minty. How good is the simplex algorithm? In Oved Shisha, editor, *Inequalities III*, pages 159–175. Academic Press, New York, 1972.

Ailsa H. Land and Alison G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960. doi: 10.2307/1910129.

Carlton E. Lemke. The dual method of solving the linear programming problem. *Naval Research Logistics Quarterly*, 1(1):36–47, 1954. doi: 10.1002/nav.3800010107.

David G. Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*. Springer, Cham, 4 edition, 2016. doi: 10.1007/978-3-319-18842-3.

Jiří Matoušek. Lineární programování: Úvod pro informatiky. Technical Report 2006-311, Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University, 2006. URL https://iti.mff.cuni.cz/series/2006/311.pdf.

Jiří Matoušek and Bernd Gärtner. *Understanding and Using Linear Programming*. Universitext. Springer, Berlin and Heidelberg, 2007. ISBN 978-3-540-30697-9. doi: 10.1007/978-3-540-30717-4.

Alexander I. Nazarov and Galina I. Sinkevich. History of Leningrad mathematics in the first half of the 20th century. *Notices of the International Congress of Chinese Mathematicians*, 7(2):47–63, 2019. doi: 10.4310/ICCM.2019.v7.n2.a5.

George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, New York, 1988. doi: 10.1002/9781118627372.

Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, Cambridge, MA, 1994. ISBN 978-0-262-65040-3.

Christos H. Papadimitriou. On the complexity of integer programming. *Journal of the ACM*, 28(4):765–768, 1981. doi: 10.1145/322276.322287.

R. Tyrrell Rockafellar. *Convex Analysis*. Number 28 in Princeton Mathematical Series. Princeton University Press, Princeton, NJ, 1970. ISBN 978-0-691-08069-7. doi: 10.1515/9781400873173.

Jan Rohn. Lineární a nelineární programování. Lecture notes, Faculty of Mathematics and Physics, Charles University, 1997. URL https://kam.mff.cuni.cz/~hladik/doc/Rohn-LP_NLP-1997.pdf.

Royal Swedish Academy of Sciences. The prize in economic sciences 1975: Press release. NobelPrize.org, 1975. URL https://www.nobelprize.org/prizes/economic-sciences/1975/press-release/.

Francisco Santos. A counterexample to the Hirsch conjecture. *Annals of Mathematics*, 176(1):383–412, 2012. doi: 10.4007/annals.2012.176.1.7.

Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Chichester, 1986. ISBN 0-471-90854-1. URL https://ir.cwi.nl/pub/10206.

George J. Stigler. The cost of subsistence. *Journal of Farm Economics*, 27(2):303–314, 1945. doi: 10.2307/1231810.

Robert J. Vanderbei. *Linear Programming: Foundations and Extensions*, volume 285 of *International Series in Operations Research & Management Science*. Springer, Cham, 5 edition, 2020. ISBN 978-3-030-39415-8. doi: 10.1007/978-3-030-39415-8.

John von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100: 295–320, 1928. doi: 10.1007/BF01448847.

Tomáš Werner. Optimalizace. Lecture notes for course B0B33OPT, Faculty of Electrical Engineering, Czech Technical University in Prague, 2026. URL https://cw.fel.cvut.cz/wiki/_media/courses/b0b33opt/opt-2up.pdf.

Günter M. Ziegler. *Lectures on Polytopes*. Springer, New York, 1995. doi: 10.1007/978-1-4613-8431-1.