Optimization

Felix Fischer
fischerf@statslab.cam.ac.uk

April 30, 2014

Contents

List of Figures

1 Introduction and Preliminaries

1.1 Constrained Optimization

We consider constrained optimization problems of the form

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & h(x) = b \\ & x \in X. \end{aligned} \tag{1.1}$$

Such a problem is given by a vector $x \in \mathbb{R}^n$ of *decision variables*, an *objective function* $f : \mathbb{R}^n \to \mathbb{R}$, a *functional constraint* $h(x) = b$ where $h : \mathbb{R}^n \to \mathbb{R}^m$ and $b \in \mathbb{R}^m$, and a *regional constraint* $x \in X$ where $X \subseteq \mathbb{R}^n$. The set $X(b) = \{ x \in X : h(x) = b \}$ is called the *feasible set*, and a problem is called *feasible* if $X(b)$ is non-empty and *bounded* if $f(x)$ is bounded from below on $X(b)$. A solution x^* is called *optimal* if it feasible and minimizes f among all feasible solutions.

The assumption that the functional constraint holds with equality is without loss of generality: an inequality constraint like $h(x) \leqslant b$ can be re-written as $h(x) + z = b$, where z is a new *slack variable* with the additional regional constraint $z \geqslant 0$.

Since minimization of $f(x)$ and maximization of $-f(x)$ are equivalent, we will often concentrate on one of the two.

1.2 Linear Programs

The special case where the objective function and constraints are linear is called a linear program (LP). In matrix-vector notation we can write an LP as

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & a_i^T x \geqslant b_i, \quad i \in M_1 \\ & a_i^T x \leqslant b_i, \quad i \in M_2 \\ & a_i^T x = b_i, \quad i \in M_3 \\ & x_j \geqslant 0, \quad j \in N_1 \\ & x_j \leqslant 0, \quad j \in N_2 \end{aligned}$$

where $c \in \mathbb{R}^n$ is a cost vector, $x \in \mathbb{R}^n$ is a vector of decision variables, and constraints are given by $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$ for $i \in \{1, \dots, m\}$. Index sets $M_1, M_2, M_3 \subseteq \{1, \dots, m\}$ and $N_1, N_2 \subseteq \{1, \dots, n\}$ are used to distinguish between different types of contraints.

An equality constraint $a_i^T x = b_i$ is equivalent to the pair of constraints $a_i^T \leqslant b_i$ and $a_i^T x \geqslant b_i$, and a constraint of the form $a_i^T x \leqslant b_i$ can be rewritten as $(-a_i)^T x \geqslant -b_i$. Each occurrence of an unconstrained variable x_j can be replaced by $x_j^+ + x_j^-$, where x_j^+

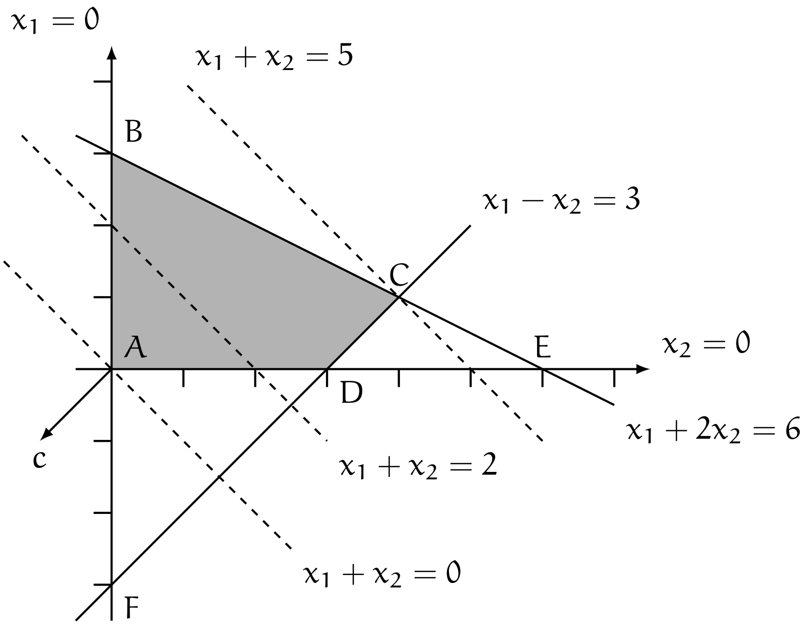

Figure 1.1: Geometric interpretation of the linear program in Example 1.1

and x_j^- are two new variables with $x_j^+ \geqslant 0$ and $x_j^- \leqslant 0$. We can thus write every linear program in the *general form*

$$\min\{\, c^T x : Ax \geqslant b, x \geqslant 0 \,\} \tag{1.2}$$

where $x, c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and $A \in \mathbb{R}^{m \times n}$. Observe that constraints of the form $x_j \geqslant 0$ and $x_j \leqslant 0$ are just special cases of constraints of the form $a_i^T x \geqslant b_i$, but we often choose to make them explicit.

A linear program of the form

$$\min\{\, c^T x : Ax = b, x \geqslant 0 \,\} \tag{1.3}$$

is said to be in *standard form*. The standard form is of course a special case of the general form. On the other hand, we can also bring every general form problem into the standard form by replacing each inequality constraint of the form $a_i^T x \leqslant b_i$ or $a_i^T x \geqslant b_i$ by a constraint $a_i^T x + s_i = b_i$ or $a_i^T x - s_i = b_i$, where s_i is a new slack variable, and an additional constraint $s_i \geqslant 0$.

The general form is typically used to discuss the theory of linear programming, while the standard form is often more convenient when designing algorithms.

EXAMPLE 1.1. Consider the following linear program, which is illustrated in Figure 1.1:

$$\begin{aligned} \text{minimize} \quad & -(x_1 + x_2) \\ \text{subject to} \quad & x_1 + 2x_2 \leqslant 6 \\ & x_1 - x_2 \leqslant 3 \\ & x_1, x_2 \geqslant 0 \end{aligned}$$

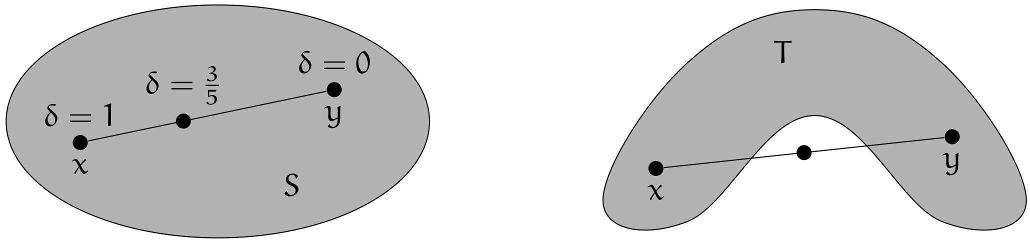

Figure 1.2: A convex set S and a non-convex set T

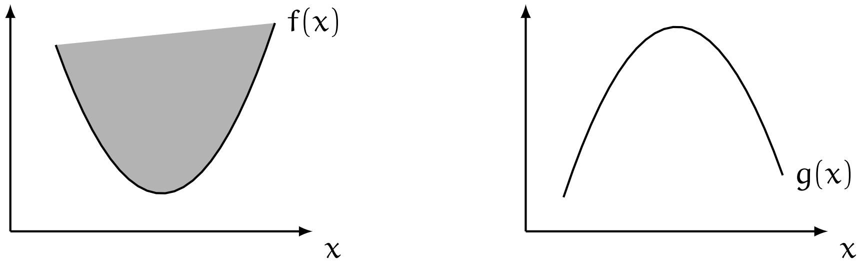

Figure 1.3: A convex function f and a concave function g

Solid lines indicate sets of points for which one of the constraints is satisfied with equality. The feasible set is shaded. Dashed lines, orthogonal to the cost vector c, indicate sets of points for which the value of the objective function is constant.

1.3 Review: Unconstrained Optimization and Convexity

Consider a function $f : \mathbb{R}^n \to \mathbb{R}$, and let $x \in \mathbb{R}^n$. A necessary condition for x to minimize f over $\mathbb{R}^n$ is that $\nabla f(x) = 0$, where

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right)^T$$

is the *gradient* of f. A general function f may have many local minima on the feasible set X, which makes it difficult to find a global minimum. However, if X is convex, and f is convex on X, then any local minimum of f on X is also a global minimum on X.

Let $S \subseteq \mathbb{R}^n$. S is called a *convex set* if for all $\delta \in [0, 1]$, $x, y \in S$ implies that $\delta x + (1 - \delta)y \in S$. An illustration in show in Figure 1.2. A function $f : S \to \mathbb{R}$ is called *convex function* if the set of points above its graph is convex, i.e., if for all $x, y \in S$ and $\delta \in [0, 1]$, $\delta f(x) + (1 - \delta)f(y) \geqslant f(\delta x + (1 - \delta)y)$. Function f is *concave* if $-f$ is convex. An illustration is shown in Figure 1.3.

If f is twice differentiable, it is convex on a convex set S if its *Hessian*

$$\mathcal{H}f = \left(\frac{\partial^2 f}{\partial x_i \partial x_j} \right)_{ij}$$

is positive semidefinite on S. A symmetric $n \times n$ matrix A is called positive semidefinite if $v^T A v \geqslant 0$ for all $v \in \mathbb{R}^n$, or equivalently, if all eigenvalues of A are non-negative.

Theorem 1.2. *Let* $X \subseteq \mathbb{R}^n$ *be convex,* $f : \mathbb{R}^n \to \mathbb{R}$ *twice differentiable on* X*. Let* $\nabla f(x^*) = 0$ *for* $x^* \in X$ *and* $\mathcal{H}f(x)$ *positive semidefinite for all* $x \in X$*. Then* x^* *minimizes* $f(x)$ *subject to* $x \in X$*.*

It is easy to see that in the case of LPs, the feasible set is convex and the objective function is both convex and concave. But even when these two conditions are satisfied, the above theorem cannot generally be used to solve constrained optimization problems, because the gradient might not be zero anywhere on the feasible set.

2 The Method of Lagrange Multipliers

A well-known method for solving constrained optimization problems is the method of Lagrange multipliers. The idea behind this method is to reduce constrained optimization to unconstrained optimization, and to take the (functional) constraints into account by augmenting the objective function with a weighted sum of them. To this end, define the *Lagrangian* associated with (1.1) as

$$L(x, \lambda) = f(x) - \lambda^T (h(x) - b), \tag{2.1}$$

where $\lambda \in \mathbb{R}^m$ is a vector of *Lagrange multipliers*.

2.1 Lagrangian Sufficiency

The following result provides a condition under which minimizing the Lagrangian, subject only to the regional constraints, yields a solution to the original constrained problem. The result is easy to prove, yet extremely useful in practice.

THEOREM 2.1 (Lagrangian Sufficiency Theorem). *Let* $x \in X$ *and* $\lambda \in \mathbb{R}^m$ *such that* $L(x, \lambda) = \inf_{x' \in X} L(x', \lambda)$ *and* $h(x) = b$*. Then* x *is an optimal solution of* (1.1)*.*

Proof. We have that

$$\begin{aligned} \min_{x' \in X(b)} f(x') &= \min_{x' \in X(b)} [f(x') - \lambda^T (h(x') - b)] \\ &\geqslant \min_{x' \in X} [f(x') - \lambda^T (h(x') - b)] \\ &= f(x) - \lambda^T (h(x) - b) = f(x). \end{aligned}$$

Equality in the first line holds because $h(x') - b = 0$ when $x' \in X(b)$. The inequality on the second line holds because the minimum is taken over a larger set. In the third line we finally use that x minimizes L and that $h(x) = b$. □

Two remarks are in order. First, a vector λ of Lagrange multipliers satisfying the conditions of the theorem is not guaranteed to exist in general, but it does exist for a large class of problems. Second, the theorem appears to be useful mainly for showing that a given solution x is optimal. In certain cases, however, it can also be used to find an optimal solution. Our general strategy in these cases will be to minimize $L(x, \lambda)$ for all values of λ, in order to obtain a minimizer $x^*(\lambda)$ that depends on λ, and then find λ^* such that $x^*(\lambda^*)$ satisfies the constraints.

2.2 Using Lagrangian Sufficiency

We begin by applying Theorem 2.1 to a concrete example.

EXAMPLE 2.2. Assume that we want to

$$\begin{aligned} \text{minimize} \quad & x_1 - x_2 - 2x_3 \\ \text{subject to} \quad & x_1 + x_2 + x_3 = 5 \\ & x_1^2 + x_2^2 = 4. \end{aligned}$$

The Lagrangian of this problem is

$$\begin{aligned} L(x, \lambda) &= x_1 - x_2 - 2x_3 - \lambda_1(x_1 + x_2 + x_3 - 5) - \lambda_2(x_1^2 + x_2^2 - 4) \\ &= \Big((1-\lambda_1)x_1 - \lambda_2 x_1^2\Big) + \Big((-1-\lambda_1)x_2 - \lambda_2 x_2^2\Big) + \Big((-2-\lambda_1)x_3\Big) + 5\lambda_1 + 4\lambda_2. \end{aligned}$$

For a given value of λ, we can minimize $L(x, \lambda)$ by independently minimizing the terms in x_1, x_2, and x_3, and we will only be interested in values of λ for which the minimum is finite.

The term $(-2 - \lambda_1)x_3$ does not have a finite minimum unless $\lambda_1 = -2$. The terms in x_1 and x_2 then have a finite minimum only if $\lambda_2 < 0$, in which case an optimum occurs when

$$\begin{aligned} \frac{\partial L}{\partial x_1} &= 1 - \lambda_1 - 2\lambda_2 x_1 = 3 - 2\lambda_2 x_1 = 0 \quad \text{and} \\ \frac{\partial L}{\partial x_2} &= -1 - \lambda_1 - 2\lambda_2 x_2 = 1 - 2\lambda_2 x_2 = 0, \end{aligned}$$

i.e., when $x_1 = 3/(2\lambda_2)$ and $x_2 = 1/(2\lambda_2)$. The optimum is indeed a minimum, because

$$\mathcal{H}L = \begin{pmatrix} \frac{\partial^2 L}{\partial x_1 \partial x_1} & \frac{\partial^2 L}{\partial x_1 \partial x_2} \\ \frac{\partial^2 L}{\partial x_2 \partial x_1} & \frac{\partial^2 L}{\partial x_2 \partial x_2} \end{pmatrix} = \begin{pmatrix} -2\lambda_2 & 0 \\ 0 & -2\lambda_2 \end{pmatrix},$$

is positive semidefinite when $\lambda_2 < 0$.

Let Y be the set of values of λ such that $L(x, \lambda)$ has a finite minimum, i.e.,

$$Y = \{\lambda \in \mathbb{R}^2 : \lambda_1 = -2, \lambda_2 < 0\}.$$

For every $\lambda \in Y$, the unique optimum of $L(x, \lambda)$ occurs at $x^*(\lambda) = (3/(2\lambda_2), 1/(2\lambda_2), x_3)^T$, and we need to find $\lambda \in Y$ such that $x^*(\lambda)$ is feasible to be able to apply Theorem 2.1. Therefore,

$$x_1^2 + x_2^2 = \frac{9}{4\lambda_2^2} + \frac{1}{4\lambda_2^2} = 4$$

and thus $\lambda_2 = -\sqrt{5/8}$. We can now use Theorem 2.1 to conclude that the minimization problem has an optimal solution at $x_1 = -3\sqrt{2/5}$, $x_2 = -\sqrt{2/5}$, and $x_3 = 5 - x_1 - x_2 = 5 + 4\sqrt{2/5}$.

Let us formalize the strategy we have used to find x and λ satisfying the conditions of Theorem 2.1 for a more general problem. To

$$\text{minimize } f(x) \text{ subject to } h(x) \leqslant b,\ x \in X \tag{2.2}$$

we proceed as follows:

1. Introduce a vector z of slack variables to obtain the equivalent problem

$$\text{minimize } f(x) \text{ subject to } h(x) + z = b,\ x \in X,\ z \geqslant 0.$$

2. Compute the Lagrangian $L(x, z, \lambda) = f(x) - \lambda^T(h(x) + z - b)$.
3. Define the set

$$Y = \{\lambda \in \mathbb{R}^m : \inf_{x \in X, z \geqslant 0} L(x, z, \lambda) > -\infty\}.$$

4. For each $\lambda \in Y$, minimize $L(x, z, \lambda)$ subject only to the regional constraints, i.e., find $x^*(\lambda), z^*(\lambda)$ satisfying

$$L(x^*(\lambda), z^*(\lambda), \lambda) = \inf_{x \in X, z \geqslant 0} L(x, z, \lambda). \tag{2.3}$$

5. Find $\lambda^* \in Y$ such that $(x^*(\lambda^*), z^*(\lambda^*))$ is feasible, i.e., such that $x^*(\lambda^*) \in X$, $z^*(\lambda^*) \geqslant 0$, and $h(x^*(\lambda^*)) + z^*(\lambda^*) = b$. By Theorem 2.1, $x^*(\lambda^*)$ is optimal for (2.2).

2.3 Complementary Slackness

It is worth pointing out a property known as *complementary slackness*, which follows directly from (2.3): for every $\lambda \in Y$ and $i = 1, \ldots, m$,

$$\begin{aligned} (z^*(\lambda))_i \neq 0 \quad &\text{implies} \quad \lambda_i = 0 \quad \text{and} \\ \lambda_i \neq 0 \quad &\text{implies} \quad (z^*(\lambda))_i = 0. \end{aligned}$$

Indeed, if the conditions were violated for some i, then the value of the Lagrangian could be reduced by reducing $(z^*(\lambda))_i$, while maintaining that $(z^*(\lambda))_i \geqslant 0$. This would contradict (2.3). Further note that $\lambda \in Y$ requires for each $i = 1, \ldots, m$ either that $\lambda_i \leqslant 0$ or that $\lambda_i \geqslant 0$, depending on the sign of b_i. In the case where where $\lambda_i \leqslant 0$, we for example get that

$$\begin{aligned} (h(x^*(\lambda^*)))_i < b_i \quad &\text{implies} \quad \lambda_i^* = 0 \quad \text{and} \\ \lambda_i^* < 0 \quad &\text{implies} \quad (h(x^*(\lambda^*)))_i = b_i. \end{aligned}$$

Slack in the corresponding inequalities $(h(x^*(\lambda^*)))_i \leqslant b_i$ and $\lambda_i^* \leqslant 0$ has to be complementary, in the sense that it cannot occur simultaneously in both of them.

EXAMPLE 2.3. Consider the problem to

$$\begin{aligned} \text{minimize} \quad & x_1 - 3x_2 \\ \text{subject to} \quad & x_1^2 + x_2^2 \leqslant 4 \\ & x_1 + x_2 \leqslant 2. \end{aligned}$$

By adding slack variables, we obtain the following equivalent problem:

$$\begin{aligned} \text{minimize} \quad & x_1 - 3x_2 \\ \text{subject to} \quad & x_1^2 + x_2^2 + z_1 = 4 \\ & x_1 + x_2 + z_2 = 2 \\ & z_1 \geqslant 0, z_2 \geqslant 0. \end{aligned}$$

The Lagrangian of this problem is

$$\begin{aligned} L(x, z, \lambda) &= x_1 - 3x_2 - \lambda_1(x_1^2 + x_2^2 + z_1 - 4) - \lambda_2(x_1 + x_2 + z_2 - 2) \\ &= \left((1 - \lambda_2)x_1 - \lambda_1 x_1^2\right) + \left((-3 - \lambda_2)x_2 - \lambda_1 x_2^2\right) - \lambda_1 z_1 - \lambda_2 z_2 + 4\lambda_1 + 2\lambda_2. \end{aligned}$$

Since $z_1 \geqslant 0$ and $z_2 \geqslant 0$, the terms $-\lambda_1 z_1$ and $-\lambda_2 z_2$ have a finite minimum only if $\lambda_1 \leqslant 0$ and $\lambda_2 \leqslant 0$. In addition, the complementary slackness conditions $\lambda_1 z_1 = 0$ and $\lambda_2 z_2 = 0$ must hold at the optimum.

Minimizing $L(x, z, \lambda)$ in x_1 and x_2 yields

$$\begin{aligned} \frac{\partial L}{\partial x_1} &= 1 - \lambda_2 - 2\lambda_1 x_1 = 0 \quad \text{and} \\ \frac{\partial L}{\partial x_2} &= -3 - \lambda_2 - 2\lambda_1 x_2 = 0, \end{aligned}$$

and we indeed obtain a minimum, because

$$\mathcal{H}L = \begin{pmatrix} \frac{\partial^2 L}{\partial x_1 \partial x_1} & \frac{\partial^2 L}{\partial x_1 \partial x_2} \\ \frac{\partial^2 L}{\partial x_2 \partial x_1} & \frac{\partial^2 L}{\partial x_2 \partial x_2} \end{pmatrix} = \begin{pmatrix} -2\lambda_1 & 0 \\ 0 & -2\lambda_1 \end{pmatrix}$$

is positive semidefinite when $\lambda_1 \leqslant 0$.

Setting $\lambda_1 = 0$ leads to inconsistent values for λ_2, so we must have $\lambda_1 < 0$, and, by complementary slackness, $z_1 = 0$. Also by complementary slackness, there are now two more cases to consider: the one where $\lambda_2 < 0$ and $z_2 = 0$, and the one where $\lambda_2 = 0$. The former case leads to a contradiction, the latter to the unique minimum at $x_1 = -\sqrt{2/5}$ and $x_2 = 3\sqrt{2/5}$.

3 Shadow Prices and Lagrangian Duality

3.1 Shadow Prices

A more intuitive understanding of Lagrange multipliers can be obtained by viewing (1.1) as a family of problems parameterized by $b \in \mathbb{R}^m$, the right hand side of the functional constraints. To this end, let $\phi(b) = \inf\{f(x) : h(x) = b, x \in \mathbb{R}^n\}$. It turns out that at the optimum, the Lagrange multipliers equal the partial derivatives of ϕ with respect to its parameters.

THEOREM 3.1. *Suppose that* f *and* h *are continuously differentiable on* $\mathbb{R}^n$*, and that there exist unique functions* $x^* : \mathbb{R}^m \to \mathbb{R}^n$ *and* $\lambda^* : \mathbb{R}^m \to \mathbb{R}^m$ *such that for each* $b \in \mathbb{R}^m$*,* $h(x^*(b)) = b$*,* $\lambda^*(b) \leqslant 0$ *and* $f(x^*(b)) = \phi(b) = \inf\{f(x) - \lambda^*(b)^T(h(x) - b) : x \in \mathbb{R}^n\}$*. If* x^* *and* λ^* *are continuously differentiable, then*

$$\frac{\partial \phi}{\partial b_i}(b) = \lambda_i^*(b).$$

Proof. We have that

$$\begin{aligned}\phi(b) &= f(x^*(b)) - \lambda^*(b)^T(h(x^*(b)) - b) \\ &= f(x^*(b)) - \lambda^*(b)^T h(x^*(b)) + \lambda^*(b)^T b.\end{aligned}$$

Taking partial derivatives of each term,

$$\frac{\partial f(x^*(b))}{\partial b_i} = \sum_{j=1}^{n} \frac{\partial f}{\partial x_j}(x^*(b)) \frac{\partial x_j^*}{\partial b_i}(b),$$

$$\begin{aligned}\frac{\partial \lambda^*(b)^T h(x^*(b))}{\partial b_i} &= \lambda^*(b)^T \frac{\partial h(x^*(b))}{\partial b_i} + h(x^*(b)) \frac{\partial \lambda^*(b)^T}{\partial b_i} \\ &= \left(\sum_{j=1}^{n} \left(\lambda^*(b)^T \frac{\partial h}{\partial x_j}(x^*(b)) \right) \frac{\partial x_j^*}{\partial b_i}(b) \right) + h(x^*(b)) \frac{\partial \lambda^*(b)^T}{\partial b_i},\end{aligned}$$

$$\frac{\partial \lambda^*(b)^T b}{\partial b_i} = \lambda^*(b)^T \frac{\partial b}{\partial b_i} + b \frac{\lambda^*(b)^T}{\partial b_i}.$$

By summing and re-arranging,

$$\begin{aligned}\frac{\partial \phi(b)}{\partial b_i} = \sum_{j=1}^{n} \left(\frac{\partial f}{\partial x_j}(x^*(b)) - \lambda^*(b)^T \frac{\partial h}{\partial x_j}(x^*(b)) \right) \frac{\partial x_j^*}{\partial b_i}(b) \\ - (h(x^*(b)) - b) \frac{\partial \lambda^*(b)^T}{\partial b_i} + \lambda^*(b)^T \frac{\partial b}{\partial b_i}.\end{aligned}$$

The first term on the right-hand side is zero, because $x^*(b)$ minimizes $L(x, \lambda^*(b))$ and thus

$$\frac{\partial L(x^*(b), \lambda^*(b))}{\partial x_j} = \frac{\partial f}{\partial x_j}(x^*(b)) - \left(\lambda^*(b)^T \frac{\partial h}{\partial x_j}(x^*(b))\right) = 0$$

for $j = 1, \dots, n$. The second term is zero as well, because $x^*(b)$ is feasible and thus $(h(x^*(b)) - b)_k = 0$ for $k = 1, \dots, m$, and the claim follows. □

It should be noted that the result also holds when the functional constraints are inequalities: if the ith constraint does not not hold with equality, then $\lambda_i^* = 0$ by complementary slackness, and therefore also $\partial \lambda_i^* / \partial b_i = 0$.

The Lagrange multipliers are also known as *shadow prices*, due to an economic interpretation of the problem to

$$\begin{aligned} \text{maximize} \quad & f(x) \\ \text{subject to} \quad & h(x) \leqslant b \\ & x \in X. \end{aligned}$$

Consider a firm that produces n different goods from m different raw materials. Vector $b \in \mathbb{R}^m$ describes the amount of each raw material available to the firm, vector $x \in \mathbb{R}^n$ the quantity produced of each good. Functions $h : \mathbb{R}^n \to \mathbb{R}^m$ and $f : \mathbb{R}^n \to \mathbb{R}$ finally describe the amounts of raw material required to produce, and the profit derived from producing, particular quantities of the goods. The goal of the above problem thus is to maximize the profit of the firm for given amounts of raw materials available to it. The *shadow price* of raw material i then is the price the firm would be willing to pay per additional unit of this raw material, which of course should be equal to the additional profit derived from it, i.e., to $\frac{\partial \phi}{\partial b_i}(b)$.

3.2 Lagrangian Duality

Another useful concept that arises from Lagrange multipliers is that of a dual problem. Again consider the optimization problem (1.1) and the Lagrangian (2.1), and define the (Lagrange) *dual function* $g : \mathbb{R}^m \to \mathbb{R}$ as the minimum value of the Lagrangian over X, i.e.,

$$g(\lambda) = \inf_{x \in X} L(x, \lambda). \tag{3.1}$$

As before, let Y be the set vectors of Lagrange multipliers for which the Lagrangian has a finite minimum, i.e., $Y = \{\lambda \in \mathbb{R}^m : \inf_{x \in X} L(x, \lambda) > -\infty\}$.

It is easy to see that the maximum value of the dual function provides a lower bound on the minimum value of the original objective function. This property is known as weak duality.

Theorem 3.2 (Weak duality). *If $x \in X(b)$ and $\lambda \in Y$, then $g(\lambda) \leqslant f(x)$, and in particular,*

$$\sup_{\lambda \in Y} g(\lambda) \leqslant \inf_{x \in X(b)} f(x). \tag{3.2}$$

Proof. Let $x \in X(b)$ and $\lambda \in Y$. Then,

$$\begin{aligned} g(\lambda) &= \inf_{x' \in X} L(x', \lambda) \\ &\leqslant L(x, \lambda) \\ &= f(x) - \lambda^T (h(x) - b) \\ &= f(x). \end{aligned}$$

□

Equality on the first and third line holds by definition of g and L, the inequality on the second line because $x \in X$. The last equality holds because $x \in X(b)$ and therefore $h(x) - b = 0$.

In light of this result, it is interesting to choose λ in order to make this lower bound as large as possible, i.e., to

$$\begin{aligned} &\text{maximize} && g(\lambda) \\ &\text{subject to} && \lambda \in Y. \end{aligned}$$

This problem is known as the *dual problem*, and (1.1) is in this context referred to as the *primal problem*. If (3.2) holds with equality, i.e., if there exists $\lambda \in Y$ such that $g(\lambda) = \inf_{x \in X(b)} f(x)$, the problem is said to satisfy *strong duality*. The cases where strong duality holds are those that can be solved using the method of Lagrange multipliers.

EXAMPLE 3.3. Again consider the minimization problem of Example 2.2, and recall that $Y = \{\lambda \in \mathbb{R}^2 : \lambda_1 = -2, \lambda_2 < 0\}$ and that for each $\lambda \in Y$ the minimum occurred at $x^*(\lambda) = (3/(2\lambda_2), 1/(2\lambda_2), x_3)$. Thus,

$$g(\lambda) = \inf_{x \in X} L(x, \lambda) = L(x^*(\lambda), \lambda) = \frac{10}{4\lambda_2} + 4\lambda_2 - 10,$$

so the dual problem is to

$$\text{maximize } \frac{10}{4\lambda_2} + 4\lambda_2 - 10 \text{ subject to } \lambda_2 < 0.$$

It should not come as a surprise that the maximum is attained for $\lambda_2 = -\sqrt{5/8}$, and that the primal and dual have the same optimal value, namely $-2(\sqrt{10}+5)$. Note that it is not actually necessary to solve the dual to see that $\lambda_2 = -\sqrt{5/8}$ is an optimizer, it suffices that the value of the dual function at this point equals the value of the objective function of the primal at some point in the feasible set of the primal.

There are several reasons why the dual is interesting. Any feasible solution of the dual provides a succinct certificate that the optimal solution of the primal is bounded by a certain value. In particular, a pair of solutions of the primal and dual that yield the same value must be optimal. If strong duality holds, the optimal value of the primal can be determined by solving the dual, which in some cases may be easier than solving the primal. In a later lecture we will express two quantities as the optimal solutions of a pair of a primal and a dual that satisfy strong duality, thereby showing that the two quantities are equal.

4 Conditions for Strong Duality

While we have already solved a few optimization problems using the method of Lagrange multipliers, it was not clear a priori whether each individual problem satisfied strong duality and whether our attempt to solve it would ultimately be successful. Our goal in this lecture will be to identify general conditions that guarantee strong duality, and classes of problems that satisfy these conditions.

4.1 Supporting Hyperplanes and Convexity

To this end, we again consider the function ϕ that describes how the optimal value behaves as we vary the right-hand side of the constraints. Fix a particular $b \in \mathbb{R}^m$ and consider $\phi(c)$ as a function of $c \in \mathbb{R}^m$. Further consider the hyperplane given by $\alpha : \mathbb{R}^m \to \mathbb{R}$ with

$$\alpha(c) = \beta + \lambda^T(c - b).$$

This hyperplane has intercept β at b and slope λ. We can now try to find $\phi(b)$ as follows:

1. For each λ, find $\beta_\lambda = \sup\{\beta : \beta + \lambda^T(c - b) \leqslant \phi(c) \text{ for all } c \in \mathbb{R}^m\}$.
2. Choose λ to maximize β_λ.

This approach is illustrated in Figure 4.1. We always have that $\beta_\lambda \leqslant \phi(b)$. In the situation on the left of Figure 4.1, this condition holds with equality because there is a tangent to ϕ at b. In fact,

$$\begin{aligned}
g(\lambda) &= \inf_{x \in X} L(x, \lambda) \\
&= \inf_{x \in X} \big(f(x) - \lambda^T(h(x) - b)\big) \\
&= \inf_{c \in \mathbb{R}^m} \inf_{x \in X(c)} \big(f(x) - \lambda^T(h(x) - b)\big) \\
&= \inf_{c \in \mathbb{R}^m} \big(\phi(c) - \lambda^T(c - b)\big) \\
&= \sup\big\{\beta : \beta + \lambda^T(c - b) \leqslant \phi(c) \text{ for all } c \in \mathbb{R}^m\big\} \\
&= \beta_\lambda
\end{aligned}$$

We again see the weak duality result as $\max_\lambda \beta_\lambda \leqslant \phi(b)$, but we also obtain a condition for strong duality. Call $\alpha : \mathbb{R}^m \to \mathbb{R}$ a *supporting hyperplane* to ϕ at b if $\alpha(c) = \phi(b) + \lambda^T(c - b)$ and $\phi(c) \geqslant \phi(b) + \lambda^T(c - b)$ for all $c \in \mathbb{R}^m$.

Theorem 4.1. *Problem* (1.1) *satisfies strong duality if and only if there exists a (non-vertical) supporting hyperplane to* ϕ *at* b.

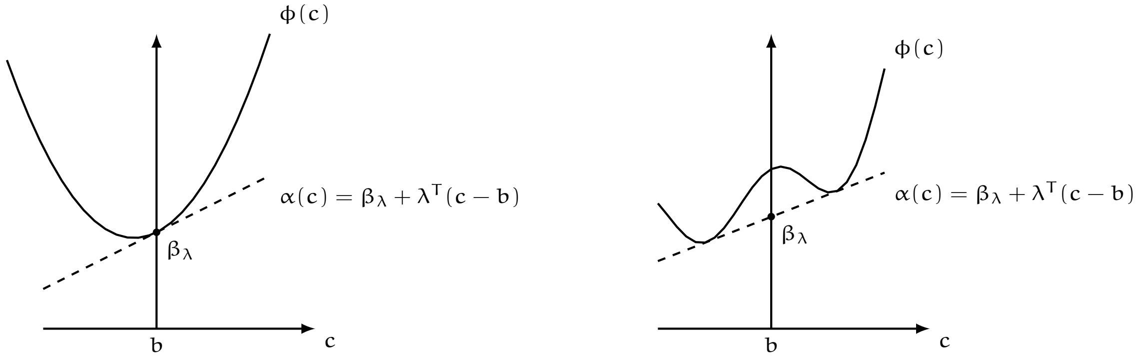

Figure 4.1: Geometric interpretation of the dual with optimal value $g(\lambda) = \beta_\lambda$. In the situation on the left strong duality holds, and $\beta_\lambda = \phi(b)$. In the situation on the right, strong duality does not hold, and $\beta_\lambda < \phi(b)$.

Proof. Suppose there exists a (non-vertical) supporting hyperplane to ϕ at b. This means that there exists $\lambda \in \mathbb{R}^m$ such that for all $c \in \mathbb{R}^m$,

$$\phi(b) + \lambda^T(c - b) \leqslant \phi(c).$$

This implies that

$$\begin{aligned} \phi(b) &\leqslant \inf_{c \in \mathbb{R}^m} \big(\phi(c) - \lambda^T(c - b)\big) \\ &= \inf_{c \in \mathbb{R}^m} \inf_{x \in X(c)} \big(f(x) - \lambda^T(h(x) - b)\big) \\ &= \inf_{x \in X} L(x, \lambda) \\ &= g(\lambda). \end{aligned}$$

On the other hand, $\phi(b) \geqslant g(\lambda)$ by Theorem 3.2, so $\phi(b) = g(\lambda)$ and strong duality holds.

Now suppose that the problem satisfies strong duality. Then there exists $\lambda \in \mathbb{R}^m$ such that for all $c \in \mathbb{R}^m$

$$\begin{aligned} \phi(b) = g(\lambda) &= \inf_{x \in X} L(x, \lambda) \\ &\leqslant \inf_{x \in X(c)} L(x, \lambda) \\ &= \inf_{x \in X(c)} (f(x) - \lambda^T(h(x) - b)) \\ &= \phi(c) - \lambda^T(c - b), \end{aligned}$$

and thus

$$\phi(b) + \lambda^T(c - b) \leqslant \phi(c).$$

This describes a (non-vertical) supporting hyperplane to ϕ at b. □

A sufficient condition for the existence of a supporting hyperplane is provided by the following basic result, which we state without proof.

THEOREM 4.2 (Supporting Hyperplane Theorem). *Suppose that* $\phi : \mathbb{R}^m \to \mathbb{R}$ *is convex and* $b \in \mathbb{R}^m$ *lies in the interior of the set of points where* ϕ *is finite. Then there exists a (non-vertical) supporting hyperplane to* ϕ *at* b.

4.2 A Sufficient Condition for Convexity

We now know that convexity of ϕ guarantees strong duality for every constraint vector b, but it is not clear how to recognize optimization problems that have this property. The following result identifies a sufficient condition.

THEOREM 4.3. *Consider the problem to*

$$\begin{aligned} \textit{minimize} \quad & f(x) \\ \textit{subject to} \quad & h(x) \leqslant b \\ & x \in X, \end{aligned}$$

and let ϕ *be given by* $\phi(b) = \inf_{x \in X(b)} f(x)$. *Then,* ϕ *is convex if* X, f, *and* h *are convex.*

Proof. Consider $b_1, b_2 \in \mathbb{R}^m$ such that $\phi(b_1)$ and $\phi(b_2)$ are defined, and let $\delta \in [0, 1]$ and $b = \delta b_1 + (1 - \delta) b_2$. Further consider $x_1 \in X(b_1)$, $x_2 \in X(b_2)$, and let $x = \delta x_1 + (1 - \delta) x_2$. Then convexity of X implies that $x \in X$, and convexity of h that

$$\begin{aligned} h(x) &= h(\delta x_1 + (1 - \delta) x_2) \\ &\leqslant \delta h(x_1) + (1 - \delta) h(x_2) \\ &\leqslant \delta b_1 + (1 - \delta) b_2 \\ &= b. \end{aligned}$$

Thus $x \in X(b)$, and by convexity of f,

$$\phi(b) \leqslant f(x) = f(\delta x_1 + (1 - \delta) x_2) \leqslant \delta f(x_1) + (1 - \delta) f(x_2).$$

This holds for all $x_1 \in X(b_1)$ and $x_2 \in X(b_2)$, so taking infima on the right hand side yields

$$\phi(b) \leqslant \delta \phi(b_1) + (1 - \delta) \phi(b_2).$$

□

Note that an equality constraint $h(x) = b$ is equivalent to the pair of constraints $h(x) \leqslant b$ and $-h(x) \leqslant -b$. In this case, the above result requires that X, f, h, and $-h$ are all convex, which in particular requires that h is linear. Indeed, in the case with equality constraints, convexity of f and h does not suffice for convexity of ϕ. To see this, consider the problem to

$$\text{minimize } f(x) = x^2 \text{ subject to } h(x) = x^3 = b$$

for some $b > 0$. Then $\phi(b) = b^{2/3}$, which is not convex. The Lagrangian is $L(x, \lambda) = x^2 - \lambda(x^3 - b) = (x^2 - \lambda x^3) + \lambda b$, and has a finite minimum if and only if $\lambda = 0$. The dual thus has an optimal value of 0, which is strictly smaller than $\phi(b)$ if $b > 0$.

Linear programs satisfy the conditions, both for equality and inequality constraints. We thus have the following.

THEOREM 4.4. *If a linear program is feasible and bounded, it satisfies strong duality.*

5 Solutions of Linear Programs

In the remaining lectures, we will concentrate on linear programs. We begin by studying the special structure of the feasible set and the objective function in this case, and how it affects the set of optimal solutions.

5.1 Basic Solutions

In the LP of Example 1.1, the optimal solution happened to lie at an extreme point of the feasible set. This was not a coincidence. Consider an LP in general form,

$$\text{maximize } c^T x \text{ subject to } Ax \leqslant b,\ x \geqslant 0. \tag{5.1}$$

The feasible set of this LP is a convex polytope in $\mathbb{R}^n$, i.e., an intersection of half-spaces. Each level set of the objective function $c^T x$, i.e., each set $L_\alpha = \{x \in \mathbb{R}^n : c^T x = \alpha\}$ of points for which the value of the objective function is equal to some constant $\alpha \in \mathbb{R}$, is a k-dimensional flat for some $k \leqslant n$. The goal is to find the largest value of α for which $L_\alpha(f)$ intersects with the feasible set. If such a value exists, the intersection contains either a single point or an infinite number of points, and it is guaranteed to contain an extreme point of the feasible set. This fact is illustrated in Figure 5.1, and we will give a proof momentarily.

Formally, $x \in S$ is an *extreme point* of a convex set S if it cannot be written as a convex combination of two distinct points in S, i.e., if for all $y, z \in S$ and $\delta \in (0, 1)$, $x = \delta y + (1-\delta)z$ implies that $x = y = z$. Since this geometric characterization of extreme points is hard to work with, we consider an alternative, algebraic characterization. To this end, consider the following LP in standard form, which can be obtained from (5.1) by introducing slack variables:

$$\text{maximize } c^T x \text{ subject to } Ax = b,\ x \geqslant 0, \tag{5.2}$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Call a solution $x \in \mathbb{R}^n$ of the equation $Ax = b$ *basic* if at most m of its entries are non-zero, i.e., if there exists a set $B \subseteq \{1, \dots, n\}$ with $|B| = m$ such that $x_i = 0$ if $i \notin B$. The set B is then called *basis*, and variable x_i is called *basic* if $i \in B$ and *non-basic* if $i \notin B$. A basic solution x that also satisfies $x \geqslant 0$ is a *basic feasible solution* (BFS) of (5.2).

We will henceforth make the following assumptions:

(i) the rows of A are linearly independent,

(ii) every set of m columns of A are linearly independent, and

(iii) every basic solution is *non-degenerate*, i.e., has exactly m non-zero variables.

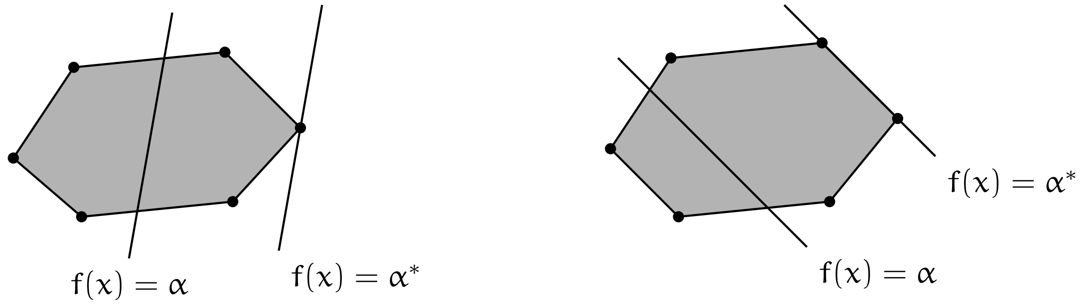

Figure 5.1: Illustration of linear programs with one optimal solution (left) and an infinite number of optimal solutions (right)

Assumptions (i) and (ii) are without loss of generality: if a set of rows are linearly dependent, one of the corresponding constraints can be removed without changing the feasible set; similarly, if a set of columns are linearly dependent, one of the corresponding variables can be removed. Extra care needs to be taken to handle degeneracies, but this is beyond the scope of this course.

If the above assumptions are satisfied, setting any subset of $n-m$ variables to zero uniquely determines the value of the remaining, basic variables. Computing the set of basic feasible solutions is thus straightforward.

EXAMPLE 5.1. Again consider the LP of Example 1.1. By adding slack variables $x_3 \geqslant 0$ and $x_4 \geqslant 0$, the functional constraint can be written as

$$\begin{pmatrix} 1 & 2 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 6 \\ 3 \end{pmatrix}.$$

The problem has the following six basic solutions corresponding to the $\binom{4}{2}$ possible ways to choose a basis, which are labeled A through F in Figure 1.1:

	x_1	x_2	x_3	x_4	$f(x)$
A	0	0	6	3	0
B	0	3	0	6	3
C	4	1	0	0	5
D	3	0	3	0	3
E	6	0	0	−3	6
F	0	−3	12	0	−3

5.2 Extreme Points and Optimal Solutions

It turns out that the basic feasible solutions are precisely the extreme points of the feasible set.

THEOREM 5.2. *A vector is a basic feasible solution of* $Ax = b$ *if and only if it is an extreme point of the set* $X(b) = \{x : Ax = b, x \geqslant 0\}$.

Proof. Consider a BFS x and suppose that $x = \delta y + (1 - \delta)z$ for $y, z \in X(b)$ and $\delta \in (0, 1)$. Since $y \geqslant 0$ and $z \geqslant 0$, $x = \delta y + (1 - \delta)z$ implies that $y_i = z_i$ whenever $x_i = 0$. By (iii), y and z are basic solutions with the same basis, i.e., both have exactly m non-zero entries, which occur in the same rows. Moreover, $Ay = b = Az$ and thus $A(y - z) = 0$. This yields a linear combination of m columns of A that is equal to zero, which by (ii) implies that $y = z$. Thus x is an extreme point of $X(b)$.

Now consider a feasible solution $x \in X(b)$ that is not a BFS. Let $i_1, \ldots, i_r$ be the rows of x that are non-zero, and observe that $r > m$. This means that the columns $a^{i_1}, \ldots, a^{i_r}$, where $a^i = (a_{1i}, \ldots, a_{mi})^T$, have to be linearly dependent, i.e., there has to exist a collection of r non-zero numbers $y_{i_1}, \ldots, y_{i_r}$ such that $y_{i_1} a^{i_1} + \cdots + y_{i_r} a^{i_r} = 0$. Extending y to a vector in $\mathbb{R}^n$ by setting $y_i = 0$ if $i \notin \{i_1, \ldots, i_r\}$, we have $Ay = y_{i_1} a^{i_1} + \cdots + y_{i_r} a^{i_r}$ and thus $A(x \pm \epsilon y) = b$ for every $\epsilon \in \mathbb{R}$. By choosing $\epsilon > 0$ small enough, $x \pm \epsilon y \geqslant 0$ and thus $x \pm \epsilon y \in X(b)$. Moreover $x = 1/2(x - \epsilon y) + 1/2(x + \epsilon y)$, so x is not an extreme point of $X(b)$. □

We are now ready to show that an optimum occurs at an extreme point of the feasible set.

THEOREM 5.3. *If the linear program* (5.2) *is feasible and bounded, then it has an optimal solution that is a basic feasible solution.*

Proof. Let x be an optimal solution of (5.2). If x has exactly m non-zero entries, then it is a BFS and we are done. So suppose that x has r non-zero entries for $r > m$, and that it is not an extreme point of $X(b)$, i.e., that $x = \delta y + (1 - \delta)z$ for $y, z \in X(b)$ with $y \neq z$ and $\delta \in (0, 1)$. We will show that there must exist an optimal solution with strictly fewer than r non-zero entries; the claim then follows by induction.

Since $c^T x \geqslant c^T y$ and $c^T x \geqslant c^T z$ by optimality of x, and since $c^T x = \delta c^T y + (1 - \delta) c^T z$, we must have that $c^T x = c^T y = c^T z$, so y and z are optimal as well. As in the proof of Theorem 5.2, $x_i = 0$ implies that $y_i = z_i = 0$, so y and z have at most r non-zero entries, which must occur in the same rows as in x. If y or z has strictly fewer than r non-zero entries, we are done. Otherwise let $x' = \delta' y + (1 - \delta')z = z + \delta'(y - z)$, and observe that x' is optimal for every $\delta' \in \mathbb{R}$. Moreover, $y - z \neq 0$, and all non-zero entries of $y - z$ occur in rows where x is non-zero as well. We can thus choose $\delta' \in \mathbb{R}$ such that $x' \geqslant 0$ and such that x' has strictly fewer than r non-zero entries. □

The result can in fact be extended to show that the maximum of a convex function f over a compact convex set X occurs at an extreme point of X. In this case any point $x \in X$ can be written as a convex combination $x = \sum_{i=1}^{k} \delta_i x^i$ of extreme points $x^1, \dots, x^k \in X$, where $\delta \in \mathbb{R}^k_{\geqslant 0}$ and $\sum_{i=1}^{k} \delta_i = 1$. Convexity of f then implies that

$$f(x) \leqslant \sum_{i=1}^{k} \delta_i f(x^i) \leqslant \max_{1 \leqslant i \leqslant k} f(x^i).$$

5.3 A Naive Approach to Solving Linear Programs

Since there are only finitely many basic solutions, a naive approach to solving an LP would be to go over all basic solutions and pick one that optimizes the objective. The problem with this approach is that it would not in general be efficient, as the number of basic solutions may grow exponentially in the number of variables. By contrast, a large body of work on the theory of computational complexity typically associates efficient computation with methods that for every problem instance can be executed in a number of steps that is at most polynomial in the size of that instance.

In one of the following lectures we will study a well-known method for solving linear programs, the so-called simplex method, which explores the set of basic solutions in a more organized way. It is usually very efficient in practice, but may still require an exponential number of steps for some contrived instances. In fact, no approach is currently know that solves linear programs by inspecting only the boundary of the feasible set and is efficient for every conceivable instance of the problem. There are, however, so-called interior-point method that traverse the interior of the feasible set in search of an optimal solution and are very efficient both in theory and in practice.

6 Linear Programming Duality

Consider the linear program (1.2) and introduce slack variables z to turn it into

$$\min\{\, c^T x : Ax - z = b, x, z \geqslant 0 \,\}.$$

We have $X = \{(x, z) : x \geqslant 0, z \geqslant 0\} \subseteq \mathbb{R}^{m+n}$. The Lagrangian is given by

$$L((x, z), \lambda) = c^T x - \lambda^T (Ax - z - b) = (c^T - \lambda^T A)x + \lambda^T z + \lambda^T b$$

and has a finite minimum over X if and only if

$$\lambda \in Y = \{\, \mu : c^T - \mu^T A \geqslant 0, \mu \geqslant 0 \,\}.$$

For $\lambda \in Y$, the minimum of $L((x, z), \lambda)$ is attained when both $(c^T - \lambda^T A)x = 0$ and $\lambda^T z = 0$, and thus

$$g(\lambda) = \inf_{(x,z)\in X} L((x, z), \lambda) = \lambda^T b.$$

We obtain the dual

$$\max\{\, b^T \lambda : A^T \lambda \leqslant c, \lambda \geqslant 0 \,\}. \tag{6.1}$$

The dual of (1.3) can be determined analogously as

$$\max\{\, b^T \lambda : A^T \lambda \leqslant c \,\}.$$

The dual is itself a linear program, and its dual is in fact equivalent to the primal.

THEOREM 6.1. *In the case of linear programming, the dual of the dual is the primal.*

Proof. The dual can be written equivalently as

$$\min\{\, -b^T \lambda : -A^T \lambda \geqslant -c, \lambda \geqslant 0 \,\}.$$

This problem has the same form as the primal (1.2), with $-b$ taking the role of c, $-c$ taking the role of b, and $-A^T$ the role of A. Taking the dual again we thus return to the original problem. □

6.1 The Relationship between Primal and Dual

EXAMPLE 6.2. Consider the following pair of a primal and dual LP, with slack variables z_1 and z_2 for the primal and μ_1 and μ_2 for the dual.

maximize	$3x_1 + 2x_2$	minimize	$4\lambda_1 + 6\lambda_2$
subject to	$2x_1 + x_2 + z_1 = 4$	subject to	$2\lambda_1 + 2\lambda_2 - \mu_1 = 3$
	$2x_1 + 3x_2 + z_2 = 6$		$\lambda_1 + 3\lambda_2 - \mu_2 = 2$
	$x_1, x_2, z_1, z_2 \geqslant 0$		$\lambda_1, \lambda_2, \mu_1, \mu_2 \geqslant 0$

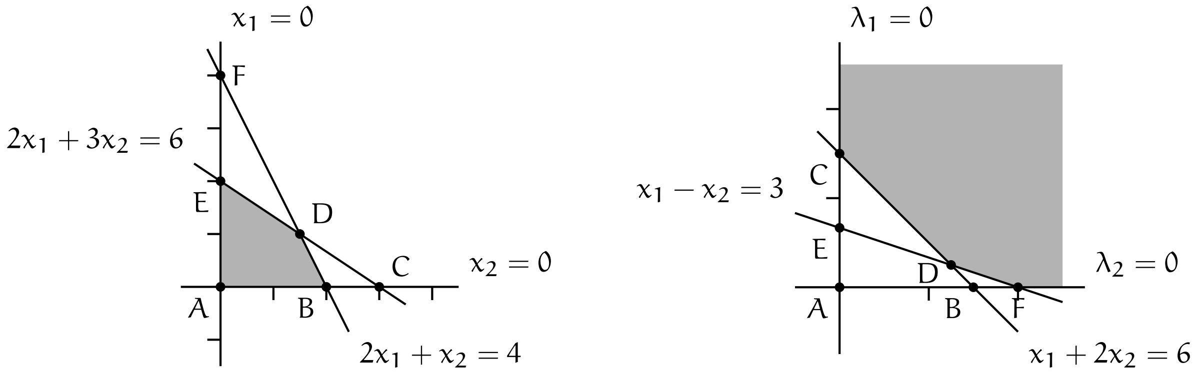

Figure 6.1: Geometric interpretation of primal and dual linear programs in Example 6.2

To see that these LPs are indeed dual to each other, observe that the primal has the form (1.2), and the dual the form (6.1), with

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

As before, we can compute all basic solutions of the primal by setting any set of $n - m = 2$ variables to zero in turn, and solving for the values of the remaining $m = 2$ variables. Given a particular basic solution of the primal, the corresponding dual solution can be found using the complementary slackness conditions $\lambda_1 z_1 = 0 = \lambda_2 z_2$ and $\mu_1 x_1 = 0 = \mu_2 x_2$. These conditions identify, for each non-zero variable of the primal, a dual variable whose value has to be equal to zero. By solving for the remaining variables, we obtain a solution for the dual, which is in fact a basic solution. Repeating this procedure for every basic solution of the primal, we obtain the following pairs of basic solutions of the primal and dual:

	x_1	x_2	z_1	z_2	$f(x)$	λ_1	λ_2	μ_1	μ_2	$g(\lambda)$
A	0	0	4	6	0	0	0	-3	-2	0
B	2	0	0	2	6	$\frac{3}{2}$	0	0	$-\frac{1}{2}$	6
C	3	0	-2	0	9	0	$\frac{3}{2}$	0	$\frac{5}{2}$	9
D	$\frac{3}{2}$	1	0	0	$\frac{13}{2}$	$\frac{5}{4}$	$\frac{1}{4}$	0	0	$\frac{13}{2}$
E	0	2	2	0	4	0	$\frac{2}{3}$	$-\frac{5}{3}$	0	4
F	0	4	0	-6	8	2	0	1	0	8

Labels A through F refer to Figure 6.2, which illustrates the feasible regions of the primal and the dual. Observe that there is only one pair such that both the primal and the dual solution are feasible, the one labeled D, and that these solutions are optimal.

6.2 Necessary and Sufficient Conditions for Optimality

In the above example, primal feasibility, dual feasibility, and complementary slackness together imply optimality. It turns out that this is true in general, and the condition is in fact both necessary and sufficient for optimality.

THEOREM 6.3. *Let x and λ be feasible solutions for the primal* (1.2) *and the dual* (6.1)*, respectively. Then x and λ are optimal if and only if they satisfy complementary slackness, i.e., if*

$$(c^T - \lambda^T A)x = 0 \quad \textit{and} \quad \lambda^T(Ax - b) = 0.$$

Proof. If x and λ are optimal, then

$$\begin{aligned} c^T x &= \lambda^T b \\ &= \inf_{x' \in X} \left(c^T x' - \lambda^T (Ax' - b)\right) \\ &\leqslant c^T x - \lambda^T (Ax - b) \\ &\leqslant c^T x. \end{aligned}$$

Since the first and last term are the same, the two inequalities must hold with equality. Therefore, $\lambda^T b = c^T x - \lambda^T(Ax - b) = (c^T - \lambda^T A)x + \lambda^T b$, and thus $(c^T - \lambda^T A)x = 0$. Furthermore, $c^T x - \lambda^T(Ax - b) = c^T x$, and thus $\lambda^T(Ax - b) = 0$.

If on the other hand $(c^T - \lambda^T A)x = 0$ and $\lambda^T(Ax - b) = 0$, then

$$c^T x = c^T x - \lambda^T(Ax - b) = (c^T - \lambda^T A)x + \lambda^T b = \lambda^T b,$$

and by weak duality x and λ must be optimal. □

While the result has been formulated here for the primal LP in general form and the corresponding dual, it is true, with the appropriate complementary slackness conditions, for any pair of a primal and dual LP.

7 The Simplex Method

Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. Further let B be a *basis*, i.e., a set $B \subseteq \{1, \dots, n\}$ with $|B| = m$, corresponding to a choice of m non-zero variables. Let $x \in \mathbb{R}^n$ such that $Ax = b$. Then we have

$$A_B x_B + A_N x_N = b,$$

where $A_B \in \mathbb{R}^{m \times m}$ and $A_N \in \mathbb{R}^{m \times (n-m)}$ respectively consist of the columns of A indexed by B and those not indexed by B, and x_B and x_N respectively consist of the rows of x indexed by B and those not indexed by B. Moreover, if x is a basic solution, then there is a basis B such that $x_N = 0$ and $A_B x_B = b$, and if x is a basic feasible solution, there is a basis B such that $x_N = 0$, $A_B x_B = b$, and $x_B \geqslant 0$.

7.1 The Simplex Tableau

For every x with $Ax = b$ and every basis B, we have that $x_B = A_B^{-1}(b - A_N x_N)$, and thus

$$\begin{aligned} f(x) = c^T x &= c_B^T x_B + c_N^T x_N \\ &= c_B^T A_B^{-1}(b - A_N x_N) + c_N^T x_N \\ &= c_B^T A_B^{-1} b + (c_N^T - c_B^T A_B^{-1} A_N) x_N \end{aligned}$$

Suppose that we want to maximize $c^T x$ and find that

$$c_N^T - c_B^T A_B^{-1} A_N \leqslant 0 \quad \text{and} \quad A_B^{-1} b \geqslant 0. \tag{7.1}$$

Then, for any feasible $x \in \mathbb{R}^n$, it holds that $x_N \geqslant 0$ and therefore $f(x) \leqslant c_B^T A_B^{-1} b$. The basic solution x^* with $x_B^* = A_B^{-1} b$ and $x_N^* = 0$, on the other hand, is feasible and satisfies $f(x^*) = c_B^T A_B^{-1} b$. It must therefore be optimal.

If alternatively $(c_N^T - c_B^T A_B^{-1} A_N)_i > 0$ for some i, then we can increase the value of the objective by increasing $(x_N)_i$. Either this can be done indefinitely, which means that the maximum is unbounded, or the constraints force some of the variables in the basis to become smaller and we have to stop when the first one reaches zero. In that case we have found a new BFS with a larger value and can repeat the process.

Assuming that the LP is feasible and has a bounded optimal solution, there exists a basis B^* for which (7.1) is satisfied. The basic idea behind the simplex method is to start from an initial BFS and then move from basis to basis until B^* is found. The information required for this procedure can conveniently be represented by the so-called *simplex tableau*. For a given basis B, it takes the following form:[1]

[1]The columns of the tableau have been permuted such that those corresponding to the basis appear on the left. This has been done just for convenience: in practice we will always be able to identify the columns corresponding to the basis by the embedded identity matrix.

	$\overbrace{\quad}^{m}$ B	$\overbrace{\quad}^{n-m}$ N	$\overbrace{\quad}^{1}$
m	$A_B^{-1}A_B = I$	$A_B^{-1}A_N$	$A_B^{-1}b$
1	$c_B^T - c_B^T A_B^{-1} A_B = 0$	$c_N^T - c_B^T A_B^{-1} A_N$	$-c_B^T A_B^{-1} b$

The first m rows consist of the matrix A and the column vector b, multiplied by the inverse of A_B. It is worth pointing out that for any basis B, the LP with constraints $A_B^{-1}Ax = A_B^{-1}b$ is equivalent to the one with constraints $Ax = b$. The first n columns of the last row are equal to $c^T - \lambda^T A$ for $\lambda^T = c_B^T A_B^{-1}$. The vector λ can be interpreted as a solution, not necessarily feasible, to the dual problem. In the last column of the last row we finally have the value $-f(x)$, where x is the BFS given by $x_B = A_B^{-1}b$ and $x_N = 0$.

We will see later that the simplex method always maintains feasibility of this solution x. As a consequence it also maintains complementary slackness for x and $\lambda^T = c_B^T A_B^{-1}$: since we work with an LP in standard form, $\lambda^T(Ax - b) = 0$ follows automatically from the feasibility condition, $Ax = b$; the condition $(c^T - \lambda^T A)x = 0$ holds because $x_N = 0$ and $c_B^T - \lambda^T A_B = c_B^T - c_B^T A_B^{-1} A_B = 0$. What it then means for (7.1) to become satisfied is that $c^T - \lambda^T A \leqslant 0$, i.e., that λ is a feasible solution for the dual. Optimality of x is thus actually a consequence of Theorem 6.3.

7.2 Using The Tableau

Consider a tableau of the following form, where the basis can be identified by the identity matrix embedded in (a_{ij}):

(a_{ij})	a_{i0}
a_{0j}	a_{00}

The simplex method then takes the following steps:

1. Find an initial BFS with basis B.
2. Check whether $a_{0j} \leqslant 0$ for every j. If yes, the current solution is optimal, so stop.
3. Choose j such that $a_{0j} > 0$, and choose $i \in \{i' : a_{i'j} > 0\}$ to minimize a_{i0}/a_{ij}. If $a_{ij} \leqslant 0$ for all i, then the problem is unbounded, so stop. If multiple rows minimize a_{i0}/a_{ij}, the problem is degenerate.
4. Update the tableau by multiplying row i by $1/a_{ij}$ and adding a $-(a_{kj}/a_{ij})$ multiple of row i to each row $k \neq i$. Then return to Step 2.

We will now describe the different steps of the simplex method in more detail and illustrate them using the LP of Example 1.1.

Finding an initial BFS

Finding an initial BFS is very easy when the constraints are of the form $Ax \leqslant b$ for $b \geqslant 0$. We can then add a vector z of slack variables and write the constraints as $Ax + z = b$, $z \geqslant 0$ and get a BFS by setting $x = 0$ and $z = b$. This can alternatively be thought of as extending x to (x, z) and setting $(x_B, x_N) = (z, x) = (b, 0)$. We then have $A_B^{-1} = I$ and $c_B = 0$, and the entries in the tableau become A_N and c_N^T for the variables x_1 and x_2 that are not in the basis, and b and 0 in the last column. For the LP of Example 1.1 we obtain the following tableau, where rows and columns have been labeled with the names of the corresponding variables:

	x_1	x_2	z_1	z_2	a_{i0}
z_1	1	2	1	0	6
z_2	1	-1	0	1	3
a_{0j}	1	1	0	0	0

If the constraints do not have this convenient form, finding an initial BFS requires more work. We will discuss this case in the next lecture.

Choosing a pivot column

If $a_{0j} \leqslant 0$ for all $j \geqslant 1$, the current solution is optimal. Otherwise we can choose a column j such that $a_{0j} > 0$ as the pivot column and let the corresponding variable enter the basis. If multiple candidate columns exist, choosing any one of them will work, but we could for example break ties toward the one that maximizes a_{0j} or the one with the smallest index. The candidate variables in our example are x_1 and x_2, so let us choose x_1.

Choosing the pivot row

If $a_{ij} \leqslant 0$ for all i, then the problem is unbounded and the objective can be increased by an arbitrary amount. Otherwise we choose a row $i \in \{i' : a_{i'j} > 0\}$ that minimizes a_{i0}/a_{ij}. This row is called the pivot row, and a_{ij} is called the pivot. If multiple rows minimize a_{i0}/a_{ij}, the problem has a degenerate BFS. In our example there is a unique choice, corresponding to variable z_2.

Pivoting

The purpose of the pivoting step is to get the tableau into the appropriate form for the new BFS. For this, we multiply row i by $1/a_{ij}$ and add a $-(a_{kj}/a_{ij})$ multiple of row i

to each row $k \neq i$, including the last one. Our choice of the pivot row as a row that minimizes a_{i0}/a_{ij} turns out to be crucial, as it guarantees that the solution remains feasible after pivoting. In our example, we need to subtract the second row from both the first and the last row, after which the tableau looks as follows:

	x_1	x_2	z_1	z_2	a_{i0}
z_1	0	3	1	−1	3
x_1	1	−1	0	1	3
a_{0j}	0	2	0	−1	−3

Note that the second row now corresponds to variable x_1, which has replaced z_2 in the basis.

We are now ready to choose a new pivot column. In our example, one further iteration yields the following tableau:

	x_1	x_2	z_1	z_2	a_{i0}
x_2	0	1	$\frac{1}{3}$	$-\frac{1}{3}$	1
x_1	1	0	$\frac{1}{3}$	$\frac{2}{3}$	4
a_{0j}	0	0	$-\frac{2}{3}$	$-\frac{1}{3}$	−5

This corresponds to the BFS where $x_1 = 4$, $x_2 = 1$, and $z_1 = z_2 = 0$, with an objective of -5. All entries in the last row are non-positive, so this solution is optimal.

8 The Two-Phase Simplex Method

The LP we solved in the previous lecture allowed us to find an initial BFS very easily. In cases where such an obvious candidate for an initial BFS does not exist, we can solve a different LP to find an initial BFS. We will refer to this as phase I. In phase II we then proceed as in the previous lecture.

Consider the LP to

$$\begin{aligned} \text{minimize} \quad & 6x_1 + 3x_2 \\ \text{subject to} \quad & x_1 + x_2 \geqslant 1 \\ & 2x_1 - x_2 \geqslant 1 \\ & 3x_2 \leqslant 2 \\ & x_1, x_2 \geqslant 0. \end{aligned}$$

We change from minimization to maximization and introduce slack variables to obtain the following equivalent problem:

$$\begin{aligned} \text{maximize} \quad & -6x_1 - 3x_2 \\ \text{subject to} \quad & x_1 + x_2 - z_1 = 1 \\ & 2x_1 - x_2 - z_2 = 1 \\ & 3x_2 + z_3 = 2 \\ & x_1, x_2, z_1, z_2, z_3 \geqslant 0. \end{aligned}$$

Unfortunately, the basic solution with $x_1 = x_2 = 0$, $z_1 = z_2 = -1$, and $z_3 = 2$ is not feasible. We can, however, add an *artificial variable* to the left-hand side of each constraint where the slack variable and the right-hand side have opposite signs, and then minimize the sum of the artificial variables starting from the obvious BFS where the artificial variables are non-zero instead of the corresponding slack variables. In the example, we

$$\begin{aligned} \text{minimize} \quad & y_1 + y_2 \\ \text{subject to} \quad & x_1 + x_2 - z_1 + y_1 = 1 \\ & 2x_1 - x_2 - z_2 + y_2 = 1 \\ & 3x_2 + z_3 = 2 \\ & x_1, x_2, z_1, z_2, z_3, y_1, y_2 \geqslant 0, \end{aligned}$$

and the goal of phase I is to solve this LP starting from the BFS where $x_1 = x_2 = z_1 = z_2 = 0$, $y_1 = y_2 = 1$, and $z_3 = 2$. If the original problem is feasible, we will be able to find a BFS where $y_1 = y_2 = 0$. This automatically gives us an initial BFS for the original problem.

In summary, the two-phase simplex method proceeds as follows:

1. Bring the constraints into equality form. For each constraint in which the slack variable and the right-hand side have opposite signs, or in which there is no slack variable, add a new artificial variable that has the same sign as the right-hand side.
2. Phase I: minimize the sum of the artificial variables, starting from the BFS where the absolute value of the artificial variable for each constraint, or of the slack variable in case there is no artificial variable, is equal to that of the right-hand side.
3. If some artificial variable has a positive value in the optimal solution, the original problem is infeasible; stop.
4. Phase II: solve the original problem, starting from the BFS found in phase I.

While the original objective is not needed for phase I, it is useful to carry it along as an extra row in the tableau, because it will then be in the appropriate form at the beginning of phase II. In the example, phase I therefore starts with the following tableau:

	x_1	x_2	z_1	z_2	z_3	y_1	y_2	
y_1	1	1	−1	0	0	1	0	1
y_2	2	−1	0	−1	0	0	1	1
z_3	0	3	0	0	1	0	0	2
II	−6	−3	0	0	0	0	0	0
I	3	0	−1	−1	0	0	0	2

Note that the objective for phase I is written in terms of the *non-basic* variables. This can be achieved by first writing it in terms of y_1 and y_2, such that we have -1 in the columns for y_1 and y_2 and 0 in all other columns because we are *maximizing* $-y_1-y_2$, and then adding the first and second row to make the entries for all variables in the basis equal to zero.

Phase I now proceeds by pivoting on a_{21} to get

	x_1	x_2	z_1	z_2	z_3	y_1	y_2	
	0	$\frac{3}{2}$	−1	$\frac{1}{2}$	0	1	$-\frac{1}{2}$	$\frac{1}{2}$
	1	$-\frac{1}{2}$	0	$-\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$
	0	3	0	0	1	0	0	2
II	0	−6	0	−3	0	0	3	3
I	0	$\frac{3}{2}$	−1	$\frac{1}{2}$	0	0	$-\frac{3}{2}$	$\frac{1}{2}$

and on a_{14} to get

	x_1	x_2	z_1	z_2	z_3	y_1	y_2	
	0	3	−2	1	0	2	−1	1
	1	1	−1	0	0	1	0	1
	0	3	0	0	1	0	0	2
II	0	3	−6	0	0	6	0	6
I	0	0	0	0	0	−1	−1	0

Note that we could have chosen a_{12} as the pivot element in the second step, and would have obtained the same result.

This ends phase I as $y_1 = y_2 = 0$, and we have found a BFS for the original problem with $x_1 = z_2 = 1$, $z_3 = 2$, and $x_2 = z_1 = 0$. After dropping the columns for y_1 and y_2 and the row corresponding to the objective for phase I, the tableau is in the right form for phase II:

x_1	x_2	z_1	z_2	z_3	
0	3	−2	1	0	1
1	1	−1	0	0	1
0	3	0	0	1	2
0	3	−6	0	0	6

By pivoting on a_{12} we obtain the following tableau, corresponding to an optimal solution of the original problem with $x_1 = 2/3$, $x_2 = 1/3$, and value -5:

x_1	x_2	z_1	z_2	z_3	
0	1	$-\frac{2}{3}$	$\frac{1}{3}$	0	$\frac{1}{3}$
1	0	$-\frac{1}{3}$	$-\frac{1}{3}$	0	$\frac{2}{3}$
0	0	2	−1	1	1
0	0	−4	−1	0	5

It is worth noting that the problem we have just solved is the dual of the LP in Example 1.1, which we solved in the previous lecture, augmented by the constraint $3x_2 \leqslant 2$. Ignoring the column and row corresponding to z_3, the slack variable for this new constraint, the final tableau is essentially the negative of the transpose of the final tableau we obtained in the previous lecture. This makes sense because the additional constraint is not tight in the optimal solution, as we can see from the fact that $z_3 \neq 0$.

9 Non-Cooperative Games

The theory of non-cooperative games studies situations in which multiple self-interested entities, or *players*, simultaneously and independently optimize different objectives and outcomes must therefore be self-enforcing.

9.1 Games and Solutions

The central object of study in non-cooperative game theory are *normal-form games*. We restrict our attention to two-player games, but note that most concepts extend in a straightforward way to games with more than two players. A two-player game with m actions for player 1 and n actions for player 2 can be represented by a pair of matrices $P, Q \in \mathbb{R}^{m \times n}$, where p_{ij} and q_{ij} are the payoffs of players 1 and 2 when player 1 plays action i and player 2 plays action j. Two-player games are therefore sometimes referred to as bimatrix games, and players 1 and 2 as the row and column player, respectively.

We will assume that players can choose their actions randomly and denote the set of possible *strategies* of the two players by X and Y, respectively, i.e., $X = \{x \in \mathbb{R}^m_{\geqslant 0} : \sum_{i=1}^m x_i = 1\}$ and $Y = \{y \in \mathbb{R}^n_{\geqslant 0} : \sum_{i=1}^n y_i = 1\}$. A *pure strategy* is a strategy that chooses some action with probability one, and we make no distinction between pure strategies and the corresponding actions. A profile $(x, y) \in X \times Y$ of strategies induces a lottery over outcomes, and we write $p(x, y) = x^T P y$ for the expected payoff of the row player in this lottery.

Consider for example the well-known prisoner's dilemma, involving two suspects accused of a crime who are being interrogated separately. If both remain silent, they walk free after spending a few weeks in pretrial detention. If one of them testifies against the other and the other remains silent, the former is released immediately while the latter is sentenced to ten years in jail. If both suspects testify, each of them receives a five-year sentence. A representation of this situation as a two-player normal-form game is shown in Figure 9.1.

It is easy to see what the players in this game should do, because action T yields a strictly larger payoff than action S for *every* action of the respective other player. More generally, for two strategies $x, x' \in X$ of the row player, x is said to *(strictly) dominate* x' if for every strategy $y \in Y$ of the column player, $p(x, y) > p(x', y)$. Dominance for the column player is defined analogously. Strategy profile (T, T) in the prisoner's dilemma is what is called a *dominant strategy equilibrium*, a profile of strategies that dominate every other strategy of the respective player. The source of the dilemma is that outcome resulting from (T, T) is strictly worse for both players than the outcome resulting from (S, S). More generally, an outcome that is weakly preferred to another outcome by all players, and strictly preferred by at least one player is said to *Pareto*

	S	T
S	$(2,2)$	$(0,3)$
T	$(3,0)$	$(1,1)$

Figure 9.1: Representation of the prisoner's dilemma as a normal-form game. The matrices P and Q are displayed as a single matrix with entries (p_{ij}, q_{ij}), and players 1 and 2 respectively choose a row and a column of this matrix. Action S corresponds to remaining silent, action T to testifying.

	C	D
C	$(2,2)$	$(1,3)$
D	$(3,1)$	$(0,0)$

Figure 9.2: The game of chicken, where players can chicken out or dare

dominate that outcome. An outcome that is Pareto dominated is clearly undesirable.

In the absence of dominant strategies, it is less obvious how players should proceed. Consider for example the game of chicken illustrated in Figure 9.2. This game has its origins in a situation where two cars drive towards each other on a collision course. Unless one of the drivers yields, both may die in a crash. If one of them yields while the other goes straight, however, the former will be called a "chicken," or coward. It is easily verified that this game does not have any dominant strategies.

The most cautious choice in a situation like this would be to ignore that the other player is self-interested and choose a strategy that maximizes the payoff in the worst case, taken over all of the other player's strategies. A strategy of this type is known as a *maximin strategy*, and the payoff thus obtained as the player's *security level*. It is easy to see that it suffices to maximize the minimum payoff over all *pure* strategies of the other player, i.e., to choose x such that $\min_{j\in\{1,\dots,n\}} \sum_{i=1}^{m} x_i p_{ij}$ is as large as possible. Maximization of this minimum can be achieved by maximizing a lower bound that holds for all $j = 1, \dots, n$, so a maximin strategy and the security level for the row player can be found as a solution of the following linear program with variables $v \in \mathbb{R}$ and $x \in \mathbb{R}^m$:

$$\begin{aligned} \text{maximize} \quad & v \\ \text{subject to} \quad & \sum_{i=1}^{m} x_i p_{ij} \geqslant v \quad \text{for } j = 1, \dots, n \\ & \sum_{i=1}^{m} x_i = 1 \\ & x \geqslant 0. \end{aligned} \tag{9.1}$$

The unique maximin strategy in the game of chicken is to yield, for a security level of 1. This also illustrates that a maximin strategy need not be optimal: assuming that the row player yields, the optimal action for the column player is in fact to go straight. Formally, strategy $x \in X$ of the row player is a *best response* to strategy $y \in Y$ of the column player if for all $x' \in X$, $p(x, y) \geqslant p(x', y)$. The concept of a best response for the column player is defined analogously. A pair of strategies $(x, y) \in X \times Y$ such that x is a best response to y and y is a best response to x is called an *equilibrium*. Equilibria are also known as Nash equilibria, because their universal existence was shown by John Nash.

THEOREM 9.1 (Nash, 1951). *Every bimatrix game has an equilibrium.*

It is easily verified that both (C, D) and (D, C) are equilibria of the game of chicken, and there is one more equilibrium, in which both players randomize uniformly between their two actions. The proof of Theorem 9.1 is beyond the scope of this course, but we show the result for the special case when the players have diametrically opposed interests.

9.2 The Minimax Theorem

A two-player game is called *zero-sum game* if $q_{ij} = -p_{ij}$ for all $i = 1, \dots, m$ and $j = 1, \dots, n$. A game of this type is sometimes called a matrix game, because it can be represented just by the matrix P containing the payoffs of the row player. Assuming invariance of utilities under positive affine transformations, results for zero-sum games in fact apply to the larger class of *constant-sum* games, in which the payoffs of the two players always sum up to the same constant. For games with more than two players, these properties are far less interesting, as one can always add an extra player who "absorbs" the payoffs of the others.

It turns out that in zero-sum games, maximin strategies are optimal.

THEOREM 9.2 (von Neumann, 1928). *Let* $P \in \mathbb{R}^{m \times n}$, $X = \{x \in \mathbb{R}^m_{\geqslant 0} : \sum_{i=1}^m x_i = 1\}$, $Y = \{y \in \mathbb{R}^n_{\geqslant 0} : \sum_{i=1}^n y_i = 1\}$. *Then,*

$$\max_{x \in X} \min_{y \in Y} p(x, y) = \min_{y \in Y} \max_{x \in X} p(x, y).$$

Proof. Again consider the linear program (9.1), and recall that the optimum value of this linear program is equal to $\max_{x \in X} \min_{y \in Y} p(x, y)$. By adding a slack variable $z \in \mathbb{R}^n$ with $z \geqslant 0$ we obtain the Lagrangian

$$\begin{aligned} L(v, x, z, w, y) &= v + \sum_{j=1}^n y_j \Big(\sum_{i=1}^m x_i p_{ij} - z_j - v \Big) - w \Big(\sum_{i=1}^m x_i - 1 \Big) \\ &= \Big(1 - \sum_{j=1}^n y_j \Big) v + \sum_{i=1}^m \Big(\sum_{j=1}^n p_{ij} y_j - w \Big) x_i - \sum_{j=1}^n y_j z_j + w, \end{aligned}$$

where $w \in \mathbb{R}$ and $y \in \mathbb{R}^n$. The Lagrangian has a finite maximum for $v \in \mathbb{R}$ and $x \in \mathbb{R}^m$ with $x \geqslant 0$ if and only if $\sum_{j=1}^n y_j = 1$, $\sum_{j=1}^n p_{ij} y_j \leqslant w$ for $i = 1, \ldots, m$, and $y \geqslant 0$. In the dual of (9.1) we therefore want to

$$\begin{aligned} &\text{minimize} && w \\ &\text{subject to} && \sum_{j=1}^n p_{ij} y_j \leqslant w \quad \text{for } i = 1, \ldots, m \\ &&& \sum_{j=1}^n y_j = 1 \\ &&& y \geqslant 0. \end{aligned}$$

It is easy to see that the optimum value of the dual is $\min_{y \in Y} \max_{x \in X} p(x, y)$, and the theorem follows from strong duality. □

The number $\max_{x \in X} \min_{y \in Y} p(x, y) = \min_{y \in Y} \max_{x \in X} p(x, y)$ is also called the *value* of the matrix game with payoff matrix P.

It is now easy to show that every matrix game has an equilibrium, and that the above result in fact characterizes the set of equilibria of such games.

THEOREM 9.3. *A pair of strategies $(x, y) \in X \times Y$ is an equilibrium of the matrix game with payoff matrix P if and only if*

$$\min_{y' \in Y} p(x, y') = \max_{x' \in X} \min_{y' \in Y} p(x', y') \quad \textit{and}$$
$$\max_{x' \in X} p(x', y) = \min_{y' \in Y} \max_{x' \in X} p(x', y').$$

10 The Minimum-Cost Flow Problem

The remaining lectures will be concerned with optimization problems on networks, in particular with flow problems.

10.1 Networks

A directed *graph*, or *network*, $G = (V, E)$ consists of a set V of *vertices* and a set $E \subseteq V \times V$ of *edges*. When the relation E is symmetric, G is called an undirected graph, and we can write edges as unordered pairs $\{u, v\} \in E$ for $u, v \in V$. The *degree* of vertex $u \in V$ in graph G is the number $|\{v \in V : (u, v) \in E \text{ or } (v, u) \in E\}|$ of other vertices connected to it by an edge. A *walk* from $u \in V$ to $w \in V$ is a sequence of vertices $v_1, \ldots, v_k \in V$ such that $v_1 = u$, $v_k = w$, and $(v_i, v_{i+1}) \in E$ for $i = 1, \ldots, k-1$. In a directed graph, we can also consider an undirected walk where $(v_i, v_{i+1}) \in E$ or $(v_{i+1}, v_i) \in E$ for $i = 1, \ldots, k-1$. A walk is a *path* if $v_1, \ldots, v_k$ are pairwise distinct, and a *cycle* if $v_1, \ldots, v_{k-1}$ are pairwise distinct and $v_k = v_1$. A graph that does not contain any cycles is called *acyclic*. A graph is called *connected* if for every pair of vertices $u, v \in V$ there is an undirected path from u to v. A *tree* is a graph that is connected and acyclic. A graph $G' = (V', E')$ is a subgraph of graph $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. In the special case where G' is a tree and $V' = V$, it is called a *spanning tree* of G.

10.2 Minimum-Cost Flows

Consider a network $G = (V, E)$ with $|V| = n$, and let $b \in \mathbb{R}^n$. Here, b_i denotes the amount of flow that enters or leaves the network at vertex $i \in V$. If $b_i > 0$, we say that i is a *source* supplying b_i units of flow. If $b_i < 0$, we say that i is a *sink* with a demand of $|b_i|$ units of flow. Further let $C, \underline{M}, \overline{M} \in \mathbb{R}^{n \times n}$, where c_{ij} denotes the cost associated with one unit of flow on edge $(i, j) \in E$, and $\underline{m}_{ij}$ and $\overline{m}_{ij}$ respectively denote lower and upper bounds on the flow across this edge. The minimum-cost flow problem then asks for flows x_{ij} that conserve the flow at each vertex, respect the upper and lower bounds, and minimize the overall cost. Formally, $x \in \mathbb{R}^{n \times n}$ is a *minimum-cost flow* of G if it is an optimal solution of the following optimization problem:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{(i,j) \in E} c_{ij} x_{ij} \\
\text{subject to} \quad & b_i + \sum_{j:(j,i) \in E} x_{ji} = \sum_{j:(i,j) \in E} x_{ij} && \text{for all } i \in V, \\
& \underline{m}_{ij} \leqslant x_{ij} \leqslant \overline{m}_{ij} && \text{for all } (i, j) \in E.
\end{aligned}
$$

The minimum-cost flow problem is a linear programming problem, with constraints of the form $Ax = b$ where

$$a_{ik} = \begin{cases} 1 & \text{kth edge starts at vertex } i, \\ -1 & \text{kth edge ends at vertex } i, \\ 0 & \text{otherwise.} \end{cases}$$

Note that $\sum_{i \in V} b_i = 0$ is required for feasibility, and that a problem satisfying this condition can be transformed into an equivalent problem where $b_i = 0$ for all i by introducing an additional vertex, and new edges from each sink to the new vertex and from the new vertex to each of the sources with upper and lower bounds equal to the flow that should enter the sources and leave the sinks. The latter problem is known as a *circulation problem*, because flow does not enter or leave the network but merely circulates. We can further assume without loss of generality that the network G is connected. Otherwise the problem can be decomposed into several smaller problems that can be solved independently.

An important special case is that of *uncapacitated flow problems*, where $\underline{m}_{ij} = 0$ and $\overline{m}_{ij} = \infty$ for all $(i,j) \in E$. Clearly, an uncapacitated flow problem is either unbounded, or has an equivalent problem with finite capacities.

10.3 Sufficient Conditions for Optimality

The Lagrangian of the minimum-cost circulation problem is

$$L(x, \lambda) = \sum_{(i,j) \in E} c_{ij} x_{ij} - \sum_{i \in V} \lambda_i \left(\sum_{j:(i,j) \in E} x_{ij} - \sum_{j:(j,i) \in E} x_{ji} \right) = \sum_{(i,j) \in E} (c_{ij} - \lambda_i + \lambda_j) x_{ij}.$$

If the Lagrangian is minimized subject to the regional constraints $\underline{m}_{ij} \leqslant x_{ij} \leqslant \overline{m}_{ij}$ for $(i,j) \in E$, Theorem 2.1 yields a set of conditions that are sufficient for optimality. It will be instructive to prove this result directly.

THEOREM 10.1. *Consider a feasible flow $x \in \mathbb{R}^{n \times n}$ for a circulation problem, and let $\lambda \in \mathbb{R}^n$ such that*

$$\begin{aligned} c_{ij} - \lambda_i + \lambda_j > 0 &\quad \textit{implies} \quad x_{ij} = \underline{m}_{ij}, \\ c_{ij} - \lambda_i + \lambda_j < 0 &\quad \textit{implies} \quad x_{ij} = \overline{m}_{ij}, \textit{ and} \\ \underline{m}_{ij} < x_{ij} < \overline{m}_{ij} &\quad \textit{implies} \quad c_{ij} - \lambda_i + \lambda_j = 0. \end{aligned}$$

Then x is optimal.

Proof. For $(i, j) \in E$, let $\bar{c}_{ij} = c_{ij} - \lambda_i + \lambda_j$. Then, for every feasible flow x',

$$\begin{aligned}
\sum_{(i,j)\in E} c_{ij}x'_{ij} &= \sum_{(i,j)\in E} c_{ij}x'_{ij} - \sum_{i\in V} \lambda_i \left(\sum_{j:(i,j)\in E} x'_{ij} - \sum_{j:(j,i)\in E} x'_{ji} \right) \\
&= \sum_{(i,j)\in E} \bar{c}_{ij}x'_{ij} \\
&\geqslant \sum_{\substack{(i,j)\in E \\ \bar{c}_{ij}<0}} \bar{c}_{ij}\overline{m}_{ij} + \sum_{\substack{(i,j)\in E \\ \bar{c}_{ij}>0}} \bar{c}_{ij}\underline{m}_{ij} \\
&= \sum_{(i,j)\in E} \bar{c}_{ij}x_{ij} = \sum_{(i,j)\in E} c_{ij}x_{ij}
\end{aligned}$$

□

The Lagrange multiplier λ_i is also referred to as a *node number*, or as a *potential* associated with vertex $i \in V$. Since only the difference between pairs of Lagrange multipliers appears in the optimality conditions, we can set $\lambda_1 = 0$ without loss of generality.

10.4 The Transportation Problem

An important special case of the minimum-cost flow problem is the *transportation problem*, where we are given a set of *suppliers* $i = 1, \dots, n$ producing s_i units of a good and a set of *consumers* $j = 1, \dots, m$ with demands d_j such that $\sum_{i=1}^{n} s_i = \sum_{j=1}^{m} d_j$. The cost of transporting one unit of the good from supplier i to consumer j is c_{ij}, and the goal is to match supply and demand while minimizing overall transportation cost. This can be formulated as an uncapacitated minimum-cost flow problem on a *bipartite network*, i.e., a network $G = (S \uplus C, E)$ with $S = \{1, \dots, n\}$, $C = \{1, \dots, m\}$, and $E \subseteq S \times C$. As far as optimal solutions are concerned, edges not contained in E are equivalent to edges with a very large cost. We can thus restrict our attention to the case where $E = S \times C$, known as the *Hitchcock transportation problem*:

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

It turns out that the transportation problem already captures the full expressiveness of the minimum-cost flow problem.

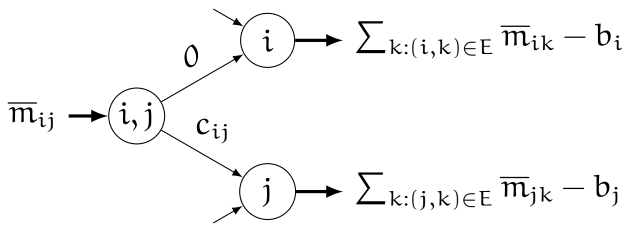

Figure 10.1: Representation of flow conservation constraints by an instance of the transportation problem

THEOREM 10.2. *Every minimum-cost flow problem with finite capacities or non-negative costs has an equivalent transportation problem.*

Proof. Consider a minimum-cost flow problem for a network (V, E) and assume without loss of generality that $\underline{m}_{ij} = 0$ for all $(i, j) \in E$. If this is not the case, we can instead consider the problem obtained by setting $\underline{m}_{ij}$ to zero, $\overline{m}_{ij}$ to $\overline{m}_{ij} - \underline{m}_{ij}$, and replacing b_i by $b_i - \underline{m}_{ij}$ and b_j by $b_j + \underline{m}_{ij}$. A solution with flow x_{ij} for the new problem then corresponds to a solution with flow $x_{ij} + \underline{m}_{ij}$ for the original problem. We can further assume that all capacities are finite: if some edge has infinite capacity but costs are non-negative then setting the capacity of this edge to a large enough number, for example $\sum_{i\in V} |b_i|$, does not affect the optimal solution of the problem.

We now construct an instance of the transportation problem as follows. For every vertex $i \in V$, we add a consumer with demand $\sum_k \overline{m}_{ik} - b_i$. For every edge $(i, j) \in E$, we add a supplier with supply $\overline{m}_{ij}$, an edge to vertex i with cost $c_{ij,j} = 0$, and an edge to vertex j with cost $c_{ij,j} = c_{ij}$. The situation is shown in Figure 10.1.

We now claim that there exists a direct correspondence between feasible flows of the two problems, and that these flows have the same costs. To see this, let the flows on edges (ij, i) and (ij, j) be $\overline{m}_{ij} - x_{ij}$ and x_{ij}, respectively. The total flow into vertex i then is $\sum_{k:(i,k)\in E}(\overline{m}_{ik} - x_{ik}) + \sum_{k:(k,i)\in E} x_{ki}$, which must be equal to $\sum_{k:(i,k)\in E} \overline{m}_{ik} - b_i$. This is the case if and only if $b_i + \sum_{k:(k,i)\in E} x_{ki} - \sum_{k:(i,k)\in E} x_{ik} = 0$, which is the flow conservation constraint for vertex i in the original problem. □

11 The Transportation Algorithm

The particular structure of basic feasible solutions in the case of the transportation problem gives rise to a special interpretation of the simplex method. This special form is sometimes called the transportation algorithm.

11.1 Optimality Conditions

The Lagrangian of the transportation problem can be written as

$$\begin{aligned} L(x,\lambda,\mu) &= \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} + \sum_{i=1}^{n}\lambda_i\left(s_i - \sum_{j=1}^{m} x_{ij}\right) - \sum_{j=1}^{m}\mu_j\left(d_j - \sum_{i=1}^{n} x_{ij}\right) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m}(c_{ij} - \lambda_i + \mu_j)x_{ij} + \sum_{i=1}^{n}\lambda_i s_i - \sum_{j=1}^{m}\mu_j d_j, \end{aligned}$$

where $\lambda \in \mathbb{R}^n$ and $\mu \in \mathbb{R}^m$ are Lagrange multipliers for the suppliers and consumers, respectively. Subject to $x_{ij} \geqslant 0$, the Lagrangian has a finite minimum if and only if

$$c_{ij} - \lambda_i + \mu_j \geqslant 0 \qquad \text{for } i = 1,\dots,n \text{ and } j = 1,\dots,m,$$

and at the optimum,

$$(c_{ij} - \lambda_i + \mu_j)x_{ij} = 0 \qquad \text{for } i = 1,\dots,n \text{ and } j = 1,\dots,m.$$

Together with feasibility of x, these dual feasibility and complementary slackness conditions are necessary and sufficient for optimality of x.

Note that the sign of the Lagrange multipliers can be chosen arbitrarily, and that this choice determines the form of the optimality conditions. The above choice is consistent with viewing demands as negative supplies.

11.2 The Simplex Method for the Transportation Problem

In solving instances of the transportation problem with the simplex method, a tableau of the following form will be useful:

	μ_1	$\cdots$	μ_m	
λ_1	x_{11} c_{11}	$\cdots$ $\cdots$	x_{1m} c_{1m}	s_1
$\vdots$	$\vdots$ $\vdots$	$\ddots$ $\ddots$	$\vdots$ $\vdots$	$\vdots$
λ_n	x_{n1} c_{n1}	$\cdots$ $\cdots$	x_{nm} c_{nm}	s_n
	d_1	$\cdots$	d_m	

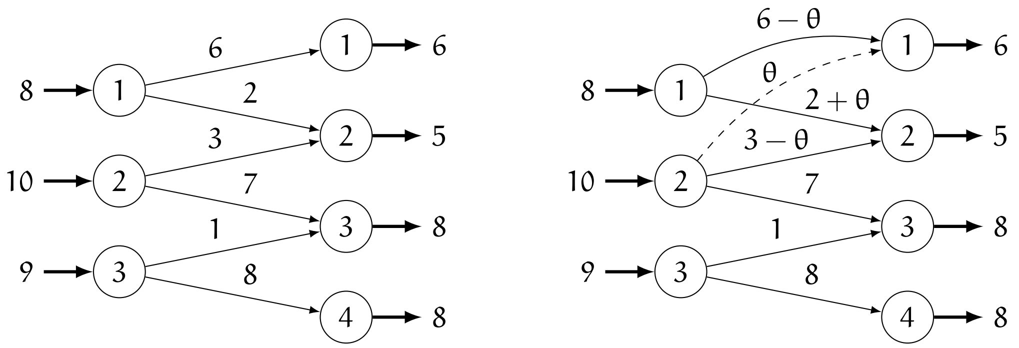

Figure 11.1: Initial basic feasible solution of an instance of the transportation problem (left) and a cycle along which the overall cost can be decreased (right)

Consider for example the Hitchcock transportation problem with three suppliers and four consumers given by the following tableau:

5	3	4	6	8
2	7	4	1	10
5	6	2	4	9
6	5	8	8	

Finding an initial BFS

An initial BFS can be found by iteratively considering pairs (i, j) of supplier i and consumer j, increasing x_{ij} until either the supply s_i or the demand d_j is satisfied, and moving to the next supplier in the former case or to the next consumer in the latter. Since $\sum_i s_i = \sum_j d_j$, this process is guaranteed to find a feasible solution. If at some intermediate step both supply and demand are satisfied at the same time, the resulting solution is degenerate. In general, degeneracies occur when a subset of the consumers can be satisfied exactly by a subset of the suppliers. In the example, we can start by setting $x_{11} = \min\{s_1, d_1\} = 6$, moving to consumer 2 and setting $x_{12} = 2$, moving to supplier 2 and setting $x_{22} = 3$, and so forth. The resulting flows are shown on the left of Figure 11.1.

Note that the initial BFS can be associated with a spanning tree (V, T) of the flow network where T is the set of edges visited by the above procedure. It then holds that $x_{ij} = 0$ when $(i, j) \notin T$, and complementary slackness dictates that $\lambda_i - \mu_j = c_{ij}$ when $(i, j) \in T$. By setting $\lambda_1 = 0$, we obtain a system of $n + m - 1$ linear equalities with $n + m - 1$ variables: each equality corresponds to an edge in T, each variable to a vertex in $(S \setminus \{1\}) \uplus C$. This system of equalities has a unique solution, allowing us to compute the values of the dual variables. We will see momentarily that every BFS can be associated with a spanning tree in this way. To verify dual feasibility, it will finally be convenient to write down $\lambda_i - \mu_j$ for $(i, j) \notin T$, and we do so in the upper right corner of the respective cells. For our example, we obtain the following tableau:

	-5	-3	0	-2	
0	6 [5]	2 [3]	0 [4]	2 [6]	8
4	9 [2]	3 [7]	7 [4]	6 [1]	10
2	7 [5]	5 [6]	1 [2]	8 [4]	9
	6	5	8	8	

Pivoting

If $c_{ij} \geqslant \lambda_i - \mu_j$ for all $(i, j) \notin T$, the current flow is optimal. Assume on the other hand that dual feasibility is violated for some edge $(i, j) \notin T$, and observe that this edge and the edges in T together form a unique cycle. In the absence of degeneracies the regional constraints for edges in T are not tight, so we can push flow around this cycle in order to increase x_{ij} and decrease the value of the Lagrangian. Due to the special structure of the network, this will alternately increase and decrease the flow for edges along the cycle until $x_{i'j'}$ becomes zero for some $(i', j') \in T$. We thus obtain a new BFS, and a new spanning tree in which (i', j') has been replaced by (i, j).

In our example dual feasibility is violated, for example, for $i = 2$ and $j = 1$. Edge $(2, 1)$ forms a unique cycle with the spanning tree T, and we would like to increase x_{21} by pushing flow along this cycle. In particular, increasing x_{21} by θ will increase x_{12} and decrease x_{11} and x_{22} by the same amount. The situation is shown on the right of Figure 11.1. If we increase x_{21} by the maximum amount of $\theta = 3$ and re-compute the values of the dual variables λ and μ, we obtain the following tableau:

	-5	-3	-7	-9
0	3 [5]	5 [3]	7 [4]	9 [6]
-3	3 [2]	0 [7]	7 [4]	6 [1]
-5	0 [5]	−2 [6]	1 [2]	8 [4]

Now, $c_{24} < \lambda_2 - \mu_4$, and we can increase x_{24} by 7 to obtain the following tableau, which satisfies $c_{ij} \geqslant \lambda_i - \mu_j$ for all $(i, j) \notin T$ and therefore yields an optimal solution:

	-5	-3	-2	-4
0	3 [5]	5 [3]	2 [4]	4 [6]
-3	3 [2]	0 [7]	−1 [4]	7 [1]
0	5 [5]	3 [6]	8 [2]	1 [4]

Let us summarize what we have done:

1. Find an initial BFS, and let T be the edges of the corresponding spanning tree.
2. Choose λ and μ such that $\lambda_1 = 0$ and $c_{ij} - \lambda_i + \mu_j = 0$ for all $(i, j) \in T$.
3. If $c_{ij} - \lambda_i + \mu_j \geqslant 0$ for all $(i, j) \in E$, the solution is optimal; stop.
4. Otherwise pick $(i, j) \in E$ such that $c_{ij} - \lambda_i + \mu_j < 0$, and push flow along the unique cycle in $(V, T \cup \{(i, j)\})$ until $x_{i'j'} = 0$ for some edge (i', j') in the cycle. Set T to $(T \setminus \{(i', j')\}) \cup \{(i, j)\}$ and go to Step 2.

12 The Maximum Flow Problem

Consider a flow network (V, E) with a single source 1, a single sink n, and finite capacities $\overline{m}_{ij} = C_{ij}$ for all $(i, j) \in E$. We will also assume for convenience that $\underline{m}_{ij} = 0$ for all $(i, j) \in E$. The *maximum flow problem* then asks for the maximum amount of flow that can be sent from vertex 1 to vertex n, i.e., the goal is to

$$\begin{aligned} &\text{maximize} \quad \delta \\ &\text{subject to} \sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = \begin{cases} \delta & \text{if } i = 1 \\ -\delta & \text{if } i = n \\ 0 & \text{otherwise} \end{cases} \\ &\qquad 0 \leqslant x_{ij} \leqslant C_{ij} \quad \text{for all } (i,j) \in E. \end{aligned} \tag{12.1}$$

This problem is in fact a special case of the minimum-cost flow problem. To see this, set $c_{ij} = 0$ for all $(i, j) \in E$, and add an edge $(n, 1)$ with infinite capacity and cost $c_{n1} = -1$. Since the new edge $(n, 1)$ has infinite capacity, any feasible flow of the original network is also feasible for the new network. Cost is clearly minimized by maximizing the flow across the edge $(n, 1)$, which by the flow conservation constraints for vertices 1 and n maximizes flow through the original network.

12.1 The Max-Flow Min-Cut Theorem

Consider a flow network $G = (V, E)$ with capacities C_{ij} for $(i, j) \in E$. A *cut* of G is a partition of V into two sets, and the capacity of a cut is defined as the sum of capacities of all edges across the partition. Formally, for $S \subseteq V$, the capacity of the cut $(S, V \setminus S)$ is given by

$$C(S) = \sum_{(i,j)\in S\times(V\setminus S)} C_{ij}.$$

Assume that x is a feasible flow vector that sends δ units of flow from vertex 1 to vertex n. It is easy to see that δ is bounded from above by the capacity of any cut S with $1 \in S$ and $n \in V \setminus S$. Indeed, for $X, Y \subseteq V$, let

$$f_x(X, Y) = \sum_{(i,j)\in E\cap(X\times Y)} x_{ij}.$$

Then, for any $S \subseteq V$ with $1 \in S$ and $n \in V \setminus S$,

$$\begin{aligned}
\delta &= \sum_{i \in S} \left(\sum_{j:(i,j) \in E} x_{ij} - \sum_{j:(j,i) \in E} x_{ji} \right) \\
&= f_x(S, V) - f_x(V, S) \\
&= f_x(S, S) + f_x(S, V \setminus S) - f_x(V \setminus S, S) - f_x(S, S) \\
&= f_x(S, V \setminus S) - f_x(V \setminus S, S) \\
&\leqslant f_x(S, V \setminus S) \leqslant C(S).
\end{aligned} \tag{12.2}$$

The following result states that this upper bound in fact tight, i.e., that there exists a flow of size equal to the minimum capacity of a cut that separates vertex 1 from vertex n.

THEOREM 12.1 (Max-flow min-cut theorem). *Let* δ *be the optimal solution of* (12.1) *for a network* (V, E) *with capacities* C_{ij} *for all* $(i, j) \in E$*. Then,*

$$\delta = \min\{ C(S) : S \subseteq V, 1 \in S, n \in V \setminus S \}.$$

Proof. It remains to be shown that there exists a cut that separates vertex 1 from vertex n and has capacity equal to δ. Consider a feasible flow vector x. A path $v_0, v_1, \dots, v_k$ is called an *augmenting path* for x if $x_{v_{i-1}v_i} < C_{v_{i-1}v_i}$ or $x_{v_i v_{i-1}} > 0$ for every $i = 1, \dots, k$. If there exists an augmenting path from vertex 1 to vertex n, then we can push flow along the path, by increasing the flow on every forward edge and decreasing the flow on every backward edge along the path by the same amount, such that all constraints remain satisfied and the amount of flow from 1 to n increases.

Now assume that x is optimal, and let

$$S = \{1\} \cup \{ i \in V : \text{there exists an augmenting path for } x \text{ from } 1 \text{ to } i \}.$$

By optimality of x, $n \in V \setminus S$. Moreover,

$$\delta = f_x(S, V \setminus S) - f_x(V \setminus S, S) = f_x(S, V \setminus S) = C(S).$$

The first equality holds by (12.2). The second equality holds because $x_{ij} = 0$ for every $(i, j) \in E \cap ((V \setminus S) \times S)$. The third equality holds because $x_{ij} = C_{ij}$ for every $(i, j) \in E \cap (S \times (V \setminus S))$. □

12.2 The Ford-Fulkerson Algorithm

The *Ford-Fulkerson algorithm* attempts to find a maximum flow by repeatedly pushing flow along an augmenting path, until such a path can no longer be found:

1. Start with a feasible flow vector x.
2. If there is no augmenting path for x from 1 to n, stop.

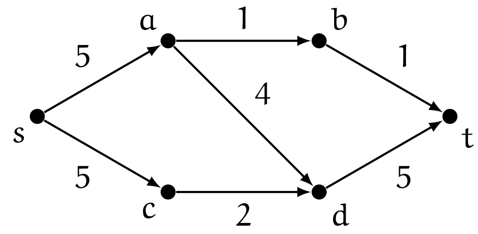

Figure 12.1: An instance of the maximum flow problem

3. Otherwise pick some augmenting path from 1 to n, and push a maximum amount of flow along this path without violating any constraints. Then go to Step 2.

Consider for example the flow network in Figure 12.1. Pushing one unit of flow along the path s, a, b, t, four units along the path s, a, d, t, and one more unit along the path s, c, d, t yields a maximum flow, and the fact that this flow is optimal is witnessed by the cut $(\{s, a, b, c, d\}, \{t\})$, which has capacity 6.

If all capacities are integral and if we start from an integral flow vector, e.g., the flow vector x such that $x_{ij} = 0$ for all $(i, j) \in E$, then the Ford-Fulkerson algorithm maintains integrality and increases the overall amount of flow by at least one unit in each iteration. The algorithm is therefore guaranteed to find a maximum flow after a finite number of iterations. Clearly, the latter also holds when all capacities are rational.

12.3 The Bipartite Matching Problem

A *matching* of a graph (V, E) is a set of edges that do not share any vertices, i.e., a set $M \subseteq E$ such for all $(s, t), (u, v) \in M$, $s \neq u$ and $s \neq v$. Matching M is called perfect if it covers every vertex, i.e., if $|M| = |V|/2$. A graph is k-regular if every vertex has degree k. Using flows it is easy to show that every k-regular bipartite graph, for $k \geqslant 1$, has a perfect matching. For this, consider a k-regular bipartite graph $(L \uplus R, E)$, orient all edges from L to R, and add two new vertices s and t and new edges (s, i) and (j, t) for every $i \in L$ and $j \in R$. Finally set the capacity of every new edge to 1, and that of every original edge to infinity. We can now send $|L|$ units of flow from s to t by setting the flow to 1 for every new edge and to $1/k$ for every original edge. The Ford-Fulkerson algorithm is therefore guaranteed to find an integral solution with at least the same value, and it is easy to see that such a solution corresponds to a perfect matching.

This result is a special case of a well-known characterization of the bipartite graphs that have a perfect matching. It should not come as a surprise that this characterization can be obtained from the max-flow min-cut theorem as well.

THEOREM 12.2 (Hall's Theorem). *A bipartite graph* $G = (L \uplus R, E)$ *with* $|L| = |R|$ *has a perfect matching if and only if* $|N(X)| \geqslant |X|$ *for every* $X \subseteq L$*, where* $N(X) = \{j \in R : i \in X, (i, j) \in E\}$.