Network Mathematics Graduate Programme

Hamilton Institute, Maynooth, Ireland

Lecture Notes

Optimization I

Angelia Nedić[1]

4th August 2008

[1]Industrial and Enterprise Systems Engineering Department, University of Illinois at Urbana-Champaign, Urbana IL 61801. E-mail: angelia@illinois.edu

Contents

Chapter 1

Review and Miscellanea

In this chapter, we review briefly without proof some basic concepts and facts of mathematical (real) analysis and linear algebra. We do assume that the reader is familiar with the elementary calculus and linear algebra such as fundamental properties and operations with scalar functions (continuity, derivatives, integrals, etc.) and matrices (addition, multiplication, inversion, determinant, etc.).

1.1 Linear Algebra

This section provides notation, definitions, and basic results of linear algebra. More on this material can be found, for example, in the textbook by Strang [32]. More general treatment of matrices can be found in the book by Horn and Johnson [18].

1.1.1 Vectors and Set Operations

Vectors

We use $\mathbb{R}^n$ to denote the set of n-dimensional vectors. We view the vectors of $\mathbb{R}^n$ as columns. Given a vector, $x \in \mathbb{R}^n$, we write x_i to denote its i-th component. We write $x \geq 0$ and $x > 0$ when, respectively, $x_i \geq 0$ and $x_i > 0$ for all components i. For any vectors $x, y \in \mathbb{R}^n$, we write $x \geq y$ and $x > y$ when $x - y \geq 0$ and $x - y > 0$, respectively. Similarly, we interpret $x \leq 0$, $x < 0$, $x \leq y$, and $x < y$.

For a vector $x \in \mathbb{R}^n$, we write x^+ to denote the vector of componentwise maximum of x and the zero vector, i.e.,

$$x^+ = \begin{bmatrix} \max\{x_1, 0\} \\ \vdots \\ \max\{x_n, 0\} \end{bmatrix}.$$

Note that $x^+ \geq 0$. Similarly, we define x^- as the componentwise minimum of x and the

zero vector,

$$x^- = \begin{bmatrix} \min\{x_1, 0\} \\ \vdots \\ \min\{x_n, 0\} \end{bmatrix}.$$

Note that $x^- \leq 0$. Furthermore, we have $x = x^+ + x^-$.

We use x^T to denote the transpose of a vector x. Accordingly, we use $x^T y$ to denote the *inner product* of two vectors $x, y \in \mathbb{R}^n$, i.e., $x^T y = \sum_{i=1}^n x_i y_i$. The vectors $x, y \in \mathbb{R}^n$ are *orthogonal* when their inner product is zero, i.e., $x^T y = 0$.

Set Operations

We denote the empty set by $\emptyset$. Given a set $X \subseteq \mathbb{R}^n$, its *complement* set is

$$X^c = \{x \in \mathbb{R}^n \mid x \notin X\}.$$

Given a scalar $\alpha \in \mathbb{R}$ and a set $X \subseteq \mathbb{R}^n$, the *scaled set* αX is given by

$$\alpha X = \{z = \alpha x \mid x \in X\}.$$

Given two sets, their *set difference* $X \setminus Y$ is the set given by

$$X \setminus Y = \{z \mid z \in X \text{ and } z \notin Y\} \qquad \text{difference.}$$

Given two sets, their set *intersection* $X \cap Y$, *union* $X \cup Y$, *Cartesian product* $X \times Y$, and *sum* $X + Y$ are the sets respectively given by

$$X \cap Y = \{z \mid z \in X \text{ and } z \in Y\} \qquad \text{intersection}$$

$$X \cup Y = \{z \mid z \in X \text{ or } z \in Y\} \qquad \text{union}$$

$$X \times Y = \{(x, y) \mid x \in X,\ y \in Y\} \qquad \text{Cartesian product}$$

$$X + Y = \{x + y \mid x \in X,\ y \in Y\} \qquad \text{sum.}$$

The preceding definitions extend naturally to more than two sets. For example, the intersection of sets $X_1, \ldots, X_m \subseteq \mathbb{R}^n$ is given by

$$X_1 \cap \cdots \cap X_m = \{x \mid x \in X_i \text{ for all } i = 1, \ldots, m\}.$$

1.1.2 Linear Combination and Independence

Given scalars $\alpha_1, \ldots, \alpha_m \in \mathbb{R}$ and vectors $x_1, \ldots, x_m \in \mathbb{R}^m$, the vector z given by

$$z = \alpha_1 x_1 + \ldots + \alpha_m x_m$$

is referred to as a *linear combination* of vectors $x_1, \ldots, x_m$.

The vectors $x_1, \ldots, x_m$ are said to be *linearly dependent* when the zero vector can be obtained as a nonzero linear combination of these vectors. Formally, $x_1, \ldots, x_m$ are linearly dependent when there exists scalars $\alpha_1, \ldots, \alpha_m$ not all equal to zero and such that

$$\alpha_1 x_1 + \ldots + \alpha_m x_m = 0.$$

The vectors $x_1, \ldots, x_m$ are said to be *linearly independent* when they are not linearly dependent. Formally, they are independent when the equality

$$\alpha_1 x_1 + \ldots + \alpha_m x_m = 0$$

holds *only for* $\alpha_1 = 0, \ldots, \alpha_m = 0$. In other words, the vectors $x_1, \ldots, x_m$ are linearly independent when the zero vector cannot be obtained as a nonzero linear combination of these vectors.

1.1.3 Subspace and Dimension

A nonempty set $S \subseteq \mathbb{R}^n$ is a *subspace* of $\mathbb{R}^n$ when every linear combination of its elements belongs to S. Formally, S is subspace when for every set of vectors $x_1, \ldots, x_m \in S$ we have

$$\alpha_1 x_1 + \ldots + \alpha_m x_m \in S \qquad \text{for any set of scalars } \alpha_1, \ldots, \alpha_m.$$

Note that $\mathbb{R}^n$ is a subspace. Also, the set containing only the zero vector is also a subspace.

Given a subspace $S \subseteq \mathbb{R}^n$ and a set of vectors $x_1, \ldots, x_m$ in S, the vectors $x_1 \ldots, x_m$ are said to *span* the subspace S when every vector in S can be expressed as a linear combination of vectors $x_1, \ldots, x_m$, i.e.,

$$S = \{\alpha_1 x_1 + \cdots + \alpha_m x_m \mid \alpha_i \in \mathbb{R} \text{ for all } i\}.$$

A set of vectors $x_1, \ldots, x_m \in S$ is said to be a *basis* of the subspace S when the vectors $x_1, \ldots, x_m$ span the subspace S and they are linearly independent. A subspace $S \subseteq \mathbb{R}^n$ can have more than one basis. However, every basis of S has the same number of vectors. This common number is the *dimension* of the subspace S. Note that the zero subspace, $S = \{0\}$, has zero dimension.

We use e_j to denote a vector in $\mathbb{R}^n$ whose j-th entry is 1 and all other entries are 0. The vectors $e_1, \ldots, e_n$ are the standard basis for $\mathbb{R}^n$. Note that these vectors are mutually orthogonal, i.e.,

$$e_i^T e_j = 0 \qquad \text{for all } i \neq j.$$

They are also refereed to as standard orthogonal basis for $\mathbb{R}^n$.

1.1.4 Affine Sets

A set $X \subseteq \mathbb{R}^n$ is an *affine set* when X is a translation of a subspace in $\mathbb{R}^n$, i.e., when X can be written as

$$X = x + S = \{x + s \mid s \in S\} \tag{1.1}$$

for some $x \in X$ and some subspace $S \subseteq \mathbb{R}^n$. We also say that the affine set X is generated by the subspace S. The dimension of an affine set X is the same as the dimension of its generating subspace S. For example, every set $\{x\}$ for $x \in \mathbb{R}^n$ is an affine (singleton) set since $\{x\} = x + \{0\}$. Each of these sets has zero dimension. The dimension of an affine set X is denoted by $\dim X$.

If X is an affine set then the subspace S that generates X is unique. However, the translation vector x in Eq. (1.1) is not unique. In fact, any vector $x \in X$ can be used. Formally, if X is an affine set generated by the subspace S, then we have

$$X = x_0 + S \qquad \text{for any } x_0 \in X.$$

1.1.5 Orthogonal Vectors and Orthogonal Subspace

We say that vectors $x, y \in \mathbb{R}^n$ are *orthogonal* when their inner product is zero, i.e., when $x^T y = 0$. We often write $x \perp y$ to denote that x and y are orthogonal.

Given a set $X \subseteq \mathbb{R}^n$, *the orthogonal complement* of X is the set $X^\perp$ of vectors y that are orthogonal to every element in X, i.e.,

$$X^\perp = \{y \mid y^T x = 0 \ \text{ for all } x \in X\}.$$

Note that $X^\perp$ is a subspace regardless whether X is a subspace or not. For a subspace S, its orthogonal complement is also referred to as *orthogonal subspace* of S. Note that $(S^\perp)^\perp = S$. Furthermore, the sum of the dimensions of a subspace $S \subseteq \mathbb{R}^n$ and its orthogonal subspace $S^\perp$ is equal to the dimension of the underlying space $\mathbb{R}^n$, i.e.,

$$\dim S + \dim S^\perp = n \qquad \text{for a subspace } S \subseteq \mathbb{R}^n.$$

1.1.6 Vector Norm

A vector norm is a scalar function that assigns a nonnegative scalar to every vector $x \in \mathbb{R}^n$. Specifically, a *norm* $\|\cdot\| : \mathbb{R}^n \to [0, +\infty)$ is a scalar function with the following defining properties:

1. *Nonnegativity* $\|x\| \geq 0$ for all $x \in \mathbb{R}^n$, where $\|x\| = 0$ if and only if $x = 0$.
2. *Homogeneity* $\|\alpha x\| = |\alpha| \, \|x\|$ for any $x \in \mathbb{R}^n$ and any $\alpha \in \mathbb{R}$.
3. *Triangle Inequality or Subadditivity* $\|x + y\| \leq \|x\| + \|y\|$ for all $x, y \in \mathbb{R}^n$.

For the most part, we use Euclidean norm given by

$$\|x\| = \sqrt{x^T x} = \sqrt{\sum_{i=1}^{n} x_i^2}.$$

Occasionally, we also use 1-norm and ∞-norm, respectively, given by

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|,$$

$$\|x\|_\infty = \max_{1\le i\le n} |x_i|.$$

When we write $\|\cdot\|$, then it is Euclidean norm. The following are important relations:

$$|x^T y| \le \|x\| \cdot \|y\| \qquad \text{for any } x, y \in \mathbb{R}^n \qquad \text{Schwartz inequality}^1.$$

$$\|x+y\|^2 = \|x\|^2 + \|y\|^2 \qquad \text{for any } x, y \in \mathbb{R}^n \text{ with } x^T y = 0 \qquad \text{Pythagorean Theorem.}$$

1.1.7 Matrices

Notation and Basics

Given a matrix A, we use A_{ij} or $[A]_{ij}$ to denote its ij-th entry. We write A_i or $[A]_i$ to denote its i-th row vector. Similarly, we use A^j or $[A]^j$ to denote its j-th column vector. We denote the *identity matrix* by I. We use A^T to denote the transpose of the matrix A.

The *rank* of A is the largest number of linearly independent columns of A. We say that A has *full column rank* when the columns of A are linearly independent. Similarly, We say that A has *full row rank* when the rows of A are linearly independent. We say that A has *full rank* when its rank is equal to the minimum of m and n. The matrix A and its transpose A^T have the same rank.

Null Space and Range

Given an $m \times n$ matrix A, the *null space of* A is the set N_A defined by

$$N_A = \{x \in \mathbb{R}^n \mid Ax = 0\}.$$

The *range of* A is the set R_A given by

$$R_A = \{y \in \mathbb{R}^m \mid y = Ax \ \text{ for some } x \in \mathbb{R}^n\}.$$

Both the null space and the range are subspaces (of $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively). There is an important orthogonality relation between the null space of A and the range of its complement A^T. In particular, we have

$$N_A^\perp = R_{A^T} \tag{1.2}$$

1.1.8 Square Matrices

For a square matrix A, we use $\det A$ to denote its determinant. The matrix is said to be *singular* when its determinant is zero; otherwise, it is *nonsingular* or invertible.

For an $n \times n$ nonsingular matrix A, its inverse is the $(n \times n)$ matrix A^{-1} such that

$$A^{-1}A = AA^{-1} = I.$$

The inverse matrix A^{-1} is unique.

For $n \times n$ nonsingular matrices A and B, the following relation holds

$$(AB)^{-1} = B^{-1}A^{-1}.$$

Additional important properties of square matrices are given in the following lemma.

Lemma 1 *For a square matrix A, the following statements are equivalent:*

(a) A is nonsingular.

(b) A^T is nonsingular.

(c) A has full column rank.

(d) A has full row rank.

(e) $Ax = 0$ if and only if $x = 0$.

(f) For every $y \in \mathbb{R}^n$, the equation $AX = y$ as a unique solution $x \in \mathbb{R}^n$.

(g) The inverse A^{-1} exists and it is unique.

1.1.9 Eigenvalues and Eigenvectors

Given an $n \times n$ square matrix A, we say that a complex number λ is an *eigenvalue* of A when

$$Ax = \lambda x \qquad \text{for some nonzero } x \in \mathbb{R}^n.$$

A vector $x \neq 0$ satisfying the preceding relation is referred to as *an eigenvector of A* associated with the eigenvalue λ. Note that the eigenvalues are in general *complex numbers*. Furthermore, every square matrix has n eigenvalues (possibly repeated). Also, if λ is a complex eigenvalue of A, then the complex conjugate $\bar{\lambda}$ is also an eigenvalue of A.

The set of all eigenvalues of A is the *spectrum of A* and is denoted by $\sigma(A)$. The *spectral radius of A* is the largest magnitude of the eigenvalues, and it is denoted by $\rho(A)$. Formally, it is given by

$$\rho(A) = \max_{\lambda \in \sigma(A)} |\lambda|.$$

The determinant of A is equal to the product of the eigenvalues of A, i.e.,

$$\det A = \lambda_1 \cdots \lambda_n.$$

The *trace* of A, denoted by $\mathrm{Tr}A$, is defined as the sum of the diagonal entries of A

$$\mathrm{Tr}A = \sum_{i=1}^{n} A_{ii}.$$

The trace of A is equal to the sum of the eigenvalues of A,

$$\mathrm{Tr}A = \sum_{i=1}^{n} \lambda_i.$$

Some additional properties of the eigenvalues of square matrices are given in the following lemma.

Lemma 2 *Let A be an $n \times n$ square matrix. We have:*

(a) *A^T has the same eigenvalues as A, i.e., $\sigma(A^T) = \sigma(A)$.*

(b) *Let B be a square matrix, and assume that $B = CAC^{-1}$ for some invertible matrix C. Then, the matrices A and B have the same eigenvalues, i.e., $\sigma(A) = \sigma(B)$.*

(c) *Let A be invertible and let its eigenvalues be $\lambda_1, \ldots, \lambda_n$. Then, the eigenvalues of A^{-1} are $\frac{1}{\lambda_1} \ldots, \frac{1}{\lambda_n}$, i.e.,*

$$\sigma(A^{-1}) = \left\{ \frac{1}{\lambda} \,\middle|\, \lambda \in \sigma(A) \right\}.$$

(d) *Let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of A. Then, the eigenvalues of the matrix $\alpha I + A$ are $\alpha + \lambda_1, \ldots, \alpha + \lambda_n$, i.e.,*

$$\sigma(\alpha I + A) = \alpha + \sigma(A).$$

(e) *Let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of A. Then, the eigenvalues of A^k are $\lambda_1^k, \ldots, \lambda_n^k$, i.e.,*

$$\sigma(A^k) = \{\lambda^k \mid \lambda \in \sigma(A)\}.$$

1.1.10 Matrix Norms

There are several matrix norms. Here, we exclusively consider the *matrix norms induced* by vector norms. We write $\|A\|$ to denote the matrix norm induced by Euclidean vector norm, which is given by

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}.$$

Similarly, we write $\|A\|_1$ and $\|A\|_\infty$ to denote the matrix norms induced by vector 1-norm and ∞-norm, respectively, which are given by

$$\|A\|_1 = \sup_{x \neq 0} \frac{\|Ax\|_1}{\|x\|_1},$$

$$\|A\|_\infty = \sup_{x \neq 0} \frac{\|Ax\|_\infty}{\|x\|_\infty},$$

The norm $\|A\|$ induced by Euclidean vector norm is also known as *spectral norm*. It satisfies the following relation

$$\|A\| = \max \left\{ \sqrt{\lambda} \mid \lambda \text{ is an eigenvalue of } A^T A \right\}.$$

The norm $\|A\|_1$ induced by vector 1-norm is also known as *maximum column sum matrix norm*. It is equivalently given by the following relation:

$$\|A\|_1 = \max_j \sum_i |A_{ij}|.$$

Similarly, the norm $\|A\|_\infty$ induced by vector ∞-norm is also known as *maximum row sum matrix norm*. The following relation holds:

$$\|A\|_\infty = \max_i \sum_j |A_{ij}|.$$

In the space of all square matrices A, the spectral radius $\rho(A)$ is a continuous function of A (the distance between A and B is measured by $\|A - B\|$, or any other matrix norm).

Note that in view of Lemma 2(a), we have for any square matrix A,

$$\|A\| = \|A^T\|.$$

1.1.11 Symmetric Matrices

A matrix is said to be symmetric when $A = A^T$. Symmetric matrices have special eigenvalue properties, as given in the following.

Lemma 3 *Let A be an $n \times n$ symmetric matrix. We have*

(a) The eigenvalues of A are real.

(b) There exist unit-norm and mutually orthogonal eigenvectors $x_1, \ldots, x_n$ associated with eigenvalues $\lambda_1, \ldots, \lambda_n$ of A, respectively, i.e., the vectors such that

$$Ax_i = \lambda_i x_i \qquad \text{with } \|x_i\| = 1 \qquad \text{for all } i = 1, \ldots, n,$$

$$x_i^T x_j = 0 \qquad \text{for all } i \neq j.$$

Furthermore, the matrix A has the following representation

$$A = \sum_{i=1}^n \lambda_i \, x_i x_i^T.$$

Additional important properties related to the norm of a square matrix A and its powers A^k are discussed in the following lemma.

Lemma 4 *Let A be a square symmetric matrix. We have:*

(a) $\|A\| = \rho(A) = \max_{\lambda \in \sigma(A)} |\lambda|$.

(b) $\|A^k\| = \|A\|^k$ for any integer $k \geq 0$.

(c) Assume that A is nonsingular. Then

$$\|A^{-1}\| = \max_{\lambda \in \sigma(A)} \frac{1}{|\lambda|}.$$

(d) Let A be of size $n \times n$ and let $\lambda_1 \leq \cdots \leq \lambda_n$ be its eigenvalues sorted in a nondecreasing order. Then,

$$\lambda_1 \|x\|^2 \leq x^T A x \leq \lambda_n \|x\|^2 \qquad \text{for all } x \in \mathbb{R}^n.$$

Positive Semidefinite and Positive Definite Matrices

Let A be an $n \times n$ symmetric matrix A. The matrix A is *positive semidefinite* when

$$x^T Ax \geq 0 \qquad \text{for all } x \in \mathbb{R}^n.$$

The matrix A is *positive definite* if the preceding inequality is strict when $x \neq 0$, i.e.,

$$x^T Ax > 0 \qquad \text{for all } x \in \mathbb{R}^n \text{ with } x \neq 0.$$

Similarly, A is *negative semidefinite* when

$$x^T Ax \leq 0 \qquad \text{for all } x \in \mathbb{R}^n,$$

and it is *negative definite* if the preceding inequality is strict when $x \neq 0$, i.e.,

$$x^T Ax < 0 \qquad \text{for all } x \in \mathbb{R}^n \text{ with } x \neq 0.$$

Note that positive semidefinite, positive definite as well as negative semidefinite and negative definite matrices are all square and symmetric by definition.

Lemma 5 *Let A be a symmetric matrix.*

(a) A is positive semidefinite if and only if all its eigenvalues are nonnegative.

(b) A is positive definite if and only if all its eigenvalues are positive.

(c) A is negative semidefinite (negative definite) if and only if all its eigenvalues are nonnegative (negative).

(d) Let A be positive definite. Then, its inverse is also positive definite.

A positive semidefinite matrix has a *symmetric square root* matrix denoted by $A^{1/2}$. The symmetric square root matrix $A^{1/2}$ is such that

$$A^{1/2}A^{1/2} = A.$$

Further properties of the square root matrix are given in the following lemma.

Lemma 6 *Let A be a positive semidefinite matrix. Then:*

(a) $A^{1/2}$ is nonsingular if and only if A is nonsingular.

(b) $A^{-1/2}A^{-1/2} = A^{-1}$.

(c) $AA^{1/2} = A^{1/2}A$.

1.2 Real Analysis and Multivariate Calculus

This section contains a brief review of basic notions and results related to vector sequences (such as accumulation points and convergence), topological properties of sets in $\mathbb{R}^n$ (such as closed and open sets), and vector functions (such as continuity, differentiability and Taylor expansions). More on these topics can be found in many textbooks on real analysis such as, for example, Rudin [27], Kolmogorov and Fomenko [20], and Zorich [35]. Also, a good summary of these results (and more) can be found in the books by Bertsekas [5], Bertsekas, Nedić and Ozdaglar [9], Boyd and Vandenberghe [13] and Polyak [25].

Throughout this section, we consider n-dimensional vector space $\mathbb{R}^n$ equipped with the standard inner product $x^T y = \sum_{i=1}^n x_i y_i$ and the Euclidean norm $\|x\| = \sqrt{x^T x} = \sqrt{\sum_{i=1}^n x_i^2}$.

1.2.1 Vector Sequence

Given a vector sequence $\{x_k\} \subset \mathbb{R}^n$ and an infinite index set $\mathcal{K} \subseteq \{1, 2, \ldots\}$, we refer to the sequence $\{x_k \mid k \in \mathcal{K}\}$ as a *subsequence of the sequence* $\{x_k\}$. We also use $\{x_k\}_{\mathcal{K}}$ or $\{x_{k_i}\}$ to denote a subsequence of $\{x_k\}$. We say that the sequence $\{x_k\}$ is *bounded* when for some scalar C, we have

$$\|x_k\| \leq C \qquad \text{for all } k.$$

The sequence $\{x_k\}$ *converges to a vector* $\tilde{x} \in \mathbb{R}^n$ if

$$\lim_{k \to \infty} \|x_k - \tilde{x}\| = 0.$$

The vector $\tilde{x}$ is referred to as *the limit of* $\{x_k\}$. When the sequence $\{x_k\}$ converges to a vector $\tilde{x}$, we also write $x_k \to \tilde{x}$.

If the sequence does not converge, we say that it *diverges* or that it is a *divergent sequence*. Somewhat abusing the terminology, we say that the sequence $\{x_k\}$ *converges to* ∞ when $\lim_{k\to\infty} \|x_k\| = \infty$.

A vector $y \in \mathbb{R}^n$ is an *accumulation point of the sequence* $\{x_k\}$ when there is a subsequence $\{x_{k_i}\}$ of the sequence $\{x_k\}$ converging to y, i.e., such that $x_{k_i} \to y$ as $i \to \infty$.

Theorem 1 *(Bolzano) A bounded sequence* $\{x_k\} \subset \mathbb{R}^n$ *has at least one accumulation point.*

Given a scalar sequence $\{x_k\} \subset \mathbb{R}$, we say that a is *an upper bound for* $\{x_k\}$ when

$$x_k \leq a \qquad \text{for all } k.$$

The smallest upper bound for $\{x_k\}$ is referred to as *the supremum* of $\{x_k\}$, and it is denoted by $\sup_k x_k$. If there exists an index $\tilde{k}$ such that

$$x_{\tilde{k}} = \sup_k x_k,$$

then we say that the *supremum of* x_k *is attained*, and we write $\max_k x_k$ instead of $\sup_k x_k$.

Similarly, we say that b is *a lower bound for* $\{x_k\}$ when

$$b \leq x_k \qquad \text{for all } k.$$

The largest lower bound for $\{x_k\}$ is referred to as *the infimum* of $\{x_k\}$, and it is denoted by $\inf_k x_k$. When there exists an index $\tilde{k}$ such that

$$x_{\tilde{k}} = \inf_k x_k,$$

then we say that the *infimum of* x_k *is attained*, and we write $\min_k x_k$ instead of $\inf_k x_k$. We note that both the infimum and the supremum of $\{x_k\}$ may be infinite. Moreover, we always have

$$\inf_k x_k \leq \sup_k x_k.$$

We refer to the largest accumulation point a as *the limit superior of* $\{x_k\}$ and we write

$$\limsup_{k\to\infty} x_k = a.$$

Note that this includes the possibility that a may be infinite (i.e., $a = -\infty$ or $a = +\infty$). Similarly, we refer to the smallest accumulation point b as *the limit inferior of* $\{x_k\}$ and we write

$$\liminf_{k\to\infty} x_k = b,$$

including the possibility that b is infinite.

We always have

$$\inf_k x_k \leq \liminf_{k\to\infty} x_k \leq \limsup_{k\to\infty} x_k \leq \sup_k x_k.$$

Furthermore, there holds

$$\liminf_{k\to\infty} x_k = \limsup_{k\to\infty} x_k$$

if and only if $\{x_k\}$ *is convergent*, including the possibilities $x_k \to +\infty$ or $x_k \to -\infty$.

1.2.2 Set Topology

Open, Closed and Compact Sets

Let $X \subseteq \mathbb{R}^n$ be a nonempty set. The set X is *bounded* when there exists a scalar C such that

$$\|x\| \leq C \qquad \text{for all } x \in X.$$

A vector $\tilde{x}$ is *an accumulation* (or a limit) *point of the set* X when there is a sequence $\{x_k\} \subseteq X$ such that $x_k \to \tilde{x}$. The set X is *closed* when it contains all of its accumulation points. The set X is *open* when its complement set $X^c = \{x \in \mathbb{R}^n \mid x \notin X\}$ is closed.

A set X is either open or closed, or none of the above (neither open nor closed). The only exceptions to this rule are the whole vector space $\mathbb{R}^n$ and the empty set $\emptyset$. By convention, the sets $\mathbb{R}^n$ and $\emptyset$ are the *only sets that are both open and closed*.

We have:

- A subspace of $\mathbb{R}^n$ is closed.
- An affine set in $\mathbb{R}^n$ is closed.
- The set $B(x,r) \subseteq \mathbb{R}^n$ given by

$$B(x,r) = \{y \in \mathbb{R}^n \mid \|y - x\| \leq r\} \qquad \text{for some } x \in \mathbb{R}^n \text{ and } r > 0$$

  is closed. This ball if also referred to as *the closed ball centered at x with radius r*.

There is an alternative (equivalent) way of defining open and closed sets. The preceding definition starts with the notion of a closed set, and then defines an open set as the set whose complement is closed.

In the following definition of an open set, we use the notion of an open ball in $\mathbb{R}^n$. An *open ball in* $\mathbb{R}^n$ centered at x and with radius $r > 0$, denoted by $B(x,r)$, is the set given by

$$B(x,r) = \{y \in \mathbb{R}^n \mid \|y - x\| < r\}.$$

Given a set $X \subseteq \mathbb{R}^n$, the set X is *open* if for every $x \in X$ there is a radius r small enough (depending on x) such that the ball $B(x,r)$ is contained in the set X, i.e., for every $x \in X$ there is $r > 0$ such that $B(x,r) \subseteq X$. The set X is *closed* if its complement X^c is open.

Let J be some set. We say that J *is finite* if J has finitely many elements i.e., the cardinality of J is finite. The following are some important properties of a family of open/closed sets.

Lemma 7 *Let $\{X_j \mid j \in J\}$ be a family of sets $X_j \subseteq \mathbb{R}^n$, where I is some index set.*

(a) If X_j is closed for each $j \in J$, then the intersection set $\cap_{j\in J} X_j$ is closed.

(b) If X_j is open for each $j \in J$, then the union set $\cup_{j\in J} X_j$ is open.

(c) If J is finite and X_j is closed for each $j \in J$, then the union $\cup_{j\in J} X_j$ is closed.

(d) If J is finite and X_j is open for each $j \in J$, then the intersection $\cap_{j\in J} X_j$ is open.

A set X is *compact* when every sequence $\{x_k\} \subseteq X$ has an accumulation point $\tilde{x}$ that belongs to the set X. Compact sets in $\mathbb{R}^n$ have another characterization as given in the following.

Lemma 8 *The set X is compact if and only if it is closed and bounded.*

A closed ball in $\mathbb{R}^n$ is compact. Neither a subspace nor an affine set is compact since none of such sets is bounded.

Closure, Interior and Boundary

Let $X \subseteq \mathbb{R}^n$ be a nonempty set. *The closure of the set* X is the set of all accumulation points of X, and it is denoted by $\mathrm{cl}X$. Note that, we always have

$$X \subseteq \mathrm{cl}X,$$

where *the equality holds only when* X *is closed*.

A vector $x \in X$ is *an interior point of* X when there exists a ball centered at x with some radius r such that $B(x, r) \subseteq X$. The set of all interior points of a set X is referred to as *the interior of* X, and it is denoted by $\mathrm{int}X$. We always have

$$\mathrm{int}X \subseteq X,$$

where *the equality holds only when* X *is open*.

A vector $\tilde{x} \in \mathbb{R}^n$ is *a boundary point of the set* X when for every radius r, the ball $B(\tilde{x}, r)$ contains the points that belong to the set X and the points that do not belong to the set X. Formally, $\tilde{x}$ is a boundary point of X when

$$B(\tilde{x}, r) \cap X \neq \emptyset \quad \text{and} \quad B(\tilde{x}, r) \cap X^c \neq \emptyset \qquad \text{for all } r > 0.$$

1.2.3 Mapping and Function

Mapping

Let $X \subseteq \mathbb{R}^n$. A *mapping* from a set X to $\mathbb{R}^m$ is an operation that to every $x \in X$ assigns a vector $y \in \mathbb{R}^m$. We write $F : X \to \mathbb{R}^m$ to denote a mapping F from the set X to $\mathbb{R}^m$. We refer to X as a *domain* of the mapping F. The domain of a mapping F is denoted by $\mathrm{dom}F$. The *image of* X *under the mapping* F is the following set

$$F(X) = \{y \in \mathbb{R}^m \mid y = F(x) \text{ for some } x \in X\}.$$

Given a set $Y \subseteq \mathbb{R}^m$, the *inverse image of* Y *under the mapping* F is the following set

$$F^{-1}(Y) = \{x \in X \mid F(x) \in Y\}.$$

Let F be a mapping $F : X \to \mathbb{R}^m$ with $X \subseteq \mathbb{R}^n$. The mapping F is *affine* when for some matrix A and a vector $b \in \mathbb{R}^m$,

$$F(x) = Ax + b \qquad \text{for all } x \in X.$$

For example, a space translation (i.e., for a given $x_0 \in \mathbb{R}^n$, $F(x) = x + x_0$ for all $x \in \mathbb{R}^n$) is an affine mapping from $\mathbb{R}^n$ to $\mathbb{R}^n$.

The mapping F is *linear* when for a matrix A,

$$F(x) = Ax \qquad \text{for all } x \in X.$$

For example, given the coordinate subspace $S = \{x \in \mathbb{R}^n \mid x_j = 0 \text{ for } j \in J\}$, where $J \subseteq \{1, \ldots, n\}$, the projection on the subspace S is a linear mapping from $\mathbb{R}^n$ to $\mathbb{R}^n$, i.e., the mapping F given by

$$F(x) = Ax \quad \text{where } A_{ii} = 1 \text{ for } i \notin J \text{ and } A_{ij} = 0 \text{ otherwise}$$

is linear from $\mathbb{R}^n$ to $\mathbb{R}^n$.

Function

When f is a mapping from a set $X \subseteq \mathbb{R}^n$ to the scalar set $\mathbb{R}$, we say that f is a *function*. Some special functions defined on $\mathbb{R}^n$ include

- *Quadratic function* given for an $n \times n$ matrix Q, a vector $a \in \mathbb{R}^n$, and a scalar $b \in \mathbb{R}$, by
$$f(x) = x^T Q x + a^T x + b \qquad \text{for all } x \in \mathbb{R}^n.$$
- *Affine function* given for a vector $a \in \mathbb{R}^n$ and a scalar $b \in \mathbb{R}$, by
$$f(x) = a^T x + b \qquad \text{for all } x \in \mathbb{R}^n.$$
- *Linear function* given for a vector $a \in \mathbb{R}^n$, by
$$f(x) = a^T x \qquad \text{for all } x \in \mathbb{R}^n.$$
- *Constant function* given for a scalar $b \in \mathbb{R}$, by
$$f(x) = b \qquad \text{for all } x \in \mathbb{R}^n.$$

1.2.4 Continuity

Let $F : X \to \mathbb{R}^m$ be a mapping with $X \subseteq \mathbb{R}^n$, and let $x \in X$ be a given vector. We say that F is *continuous at the vector* x when the vectors $F(x_k)$ converge to $F(x)$ for every sequence $\{x_k\} \subseteq X$ converging to x, i.e.,
$$F(x_k) \to F(x) \qquad \text{for every } \{x_k\} \subseteq X \text{ with } x_k \to x.$$
We say F is *continuous over* X when F is continuous at every $x \in X$. When $F : \mathbb{R}^n \to \mathbb{R}^m$ is continuous over $\mathbb{R}^n$, we just say that F is *continuous*. For example, *any vector norm in* $\mathbb{R}^n$ *is a continuous function*.

A mapping $F : X \to \mathbb{R}^m$, with $X \subseteq \mathbb{R}^n$, is *Lipschitz (continuous) over* X if there exists a scalar $c > 0$ (possibly depending on X) such that
$$\|F(x) - F(y)\| \leq c\|x - y\| \qquad \text{for all } x, y \in X.$$
When the preceding relation holds for $X = \mathbb{R}^n$, we simply say F is *Lipschitz (continuous)*.

A function $f : X \to \mathbb{R}$ is *lower semicontinuous at a vector* $x \in X$ when
$$f(x) \leq \liminf_{k \to \infty} f(x_k) \qquad \text{for every } \{x_k\} \subseteq X \text{ with } x_k \to x.$$
A function f *is upper semicontinuous at a vector* $x \in X$ when the function $-f$ is lower semicontinuous at x, i.e.,
$$f(x) \geq \limsup_{k \to \infty} f(x_k) \qquad \text{for every } \{x_k\} \subseteq X \text{ with } x_k \to x.$$
When f is lower semicontinuous at every point x in some set X, we say f *is lower semicontinuous over* X. When f is lower semicontinuous at every point $x \in \mathbb{R}^n$, we say f *is lower semicontinuous*. Analogous terminology is used for upper semicontinuity.

Lemma 9 *The following are some properties of continuous mappings.*

(a) Let $Y \subseteq \mathbb{R}^p$. Let $F : \mathbb{R}^n \to \mathbb{R}^m$ be a continuous mapping over $\mathbb{R}^n$ and $G : Y \to \mathbb{R}^n$ be a continuous mapping over Y. Then, their composition $F \circ G : Y \to \mathbb{R}^m$ given by

$$(F \circ G)(y) = F(G(y)) \qquad \text{for all } y \in Y$$

is continuous over Y.

(b) Let $F : \mathbb{R}^n \to \mathbb{R}^m$ be a continuous map.

(i) Let $X \subset \mathbb{R}^n$ be a compact set. Then, the image $F(X)$ of X under F is compact.

(ii) Let $Y \subseteq \mathbb{R}^m$. If Y is open, then the inverse image $F^{-1}(Y)$ is open. If Y is closed, then the inverse image $F^{-1}(Y)$ is closed.

Given a function $f : \mathbb{R}^n \to \mathbb{R}$, *the (lower) level set $L_\gamma(f)$ of f* for a given $\gamma \in \mathbb{R}$ is the set defined by

$$L_\gamma(f) = \{x \in \mathbb{R}^n \mid f(x) \leq \gamma\}.$$

When f is lower semicontinuous the level set $L_\gamma(f)$ is closed for each γ. In fact, a stronger relation holds, as given in the following.

Theorem 2 *The function $f : \mathbb{R}^n \to \mathbb{R}$ is lower semicontinuous if and only if the level set $L_\gamma(f)$ is closed for each $\gamma \in \mathbb{R}$.*

Let $X \subseteq \mathbb{R}^n$ and $f : X \to \mathbb{R}$. The function f *is coercive over X* when it satisfies the following relation

$$\lim_{k\to\infty} f(x_k) = +\infty \qquad \text{for every sequence } \{x_k\} \subseteq X \text{ with } \|x_k\| \to \infty.$$

When $X = \mathbb{R}^n$ in the preceding relation, we just say f *is coercive.*

In the next theorem, we provide a condition on the function f and the given set X that is necessary for the attainment of both the infimum of $f(x)$ over X and the supremum of f over X.

Theorem 3 *(Weierstrass) Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous function and $X \subseteq \mathbb{R}^n$ be a nonempty compact set. Then, both $\inf_{x\in X} f(x)$ and $\sup_{x\in X} f(x)$ are finite, and there exist vectors $x^* \in X$ and $\tilde{x} \in X$ such that*

$$f(x^*) = \inf_{x\in X} f(x) \qquad \text{and} \qquad f(\tilde{x}) = \sup_{x\in X} f(x).$$

The following theorem provides some conditions on the function f and the given set X that can guarantee only the attainment of the infimum of $f(x)$ over X.

Theorem 4 *Let $X \subseteq \mathbb{R}^n$ be a nonempty set and let $f : X \to \mathbb{R}$ be a function lower semicontinuous over X. Furthermore, let any of the following conditions be satisfied:*

(i) The function f is coercive over X and the set X is closed.

(ii) For some $\gamma \in \mathbb{R}$, the set $\{x \in X \mid f(x) \leq \gamma\}$ is nonempty and compact.

(iii) The set X is compact.

Then, $\inf_{x \in X} f(x)$ *is finite and there exists a vector $x^* \in X$ such that*

$$f(x^*) = \inf_{x \in X} f(x).$$

1.2.5 Differentiability

Differentiable Functions

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function, and let a vector $x \in \mathbb{R}^n$ and a direction $d \in \mathbb{R}^n$ be given. Consider the limit

$$f'(x; d) = \lim_{\lambda \downarrow 0} \frac{f(x + \lambda d) - f(x)}{\lambda},$$

where $\lambda \downarrow 0$ means that $\lambda \to 0$ with $\lambda > 0$. When this limit exists, we say that $f'(x; d)$ is *the directional derivative of f along direction d at the point x.*

Suppose that f at the point x has directional derivatives $f'(x; d)$ in all directions $d \in \mathbb{R}^n$. If the directional derivative function $f'(x; \cdot) : \mathbb{R}^n \to \mathbb{R}$ is linear, we say that f *is differentiable at x.* This type of differentiability is also known as *Gateaux differentiability.* It is equivalent to *the existence of the gradient $\nabla f(x)$ of f at x*, which satisfies

$$f'(x; d) = \nabla f(x)^T d \qquad \text{for all } d \in \mathbb{R}^n.$$

The gradient $\nabla f(x)$ is a column vector given by

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix},$$

where $\frac{\partial f(x)}{\partial x_i}$ for each $i = 1, \ldots, n$ is a *partial derivative of f at x* given by

$$\frac{\partial f(x)}{\partial x_i} = \lim_{\lambda \to 0} \frac{f(x + \lambda e_i) - f(x)}{\lambda},$$

with e_i being i-th basis vector in $\mathbb{R}^n$ (see Section 1.1.3). When f is differentiable at x for all x in some (open) set X, we say f *is differentiable over X.* If f is differentiable over $\mathbb{R}^n$, we just say f *is differentiable.*

Let f be differentiable over some (open) set X and assume that $\nabla f(\cdot)$ is continuous function over X. We then say that f *is continuously differentiable over X.* In this case, the following relation holds

$$\lim_{\|d\| \to 0} \frac{f(x + d) - f(x) - \nabla f(x)^T d}{\|d\|} = 0 \qquad \text{for all } x \in X.$$

Let $f : \mathbb{R}^n \to \mathbb{R}$ have second partial derivatives $\frac{\partial^2 f(x)}{\partial x_i \partial x_j}$ for all i, j at a given vector x. Then, we say *f is twice differentiable at x*. The matrix $\nabla^2 f(x)$ with entries $[\nabla^2 f(x)]_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$ is *the Hessian of f at x*, i.e., the Hessian $\nabla^2(x)$ is given by

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_n \partial x_n} \end{bmatrix}.$$

Since the second partial derivatives are symmetric,

$$\frac{\partial^2 f(x)}{\partial x_i \partial x_j} = \frac{\partial^2 f(x)}{\partial x_j \partial x_i} \qquad \text{for all } i, j,$$

the *Hessian $\nabla^2 f(x)$ is a symmetric matrix.*

If the Hessian $\nabla^2 f(x)$ exists at every x in an (open) set X, we say that *f is twice differentiable over X*. In addition, when the Hessian is continuous over X, we say *f is twice continuously differentiable over X*. Similarly, we say *f is twice (continuously) differentiable*, when f is twice (continuously) differentiable over $\mathbb{R}^n$.

Mean Value Theorem

The following theorem provide some useful properties of continuous and twice continuous differentiable functions.

Theorem 5 *Let $X \subseteq \mathbb{R}^n$ be an open set and let $x, y \in X$ be arbitrary. Also, let $f : X \to \mathbb{R}$.*

(a) If f is continuously differentiable over X, then

$$f(y) = f(x) + \nabla f(z)^T (y - x),$$

where $z = \alpha x + (1 - \alpha) y$ for some scalar $\alpha \in [0, 1]$.

(b) If f is twice continuously differentiable over X, then

$$f(y) = f(x) + \nabla f(x)^T (y - x) + \frac{1}{2}(y - x)^T \nabla^2 f(\xi)(y - x),$$

where $\xi = \beta x + (1 - \beta) y$ for some scalar $\beta \in [0, 1]$.

Taylor Expansions

The following theorem provides useful function value approximations based on Taylor's series expansion.

Theorem 6 *Let $X \subseteq \mathbb{R}^n$ be an open set and let $x, y \in X$ be arbitrary. Also, let $f : X \to \mathbb{R}$.*

(a) If f is continuously differentiable over X, then

$$f(y) = f(x) + \nabla f(x)^T (y - x) + o(\|y - x\|).$$

(b) If f is twice continuously differentiable over X, then

$$f(y) = f(x) + \nabla f(x)^T(y-x) + \frac{1}{2}(y-x)^T\nabla^2 f(x)(y-x) + o(\|y\|^2).$$

Here, $o(\alpha)$ is a continuous function such that $\lim_{\alpha\to 0}\frac{o(\alpha)}{\alpha} = 0$.

A function with Lipschitz continuous gradient can be approximated by a quadratic function as seen in the following theorem.

Theorem 7 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function with Lipschitz gradient, i.e., for some scalar $c > 0$,*

$$\|\nabla f(x) - \nabla f(y)\| \le c\|x-y\| \qquad \text{for all } x, y \in \mathbb{R}^n.$$

Then, we have for all $x, y \in \mathbb{R}^n$,

$$f(y) \le f(x) + \nabla f(x)^T(y-x) + \frac{c}{2}\|y-x\|^2.$$

Differentiable Mappings

Let $F : \mathbb{R}^n \to \mathbb{R}^m$ be a mapping given by

$$F(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{bmatrix},$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ for all i. When each f_i is differentiable function at a given x, then the mapping F *is differentiable at* x. The matrix whose rows are $\nabla f_1(x)^T, \ldots, \nabla f_m(x)^T$ is *the Jacobian of F at x*. The Jacobian is an $m \times n$ matrix denoted by $JF(x)$, i.e.,

$$JF(x) = \begin{bmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_m(x)^T \end{bmatrix}.$$

Chain Rules

Let $G : \mathbb{R}^p \to \mathbb{R}^n$ be a mapping differentiable at x, and let $F : \mathbb{R}^n \to \mathbb{R}^m$ be another mapping differentiable at $G(x)$. Then, the composite mapping $F \circ G$ is differentiable at x and the following *chain rule holds for the Jacobian $J(F \circ G)(x)$*:

$$J(F \circ G)(x) = JF(G(x))\, JG(x).$$

When G is a linear mapping given by

$$G(x) = Ax \qquad \text{for all } x \in \mathbb{R}^p \text{ and some } n \times p \text{ matrix } A,$$

then

$$J(F \circ G)(x) = JF(Ax))\, A.$$

Let $G : \mathbb{R}^p \to \mathbb{R}^n$ be a mapping differentiable at x, and let $f : \mathbb{R}^n \to \mathbb{R}$ be a function differentiable at $G(x)$. Then, the composite function $f \circ G$ is differentiable at x and the following *chain rule holds for the gradient* $\nabla(f \circ G)(x)$:

$$\nabla(f \circ G)(x) = JG(x)^T\, \nabla f(G(x)).$$

When G is a linear mapping given by

$$G(x) = Ax \qquad \text{for all } x \in \mathbb{R}^p \quad \text{and for an } n \times p \text{ matrix } A,$$

then

$$\nabla(f \circ G)(x) = A^T \nabla f(Ax)).$$

Moreover, if f is twice differentiable at Ax, then the following *chain rule holds for the Hessian* $\nabla^2(f \circ G)(x)$:

$$\nabla^2(f \circ G)(x) = A^T\, \nabla^2 f(Ax) A.$$

Let $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ be a function of the variable (x, y) where $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$. Then, the gradients $\nabla_x f(x, y)$ and $\nabla_y f(x, y)$ of f at (x, y) with respect to x and y, respectively, are given by

$$\nabla_x f(x, y) = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x,y)}{\partial x_n} \end{bmatrix}, \qquad \nabla_y f(x, y) = \begin{bmatrix} \frac{\partial f(x,y)}{\partial y_1} \\ \vdots \\ \frac{\partial f(x,y)}{\partial y_m} \end{bmatrix}.$$

Let $F : \mathbb{R}^p \to \mathbb{R}^n$ and $G : \mathbb{R}^p \to \mathbb{R}^m$ be two mappings differentiable at $z \in \mathbb{R}^p$. Let $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ be a function of (x, y) differentiable at $(F(z), G(z))$. Then, the gradient $\nabla_z f(F(z), G(z))$ is given by

$$\nabla_z f(F(z), G(z)) = JF(z)^T \nabla_x f(F(z), G(z)) + JG(z)^T \nabla_y f(F(z), G(z)).$$

Chapter 2

Fundamental Concepts in Convex Optimization

Convex optimization is an important topic with many practical application areas such as communication and networks, estimation and statistical signal processing, and control systems. In this section, we discuss the fundamental notions, principles and results of convex optimization.

In particular, we introduce the notion of a convex set and convex function, and discuss the basic operations on such sets and functions that preserve convexity. We then focus on convex constrained optimization problems and provide necessary conditions for existence of solutions as well as necessary and sufficient optimality conditions. We study the projection problem on a convex set, and discuss separation results which provide a link to the Lagrangian duality. We then focus on Lagrangian duality and discuss the linear programming duality, Slater condition, and Karush-Kuhn-Tucker conditions characterizing a primal-dual optimal pair.

The interested reader can find a more in depth coverage of the convexity theory in the seminal book by Rockafellar [26], convex analysis and optimization in the textbook by Bertsekas, Nedić, and Ozdaglar [9], convex optimization with wide range of engineering applications in the books by Boyd and Vandenberghe [13] and Ben-Tal and Nemirovski [3], convex analysis and nonlinear optimization by Borwein and Lewis [12].

2.1 Convex Sets

2.1.1 Definition

Convexity is defined through the notion of a line segment. Given two vectors x and y in $\mathbb{R}^n$, the line segment connecting x and y is the set $[x, y]$ formally given by

$$[x, y] = \{\alpha x + (1 - \alpha)y \mid \alpha \in [0, 1]\}.$$

Definition 1 *A set X is convex when with any two points $x, y \in X$, the line segment $[x, y]$ also belongs to the set X, i.e.,*

$$\alpha x + (1 - \alpha)y \in X \qquad \text{for any } x, y \in X \text{ and } \alpha \in (0, 1).$$

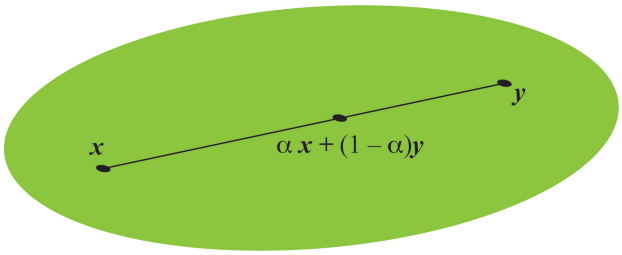

Figure 2.1: A convex set contains a segment $[x, y]$ for any x, y belonging to the set.

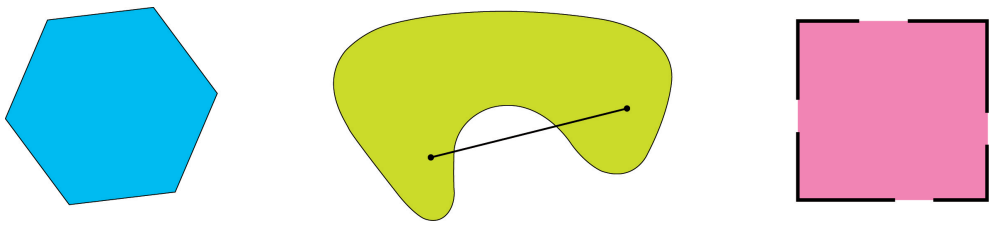

Figure 2.2: The hexagon to the left is convex, the other two sets are nonconvex.

Note that the empty set is convex by convention. A convex set definition is illustrated in Figure 2.1. Some additional pictorial examples of convex and nonconvex sets are given in Figure 2.2. Furthermore, using the definition of a convex set, one can verify that the *following sets are convex*:

(a) Any subspace of $\mathbb{R}^n$.

(b) Any affine set in $\mathbb{R}^n$.

(c) Any open ball in $\mathbb{R}^n$; also, any closed ball in $\mathbb{R}^n$.

(d) A singleton set, i.e., the set $\{x\}$ for a vector x.

(e) A line given by two vectors x and y, which is the set $\{x + t(y - x) \mid t \in \mathbb{R}\}$.

(f) A ray defined by a vector x, which is the set $\{\lambda x \mid \lambda \geq 0\}$.

(g) The nonnegative orthant in $\mathbb{R}^n$, which is the set $\mathbb{R}^n_+ = \{x \in \mathbb{R}^n \mid x \geq 0\}$.

(h) The positive orthant in $\mathbb{R}^n$, which is the set $\mathbb{R}^n_{++} = \{x \in \mathbb{R}^n \mid x > 0\}$.

(i) The set $\{x \in \mathbb{R}^2 \mid x_1 > 0,\ x_2 \geq 0\}$.

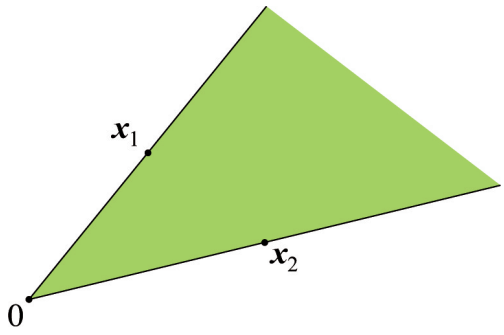

Figure 2.3: A convex cone containing the origin.

2.1.2 Special Convex Sets

We next discuss some special convex sets. We start with a cone, which in general need not be convex. We then provide a necessary and sufficient condition for convexity of a cone.

A set $C \subseteq \mathbb{R}^n$ is a *cone* when with every $x \in C$, the whole ray $\{\lambda x \mid \lambda > 0\}$ also belongs to the set C, i.e.,

$$\lambda x \in C \qquad \text{for all } x \in C \text{ and } \lambda > 0.$$

A cone may or may not contain the origin. Also, it may or may not be convex. For example, the set $\{x \in \mathbb{R}^2 \mid x_1 x_2 = 0\}$ is a cone that contains the origin and it is nonconvex. The set $\{x \in \mathbb{R}^2 \mid x_1 x_2 = 0,\ x \neq 0\}$ is a cone that does not contain the origin and it is nonconvex. The positive orthant $R^n_{++} = \{x \in \mathbb{R}^n \mid x > 0\}$ is a cone that does not contain the origin and is convex. Another convex cone containing the origin is illustrated in Figure 2.3.

Lemma 10 *(Convex Cone): A cone C is convex if and only if $C + C \subseteq C$.*

Proof. Suppose C is convex, and let $x, y \in C$. By convexity of C, we have $z = \frac{1}{2}(x+y) \in C$. Since C is a cone, it follows that $2z \in C$, thus implying that $x + y \in C$. Hence $C + C \subseteq C$.

Now assume that $C + C \subseteq C$, and let $x, y \in C$ and $\alpha \in (0, 1)$ be arbitrary. Note that $1 - \alpha > 0$. Since C is a cone, it follows that $\alpha x \in C$ and $(1 - \alpha)y \in C$. Using $C + C \subseteq C$, we obtain $\alpha x + (1 - \alpha)y \in C$, showing that C is convex. ■

A hyperplane is a set of the form $\{x \in \mathbb{R}^n \mid a^T x = b\}$ for a nonzero vector $a \in \mathbb{R}^n$ and a scalar b. The vector a is referred to as the *normal vector* of the hyperplane. A hyperplane is illustrated in Figure 2.4.

A half-space is a set of the form $\{x \mid a^T x \leq b\}$ with a nonzero vector $a \in \mathbb{R}^n$. *A hyperplane in $\mathbb{R}^n$ divides the space into two half-spaces*: $\{x \mid a^T x \leq b\}$ and $\{x \mid a^T x \geq b\}$ (cf. Figure 2.4). Note that hyperplanes and half-spaces are convex. In addition, hyperplanes are affine sets.

A *polyhedral set* is a set given by finitely many linear inequalities, i.e., a set of the form

$$\{x \in \mathbb{R}^n \mid a_i^T x \leq b_i,\ i = 1, \ldots, m\},$$

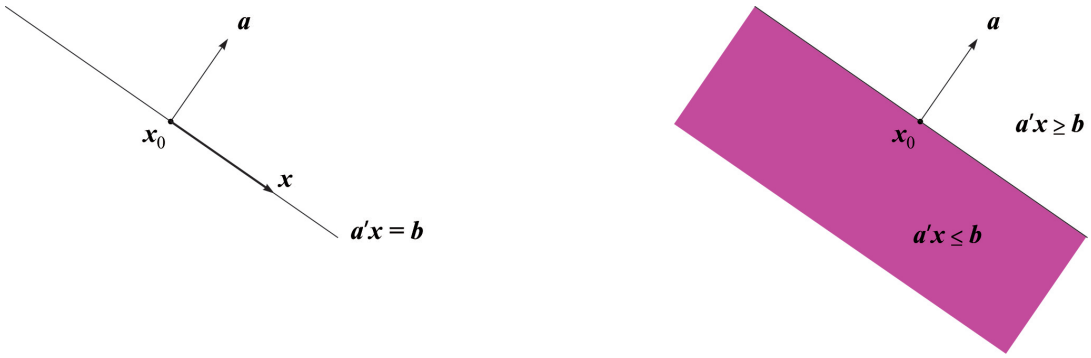

Figure 2.4: A hyperplane is shown in the plot to the left, while the two half-spaces generated by this hyperplane are depicted in the plot to the right.

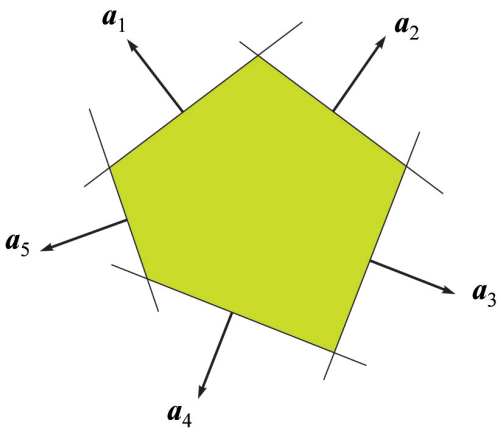

Figure 2.5: A polyhedral set.

where $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$ for all i, or compactly

$$\{x \in \mathbb{R}^n \mid Ax \leq b\} \quad \text{where } A \text{ is an } m \times n \text{ matrix.}$$

Such a set is illustrated in Figure 2.5. Note that the polyhedral set description may include linear equalities in addition to linear inequalities, i.e., the set $\{x \in \mathbb{R}^n \mid Ax \leq b,\ Bx = d\}$ is polyhedral. *Every polyhedral set is convex.*

An ellipsoid is a set of the form

$$\mathcal{E} = \{x \in \mathbb{R}^n \mid (x - x_0)^T P^{-1}(x - x_0) \leq 1\},$$

where $x_0 \in \mathbb{R}^n$ is some vector and P is a symmetric and positive definite matrix. An ellipsoid is illustrated in Figure 2.6. The vector x_0 is *the center of the ellipsoid* $\mathcal{E}$. Note that a (closed) ball $\{x \in \mathbb{R}^n \mid \|x - x_0\| \leq r\}$ is a special case of the ellipsoid $\mathcal{E}$ where $P = r^2 I$. *Ellipsoids are convex sets.*

A *norm cone* is the set of the form

$$C = \{(x, t) \in \mathbb{R}^n \times \mathbb{R} \mid \|x\| \leq t\},$$

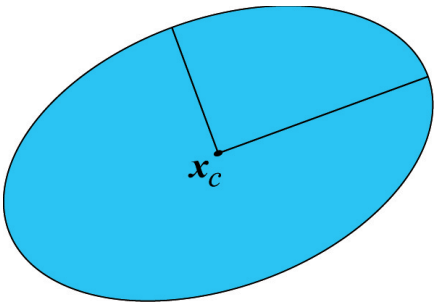

Figure 2.6: An ellipsoid in $\mathbb{R}^2$ with the center at x_c.

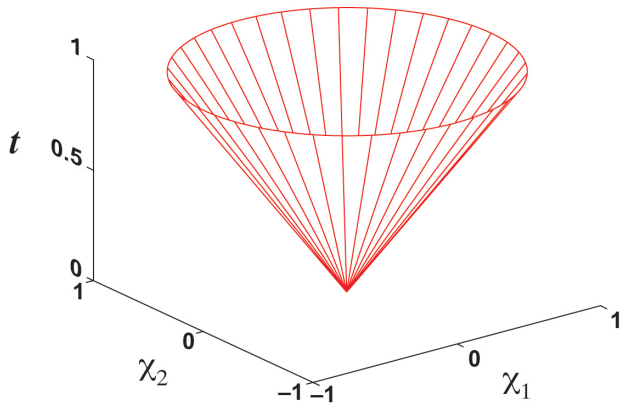

Figure 2.7: An ice-cream cone, a norm cone for Euclidean norm.

where the norm $\|\cdot\|$ can be any norm in $\mathbb{R}^n$. The norm cone for Euclidean norm is also known as *ice-cream cone*, depicted in Figure 2.7. *Any norm cone is convex.*

A simplex is a set of the form

$$\left\{ \alpha_1 v_1 + \ldots + \alpha_m v_m \,\middle|\, \sum_{i=1}^{m} \alpha_i = 1, \quad \alpha_i \geq 0 \text{ for all } i = 1, \ldots, m \right\},$$

where $v_1, \ldots, v_m \in \mathbb{R}^n$ are some vectors. The *dimension of the simplex* is the maximum number of linearly independent vectors among $v_2 - v_1, \ldots, v_m - v_1$. *Every simplex is convex.* Some special simplices include:

$$\textit{Unit simplex} \qquad \left\{ x \in \mathbb{R}^n \,\middle|\, x \geq 0, \ \sum_{i=1}^{n} x_i \leq 1 \right\},$$

$$\textit{Probability simplex} \qquad \left\{ x \in \mathbb{R}^n \,\middle|\, x \geq 0, \ \sum_{i=1}^{n} x_i = 1 \right\}.$$

2.1.3 Set Operations Preserving Convexity

Convexity of a given set X can be verified by

- Recognizing that the set is one of the "special convex sets" such as polyhedral, simplex, norm cone, etc.
- Proving that the set is convex by directly applying the definition, i.e., showing that $\alpha x + (1-\alpha)y \in X$ for every $x, y \in X$ and $\alpha \in (0,1)$.
- Show that the set is obtained from one of the "recognizable" (special) convex sets through an operation that preserves convexity.

The following theorem provides some basic operations on convex sets that preserve convexity.

Theorem 8 *Let $X, X_1, X_2 \subseteq \mathbb{R}^n$ be convex sets. Then, the following sets are also convex:*

(a) The scaled set $tX = \{tx \mid x \in X\}$ for any (fixed) $t \in \mathbb{R}$.

(b) The set sum $X_1 + X_2$.

(c) The set intersection $X_1 \cap X_2$.

(d) The Cartesian product $X_1 \times X_2$.

Proof. (a) If $t = 0$, then $tX = \{0\}$, which is evidently convex, so assume that $t \neq 0$. Let $x, y \in tX$ and $\alpha \in (0,1)$. Then, $\frac{x}{t} \in X$ and $\frac{y}{t} \in X$. Since X is convex, it follows $\alpha\frac{x}{t} + (1-\alpha)\frac{y}{t} \in X$, and therefore, by multiplying with t we see that $\alpha x + (1-\alpha)y \in tX$. Hence, tX is convex.

(b) Let $x, y \in X_1 + X_2$ and $\alpha \in (0,1)$. Then, $x = x_1 + x_2$ for some $x_1 \in X_1$ and $x_2 \in X_2$. Similarly, $y = y_1 + y_2$ for some $y_1 \in X_1$ and $y_2 \in X_2$. Since X_1 and X_2 are convex, we have $\alpha x_1 + (1-\alpha)y_1 \in X_1$ and $\alpha x_2 + (1-\alpha)y_2 \in X_2$. Hence, $\alpha x_1 + (1-\alpha)y_1 + \alpha x_2 + (1-\alpha)y_2 \in X_1 + X_2$, or equivalently $\alpha(x_1 + x_2) + (1-\alpha)(y_1 + y_2) = \alpha x + (1-\alpha)y \in X_1 + X_2$, thus showing that $X_1 + X_2$ is convex.

(c) Let $x, y \in X_1 \cap X_2$ and $\alpha \in (0,1)$. Since $x, y \in X_1$ and the set X_1 is convex, it follows that $\alpha x + (1-\alpha)y \in X_1$. Similarly, we conclude that $\alpha x + (1-\alpha)y \in X_2$. Therefore, $\alpha x + (1-\alpha)y \in X_1 \cap X_2$, showing that $X_1 \cap X_2$ is convex.

(d) Let $x, y \in X_1 \times X_2$ and $\alpha \in (0,1)$. Since $x, y \in X_1 \times X_2$, we have $x = (x_1, x_2)$ and $y = (y_1, y_2)$ with $x_i, y_i \in X_i$ for $i = 1, 2$. The sets X_1 and X_2 are convex, so that $\alpha x_1 + (1-\alpha)y_1 \in X_1$ and $\alpha x_2 + (1-\alpha)y_2 \in X_2$. Hence $(\alpha x_1 + (1-\alpha)y_1, \alpha x_2 + (1-\alpha)y_2) \in X_1 \times X_2$, or equivalently $\alpha(x_1, x_2) + (1-\alpha)(y_1, y_2) = \alpha x + (1-\alpha)y \in X_1 \times X_2$, thus showing that $X_1 \times X_2$ is convex. ■

As a consequence of Theorem 8(b), for a given set X and a vector a, *the translated set $X + a$ is convex when X is convex.* In particular, this follows by letting $X_1 = X$ and $X_2 = \{a\}$ and by noting that the singleton set $X_2 = \{a\}$ is convex.

Convexity of a set is preserved under linear transformations, as shown in the following theorem.

Theorem 9 *Let $X \subseteq \mathbb{R}^n$ and $Y \subseteq \mathbb{R}^m$ be convex sets, and let A be an $m \times n$ matrix. Then, the following sets are also convex:*

(a) The image set AX of X under A.

(b) The inverse image $A^{-1}Y$ of Y under A.

Proof. (a) Recall that the image AX is given by

$$AX = \{y \in \mathbb{R}^m \mid y = Ax \text{ for some } x \in X\}.$$

Let $y_1, y_2 \in AX$ and $\alpha \in (0,1)$. By the definition of the image set AX, we have $y_1 = Ax_1$ and $y_2 = Ax_2$ for some vectors $x_1, x_2 \in X$. Since the set X is convex, the vector $\alpha x_1 + (1-\alpha)x_2$ belongs to X. We have $A(\alpha x_1 + (1-\alpha)x_2) = \alpha Ax_1 + (1-\alpha)Ax_2 = \alpha y_1 + (1-\alpha)y_2$, and since $\alpha x_1 + (1-\alpha)x_2 \in X$, it follows that $\alpha y_1 + (1-\alpha)y_2 \in AX$. Hence AX is convex.

(b) Recall that the inverse image $A^{-1}Y$ is given by

$$A^{-1}Y = \{x \in \mathbb{R}^n \mid Ax \in Y\}.$$

Let $x_1, x_2 \in A^{-1}Y$ and $\alpha \in (0,1)$. By the definition of the inverse image $A^{-1}Y$, we have $Ax_1 \in Y$ and $Ax_2 \in Y$. For the vector $\alpha x_1 + (1-\alpha)x_2$, we have $A(\alpha x_1 + (1-\alpha)x_2) = \alpha Ax_1 + (1-\alpha)Ax_2$, which belongs to the set Y by convexity Y. Therefore, $\alpha x_1 + (1-\alpha)x_2 \in A^{-1}Y$, thus showing that $A^{-1}Y$ is convex. ■

As a consequence of Theorem 9(a), *a coordinate projection of a convex set X is convex.* For example, given a convex set $X \subseteq \mathbb{R}^2$, the set $\{x_1 \mid (x_1, x_2) \in X \text{ for some } x_2\}$ is convex.

2.2 Convex Functions

Informally speaking, a function $f : \mathbb{R}^n \to \mathbb{R}$ is convex when for every segment $[x_1, x_2]$, as the vector $x_\alpha = \alpha x_1 + (1-\alpha)x_2$ varies within the line segment $[x_1, x_2]$, the points $(x_\alpha, f(x_\alpha))$ on the graph $\{(x, f(x) \mid x \in \mathbb{R}^n\}$ lie below the segment connecting $(x_1, f(x_1))$ and $(x_2, f(x_2))$, as illustrated in Figure 2.8.

Let f be a function. *The domain of f* is a set in $\mathbb{R}^n$ defined by

$$\text{dom}(f) = \{x \in \mathbb{R}^n \mid f(x) \text{ is well defined (finite)}\}.$$

Definition 2 *A function f is convex if its domain* $\text{dom}(f)$ *is convex set and for all $x_1, x_2 \in$* $\text{dom}(f)$ *and $\alpha \in (0,1)$, the following relation holds*

$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha)f(x_2).$$

When the inequality in the preceding definition is strict for $x_1 \neq x_2$, the function f *is strictly convex.*

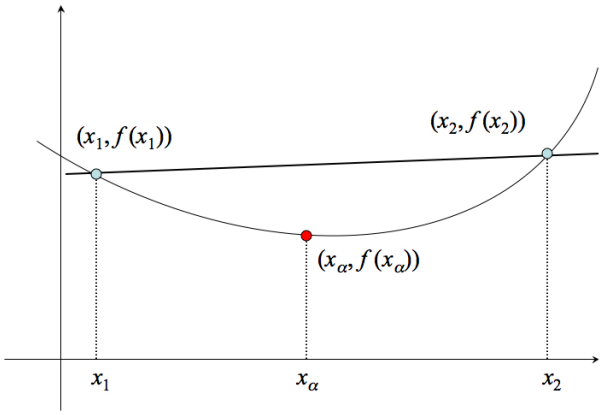

Figure 2.8: A convex function values at a point $x_\alpha = \alpha x_1 + (1-\alpha)x_2$ do not exceed the value $\alpha f(x_1) + (1-\alpha)f(x_2)$ for any $\alpha \in (0,1)$ and any x_1, x_2.

Definition 3 *A convex function f is strictly convex if for all $x_1, x_2 \in \mathrm{dom}(f)$ with $x_1 \neq x_2$, and any $\alpha \in (0,1)$, the following strict inequality holds*

$$f(\alpha x_1 + (1-\alpha)x_2) < \alpha f(x_1) + (1-\alpha)f(x_2).$$

Closely related to convex functions are concave functions.

Definition 4 *A function f is concave if $-f$ is convex.*

Since f and $-f$ have the same domain, from Definitions 2 and 4, we see that a function f is concave when its domain $\mathrm{dom}(f)$ is convex and, for all $x_1, x_2 \in \mathrm{dom}(f)$ and $\alpha \in (0,1)$, the following inequality holds

$$f(\alpha x_1 + (1-\alpha)x_2) \geq \alpha f(x_1) + (1-\alpha)f(x_2).$$

Similarly, a function f is strictly concave if $-f$ is strictly convex. Thus, a concave function f is strictly concave when the preceding inequality is strict whenever $x_1 \neq x_2$.

By using the definitions of a convex and concave function, one can verify that the *following scalar functions are convex*:

(a) Affine function $f(x) = ax + b$ for any $a, b \in \mathbb{R}$.

(b) Exponential function $f(x) = e^{ax}$ for any $a \in \mathbb{R}$.

(c) Power $f(x) = x^p$ for $x \in (0, +\infty)$ and $p \geq 1$ or $p \leq 0$.

(d) Power of absolute value $|x|^p$ for $p \geq 1$.

(e) Negative entropy $f(x) = x \ln x$ for $x \in (0, +\infty)$.

The following are some examples of *concave scalar functions*:

(a) Affine function $f(x) = ax + b$ for any $a, b \in \mathbb{R}$.

(b) Power $f(x) = x^p$ for $x \in (0, +\infty)$ and $0 \leq p \leq 1$.

(c) Logarithm $f(x) = \ln x$ for $x \in (0, +\infty)$.

Note that *the affine functions are both convex and concave*, and there are no other functions that are both convex and concave.

Any norm in $\mathbb{R}^n$ is convex. In particular, a general the norm $\|\cdot\|_p$ for $p \geq 1$,

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}$$

is convex.

There are some well-known functions of matrices that are also convex. For example, *the trace of a square matrix X is convex*, where the trace is viewed as a function of X. More specifically, the trace is a linear function: for any two $n \times n$ square matrices X and Y, and any two scalars a and b, we have

$$\mathrm{Tr}(aX + bY) = \sum_{i=1}^{n} (a[X]_{ii} + b[Y]_{ii}) = a\mathrm{Tr}X + b\mathrm{Tr}Y.$$

Some additional examples of convex functions of matrices are provided in Examples 1 and 2.

Example 1 *An affine function on the space of $m \times n$ matrices is given by*

$$f(X) = \mathrm{Tr}(A^T X) + b = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_{ij} + b,$$

where A is a given (fixed) $m \times n$ matrix and b is a scalar. Affine functions on the space of $m \times n$ matrices are convex (in X).

Example 2 *Spectral norm of a matrix is given by*

$$f(X) = \|X\| = \sqrt{\lambda_{\max}(X^T X)},$$

where $\lambda_{max}(A)$ denotes the maximum eigenvalue of a matrix A (see Section 1.1.10). Spectral norm is convex function.

A convex function f satisfies *general convex inequality*

$$f(\alpha_1 x_1 + \ldots + \alpha_m x_m) \leq \alpha_1 f(x_1) + \ldots + \alpha_m f(x_m)$$

for any integer $m \geq 2$, and for all vectors$x_1, \ldots, x_m \in \mathrm{dom} f$ and scalars $\alpha_1, \ldots, a_m$ such that $\alpha_i > 0$ for all i and $\sum_{i=1}^{m} \alpha_i = 1$.

Convex functions have a special continuity property. In particular, a continuity over some subset of $\mathrm{dom} f$ follows immediately from convexity. We state one such result without proof. A more general result (with the proof) can be found, for example, in Bertsekas, Nedić and Ozdaglar [9], Proposition 1.4.6. This result uses a notion of relative interior of a set, which is the intersection of the interior of the set and the smallest (in the sense of inclusion) affine set containing the given set.

Theorem 10 *A convex function is continuous over the relative interior of its domain.*

As an immediate consequence of Theorem 10, any function f with domain $\mathrm{dom} f = \mathbb{R}^n$ is continuous (over $\mathbb{R}^n$). Furthermore, if the interior of the domain $\mathrm{dom} f$ is nonempty, by Theorem 10, it follows that f is continuous over $\mathrm{int}(\mathrm{dom} f)$. For example, consider the logarithmic function $f(x) = \ln x$ whose domain is $\mathrm{dom} f = (0, +\infty)$. The interior of its domain is the domain itself [since $(0, +\infty)$ is open in $\mathbb{R}$]. Hence, $f(x) = \ln x$ is continuous over $(0, +\infty)$.

Now, we consider the level sets of a convex function. Recall that for a function f and a scalar γ, the (lower) level set $L_\gamma(f)$ is given by

$$L_\gamma(f) = \{x \mid f(x) \leq \gamma\}.$$

From Definition 2 of a convex function it is straightforward to verify that *every (lower) level set of a convex function f is convex.* (recall that, by convention, the empty set is convex). However, the reverse statement is false, i.e., if every (lower) level set of f is convex, the function f need not be convex. Consider, for example, the function $f(x) = -e^x$ for $x \in \mathbb{R}$. *For a concave function g, every (upper) level set $\{x \mid g(x) \geq \gamma\}$ is convex.*

We can verify that a given function is convex by

- Using the definition
- Considering reduction to a scalar function
- Applying some special criteria, such as the second-order or the first order conditions
- Showing that the function is obtained from some other (easily recognizable) convex functions through operations preserving convexity.

We next discuss the reduction to a scalar function. The convexity criteria applicable to differentiable functions are discussed in Section 2.2.1, while the operations preserving the convexity of functions are discussed in Section 2.2.2.

The following lemma provides a convexity test for functions based on the convexity along lines. We state the result without a proof (the result follows straightforward from the definition of a convex function).

Lemma 11 *A function f is convex if and only if* dom$f$ *is convex, and for each $x \in$ dom$f$ and $v \in \mathbb{R}^n$, the function $g_{x,v} : \mathbb{R} \mapsto \mathbb{R}$ given by*

$$g_{x,v}(t) = f(x + tv) \qquad with \quad \text{dom}g_{x,v} = \{t \in \mathbb{R} \mid x + tv \in \text{dom}f\}$$

is convex (in t).

In view of Lemma 11, *checking convexity of multivariable functions can be done by checking convexity of a family of scalar functions.*

The following example demonstrates an application of Lemma 11.

Example 3 *Let $\mathcal{S}^n$ be the space of all $n \times n$ symmetric matrices, and let $f : \mathcal{S}^n \to \mathbb{R}$ be a function defined by*

$$f(X) = -\ln \det X.$$

The domain of f is dom$f = \mathcal{S}^n_{++}$*, where $\mathcal{S}^n_{++}$ is the space of all $n \times n$ positive definite (symmetric) matrices. Let $X \in \mathcal{S}^n_{++}$ and $V \in \mathcal{S}^n$, and let*

$$g(t) = -\ln \det(X + tV).$$

By writing $X + tV = X^{1/2}X^{1/2} + X^{1/2}(X^{-1/2}tVX^{-1/2})X^{1/2}$, we have

$$X + tV = X^{1/2}\,(I + tX^{-1/2}VX^{-1/2})X^{1/2}.$$

Therefore

$$\begin{aligned} g(t) &= -\ln \det X - \ln \det(I + tX^{-1/2}VX^{-1/2}) \\ &= -\ln \det X - \sum_{i=1}^{n} \ln(1 + t\lambda_i), \end{aligned}$$

where λ_i are the eigenvalues of $X^{-1/2}VX^{-1/2}$. The last equality in the preceding relation follows from the fact that the determinant of a (square) matrix is equal to the sum of the eigenvalues of a matrix (see Section 1.1.9), and the fact that the eigenvalues of $I + A$ are $1 + \lambda_i$, where λ_i are eigenvalues of A [see Section 1.1.9, Lemma 2(d)]. Since g is convex in t (for any $X \in$ dom$f$ and $V \in \mathcal{S}^n$), by Lemma 11, f is convex in X over $\mathcal{S}^n$.

2.2.1 Differentiable Convex Functions

Let f be twice differentiable at a vector x. Recall that the Hessian $\nabla^2 f(x)$ is a symmetric $n \times n$ matrix whose entries are the second-order partial derivatives of f at x,

$$\left[\nabla^2 f(x)\right]_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j} \qquad \text{for } i, j = 1, \ldots, n$$

(see Section 1.2.5). The following theorem provides a necessary and sufficient second-order condition for convexity of a function.

Theorem 11 *Let f be a twice differentiable function with convex domain* $\mathrm{dom} f$. *Then:*

(a) f is convex over $\mathrm{dom} f$ *if and only if $\nabla^2 f(x) \geq 0$ for all $x \in \mathrm{dom} f$.*

(b) f is strictly convex over $\mathrm{dom} f$ *if $\nabla^2 f(x) > 0$ for all $x \in \mathrm{dom} f$.*

The following are some examples of the use of Theorem 11.

Example 4 *(Quadratic function)*
Let $f(x) = x'Px + q'x + r$ for an $n \times n$ matrix P. Then,

$$\nabla f(x) = (P + P^T)x + q, \qquad \nabla^2 f(x) = P + P^T.$$

Thus, by Theorem 11, f is convex if and only if $P + P^T$ is positive semidefinite. When in addition P is symmetric, f is convex if and only if P is positive semidefinite.

Example 5 *(Least-squares)*
Let $f(x) = \|Ax - b\|^2$ for an $m \times n$ matrix A. Then,

$$\nabla f(x) = 2A^T(Ax - b), \qquad \nabla^2 f(x) = 2A^T A.$$

Since $A^T A$ is positive semidefinite for any A, by Theorem 11, the function $f(x) = \|Ax - b\|^2$ is always convex.

Example 6 *(Quadratic-over-linear, see [13], page73.)*
Let $f(x, y) = x^2/y$ with $x, y \in \mathbb{R}$. Then,

$$\nabla^2 f(x, y) = \frac{2}{y^3} \begin{bmatrix} y \\ -x \end{bmatrix} \begin{bmatrix} y \\ -x \end{bmatrix}^T.$$

The Hessian $\nabla^2 f(x, y)$ is positive semidefinite if and only if $y > 0$. Thus, $f(x, y) = x^2/y$ is convex over the set $\{(x, y) \in \mathbb{R}^2 \mid x \in \mathbb{R},\ y > 0\}$.

The following theorem provides the first-order condition.

Theorem 12 *Let f be differentiable function over its domain. Then, f is convex if and only if its domain is convex and*

$$f(x) + \nabla f(x)^T(z - x) \leq f(z) \qquad \text{for all } x, z \in \mathrm{dom} f.$$

The result of Theorem 12 has far reaching consequences. In particular, it implies that for a convex function, *a first order approximation provides a global underestimate of f*. This is a very important property frequently used in the development of optimization algorithms for convex problems as well as in the performance analysis of these algorithms.

2.2.2 Operations Preserving Convexity of Functions

Here, we discuss some operations with convex functions that preserve the convexity. In particular, we consider scaling, sum, pointwise maximum/supremum, partial minimization, and special compositions.

Theorem 13 *We have:*

(a) Let f be a convex function and let $\lambda > 0$ be a scalar. Then, λf is convex.

(b) Let f_1 and f_2 be convex functions over their respective domains $\mathrm{dom} f_1$ and $\mathrm{dom} f_2$, and such that $\mathrm{dom} f_1 \cap \mathrm{dom} f_2 \neq \emptyset$. Then, $f_1 + f_2$ is convex over $\mathrm{dom} f_1 \cap \mathrm{dom} f_2$.

(c) Let $\mathcal{A} \subseteq \mathbb{R}^p$ and $f : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$. Let $f(x, z)$ be convex in x for each $z \in \mathcal{A}$. Then, the supremum function over the set $\mathcal{A}$ is convex, i.e., the function

$$g(x) = \sup_{z \in \mathcal{A}} f(x, z)$$

is convex.

(d) Let $C \subseteq \mathbb{R}^n \times \mathbb{R}^p$ be a nonempty convex set, and let $f : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$ be a convex function [in $(x, z) \in \mathbb{R}^n \times \mathbb{R}^p$]. Then, the function $g(x)$ obtained by the partial minimization is convex, i.e., the function

$$g(x) = \inf_{z \in C} f(x, z)$$

is convex.

Proof. (a) Let $\lambda > 0$, and note that f and λf have the same domain $\mathrm{dom} f$, which is convex since f is convex. Let $x_1, x_2 \in \mathrm{dom} f$ and $\alpha \in (0, 1)$. Evidently, by convexity of f, we have $\lambda f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha \lambda f(x_1) + (1-\alpha)\lambda f(x_2)$, showing that λf is convex.

(b) Note that $\mathrm{dom} f_1 \cap \mathrm{dom} f_2$ is convex since it is the intersection of two convex sets [see Theorem 8(c)]. Let $x_1, x_2 \in \mathrm{dom} f_1 \cap \mathrm{dom} f_2$ and $\alpha \in (0, 1)$. Then,

$$\begin{aligned} (f_1 + f_2)(\alpha x_1 + (1-\alpha)x_2) &= f_1(\alpha x_1 + (1-\alpha)x_2) + f_2(\alpha x_1 + (1-\alpha)x_2) \\ &\leq \alpha f_1(x_1) + (1-\alpha) f_1(x_2) + \alpha f_2(x_1) + (1-\alpha) f_2(x_2) \\ &= \alpha (f_1 + f_2)(x_1) + (1-\alpha)(f_1 + f_2)(x_2), \end{aligned}$$

where the inequality follows from convexity of f_1 and f_2. The preceding relation shows that $f_1 + f_2$ is convex over $\mathrm{dom} f_1 \cap \mathrm{dom} f_2$.

(c) Let $x_1, x_2 \in \mathbb{R}^n$ and $\alpha \in (0, 1)$. Consider the point $\alpha x + (1-\alpha)x_2$ and the function value

$$g(\alpha x_1 + (1-\alpha)x_2) = \sup_{z \in \mathcal{A}} f(\alpha x_1 + (1-\alpha)x_2, z). \tag{2.1}$$

Let $\epsilon > 0$ be arbitrary. By definition of the supremum, for $\epsilon > 0$, there exists z_ϵ such that

$$\sup_{z \in \mathcal{A}} f(\alpha x_1 + (1-\alpha)x_2, z) - \epsilon \leq f(\alpha x_1 + (1-\alpha)x_2, z_\epsilon). \tag{2.2}$$

For the vector z_ϵ, the function $f(x, z_\epsilon)$ is convex in x, so that

$$f(\alpha x_1 + (1-\alpha)x_2, z_\epsilon) \leq \alpha f(x_1, z_\epsilon) + (1-\alpha) f(x_2, z_\epsilon).$$

Since $f(x_1, z_\epsilon) \leq \sup_{z\in\mathcal{A}} f(x_1, z)$ and $f(x_2, z_\epsilon) \leq \sup_{z\in\mathcal{A}} f(x_2, z)$, from the preceding relation and the definition of $g(x)$, it follows

$$f(\alpha x_1 + (1-\alpha)x_2, z_\epsilon) \leq \alpha g(x_1) + (1-\alpha) g(x_2). \tag{2.3}$$

By combining Eqs. (2.1)–(2.3), we obtain

$$g(\alpha x_1 + (1-\alpha)x_2) - \epsilon \leq \alpha g(x_1) + (1-\alpha) g(x_2).$$

Letting $\epsilon \to 0$, we see that

$$g(\alpha x_1 + (1-\alpha)x_2) \leq \alpha g(x_1) + (1-\alpha) g(x_2),$$

thus establishing the convexity of g.

(d) Let $x_1, x_2 \in \mathbb{R}^n$ and $\alpha \in (0,1)$ be arbitrary. Let $\epsilon > 0$ be arbitrarily small. Then, there exist $z_1, z_2 \in C$ such that $f(x_1, z_1) \leq g(x_1) + \epsilon$ and $f(x_2, z_2) \leq g(x_2) + \epsilon$. Consider $f(\alpha x_1 + (1-\alpha)x_2, \alpha z_1 + (1-\alpha)z_2)$. Since C is convex, the vector $\alpha z_1 + (1-\alpha)z_2)$ belongs to the set C. Therefore,

$$g(\alpha x_1 + (1-\alpha)x_2) = \inf_{z\in C} f(\alpha x_1 + (1-\alpha)x_2, z) \leq f(\alpha x_1 + (1-\alpha)x_2, \alpha z_1 + (1-\alpha)z_2). \tag{2.4}$$

Note that we can write

$$(\alpha x_1 + (1-\alpha)x_2, \alpha z_1 + (1-\alpha)z_2) = \alpha(x_1, z_1) + (1-\alpha)(x_2, z_2),$$

so that by convexity of f, we obtain

$$\begin{aligned} f(\alpha x_1 + (1-\alpha)x_2, \alpha z_1 + (1-\alpha)z_2) &= f\left(\alpha(x_1, z_1) + (1-\alpha)(x_2, z_2)\right) \\ &\leq \alpha f(x_1, z_1) + (1-\alpha) f(x_2, z_2). \end{aligned}$$

By the choice of $z_1, z_2 \in C$, we further have

$$f(\alpha x_1 + (1-\alpha)x_2, \alpha z_1 + (1-\alpha)z_2) \leq \alpha g(x_1) + (1-\alpha) g(x_2) + \epsilon,$$

which when combined with Eq. (2.4) yields

$$g(\alpha x_1 + (1-\alpha)x_2) \leq \alpha g(x_1) + (1-\alpha) g(x_2) + \epsilon.$$

Letting $\epsilon \to 0$, we see that g satisfies the convexity relation and therefore, g is convex. ■

As a special case of Theorem 13(c), when $\mathcal{A}$ is a finite set, say $\mathcal{A} = \{1, \ldots, m\}$, we see that *the pointwise maximum of a finite collection of convex functions is a convex function*, i.e., the function

$$f(x) = \max\{f_1(x), \ldots, f_m(x)\}$$

is convex when each f_i is convex. Its domain is the intersection of domains of f_i, i.e., $\mathrm{dom} f = \cap_{i=1}^m \mathrm{dom} f_i$. For example, *a polyhedral function f given by*

$$f(x) = \max\{a_1^T x + b_1, \ldots, a_m^T x + b_m\}$$

is convex. Another example is *the sum of r largest components of a vector $x \in \mathbb{R}^n$*, as discussed next.

Example 7 *(Example 3.6 of [13], page 80.) The function defined as the sum of r largest components of a vector $x \in \mathbb{R}^n$ is convex. In particular, this sum is given by*

$$f(x) = \max_{(i_1,\ldots,i_r)\in\mathcal{I}_r} \{x_{i_1} + \ldots + x_{i_r}\},$$

where $\mathcal{I}_r$ is the set of ordered r-tuples with components from $\{1,\ldots,n\}$, defined as follows:

$$\mathcal{I}_r = \{(i_1,\ldots,i_r) \mid i_1 < \ldots < i_r,\ i_j \in \{1,\ldots,n\} \ \text{ for } j = 1,\ldots,r\}.$$

By applying Theorem 13(c), with $\mathcal{A} = \mathcal{I}_r$, we see that f is convex.

We next discuss some applications of Theorem 13(c), where $\mathcal{A}$ is infinite. Given a nonempty set $C \subseteq \mathbb{R}^n$, *the set support function $f_C(x) = \sup_{z\in C} z^T x$ is convex.* Note that, for each fixed z, the function $x \mapsto z^T x$ is convex (in fact, it is linear). Thus, f_C is convex by Theorem 13(c), where $\mathcal{A} = C$. In addition, *set farthest-distance is also convex*, i.e., the function

$$f(x) = \sup_{z\in C} \|x - z\| \qquad \text{for } x \in \mathbb{R}^n$$

is convex. This follows from Theorem 13(c), by noting that for each fixed z, the function $x \mapsto \|x - z\|$ is convex.

Maximum-eigenvalue function over symmetric matrices is convex. In particular, let $\mathcal{S}^n$ be the set of all $n \times n$ symmetric matrices. The maximum-eigenvalue function $\lambda_{\max}(X)$ of a matrix $X \in \mathcal{S}^n$ is

$$\lambda_{\max}(X) = \sup_{\|z\|=1} z^T X z.$$

For a fixed $z \in \mathbb{R}^n$, the function $X \mapsto z^T X z$ is convex (in fact, linear). Hence, by Theorem 13(c) where $C = \{z \in \mathbb{R}^n \mid \|z\| = 1\}$, it follows that $\lambda_{\max}(X)$ is convex.

In the following theorems, we provide some special compositions involving convex functions that preserve convexity. The theorems are given without proofs. One can verify the results of the theorems directly by using the definition of a convex function [Definition 2].

Theorem 14 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex, and let A be an $n \times m$ matrix and $b \in \mathbb{R}^n$. Then,*

$$g(x) = f(Ax + b) \qquad \textit{for } x \in \mathbb{R}^n$$

is a convex function.

As a direct consequence of Theorem 14, the following two functions are convex

$$g(x) = \|Ax + b\| \qquad \text{for } x \in \mathbb{R}^n,$$

$$f(x) = -\sum_{i=1}^{m} \ln(b_i - a_i^T x) \qquad \textit{Log-barrier function},$$

where the domain of the log-barrier function is $\mathrm{dom} f = \{x \in \mathbb{R}^n \mid a_i^T x < b_i,\ i = 1,\ldots,m\}$.

Theorem 15 *Let $f : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R} \to \mathbb{R}$. Consider*

$$g(x) = h(f(x)) \qquad \text{for all } x \in \mathbb{R}^n.$$

The function g is convex if either of the following two conditions is satisfied:

(1) f is convex, h is nondecreasing and convex.

(2) f is concave, h is nonincreasing and convex.

By applying Theorem 15, we can see that:

$$e^{f(x)} \qquad \text{is convex if } f \text{ is convex,}$$

$$\frac{1}{f(x)} \qquad \text{is convex if } f \text{ is concave and } f(x) > 0 \text{ for all } x.$$

2.3 Convex Constrained Optimization Problems

In this section, we consider a generic convex constrained optimization problem. We introduce the basic terminology, and study the existence of solutions and the optimality conditions. We conclude this section with the projection problem and projection theorem. which is important for the subsequent algorithmic development

2.3.1 Constrained Problem

Consider the following constrained optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_1(x) \le 0, \ldots, g_m(x) \le 0 \\ & Bx = d \\ & x \in X, \end{array} \tag{2.5}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is an objective function, $X \subseteq \mathbb{R}^n$ is a given set, $g_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$ are constraint functions, B is a $p \times n$ matrix, and $d \in \mathbb{R}^p$. Let $g(x) \le 0$ compactly denote the inequalities $g_1(x) \le 0, \ldots, g_m(x) \le 0$. Define

$$C = \{x \in \mathbb{R}^n \mid g(x) \le 0,\ Bx = d,\ x \in X\}. \tag{2.6}$$

We refer to the set C as *constraint set or feasible set.* The problem is *feasible* when C is nonempty. We refer to the value $\inf_{x \in C} f(x)$ as the *optimal value* and denote it by f^*, i.e.,

$$f^* = \inf_{x \in C} f(x),$$

where C is given by Eq. (2.6). A vector x^* is *optimal (solution)* when x^* is *feasible* and attains the optimal value f^*, i.e.,

$$g(x^*) \le 0, \qquad Bx^* = d, \qquad x^* \in X, \qquad f(x^*) = f^*.$$

Before attempting to solve problem (2.5), there are some important questions to be answered, such as:

- Is the problem infeasible, i.e., is C empty?
- Is $f^* = +\infty$?[1].
- Is $f^* = -\infty$?

If the answer is "yes" to any of the preceding questions, then it does not make sense to consider the problem in (2.5) any further. The problem is of interest only when f^* is finite. In this case, the particular instances when the problem has a solution are of interest in many applications.

A feasibility problem is the problem of determining whether the constraint set C in Eq. (2.6) is empty or not. It can be posed as an optimization problem with the objective function $f(x) = 0$ for all $x \in \mathbb{R}^n$. In particular, *a feasibility problem* can be reduced to the following minimization problem:

$$\begin{aligned} &\text{minimize} && 0 && (2.7)\\ &\text{subject to} && g(x) \leq 0,\ Bx = d,\ x \in X. && (2.8) \end{aligned}$$

In many applications, the feasibility problem can be a hard problem on its own. For example, the stability in linear time invariant systems often reduces to the feasibility problem where we want to determine whether there exist positive definite matrices P and Q such that

$$A^T P + PA = -Q.$$

Equivalently, the question is whether the set

$$\{(P, Q) \in S^n_{++} \times S^n_{++} \mid A^T P + PA = -Q\}$$

is nonempty, where S^n_{++} denotes the set of $n \times n$ positive definite matrices.

From now on, we assume that the problem in Eq. (2.5) is feasible, and we focus on the issues of finiteness of the optimal value f^* and the existence of optimal solutions x^*. In particular, for problem (2.5), we use the following assumption.

Assumption 1 *The functions f and $g_i, i = 1, \ldots, m$ are convex over $\mathbb{R}^n$. The set X is closed and convex. The set $C = \{x \in \mathbb{R}^n \mid g(x) \leq 0,\ Bx = d,\ x \in X\}$ is nonempty.*

Under Assumption 1, the functions f and $g_i, i = 1, \ldots, m$ are continuous over $\mathbb{R}^n$ (see Theorem 10).

In what follows,*we denote the set of optimal solutions of problem (2.5) by X^*.*

2.3.2 Existence of Solutions

Here, we provide some results on the existence of solutions. Under Assumption 1, these results are consequences of Theorems 3 and 4 of Section 1.2.4.

[1]This happens in general only when $\text{dom} f \cap C = \emptyset$.

Theorem 16 *Let Assumption 1 hold. In addition, let $X \subseteq \mathbb{R}^n$ be bounded. Then, the optimal set X^* of problem (2.5) is nonempty, compact, and convex.*

Proof. At first, we show that the constraint set C is compact. The set C is the intersection of the level sets of continuous functions g_i and hyperplanes (for $j = 1, \ldots, p$, each set $\{x \in \mathbb{R}^n \mid b_j^T x = d_j\}$ is a hyperplane), which are all closed sets. Therefore, C is closed. Since $C \subseteq X$ and X is bounded (because it is compact), the set C is also bounded. Hence, by Lemma 8 of Section 1.2.2, the set C is compact. The function f is continuous (by convexity over $\mathbb{R}^n$). Hence, by Weierstrass Theorem (Theorem 3), the optimal value f^* of problem (2.5) is finite and its optimal set X^* is nonempty.

The set X^* is closed since it can be represented as the intersection of closed sets:

$$X^* = C \cap \{x \in \mathbb{R}^n \mid f(x) \leq f^*\}.$$

Furthermore, X^* is bounded since $X^* \subseteq C$ and C is bounded. Hence, X^* is compact.

We now show that X^* is convex. Note that C is convex as it is given as the intersection of convex sets. Furthermore, the level set $\{x \in \mathbb{R}^n \mid f(x) \leq f^*\}$ is convex by convexity of f. Hence, X^* is the intersection of two convex sets and, thus, X^* is convex. ■

As seen in the proof of Theorem 16, the set C is closed and convex under Assumption 1. Also, under this assumption, the set X^* is closed and convex but possibly empty. The boundedness of X is the key assumption ensuring nonemptiness and boundedness of X^*.

The following theorem is based on Theorem 4. We provide it without a proof. (The proof can be constructed similar to that of Theorem 16. The only new detail is in part (i), where using the coercivity of f, we show that the level sets of f are bounded.)

Theorem 17 *Let Assumption 1 hold. Furthermore, let any of the following conditions be satisfied:*

(i) The function f is coercive over C.

(ii) For some $\gamma \in \mathbb{R}$, the set $\{x \in C \mid f(x) \leq \gamma\}$ is nonempty and compact.

(iii) The set C is compact.

Then, the optimal set X^* of problem (2.5) is nonempty, compact, and convex.

For a quadratic convex objective and a linear constraint set, the existence of solutions is equivalent to finiteness of the optimal value. Furthermore, the issue of existence of solutions can be resolved by checking a "linear condition", as seen in the following theorem.

Theorem 18 *Consider the problem*

$$\begin{array}{ll} \textit{minimize} & f(x) = x^T P x + c^T x \\ \textit{subject to} & Ax \leq b, \end{array}$$

where P is an $n \times n$ positive semidefinite matrix, $c \in \mathbb{R}^n$, A is an $m \times n$ matrix, and $b \in \mathbb{R}^m$. The following statements are equivalent:

(1) The optimal value f^* is finite.

(2) The optimal set X^* is nonempty.

(3) If $Ay \leq 0$ and $Py = 0$ for some $y \in \mathbb{R}^n$, then $c^T y \geq 0$.

The proof of Theorem 18 requires the notion of recession directions of convex closed sets, which is beyond the scope of these notes. The interested reader can find more discussion on this in Bertsekas, Nedić and Ozdaglar [9] (see there Proposition 2.3.5), or in Auslender and Teboulle [2].

As an immediate consequence of Theorem 18, we can derive the conditions for existence of solutions of linear programming problems.

Corollary 1 *Consider a linear programming problem*

$$\begin{array}{ll} \textit{minimize} & f(x) = c^T x \\ \textit{subject to} & Ax \leq b. \end{array}$$

The following conditions are equivalent for the LP problem:

(1) The optimal value f^* is finite.

(2) The optimal set X^* is nonempty.

(3) If $Ay \leq 0$ for some $y \in \mathbb{R}^n$, then $c^T y \geq 0$.

A linear programming (LP) problem that has a solution, it always has a solution of a specific structure. This specific solution is due to the geometry of the polyhedral constraint set. We describe this specific solution for an LP problem in a *standard form*:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0, \end{array} \tag{2.9}$$

where A is an $m \times n$ matrix and $b \in \mathbb{R}^m$. The feasible set for the preceding LP problem is the polyhedral set $\{x \in \mathbb{R}^n \mid Ax = b,\ x \geq 0\}$.

Definition 5 *We say that x is a basic feasible solution for the LP in Eq. (2.9), when x is feasible and there are n linearly independent constraints among the constraints that x satisfies as equalities.*

The preceding definition actually applies to an LP problem in any form and not necessarily in the standard form. Furthermore, a basic solution of an LP is a vertex (or extreme point) of the (polyhedral) constraint set of the given LP, which are out of the scope of these lecture notes. The interested readers may find more on this, for example, in the textbook on linear optimization by Bertsimas and Tsitsiklis [11].

Note that for a given polyhedral set, there can be only finitely many basic solutions. However, the number of such solutions may be very large. For example, the cube $\{x \in \mathbb{R}^n \mid 0 \leq x_i \leq 1,\ i = 1, \ldots, n\}$ is given by $2n$ inequalities, and it has 2^n basic solutions.

We say that a vector x is *a basic solution* if it satisfies Definition 5 apart from being feasible. Specifically, x is a basic solution for (2.9) if there are n linearly independent constraints among the constraints that x satisfies as equalities. A basic solution x is *degenerate* if more than n constraints are satisfied as equalities at x (active at x). Otherwise, it is *nondegenerate*.

Example 8 *Consider the polyhedral set given by*

$$\begin{array}{ll} \textit{minimize} & x_1 + x_2 + x_3 \leq 2 \\ \textit{subject to} & x_2 + 2x_3 \leq 2 \\ & x_1 \leq 1 \\ & x_3 \leq 1 \\ & x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0. \end{array}$$

The vector $\tilde{x} = (1, 1, 0)$ is a nondegenerate basic feasible solution since there are exactly three linearly independent constraints that are active at $\tilde{x}$, specifically,

$$x_1 + x_2 + x_3 \leq 2, \qquad x_1 \leq 1, \qquad x_3 \geq 0.$$

The vector $\hat{x} = (1, 0, 1)$ is a degenerate feasible solution since there are five constraints active at $\hat{x}$, namely

$$x_1 + x_2 + x_3 \leq 2, \qquad x_2 + 2x_3 \leq 2, \qquad x_1 \leq 1, \qquad x_3 \leq 1, \qquad x_2 \geq 0.$$

Out of these, for example, the last three are linearly independent.

We are now ready to state a fundamental result for linear programming solutions.

Theorem 19 *Consider an LP problem. Assume that its constraint set has at least one basic feasible solution and that the LP has an optimal solution. Then, there exists an optimal solution which is also a basic feasible solution.*

2.3.3 Optimality Conditions

In this section, we deal with a differentiable convex function. We have the following.

Theorem 20 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable convex function, and let $C \subseteq \mathbb{R}^n$ be a nonempty closed convex set. Consider the problem*

$$\begin{array}{ll} \textit{minimize} & f(x) \\ \textit{subject to} & x \in C. \end{array}$$

A vector x^* is optimal for this problem if and only if $x^* \in C$ and

$$\nabla f(x^*)^T (z - x^*) \geq 0 \qquad \textit{for all } z \in C.$$

Proof. Let x^* be optimal. Suppose that for some $\hat{z} \in C$ we have

$$\nabla f(x^*)^T(\hat{z} - x^*) < 0.$$

Since f is continuously differentiable, by the first-order Taylor expansion [Theorem 6(a)], we have for all sufficiently small $\alpha > 0$,

$$f(x^* + \alpha(\hat{z} - x^*)) = f(x^*) + \alpha \nabla f(x^*)^T(\hat{z} - x^*) + o(\alpha) < f(x^*),$$

with $x^* \in C$ and $\hat{z} \in C$. By the convexity of C, we have $x^* + \alpha(\hat{z} - x^*) \in C$. Thus, this vector is feasible and has a smaller objective value than the optimal point x^*, which is a contradiction. Hence, we must have $\nabla f(x^*)^T(z - x^*) \geq 0$ for all $z \in C$.

Suppose now that $x^* \in C$ and

$$\nabla f(x^*)^T(z - x^*) \geq 0 \qquad \text{for all } z \in C. \tag{2.10}$$

By convexity of f [see Theorem 12], we have

$$f(x^*) + \nabla f(x^*)^T(z - x^*) \leq f(z) \qquad \text{for all } z \in C,$$

implying that

$$\nabla f(x^*)^T(z - x^*) \leq f(z) - f(x^*).$$

This and Eq. (2.10) further imply that

$$0 \leq f(z) - f(x^*) \qquad \text{for all } z \in C.$$

Since $x^* \in C$, it follows that x^* is optimal. ■

We next discuss several implications of Theorem 20, by considering some special choices for the set C. Let C be the entire space, i.e., $C = \mathbb{R}^n$. The condition

$$\nabla f(x^*)^T(z - x^*) \geq 0 \qquad \text{for all } z \in C$$

reduces to

$$\nabla f(x^*)^T d \geq 0 \qquad \text{for all } d \in \mathbb{R}^n. \tag{2.11}$$

In turn, this is equivalent to

$$\nabla f(x^*) = 0.$$

Thus, by Theorem 20, *a vector x^* is a minimum of f over $\mathbb{R}^n$ if and only if $\nabla f(x^*) = 0$.*

Let the set C be affine, i.e., the problem of interest is

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b, \end{array} \tag{2.12}$$

where A is an $m \times n$ matrix and $b \in \mathbb{R}^m$. In this case, the condition of Theorem 20 reduces to

$$\nabla f(x^*)^T y \geq 0 \qquad \text{for all } y \in N_A,$$

where N_A is the null space of the matrix A. Thus, the gradient $\nabla f(x^*)$ is orthogonal to the null space N_A. Since the range of A^T is orthogonal to N_A [see Eq. (1.2)], it follows that $\nabla f(x^*)$ belongs to the range of A^T, implying that

$$\nabla f(x^*) + A^T\lambda^* = 0 \qquad \text{for some } \lambda^* \in \mathbb{R}^m.$$

Hence, by Theorem 20, *a vector x^* solves problem (2.12) if and only if $Ax^* = b$ and there exists $\lambda^* \in \mathbb{R}^m$ such that $\nabla f(x^*) + A^T\lambda^* = 0$.* The relation $\nabla f(x^*) + A^T\lambda^* = 0$ is known as *primal optimality condition*. It is related to Lagrangian duality, which is the focus of Section 2.5.

Let C be the nonnegative orthant in $\mathbb{R}^n$, i.e., the problem of interest is

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \geq 0. \end{array} \tag{2.13}$$

For this problem, the condition of Theorem 20 is equivalent to

$$\nabla f(x^*)^T x^* = 0.$$

Therefore, *a vector x^* solves problem (2.13) if and only if $x^* \geq 0$ and $\nabla f(x^*)^T x^* = 0$.* The relation $\nabla f(x^*)^T x^* = 0$ is known as *complementarity condition*, and the terminology comes again from the Lagrangian duality theory.

Let C be a simplex in $\mathbb{R}^n$, i.e., the problem of interest is

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \geq 0,\ \displaystyle\sum_{i=1}^n x_i = a, \end{array} \tag{2.14}$$

where $a > 0$ is a scalar. By Theorem 20, x^* is optimal if and only if

$$\sum_{i=1}^n \frac{\partial f(x^*)}{\partial x_i}\,(x_i - x_i^*) \geq 0 \qquad \text{for all } x_i \geq 0 \text{ with } \textstyle\sum_{i=1}^n x_i = a.$$

Consider an index i with $x_i^* > 0$. Let $j \neq i$ and consider a feasible vector x with $x_i = 0$, $x_j = x_j^* + x_i^*$ and all the other coordinates the same as those of x^*. By using this vector in the preceding relation, we obtain

$$\left(\frac{\partial f(x^*)}{\partial x_j} - \frac{\partial f(x^*)}{\partial x_i}\right) x_i^* \geq 0 \qquad \text{for all } i \text{ such that } x_i^* > 0,$$

or equivalently

$$\frac{\partial f(x^*)}{\partial x_i} \leq \frac{\partial f(x^*)}{\partial x_j} \qquad \text{for all } i \text{ such that } x_i^* > 0. \tag{2.15}$$

Hence, x^* is an optimal solution to problem (2.14) if and only if x^* satisfies relation (2.15).

Let us illustrate the optimality conditions for a simplical constraint set on the problem of optimal routing in a communication network (see [5] and [17]).

Example 9 *(Optimal Routing) Consider a directed graph modeling a data communication network. Let $\mathcal{S}$ be a set of origin-destination pairs, i.e., each $s \in \mathcal{S}$ is an ordered pair (i_s, j_s) of nodes i_s and j_s in the network, with i_s being the origin and j_s being the destination of s. Let y_s be the traffic flow of s (data units/second) i.e., the arrival rate of traffic entering the network at the origin of s and exiting the network at the destination of s. The traffic flow of s is routed through different paths in the network. There is a cost associated with using the links $\mathcal{L}$ of the network, namely, the cost of sending a flow z_{ij} on the link $(i,j) \in \mathcal{L}$ is $f_{ij}(z_{ij})$, where f_{ij} is convex and continuously differentiable. The problem is to decide on paths along which the flow y_s should be routed, so as to minimize the total cost.*

To formalize the problem, we introduce the following notation:

- $\mathcal{P}_s$ is the set of all directed paths from the origin of s to the destination of s.

- x_s is the part of the flow y_s routed through the path p with $p \in \mathcal{P}_s$.

Let x denote a vector of path-flow variables, i.e.,

$$x = \{x_p \mid p \in \mathcal{P}_s,\ s \in \mathcal{S}\}.$$

Then, the routing problem can be casted as the following convex minimization:

$$\begin{aligned} \textit{minimize} \quad & f(x) = \sum_{(i,j)\in\mathcal{L}} f_{ij}\left(\sum_{\{p\,|\,(i,j)\in p\}} x_p\right) \\ \textit{subject to} \quad & \sum_{p\in\mathcal{P}_s} x_p = y_s \qquad \textit{for all } s \in \mathcal{S} \\ & x_p \geq 0 \qquad \textit{for all } p \in \mathcal{P}_s \textit{ and all } s \in \mathcal{S}. \end{aligned}$$

The cost on link (i,j) depends on the total flow through that link, i.e., the sum of all flows x_p along paths p that contain the link (i,j). The problem is convex in variable x, with differentiable objective function and a constraint set given by a Cartesian product of simplices (one simplex per s).

We now consider the optimality conditions for the routing problem. Note that

$$\frac{\partial f(x)}{\partial x_p} = \sum_{(i,j)\in p} f'_{ij}(z_{ij}),$$

with z_{ij} being the total flow on the link (i,j). When $f'_{ij}(z_{ij})$ is viewed as the length of the link (i,j) evaluated at z_{ij}, the partial derivative $\frac{\partial f(x)}{\partial x_p}$ is the length of the path p. By the necessary and sufficient conditions for a simplex [cf. Eq. 2.14], for all $s \in \mathcal{S}$, we have $x_p^* > 0$ when

$$\frac{\partial f(x^*)}{\partial x_p} \leq \frac{\partial f(x^*)}{\partial x_{\tilde{p}}} \qquad \textit{for all } \tilde{p} \in \mathcal{P}_s.$$

This relation means that a set of path-flows is optimal if and only if the flow is positive only on the shortest paths (where link length is measured by the first derivative). It also means that at an optimum x^*, for an $s \in \mathcal{S}$, all the paths $p \in \mathcal{P}_s$ carrying a positive flow $x_p^* > 0$ have the same length (i.e., the traffic y_s is routed through the paths of equal length).

In the absence of convexity, the point $x^* \in C$ satisfying the condition

$$\nabla f(x^*)^T(z - x^*) \geq 0 \qquad \text{for all } z \in C$$

is referred to as *a stationary point*. Such a point may be a local or global minimum of f over C. *A global minimum of f over C* is any solution to the problem of minimizing f over C. *A local minimum of f over C* is a point $\tilde{x} \in C$ for which there exists a ball $B(\tilde{x}, r)$ such that there is no "better" point among the points that belong to the ball $B(\tilde{x}, r)$ and the set C, i.e., a ball $B(\tilde{x}, r)$ such that

$$f(x) \geq f(\tilde{x}) \qquad \text{for all } x \in C \cap B(\tilde{x}, r).$$

For convex problems (i.e., convex f and C), there is no distinction between local and global minima: *every local minimum is also global in convex problems*. This makes solving convex minimization problems "easier" than solving a more general "nonconvex" problems.

Let us note that for a strictly convex function, the optimal solution to the problem of minimizing f over C is unique (of course, when a solution exists). We state this result in the following theorem, whose proof follows from the definition of strict convexity.

Theorem 21 *Let $C \subseteq \mathbb{R}^n$ be a nonempty closed convex set and let f be a strictly convex function over C. If the problem of minimizing f over C has a solution, then the solution is unique.*

Proof. To arrive at a contradiction, assume that the optimal set X^* has more than one point. Let x_1^* and x_2^* be two distinct solutions, i.e., $f(x_1^*) = f(x_2^*) = f^*$ and $x_1^* \neq x_2^*$. Also, let $\alpha \in (0, 1)$. Since f is convex, the set X^* is also convex implying that $\alpha x_1^* + (1 - \alpha)x_2^* \in X^*$. Hence,

$$f(\alpha x_1^* + (1 - \alpha)x_2^*) = f^*. \tag{2.16}$$

At the same time, by strict convexity of f over C and the relation $X^* \subseteq C$, we have that f is strictly convex over X^*, so that

$$f(\alpha x_1^* + (1 - \alpha)x_2^*) < \alpha f(x_1^*) + (1 - \alpha)f(x_2^*) = f^*,$$

which contradicts relation (2.16). Therefore, the solution must be unique. ■

2.3.4 Projection Theorem

One special consequence of Theorems 17 and 20 is the Projection Theorem. The theorem guarantees the existence and uniqueness of the projection of a vector on a closed convex set. This result has a wide range of applications.

For a given nonempty set $C \subseteq \mathbb{R}^n$ and a vector $\hat{x}$, the projection problem is the problem of determining the point $x^* \in C$ that is the closest to $\hat{x}$ among all $x \in C$ (with respect to the Euclidean distance). Formally, the problem is given by

$$\begin{array}{ll} \text{minimize} & \|x - \hat{x}\|^2 \\ \text{subject to} & x \in C. \end{array} \tag{2.17}$$

In general, such a problem may not have an optimal solution and the solution need not be unique (when it exists). However, when the set C is closed and convex set, the solution exists and it is unique, as seen in the following theorem.

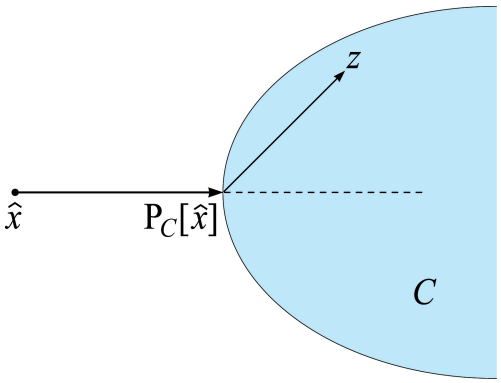

Figure 2.9: The projection of a vector $\hat{x}$ on the closed convex set C is the vector $P_C[\hat{x}] \in C$ that is the closest to $\hat{x}$ among all $x \in C$, with respect to the Euclidean distance.

Theorem 22 *(Projection Theorem) Let $C \subseteq \mathbb{R}^n$ be a nonempty closed convex set and $\hat{x} \in \mathbb{R}^n$ be a given arbitrary vector.*

(a) The projection problem in Eq. (2.17) has a unique solution.

(b) A vector $x^* \in C$ is the solution to the projection problem if and only if

$$(x^* - \hat{x})^T (x - x^*) \geq 0 \qquad \text{for all } x \in C.$$

Proof. (a) The function $f(x) = \|x - \hat{x}\|^2$ is coercive over $\mathbb{R}^n$, and therefore coercive over C (i.e., $\lim_{\|x\| \to \infty,\ x \in C} f(x) = \infty$). The set C is closed, and therefore by Theorem 17(i), the optimal set X^* for projection problem (2.17) is nonempty.

Furthermore, the Hessian of f is given by $\nabla^2 f(x) = 2I$, which is positively definite everywhere. Therefore, by Theorem 11(b), the function f is strictly convex and the optimal solution is unique [cf. Theorem 21].

(b) By the first-order optimality condition of Theorem 20, we have $x^* \in C$ is a solution to the projection problem if and only if

$$\nabla f(x^*)^T (x - x^*) \geq 0 \qquad \text{for all } x \in C.$$

Since $\nabla f(x) = 2(x - \hat{x})$, the result follows. ■

The projection of a vector $\hat{x}$ to a closed convex set C is illustrated in Figure 2.9. The unique solution x^* to the projection problem is referred to as *the projection of $\hat{x}$ on C*, and it is denoted by $P_C[\hat{x}]$. The projection $P_C[\hat{x}]$ has some special properties, as given in the following theorem.

Theorem 23 *Let $C \subseteq \mathbb{R}^n$ be a nonempty closed convex set.*

(a) The projection mapping $P_C : \mathbb{R}^n \to C$ is nonexpansive, i.e.,

$$\|P_C[x] - P_C[y]\| \leq \|x - y\| \qquad \text{for all } x, y \in \mathbb{R}^n.$$

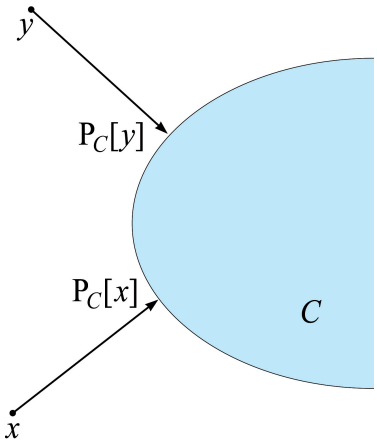

Figure 2.10: The projection mapping $x \mapsto P_C[x]$ is nonexpansive, $\|P_C[x]-P_C[y]\| \leq \|x-y\|$ for all x, y.

(b) The set distance function $d : \mathbb{R}^n \to \mathbb{R}$ given by

$$dist(x, C) = \|P_C[x] - x\|$$

is convex.

Proof. (a) The relation evidently holds for any x and y with $P_C[x] = P_C[y]$. Consider now arbitrary $x, y \in \mathbb{R}^n$ with $P_C[x] \neq P_C[y]$. By Projection Theorem 22(b), we have

$$(P_C[x] - x)^T(z - P_C[x]) \geq 0 \qquad \text{for all } z \in C, \tag{2.18}$$

$$(P_C[y] - y)^T(z - P_C[y]) \geq 0 \qquad \text{for all } z \in C. \tag{2.19}$$

Using $z = P_C[y]$ in Eq. (2.18) and $z = P_C[x]$ in Eq. (2.19), and by summing the resulting inequalities, we obtain,

$$(P_C[y] - y + x - P_C[x])^T(P_C[x] - P_C[y]) \geq 0.$$

Consequently,

$$(x - y)^T(P_C[x] - P_C[y]) \geq \|P_C[x] - P_C[y]\|^2.$$

Since $P_C[x] \neq P_C[y]$, it follows that $\|y - x\| \geq \|P_C[x] - P_C[y]\|$.

(b) The distance function is equivalently given by

$$dist(x, C) = \min_{z \in C} \|x - z\| \qquad \text{for all } x \in \mathbb{R}^n.$$

The function $h(x, z) = \|x - z\|$ is convex in (x, z) over $\mathbb{R}^n \times \mathbb{R}^n$, and the set C is convex. Hence, by Theorem 13(d), the function $dist(x, C)$ is convex. ■

2.4 Problem Reformulation

Often, a convex minimization problem can be re-formulated in an equivalent form that may have certain advantages from the point of solution approaches. In this section, we provide several examples illustrating some of the useful problem re-formulations.

Consider the problem with *a polyhedral objective*, given as follows:

$$\begin{array}{ll} \text{minimize} & \max\{c_1^T x + d_1, \ldots, c_m^T x + d_m\} \\ \text{subject to} & Ax \leq b, \end{array}$$

where $c_j \in \mathbb{R}^n$, A is an $m \times n$ matrix, and $b \in \mathbb{R}^m$. The problem is convex, but the objective function is not differentiable. However, the problem can be casted as a linear program by augmenting the space of variables. In particular, it is equivalent to the following linear programming problem

$$\begin{array}{ll} \text{minimize} & w \\ \text{subject to} & c_j^T x + d_j \leq w, \quad j = 1, \ldots, m \\ & Ax \leq b, \end{array}$$

where the minimization is taken over the variable (x, w) with $x \in \mathbb{R}^n$ and $w \in \mathbb{R}$. The preceding problem is linear in (x, w).

The following problem is a *linear-fractional programming problem*

$$\begin{array}{ll} \text{minimize} & \dfrac{c^T x + d}{g^T x + h} \\ \text{subject to} & Ax \leq a,\ Bx = b. \end{array}$$

There is another (implicit) constraint in this problem imposed by the domain of the objective f:

$$\text{dom} f = \{x \in \mathbb{R}^n \mid g^T x + h > 0\}.$$

By introducing the change of variables $x = \frac{y}{z}$ with $y \in \mathbb{R}^n$, $z \in \mathbb{R}$ and $z \geq 0$, the problem can be re-formulated as a linear problem

$$\begin{array}{ll} \text{minimize} & c^T y + dz \\ \text{subject to} & Ay \leq za,\ By = zb \\ & g^T y + hz = 1,\ z \geq 0. \end{array}$$

Note that, the preceding problem is linear in variable (y, z) where $y \in \mathbb{R}^n$ and $z \in \mathbb{R}$.

Example 10 *(Example 4.7 of [13], page 152.) Von Neumann Model of a growing economy is the following:*

$$\begin{array}{ll} \textit{maximize} & f(x, x^+) = \min\limits_{1 \leq i \leq n} \dfrac{x_i^+}{x_i} \\ \textit{subject to} & x^+ \geq 0,\ x > 0,\ Bx^+ \leq Ax, \end{array}$$

where

- *x_i and x_i^+ represent the activity levels of sector i, in the current and next period respectively.*

- *$[Ax]_j$ and $[Bx^+]_j$ are respectively the produced and consumed amounts of good j.*

- *x_i^+/x_i is the growth rate of sector i.*

The problem is to allocate activity levels x and x^+ so as to maximize the growth rate of the slowest growing sector. This problem is a linear-fractional programming problem.

A semidefinite programming problem (SDP) is a convex problem of the following form:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & x_1 F_1 + x_2 F_2 + \cdots + x_n F_n + G \leq 0 \\ & Ax = b, \end{array} \tag{2.20}$$

with $x \in \mathbb{R}^n$, symmetric $m \times m$ matrices F_i and G, a $p \times n$ matrix A, and a vector $b \in \mathbb{R}^p$. The inequality constraint is referred to as a *linear matrix inequality* (LMI). An SDP problem with multiple LMI constraints can be re-written as an SDP with a single LMI constraint by enlarging the matrices. For example, the system with two LMIs

$$x_1 \hat{F}_1 + \cdots + x_n \hat{F}_n + \hat{G} \leq 0, \qquad x_1 \tilde{F}_1 + \cdots + x_n \tilde{F}_n + \tilde{G} \leq 0$$

is equivalent to the following single LMI

$$x_1 \begin{bmatrix} \hat{F}_1 & 0 \\ 0 & \tilde{F}_1 \end{bmatrix} + x_2 \begin{bmatrix} \hat{F}_2 & 0 \\ 0 & \tilde{F}_2 \end{bmatrix} + \cdots + x_n \begin{bmatrix} \hat{F}_n & 0 \\ 0 & \tilde{F}_n \end{bmatrix} + \begin{bmatrix} \hat{G} & 0 \\ 0 & \tilde{G} \end{bmatrix} \leq 0.$$

The SDP problem in Eq. (2.20) reduces to an LP problem when all matrices F_i and G are diagonal. An LP problem is equivalent to an SDP problem. In particular, consider the following LP problem:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \leq b. \end{array}$$

Its equivalent SDP problem is:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \operatorname{diag}(Ax - b) \leq 0, \end{array}$$

where

$$\operatorname{diag}(Ax - b) = \begin{bmatrix} [Ax-b]_1 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & [Ax-b]_m \end{bmatrix}.$$

The following is an example of an SDP problem.

Example 11 *(Matrix Norm Minimization example of [13], Section 4.6.3, page 169.) Consider the problem*

$$\begin{array}{ll} \textit{minimize} & \|A(x)\| \\ \textit{subject to} & x \in \mathbb{R}^n, \end{array}$$

where

$$A(x) = A_0 + x_1 A_1 + \cdots + x_n A_n$$

with each A_i being an $p \times q$ matrix. The matrix norm $\|A\|$ is the norm induced by the Euclidean vector norm, i.e.,

$$\|A\| = \sqrt{\lambda_{\max}(A^T A)},$$

and $\lambda_{\max}(A^T A)$ is the maximum eigenvalue of $A^T A$.

The preceding matrix norm minimization problem is equivalent to the following problem:

$$\begin{array}{ll} \textit{minimize} & t \\ \textit{subject to} & \|A(x)\| \leq t \\ & x \in \mathbb{R}^n. \end{array}$$

Since the relation $\|A\| \leq t$ is equivalent to $A^T A \leq t^2 I$, and this in turn is equivalent to

$$\begin{bmatrix} tI & A \\ A^T & tI \end{bmatrix} \geq 0,$$

the matrix norm minimization is equivalent to the following SDP problem

$$\begin{array}{ll} \textit{minimize} & t \\ \textit{subject to} & \begin{bmatrix} tI & A(x) \\ A(x)^T & tI \end{bmatrix} \geq 0 \\ & x \in \mathbb{R}^n, \quad t \in \mathbb{R}. \end{array}$$

2.5 Lagrangian Duality

Lagrangian duality is an important concept in convex optimization, and in optimization theory in general. Its implications are far reaching both in theory and practice. Lagrangian duality is a powerful tool providing:

- A basis for the development and analysis of a rich class of optimization algorithms.
- A general systematic way for generating bounding strategies (both in continuous and discrete optimization).
- A basis for sensitivity analysis.

The main idea in Lagrangian duality is to associate, with a given constrained "primal problem", an "equivalent dual problem", which may be easier to solve than the original problem. This methodology is applicable to general constrained optimization problems, but here, we will dominantly focus on constrained convex problems. The questions that are investigated in the duality theory include:

(1) Is there a general relation between the primal and its associated dual problem?

(2) Do the primal and the dual problems have the same optimal values?

(3) When do the primal and dual optimal solutions exist?

(4) What are the relations between primal and dual optimal solutions?

(5) What kind of information (if any) does a dual optimal solution provide about the primal problem?

In this section, we provide the basic duality framework, and address some of the questions listed above. We start with an abstract geometric primal problem and its dual, and then we show that a constrained optimization problem and its dual can be embedded in the geometric primal-dual setting.

We then consider some important duality results including linear programming and quadratic programming duality. We discuss the well-known Slater condition guaranteeing the strong duality relation as well as the existence of a dual optimal solution. We consider the Karush-Kuhn-Tucker conditions characterizing a primal-dual optimal pair. We also provide several examples illustrating the use of duality and, in particular, the Karush-Kuhn-Tucker conditions.

2.5.1 Geometric Primal and Dual Problems

We illustrate duality using an abstract "geometric framework", which provides a "visualization of duality" and the insights into the weak duality relation for optimal primal and optimal dual values, and the strong duality relation (equality between the optimal primal and dual values).

We start by introducing some terminology. Consider a hyperplane $H \subseteq \mathbb{R}^n$ given by

$$H = \{x \in \mathbb{R}^n \mid a^T x = b\} \qquad \text{for some nonzero } a \in \mathbb{R}^n.$$

We say that *the hyperplane H passes through a vector x_0* when $x_0 \in H$, or equivalently

$$a^T x_0 = b.$$

Given a nonempty set C, we say that *the hyperplane H supports the set C* when $a^t z \leq b$ for all $z \in C$, or equivalently

$$\sup_{x \in C} a^T x \leq b.$$

In this case, we also say that *H is a supporting hyperplane for the set C*. If there is a point $x_0 \in C$ attaining the supremum $\sup_{x \in C} a^T x$ and $a^T x_0 = b$, then we say that *H supports C at the point $x_0 \in C$*. A supporting hyperplane is illustrated in Figure 2.11.

We now define a "geometric primal problem" using an abstract set $V \subseteq \mathbb{R}^m \times \mathbb{R}$ and its corresponding "geometric dual problem" using the hyperplanes supporting the set V.

Consider an abstract (nonempty) set V of vectors $(u, w) \in \mathbb{R}^m \times \mathbb{R}$, which intersects the w-axis, i.e.,

$$(0, w) \in V \quad \text{for some } w \in \mathbb{R}.$$

Let the set V extend "north" and "east", i.e.,

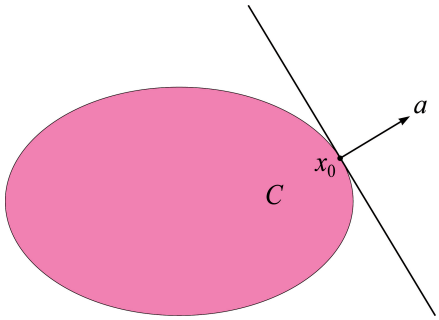

Figure 2.11: A hyperplane supporting the set C at the point x_0.

(a) *North:* For any $(u, w) \in V$ and $u \in \mathbb{R}^m$ with $u \leq \tilde{u}$, we have $(\tilde{u}, w) \in V$.

(b) *East:* For any $(u, w) \in V$ and $w \in \mathbb{R}$ with $w \leq \tilde{w}$, we have $(u, \tilde{w}) \in V$.

Geometric Primal Problem: Determine the minimum intercept of the set V and the w-axis:

$$\begin{array}{ll} \text{minimize} & w \\ \text{subject to} & (0, w) \in V. \end{array}$$

The minimum intercept value is denoted by f^*, i.e., $f^* = \inf_{(0,w)\in V} w$.

Consider the hyperplane $H = \{(u, w) \in \mathbb{R}^m \times \mathbb{R} \mid \mu^T u + \mu_0 w = \xi\}$ where $\mu \in \mathbb{R}^m$, $\mu_0, \xi \in \mathbb{R}$, and $(\mu, \mu_0) \neq (0, 0)$. We say that the hyperplane H is *nonvertical* when $\mu_0 \neq 0$. Let $H_{\mu,\xi}$ denote a nonvertical hyperplane in $\mathbb{R}^m \times \mathbb{R}$, i.e.,

$$H_{\mu,\xi} = \{(u, w) \in \mathbb{R}^m \times \mathbb{R} \mid \mu^T u + w = \xi\},$$

with $\mu \in \mathbb{R}^m$ and $\xi \in \mathbb{R}$. Let $q(\mu)$ be the minimum value of $\mu^T u + w$ for $(u, w) \in V$, i.e.,

$$q(\mu) = \inf_{(u,w)\in V} \{\mu^T u + w\}.$$

A nonvertical hyperplane $H_{\mu,\hat{\xi}}$ *supports* a set V when $\hat{\xi} = q(\mu)$, as illustrated in Figure 2.12. A hyperplane supporting the set V intersects the w-axis at $(0, q(\mu))$ (see Figure 2.12).

Geometric Dual Problem: Determine the maximum intercept with the w-axis for the nonvertical hyperplanes that support the set V:

$$\begin{array}{ll} \text{maximize} & q(\mu) \\ \text{subject to} & \mu \in \mathbb{R}^m. \end{array}$$

Note that $q(\mu) = \inf_{(u,w)\in V}\{\mu^T u + w\}$ takes value $-\infty$ for $\mu \not\geq 0$. The geometric optimal dual value is denoted by q^*. An illustration of the geometric dual problem and its optimal value is given in Figure 2.13.

Figure 2.13 also illustrates a geometric primal optimal value and its relation to the geometric dual value. In particular, one can observe the following:

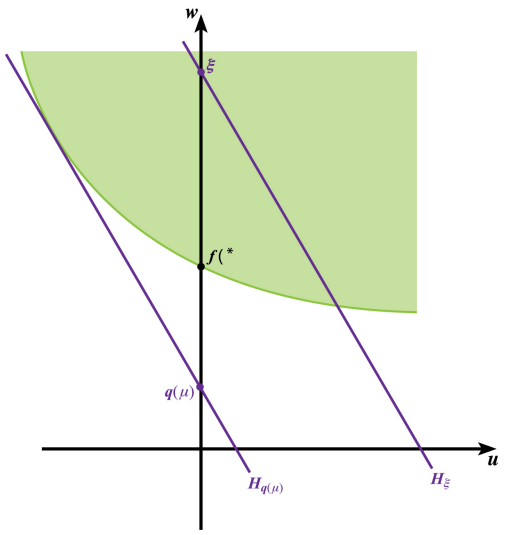

Figure 2.12: A nonvertical hyperplane supporting the shaded set and intersecting the w-axis at the point $(0, q(\mu))$.

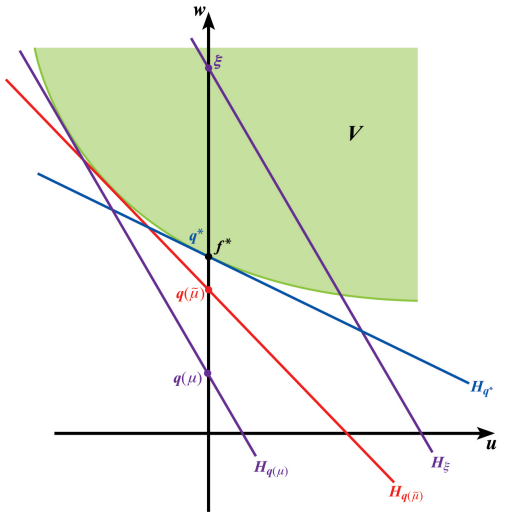

Figure 2.13: Geometric dual problem and its optimal value q^*.

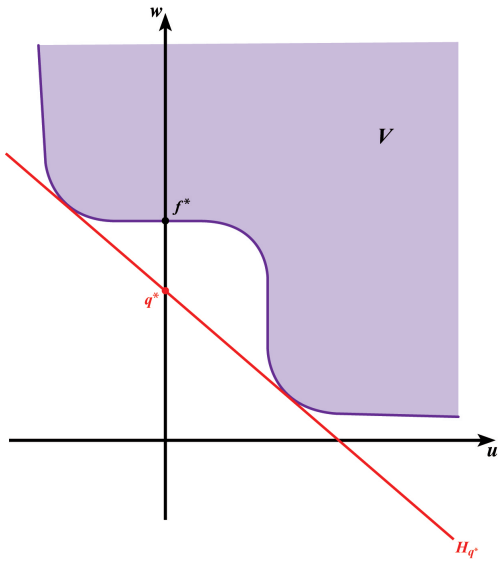

Figure 2.14: A duality gap for a nonconvex set V.

- Dual values $q(\mu)$ are always below f^* and below any w with $(0, w) \in V$,

$$q(\mu) \leq f^* \leq w \qquad \text{for any } \mu \geq 0 \text{ and any } w \text{ such that } (0, w) \in V.$$

- Dual optimal value q^* never exceeds the primal optimal value f^*, which is known as *weak duality* relation, i.e.,

$$q^* \leq f^* \qquad \textit{weak duality.}$$

  This relation *always holds* regardless of the structure of the primal problem (i.e., regardless of the structure of the set V).

The weak duality relation may be strict i.e., $q^* < f^*$, in which case we say that *there is a duality gap*. Some illustrations of duality gaps are provided in Figures 2.14 and 2.15. As seen from Figure 2.15, a duality gap may exist even for a convex set V.

When the primal optimal and dual optimal values are equal, we say that *strong duality* holds,

$$q^* = f^* \qquad \textit{strong duality.}$$

In this case, however, a nonvertical hyperplane achieving the maximum intercept may not exist. In other words, the optimal dual μ^* achieving the dual optimal value q^* may not exist, as shown in Figure 2.16.

2.5.2 Constrained Optimization Duality

In this section, we consider a general constrained optimization problem and derive its dual, by embedding the problem in the geometric setting. Also, we provide some examples of primal-dual problem pairs, problems with duality gaps, and strong duality examples.

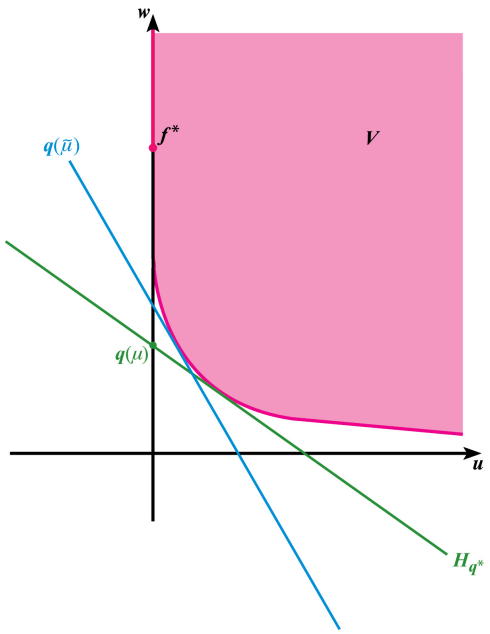

Figure 2.15: A duality gap for a convex set V.

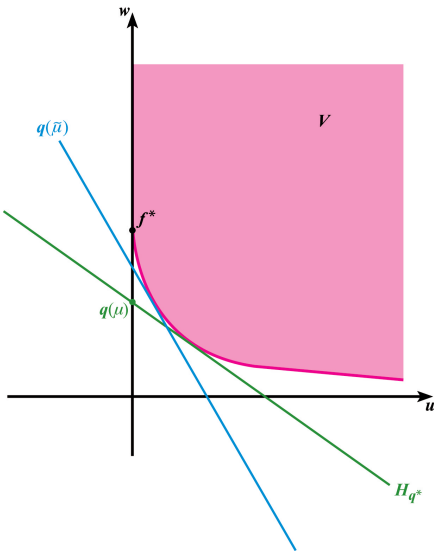

Figure 2.16: A situation where $f^* = q^*$ but a dual optimal solution does not exist.

Consider the following (not necessarily convex) optimization problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_j(x) = 0, \quad j = 1, \ldots, r \\ & x \in X, \end{array} \tag{2.21}$$

where $X \subseteq \mathbb{R}^n$ is a given set, and $f : X \to \mathbb{R}$, $g_i : X \to \mathbb{R}$ and $h_j : X \to \mathbb{R}$ are given functions. We refer to the preceding problem as *primal problem*, and we let f^* denote its optimal value.

We make no assumptions on the set X (may even be discrete) or the functions f, g_i, h_j (may even be discontinuous). We write the inequality constraints $g_i(x) \leq 0, i = 1, \ldots, m$ compactly as $g(x) \leq 0$. Also, write the equality constraints $h_j(x) \leq 0, j = 1, \ldots, r$ compactly as $h(x) = 0$. We embed primal problem (2.21) in the geometric setting by defining the set V as follows:

$$V = \{(u, v, w) \in \mathbb{R}^m \times \mathbb{R}^r \times \mathbb{R} \mid \text{there is } x \in X \text{ such that } g(x) \leq u,\ h(x) = v,\ f(x) \leq w\}.$$

The dual function q is given by

$$q(\mu, \lambda) = \inf_{(u,v,w) \in V} \{w + \mu^T u + \lambda^T v\} = \inf_{x \in X} \{f(x) + \mu^T g(x) + \lambda^T h(x)\} \quad \text{for } \mu \geq 0,\ \lambda \in \mathbb{R}^r.$$

The function appearing under the minimization is *Lagrangian function*, denoted by $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^r \to \mathbb{R}$ and given by

$$\begin{aligned} \mathcal{L}(x, \mu, \lambda) &= f(x) + \mu^T g(x) + \lambda^T h(x) \\ &= f(x) + \sum_{j=1}^{m} \mu_j g_j(x) + \sum_{j=1}^{r} \lambda_j h_j(x). \end{aligned}$$

The infimum above has an implicit constraint on the primal problem domain.

Note that the Lagrangian function is the weighted sum of the objective and the constraint functions. The weights μ and λ of the constraint functions can be viewed as penalties associated respectively with violations of inequality and equality constraints. The vector $\mu \in \mathbb{R}^m$, with $\mu \geq 0$, is *a Lagrange multiplier associated with constraints* $g(x) \leq 0$, where $g = (g_1, \ldots, g_m)$. The vector $\lambda \in \mathbb{R}^r$ is *a Lagrange multiplier associated with* $h(x) = 0$, where $h = (h_1, \ldots, h_r)$. Often, the variable x in the primal problem is referred to as *primal variable*, while the variables μ and λ in the dual problem are referred to as *dual variables* or *prices*.

The problem given by

$$\begin{array}{ll} \text{maximize} & q(\mu, \lambda) \\ \text{subject to} & \mu \geq 0,\ \lambda \in \mathbb{R}^r \end{array} \tag{2.22}$$

is it the dual problem of the problem in Eq. (2.21). We denote its optimal value by q^*.

The dual problem has some important properties that hold without any assumptions on the primal problem, as shown in the next theorem

Theorem 24 *For the primal-dual problem pair of Eqs. (2.21)–(2.22), we have:*

(a) The dual constraint set $\{\mu, \lambda) \in \mathbb{R}^m \times \mathbb{R}^r \mid \mu \geq 0\}$ *is convex, and the dual function* $q(\mu, \lambda)$ *is concave.*

(b) For any $\mu \geq 0$ *and* $\lambda \in \mathbb{R}^r$*, we have*

$$q(\mu, \lambda) \leq f^*.$$

(c) Weak duality holds, i.e., $q^* \leq f^*$.

Proof. (a) The dual constraint set is the Cartesian product of two convex sets, the nonnegative orthant in $\mathbb{R}^m$ and $\mathbb{R}^r$. Hence, it is convex by Theorem 8(d). The concavity of q follows by viewing the dual function as the infimum over $x \in X$ of the affine functions $(\mu, \lambda) \mapsto f(x) + \mu^T g(x) + \lambda^T h(x)$ [see Theorem 13(c)].

(b) Note that

$$q(\mu, \lambda) \leq \inf_{\substack{g(x) \leq 0, \ h(x)=0 \\ x \in X}} \left\{ f(x) + \mu^T g(x) + \lambda^T h(x) \right\}.$$

Furthermore, for a $\mu \geq 0$ and a primal feasible x, i.e., $g(x) \leq 0$, $h(x) = 0$, $x \in X$, we have $\mu^T g(x) + \lambda^T h(x) \leq 0$. Thus, for any $\mu \geq 0$ and $\lambda \in \mathbb{R}^r$,

$$q(\mu, \lambda) \leq \inf_{\substack{g(x) \leq 0, \ h(x)=0 \\ x \in X}} f(x) = f^*.$$

(c) The weak duality relation follows by taking the supremum over $\mu \geq 0$ and $\lambda \in \mathbb{R}^r$ in the preceding relation. ■

The following are some examples of primal-dual optimization problems.

Example 12 *(Least-Norm Solution of Linear Equations) Consider the problem of minimizing the Euclidean norm subject to linear equation constraints, i.e.,*

$$\begin{array}{ll} \textit{minimize} & x^T x \\ \textit{subject to} & Bx = d, \end{array}$$

for an $r \times n$ *matrix* B *and a vector* $d \in \mathbb{R}^r$*. The associated Lagrangian function is*

$$\mathcal{L}(x, \lambda) = x^T x + \lambda^T (Bx - d).$$

For a fixed λ*, to minimize* $\mathcal{L}$ *over* $x \in \mathbb{R}^n$*, we set the gradient* $\nabla_x \mathcal{L}(x, \lambda)$ *equal to zero [cf. Eq (2.11)] and obtain*

$$\nabla_x \mathcal{L}(x, \lambda) = 2x + B^T \lambda = 0,$$

implying that

$$x_\lambda = -\frac{1}{2} B^T \lambda.$$

By substituting x_λ *in* $\mathcal{L}$*, we obtain the dual function value* $q(\lambda)$*,*

$$q(\lambda) = \mathcal{L}(x_\lambda, \lambda) = -\frac{1}{4} \lambda^T B B^T \lambda - d^T \lambda,$$

which is a concave function of λ. *Furthermore, we have*

$$-\frac{1}{4}\lambda^T BB^T\lambda - d^T\lambda \leq f^* \qquad \text{for all } \lambda \in \mathbb{R}^r,$$

where f^* *is the optimal value of the minimum norm (primal) problem, i.e.,* $f^* = \inf_{Bx=d} x^T x$.

Example 13 *Consider the following norm minimization problem:*

$$\text{minimize} \quad \|Ax - b\| \quad \text{over } x \in \mathbb{R}^n.$$

This is an unconstrained problem, which can be reformulated as follows:

$$\begin{array}{ll} \text{minimize} & \|y\| \\ \text{subject to} & y = Ax - b. \end{array}$$

The dual function of the preceding problem is:

$$q(\lambda) = \inf_{x,y\in\mathbb{R}^n} \{\|y\| + \lambda^T y - \lambda^T Ax + b^T\lambda\} = \begin{cases} b^T\lambda + \inf_{y\in\mathbb{R}^n} \left\{\|y\| + \lambda^T y\right\} & \text{for } A^T\lambda = 0 \\ -\infty & \text{otherwise.} \end{cases}$$

Note that

$$\inf_{y\in\mathbb{R}^n} \{\|y\| + \lambda^T y\} = \inf_y \{\|y\| - \lambda^T y\}.$$

$$q(\lambda) = \begin{cases} b^T\lambda & \text{for } A^T\lambda = 0, \quad \|\lambda\| \leq 1 \\ -\infty & \text{otherwise.} \end{cases}$$

The dual of the norm minimization problem is

$$\begin{array}{ll} \text{maximize} & b^T\lambda \\ \text{subject to} & A^T\lambda = 0, \quad \|\lambda\| \leq 1. \end{array}$$

Example 14 *(Standard Form LP) Consider an LP in a standard form*

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0, \end{array}$$

for an $r \times n$ *matrix* A *and a vector* $b \in \mathbb{R}^r$. *Its Lagrangian is given by*

$$\begin{aligned} \mathcal{L}(x,\mu,\lambda) &= c^T x + \lambda^T(Ax - b) - \mu^T x \\ &= -b^T\lambda + (c + A^T\lambda - \mu)^T x. \end{aligned}$$

The Lagrangian $\mathcal{L}(x,\mu,\lambda)$ *is linear in* x *for a fixed* μ *and* λ. *Hence,*

$$q(\mu,\lambda) = \inf_x \mathcal{L}(x,\mu,\lambda) = \begin{cases} -b^T\lambda & \text{when } A^T\lambda - \mu + c = 0, \\ -\infty & \text{otherwise.} \end{cases}$$

Therefore, the domain of q *is the affine set* $\{(\mu,\lambda) \in \mathbb{R}^n \times \mathbb{R}^r \mid A^T\lambda - \mu + c = 0,\ \mu \geq 0\}$. *The dual function* $q(\mu,\lambda)$ *is linear on its domain and hence, concave. Moreover, we have*

$$-b^T\lambda \leq f^* \qquad \text{for } \lambda \text{ and } \mu \geq 0 \text{ such that } A^T\lambda + c - \mu = 0,$$

where $f^* = \inf_{Ax=b,\ x\geq 0} c^T x$. *Note that the preceding relation is equivalent to*

$$-b^T\lambda \leq f^* \qquad \text{for } \lambda \text{ such that } A^T\lambda + c \geq 0.$$

We discuss the LP duality in greater detail later on in Section 2.5.3.

Example 15 *(Two-Way Partitioning Problem in [13], pages 219–221.)*
Consider the following problem

$$\begin{array}{ll} \textit{minimize} & x^T W x \\ \textit{subject to} & x_i^2 = 1, \quad i = 1, \ldots, n, \end{array}$$

where W is an $n \times n$ matrix. This is a nonconvex optimization problem. In particular, it is a discrete optimization problem since the feasible set contains 2^n discrete points. The problem can be interpreted as the partitioning of the index set $\{1, \ldots, n\}$ in two sets, in the presence of a cost W_{ij} associated with assigning i and j to the same set and the cost $-W_{ij}$ when assigning i and j to different sets. The dual function is

$$\begin{aligned} q(\lambda) = \inf_x \left\{ x^T W x + \sum_{i=1}^n \lambda_i (x_i^2 - 1) \right\} &= \inf_x x^T [W + \operatorname{diag}(\lambda)] x - e^T \lambda \\ &= \begin{cases} -e^T \lambda & \textit{when } W + \operatorname{diag}(\lambda) \geq 0, \\ -\infty & \textit{otherwise}, \end{cases} \end{aligned}$$

where $\operatorname{diag}(\lambda)$ *is the diagonal matrix with diagonal entries λ_i. Given any $\lambda \in \mathbb{R}^n$, we have*

$$-e^T \lambda \leq f^* \qquad \textit{if } W + \operatorname{diag}(\lambda) \geq 0,$$

where $f^* = \inf_{x_i^2 = 1,\, i = 1, \ldots, n} x^T W x$. For example, $\lambda = -\lambda_{\min}(W) e$ gives the following lower bound

$$n \lambda_{\min}(W) \leq f^*.$$

As shown earlier [Theorem 24(c)], the weak duality relation $q^* \leq f^*$ always holds. It can be used to compute nontrivial lower bounds for difficult problems. For example, a lower bound for the two-way partitioning problem of Example 15 can be obtained by solving the following SDP:

$$\begin{array}{ll} \text{maximize} & -e^T \lambda \\ \text{subject to} & W + \operatorname{diag}(\lambda) \geq 0. \end{array}$$

The following are some examples of problems with a duality gap ($q^* < f^*$).

Example 16 *Consider a discrete optimization problem of the form:*

$$\begin{array}{ll} \textit{minimize} & -x \\ \textit{subject to} & x \leq 1, \ x \in \{0, 2\}. \end{array}$$

The primal optimal value is $f^* = 0$. the dual function is

$$q(\mu) = \inf_{x \in \{0,2\}} -x + \mu(x - 1) = (\mu - 1)x - \mu = \begin{cases} \mu - 2 & \textit{if } 0 \leq \mu \leq 1, \\ -\mu & \textit{if } \mu \geq 1. \end{cases}$$

The dual optimal value is $q^* = -1$, and we have a duality gap.

Example 17 *Consider a convex optimization problem of the form:*

$$\begin{array}{ll} \textit{minimize} & e^{-\sqrt{x_1 x_2}} \\ \textit{subject to} & x_1^2 \leq 0,\ x_1 \geq 0,\ x_2 \geq 0. \end{array}$$

Here, it can be seen the primal optimal value is $f^* = 1$, while the dual optimal value is $q^* = 0$. Thus, there is a duality gap. Note that the problem is convex.

As indicated with the preceding example, the duality gap may exist even for convex problems. In general, some additional conditions of the objective function f and the constraint set are needed for strong duality to hold.

The following are some examples of problems for which strong duality holds. These examples show that, in general, the strong duality relation $q^* = f^*$ provides no information about the existence of dual optimal solutions.

Example 18 *(Unique Dual Optimal Solution) Consider the following primal problem*

$$\begin{array}{ll} \textit{minimize} & \frac{1}{2}\left(x_1^2 + x_2^2\right) \\ \textit{subject to} & x_1 \leq 1. \end{array}$$

Its optimal value is $f^* = 0$, which is attained at $(0, 0)$. The dual function is

$$q(\mu_1) = \inf_{x \in \mathbb{R}^2} \left\{ \frac{1}{2}\left(x_1^2 + x_2^2\right) + \mu_1 (x_1 - 1) \right\}.$$

To determine the infimum in the dual function, we set the gradient of the Lagrangian function to zero, and obtain

$$x_1 + \mu_1 = 0, \qquad x_2 = 0.$$

For a fixed $\mu_1 \geq 1$, the infimum is attained at $(-\mu_1, 0)$, implying that

$$q(\mu_1) = -\frac{1}{2}\mu_1^2 - \mu_1.$$

The dual optimal value over $\mu_1 \geq 0$ is $q^* = 0$, which is attained at $\mu_1 = 0$.

Example 19 *(Multiple Dual Optimal Solutions) Consider the following primal problem*

$$\begin{array}{ll} \textit{minimize} & |x| \\ \textit{subject to} & x \leq 0. \end{array}$$

Its optimal value is $f^* = 0$, attained at $x = 0$. The dual problem is

$$q(\mu_1) = \inf_{x \in \mathbb{R}} \{|x| + \mu_1 x\} = \inf_{x \in \mathbb{R}} \left(\mu_1 + sgn(x)\right) x = \begin{cases} 0 & \textit{if } 0 \leq \mu_1 \leq 1 \\ -\infty & \textit{otherwise.} \end{cases}$$

Thus, the dual optimal value is $q^* = 0$ and the dual optimal set is the interval $[0, 1]$.

Example 20 *(No Dual Optimal Solution)*

$$\begin{array}{ll} \textit{minimize} & x \\ \textit{subject to} & x^2 \leq 0. \end{array}$$

The primal optimal value is $f^* = 0$*, which is attained at* $x = 0$*. The dual function is*

$$q(\mu_1) = \inf_{x \in \mathbb{R}} \{x + \mu_1 x^2\}.$$

The Lagrangian function $x \mapsto x + \mu_1 x^2$ *attains its minimum at* $x = -\frac{1}{2\mu_1}$ *for* $\mu_1 > 0$*. Hence, the dual function is*

$$q(\mu_1) = \begin{cases} -\frac{1}{4\mu_1^2} & \textit{for } \ \mu_1 > 0, \\ -\infty & \textit{otherwise}, \end{cases}$$

whose supremum over $\mu_1 \geq 0$ *is* $q^* = 0$*, but there is no* $\mu_1 > 0$ *achieving the optimal value.*

Similar to the preceding, we can construct examples when $q^* = f^*$ and there is no information about the existence of the primal optimal solutions.

2.5.3 Linear Programming Duality

In this section, we study the LP duality, which is special due to the linear structure of LP problems. The LP duality provides insights leading to many applications and algorithm designs, such as the dual simplex algorithm.

The linear problem of the form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0 \end{array} \tag{2.23}$$

is the *LP in the standard form*. The linear program given in any other from can be transformed into the standard form. For example, let the LP be given as maximization problem

$$\begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0. \end{array}$$

Then, it is equivalent to

$$\begin{array}{ll} \text{minimize} & -c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0. \end{array}$$

Let the LP be given by

$$\text{minimize} \qquad c^T x$$

$$\text{subject to} \quad Ax \leq b, \tag{2.24}$$

for an $m \times n$ matrix A and a vector $b \in \mathbb{R}^m$. It can be transformed to the standard form by introducing the *slack variables* and some additional variables. In particular, define the slack variables $s \in \mathbb{R}^m$, $s \geq 0$ such that

$$Ax + s = b.$$

Then, LP problem (2.24) reduces to

$$\begin{array}{ll} \text{minimize} & c^T x + 0^T s \\ \text{subject to} & Ax + s = b \\ & x \in \mathbb{R}^n,\ s \geq 0. \end{array} \tag{2.25}$$

Further, we can write the vector $x \in \mathbb{R}^n$ as $x = x^+ + x^-$, with $x^+ \geq 0$ and $x^- \leq 0$. Then, by introducing the variable $y^+ = -x^- \geq 0$, we can write

$$x = x^+ - y^+ \qquad \text{with } x^+ \geq 0 \text{ and } y^+ \geq 0.$$

Thus, the LP of Eq. (2.25) is equivalent to the following LP

$$\begin{array}{ll} \text{minimize} & c^T x^+ - c^T y^+ + 0^T s \\ \text{subject to} & Ax^+ - Ay^+ + s = b \\ & x^+ \geq 0,\ y^+ \geq 0,\ s \geq 0. \end{array} \tag{2.26}$$

Hence, the original LP of Eq. (2.24) is equivalent to an LP in a standard form but with a larger size (the number of variables is $2n + m$, as opposed to n in the original LP).

The dual of an LP problem is also an LP problem. To illustrate this, we consider an LP in a standard form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0. \end{array} \tag{2.27}$$

By assigning prices $\lambda \in \mathbb{R}^m$ (a price λ_i per equality constraint), we obtain the Lagrangian function

$$q(\lambda) = \inf_{x \geq 0} \{c^T x + \lambda^T (Ax - b)\} = -b^T \lambda + \inf_{x \geq 0} (c + A^T \lambda)^T x.$$

Note that the infimum in the right hand side of the preceding relation is $-\infty$ if $c_i + [A^T \lambda]_i < 0$ for some i, and otherwise it is 0. Hence,

$$q(\lambda) = \begin{cases} -b^T \lambda & \text{when } A^T \lambda + c \geq 0, \\ -\infty & \text{otherwise.} \end{cases}$$

When maximizing $q(\lambda)$, the values $-\infty$ need not be considered, so the dual problem is the following

$$\text{maximize} \qquad -b^T \lambda$$

$$\text{subject to} \quad A^T\lambda \geq -c. \tag{2.28}$$

This is also an LP problem in the dual variables (prices) $\lambda \in \mathbb{R}^m$. Furthermore, note that we could have considered another dual of problem (2.27) by assigning additional prices μ to the inequality constraints $x \geq 0$. This has been considered in Example 14. As seen there, the alternative dual is equivalent to the dual considered here.

To summarize, a dual of given LP is not always of the same form. The form of the dual problem depends on which constraints are selected to be "priced". However, all dual problems corresponding to the same primal problem are equivalent.

If replace λ by $-p$ in the dual problem of Eq. (2.28), we obtain an equivalent LP problem

$$\begin{array}{ll} \text{maximize} & b^T p \\ \text{subject to} & A^T p \leq c. \end{array} \tag{2.29}$$

When compared to the primal LP of Eq. (2.27), we see that the primal problem is a *minimization problem with the objective* $c^T x$, while its dual (2.29) is a *maximization problem with objective* $b^T p$. The primal problem (2.27) has *equality constraints* $Ax = b$, while the dual (2.29) has *inequality constraints* $A^T p \leq c$. The primal (2.27) has *the sign constraint* $x \geq 0$ *on the primal variable* x, while the dual (2.29) has *no constraints on the dual variable* p (the dual variable is free).

The weak duality relation [cf. Theorem 24(c)] for the primal-dual LP problems, yields

$$b^T p \leq c^T x$$

for all primal feasible x and dual feasible p, i.e., for all $x \geq 0$ with $Ax = b$ and all p with $A^T p \leq c$.

The LP duality has a property that "the dual of the dual is the primal", meaning that if we form the dual of the dual of a given (primal) LP, we obtain the original (primal) LP. This is formally stated in the following theorem.

Theorem 25 *By taking the dual of the transformed dual in a minimization problem, we obtain the primal problem.*

Proof. We prove this theorem by using the primal-dual LP problems of Eqs. (2.27) and (2.29). The dual problem in Eq. (2.29) is equivalent to the following

$$\begin{array}{ll} \text{minimize} & -b^T p \\ \text{subject to} & A^T p \leq c. \end{array}$$

By assigning prices y, $y \geq 0$, to the inequality constraints $A^T p \leq c$, we have the following dual function

$$\tilde{q}(y) = \inf_{y \in \mathbb{R}^n} \{-b^T p + y^T (A^T p - c)\} = -c^T p + \inf_{y \in \mathbb{R}^n} (Ay - b)^T p.$$

Since

$$\inf_{y \in \mathbb{R}^n} (Ay - b)^T p = \begin{cases} 0 & \text{when } Ay - b = 0, \\ -\infty & \text{otherwise,} \end{cases}$$

it follows

$$\tilde{q}(y) = \begin{cases} -c^T y & \text{when } Ay = b, \\ -\infty & \text{otherwise.} \end{cases}$$

Hence, the dual of the dual problem is

$$\begin{array}{ll} \text{maximize} & -c^T y \\ \text{subject to} & Ay = b \\ & y \geq 0, \end{array}$$

or equivalently

$$\begin{array}{ll} \text{minimize} & c^T y \\ \text{subject to} & Ay = b \\ & y \geq 0, \end{array}$$

which is the same as the primal problem of Eq. (2.27). ■

We have the following results for the LP feasibility. These results are a consequence of the weak duality and the LP duality symmetry of Theorem 25.

Theorem 26 *The following hold:*

(a) If a primal LP has optimal value $f^* = -\infty$, then the dual LP problem is infeasible.

(b) If a dual LP has optimal value $q^* = +\infty$, then the primal LP problem is infeasible.

Proof. Let the primal optimal value be $f^* = -\infty$, and assume that the dual problem is feasible. Then, there exists a dual feasible $\hat{p}$ such $q(\hat{p}) > -\infty$. On the other hand, by weak duality, for all dual feasible p, we have $q(p) \leq -\infty$, implying that $q(p) = -\infty$ for all feasible p, which contradicts the feasibility of the dual at $\hat{p}$ [i.e., the relation $q(\hat{p}) > -\infty$]. Hence, the dual must be infeasible. To show the statement in part (b), we use a symmetrical argument. ■

It is possible that both the primal and the dual LP are infeasible, as seen in the following example.

Example 21 *Consider the following primal LP problem*

$$\begin{array}{ll} \textit{minimize} & x_1 + 2x_2 \\ \textit{subject to} & x_1 + x_2 = 1 \\ & x_1 + x_2 = 2. \end{array}$$

Its dual is

$$\begin{array}{ll} \textit{maximize} & p_1 + 2p_2 \\ \textit{subject to} & p_1 + p_2 = 1 \\ & p_1 + p_2 = 2. \end{array}$$

Clearly, both problems are infeasible.

In general for an LP problem, there are three possible situations:

(1) The problem is infeasible ($f^* = +\infty$).

(2) The problem is unbounded ($f^* = -\infty$).

(3) The problem has an optimal solution (f^* is finite).

Note that the case (3) follows from Corollary 1. In particular, by Corollary 1, we have that an LP problem is feasible if and only if it has an optimal solution.

The weak duality implies a further characterization of a primal-dual optimal pair, as seen in the following theorem.

Theorem 27 *Let x be a primal feasible and p be a dual feasible for an LP problem. If the following relation is satisfied*

$$c^T x = b^T p,$$

then x is primal optimal and p is dual optimal.

Proof. Since (x, p) is a primal-dual feasible pair, by weak duality we have $b^T p \leq c^T x$. If $c^T x = b^T p$, it follows that x must be primal optimal and p must be dual optimal. ■

The reverse statement to that of Theorem 27 also holds. It actually holds for a more general class of convex problems than the class of LP problems, as we will see later on in Section 2.5.7.

In the following theorem, we provide the strong duality result for LP problems.

Theorem 28 *Let an LP problem have a finite optimal value f^*. Then:*

(a) A primal optimal solution exists.

(b) A dual optimal solution exists.

(c) The primal and the dual optimal values are equal.

Another relation characterizes an LP primal-dual optimal pair (x^*, p^*), a relation known as *complementarity slackness* condition. We will prove this relation later for a more general class in Section 2.5.7.

Theorem 29 *(Complementarity Slackness) Let x^* be a primal feasible and p^* be a dual feasible for an LP problem. The vectors x^* and p^* are respectively primal and dual optimal if and only if the following relations are satisfied*

$$p_i^* \left([A]_i x^* - b_i\right) = 0 \qquad \text{for all } i = 1, \ldots, m,$$

$$\left(c_j - (p^*)^T [A]^j\right) x_j = 0 \qquad \text{for all } j = 1, \ldots, n.$$

The complementarity slackness condition for the problem

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \leq b, \end{array}$$

implies that for the optimal primal solution x^* and the optimal dual price p^*, we have:

- If the constraint $[A]_i x \leq b_i$ is not active at x^*, i,e., $[A]_i x^* < b_i$, then the price p_i^* must be zero.

- If the constraint $[A]_i x \leq b_i$ is active at x^*, i,e., $[A]_i x^* = b_i$, then the price p_i^* may be nonzero.

Thus, the dual optimal price p_i^* is zero unless the constraint $[A]_i x^* \leq b_i$ is active. Intuitively, this means that a constraint that is not active at an optimal solution may be removed from the problem (or slightly perturbed) without affecting the optimal value.

In the following example, we illustrate how the complementarity slackness condition can be used to determine a dual optimal solution.

Example 22 *Consider the following problem and its dual*

$$\begin{array}{ll} \textit{minimize} & 3x_1 + 10x_2 + 3x_3 \\ \textit{subject to} & x_1 + x_2 + 3x_3 = 4 \\ & x_1 + x_2 = 1 \\ & x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0, \end{array}$$

$$\begin{array}{ll} \textit{maximize} & 4p_1 + p_2 \\ \textit{subject to} & p_1 + p_2 \leq 3 \\ & p_1 + p_2 \leq 10 \\ & 3p_1 \leq 3. \end{array}$$

Consider the vector $x^* = (1, 0, 1)$. We will use the complementarity slackness to show that x^* is a primal optimal and to construct a dual optimal p_i^*. The relation $p_i\left([A]_i x^* - b_i\right) = 0$ is satisfied for all i, since the primal is in the standard form. The relation $\left(c_j - (p^*)^T [A]^j\right) x_j = 0$ is satisfied for $j = 2$, since $x_2^* = 0$. Since $x_1^* = x_3^* = 1$, the relations $\left(c_j - (p^*)^T [A]^j\right) x_j = 0$ for $j = 1$ and $j = 3$ imply

$$p_1^* + p_2^* = 3, \qquad 3p_1^* = 3.$$

Hence, $p_1^* = 1$ and $p_2^* = 2$. The vector $x^* = (1, 0, 1)$ is primal feasible, and $p^* = (1, 0, 2)$ is dual feasible. Thus, by Theorem 29, the vectors x^* and p^* are primal and dual optimal, respectively.

We note that, here, the dual optimal p^* has been uniquely determined from complementarity slackness. However, in general, this may not be the case.

2.5.4 Slater Condition

Here, we consider convex constrained optimization problem of the following form:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_j(x) \leq 0, \quad j = 1, \ldots, m \\ & x \in X. \end{array} \tag{2.30}$$

Throughout this section, we assume that the problem is feasible and convex. In particular, we use the following assumption.

Assumption 2 *The following hold:*

(a) The optimal value f^* of the problem (2.30) is finite.

(b) The objective function f and all constraint functions g_j are convex.

(c) The set $X \subseteq \mathbb{R}^n$ is nonempty and convex.

The Slater condition is a condition imposed on the constraint set $\{x \in \mathbb{R}^n \mid g_j(x) \leq 0, j = 1, \ldots, m,\ x \in X\}$.

Assumption 3 *There is a vector $\bar{x} \in X \cap \mathrm{dom} f$ such that*

$$g_j(x) < 0 \qquad \textit{for all } j = 1, \ldots, m.$$

We often refer to a vector satisfying the Slater condition as a *Slater vector*.

For the problem of Eq. (2.30), consider the set $V \subset \mathbb{R}^m \times \mathbb{R}$ given by

$$V = \{(u, w) \mid g(x) \leq u,\ f(x) \leq w,\ x \in X\}.$$

The set V is convex by the convexity of f, $g_j's$, and X. The Slater condition for this set is illustrated in Figure 2.17.

For example, let X be a polyhedral set given by

$$X = \{x \in \mathbb{R}^n \mid Ax \leq b,\ Bx = d\},$$

where A is an $m \times n$ matrix, $b \in \mathbb{R}^m$, B is a $p \times n$ matrix, and $d \in \mathbb{R}^p$. Also let f be defined over $\mathbb{R}^n$, then the Slater condition becomes: there is a vector $\bar{x}$ such that $A\bar{x} \leq b$, $B\bar{x} = d$, and

$$g_j(x) < 0 \qquad \text{for all } j = 1, \ldots, m.$$

We next provide an example of a problem satisfying the Slater condition.

Example 23 *Consider the following $\|\cdot\|_\infty$ minimization problem:*

$$\textit{minimize} \quad \|Ax - b\|_\infty.$$

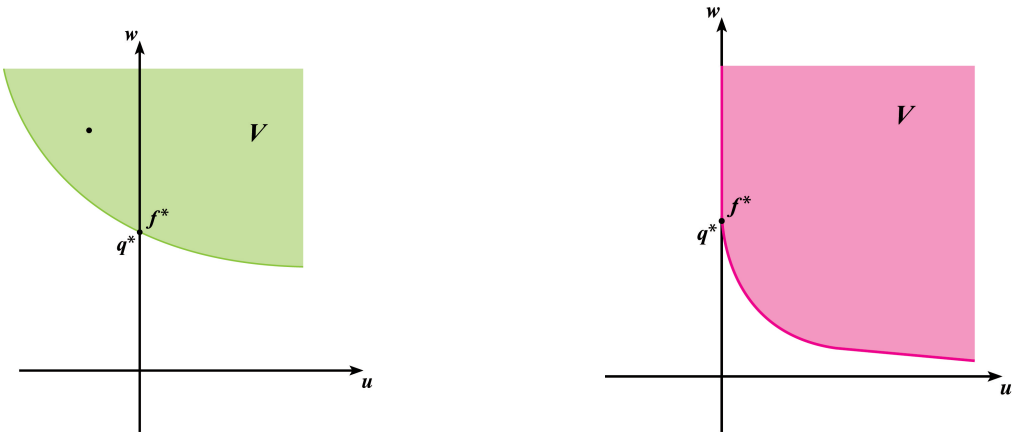

Figure 2.17: The plot to the left illustrates the set V with a Slater vector, while the plot to the left illustrates a set V without a Slater vector.

The problem is equivalent to the following LP problem.

$$\begin{array}{ll} \textit{minimize} & t \\ \textit{subject to} & a_j^T x - b_j - t \leq 0, \ j = 1, \ldots, m \\ & b_j - a_j^T x - t \leq 0, \ j = 1, \ldots, m. \end{array}$$

The LP minimization is with respect to $(x, t) \in \mathbb{R}^n \times \mathbb{R}$. Consider the vector $(\bar{x}, \bar{t}\,)$ given by

$$\bar{x} = 0 \quad \textit{and} \quad \bar{t} = \epsilon + \max_{1 \leq j \leq m} |b_j| \quad \textit{for some } \epsilon > 0.$$

The vector $(\bar{x}, \bar{t})$ satisfies the Slater condition.

Under the Slater condition, we have the following well-known strong duality result.

Theorem 30 *Let Assumption 2 and the Slater condition hold [Assumption 3]. Then for the problem of Eq. (2.30), we have:*

(a) There is no duality gap, i.e., $q^* = f^*$.

(b) The set of dual optimal solutions is nonempty and bounded.

The following is an example of application of Theorem 30.

Example 24 *Consider the canonical utility-based network resource allocation model proposed in the seminal work of Kelly et. al [19]. We are given a network that consists of a set $\mathcal{S} = \{1, \ldots, S\}$ of sources and a set $\mathcal{L} = \{1, \ldots, L\}$ of undirected links, where a link l has capacity c_l. Let $\mathcal{L}(i) \subset \mathcal{L}$ denote the set of links used by source i. The application requirements of source i is represented by a concave increasing utility function $u_i : [0, \infty) \to [0, \infty)$, i.e., each source i gains a utility $u_i(x_i)$ when it sends data at a rate x_i. We further assume that rate x_i is constrained to lie in the interval $I_i = [0, M_i]$ for all $i \in \mathcal{S}$, where the scalar M_i denotes the maximum allowed rate for source i. Let $\mathcal{S}(l) = \{i \in \mathcal{S} \mid l \in \mathcal{L}(i)\}$ denote*

the set of sources that use link l. The goal of the network utility maximization problem is to allocate the source rates as the optimal solution of the problem

$$\begin{aligned} &\textit{maximize} && \sum_{i\in\mathcal{S}} u_i(x_i) \\ &\textit{subject to} && \sum_{i\in\mathcal{S}(l)} x_i \le c_l \quad \textit{for all } l \in \mathcal{L} \\ & && x_i \in I_i \quad \textit{for all } i \in \mathcal{S}. \end{aligned}$$

We let x be the vector with components x_i for $i \in \mathcal{S}$. Note that the preceding problem can be equivalently casted as a minimization of a convex function $f(x) = -\sum_{i\in\mathcal{S}} u_i(x_i)$. Also, we let

$$X = \{x \mid x_i \in I_i \quad \textit{for all } i \in \mathcal{S}\},$$

and

$$g_l(x) = \sum_{i\in\mathcal{S}(l)} x_i - c_l \quad \textit{for all } l \in \mathcal{L}.$$

In this form, we see that the zero vector is such that $0 \in \mathrm{dom} f$, and $0 \in X$. Furthermore

$$g_l(0) = -c_l < 0 \quad \textit{for all } l \in \mathcal{L}.$$

Hence, for the network utility maximization problem, the zero vector satisfies the Slater condition. Thus, according to Theorem 30, there is no duality gap and the dual of the network utility problem has nonempty and bounded set of optimal prices.

2.5.5 Linear Constraint Condition

Here, we consider a strong duality result for a special convex minimization problem. In particular, we consider minimization of a convex function subject to linear constraints. In this case, as we will see the strong duality also holds under another condition different from the Slater condition.

Consider the following primal problem with linear constraints

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && Ax \le b,\ Bx = d \\ & && x \in X. \end{aligned} \tag{2.31}$$

We have the following strong duality result.

Theorem 31 *Let the optimal value f^* of problem (2.31) be finite. Assume that there exists a feasible vector for the problem that belongs to the relative interior of the set X and the domain of f, i.e., a vector $\tilde{x} \in \mathbb{R}^n$ such that*

$$A\tilde{x} \le b, \qquad B\tilde{x} = d \qquad \textit{and} \qquad \tilde{x} \in \mathrm{relint}(X) \cap \mathrm{relint}(\mathrm{dom} f).$$

Then, there is no duality gap and a dual optimal solution exists.

Recall that the relative interior of a given set C is the interior of the set relative to the smallest affine set that contains C. In particular, when the dimensions of X and $\mathrm{dom} f$ are equal to n, then the relative interior condition of Theorem 31 reduces to the interior condition as follows:

$$A\tilde{x} \leq b, \qquad B\tilde{x} = d \qquad \text{and} \qquad \tilde{x} \in \mathrm{int}(X) \cap \mathrm{int}(\mathrm{dom} f).$$

This condition becomes redundant in Theorem 31 when $X = \mathbb{R}^n$ and f is defined over $\mathbb{R}^n$ (i.e., $\mathrm{dom} f = \mathbb{R}^n$). In particular, in this case, it reduces to

$$A\tilde{x} \leq b, \qquad B\tilde{x} = d \qquad \text{for some} \qquad \tilde{x} \in \mathbb{R}^n,$$

which is just the feasibility of the problem (2.31). However, the feasibility is already satisfied in view of the assumption of Theorem 31 that the optimal value f^* is finite. A particular direct consequence of this is the LP duality, studied in Section 2.5.3.

Observe that, similar to Theorem 30 (the Slater condition), the relative interior condition of Theorem 31 is imposed only on the constraint set, and it guarantees the nonemptiness of the dual optimal set. However, unlike Theorem 30, the relative interior condition of Theorem 31 does not guarantee the boundedness of the dual optimal set.

The following is an example of a problem not satisfying the relative interior condition of Theorem 31. In particular, consider the problem given by

$$\begin{array}{ll} \text{minimize} & e^{-\sqrt{x_1 x_2}} \\ \text{subject to} & x_1 \leq 0, \quad x \in \mathbb{R}^2. \end{array}$$

Here, $X = \mathbb{R}^2$ and hence $\mathrm{relint}(X) = \mathbb{R}^2$. The feasible set is

$$C = \{x \in \mathbb{R}^2 \mid x_1 = 0,\ x_2 \in \mathbb{R}\}.$$

The domain of f is the set $\{x \in \mathbb{R}^2 \mid x_1 \geq 0,\ x_2 \geq 0\}$, whose relative interior coincides with the interior, i.e.,

$$\mathrm{relint}(X) = \mathrm{int}(X) = \{x \mid x_1 > 0,\ x_2 > 0\}.$$

Hence,

$$\mathrm{relint}(X) \cap \mathrm{relint}(\mathrm{dom} f) = \mathbb{R}^2 \cap \{x \mid x_1 > 0,\ x_2 > 0\} = \{x \mid x_1 > 0,\ x_2 > 0\}.$$

However, none of the feasible vectors lies in $\mathrm{relint}(X) \cap \mathrm{relint}(\mathrm{dom} f)$, i.e.,

$$C \cap \mathrm{relint}(X) \cap \mathrm{relint}(\mathrm{dom} f) = C \cap \{x \mid x_1 > 0,\ x_2 > 0\} = \emptyset.$$

Hence, the relative interior condition fails. Note that $f^* = 1$ and $q^* = 0$, and there is a duality gap.

Consider the same constraint set with an objective $f(x) = -\sqrt{x_1}$. Again, the relative interior condition is not satisfied. In this case, it can be seen that $f^* = q^* = 0$. Thus, the strong duality holds, but a dual optimal solution does not exist.

2.5.6 Quadratic Convex Problem

Another special problem is minimization of a convex quadratic objective over a quadratic constraint set:

$$\begin{array}{ll} \text{minimize} & x^T Q_0 x + a_0^T x + b_0 \\ \text{subject to} & x^T Q_j x + a_j^T x + b_j \leq 0, \quad j = 1, \ldots, m \\ & x \in \mathbb{R}^n, \end{array} \tag{2.32}$$

where each Q_i is an $n \times n$ symmetric positive semidefinite matrix, each $a_j \in \mathbb{R}^n$ and $b_j \in \mathbb{R}$. For such a problem, there is no duality gap whenever the optimal value f^* is finite, as seen in the next theorem.

Theorem 32 *Let each Q_j in problem (2.32) be a symmetric positive semidefinite matrix. Also, let the optimal value f^* of the problem be finite. Then, there is no duality gap and the primal optimal set is nonempty.*

Note that Theorem 32 says nothing about the existence of dual optimal solutions. This is opposed to the Slater and the relative interior condition of Theorems 30 and 31, which say nothing about the existence of the primal optimal solutions.

Also note that an LP problem can be viewed as a special case of problem (2.32), where $b_0 = 0$ and $Q_j = 0$ for all j.

An example of a problem that can be reduced to the form (2.32) is the following:

$$\begin{array}{ll} \text{minimize} & \|x\| \\ \text{subject to} & Ax = b \\ & x \in \mathbb{R}^n. \end{array}$$

In particular, note that the preceding is equivalent to the minimization of $x^T I x$ subject to linear constraints $[A]_i\, x \leq b_i$, $i = 1, \ldots m$, $-[A]_i\, x \leq -b_i$, $i = 1, \ldots m$.

2.5.7 Karush-Kuhn-Tucker Conditions

In this section, we consider the primal-dual optimality condition for a general convex problem. As a special case of this condition, we obtain the Karush-Kuhn-Tucker conditions characterizing primal-dual optimal pairs.

We consider the following primal problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_j(x) \leq 0, \quad j = 1, \ldots, m \\ & a_i^T x = b_i, \quad i = 1, \ldots, p \\ & x \in X. \end{array} \tag{2.33}$$

Its Lagrangian function is given by

$$\mathcal{L}(x, \mu, \lambda) = f(x) + \mu^T g(x) + \lambda^T (Ax - b), \qquad \mu \in \mathbb{R}^m,\ \mu \geq 0,\ \lambda \in \mathbb{R}^p,$$

where $g = (g_1, \ldots, g_m)^T$ and A is a matrix with rows a_i^T, $i = 1, \ldots, p$. The dual function is

$$q(\mu, \lambda) = \inf_{x \in X} \mathcal{L}(x, \mu, \lambda) = \inf_{x \in X} \left\{ f(x) + \mu^T g(x) + \lambda^T (Ax - b) \right\}.$$

Note that the infimum in the dual function is actually taken over $X \cap \text{dom} f \cap \text{dom} g_1 \cap \ldots \cap \text{dom} g_m$. Thus, the dual problem is

$$\max_{\mu \geq 0,\ \lambda \in \mathbb{R}^p} q(\mu, \lambda).$$

The following theorem provides necessary and sufficient conditions for optimality of primal-dual pairs. These conditions apply for as long as the primal optimal value f^* is finite and the strong duality holds, i.e., $f^* = q^*$.

Theorem 33 *(Optimality Condition for Primal-Dual Pairs)*
Consider the problem of Eq. (2.33). Assume that the problem is convex [Assumption 2] and that the optimal value f^* is finite. Furthermore, assume that the strong duality holds, i.e., $q^* = f^*$. Then x^* is a primal optimal and (μ^*, λ^*) is a dual optimal if and only if the following conditions are satisfied:

(1) Primal Feasibility: x^* is primal feasible i.e.,

$$g(x^*) \leq 0, \quad Ax^* = b, \quad x^* \in X \cap \text{dom} f.$$

(2) Dual Feasibility: (μ^*, λ^*) is dual feasible i.e., $\mu^* \geq 0$.

(3) Lagrangian Optimality in x: x^* attains the minimum in $\inf_{x \in X} \mathcal{L}(x, \mu^*, \lambda^*)$.

(4) Complementarity Slackness: The multiplier μ^* is such that

$$\mu_j^* g_j(x^*) = 0 \qquad \text{for all } j = 1, \ldots, m.$$

The condition in part (4) of Theorem 33 is also known as *Lagrangian Optimality in* (μ, λ), since it is equivalent to the following statement:

$$(\mu^*, \lambda^*) \quad \text{attains the maximum in} \quad \sup_{\mu \geq 0,\ \lambda \in \mathbb{R}^p} \mathcal{L}(x^*, \mu, \lambda).$$

We now discuss some implications of the optimality conditions of Theorem 33. Suppose there is no duality gap and we have an optimal dual multiplier (μ^*, λ^*) for problem 2.33). We may consider minimizing the Lagrangian $\mathcal{L}(x, \mu^*, \lambda^*)$ over $x \in X$, i.e.,

$$\text{minimize} \quad f(x) + (\mu^*)^T g(x) + (\lambda^*)^T (Ax - b) \quad \text{over } x \in X.$$

The possibilities for this problem are:

(i) There is a unique minimizer x^* and this minimizer is feasible. Then, according to Theorem 33, the solution x^* is primal optimal. For example, a minimizer is unique when $\mathcal{L}(x, \mu^*, \lambda^*)$ is strictly convex in x.

(ii) A unique minimizer exists but it is not feasible. Then, the primal problem has no optimal solution (no primal feasible x achieving f^*).

(iii) There are multiple minimizers. Then, only those that are primal feasible are actually primal optimal.

We illustrate an application of Theorem 33 in the following example.

Example 25 *(Entropy Maximization, see [13] page 228.)*
Consider the entropy maximization problem

$$\begin{array}{ll} \textit{minimize} & \sum_{i=1}^n x_i \ln x_i \\ \textit{subject to} & Ax \leq b,\ \sum_{i=1}^n x_i = 1, \end{array}$$

with domain $x \geq 0$. Its dual is given by

$$\begin{array}{ll} \textit{maximize} & -b^T\mu - \lambda - e^{-\lambda-1}\sum_{i=1}^n e^{-a_i^T\mu} \\ \textit{subject to} & \mu \geq 0. \end{array}$$

Suppose that the Slater condition holds, i.e., there is a vector $\bar{x}$ such that

$$A\bar{x} \leq b, \quad \sum_{i=1}^n \bar{x}_i = 1, \quad \bar{x} \geq 0.$$

Thus, there is no gap and a dual optimal solution (μ^*, λ^*) exists.

The Lagrangian $\mathcal{L}(x, \mu, \lambda)$ at (μ^*, λ^*) is given by

$$\mathcal{L}(x, \mu^*, \lambda^*) = \sum_{i=1}^n x_i \ln x_i + (\mu^*)^T(Ax - b) + (\lambda^*)^T\left(e^T x - 1\right),$$

which is strictly convex in x over the domain and has a unique minimizer x^*, with components x_i^* given by

$$x_i^* = e^{-\left(a_i^T\mu^* + \lambda^* + 1\right)} \qquad \textit{for all } i = 1, \ldots, n.$$

If x^* is primal feasible, then x^* is a primal optimal solution. If x^* is not primal feasible, then the primal problem has no solution.

As a special case of Theorem 33 when $X = \mathbb{R}^n$ and the functions f and g_j are differentiable, we obtain a well-known *Karush-Kuhn-Tucker (KKT) conditions.*

Theorem 34 *Consider the primal problem*

$$\begin{array}{lll} \textit{minimize} & f(x) & \\ \textit{subject to} & g_j(x) \leq 0, & j = 1, \ldots, m \\ & Ax = b, & \end{array}$$

where f and all g_j are differentiable. Assume that the problem is convex [Assumption 2] and that the optimal value f^* is finite. Furthermore, assume that the strong duality holds, i.e., $q^* = f^*$. Then x^* is a primal optimal and (μ^*, λ^*) is a dual optimal if and only if the following conditions are satisfied:

(1) Primal Feasibility: $g_j(x^*) \leq 0,\ j = 1, \ldots, m,\ \ Ax^* = b.$

(2) Dual Feasibility: $\mu^* \geq 0.$

(3) Lagrangian Optimality in x: *The gradient of $\mathcal{L}(x, \mu^*, \lambda^*)$ with respect to x vanishes,*

$$\nabla f(x^*) + \sum_{j=1}^{m} \mu_j^* \nabla g_j(x^*) + \sum_{i=1}^{p} \lambda_i a_i = 0,$$

where a_i is the i-th row of the matrix A.

(4) Complementary Slackness: $\mu_j^*\, g_j(x^*) = 0 \quad$ *for all* $j = 1, \ldots, m.$

As seen from Theorem 34 for convex problems with no duality gap and finite optimal value f^*, the KKT conditions are necessary and sufficient for optimality. However, for a general nonconvex problem with no duality gap and finite f^*, the KKT conditions are only necessary.

The KKT conditions are important since they provide a certificate of optimality for primal-dual pairs. This is often exploited in the design of optimization algorithms and their performance analysis. These conditions are also used for verifying optimality. They are also used as a design principle, i.e., some algorithms are designed for solving KKT equations, thus producing both primal and dual optimal solutions.

We now provide some examples.

Example 26 *(Power Allocation to Communication Channels)*
The problem of power allocation among n communication channels can be formulated as a minimization problem of the following form:

$$\begin{array}{ll} \textit{minimize} & -\sum_{i=1}^{n} \ln(\alpha_i + x_i) \\ \textit{subject to} & x \geq 0,\ \sum_{i=1}^{n} x_i = 1, \end{array}$$

where each α_i is a scalar such that $\alpha_i > 0$. The problem arises in information theory when allocating power among n channels. The decision variable x_i represents the power allocated to the i-th channel. The total power is normalized, so that x_i actually represents a fraction of the total power allocated to channel i. The function $\ln(\alpha_i + x_i)$ gives the capacity (communication rate) of the i-th channel. Thus, the problem consists of allocating a total power of one unit to the channels so as to maximize the total communication rate, or equivalently minimize $-\sum_{i=1}^{n} \ln(\alpha_i + x_i)$.

The domain of the objective function is $\mathrm{dom} f = \{x \mid x + \alpha \geq 0\}$, *and the objective function is continuous over its domain. The constraint set is compact and contained in the domain* $\mathrm{dom} f$. *Hence, the objective function is bounded over the constraint set, and therefore the optimal value f^* is finite. Furthermore, the Slater condition is satisfied for $\bar{x} = (1/n, \ldots, 1/n)$. Hence, there is no duality gap.*

The KKT conditions for the power allocation problem are:

$$x^* \geq 0, \qquad \sum_{i=1}^{n} x_i^* = 1, \qquad \mu^* \geq 0,$$

$$\mu_i^* x_i^* = 0, \qquad \lambda^* - \frac{1}{\alpha_i + x_i^*} - \mu_i^* = 0 \qquad \textit{for all } i.$$

By eliminating μ_i^* from the last relation, we obtain the following:

$$x^* \geq 0, \qquad \sum_{i=1}^n x_i^* = 1, \qquad \lambda^* \geq \frac{1}{\alpha_i + x_i^*}, \qquad x_i^* \left(\lambda^* - \frac{1}{\alpha_i + x_i^*} \right) = 0 \qquad \textit{for all } i.$$

If $\lambda^* < 1/\alpha_i$, by the third relation, we obtain $x_i^* > 0$. By Complementarity Slackness (the last relation), it follows that $\lambda^* = 1/(\alpha_i + x_i^*)$, so that $x_i^* = 1/\lambda^* - \alpha_i$. If $\lambda^* \geq 1/\alpha_i$, by Complementarity Slackness, we have $x_i^* = 0$. Hence,

$$x_i^* = \begin{cases} \frac{1}{\alpha_i + x_i^*} & \textit{if } \lambda^* < \frac{1}{\alpha_i} \\ 0 & \textit{otherwise} \end{cases} \qquad \textit{for all } i = 1, \ldots, n,$$

or equivalently

$$x_i^* = \max \left\{ 0, \frac{1}{\lambda^*} - \alpha_i \right\} \qquad \textit{for all } i = 1, \ldots, n,$$

where λ^* is determined from the relation $\sum_{i=1}^n x_i^* = 1$, i.e., $\sum_{i=1}^n \max\{0, 1/\lambda^* - \alpha_i\} = 1$. Note that $\sum_{i=1}^n \max\{0, 1/\lambda^* - \alpha_i\}$ is a piece-wise linear function increasing in $1/\lambda^*$, and it has a unique solution.

Example 27 *(Separable Objective with Equality Constraint, Example 5.4 of [13] page 248.) Consider the problem*

$$\begin{array}{ll} \textit{minimize} & \sum_{i=1}^n f_i(x_i) \\ \textit{subject to} & a^T x = b, \end{array}$$

where $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$. Each function $f_i : \mathbb{R} \to \mathbb{R}$ is differentiable and strictly convex. The objective function $\sum_{i=1}^n f_i(x_i)$ is referred to as separable, since it is a sum of functions of the individual components x_i of the vector x. Assume that an optimal solution exist. Then, in view of the strict convexity of the objective function, the optimal solution is unique. Furthermore, note that by Theorem 31, there is no duality gap.

The Lagrangian function is

$$\mathcal{L}(x, \lambda) = \sum_{i=1}^n f_i(x_i) + \lambda(a^T x - b) = -\lambda b + \sum_{i=1}^n \left(f_i(x_i) + \lambda\, a_i x_i \right),$$

where $\lambda \in \mathbb{R}$. Note that the Lagrangian is also separable in x. The dual function is given by

$$q(\lambda) = -\lambda b + \inf_{x \in \mathbb{R}^n} \sum_{i=1}^n \left(f_i(x_i) + \lambda\, a_i x_i \right) = -\lambda b + \sum_{i=1}^n \inf_{x_i \in \mathbb{R}} \left(f_i(x_i) + \lambda\, a_i x_i \right).$$

Since f_i is differentiable, for a given λ, the infimum $\inf_{x_i \in \mathbb{R}} \left(f_i(x_i) + \lambda\, a_i x_i \right)$ is attained at $x_i(\lambda)$ such that

$$f_i'(x_i(\lambda)) = -\lambda a_i.$$

Let λ^* be an optimal dual solution, and let $x^* = (x_1^*, \ldots, x_n^*)$ be such that

$$f_i'(x_i^*) = -\lambda^* a_i.$$

Since the original problem has a unique solution, this solution is x^*.

2.5.8 Representation and Constraint Relaxation Issues

The presence of duality gap is due to the "problem representation". To illustrate this consider the following problem:

$$\begin{array}{ll} \text{minimize} & -x_2 \\ \text{subject to} & \|x\| \leq x_1 \\ & x \in X,\ X = \{(x_1, x_2) \mid x_2 \geq 0\}. \end{array}$$

Note that its optimal value is $f^* = 0$.

Consider relaxing (assigning a price) to the inequality constraint $\|x\| \leq x_1$. As a result, it can be seen that the corresponding dual problem is such that the dual value $q(\mu)$ is $-\infty$ for any $\mu \geq 0$. Thus, $q^* = -\infty$, while $f^* = 0$. Hence, there is a duality gap.

However, by taking a closer look into the constraint set

$$C = \{x \in \mathbb{R}^2 \mid \|x\| \leq x_1,\ x \in X\},$$

we can see that

$$C = \{x \in \mathbb{R}^2 \mid x_1 \geq 0,\ x_2 = 0\}.$$

Thus the problem is equivalent to

$$\begin{array}{ll} \text{minimize} & -x_2 \\ \text{subject to} & x_1 \geq 0,\ x_2 = 0, \end{array}$$

which is an LP problem. According to the LP strong duality, there is no duality gap for this problem.

As seen from the preceding example, *the duality gap issue is closely related to the "representation" of the constraints*. Thus, often it is advantageous to reformulate a given problem into an equivalent form (if possible) for which some strong duality result is applicable.

Another issue that often arises when dealing with constraint problems is *the existence of multiple choices for a dual problem*. For example, consider the problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_j(x) \leq 0, \quad j = 1, \ldots, m \\ & Ax \leq b,\ Bx = d \\ & x \in X. \end{array}$$

For this problem, there are multiple choices for the relaxation of the constraints (assigning the prices to them). There is no general rule prescribing how to choose the "right one". For example, it is often convenient to keep (not to relax):

(1) Box constraints, i.e., interval constraints on the vector components ($a_i \leq x_i \leq b_i$).

(2) Sign constraints, such as $x \geq 0$ or $x \leq 0$.

(3) Constraints for which the dual function can be easily evaluated.

The choice in (3) requires familiarity with the structure of a given problem.

The following example illustrates choosing a "good dual".

Example 28 *Consider an LP with box constraints:*

$$\begin{array}{ll} \textit{minimize} & c^T x \\ \textit{subject to} & Ax = b \\ & -1 \leq x_i \leq 1 \ i = 1, \ldots, n, \end{array}$$

where A is an $m \times n$ matrix and $b \in \mathbb{R}^m$. If we relax all the constraints, we obtain the dual problem of the form

$$\begin{array}{ll} \textit{maximize} & -b^T\lambda - e^T\mu - e^T\nu \\ \textit{subject to} & c + A^T\lambda + \mu - \nu = 0 \\ & \mu \geq 0, \ \nu \geq 0. \end{array}$$

This dual problem is somewhat complicated. By keeping the box constraints (equivalent to keeping the box constraints in the set X), we obtain the dual of the form:

$$q(\lambda) = \inf_{-1 \leq x_i \leq 1, \ i=1\ldots,n} \left\{ c^T x + \lambda^T (Ax - b) \right\} = -b^T\lambda - \|A^T\lambda + c\|_1.$$

The dual problem is:

$$\textit{maximize} \ \ -b^T\lambda - \|A^T\lambda + c\|_1 \ \ \textit{over } \lambda \in \mathbb{R}^m.$$

There is another useful result, which we refer to "relax-all rule." This rule says that if the strong duality holds for the dual problem resulting from the relaxation of all constraints, then the strong duality holds for a dual problem resulting from any partial relaxation of the constraints. This rule is formalized in the following theorem

Theorem 35 *Consider the following problem*

$$\begin{array}{ll} \textit{minimize} & f(x) \\ \textit{subject to} & g_j(x) \leq 0, \ j = 1, \ldots, m. \end{array}$$

Let the optimal value f^* of the problem be finite. Consider a dual corresponding to the relaxation of all the constraints, and assume that the strong duality holds. Then, there is no duality gap when partially relaxing the constraints.

Note that Theorem 35 applies to the problems that include linear equality constraints. In this case, each linear equality is represented by two linear inequalities.

Chapter 3

Vector Space Methods for Static Optimization

In this chapter, we discuss the basic algorithms for solving convex optimization problems in n-dimensional vector space. The optimization problems we study here are static in the sense that there is no underlying dynamic. In other words, we are given a single instance of problem data that we need to solve optimally.

We start with a linear programming problem and simplex method. We then consider a general convex optimization problem with constrains and study gradient projection methods. Both simplex and gradient projection method are viewed as primal problems. We then focus on dual problems and consider dual methods.

3.1 Simplex Algorithm

We consider a linear programming problem in a standard form and introduce the notion of a basic feasible solution. In Section 2.3.2 of Chapter 2, we have seen that, when LP problem has an optimal solution and the constraint set contains a basic feasible solution, then the LP has an optimal solution which is also a basic feasible solution. Simplex method exploits this fact by searching the basic feasible solutions until an optimal one is found. The search is based on moving from one basic feasible solution to another in the direction of a cost decrease. Here, we discuss precisely how the method operates.

In what follows, we consider an LP problem in the standard form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0, \end{array}$$

where A is an $m \times n$ matrix and $b \in \mathbb{R}^m$. Throughout this section, we assume that $m \leq n$ and that the rows of A are linearly independent. Recall our notation of $[A]_i$ being the i-th row of the matrix A and $[A]^j$ being the j-th column of A.

3.1.1 Optimal Basic Feasible Solutions

Here, we introduce and characterize feasible directions of an LP problem, and provide a characterization of basic feasible solutions.

In particular, we have the following characterization of basic feasible solutions for the standard LP problem.

Theorem 36 *A vector x is a basic feasible solution for the standard LP problem if and only if $Ax = b$ and there exist indices $B(1), \ldots, B(m)$ such that the columns $[A]^{B(1)}, \ldots, [A]^{B(m)}$ are linearly independent and $x_i = 0$ for $i \notin \{B(1), \ldots, B(m)\}$.*

Based on Theorem 36, we can come up with a simple *procedure for generating a basic solution*. In particular we can do this as follows:

(1) Select m linearly independent columns of A, say $[A]^{B(1)}, \ldots, [A]^{B(m)}$.

(2) Set $x_i = 0$ for all $i \notin \{B(1), \ldots, B(m)\}$.

(3) Determine the remaining m components $x_{B(1)}, \ldots, x_{B(m)}$, by solving the system of m equations $Ax = b$.

If the basic solution that is produced by the preceding procedure has nonnegative components, then its is a basic feasible solution. A basic solution with more than $n - m$ components equal to zero is *degenerate*; otherwise, it is *nondegenerate*.

For a basic solution, the variables $x_{B(1)}, \ldots, x_{B(m)}$ are *basic*, and the remaining variables are *nonbasic*. The columns $[A]^{B(1)}, \ldots, [A]^{B(m)}$ are *basic columns*. Note that these columns form a basis in $\mathbb{R}^m$, since they are linearly independent. The index set $\{B(1), \ldots, B(m)\}$ is the set of *basic indices*. Two basis are distinct if they involve different sets of basic indices (the order of the indices does not matter).

By forming a matrix from m basic columns, we obtain an $m \times m$ matrix B, which we refer to as a *basis* matrix. Every basis matrix is invertible, since its columns are linearly independent. Similarly, we define a vector x_B whose components are the basic variables. Specifically, we have

$$B = \begin{bmatrix} [A]^{B(1)} & [A]^{B(2)} & \cdots & [A]^{B(m)} \end{bmatrix}, \qquad x_B = \begin{bmatrix} x_{B(1)} \\ x_{B(2)} \\ \vdots \\ x_{B(m)} \end{bmatrix}.$$

Example 29 *Let the equations $AX = b$ be given by*

$$\begin{bmatrix} 2 & 1 & 0 & 0 & 0 \\ 3 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 1 \\ 1 \end{bmatrix}.$$

Choosing basic index set $\{2, 3, 4, 5\}$, we obtain a basic matrix B to be the 4×4 identity matrix, and the corresponding basic solution is $x = [0\ 2\ 2\ 1\ 1]^T$. Note that this solution is also basic feasible.

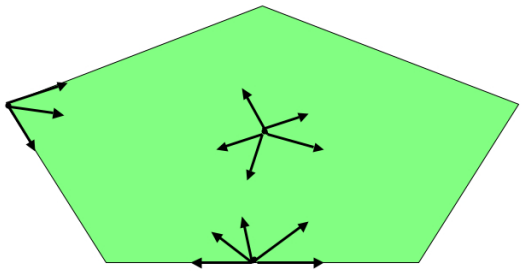

Figure 3.1: Feasible directions at different feasible points.

Choose now the basic index set $\{1, 3, 4, 5\}$. *The corresponding basic matrix* B *is*

$$B = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and its corresponding basic solution is $x = [1\ 0\ -1\ 1\ 1]^T$. *This solution is basic but not feasible since* $x_3 = -1 < 0$.

Given a feasible vector x for an LP problem, we say that a vector d is *a feasible direction at* x when there exists a scalar λ with $\lambda > 0$ such that $x + \lambda d$ is also feasible for the problem. Feasible directions are illustrated in Figure 3.1

Suppose now we are at some feasible vector x and we want to move from x along a feasible direction d to another feasible vector of the form $x + \lambda d$. In order for the vector $x + \lambda d$ to be feasible, we must have $Ax + \lambda Ad = b$. Since x is feasible and $\lambda > 0$, it follows that $Ad = 0$.

Suppose that x is a basic feasible solution to the standard form LP problem. Let B be the corresponding basic matrix, and let x_B be the corresponding vector of basic variables. We then have

$$x_B = B^{-1}b,$$

while the nonbasic variables are all zero. Consider a direction d at x of the form $d_j = 1$ for some nonbasic index j and $d_l =$ for all other nonbasic indices l. For such d to be feasible we must have $Ad = 0$, implying that

$$0 = Ad = \sum_{i=1}^{n} [A]^i \, d_i = \sum_{i=1}^{m} [A]^{B(i)} \, d_{B(i)} + [A]^j = Bd_B + [A]^j.$$

Hence, since B is invertible, we have

$$d_B = -B^{-1}[A]^j.$$

Such direction d as just constructed is *the j-th basic direction*. The preceding construction guarantees that the equality constraints $Ax = b$ are satisfied as me move away from x along the direction d. But how far we can move (if any at all) and maintain the feasibility depends on the degeneracy of x. In particular, there are two possibilities:

(1) If x is nondegenerate, then $x_B > 0$ and for some (possibly small) $\lambda > 0$, we can still have $x_B + \lambda d_B \geq 0$. Hence, the feasibility can be maintained (d is feasible).

(2) If x is degenerate, then we may have for some basic variable $x_{B(i)} = 0$ and the corresponding component $d_{B(i)} < 0$. In this case, all the vectors of the form $x + \lambda d$ with $\lambda > 0$ will be infeasible, i.e., violate nonnegativity constraint for $x_{B(i)}$ (d is infeasible).

We now describe how the cost $c^T x$ changes as we move along a basic direction d. Moving from x to $x + d$ corresponds to the cost change from $c^T x$ to $c^T x + c^T d$. Thus the rate of cost change is $c^T d$. Since

$$d_B = -B^{-1}[A]^j, \quad d_j = 1 \ \text{ for a nonbasic index } j, \quad d_l = 0 \ \text{ for other nonbasic indices } l,$$

it follows that the rate of change along d is

$$c^T d = c_j - c_B^T B^{-1}[A]^j.$$

This is the cost per unit increase in the variable x_j. It plays an important role in the simplex algorithm, so it has a special name.

Definition 6 *Let x be a basic solution with the corresponding basis matrix B. Let c_B be the cost vector associated with the basis matrix B. For every j, the reduced cost $\bar{c}_j$ of the variable x_j is given by*

$$\bar{c}_j = c_j - c_B^T B^{-1}[A]^j.$$

It can be seen that for any basic variable $x_{B(i)}$, the reduced cost $\bar{c}_{B(i)}$ is zero. This follows from the fact $B^{-1}[A]^{B(i)} = e_i$.

We next provide an example for a basic direction and reduced cost.

Example 30 *(Based on Example 3.1 of [11]) Consider the following LP problem:*

$$\begin{array}{ll} \textit{minimize} & x_1 + x_2 - 2x_3 + x_4 \\ \textit{subject to} & x_1 + x_2 + x_3 + x_4 = 2 \\ & 2x_1 + \ \ +3x_3 + 4x_4 = 2 \\ & x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0,\ x_4 \geq 0. \end{array}$$

Let B be the matrix formed from the first two columns of A, i.e.,

$$B = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix}.$$

We let $x_3 = 0$ and $x_4 = 0$, and obtain x_1 and x_2 from the relation $x_B = B^{-1}b$, implying that $x_1 = 1$ and $x_2 = 1$. The resulting vector x is a basic feasible solution and nondegenerate.

Consider the nonbasic variable x_3 and the corresponding basic direction d. We have $d_3 = 1$ and $d_4 = 0$, while d_1 and d_2 are obtained from the relation $d_B = -B^{-1}[A]^3$, yielding $d_1 = -3/2$ and $d_2 = 1/2$. The reduced cost of the variable x_3 is

$$\bar{c}_3 = c_3 - c_B^T B^{-1}[A]^3 = -\frac{3}{2} + \frac{1}{2} - 2 = -3.$$

In line with the optimality conditions of Section 2.3.3, we have the following result.

Theorem 37 *Let x be a basic feasible solution with the corresponding basis matrix B, and let $\bar{c}$ be the corresponding vector of reduced costs. We then have:*

(a) If $\bar{c} \geq 0$, then x is optimal solution.

(b) If x is optimal and nondegenerate, then $\bar{c} \geq 0$.

In light of Theorem 37, a basic solution is optimal if it is feasible and its corresponding reduced cost is nonnegative. We use this to define an optimal basis, as follows.

Definition 7 *A basis matrix B is optimal when $B^{-1}b \geq 0$ and $\bar{c}^T = c^T - c_B^T B^{-1}A \geq 0$.*

Clearly, a basic solution corresponding to an optimal basis is an optimal solution.

3.1.2 Algorithm

Here, we complete the details needed for the development of the simplex method, and present the method in a basic form. We have discussed basic feasible directions, and now we need to see how we choose a good direction and specifically, how we generate a new basic feasible solution from the current one. We first consider a nondegenerate LP problem and then, comment on issues and discuss some modifications when the problem is degenerate.

Nondegenerate Linear Problem

Assume that all feasible solutions of the problem are nondegenerate. Let x be the current basic feasible solution. Also, assume that we have evaluated the reduced cost $\bar{c}_j$ for each of the nonbasic variables. If they are all nonnegative, then by Theorem 37, the solution x is optimal. Otherwise, there is some nonbasic variable x_j with the reduced cost $\bar{c}_j < 0$. For this nonbasic variable, we compute the corresponding (j-th) basic feasible direction,

$$\begin{aligned} d_B = -B^{-1}[A]^j, \qquad & d_j = 1 \ \text{ for the nonbasic index } j, \\ & d_l = 0 \ \text{ for all other nonbasic indices } l. \end{aligned} \tag{3.1}$$

We have $x_j = 0$. Moving from x along d, we generate the vectors $x + \lambda d$ with λ increasing (starting from $\lambda = 0$). When increasing λ, the nonbasic variable x_j increases and becomes positive, while the other nonbasic variables remain unchanged (stay at 0). In this case, we say that x_j *enters the basis*.

The cost of a vector $x + \lambda d$ with $\lambda \geq 0$ is $c^T x + \lambda c^T d$. The cost change resulting from moving from x to some $x + \lambda d$ is equal to $\lambda c^T d$. This is the same as the cost $\lambda\, \bar{c}_j$, which

is negative for $\lambda > 0$. Hence, the cost decreases along d and to find the best feasible vector along d, we should move as far as possible and stay feasible. This corresponds to determining largest $\lambda^* > 0$ such that $x + \lambda^* d$ is feasible.

By the construction of d in Eq. (3.1), we have that $Ad = 0$, and therefore, the vectors $x + \lambda d$ satisfy $AX = b$ for any $\lambda \geq 0$. A vector $x + \lambda d$ may be infeasible only when some of its components are negative. Recall that the nonbasic variables other than x_j will stay at 0, while x_j becomes positive as λ increases from zero. Thus, the only components of $x + \lambda d$ that may become negative as λ increases are the basic variables, which are affected by d_B. There are two possibilities:

Case $d_B \geq 0$. Then, $x_B + \lambda d_B \geq 0$ for all $\lambda > 0$, and the vector $x + \lambda d$ is always feasible for $\lambda \geq 0$. Hence, $\lambda^* = \infty$.

Case $d_B \not\geq 0$. Then, there exists a basic index $B(i)$ such that $d_{B(i)} < 0$. The constraint $x_{B(i)} + \lambda d_{B(i)} \geq 0$ implies that $\lambda \leq -x_{B(i)}/d_{B(i)}$. This inequality should be satisfied for all basic indices $B(i)$ for which $d_{B(i)} < 0$. Therefore,

$$\lambda^* = \min_{\{i \,|\, d_{B(i)} < 0\}} \left\{ -\frac{x_{B(i)}}{d_{B(i)}} \right\}.$$

Observe that $\lambda^* > 0$, since $x_{B(i)} > 0$ (nondegenerate x) and the minimum is taken over the basic indices i corresponding to $d_{B(i)} < 0$.

Suppose we have computed λ^*, and let λ^* be finite. We then move to a new vector $\tilde{x} = x + \lambda^* d$. Note that $\tilde{x}_j = \lambda^*$, since $x_j = 0$ and $d_j = 1$. Let p be the index achieving the minimum in λ^*, i.e.

$$-\frac{x_{B(p)}}{d_{B(p)}} \min_{\{i \,|\, d_{B(i)} < 0\}} \left\{ -\frac{x_{B(i)}}{d_{B(i)}} \right\} = \lambda^*.$$

Then, we have

$$x_{B(p)} + \lambda^* d_{B(p)} = 0,$$

and by the definition of λ^*, we have

$$x_{B(i)} + \lambda^* d_{B(i)} > 0 \qquad \text{for all } i \neq p, \;\; i = 1, \ldots, m, \tag{3.2}$$

where the strict inequality holds by nondegeneracy. Thus, the new vector $\tilde{x} = x + \lambda^* d$ is feasible. Let $\tilde{B}$ be the matrix obtained from the basis matrix B by replacing p-th column of B (which is $[A]^{B(p)}$) with j-th column of A (i.e., $[A]^j$). Equivalently, we replace the basic index set $\{B(1), \ldots, B(m)\}$ with the basic index set corresponding to $\tilde{x}$, which is given by

$$\{B(1), \ldots, B(p-1), j, B(p+1), \ldots, B(m)\}.$$

It can be seen that the new matrix $\tilde{B}$ also has linearly independent columns and thus, the new vector $\tilde{x}$ is a basic feasible solution.

In the preceding procedure, we say that the nonbasic variable x_j *enters the basis*, while the basic variable $x_{B(p)}$ *leaves the basis*. Furthermore, the new basic feasible solution $\tilde{x}$ is

distinct from x and the cost is reduced, i.e., the cost of $\tilde{x}$ is strictly smaller than the cost of x.

The preceding construction is a typical iteration of a simplex method, by which the method moves from one feasible basic solution to another. We next summarize *the steps involved in a typical simplex iteration*:

1. Given a basic feasible solution x and its basis matrix B, compute the reduced costs
$$\bar{c}_j = c_j - c_B^T B[A]^j \qquad \text{for all nonbasic indices } j.$$
2. If these reduced costs are nonnegative, then stop. The current solution x is optimal. Otherwise choose some j for which $\bar{c}_j < 0$.
3. Compute $d_B = -B^{-1}[A]^j$. If $d_B \geq 0$, then $\lambda^* = +\infty$ and the optimal cost is $f^* = -\infty$; terminate. Otherwise, go to step 4.
4. Compute
$$\lambda^* = \min_{\{i \,|\, d_{B(i)} < 0\}} \left\{ -\frac{x_{B(i)}}{d_{B(i)}} \right\}.$$
5. Select p such that $\lambda^* = -x_{B(p)}/d_{B(p)}$. Form a new basis matrix by replacing $[A]^{B(p)}$ with $[A]^j$. The new feasible solution $\tilde{x}$ has basic variables
$$\tilde{x}_j = \lambda^* \qquad \text{and} \qquad \tilde{x}_{B(i)} = x_{B(i)} + \lambda^* d_{B(i)} \qquad \text{for } i \neq p.$$

There are few remaining issues to be discussed, namely, how to initially choose a basic feasible solution, and how to select the variable x_j entering the basis and the variable $x_{B(p)}$ leaving the basis when there are multiple choices.

In general, finding an initial basic solution is not easy. It requires solving an auxiliary linear problem, which is beyond the scope of these notes. The interested reader can find in depth discussion on this issue, for example, in textbook [11], Section 3.5.

In a typical simplex iteration, at Step 2, there may exist several basic indices j for which $\bar{c}_j < 0$. Similarly at step 5, there may be more than one index p for which the minimum in the expression for λ^* is attained. Thus, there may be multiple options for selecting j and p, and there are some rules guiding these selections. The rules are known as *pivoting rules*.

We discuss pivoting rules regarding the choices of the nonbasic variable entering the bases. Some possible choices are:

- Choose a nonbasic variable x_j for which the reduced cost $\bar{c}_j < 0$ is the smallest. This rule chooses the direction d along which the cost decreases at the fastest rate, but this not necessarily yields the largest cost decrease, since the cost decrease depends on how far we move along the chosen direction d.
- Choose a nonbasic variable x_j with $\bar{c}_j < 0$, for which $\lambda^* \bar{c}_j$ is the smallest. This rule chooses the direction d corresponding to the largest cost decrease. However, note that the computational load at each iteration can be high, since we need to compute λ^* for each nonbasic variable x_j with $\bar{c}_j < 0$.

In the nondegenerate case, the simplex method method finds an optimal basic solution in a finite number of iterations, as stated in the following theorem.

Theorem 38 *Assume that every basic feasible solution is nondegenerate, and that the problem is feasible. Then, in finitely many iterations,the simplex method terminates at one of the following possible cases:*

(1) The basis B is optimal and the associated basic feasible solution is optimal.

(2) The direction d is such that $Ad = 0$, $d \geq 0$, and $c^T d < 0$.

Proof. If the algorithm terminates at step 2, then by Theorem 37, the basis B is optimal and its corresponding basic feasible solution is optimal.

If the algorithm terminates at step 3, we then have a direction d with the reduced cost $\bar{c}_j < 0$, and $Ad = 0$ and $d_B \geq 0$. But then, by construction of d [cf. Eq. (3.1)], we have $d \geq 0$. Furthermore, we have $\bar{c}_j = c^T d < 0$. By letting $\lambda \to \infty$, we obtain $c^T(x + \lambda d) \to -\infty$, while the vectors $x + \lambda d$ stay feasible. Hence $f^* = -\infty$.

Note that the algorithm reduces the cost at each iteration. Thus it moves from one feasible basic solution to another, without ever revisiting the same basic feasible solution. Since there are finitely many basic feasible solutions, the algorithm must terminate in a finite number of iterations. ■

Degenerate Linear Problem

The simplex algorithm, as described, can be applied to a linear problem with degenerate basic feasible solutions. However, we may face some new possibilities, such as:

- When the current basic feasible solution x is degenerate, λ^* may be zero, and the new basic feasible solution is the same as x. This can happen, when for some basic variable $x_{B(i)} = 0$ and the corresponding $d_{B(i)} < 0$. However, we can still find a new basis and proceed with the iteration. Theorem 38 still applies.

- When λ^* is positive, it may happen that more than one of the original basic variables becomes 0 [i.e., the inequality in Eq. (3.2) need not be strict anymore]. Among these basic variables, one of them leaves the basis, but the others stay in the basis (at the zero value). Thus, the new basic feasible solution is degenerate.

Changing the basis while still staying at the same basic feasible solution is still worth doing. This is because, eventually, through a sequence of such changes, we may find a feasible direction along which the cost decreases. However, a sequence of such changes may also result in reaching the same basic feasible solution that we have started with. Thus, is known as *cycling*, which can happen in the presence of degenerate basic feasible solutions. There are some bookkeeping rules that track the variables entering the basis and prevent cycling. For more on this, see for example [11], Section 3.4.

Other Versions of the Simplex Algorithm

There are versions of the simplex method that exploit the dual LP. In particular, as seen in Section 2.5.3 the dual of the standard LP is given by

$$\begin{array}{ll} \text{maximize} & b^T p \\ \text{subject to} & A^T p \leq c. \end{array} \tag{3.3}$$

The primal optimality condition $c^T - c_B^T B^{-1} A \geq 0$, can be written in terms of the dual variable p, by letting $p^T = c_B^T B^{-1}$. Then, the primal optimality condition reduces to $p^T A \leq c^T$, which is the feasibility condition for the dual problem (3.3). Thus, the simplex method that generates basic feasible solutions, maintains the primal feasibility and strives for dual feasibility which is attained at the termination of the algorithm when f^* is finite [cf. Theorem 38]. This method is also known as *primal simplex method*.

Alternatively, the simplex method can start at a dual feasible solution and strive to achieve primal feasibility. Such a simplex algorithm is known as *dual simplex method*. More on the dual simplex algorithm can be found in [11], Section 4.5.

3.2 Gradient Projection Method

In this section, we consider a standard gradient projection method as applied to a convex optimization problem subject to a simple set of constraints. In particular, we focus on the following problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in X, \end{array} \tag{3.4}$$

where $X \subseteq \mathbb{R}^n$ and $f : X \to \mathbb{R}$. Throughout this section, we use the following assumption for the set X and the function f.

Assumption 4 *The set X is convex and closed. The function f is convex and differentiable over the set X. The optimal value f^* of the problem (3.4) is finite.*

The gradient projection method is an iterative method that starts with some initial feasible vector $x_0 \in X$, and generates the next iterate by taking a step along the negative gradient direction $-\nabla f(x_k)$ of f at x_k and then, by projecting on the set X to maintain feasibility. Formally, a typical iteration of the gradient projection method is given by

$$x_{k+1} = P_X \left[x_k - \alpha_k \nabla f(x_k) \right], \tag{3.5}$$

where the scalar $\alpha_k > 0$ is a stepsize and x_k is the current iterate, and $P_X[y]$ is the projection of a vector y on the set X. By the Projection Theorem [see Theorem 22 of Section 2.3.4], the projection exists and it is unique since X is closed and convex. A typical iteration of the gradient projection method is illustrated in Figure 3.2.

We assume that the set X is simple so that the projection on X is easy. Examples of such sets X include a nonnegative orthant, a box, and a ball. When X is the nonnegative orthant

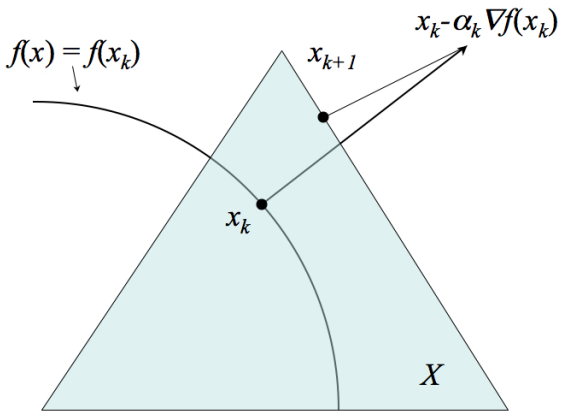

Figure 3.2: An iteration of the gradient projection method.

$\mathbb{R}^n_+$, then the projection on X decomposes into projections per coordinate. In particular, in this case, the projection $P_X[x]$ is the vector x^+ with components $x_i^+ = \max\{x_i, 0\}$. When X is the box,

$$X = \{x \in \mathbb{R}^n \mid a_i \leq x_i \leq b_i \ \text{ for all } i\},$$

again the projection on X decomposes into projections per coordinate, and the components of the projection $P_X[x]$ vector are given by

$$[P_X[x]]_i = \begin{cases} a_i & \text{if } x_i < a_i \\ x_i & \text{if } a_i \leq x_i \leq b_i \\ b_i & \text{if } b_i < x_i. \end{cases}$$

We consider the gradient projection method with several stepsize rules:

- *Constant stepsize*, where for some $\alpha > 0$, we have $\alpha_k = \alpha$ for all k.
- *Diminishing stepsize*, where $\alpha_k \to 0$ and $\sum_k \alpha_k = \infty$.
- *Polyak's stepsize*, where $\alpha_k = \frac{f(x_k) - f^*}{\|\nabla f(x_k)\|^2}$.
- *Modified Polyak's stepsize*, where $\alpha_k = \frac{f(x_k) - \hat{f}_k}{\|\nabla f(x_k)\|^2}$ and $\hat{f}_k = \min_{0 \leq j \leq k} f(x_j) - \delta$ for some scalar $\delta > 0$.

The constant stepsize rule is suitable when we are interested in finding an approximate solution to the problem (3.4). Diminishing stepsize rule is an off-line rule and is typically used with $\alpha_k = \frac{c}{k+1}$ or $\frac{c}{\sqrt{k+1}}$ for some $c > 0$. The constant and the diminishing stepsize are also well suited for some distributed implementations of the method.

Let us mention that Polyak's stepsize in general form involves a parameter γ_k. In particular, the stepsize is given by

$$\alpha_k = \gamma_k \frac{f(x_k) - f^*}{\|\nabla f(x_k)\|^2},$$

where γ_k is bounded away from zero and away from 2, i.e., $0 < \underline{\gamma} \leq \gamma_k \leq \overline{\gamma} < 2$ for all k. Here, however, we simply use $\gamma_k = 1$ for all k, since in general it is not clear what should guide the choice for γ_k at each iteration k. Furthermore, all the results that we establish here, also hold for the Polyak's rule in general form.

Polyak's stepsize is suitable when the optimal function value f^* is known and the evaluation of the function f is possible and "easy". There are some optimization problems for which f^* is known and the algorithm is applied to generate an optimal solution. This is the case, for example, in the feasibility problem (2.8) of Section 2.3.1, where $f^* = 0$.

The modified Polyak's stepsize is a simple adaptation of the Polyak's stepsize to accommodate the situations when f^* is not known. Note that the modified Polyak's stepsize is using a fixed positive parameter δ and, as we will see, the method with this stepsize exhibits behavior similar to that of the method with the constant stepsize, thus only generating an approximate solution. There exist other more complex modifications of the Polyak's rule for which the method generates optimal solutions. Some of these adaptations can be found in [22] and [9] [and the references therein] although for the subgradient method and the incremental implementation. However, the results shown there still hold by replacing the subgradient with a gradient, and by setting $m = 1$ in the incremental implementation.

There are some other stepsize rules that use a line search. These have been developed for the gradient method when $X = \mathbb{R}^n$, but can be adjusted for the gradient projection method [when $X \neq \mathbb{R}^n$]. We comment on these in Section 3.2.4.

In what follows, we study the convergence properties of the method under the stepsize rules described above. We note that the gradient projection method does not necessarily generate iterates with decreasing cost. What makes the method work is the property that at each new iteration, the method either decreases the function value or decreases the distance to the optimal set. This property is captured in the following lemma, providing a basis for the analysis of the method.

Lemma 12 *Let Assumption 4 hold. Let $y \in X$ be arbitrary but fixed. Then, for the gradient projection method with any stepsize rule, we have for all k,*

$$\|x_{k+1} - y\|^2 \leq \|x_k - y\|^2 - 2\alpha_k(f(x_k) - f(y)) + \alpha_k^2\|\nabla f(x_k)\|^2.$$

Proof. Since $y \in X$ and the projection is a nonexpansive mapping [see Theorem 23 of Section 2.3.4], it follows that for any k,

$$\|x_{k+1} - y\|^2 \leq \|x_k - \alpha_k \nabla f(x_k) - y\|^2 = \|x_k - y\|^2 - 2\alpha_k \nabla f(x_k)^T(x_k - y) + \alpha_k^2\|\nabla f(x_k)\|^2.$$

By the convexity of f, we have [see Theorem 12 of Section 2.2.1] for any k and any $y \in X$,

$$f(x_k) - f(y) \leq \nabla f(x_k)^T(x_k - y).$$

The desired relation follows by combining the preceding two inequalities. ■

3.2.1 Convergence for Constant and Diminishing Rule

In our analysis in this section, we use an additional assumption on gradients of f, namely, we assume that the gradients are bounded uniformly over X.

Assumption 5 *There exists a constant $L > 0$ such that*

$$\|\nabla f(x_k)\| \leq L \qquad \text{for all } x \in X.$$

This assumption is satisfied for example, when X is a compact set and f is continuously differentiable over X. Under this assumption, for the method with the constant stepsize, we have the following result.

Theorem 39 *Let Assumptions 4 and 5 hold. Then, for the gradient projection method with the constant stepsize α, we have*

$$\liminf_{k\to\infty} f(x_k) \leq f^* + \frac{\alpha L^2}{2}.$$

Proof. To arrive at a contradiction, assume that the given relation does not hold, i.e., assume that

$$\liminf_{k\to\infty} f(x_k) > f^* + \frac{\alpha L^2}{2}.$$

Then, for some sufficiently small $\epsilon > 0$, we have

$$f(x_k) \geq f^* + \frac{\alpha L^2}{2} + 2\epsilon \qquad \text{for all } k.$$

The function f is continuous over X, so that there exists $\hat{y} \in X$ such that $f(\hat{y}) = f^* + \epsilon$, implying that

$$f(x_k) - f(\hat{y}) \geq \frac{\alpha L^2}{2} + \epsilon \qquad \text{for all } k.$$

By using the relation of Lemma 12 with $\alpha_k = \alpha$ and $y = \hat{y}$, we obtain for all k,

$$\begin{aligned} \|x_{k+1} - \hat{y}\|^2 &\leq \|x_k - \hat{y}\|^2 - 2\alpha(f(x_k) - f(\hat{y})) + \alpha^2\|\nabla f(x_k)\|^2 \\ &\leq \|x_k - \hat{y}\|^2 - 2\alpha\left(\frac{\alpha L^2}{2} + \epsilon\right) + \alpha^2 L^2, \end{aligned}$$

where in the last inequality we use the uniform boundedness of the gradients. Hence

$$\|x_{k+1} - \hat{y}\|^2 \leq \|x_k - \hat{y}\|^2 - 2\alpha\epsilon \qquad \text{for all } k,$$

and by summing the preceding inequalities over k, we obtain

$$\|x_k - \hat{y}\|^2 \leq \|x_0 - \hat{y}\|^2 - 2k\alpha\epsilon. \tag{3.6}$$

However, the preceding relation fails to hold for sufficiently large k. Therefore, we must have

$$\liminf_{k\to\infty} f(x_k) \leq f^* + \frac{\alpha L^2}{2}.$$

■

When the set X is compact, and we know the maximal distance d of the set X, i.e., $d = \max_{x,y\in X} \|x - y\|$, we can use relation (3.6) to compute an upper bound of the minimal

number of iterations N needed to guarantee that the error level αL^2 is achieved, i.e., to guarantee that

$$\min_{0\leq k\leq N} f(x_k) \leq f^* + \alpha L^2.$$

In particular, by choosing $\epsilon = \frac{\alpha L^2}{2}$ and using relation (3.6), we can see that the number N is given by

$$N = \left\lfloor \frac{d^2}{\alpha^2 L^2} \right\rfloor.$$

Theorem 40 *Let Assumptions 4 and 5 hold. Then, for the gradient projection method with the diminishing stepsize α_k, we have*

$$\liminf_{k\to\infty} f(x_k) = f^*.$$

Proof. To obtain a contradiction, assume that the given relation does not hold, i.e., assume that for some sufficiently small $\epsilon > 0$,

$$\liminf_{k\to\infty} f(x_k) > f^* + 2\epsilon.$$

The function f is continuous over X, so there exists $\hat{y} \in X$ such that $f(\hat{y}) = f^* + \epsilon$, implying that

$$f(x_k) - f(\hat{y}) \geq \epsilon \qquad \text{for all } k.$$

By using the relation of Lemma 12 with $y = \hat{y}$, we obtain for all k,

$$\begin{aligned} \|x_{k+1} - \hat{y}\|^2 &\leq \|x_k - \hat{y}\|^2 - 2\alpha_k(f(x_k) - f(\hat{y})) + \alpha_k^2 \|\nabla f(x_k)\|^2 \\ &\leq \|x_k - \hat{y}\|^2 - 2\alpha_k\epsilon + \alpha_k^2 L^2, \end{aligned}$$

where in the last inequality we also use the uniform boundedness of the gradients. Hence

$$\|x_{k+1} - \hat{y}\|^2 \leq \|x_k - \hat{y}\|^2 - \alpha\left(2\epsilon - \alpha_k L^2\right) \qquad \text{for all } k.$$

Since $\alpha_k \to 0$, there exists $\hat{k}$ such that $\epsilon \geq \alpha_k L^2$, implying that

$$\|x_{k+1} - \hat{y}\|^2 \leq \|x_k - \hat{y}\|^2 - \alpha\epsilon \qquad \text{for all } k \geq \hat{k}.$$

By summing the preceding inequalities over $k = \hat{k}, \ldots, K$, we obtain

$$\|x_K - \hat{y}\|^2 \leq \|x_{\hat{k}} - \hat{y}\|^2 - \epsilon \sum_{i=\hat{k}}^{K-1} \alpha_i. \tag{3.7}$$

By letting $K \to \infty$ and using the fact $\sum_k \alpha_k = +\infty$, we see that the left hand side of relation (3.7) tends to $-\infty$, while its right hand side is nonnegative - a contradiction. Therefore, we must have $\liminf_{k\to\infty} f(x_k) = f^*$. ■

When the optimal set X^* of the problem in Eq. (3.4) is nonempty and we impose a stronger condition on the diminishing stepsize, namely $\sum_k \alpha_k^2 < \infty$, we can show that the whole sequence $\{x_k\}$ generated by the gradient projection method converges to some optimal solution x^*. This result is formally given in the following theorem.

Theorem 41 *Let Assumptions 4 and 5 hold. Also, assume that the optimal set X^* of the problem (3.4) is nonempty. Suppose that the stepsize is such that $\sum_k \alpha_k = \infty$ and $\sum_k \alpha_k^2 < \infty$. Then, for the iterate sequence $\{x_k\}$ generated by the gradient projection method with such a stepsize α_k, we have*

$$\lim_{k\to\infty} \|x_k - x^*\| = 0 \qquad \text{for some } x^* \in X^*.$$

Proof. By using the relation of Lemma 12 with $y = x^*$, we obtain for all k,

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &\leq \|x_k - x^*\|^2 - 2\alpha_k(f(x_k) - f^*) + \alpha_k^2 \|\nabla f(x_k)\|^2 \\ &\leq \|x_k - x^*\|^2 + \alpha_k^2 L^2, \end{aligned}$$

where in the last inequality we use $f(x_k) \geq f^*$ and the uniform boundedness of the gradients. By summing the preceding inequalities over $k = \hat{k}, \ldots, K$ for some arbitrary $\hat{k}$ and K with $\hat{k} < K$, we obtain

$$\|x_K - x^*\|^2 \leq \|x_{\hat{k}} - x^*\|^2 + L^2 \sum_{k=\hat{k}}^{K-1} \alpha_k^2.$$

Therefore,

$$\limsup_{K\to\infty} \|x_K - x^*\|^2 \leq \|x_{\hat{k}} - x^*\|^2 + L^2 \sum_{k=\hat{k}}^{\infty} \alpha_k^2. \tag{3.8}$$

By letting $\hat{k} = 0$, we see that the sequence $\{\|x_k\|\}$ is bounded and hence, it has at least one accumulation point. Since $\sum_k \alpha_k^2 < \infty$, it follows that $\alpha_k \to 0$. By Theorem 40 we have $\liminf_{k\to\infty} f(x_k) = f^*$. Thus, one of the accumulation points of $\{x_k\}$ must belong to the optimal set X^*. Let $\{x_{k_i}\}$ be a subsequence such that $x_{k_i} \to \hat{x}^*$ with $\hat{x}^* \in X^*$.

By setting $x^* = \hat{x}^*$ and $\hat{k} = k_i$ in Eq. (3.8), and by letting $i \to \infty$, we obtain

$$\limsup_{K\to\infty} \|x_K - \hat{x}^*\|^2 \leq \lim_{i\to\infty} \|x_{k_i} - \hat{x}^*\|^2 + \lim_{i\to\infty} \sum_{k=k_i}^{\infty} \alpha_k^2 = 0,$$

where we have used $\lim_{i\to\infty} \|x_{k_i} - \hat{x}^*\| = 0$ and

$$\lim_{i\to\infty} \sum_{k=k_i}^{\infty} \alpha_k^2 = 0,$$

which follows from $\sum_k \alpha_k^2 < \infty$. Hence, the whole sequence $\{x_k\}$ converges to $\hat{x}^* \in X^*$. ■

3.2.2 Convergence for Polyak's Stepsize and its Modification

For the case when the optimal value f^* is known Polyak in [24] had suggested the stepsize rule of the form

$$\alpha_k = \frac{f(xk) - f^*}{\|\nabla f(x_k)\|^2}.$$

This stepsize rule has also been studied by Shor in [29, 30] and Bränlund [14].

For the convergence analysis of the Polyak's stepsize, we do not need the assumption on uniform boundedness of the gradients. This is because, the stepsize "normalizes" the directions that the method is using. In particular, the direction that the method is using for the Polyak's stepsize is given by

$$-\alpha_k \nabla f(x_k) = \frac{f(x_k) - f^*}{\|\nabla f(x_k)\|} \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|}.$$

We have the following convergence result.

Theorem 42 *Let Assumption 4 hold. Also, assume that the optimal set X^* of the problem (3.4) is nonempty. Then, for the iterate sequence $\{x_k\}$ generated by the gradient projection method with Polyak's stepsize α_k, we have*

$$\lim_{k\to\infty} \|x_k - x^*\| = 0 \qquad \text{for some } x^* \in X^*.$$

Proof. By using the relation of Lemma 12 with $y = x^*$, we obtain for all k and any $x^* \in X^*$,

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &\leq \|x_k - x^*\|^2 - 2\alpha_k (f(x_k) - f^*) + \alpha_k^2 \|\nabla f(x_k)\|^2 \\ &= \|x_k - x^*\|^2 - \frac{(f(x_k) - f^*)^2}{\|\nabla f(x_k)\|^2}, \end{aligned}$$

where in the last inequality we use the stepsize expression $\alpha_k = \frac{f(x_k)-f^*}{\|\nabla f(x_k)\|^2}$. Therefore, it follows that for any $x^* \in X^*$, and any k and s with $k > s$,

$$\|x_{k+1} - x^*\|^2 \leq \|x_s - x^*\|^2 - \sum_{i=s}^{k} \frac{(f(x_i) - f^*)^2}{\|\nabla f(x_i)\|^2}. \tag{3.9}$$

By letting $s = 0$, we see that the iterate sequence $\{x_k\}$ is bounded and therefore, it has an accumulation point. Furthermore, note that Eq. (3.9) implies that

$$\sum_{i=0}^{\infty} \frac{(f(x_i) - f^*)^2}{\|\nabla f(x_i)\|^2} \leq \|x_0 - x^*\|^2 < \infty. \tag{3.10}$$

Suppose that none of the accumulation points of $\{x_k\}$ belongs to the optimal set X^*. Then, for some small scalar $\epsilon > 0$, we have

$$f(x_k) > f^* \qquad \text{for all } k.$$

Since $\{x_k\}$ is bounded, so is the gradient sequence $\{\nabla f(x_k)\}$, i.e., there is a scalar $c > 0$ such that

$$\|\nabla f(x_k)\| \leq c \qquad \text{for all } k.$$

Therefore,

$$\frac{(f(x_i) - f^*)^2}{\|\nabla f(x_i)\|^2} \geq \frac{\epsilon^2}{c^2} \qquad \text{for all } k.$$

By summing the preceding relations over k, we obtain

$$\sum_{i=0}^{\infty} \frac{(f(x_i) - f^*)^2}{\|\nabla f(x_i)\|^2} \geq \sum_{i=0}^{\infty} \frac{\epsilon^2}{c^2} = \infty,$$

thus contradicting the relation in Eq. (3.10). Hence, every accumulation point of $\{x_k\}$ must belong to the set X^*.

Let $\hat{x}^*$ be an accumulation point of the sequence $\{x_k\}$, and let $\{x_{k_j}\}$ be a subsequence of $\{x_k\}$. By letting $x^* = \hat{x}^*$ and $s = k_j$ in Eq. (3.9), we obtain for all $k > k_j$.

$$\|x_{k+1} - \hat{x}^*\|^2 \leq \|x_{k_j} - \hat{x}^*\|^2 - \sum_{i=k_j}^{k} \frac{(f(x_i) - f^*)^2}{\|\nabla f(x_i)\|^2}.$$

Therefore,

$$\limsup_{k\to\infty} \|x_{k+1} - \hat{x}^*\|^2 \leq \|x_{k_j} - \hat{x}^*\|^2 - \sum_{i=k_j}^{\infty} \frac{(f(x_i) - f^*)^2}{\|\nabla f(x_i)\|^2}.$$

By letting $j \to \infty$, and by using the relation $\|x_{k_j} - \hat{x}^*\| \to 0$ and Eq. (3.10), we obtain

$$\limsup_{k\to\infty} \|x_{k+1} - \hat{x}^*\|^2 \leq \lim_{j\to\infty} \left(\|x_{k_j} - \hat{x}^*\|^2 - \sum_{i=k_j}^{\infty} \frac{(f(x_i) - f^*)^2}{\|\nabla f(x_i)\|^2} \right) = 0,$$

thus showing that the entire sequence converges to $\hat{x}^* \in X^*$. ■

The relation shown at the beginning of the proof of Theorem 42, namely,

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \frac{(f(x_k) - f^*)^2}{\|\nabla f(x_k)\|^2} \qquad \text{for all } k \text{ and any } x^* \in X^* \tag{3.11}$$

will be important in our assessment of the convergence rate of the gradient projection method in Section 3.2.3.

We now provide a convergence result for the modified Polyak's stepsize. Note that for this stepsize, we have

$$\alpha_k = \frac{f(x_k) - \hat{f}_k}{\|\nabla f(x_k)\|^2} = \frac{f(x_k) - \min_{0\leq j\leq k} f(x_j) + \delta}{\|\nabla f(x_k)\|^2} \geq \frac{\delta}{\|\nabla f(x_k)\|^2}.$$

As we will see, this stepsize remains bounded away from zero, so the convergence result is similar to that of Theorem 39.

Theorem 43 *Let Assumption 4 hold. Then, for the iterate sequence $\{x_k\}$ generated by the gradient projection method with modified Polyak's stepsize α_k, we have*

$$\liminf_{k\to\infty} f(x_k) \leq f^* + \delta.$$

Proof. To obtain a contradiction, assume that the given relation does not hold, i.e., assume that for some sufficiently small $\epsilon > 0$,

$$\liminf_{k\to\infty} f(x_k) > f^* + \delta + \epsilon.$$

The function f is continuous over X, so there exists $\hat{y} \in X$ such that $f(\hat{y}) = f^* + \epsilon$, implying that

$$f(x_k) \geq f(\hat{y}) + \delta \qquad \text{for all } k.$$

Therefore

$$\hat{f}_k = \min_{0\leq j\leq k} f(x_j) - \delta \geq f(\hat{y}) \qquad \text{for all } k,$$

implying that

$$f(x_k) - f(\hat{y}) \geq f(x_k) - \hat{f}_k \qquad \text{for all } k. \tag{3.12}$$

By using the relation of Lemma 12 with $y = \hat{y}$, we obtain for all k,

$$\begin{aligned} \|x_{k+1} - \hat{y}\|^2 &\leq \|x_k - \hat{y}\|^2 - 2\alpha_k(f(x_k) - f(\hat{y})) + \alpha_k^2\|\nabla f(x_k)\|^2 \\ &\leq \|x_k - \hat{y}\|^2 - 2\alpha_k(f(x_k) - \hat{f}_k) + \alpha_k^2\|\nabla f(x_k)\|^2, \end{aligned}$$

where the last inequality is obtained by using the relation (3.12). Using the stepsize expression $\alpha_k = \frac{f(x_k)-\hat{f}_k}{\|\nabla f(x_k)\|^2}$, we see that for all k,

$$\|x_{k+1} - \hat{y}\|^2 \leq \|x_k - \hat{y}\|^2 - \frac{(f(x_k) - \hat{f}_k)^2}{\|\nabla f(x_k)\|^2}. \tag{3.13}$$

By summing the preceding inequalities, we obtain for all k,

$$\|x_{k+1} - x^*\|^2 \leq \|x_0 - x^*\|^2 - \sum_{i=0}^{k} \frac{(f(x_i) - \hat{f}_k)^2}{\|\nabla f(x_i)\|^2}. \tag{3.14}$$

In particular, from the relation (3.14) it follows that the iterate sequence $\{x_k\}$ is bounded and that

$$\sum_{i=0}^{\infty} \frac{(f(x_i) - \hat{f}_k)^2}{\|\nabla f(x_i)\|^2} \leq \|x_0 - x^*\|^2 < \infty. \tag{3.15}$$

Since $\{x_k\}$ is bounded, so is the gradient sequence $\{\nabla f(x_k)\}$, i.e., there is a scalar $c > 0$ such that

$$\|\nabla f(x_k)\| \leq c \qquad \text{for all } k.$$

Therefore,

$$\frac{(f(x_i) - \hat{f}_k)^2}{\|\nabla f(x_i)\|^2} \geq \frac{\delta^2}{c^2} \qquad \text{for all } k,$$

implying that

$$\sum_{i=0}^{\infty} \frac{(f(x_i) - f^*)^2}{\|\nabla f(x_i)\|^2} \geq \sum_{i=0}^{\infty} \frac{\delta^2}{c^2} = \infty,$$

contradicting the relation in Eq. (3.15). Hence, we must have $\liminf_{k\to\infty} f(x_k) \leq f^* + \delta$. ■

When the set X is compact, and we know the maximal distance d of the set X, i.e., $d = \max_{x,y\in X} \|x - y\|$, we can use relation (3.13) to compute an upper bound of the minimal number of iterations N needed to guarantee that the error level δ is achieved, i.e., to guarantee that

$$\min_{0\leq k\leq N} f(x_k) \leq f^* + \delta.$$

In particular, by using relation (3.6), we can see that the number N is given by

$$N = \left\lfloor \frac{d^2 L^2}{\delta^2} \right\rfloor,$$

where L is an upper bound on the gradient norms $\|\nabla f(x_k)\|$.

3.2.3 Convergence Rate

The convergence rate of the projection gradient method is at best linear. The linear convergence is attained with Polyak's stepsize and for an objective function with *a sharp set of minima*. In particular, we say that f has *a sharp set of minima over* X when for some scalar $\mu > 0$,

$$f(x) - f^* \geq \mu \operatorname{dist}(x, X^*) \qquad \text{for all } x \in X, \tag{3.16}$$

where $\operatorname{dist}(x, Y)$ is the distance from the vector x to the set Y. It can be seen that a (polyhedral) function has a sharp set of minima. Let f be *a polyhedral function*, i.e., a function of the form

$$f(x) = \max_{1\leq i\leq m} \{a_i^T x + b\},$$

where $b_i \in \mathbb{R}$ and $a_i \in \mathbb{R}^n$ with $a_i \neq 0$ for all i. Then, it can be seen that f satisfies the relation for a sharp set of minima with $\mu = \min_{1\leq i\leq m} \|a_i\|$, provided that X^* is not empty.

In following theorem, we establish a linear convergence rate for the gradient projection method. In the proof of the theorem, we use relation (3.11) mentioned after the proof of Theorem 42.

Theorem 44 *Let Assumption 4 hold and let the optimal set X^* be nonempty. Also, assume that f has a sharp set of minima over X [cf. Eq. (3.16)]. Then, the gradient projection method with Polyak's stepsize α_k converges linearly, i.e., we have*

$$\operatorname{dist}(x_k, X^*)^2 \leq \left(1 - \frac{\mu^2}{L^2}\right)^k \operatorname{dist}(x_0, X^*)^2 \qquad \textit{for all } k,$$

where L is the gradient norm bound uniform over X [$\|\nabla f(x)\| \leq L$ for all $x \in X$].

Proof. We start with the relation (3.11), i.e.,

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \frac{(f(x_k) - f^*)^2}{\|\nabla f(x_k)\|^2} \qquad \text{for all } k \text{ and any } x^* \in X^*.$$

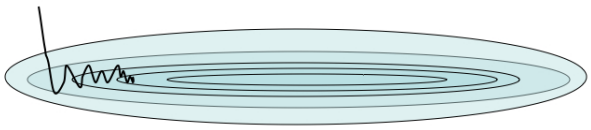

Figure 3.3: A zig-zagging behavior of the gradient method when the level sets of the function f are prolonged.

It follows that the sequence $\{x_k\}$ is bounded, so that the gradient sequence $\{\nabla f(x_k)\}$ is also bounded. Let L be an upper bound on the gradient norms. By taking the minimum with respect to $x^* \in X^*$ in both sides of the preceding relation, and also using the gradient boundedness, we obtain

$$\operatorname{dist}(x_{k+1}, X^*)^2 \leq \operatorname{dist}(x_k, X^*)^2 - \frac{(f(x_k) - f^*)^2}{L^2} \qquad \text{for all } k.$$

Using the sharp minima relation (3.16), we obtain

$$\operatorname{dist}(x_{k+1}, X^*)^2 \leq \operatorname{dist}(x_k, X^*)^2 - \frac{\mu^2}{L^2} \operatorname{dist}(x_k, X^*)^2 = \left(1 - \frac{\mu^2}{L^2}\right) \operatorname{dist}(x_k, X^*)^2 \qquad \text{for all } k,$$

which implies the desired result. ■

Linear convergence rate is not the best known rate. There are methods with superlinear (quadratic) rate such as Newton method and the interior point method, which uses Newton's directions. These methods, however, require f to be twice differentiable among other conditions.

The potential drawback of the projection gradient method is that it can be very slow when the gradient directions are almost perpendicular to the directions pointing toward the optimal set X^*, corresponding to

$$\nabla f(x_k)^T (x_k - x^*) \approx 0.$$

In this case, the method may exhibit zig-zag behavior, depending on the initial iterate x_0. This may happen in particular when the level sets of the function f are prolonged in certain directions (known as ill-posedness), as illustrated in Figure 3.3 [for $X = \mathbb{R}^n$].

3.2.4 Non-Projected Gradient

When the problem (3.4) is unconstrained i.e., $X = \mathbb{R}^n$, the gradient projection method reduces to *unconstrained gradient method*, where

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) \qquad \text{for all } k,$$

which is also known as *steepest descent method*. In this method, we can use stepsizes based on line search, such as *the exact line search stepsize*, where

$$\alpha_k = \operatorname*{Argmin}_{\alpha>0} f(x_k - \alpha \nabla f(x_k)),$$

or *backtracking line search* [also known as *Armijo stepsize*]. What is more surprising, we can use a carefully selected constant step and still have the convergence of the method, as seen from the following theorem.

Theorem 45 *Let Assumption 4 hold with $X = \mathbb{R}^n$. Let the gradient mapping of the function f be Lipschitz continuous, i.e.,*

$$\|\nabla f(x) - \nabla f(y) \leq L\|x - y\| \qquad \text{for all } x, y \in \mathbb{R}^n.$$

Let the stepsize α_k be such that for some $\alpha > 0$,

$$\alpha \leq \alpha_k < \frac{2}{L} \qquad \text{for all } k.$$

Then, every accumulation point of the iterate sequence $\{x_k\}$ generated by the unconstrained gradient method with this stepsize is an optimal solution.

In practical implementations, in order to use the constant stepsize with guaranteed convergence, we need to have the Lipschitz constant L available, or some upper bound for L. The analogous result to that of Theorem 45 holds for the gradient projection method, under Assumption 4 and assuming that the gradient is Lipschitz continuous over X, i.e.,

$$\|\nabla f(x) - \nabla f(y) \leq L\|x - y\| \qquad \text{for all } x, y \in X.$$

This result for a more general nonconvex function f can be found in [5], Proposition 2.3.2.

The *backtracking line search* determines the stepsize α_k by searching for a point on the line segment connecting x_k and $x_k - \nabla f(x_k)$ that has a sufficient decrease. The search is performed by successive halving of the intermediate segments. Specifically, the stepsize α_k is determined through the following procedure: Parameters $\sigma \in (0, 1]$ and $\beta \in (0, 1)$ are selected when the algorithm is initiated. At the current iterate x_k, we perform the following:

Backtracking Line Search

1. Set $i = 1$ and $t_i = 1$.

2. *Sufficient Descent Test* If

$$f(x_k - t_i \nabla f(x_k)) \leq f(x_k) - \sigma t_i \|\nabla f(x_k)\|^2,$$

   then stop. Set the stepsize $\alpha_k = t_i$. Otherwise, go to step 3.

2. Increase i by 1 and set $t_{i+1} = \beta t_i$. Go to step 2.

When function f is continuously differentiable, by the first-order Taylor expansion at any x, we have

$$f(x - \alpha\nabla f(x)) = f(x) - \alpha\|\nabla f(x)\|^2 + o(\alpha) = f(x) - \alpha\left(\|\nabla f(x)\|^2 - \frac{o(\alpha)}{\alpha}\right).$$

Since $\lim_{\alpha\to 0}\frac{o(\alpha)}{\alpha} = 0$, it follows that for sufficiently small $\alpha > 0$, we have

$$\|\nabla f(x)\|^2 - \frac{o(\alpha)}{\alpha} > 0.$$

This guarantees that the backtracking line search finds α_k in a finite number of trials i.

At every iteration of the unconstrained gradient method with the backtracking rule, we have

$$x_{k+1} = x_k - \alpha_k\nabla f(x_k), \tag{3.17}$$

with α_k such that the sufficient descent is guaranteed, i.e.,

$$f(x_{k+1}) \le f(x_k) - \sigma\alpha_k\|\nabla f(x_k)\|^2, \tag{3.18}$$

with $a_k = \beta^{i_k}$ for some integer $i_k \ge 0$

For the gradient method with backtracking line search, we have the following result. Its proof for a more general nonconvex function [using $\sigma \in (0, 2)$[1]] can be found for example in [5], Section 1.2.2.

Theorem 46 *Let Assumption 4 hold with $X = \mathbb{R}^n$. Then, every accumulation point of the iterate sequence $\{x_k\}$ generated by the unconstrained gradient method with backtracking line search stepsize is an optimal solution.*

The backtracking line search can be modified to be used for the constrained minimization problem [$X \neq \mathbb{R}^n$]. In this case, the line search is replaced with the "arc" search, as follows: at the current iterate x_k, we determine the stepsize by:

Backtracking Arc Search

1. Set $i = 1$ and $t_i = 1$.

2. *Sufficient Descent Test* If

$$f\left(P_X[x_k - t_i\nabla f(x_k)]\right) \le f(x_k) - \sigma t_i\|\nabla f(x_k)\|^2,$$

then stop. Set the stepsize $\alpha_k = t_i$. Otherwise, go to step 3.

2. Increase i by 1 and set $t_{i+1} = \beta t_i$. Go to step 2.

[1]When f is convex, we cannot use $\sigma > 1$. This follows from linearization property for a convex function $f(x_k) + \nabla f(x_k)^T(x_{k+1} - x_k) \le f(x_{k+1})$, and the relations (3.17)–(3.18).

In this case, the result of Theorem 46 holds for the gradient projection method and the backtracking arc search, as seen for example, in Proposition 2.3.3 of [5] for a general non-convex function.

The backtracking line search can be efficiently used when the function evaluation $f(x_k)$ are "easy". However, if these evaluations are expensive or impossible such as in some distributed systems[2], then one should consider an alternative stepsize.

Finally, note that the results of Theorems 46 and 45 do not guarantee the existence of the accumulation points of $\{x_k\}$. These results merely say that: if accumulation points exist, then they are optimal. Let us also note that the results of Theorems 46 and 45, as well as the results of their extensions to the gradient projection method, hold only for differentiable functions[3]. This is in contrast with the results of Theorems 39–43, which hold when f is convex but not necessarily nondifferentiable. Basically, when f is not differentiable due to multiple directions that may "play role of the gradient", the backtracking line search procedure may never exit the loop of trials, and thus, cannot be used.

3.2.5 Gradient Scaling

To overcome a possibly slow convergence of the gradient projection method mentioned in Section 3.2.3, the gradient is often scaled. In this case, the method takes the form

$$x_{k+1} = P_X[x_k - \alpha_k \Lambda_k \nabla f(x_k)], \tag{3.19}$$

where Λ_k is a diagonal matrix with positive entries on its diagonal, i.e., $[\Lambda_k]_{ii} > 0$ for all $i = 1, \ldots, n$ and all k.

When the function f is twice differentiable, one possibility is to use Λ_k based on Hessian information of f at x_k. This choice is motivated by the Newton method which uses the direction $[\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$ instead of the gradient. Since the inverse of the Hessian $\nabla^2 f(x_k)$ may be computationally very expensive a simpler scaling can be used instead that takes the inverse values of the diagonal elements of the Hessian $\nabla^2 f(x_k)$, i.e., the scaling matrix Λ_k is given by

$$[\Lambda_k]_{ii} = \left(\frac{\partial^2 f(x_k)}{\partial x_i^2} \right)^{-1} \qquad \text{for all } i \text{ and } k,$$

where $\frac{\partial^2 f(x_k)}{\partial x_i^2}$ is the (i,i)th entry of the Hessian $\nabla^2 f(x_k)$.

The analysis of such methods uses the successive transformation of space variables x_k based on the scaling matrix Λ_k. Namely, new variables are defined by

$$y_k = \Lambda_k^{\frac{1}{2}} x_k,$$

where $\Lambda_k^{\frac{1}{2}}$ is the square root of the matrix Λ_k.

[2] In some multi agent systems, agents have access to partial information about f, and no agent knows the entire function f.

[3] This can be seen from the the proofs for these results.

More generally, one may consider a non-diagonal scaling matrices Λ_k. However, these matrices have to be positively definite. In fact, one can establish the convergence of the scaled gradient projection method (3.19) when the maximum and the minimum of all eigenvalues of Λ_k are uniformly bounded over k.

We next provide a result showing a convergence of the scaled gradient projection method with the backtracking arc search rule (see a more general result for nonconvex objective function in [5], Proposition 2.3.4).

Theorem 47 *Let Assumption 4 hold. Let $\{x_k\}$ be a sequence generated by the scaled gradient projection method (3.19) with backtracking arc search stepsize. Also, let the matrices Λ_k be all positive definite and assume that for some scalars $\nu_1 > 0$ and $\nu_2 > 0$ the following relation is satisfied*

$$\nu_1\|x\|^2 \leq x^T\Lambda_k x \leq \nu_2\|x\|^2 \qquad \text{for all } \in \mathbb{R}^n \text{ and all } k.$$

Then, every accumulation point of the iterate sequence $\{x_k\}$ is an optimal solution.

More discussion on the scaled gradient projection algorithm can be found, for example, in [5] Section 2.3.

3.2.6 Feasible Descent Method

This is a general scheme for minimizing differentiable function over a constraint set. In particular, the idea is to generate an iterate sequence using "gradient related" directions d_k as opposed to gradient directions. The method takes the form

$$x_{k+1} = x_k + \alpha_k d_k \qquad \text{for all } k,$$

with d_k being a feasible direction of the set X at the iterate x_k, i.e.,

$$x_k + \alpha_k d_k \in X \qquad \text{for all } k.$$

For convergence of such a method to a solution of the problem, the directions $\{d_k\}$ need to be related to the gradient directions. This is ensured for example when for some $\eta > 0$,

$$d_k^T\nabla f(x_k) \leq -\eta\|d_k\|\,\|\nabla f(x_k)\| \qquad \text{for all } k.$$

In other words, this condition ensures that the angle formed by the direction d_k and the gradient $\nabla f(x_k)$ is bounded away from 90 degrees uniformly over k.

Frank-Wolfe Method

One idea of generating a feasible direction d_k is based on the use of the linear approximation of f at the current iterate x_k. In particular, by writing $d_k = \hat{x}_k - x_k$, where $\hat{x}_k$ is a solution to the following problem

$$\text{minimize} \qquad \nabla f(x_k)^T(x - x_k)$$

$$\text{subject to} \quad x \in X. \tag{3.20}$$

This method is known as *Frank-Wolfe method* and also as *conditional gradient method*. Note that the preceding problem may not have solution when X is unbounded. To ensure the existence of a solution, an additional assumption that X is bounded is typically used.

A typical iteration of the Frank-Wolfe method has the following steps: at the current iterate x_k, solve the minimization problem (3.20) to obtain $\hat{x}_k$, i.e.,

$$\hat{x}_k \in \operatorname*{Argmin}_{x \in X} \nabla f(x_k)^T (x - x_k).$$

Select the stepsize α_k, and define

$$x_{+1} = x_k + \alpha_k (\hat{x}_k - x_k).$$

Typically, the backtracking line search is used to determine the stepsize α_k.

The convergence results for the Frank-Wolfe method, as well as for more general feasible direction methods, can be found in [5] Section 2.2.

3.3 Dual Method

In this section, we consider a constrained problem where, in addition to the constraint set X, there are also inequality and linear equality constraints. Specifically the minimization problem of interest has the following form

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_1(x) \leq 0, \ldots, g_m(x) \leq 0 \\ & a_1^T x = b_1, \ldots, a_r^T x = b_r \\ & x \in X, \end{array}$$

where $X \subset \mathbb{R}^n$, $g_j : X \to \mathbb{R}$ for all j, and $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$ for all i. We will also use a more compact formulation of the preceding problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g(x) \leq 0 \\ & Ax = b \\ & x \in X, \end{array} \tag{3.21}$$

where $g = [g_1, \ldots, g_m]^T : X \to \mathbb{R}^m$, A is an $r \times n$ matrix with rows a_i^T, and $b \in \mathbb{R}^r$ is a vector with components b_i. Throughout this section, we use the following assumption.

Assumption 6 *The set X is convex and closed. The objective f and the constraint functions $g_1, \ldots, g_m$ are convex over the set X. The optimal value f^* of problem (3.21) is finite.*

A silent assumption is that the set X has a simple structure for projection operation. Solving the problem of the form (3.21) can be very complex due to the presence of (possibly

nonlinear) inequality constraints $g(x) \leq 0$. Here, we consider the algorithms for solving the problem (3.21) through the use dual methods as applied to solve the dual problem.

Consider the dual problem obtained by relaxing all the inequality and equality constraints (assigning prices to them). In this case, the dual problem is

$$\begin{array}{ll} \text{maximize} & q(\mu, \lambda) \\ \text{subject to} & \mu \geq 0, \end{array} \tag{3.22}$$

where the dual function $q(\mu, \lambda)$ is given by

$$q(\mu, \lambda) = \inf_{x \in X} \{f(x) + \mu^T g(x) + \lambda^T (Ax - b)\} \qquad \text{for } \mu \in \mathbb{R}^m \text{ with } \mu \geq 0 \text{ and } \lambda \in \mathbb{R}^r. \tag{3.23}$$

Note that the constraint $(\mu, \lambda) \in \operatorname{dom} q$ is an implicit constraint of the dual problem (silently assumed). In the dual problem, the multiplier μ is constrained to the nonnegative orthant, while the multiplier λ is a free variable. Furthermore, the dual function $q(\mu, \lambda)$ is concave, so that the dual is a constrained concave maximization problem, which is equivalent to a constrained convex minimization problem (through a sign change in the objective). Hence, if the dual function is differentiable we could apply gradient projection methods (for maximization) and solve the dual.

In some situations, a partial dual problem is considered and is still referred to as the dual. In particular, consider relaxing only the inequality constraints of the problem (3.21), yielding a dual problem of the form

$$\begin{array}{ll} \text{maximize} & \tilde{q}(\mu) \\ \text{subject to} & \mu \geq 0, \end{array} \tag{3.24}$$

where the dual function $\tilde{q}(\mu)$ is given by

$$\tilde{q}(\mu) = \inf_{Ax = b,\ x \in X} \{f(x) + \mu^T g(x)\} \qquad \text{for } \mu \in \mathbb{R}^m \text{ with } \mu \geq 0. \tag{3.25}$$

In this dual problem, the multiplier μ is constrained to the nonnegative orthant, while the dual function $\tilde{q}(\mu)$ is concave. To distinguish between these two different formulations of the dual problem, we will refer to problem (3.22)–(3.23) as *the dual problem* and to problem (3.24)–(3.25) as *the partial dual problem*.

The main difficulty in dealing with dual problems is the evaluation of the dual function, since it involves solving a constrained minimization problem per each value of the dual variables. The use of dual problems is the most advantageous in the situations when the dual function evaluation is "easy", i.e., when a dual solution is explicitly given. Fortunately, this is the case in many problems arising in various applications. We discuss some of them in Chapter 4.

In what follows, we focus on the dual problems where the minimization problem involved in the dual function evaluation has solutions. Under this assumption, for the dual function $q(\mu, \lambda)$ of Eq. (3.23), we have for any $\mu \geq 0$ and any λ,

$$q(\mu, \lambda) = f(x_{\mu\lambda}) + \mu^T g(x_{\mu\lambda}) + \lambda^T (Ax_{\mu\lambda} - b),$$

where $x_{\mu\lambda}$ is an optimal solution for the following problem

$$\begin{array}{ll} \text{minimize} & f(x) + \mu^T g(x) + \lambda^T (Ax - b) \\ \text{subject to} & x \in X. \end{array} \tag{3.26}$$

Similarly, under the assumption that the minimization problem defining the dual function $\tilde{q}(\mu)$ of Eq. (3.25) has optimal solutions, we have for any $\mu \geq 0$,

$$\tilde{q}(\mu) = f(x_\mu) + \mu^T g(x_\mu),$$

where x_μ is an optimal solution for the following problem

$$\begin{array}{ll} \text{minimize} & f(x) + \mu^T g(x) \\ \text{subject to} & Ax = b,\ x \in X. \end{array} \tag{3.27}$$

Let us now consider the relations that characterize the minimizers $x_{\mu\lambda}$ and x_μ of the problems (3.26) and (3.27) respectively, when f and all g_j are convex, and X is closed and convex. These relations require differentiability of f and all g_j. The gradient of the Lagrangian function of Eq. (3.26) is

$$\nabla f(x) + \sum_{j=1}^{m} \mu_j \nabla g_j(x) + \lambda^T A.$$

Thus, by the first-order optimality conditions, $x_{\mu\lambda}$ is an optimal solution for problem (3.26) if and only if

$$\left(\nabla f(x_{\mu\lambda}) + \sum_{j=1}^{m} \mu_j \nabla g_j(x_{\mu\lambda}) + \lambda^T A \right)^T (x - x_{\mu\lambda}) \geq 0 \qquad \text{for all } x \in X.$$

Similarly, the gradient of the Lagrangian function of Eq. (3.27) is

$$\nabla f(x) + \sum_{j=1}^{m} \mu_j \nabla g_j(x).$$

By the first-order optimality conditions, x_μ is an optimal solution for problem (3.27) if and only if

$$\left(\nabla f(x_{\mu\lambda}) + \sum_{j=1}^{m} \mu_j \nabla g_j(x_{\mu\lambda}) \right)^T (x - x_{\mu\lambda}) \geq 0 \qquad \text{for all } x \in X,\ Ax = b.$$

To this end, we discussed both the dual and the partial dual problem, since both have been traditionally used, depending on which one is more suitable for a given problem at hand. For the rest of this section, we will focus only on the dual problem (3.22). Analogous results hold for the partial dual.

With respect to the existence of optimal solutions for the problem (3.26), we consider two cases:

(1) The minimizer $x_{\mu\lambda}$ is unique for each $\mu \geq 0$ and $\lambda \in \mathbb{R}^r$.

(2) The minimizer $x_{\mu\lambda}$ is not unique.

The uniqueness of the minimizers ties closely with the differentiability of the dual function $q(\mu, \lambda)$, which we discuss next.

3.3.1 Differentiable Dual Function

In this section, we discuss the dual methods for the case when the dual function is differentiable. As we will see the differentiability of the dual function is guaranteed when the minimizer of the problem (3.26) is unique. We focally impose this condition, as follows.

Assumption 7 *For every $\mu \geq 0$ and $\lambda \in \mathbb{R}^r$, the minimizer $x_{\mu\lambda}$ in the problem (3.26) exists and it is unique.*

Under convexity assumption and Assumption 7, we have the following result.

Lemma 13 *Let Assumptions 6 and 7 hold. Then, for every $\mu \geq 0$ and $\lambda \in \mathbb{R}^r$, the inequality and equality constraints of the problem (3.21) evaluated at the minimizer $x_{\mu\lambda}$ constitute the gradient of q at (μ, λ), i.e.,*

$$\nabla q(\mu, \lambda) = \begin{bmatrix} g_1(x_{\mu\lambda}) \\ \vdots \\ g_m(x_{\mu\lambda}) \\ a_1^T x_{\mu\lambda} - b_1 \\ \vdots \\ a_r^T x_{\mu\lambda} - b_r \end{bmatrix}.$$

Note that, in view of Lemma 13, *the differentiability of the dual function has nothing to do with the differentiability of the objective f or constraint functions g_j.* The differentiability of q is strongly related to the uniqueness of the optimizer in the problem defining the dual function value.

According to Lemma 13, for the partial gradients $\nabla_\mu q(\mu, \lambda)$ and $\nabla_\lambda q(\mu, \lambda)$, we have

$$\nabla_\mu q(\mu, \lambda) = \begin{bmatrix} g_1(x_{\mu\lambda}) \\ \vdots \\ g_m(x_{\mu\lambda}) \end{bmatrix} = g(x_{\mu\lambda}),$$

$$\nabla_\lambda q(\mu, \lambda) = \begin{bmatrix} a_1^T x_{\mu\lambda} - b_1 \\ \vdots \\ a_r^T x_{\mu\lambda} - b_r \end{bmatrix} = A x_{\mu\lambda} - b.$$

To solve the dual problem, under Assumptions 6 and 7, we can now apply the projection gradient method, which is adapted to handle maximization. The projected gradient method for the dual problem has the form:

$$\begin{aligned} \mu_{k+1} &= [\mu_k + \alpha_k \nabla_\mu q(\mu, \lambda)]^+, \\ \lambda_{k+1} &= \lambda_k + \alpha_k \nabla_\lambda q(\mu, \lambda), \end{aligned}$$

where $[\cdot]^+$ denotes the projection on the nonnegative orthant $\mathbb{R}^m_+$, $\alpha_k > 0$ is the stepsize, and $\mu_0 \geq 0$ and λ_0 are initial multipliers. Note that, since the dual problem involves

maximization, the gradient method takes steps along the gradients of q. We refer to this gradient method as *the dual gradient projection method*.

By using Lemma 13, we see that the method is equivalently given by

$$\mu_{k+1} = [\mu_k + \alpha_k \, g(x_{\mu\lambda})]^+, \tag{3.28}$$

$$\lambda_{k+1} = \lambda_k + \alpha_k \, (Ax_{\mu\lambda} - b). \tag{3.29}$$

The dual gradient method can be used with all the stepsizes discussed in Section 3.2, including the backtracking line search. However, the Polyak's stepsize and its modification, and the backtracking line search have to be suitably adjusted to account for the maximization aspect. In particular, for the dual gradient projection method the Polyak's stepsize is given by:

$$\alpha_k = \frac{q^* - q(\mu_k, \lambda_k)}{\|\nabla q(\mu_k, \lambda_k)\|^2},$$

where q^* is the optimal value of the dual problem (of course, q^* should be finite in order to use this stepsize).

Denote by x_k the minimizer $x_{\mu\lambda}$ when $(\mu, \lambda) = (\mu_k, \lambda_k)$. Note that $\|\nabla q(\mu_k, \lambda_k)\|^2 = \|g(x_k)\|^2 + \|Ax_k - b\|^2$, or equivalently

$$\|\nabla q(\mu_k, \lambda_k)\|^2 = \sum_{j=1}^{m} g_j^2(x_k) + \sum_{i=1}^{r} (a_i^T x_k - b_i)^2.$$

For the dual gradient projection method, the modified Polyak's stepsize has the following form

$$\alpha_k = \frac{\hat{q}_k - q(\mu_k, \lambda_k)}{\|\nabla q(\mu_k, \lambda_k)\|^2} \qquad \text{with} \qquad \hat{q}_k = \delta + \max_{0 \le \kappa \le k} q(\mu_\kappa, \lambda_\kappa).$$

Finally, in the backtracking arc search procedure of Section 3.2.4, the relation at step 2 should be *Sufficient Ascent Test* and the inequality in the test should be

$$q(v_i) \ge q(\mu_k, \lambda_k) + \sigma t_i \|\nabla q(\mu_k, \lambda_k)\|^2,$$

where

$$v_i = \begin{bmatrix} [\mu_k + t_i \nabla_\mu q(\mu_k, \lambda_k)]^+ \\ \lambda_k + t_i \nabla_\lambda q(\mu_k, \lambda_k) \end{bmatrix} = \begin{bmatrix} [\mu_k + t_i g(x_k)]^+ \\ \lambda_k + t_i p(Ax_k - b) \end{bmatrix}.$$

Note that, when appropriately interpreted, all the results for the gradient projection method that we established in Section 3.2 apply to the dual maximization problem.

We next provide an example demonstrating computation of the gradient of the dual function. In particular, we revisit the Kelly's canonical utility-based network resource allocation problem (see [19]).

Example 31 *Consider a network consisting of a set $\mathcal{S} = \{1, \ldots, S\}$ of sources and a set $\mathcal{L} = \{1, \ldots, L\}$ of undirected links, where a link l has capacity c_l. Let $\mathcal{L}(i) \subset \mathcal{L}$ denote the set of links used by source i. The application requirements of source i is represented by a differentiable concave increasing utility function $u_i : [0, \infty) \to [0, \infty)$, i.e., each source i*

gains a utility $u_i(x_i)$ when it sends data at a rate x_i. Let $\mathcal{S}(l) = \{i \in \mathcal{S} \mid l \in \mathcal{L}(i)\}$ denote the set of sources that use link l. The goal of the network utility maximization problem is to allocate the source rates as the optimal solution of the problem

$$\begin{aligned} &\textit{maximize} && \sum_{i\in\mathcal{S}} u_i(x_i) \\ &\textit{subject to} && \sum_{i\in\mathcal{S}(l)} x_i \le c_l \quad \textit{for all } l \in \mathcal{L} \\ & && x_i \ge 0 \quad \textit{for all } i \in \mathcal{S}. \end{aligned}$$

Alternatively, we may view the problem as the minimization of differentiable convex and decreasing function $f(x) = -\sum_{i\in\mathcal{S}} u_i(x_i)$ subject to the above constraints. Note that the constraint set of the problem is compact [since $0 \le x_l \le c_l$ for all links $l \in \mathcal{L}$]. Since f is continuous over the constraint set, the optimal value f^* is finite [in fact, a unique optimal solution x^* exists]. Assumption 6 is satisfied.

By relaxing the link capacity constraints, the dual function takes the form

$$\begin{aligned} q(\mu) &= \min_{x_i\ge 0,\ i\in\mathcal{S}} \sum_{i\in\mathcal{S}} = u_i(x_i) + \sum_{l\in\mathcal{L}} \mu_l \left(\sum_{i\in\mathcal{S}(l)} x_i - c_l \right) \\ &= \min_{x_i\ge 0,\ i\in\mathcal{S}} \sum_{i\in\mathcal{S}} \left(-u_i(x_i) + x_i \sum_{l\in\mathcal{L}(i)} \mu_l \right) - \sum_{l\in\mathcal{L}} \mu_l c_l. \end{aligned}$$

Since the optimization problem on the right-hand side of the preceding relation is separable in the variables x_i, the problem decomposes into subproblems for each source i. Letting $\mu_i = \sum_{l\in\mathcal{L}(i)} \mu_l$ for each i (i.e., μ_i is the sum of the multipliers corresponding to the links used by source i), we can write the dual function as

$$q(\mu) = \sum_{i\in\mathcal{S}} \min_{x_i\ge 0} \{x_i\mu_i - u_i(x_i)\} - \sum_{l\in\mathcal{L}} \mu_l c_l.$$

Hence, to evaluate the dual function, each source i needs to solve the one-dimensional optimization problem $\min_{x_i\ge 0}\{x_i\mu_i - u_i(x_i)\}$. Note that $\mu_i = 0$ is not in the domain of the dual function [since each u_i is increasing, it follows that $\min_{x_i\ge 0}\{-u_i(x_i)\} = -\infty$]. Thus, we must have $\mu_i > 0$ for all $i \in \mathcal{S}$ for the dual function to be well defined.

For $\mu_i > 0$, by the first-order optimality conditions [see Theorem 20 of Section 2.3.3], the optimal solution $x_i(\mu_i)$ for the one-dimensional problem satisfies the following relation

$$u_i'(x_i(\mu_i)) = \mu_i,$$

where $u_i'(x_i)$ denotes the derivative of $u_i(x_i)$. Thus,

$$x_i(\mu_i) = u_i'^{-1}(\mu_i),$$

where $u_i'^{-1}$ is the inverse function of u_i', which exists since u_i is differentiable and increasing. Hence, for each dual variable $\mu > 0$, the minimizer $x(\mu) = \{\mu_i,\ i \in \mathcal{S}\}$ in the problem of dual function evaluation exists and it is unique.

The following implication of the KKT conditions [cf. Theorem 34 of Section 2.5.7] is often useful when a dual optimal solution is available.

Lemma 14 *Let Assumption 6 hold, and let strong duality hold [$f^* = q^*$]. Let (μ^*, λ^*) be an optimal multiplier pair. Then, x^* is a solution of the primal problem 3.21 if and only if*

- *x^* is primal feasible, i.e., $g(x^*) \leq 0,\ Ax^* = 0,\ x^* \in X$.*
- *x^* is a minimizer of the problem (3.26) with $(\mu, \lambda) = (\mu^*, \lambda^*)$.*
- *x^* and μ^* satisfy the complementarity slackness, i.e., $(\mu^*)^T g(x^*) = 0$.*

Chapter 4

Network Applications

In this chapter, we consider a classic network flow problem and the use of this model in routing and congestion control in communication networks.

The network flow problem is just a special case of linear programming. We discuss the simplex method as applied to the network flow problems, and we also consider some special cases of the network flow problem, such as the shortest path and the maximum flow problem. Some of these problems have a special structure that can be exploited to simplify the simplex method, as well as to design new special algorithms. In depth treatment of network flow problems can be found in the textbooks by Bertsekas [4], Bertsimas and Tsitsiklis [11], and Ahuja, Magnanti, and Orlin [1].

A more general network flow model, with convex objective, is discussed as a flow model for routing and congestion control in communication networks. In particular, we present a classic routing problem for data networks and a more recent utility-based model for joint routing and congestion control. The material is focused on the use of convex optimization highlighting the role of optimality conditions and the duality theory in the development of the algorithms for the routing, congestion control, and general resource allocation in networks. More on flow algorithms for data routing can be found in [8]. The utility-based flow model for network resource allocation, of particular interest for joint routing and congestion control, has been proposed by Kelly *et. al* in [19], and further studied by Low and Lapsley [21]. An in depth coverage of the state-of-art algorithms for joint routing and congestion control can be found in the book by Srikant [31], and in the article by Shakkottai and Srikant [28]. The analytical tools for the design and analysis of communication networks are covered in the lecture notes by Hajek [17].

4.1 Graphs

A network is modeled by a graph. We distinguish two types of graphs, namely, undirected and directed.

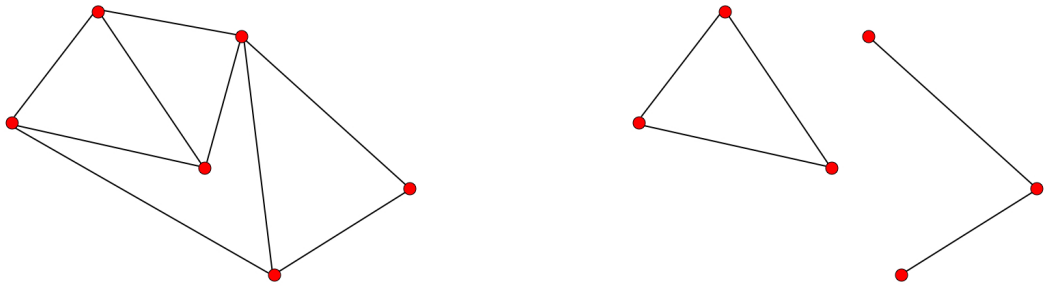

Figure 4.1: The graph to the left is connected, while the graph to the right is not connected.

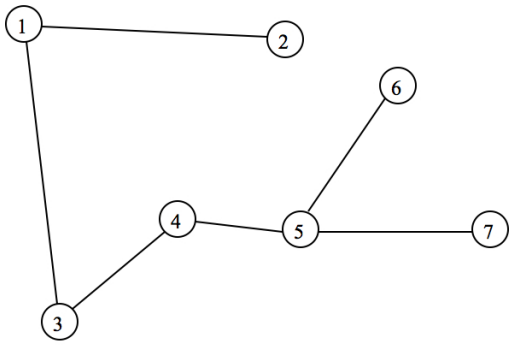

Figure 4.2: This graph is a tree with leaves 2, 6, and 7.

Undirected Graph

An *undirected graph* $G = (\mathcal{N}, \mathcal{E})$ with a set N of nodes and a set $\mathcal{E}$ of edges. An edge is an unordered pair i, j of two distinct nodes $i, j \in \mathcal{N}$. In our setting here, the self-edges $\{i, i\}$ are not permitted. The number of nodes is denoted by $|\mathcal{N}|$ or n, while the number of edges is denoted by $|\mathcal{E}|$ or m.

We say that the edge $\{i, j\}$ is *incident* to nodes i and j. The *degree of a node* in an undirected graph is the number of edges incident to the node. The *degree of an undirected graph* is the maximum of the degrees of the nodes in that graph.

A*walk in undirected graph* from node i_1 to node i_s is a finite (ordered) sequence of nodes $i_1, \ldots, i_s$ such that $\{i_k, i_{k+1}\} \in \mathcal{E}$ for $k = 1, \ldots, s-1$. A *path in undirected graph* is a walk with no repeated nodes (i.e., the nodes $i_1, \ldots, i_s$ are distinct). A walk $i_1, \ldots, i_s$ is a cycle if $i_1 = i_s$ and all the other nodes $i_2, \ldots, i_{s-1}$ are distinct, and there are at least 3 distinct nodes. This is because we want to exclude a cycle of the form i, j, i where the edge $\{i, j\}$ is traversed twice (back and forth). An undirected graph is *connected* when for every two distinct nodes i and j, there is a path from i to j. Figure 4.1 shows an undirected graph that is connected and an undirected graph that is not connected.

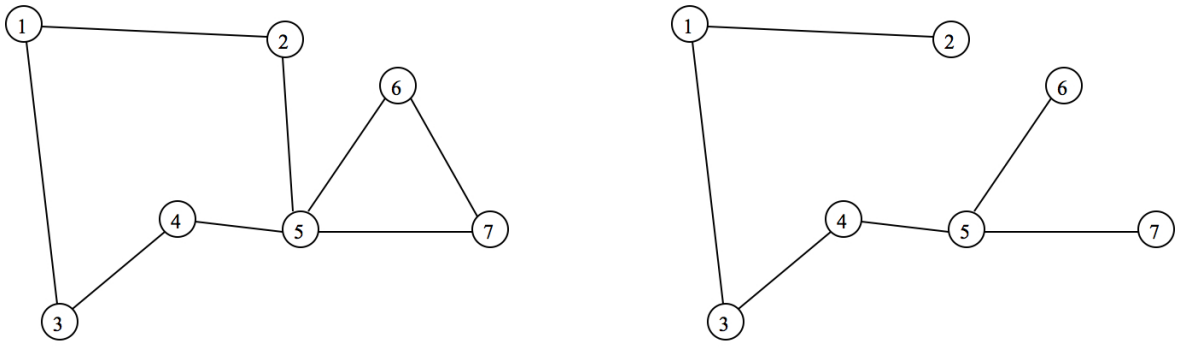

Figure 4.3: An undirected graph is to the left. A spanning tree of the graph is to the right.

An undirected graph $G = (\mathcal{N}, \mathcal{E})$ is a *tree*, if it is connected and has no cycles. A node in a tree with degree 1 is a *leaf*. Figure 4.2 illustrates a tree and its leaves. A tree has some special properties summarized in the following.

Theorem 48 *We have:*

(a) Every tree with more than two nodes has a leaf.

(b) An undirected graph is a tree if and only if it is connected and has $|\mathcal{N}| - 1$ edges.

(c) For every pair of distinct nodes i and j in a tree, there is a unique path from i to j.

(d) Given a tree, if we add a new edge, the resulting graph has exactly one cycle.

Let $G = (\mathcal{N}, \mathcal{E})$ be a undirected graph. Let $\mathcal{E}_1 \subseteq \mathcal{E}$ be a subset of edges such that the subgraph $T = (\mathcal{N}, \mathcal{E}_1)$ is a tree. Such a tree is *a spanning tree*. A graph and its spanning tree are illustrated in Figure 4.3. Every connected undirected graph contains a spanning tree. This follows from the following theorem.

Theorem 49 *Let $G = (\mathcal{N}, \mathcal{E})$ be a connected undirected graph. Let $\mathcal{E}_0 \subseteq \mathcal{E}$ be a set of edges that cannot form a cycle. Then, the set $\mathcal{E}_0$ can be augmented to a set $\mathcal{E}_1$ such that the graph $(\mathcal{N}, \mathcal{E}_1)$ is a spanning tree.*

Directed Graph

A *directed graph* $G = (\mathcal{N}, \mathcal{L})$ with a set N of nodes and a set $\mathcal{L}$ of directed links. A directed link is an ordered pair (i, j) of two distinct nodes $i, j \in \mathcal{N}$. A directed graph is shown in Figure 4.4. In our setting, the self-links (i, i) are not permitted. The number of nodes is denoted by $|\mathcal{N}|$ or n, while the number of links is denoted by $|\mathcal{L}|$ or m.

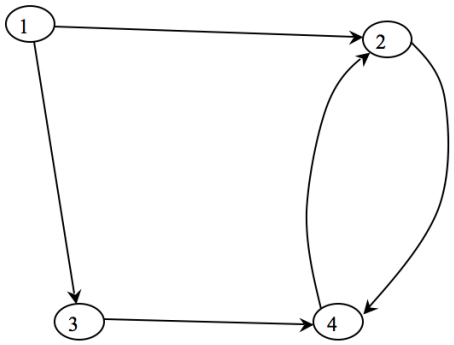

Figure 4.4: A directed graph.

We say that the link (i, j) is *from i to j*, or *outgoing from i and incoming to j*. The set of all nodes with links incoming to i is denoted by $I(i)$, while the set of links outgoing from node i is denoted by $O(i)$,

$$I(i) = \{j \in \mathcal{N} \mid (j, i) \in \mathcal{L}\},$$

$$O(i) = \{j \in \mathcal{N} \mid (i, j) \in \mathcal{L}\}.$$

A*walk in a directed graph* from node i_1 to node i_s is a finite (ordered) sequence of nodes $i_1, \ldots, i_s$ and links $l_1, \ldots, l_{s-1}$ such that for all $k = 1, \ldots s - 1$,

$$\text{either} \qquad l_k = (i_k, i_{k+1}) \in \mathcal{N} \quad \text{(forward link)}$$

$$\text{or} \qquad l_k = (i_{k+1}, i_k) \in \mathcal{N} \quad \text{(backward link)}.$$

A *path in undirected network* is a walk with no repeated nodes (i.e., the nodes $i_1, \ldots, i_s$ are distinct). A walk $i_1, \ldots, i_s$ is a cycle if $i_1 = i_s$ and all the other nodes $i_2, \ldots, i_{s-1}$ are distinct. A walk, a path, or a cycle is directed when it only contains forward links.

A directed graph is *connected* when the undirected graph obtained by ignoring the directions and deleting the repeated edges is connected. A directed graph is *strongly connected* when for every two distinct nodes $i, j \in \mathcal{N}$, there is a directed path from i to j. The directed graph shown in Figure 4.4 is connected but not strongly connected.

4.2 Minimum Cost Network Flow Problem

A network is a directed [or undirected] graph $G = (\mathcal{N}, cL)$ [or $G = (\mathcal{N}, \mathcal{E})$] with some additional information such as

- Link capacity $u_{ij} > 0$ for $(i, j) \in \mathcal{L}$, which is the maximum possible flow on the link.
- Link cost c_{ij} for $(i, j) \in \mathcal{L}$, representing the cost per unit flow on the link.

- External supply b_i for each node $i \in \mathcal{N}$, representing the amount of flow that enters the network at node i.

A node i with a positive supply $[b_i > 0]$ is a *source node*, while a node with a negative supply $[b_i < 0]$ is a *destination node* or *a sink*. A node with no supply $[b_i = 0]$ is a *transient node*.

In a general network problem, we want to send a flow from source nodes to destination nodes at the minimal cost, subject to capacity constraints and the network balance relations enforcing the law of flow conservation. This law states that, at every node, the amount of flow into a node must be equal to the amount of flow out of that node.

Formally, the general minimum cost network problem is a linear problem of the following form:

$$\begin{aligned} \text{minimize} \quad & \sum_{(i,j)\in\mathcal{L}} c_{ij}\, x_{ij} \\ \text{subject to} \quad & \sum_{j\in O(i)} x_{ij} - \sum_{j\in I(i)} x_{ij} = b_i \qquad \text{for all nodes } i \in \mathcal{N} \\ & 0 \le x_{ij} \le u_{ij} \qquad \text{for all links } (i,j) \in \mathcal{L}. \end{aligned} \tag{4.1}$$

The equality constraints

$$\sum_{j\in O(i)} x_{ij} - \sum_{j\in I(i)} x_{ij} = b_i \qquad \text{for all nodes } i \in \mathcal{N} \tag{4.2}$$

model the network balance. Note that by summing these equations, we have

$$\sum_{i\in\mathcal{N}} b_i = 0,$$

which models the law of network flow conservation.

The inequalities

$$0 \le x_{ij} \le u_{ij} \qquad \text{for all links } (i,j) \in \mathcal{L} \tag{4.3}$$

model the capacity constraints for the links. When the network links are uncapacitated $[u_{ij} = +\infty]$, the network problem is a linear problem in the standard form.

We rewrite the network problem of Eq. (4.1) in a more convenient form, as follows. Let $|\mathcal{N}| = n$ and $|\mathcal{L}| = m$. We put the links $(i,j) \in \mathcal{L}$ in an order, and we order the flow variables x_{ij} accordingly. define the *node-link incidence matrix* A as an $n \times m$ matrix (a row per node and a column per link) with entries a_{ik} given by

$$a_{ik} = \begin{cases} 1 & \text{if } i \text{ is the start node of the } k\text{th link} \\ -1 & \text{if } i \text{ is the end node of the } k\text{th link} \\ 0 & \text{otherwise.} \end{cases}$$

As an example, consider the directed graph of Figure 4.4. Take the links in the following order $(1,2)$, (1,3), (2,4), (4,2), and (3, 4). The matrix A associated with this order of the

links is

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1 & -1 \end{bmatrix}. \tag{4.4}$$

The matrix A has two nonzero entries in each column, namely 1 and -1 indicating the start and the end node of the corresponding link.

Letting A_i denote the ith row of A, we have

$$A_i\, x = \sum_{j \in O(i)} x_{ij} - \sum_{j \in I(i)} x_{ij},$$

which in view of the network balance equations (4.2) implies

$$A_i\, x = b_i \qquad \text{for all } i \in \mathcal{N}.$$

Equivalently, as a matrix equation, we have

$$Ax = b,$$

where $b \in \mathbb{R}^n$ is a vector with entries b_i, $i = 1, \ldots, n$.

The sum of rows of A is 0, implying that the rows of A are linearly dependent. This violates the assumption used in the development of the simplex method in Section 3.1. However, we can eliminate some of the equalities and obtain a subset of linearly independent rows, without affecting the feasible set of the problem. In particular, let $\tilde{A}$ be the matrix obtained from the node-link incidence matrix A by taking the first $n-1$ rows of A. The matrix $\tilde{A}$ is the *truncated node-link incidence matrix*. It can be seen that the matrix $\tilde{A}$ has linearly independent rows. Let $\tilde{b}$ be the correspondingly truncated supply vector b, i.e., $\tilde{b}$ consists of the first $n-1$ components of b. We can replace the original balance equation $Ax = b$ with $\tilde{A}x = \tilde{b}$.

As an example, consider the node incidence matrix A of Eq. (4.4) corresponding to the directed graph of Figure 4.4. By removing the last row of A, we obtain the reduced matrix $\tilde{A}$ given by

$$\tilde{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 0 & 1 \end{bmatrix}.$$

This matrix has linearly independent rows.

We now discuss some concepts used in many network flow algorithms. A vector x is a *circulation* if

$$Ax = 0.$$

Note that such a flow need not be feasible.

Let C be a cycle, and let F and B be respectively the set of forward and backward links in the cycle C. The flow y^C defined by

$$y^C_{ij} = \begin{cases} 1 & \text{if } (i,j) \in F \\ -1 & \text{if } (i,j) \in B \\ 0 & \text{otherwise} \end{cases}$$

is *the circulation associated with the cycle* C. We have

$$Ay^C = 0.$$

The *cost of the cycle* C *is given by*

$$c^T y^C = \sum_{(i,j)\in F} c_{ij} - \sum_{(i,j)\in B} c_{ij}.$$

Given a flow x a cycle C, and a scalar $\theta > 0$, the flow vector $x + \theta y^C$ is obtained from x by *pushing* θ *units of flow around the cycle* C. The cost change associated with the flow push around the cycle C is $c^T y^C$.

4.2.1 Simplex Algorithm for Uncapacitated Min-Cost Flow

Consider the uncapacitated network flow problem

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0, \end{array}$$

where A is the node-link incidence matrix of a directed graph $G = (\mathcal{N}, \mathcal{L})$. We assume that $\sum_{i\in\mathcal{N}} b_i = 0$ and *that the graph* $G = (\mathcal{N}, \mathcal{L})$ *is connected.* A flow vector x is *a tree solution* if it can be obtained as follows:

(1) Select a set $\mathcal{T}$ of $n-1$ links in $\mathcal{L}$, and form a tree (ignoring the link directions).

(2) Set $x_{ij} = 0$ for $(i,j) \notin \mathcal{T}$.

(3) Use the (reduced) flow conservation equation $\tilde{A}x = \tilde{b}$ to determine the variables x_{ij} for $(i,j) \in \mathcal{T}$.

A tree solution such that $x \geq 0$ is *a feasible tree solution.*

With respect to the terminology developed for the simplex method in Section 3.1, we have that *a flow vector* x *is a basic solution if and only if* x *is a tree solution.*

Change of basis in the simplex method, corresponds to generating a new feasible tree solution from the current one. In particular, let $\mathcal{T}$ be the current basic tree solution. Selecting a nonbasic variable in the simplex method corresponds to selecting a link $(i,j) \notin \mathcal{T}$. Then, the link (i,j) and some other links in $\mathcal{T}$ form a a cycle C. Choose the orientation of the cycle C so that (i,j) is forward link in the cycle. If we can increase the flow on the link (i,j) to some value $\theta > 0$, then we have to adjust the old basic variables to maintain the flow balance. This corresponds to pushing θ units of flow around the cycle C. Furthermore, we want to push the maximal possible amount of θ^* units. Thus, we move from a feasible tree solution x to another feasible tree solution $\tilde{x}$, whose components are given by

$$\tilde{x}_{kj} = \begin{cases} x_{kj} + \theta^* & \text{if } (k,j) \in F \\ x_{kj} - \theta^* & \text{if } (k,j) \in B \\ x_{kj} & \text{otherwise,} \end{cases}$$

If $B \neq \emptyset$, then

$$\theta^* = \min_{(k,j)\in B} x_{kj},$$

and otherwise $\theta^* = +\infty$.

The cost associated with this change is equal to

$$\theta^* \left(\sum_{(k,j)\in F} c_{kj} - \sum_{(k,j)\in B} c_{kj} \right).$$

The variable x_{ij} enters the basis only if this cost is negative. The reduced cost of the nonbasic variable x_{ij} is given by

$$\bar{c}_{ij} = \sum_{(k,j)\in F} c_{kj} - \sum_{(k,j)\in B} c_{kj}.$$

This is exactly the cost of the cycle C around which the flow is being pushed.

An alternative, a more efficient, formula for computing the reduced cost is given in terms of the dual variables p. Recall that the general formula for the reduced cost $\bar{c}$ is

$$\bar{c}^T = c^T - p^T \tilde{A} \qquad \text{with} \;\; p^T = c_B^T B^{-1},$$

with B being the current basis matrix. For the network problem, the dimension of p is the same as the number of rows of the reduced node-link incident matrix $\tilde{A}$, which is $n-1$. Using the structure of $\tilde{A}$, it can be seen that

$$\bar{c}_{kj} = \begin{cases} c_{kj} - (p_k - p_j) & \text{if } (i,j) \neq n \\ c_{kj} - p_k & \text{if } j = n \\ c_{kj} + p_j & \text{if } k = n. \end{cases}$$

By defining $p_n = 0$, we can compactly write the preceding relation as

$$\bar{c}_{kj} = c_{kj} - (p_k - p_j) \qquad \text{for all} \;\; (k,j) \in \mathcal{L}.$$

To compute the dual vector p for the feasible tree solution $\mathcal{T}$, we use the fact that the reduced cost associated with the basic variables x_{kj} with $(k,j) \in \mathcal{T}$ is zero. Thus, we have

$$\begin{array}{ll} p_k - p_j = c_{kj} & \text{for all} \;\; (k,j) \in \mathcal{T}, \\ p_n = 0 & . \end{array} \tag{4.5}$$

The uncapacitated min-cost flow problem has important integrality properties. Specifically, when the network is connected, the matrix B^{-1} has integer entries for any basis matrix B. This is the consequence of the structure of the node-link incidence matrix A. In addition, we have

- When the supplies b_i are integer, every basic solution has integer components.
- When the link costs c_{ij} are integer, every dual basic solution has integer components.

The simplex method can be adapted to apply to the capacitated network flow problem (4.1). However, a potential disadvantage is coming from degeneracy of basic solutions, in which case a change of basis may not decrease the cost. There is an alternative algorithm that, similar to simplex method, improves the cost by identifying a negative cost cycle, but unlike the simplex method, the cost improvement is nonzero at every iteration. This method is known as *negative cost cycle algorithm*. The interested reader can find more on this method, for example, in [11] Section 7.4.

4.3 Shortest Path Problem

There are several shortest path problems that one may consider, such as single origin-multiple destinations, or single origin-single destination. Here, we consider an all origin-single destination shortest path problem in the graph $G = (\mathcal{N}, \mathcal{L})$. Without loss of generality, assume that $\mathcal{N} = \{1, \ldots, n\}$ and that we are interested in paths from any node i to node n. We assume that there is a directed path from each node i to node n, and that node n has no outgoing links. This shortest path problem can be formulated as a min-cost flow problem, where each node i other than node n has one unit supply [$b_i = 1$], while node n has demand of $n-1$ units [$b_n = -(n-1)$]. Thus, the shortest path problem can be modeled as:

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{(ij)\in\mathcal{L}} c_{ij}x_{ij} & \\ \text{subject to} & \displaystyle\sum_{j\in O(i)} x_{ij} - \sum_{j\in I(i)} x_{ij} = \begin{cases} 1 & \text{if } i \neq n, \\ -(n-1) & \text{if } i = n, \end{cases} & \text{for all } i \in \mathcal{N} \\ & x_{ij} \geq 0 \qquad \text{for all } (i,j) \in \mathcal{L}, & \end{array}$$

which is a min-cost network flow problem [cf. Eq. (4.1)]. Using the node-link incidence matrix, the problem can be written as a standard LP problem:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0, \end{array}$$

where the vector $b \in \mathbb{R}^n$ has a special structure,

$$b = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ -(n-1) \end{bmatrix}.$$

Since the supply vector b has integer components, every basic solution also has integer components. Any forward path $\mathcal{P}_i$ from node i to node n, can be represented by a flow x with components

$$x_{kj} = \begin{cases} 1 & \text{if } (k,j) \in \mathcal{P}_i \\ 0 & \text{otherwise .} \end{cases}$$

Thus, each node i other than node n should send one unit of flow to node n, and the flow should be pushed along the shortest path from i to n.

If the graph $G = (\mathcal{N}, \mathcal{L})$ contains a cycle with negative cost, then the optimal cost of the min-cost flow problem is $f^* = -\infty$. If the graph does not contain negative cycles, then the optimal cost is finite and the simplex algorithm can be applied to obtain a tree optimal solution. Such an optimal solution consists of paths $\mathcal{P}_i^*$, $i = 1, \ldots, n-1$, where each $\mathcal{P}_i^*$ is a shortest path from node i to node n.

Furthermore, there is a dual optimal solution with a special structure. In particular, recall that the dual of the standard form LP is

$$\begin{array}{ll} \text{maximize} & b^T p \\ \text{subject to} & A^T p \leq c. \end{array}$$

Suppose that $\mathcal{T}^*$ is an optimal tree solution for the min-cost flow problem modeling the shortest path problem. Let p^* be the dual variables associated with the tree solution $\mathcal{T}^*$, i.e., p^* and $\mathcal{T}^*$ satisfy relations in Eq. (4.5). Then, for each link $(k, j) \in \mathcal{T}^*$, we have

$$c_{kj} = p_k^* - p_j^*.$$

Hence, for the optimal path $\mathcal{P}_i^* \subset \mathcal{T}^*$ from node i to n, we have

$$c_{kj} = p_k^* - p_j^* = c_{kj} \qquad \text{for all } (k, j) \in \mathcal{P}_i^*.$$

By adding the costs of the links in the path $\mathcal{P}_i^*$, we obtain

$$\sum_{(k,j) \in \mathcal{P}_i^*} c_{kj} = p_i^* - p_n^* = p_i^*,$$

where we use the fact $p_n^* = 0$ [cf. Eq. (4.5)]. Thus, the dual variable associated with the tree optimal solution is unique [when the last component p_n is fixed to zero]. Note that the tree solution is nondegenerate, since $x_{kj}^* = 1$ for all $(k, j) \in \mathcal{T}^*$ (none of the basic variables is zero). Hence, by Theorem 37, we have that the reduced cost of $\mathcal{T}^*$ is zero. Since the reduced cost of $\mathcal{T}^*$ is $\bar{c}^T = c^T - (p^*)^T A$, it follows that

$$c^T - (p^*)^T A \geq 0,$$

implying that p^* is dual feasible, and hence (by KKT conditions), p^* is dual optimal.

Aside from simplex method, there are other special methods for solving the shortest path problems, such as Dijkstra's algorithm discussed later on in Chapter 5.

4.4 Maximum Flow Problem

Maximum flow problem is defined for a capacitated directed graph $G = (\mathcal{N}, \mathcal{L})$ with link capacities $u_{ij} > 0$ for $(i, j) \in \mathcal{L}$. There is a source node s and a terminal node t, and we want to send the largest possible amount of flow from node s to node t while obeying the link capacities.

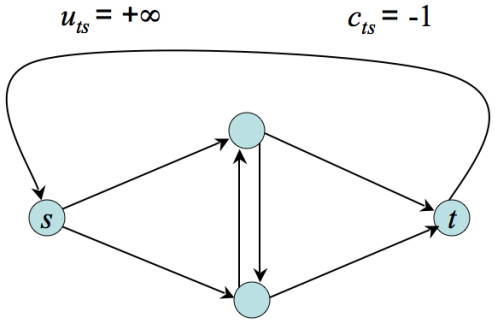

Figure 4.5: Augmentation of the original graph with the new link to accommodate the max-flow formulation as a min-cost problem.

The maximum flow problem can be formulated as follows:

$$\begin{array}{lll} \text{maximize} & b_s & \\ \text{subject to} & Ax = b & \\ & b_t = -b_s & \\ & b_i = 0 \qquad \text{for all } i \neq s, t & \\ & 0 \leq x_{ij} \leq u_{ij} \qquad \text{for all } (i,j) \in \mathcal{L}. & (4.6) \end{array}$$

This problem can be modeled as a min-cost network problem, as follows: we introduce a new link (t, s) with the infinite capacity, $u_{ts} = +\infty$. The augmented network is illustrated in Figure 4.5. We define the cost of each link $(i, j) \in \mathcal{L}$ to be zero, and the cost of the new link to be -1, i.e.

$$c_{ts} = -1 \qquad \text{and} \qquad c_{ij} = 0 \quad \text{for all } (i,j) \in \mathcal{L}.$$

Thus minimizing the cost $\sum_{(i,j)} c_{ij}x_{ij}$ in the the new network is the same as maximizing x_{ts} on the new link. This flow is the returned flow of what comes from node s to node t, thus maximizing x_{ts} is the same as solving the original problem.

Again, a simplex method can be applied to solve the min-cost network flow formulation. However, there is another method that applies to the max-flow problem directly and exploits the special structure of the problem. The method is an adaptation of the negative cost cycle algorithm, and it is known as the *Ford-Fulkerson algorithm.*

The Ford-Fulkerson algorithm starts with a feasible flow x, which is a flow that satisfies the constraints in the max-flow problem (4.6). Then, the algorithm searches for possibility to increase the flow along some additional path. A path along which a positive flow can be pushed from the source node s to the terminal node t is an *augmenting path*, formally defined as follows.

Definition 8 *Let x be a feasible flow for the max-flow problem. An augmenting path $\mathcal{P}$ is a path from s to t such that*

$$x_{ij} < u_{ij} \qquad \textit{for all } (i,j) \in F \qquad \text{(forward links in the path } \mathcal{P}),$$

$$x_{ij} > 0 \qquad \textit{for all } (i,j) \in B \qquad \text{(backward links in the path } \mathcal{P}\text{)}.$$

As seen from the definition, an augmenting path is a path along which an additional positive flow can be sent. This is because none of the links in such a path are utilized to the full capacity: for each forward link (i,j) in the path, the flow x_{ij} is not at the maximum capacity u_{ij}, while for each backward link (i,j) in the path, the flow x_{ij} is not zero. Hence, we can push an additional amount of flow $\delta > 0$ along the path $\mathcal{P}$. This is referred to as *flow augmentation*. The maximum amount of flow that can be pushed along $\mathcal{P}$ is $\delta(\mathcal{P})$ given by

$$\delta(\mathcal{P}) = \min \left\{ \min_{(i,j)\in F} (u_{ij} - x_{ij}), \quad \max_{(i,j)\in B} x_{ij} \right\}. \tag{4.7}$$

If the path $\mathcal{P}$ has no backward links [$B = \emptyset$] and all the capacities of the (forward) links are infinity [$u_{ij} = +\infty$ for all $(i,j) \in \mathcal{P}$], then

$$\delta(\mathcal{P}) = +\infty.$$

In this case, the maximal flow from s to t is unbounded [$f^* = +\infty$].

If all the forward links in the path $\mathcal{P}$ have finite capacities, then after pushing $\delta(\mathcal{P})$ units of flow along $\mathcal{P}$, at least one of the links in the path $\mathcal{P}$ will be used to its maximum capacity. Such a link is referred to as *a saturated link*.

The typical *iteration of the Ford-Fulkerson algorithm* for the max-flow problem is as follows:

1. Start with a feasible flow x.
2. Find an augmenting path. If such a path does not exist, terminate.
3. Determine the maximum possible amount of flow $\delta(\mathcal{P})$. If $\delta(\mathcal{P}) = +\infty$, terminate.
4. Push the amount $\delta(\mathcal{P})$ along the path $\mathcal{P}$ and obtain a new feasible solution. Go to step 2.

We have the following result for the algorithm.

Theorem 50 *Suppose that the capacity u_{ij} of each link $(i,j) \in \mathcal{L}$ is integer. Also, assume that the Ford-Fulkerson algorithm is initiated with a feasible integral flow. Then, the link flows remain integer during the execution of the algorithm. Furthermore, the algorithm terminates in a finite number of iterations.*

Proof. Since all link capacities are integer, they are finite, so the maximum flow from s to t is finite. When the initial feasible flow is integer, then according to Eq. (4.7), the flow amount $\delta(\mathcal{P})$ is integer for any augmenting path $\mathcal{P}$. Hence, the integrality of the flows is maintained throughout the algorithm. At every iteration, the algorithm sends an additional positive flow $\delta(\mathcal{P})$, which is at least 1 [since $\delta(\mathcal{P})$ is integer]. Since the maximal possible flow from s to t is finite, the algorithm must terminate in a finite number of steps. ■

The maximum flow is related to the minimum cut. To show this relation, we define a specific cut in a graph $G = (\mathcal{N}, \mathcal{L})$ corresponding to two special nodes s and t.

Definition 9 *Let $G = (\mathcal{N}, \mathcal{L})$ be a given directed graph, and let s and t be two given nodes in the graph. A subset $S \subset \mathcal{N}$ is an s-t cut if $s \in S$ and $t \notin S$.*

We refer to the set S as *a cut*. We define the capacity $C(S)$ of the cut S as follows:

$$C(S) = \sum_{\{(i,j)\in\mathcal{L} | i\in S,\ j\notin S\}} u_{ij}.$$

Any path from s to t must cross the cut S, i.e., must contain a link $(i, j) \in \mathcal{L}$ with $i \in S$ and $j \notin S$. Thus, any feasible flow x from s to t cannot exceed the capacity of the cut S,

$$x \leq C(S).$$

In particular, the maximum flow f^* cannot exceed the capacity of the cut,

$$f^* \leq C(S). \tag{4.8}$$

Hence, the capacity of any cut S is an upper bound on the maximum flow. In particular, the cut with the minimum capacity provides the tightest upper bound, which is equal to the maximum flow f^*. This can be seen by considering a suitable primal-dual pair of linear problems, and then applying the linear duality theorem. Here, we provide a different proof, using Ford-Fulkerson algorithm.

Theorem 51 *(Max-Flow Min-Cut) The value of the maximum flow is equal to the minimum cut capacity.*

Proof. If the max flow is infinite, then from inequality (4.8), it follows that the minimum cut capacity is infinite.

Suppose now that the maximum flow is finite. Let x^* be the network flow obtained at the end of the Ford-Fulkerson algorithm. Note that this flow is optimal, and let the optimal flow value from s to t be f^*. Define the cut S, as follows

$$S = \{s\} \cup \{i \in \mathcal{N} \mid \text{there is a path from } s \text{ to } i \text{ with unsaturated links}\}.$$

Note that $t \notin S$, for otherwise there is an augmenting path from s to t, in which case the Ford-Fulkerson algorithm would increase the flow - a contradiction. Each link $(i, j) \in \mathcal{L}$ with $i \in S$ and $j \notin S$ must be saturated, i.e., the flow is $x^*_{ij} = u_{ij}$; otherwise, the node j would also be in the set S. Hence, for the cut S capacity, we have

$$C(S) = \sum_{\{(i,j)\in\mathcal{L} | i\in S,\ j\notin S\}} u_{ij} = \sum_{\{(i,j)\in\mathcal{L} | i\in S,\ j\notin S\}} x^*_{ij} = f^*.$$

Hence, S is the minimum capacity cut, and its capacity is equal to the max flow. ■

4.5 Routing in Communication Network

Here, we discuss the problem of routing in the presence of congestion. In this problem, the rates at which the users generate the traffic are fixed, and the routes (or paths) of the user flow should be sent through the network optimally, given the costs of the links in the network. We have considered this problem in Example 9 of Section 2.3.3. We slightly modify the notation used there.

Given a network, let $\mathcal{S}$ be a set of users (origin-destination pairs). Let x_s be the traffic flow generated by user s. The traffic flow of user s is to be routed through the network along directed paths. If the traffic of user s is routed along a path p, we write $p \in s$. Define a user-path incidence matrix D with entries

$$D_{sp} = 1 \quad \text{if } p \in s \qquad \text{and} \qquad D_{sp} = 0 \quad \text{otherwise.}$$

Let $\mathcal{L}$ be the set of links in the network, and $\mathcal{P}$ be the set of paths. If a link l is traversed by a path p, we write $l \in p$. Let A be the link-path incidence matrix, with entries given by

$$A_{lp} = 1 \quad \text{if } l \in s \qquad \text{and} \qquad A_{lp} = 0 \quad \text{otherwise.}$$

Let v_p denote the flow along a path p. The corresponding flow on link l is denoted by z_l, and we have

$$z_l = \sum_{\{p \,|\, l \in p\}} v_p \qquad \text{for all } l \in \mathcal{L},$$

or compactly $z = Av$.

There is a cost associated with using the links $\mathcal{L}$ of the network, namely, the cost of sending a flow z_l on the link l is $f_l(z_l)$. The problem is to decide on paths along which the user flows x_s should be routed, so as to minimize the total cost. More formally, the problem is to determine the variables v_p that minimize the cost, while supporting the given flow rate x_s for each user s. The routing problem is given by

$$\begin{array}{ll} \text{minimize} & f(v) = \displaystyle\sum_{l \in \mathcal{L}} f_l \left(\sum_{\{p \,|\, l \in p\}} v_p \right) \\ \text{subject to} & Dv = x \\ & v \geq 0, \end{array}$$

where $v = [v_p,\ p \in \mathcal{P}]$ is a variable vector of path flows and $x = [x_s,\ s \in \mathcal{S}]$ is a given vector of user rates.

This mathematical model can represent a general resource allocation problem, where some resource l other than links is considered. However, here we focus on routing and view the links as the resource to be allocated.

Assume that the link costs f_l are continuously differentiable convex functions. In this case, as seen in Example 9 of Section 2.3.3, the optimality condition for the routing problem is: for $p \in s$ with $v_p^* > 0$ we have

$$\sum_{l \in p} f_l'(z_l^*) \leq \sum_{j \in \tilde{p}} f_j'(z_j^*) \qquad \text{for all } \tilde{p} \in s,$$

with $z_l^* = \sum_{\{p \,|\, l \in p\}} v_p^*$. The vector v^* must be feasible i.e., $Hv^* = x$ and $v^* \geq 0$.

We summarize this in the following lemma.

Lemma 15 *A feasible flow vector v, supporting the given rate demand x, is an optimal solution to the routing problem if and only if there exists a vector $d = [d_s,\ s \in \mathcal{S}]$ such that*

$$d_s \leq \sum_{j \in p} f_j'(z_j) \qquad \textit{for all } p \in s$$

with equality holding only when $v_p > 0$. Here, z_j represents the total flow on link j resulting from the flow vector v, i.e., $z_l = \sum_{\{\tilde{p} \,|\, j \in \tilde{p}\}} v_{\tilde{p}}$.

To interpret the result, let the length of a path p be $\sum_{l \in p} f_l'([Av]_l)$ when a flow is x (recall $z = Av$). Let d_s represents the length of the shortest path for user s. The lemma says that a feasible vector v is a solution, when for each user s, the flows v_p on paths $p \in s$ are positive only on the shortest paths for that user.

We now describe the *flow deviation algorithm* for solving the routing problem. This method is a version of the Frank-Wolfe method exploiting the special structure of the problem. Let v^k be the current flow. We formulate the auxiliary minimization problem

$$\begin{array}{ll} \text{minimize} & \nabla f(v^k)(v - v^k) \\ \text{subject to} & Dv = x \\ & v \geq 0, \end{array}$$

where $v^k = [v_p^k,\ p \in \mathcal{P}]$ is the current flow route. This a standard LP problem with a compact constraint set. Thus, an optimal solution exists $\hat{v}^k$ and satisfies the following (necessary and sufficient) optimality conditions:

$$\nabla f(v^k)(v - \hat{v}^k) \geq 0 \qquad \text{for all } v \text{ such that } Dv = x \text{ and } v \geq 0.$$

The optimality condition suggests that, for each user s, the flow $\hat{v}_p^k$ should be positive only on the shortest paths for that user with respect to the link prices $f_j'(z_j^k)$ (where z_j^k is the flow induced by the current flow route v^k). Once the shortest-path route $\hat{v}_k$ is determined, we select the stepsize α by the exact search, i.e., find $\alpha_k \in (0, 1]$ that minimizes $f(v^k + \alpha(\hat{v}^k - v^k))$ (the smallest cost achievable on the segment $[v^k, \hat{v}^k]$),

$$f(v^k + \alpha_k(\hat{v}^k - v^k)) = \min_{0 \leq \alpha \leq 1} f(v^k + \alpha(\hat{v}^k - v^k)).$$

Define the new flow v^{k+1} as

$$v^{k+1} = v^k + \alpha_k(\hat{v}^k - v^k).$$

Note that the flow $v^k + \alpha(\hat{v}^k - v^k)$ corresponds to, for each user s, removing αv_p^k-amount of flow from the flow v_p^k for $p \in s$ if p is not the shortest path for user s (shortest with respect to the prices induced by $\nabla f(v^k)$, and then re-allocating these amounts on the shortest path of user s.

The preceding describes a typical iteration of the flow deviation algorithm. We summarize it formally as follows.

Flow Deviation Iteration. Given the current flow v^k, perform the following:

1. Compute the link prices $f_j'(z_j^k)$, where z_j^k is the flow induced by the current flow v^k, i.e., $z_j^* = \sum_{\{p \,|\, j \in p\}} v_p^k$.

2. For each user s, determine the shortest path p_s with respect to the prices $f_j'(z_j^k)$.

3. Choose $\alpha \in (0, 1]$. For each user s and each path p that is not the shortest path for user s, reallocate an α-fraction of the flow v_p^k to the shortest path, i.e., generate a deviation flow $\tilde{v}$ as follows: for each user s,

$$\tilde{v}_p = \begin{cases} (1-\alpha)v_p & \text{if } p \in S \text{ and } p \neq p_s \\ v_p + \alpha \sum_{\{\tilde{p} \,|\, \tilde{p} \neq p, \tilde{p} \in s\}} v_{\tilde{p}} & \text{if } p = p_s, \end{cases}$$

4. Determine $\alpha \in [0, 1]$ minimizing the cost $f(\tilde{v})$. The resulting flow defines v^{k+1}.

An application of a scaled projection gradient method [using the second order information $\nabla^2 f(v)$] to the routing problem can be found in [8] Section 5.7.

4.6 Joint Routing and Congestion Control

Here, we are interested in the network problem when the user rates are not fixed. The congestion control adjusts the traffic rates so that the network resources are fairly shared among the users, and the network is at a reasonable operating point balancing the throughput with the delay. The model for joint routing and congestion control presented in this section is based on utility maximization framework (this is the model developed in [19]).

The utility functions have been used in economic market models to quantify preferences of users (consumers) for certain resources (commodities). A utility function $u(x)$ "measures" the value of a resource amount x to a user. Typically, it is assumed that a utility function is concave, nondecreasing, and continuously differentiable (scalar) function, defined on the interval $[0, +\infty)$. Some of the common examples include

- *Log-function*: $u(x) = w \ln x$ for some scalar $w > 0$.
- *Power-function*: $u(x) = w \frac{x^{1-\alpha}}{1-\alpha}$ for some scalars $w > 0$ and $\alpha > 0$.

The derivative of u is referred to as the marginal utility per unit resource, since $u(x + \delta) = u(x) + \delta u'(x) + o(\delta)$ for a small $\delta > 0$. Due to the assumed concavity of u, the marginal utility $u'(x)$ is nonincreasing.

Congestion control is a mechanism for adjusting the user rates $x = [x_s,\ s \in \mathcal{S}]$ fairly with respect to user utility functions $u_s,\ s \in \mathcal{S}]$. The network performance is quantified in terms of the user utilities and the cost of routing the traffic. The joint routing and congestion control problem is to determine both the user rates $x = [x_s,\ s \in \mathcal{S}]$ and the paths $v = [v_p,\ p \in \mathcal{P}]$ so as to maximize the network performance. Formally, the problem is given by

$$\text{maximize} \qquad U(x) = \sum_{s \in \mathcal{S}} u_s(x_s) - \sum_{l \in \mathcal{L}} f_l \left(\sum_{\{p \,|\, l \in p\}} v_p \right)$$

$$\begin{array}{ll} \text{subject to} & Dv = x \\ & x \geq 0,\ v \geq 0. \end{array} \tag{4.9}$$

This is a concave maximization problem with a convex (in fact, polyhedral) constraint set. For this problem, the optimality condition of Theorem 20 reduces to the following: the feasible vectors x and v are solution to problem (4.9) if and only if x and v are feasible [i.e., $Dv = x$, $x \geq 0,\ v \geq 0$] and for each $s \in \mathcal{S}$,

$$u_s'(x_s) \leq \sum_{l \in p} f_l' \left(\sum_{\{p \,|\, l \in p\}} v_p \right) \qquad \text{for all } p \in s, \tag{4.10}$$

with equality holding when $v_p > 0$.

We interpret the length $f_l' \left(\sum_{\{p \,|\, l \in p\}} v_p \right)$ of link l as the cost of using the link l. In view of this, the preceding relation means that, at optimal x and v, for the paths p carrying the flow of user s, the cost of any path p with $v_p > 0$ is equal to the user's marginal utility, while the cost of any path p with $v_p = 0$ is no less than the user marginal utility.

In some applications, there are explicit constraints on the link capacities, and the problem of joint routing and congestion control is given by

$$\begin{array}{ll} \text{maximize} & U(x) = \sum_{s \in \mathcal{S}} u_s(x_s) \\ \text{subject to} & Dv = x,\ Av \leq c \\ & x \geq 0,\ v \geq 0, \end{array} \tag{4.11}$$

where $c = [c_l,\ l \in \mathcal{L}]$ is the vector with entries c_l representing the capacity of link l.

Consider the dual of problem (4.11) obtained by assigning the prices to the link constraints $Av \leq c$. The dual function is given by

$$q(\mu) = \max_{Dv = x,\ x \geq 0,\ v \geq 0} \left\{ \sum_{s \in \mathcal{S}} u_s(x_s) - \mu^T A x \right\} + \mu^T c \qquad \text{for } \mu \geq 0.$$

By the KKT conditions of Theorem 34, we have that x^* and v^* are primal optimal (and μ^* dual optimal) if and only if: x^* and v^* are feasible, $\mu^* \geq 0$, and such that they satisfy the complementarity slackness and (x^*, v^*) attains the maximum in $q(\mu^*)$. Formally, the complementarity slackness is given by

$$\mu_l^* = 0 \quad \text{only if} \quad \textstyle\sum_{\{p \,|\, l \in p\}} v_p^* < c_l. \tag{4.12}$$

Furthermore, (x^*, v^*) attains the maximum in $q(\mu^*)$ if and only if for each s,

$$u_s'(x_s^*) \leq \sum_{l \in p} \mu_l^* \qquad \text{for all } p \in s. \tag{4.13}$$

with equality only when $v_p^* > 0$. This relation is similar to the relation in Eq. (4.10), where the cost f_l' "plays the role of" the multiplier μ_l^*.

Note that when $v_p^* > 0$, the values $\sum_{l \in p} \mu_l^*$ for $p \in s$ are the same. Denote this value by μ_s^*. By interpreting μ_s^* as the price per unit rate for user s, from (4.13), we have

$$u_s'(x_s^*) = \mu_s^* \text{ when } \mu_s^* > 0 \qquad \text{and} \qquad u_s'(x_s^*) \leq \mu_s^* \text{ when } \mu_s^* = 0.$$

The preceding is the optimality condition for the following problem for user s,

$$\begin{array}{ll} \text{maximize} & u(x_s) - \mu_s^* x_s \\ \text{subject to} & x_s \geq 0. \end{array}$$

By introducing a new variable $w_s = \mu_s^* x_s$, we can rewrite the preceding problem as follows:

$$\begin{array}{ll} \text{maximize} & u\left(\dfrac{w_s}{\mu_s^*}\right) - w_s \\ \text{subject to} & w_s \geq 0. \end{array} \tag{4.14}$$

The relation $w_s = \mu_s^* x_s$ implies that $w_s/x_s = \mu_s^*$ can be interpreted as optimality condition at x^*, as follows

$$\frac{w_s^*}{x_s^*} = \mu_s^* \text{ when } \mu_s^* > 0,$$

which corresponds to maximizing $w_s^* \ln x_s - \mu_x^* x_s$ over $x_s \geq 0$. This together with the feasibility of x^* and v^*, and the complementarity slackness of Eq. (4.12), imply by KKT conditions that w^*, x^* and v^* constitute an optimal solution to the problem

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{s \in \mathcal{S}} w_s \ln x_s \\ \text{subject to} & Dv = x,\ Av \leq c \\ & x \geq 0,\ v \geq 0, \end{array} \tag{4.15}$$

Thus, by introducing a new variable $w_s = \mu_s^* x_s$ and through the use of KKT conditions, we have found that the original joint routing and congestion control problem (4.11) is equivalent to the set of users problems of Eq. (4.14) and a network problem of Eq. (4.15).

A similar transformation can be considered for joint routing and congestion control problem without the link capacity constraints [cf. problem (4.11)]. This is discussed in detail in lecture notes [17].

The approach discussed above is precisely the approach proposed by Kelly *et. al* in [19]. The key idea in the approach is to introduce a price w_s that user s is willing to pay for his rate x_s. The price per unit rate is w_s/x_s for user s. The network receives x_s and w_s from each user $s \in \mathcal{S}$, and interprets the ratio w_s/x_s as "marginal utility" per unit of flow for user s. Thus, the network generates "surrogate utility functions"

$$\tilde{u}_s(x_s) = w_s \ln x_s \qquad \text{for each } s \in \mathcal{S}.$$

With these utility functions, the network problem of Eq.(4.11) becomes

$$\text{maximize} \quad \sum_{s \in \mathcal{S}} w_s \ln x_s$$

$$\begin{aligned} \text{subject to} \quad & Dv = x,\ Av \le c \\ & x \ge 0,\ v \ge 0. \end{aligned} \tag{4.16}$$

The network chooses the link prices μ_l as as optimal multipliers (link prices) for this problem. For each user s, the resulting price μ_s per unit of flow is given to user s, where $\mu_s = \sum_{l \in p} \mu_p$ for any $p \in s$ (these are the same for any $p \in s$ with $x_p > 0$). The user problem is to maximize $u_s(x_s)$ minus the pay $w_r = s$, subject to $w_s \ge 0$. Since $x_s = w_s/\mu_s$, a user problem is

$$\begin{aligned} \text{maximize} \quad & u\left(\frac{w_s}{\mu_s}\right) - w_s \\ \text{subject to} \quad & w_s \ge 0. \end{aligned} \tag{4.17}$$

The importance of the preceding formulations is the decomposition of the problem. The original problem of joint routing and congestion control formulated in Eq. (4.11) is a large optimization problem involving both user and network information. Through the use of "willingness to pay" variable and the KKT conditions, the problem is suitably decomposed into: a network problem (4.16) that does not require the information about user utility functions, and a set of user problems (4.17) that does not require any knowledge about the network (topology). Evidently, the users and the network have to exchange some information.

4.7 Rate Allocation in Communication Network

The rate allocation problem is a special case of problem (4.11), where the routing is fixed (i.e., v is now given, and $Dv = x$ is satisfied) and the problem is to allocate rates x_s for users $s \in \mathcal{S}$ optimally. The resulting rate allocation problem is

$$\begin{aligned} \text{maximize} \quad & \sum_{s \in \mathcal{S}} u_s(x_s) \\ \text{subject to} \quad & \sum_{s \in \mathcal{S}(l)} x_s \le c_l \qquad \text{for all } l \in \mathcal{L} \\ & x \ge 0, \end{aligned}$$

where for each s, the set $\mathcal{S}(l)$ is the set of all users s whose traffic uses link l. We consider this problem with additional constraints, namely, the rate of user s is constrained within an interval $x_s \in [m_s, M_s]$, where $m_s \ge 0$ is the minimum and $M_s < \infty$ is the maximum rate for user s.

With these additional rate constraints, the rate allocation problem is given by

$$\begin{aligned} \text{maximize} \quad & \sum_{s \in \mathcal{S}} u_s(x_s) \\ \text{subject to} \quad & \sum_{s \in \mathcal{S}(l)} x_s \le c_l \qquad \text{for all } l \in \mathcal{L} \\ & x_s \in I_s, \quad I_s = [m_s, M_s] \qquad \text{for all } s \in \mathcal{S}. \end{aligned} \tag{4.18}$$

In what follows, we discussed a dual algorithm given by Low and Lapsley [21] for solving problem (4.18). It is assumed that *each utility u_s is strictly concave and increasing.* Under this assumption, the problem has an optimal solution [the constraint set is compact], and the optimal solution is unique by strict concavity of the utility functions.

The objective function of problem (4.18) is separable in the variables x_s, and these variables are coupled only through the link capacity constraints. Thus, by assigning prices to the link capacities, we obtain a dual problem of the form

$$\begin{array}{ll} \text{minimize} & q(\mu) \\ \text{subject to} & \mu \geq 0, \end{array} \tag{4.19}$$

where the dual function is

$$\begin{aligned} q(\mu) &= \max_{x_s \in I_s} \sum_{s \in \mathcal{S}} u_s(x_s) - \sum_{l \in \mathcal{L}} \mu_l \left(\sum_{s \in \mathcal{S}(l)} x_s - c_l \right) \\ &= \max_{x_s \in I_s} \sum_{s \in \mathcal{S}} \left(u_s(x_s) - x_s \sum_{l \in \mathcal{L}(s)} \mu_l \right) + \sum_{l \in \mathcal{L}} \mu_l c_l, \end{aligned}$$

where $\mathcal{L}(s)$ is the set of all links l carrying the flow of user s. Defining the variables $p_s = \sum_{l \in \mathcal{L}(s)} \mu_l$ and the functions

$$Q_s(p_s) = \max_{x_s \in I_s} \{u_s(x_s) - x_s p_s\}, \tag{4.20}$$

the dual function can be expressed as

$$q(\mu) = \sum_{s \in \mathcal{S}} Q_s(p_s) + \sum_{l \in \mathcal{L}} \mu_l c_l. \tag{4.21}$$

Given the link prices $\mu_l,\ l \in \mathcal{L}$ and the resulting prices $p_s = \sum_{l \in \mathcal{L}(s)} \mu_l$ as seeing by the users $s \in \mathcal{S}$, for each s, the rate attaining the dual function value $Q_s(p_s)$ is denoted by $x_s(p_s)$. Note that the maximizer $x_s(p_s)$ in the problem of (4.20) is unique and given by

$$x_s(p_s) = P_{I_s}[u_s'(p_s)^{-1}], \tag{4.22}$$

where ${u_s'}^{-1}$ is the inverse function of the derivative u_s', and $P_{I_s}[z]$ denotes the projection on the (closed convex) interval I_s, which is in particular given by $P_{I_s}[z] = \min\{\max\{m_s, z\},\ M_s\}$.

We now consider a dual algorithm for rate allocation problem (4.18). In what follows, we assume that the problem is feasible and that each utility function u_s is strictly concave, twice differentiable and increasing on the interval I_s. Furthermore, for each s, the curvature of u_s is bounded away from zero on the interval I_s, i.e.,

$$-u_s''(z) \geq \frac{1}{a_s} > 0 \qquad \text{for all } z \in I_s \text{ and some } a_s > 0.$$

Under this condition, the rate problem has a unique optimal solution. Furthermore, note that the strong duality holds for the primal problem (4.18) and the dual problem (4.19)–(4.21) (recall the strong duality result of Theorem 31 of Section 2.5.5 for convex objective and linear constraints).

As noted earlier, under the preceding assumptions, the maximizer $x_s(p_s)$ in problem (4.20) exists and it is unique, implying that the dual function $q(\mu)$ is differentiable with the partial derivatives given by

$$\frac{\partial q(\mu)}{\partial \mu_l} = c_l - \sum_{s \in \mathcal{S}(l)} x_s(p_s).$$

As a matter of fact, the dual function $q(\mu)$ is continuously differentiable (see Bertsekas and Tsitsiklis [10], page 669).

Let $\mu_l(k)$ be the link prices at a given time k, and $\mu(k)$ be the vector of these prices. Let $x_s(k)$ be the maximizer given by Eq. (4.22) for $p_s = \sum_{l \in \mathcal{L}(s)} \mu_l(k)$. Consider the following gradient projection method for minimizing $q(\mu)$:

$$\mu_l(k+1) = \left[\mu_l(k) - \alpha_k \frac{\partial q(\mu(k))}{\partial \mu_l} \right]^+ \qquad \text{for all } l \in \mathcal{L},$$

where $\alpha_k > 0$ is a stepsize. Equivalently, the method is given by

$$\mu_l(k+1) = \left[\mu_l(k) + \alpha_k \left(\sum_{s \in \mathcal{S}(l)} x_s(k) - c_l \right) \right]^+ \qquad \text{for all } l \in \mathcal{L}. \tag{4.23}$$

Note that, given the aggregate rate $\sum_{s \in \mathcal{S}(l)} x_s(k)$ of the traffic through link l, the iterations of algorithm (4.23) are completely distributed over the links, and can be implemented by individual links using local information only.

By interpreting the set $\mathcal{L}$ of links and the set $\mathcal{S}$ of users as processors in a distributed system, the dual problem can be solved. In particular, given the link prices $\mu_l(k)$, the aggregate link price $\sum_{l \in \mathcal{L}(s)} \mu_l(k)$ is communicated to user s. Each user s evaluates its corresponding dual function $Q_s(p_s)$ of Eq. (4.20) [i.e., user s computes the maximizer $x_s(p_s)$]. Each user s communicates its rate $x_s(p_s)$ to links $l \in \mathcal{L}(s)$ [the links carrying the flow of user s]. Every link l updates its price $\mu_l(k)$ according to the gradient projection algorithm [cf. Eq. (4.23)]. The updated aggregate link prices $\sum_{l \in \mathcal{L}(s)} \mu_l(k+1)$ are communicated to users, and the process is repeated. We formally summarize these steps in the following algorithm.

Dual Gradient Projection Algorithm. At times $k = 1, \ldots$, each link $l \in \mathcal{L}$ performs the following steps:

1. Receives the rates $x_s(k)$ from users $s \in \mathcal{S}(l)$ using the link.

2. Updates its price

$$\mu_l(k+1) = \left[\mu_l(k) + \alpha_k \left(\sum_{s \in \mathcal{S}(l)} x_s(k) - c_l \right) \right]^+ .$$

3. Communicates the new price $\mu_l(k+1)$ to all users $s \in \mathcal{S}(l)$ using the link l.

At times $k = 1, \ldots$, each user $s \in \mathcal{S}$ performs the following steps:

1. Receives the aggregate price $p_s(k) = \sum_{l \in \mathcal{L}(s)} \mu_l(k)$ [sum of link prices over the links carrying its flow].

2. Computes its new rate by $x_s(k+1) = x_s(p_s(k)$ [i.e., determines the maximizer in $Q_s(p_s(k))$].

3. Communicates the new rate $x_s(k+1)$ to all links $l \in \mathcal{L}(s)$ [the links in its flow path].

In the preceding, we have not specified the stepsize α_k. Under the assumption that the second derivative of each utility u_s is bounded away from zero by $1/a_s$, it can be seen that the gradient of the dual function is Lipschitz continuous, i.e.,

$$\|\nabla q(\mu) - \nabla q(\tilde{\mu})\| \leq L \|\mu - \tilde{\mu}\| \qquad \text{for all } \mu, \tilde{\mu} \geq 0.$$

with constant L given by

$$L = \max_{s \in \mathcal{S}} a_s \ \max_{s \in \mathcal{S}} |\mathcal{L}(s)| \ \max_{l \in \mathcal{L}} |S(l)|. \tag{4.24}$$

We next discuss a convergence result for the method. We use x to denote the vector of user rates $[x_s,\ s \in \mathcal{S}]$ and μ to denote the vector of link prices $[\mu_l,\ l \in \mathcal{L}]$. We assume that the method is started with initial rates $x_s(0) \in I_s$ for all s and initial prices $\mu_l(0) \geq 0$ for all l. The constant stepsize can be used, as seen from the following theorem, which is established in [21].

Theorem 52 *Assume that each utility function u_s is strictly concave, twice differentiable and increasing on the interval I_s. Furthermore, assume that for each s, the curvature of u_s is bounded away from zero on the interval I_s [i.e., $-u_s''(z) \geq 1/a_s > 0$ for all $z \in I_s$]. Also, assume that the constant stepsize $\alpha_k = \alpha$ is used in the dual gradient projection method, where $0 < \alpha < \frac{2}{L}$ and L as given in Eq. (4.24). Then, every accumulation point (x^*, μ^*) of the sequence $\{(x(k), \mu(k))\}$ generated by the dual gradient projection algorithm is primal-dual optimal.*

Chapter 5

Dynamic Programming

In this chapter, we study the dynamic programming approach to for sequential decision making in the presence of uncertainties. In particular, the decisions are made in stages and, at each stage, the decision is influencing the future outcomes. The overall goal is to find decisions that optimize the long term cost in the presence of dynamics and uncertainties.

We discuss the basic ingredients of a dynamic programming problem and the basic underlying optimality principle, summarized in Bellman's equation. Our focus is on infinite horizon problems, and in particular, stochastic shortest path problems, discounted cost problems, and average cost problems. An in depth treatment of dynamic programming is given in the two-volume textbook by Bertsekas [6] and [7].

5.1 Fundamental Concepts and Problem Formulation

A dynamic programming (DP) problem is the problem of determining the optimal decisions over a horizon of time. The time is discrete, and we use the subscript k to denote the time index. Any DP problem has the following basic components:

1. *Dynamics.* An underlying dynamic system whose state at time k is denoted by x_k. The system dynamics is described by an evolution equation specifying the state at time $k+1$ as a function of the current state x_k, a decision (control) u_k made at time k and an uncertainty ξ_k in the decision outcome. Formally, we have

$$x_{k+1} = F_k(x_k, u_k, \xi_k),$$

   where F_k is some function. The set of possible states x_k at time k is denoted by S_k.

2. *Decisions.* At each time k, the possible decisions at a given state x_k are specified by a set $U_k(x_k)$. Formally, the decisions are constrained

$$u_k \in U_k(x_k) \qquad \text{for all } k.$$

   Note that the decisions can be time dependent and state dependent.

3. *Uncertainties.* At each time k, the uncertainty ξ_k has a probability distribution $P_k(\cdot\,; x_k, u_k)$ that depends on the current state x_k and the decision u_k, but does not depend on the past uncertainties $\xi_0, \ldots, \xi_{k-1}$.

4. *Cost.* At time k, the cost $g_k(x_k, u_k, \xi_k)$ is associated when at state x_k, the decision u_k is made with uncertainty ξ_k in the outcome. The overall *cost incurred over the time horizon is additive.*

When the time horizon is finite, the *total expected cost* accumulated starting from an initial state x_0 is given by

$$\mathrm{E}\left\{\sum_{k=0}^{N-1} g_k(x_k, u_k, \xi_k) + g_N(x_N) \,\middle|\, x_0\right\},$$

where N is the last time period with terminal cost g_N. The conditional expectation is taken with respect to the random variables x_k and ξ_k.

Example 32 *(Inventory Control, see Example 1.1.1 of [6])*
Consider a problem of ordering a stock quantity at each of N periods of time to minimize the expected cost. We introduce the following notation:

- *x_k stock quantity at time k.*

- *u_k stock quantity ordered (and available) at time k.*

- *ξ_k uncertain demand at time k.*

The stock quantity evolves in time according to the following relation

$$x_{k+1} = x_k + u_k - \xi_k,$$

where negative stock represents the back-logged demand.

The cost at time k is given by

$$g_k(x_k, u_k, \xi_k) = cu_k + r(x_k + u_k - \xi_k),$$

where c is the cost per unit of the stock, and $r(x_k + u_k - \xi_k)$ is the penalty for positive stock (holding cost for excess inventory) or penalty for negative stock (shortage cost for unfulfilled demand).

The total expected cost incurred over N stages is

$$\mathrm{E}\left\{\sum_{k=0}^{N-1} cu_k + r(x_k + u_k - \xi_k)\right\}.$$

The problem is to minimize this cost subject to $u_0 \geq 0, \ldots, u_{N-1} \geq 0$ (i.e., $U_k = [0, +\infty)$ for all k).

A *policy* π is an ordered set of functions

$$\pi = (\mu_0, \ldots, \mu_{N-1}),$$

where each $\mu_k : x_k \mapsto u_k$ such that $u_k \in U_k(x_k)$ for all $x_k \in S_k$. The expected cost of policy π starting from state x_0 is

$$V_\pi(x_0) = \mathrm{E}\left\{ \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), \xi_k) + g_N(x_N) \,\middle|\, x_0 \right\},$$

An *optimal policy* π^* is the policy that minimizes the expected cost i.e., π^* such that

$$V_{\pi^*}(x_0) = \min_{\pi \in \Pi} V_\pi(x_0),$$

where Π is the set of all policies. The (finite horizon) DP problem is to determine the optimal policy $\pi^* = \left(\mu_0^*, \ldots, \mu_{N-1}^*\right)$. *The optimal cost* of starting at x_0 is defined by

$$V^*(x_0) = \min_{\pi \in \Pi} V_\pi(x_0).$$

It is often convenient to view the optimal cost as a function that to each state x_0 assigns the optimal cost value $V^*(x_0)$, i.e., as function $V^* : x_0 \mapsto \min_{\pi \in \Pi} V_\pi(x_0)$. We will often refer to this function as *the optimal value function*.

The dynamic programming technique is based on a simple *optimality principle*, which can be stated as: for an optimal policy $\pi^* = \left(\mu_0^*, \ldots, \mu_{N-1}^*\right)$, every truncated policy $(\mu_i^*, \ldots, \mu_{N-1})$ is optimal for the subproblem starting at state x_i and minimizing the "cost-to-go" from time i to time N,

$$\mathrm{E}\left\{ \sum_{k=i}^{N-1} g_k(x_k, \mu_k(x_k), \xi_k) + g_N(x_N) \,\middle|\, x_i \right\}.$$

Intuitively, if the truncated policy $(\mu_i^*, \ldots, \mu_{N-1})$ were not optimal for the subproblem, we could replace it by a subproblem optimal policy, and construct a new policy that at state x_i switches to the subproblem optimal policy. This new policy would have a smaller cost, thus violating the optimality of π^*. Another interpretation of the optimality principle is: given any point C on "the shortest path from point A to point B, the truncated path from C to B is the shortest path from C to B.

The optimality principle suggests that the optimal policy can be constructed by going *backward in time*, i.e., by solving the subproblems

$$\mathrm{E}\left\{ \sum_{k=i}^{N-1} g_k(x_k, \mu_k(x_k), \xi_k) + g_N(x_N) \,\middle|\, x_i \right\}$$

going from $i = N - 1$ down to $i = 0$.

Our interest is the problems with "a stationary" description:

1. The system dynamic is given by a function F that does not vary with time, i.e,

$$x_{k+1} = F(x_k, u_k, \xi_k) \qquad \text{for all } k.$$

   The set S of possible states x_k does not depend on time.

2. The set $U(x_k)$ of possible decisions at a given state x_k does not depend on time.

3. The probability distribution $P(\cdot\,; x, u)$ of the uncertainty ξ does not depend on time.

4. The cost $g(x, u, \xi)$ does not depend on time.

The problems with stationary description are more tractable. Note that the inventory problem in Example 32 has stationary description. Here is another example of a problem with such a description.

Example 33 *(M/M/1 Queue Control, see [17] Chapter 9)*
Consider a system with one server and one queue. Both server and the customer arrival are of exponential type, i.e., the server is characterized by its service rate $\mu > 0$ and the customers arrival rate is λ. Let x_k be the state of the queue (number of the customers) at time k. The customers arrival rate in the queue is λu_k, where $u_k \in [0, 1]$ is a control variable at time k.

The state of the system can go up by 1 or down by 1 depending whether the departure or an arrival occurred. The evolution of the system can be modeled by a Markov chain with states $S = \{0, 1, \ldots\}$ with transition probabilities

$$p_{ij}(u) = \text{Prob}\,\{x_{k+1} = j \mid x_k = i,\, u_k = u\}.$$

The probability matrix $P(u) = [p_{ij}(u)]$ is given by

$$P(u) = \begin{bmatrix} \frac{\mu+\lambda(1-u)}{\lambda+\mu} & \frac{\lambda u}{\lambda+\mu} & 0 & \cdots \\ \frac{\mu}{\lambda+\mu} & \frac{\lambda(1-u)}{\lambda+\mu} & \frac{\lambda u}{\lambda+\mu} & \cdots \\ 0 & \frac{\mu}{\lambda+\mu} & \frac{\lambda(1-u)}{\lambda+\mu} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

There is a cost associated with the size of the queue

$$g(x) = cx + \mu\, I_{x=0},$$

where cx represents the holding cost of customers in the queue, $I_{x=0}$ is the indicator function of the event $x = 0$ (i.e., $I_{x=0} = 1$ if $x = 0$, and otherwise it is zero). The cost $g(0) = \mu$ is the cost associated with lost departure opportunities when the queue is empty.

The goal is to select the controls $u_0, \ldots, u_N$ so that the total expected cost accumulated over N periods of time is minimized, i.e., the goal is to minimize

$$\text{E}\left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k) \,\middle|\, x_0 = x \right\}.$$

The cost here is discounted by a factor $\alpha \in (0, 1)$ at each stage.

5.1.1 DP Algorithm for Finite Horizon Problem

Here, we describe the DP algorithm for finite horizon problem with a stationary description. When the time horizon is finite but large, often a discounted cost model is used, where the cost $g(x, u, \xi)$ is discounted by a factor $\alpha(0, 1)$ at each stage. To accommodate both situations, we will use a discounted model and allow α to take value 1.

The DP algorithm is based on the optimality principle, which under the stationary problem description can be written as: for all $x \in S$ and $k = 0, \ldots, N-1$,

$$V_{k+1}(x) = \min_{u \in U(x)} \mathrm{E}\left\{g(x, u, \xi) + \alpha V_k(F(x, u, \xi))\right\}, \tag{5.1}$$

with initial condition $V_0 = 0$ and discount factor $\alpha \in (0, 1)$. The value $V_k(x)$ is the minimum expected cost accumulated over k stages to go, starting from state x. This basically gives the DP algorithm with backward recursion.

Note that we can compute all the optimal finite horizon cost functions using the DP recursion. This is possible because of the stationary problem description. Note however that the DP recursion has to be solved for all possible states $x \in S$. This may not be efficient especially when the set of states S is large.

The optimal control u_k for $k+1$ stages to go starting from state x is the control that minimizes $\mathrm{E}\left\{g(x, u, \xi) + V_k(F(x, u, \xi))\right\}$ over all $u \in U(x)$. Consider the mapping $\mu_k^*(x)$ such that

$$\mu_k^*(x) \in \operatorname*{Argmin}_{u \in U(x)} \mathrm{E}\left\{g(x, u, \xi) + \alpha V_k(F(x, u, \xi))\right\} \qquad \text{for all } x \in S.$$

Then, the policy $\pi^* = (\mu_0^*, \ldots, \mu_{N-1}^*)$ is optimal.

Again, due to number of the states $x \in S$, it may be computationally inefficient to determine the optimal policy. Often some approximations are used for cost functions or for the policy space (i.e., the search is restricted to a smaller class of policies).

To illustrate the DP algorithm, we revisit Example 33.

Example 34 *The DP algorithm applied to the queue control problem of Example 33 reduces to: for $x \geq 1$,*

$$V_{k+1}(x) = \min_{u \in [0,1]} \left\{ cx + \mu\, I_{x=0} + \alpha \left\{ \frac{\mu}{\lambda+\mu} V_k(x-1) + \frac{\lambda(1-u)}{\lambda+\mu} V_k(x) + \frac{\lambda u}{\lambda+\mu} V_k(x+1) \right\} \right\},$$

and for $x = 0$,

$$V_{k+1}(x) = \min_{u \in [0,1]} \left\{ cx + \mu\, I_{x=0} + \alpha \left\{ \frac{\mu + \lambda(1-u)}{\lambda+\mu} V_k(x) + \frac{\lambda u}{\lambda+\mu} V_k(x+1) \right\} \right\}.$$

By noting that certain terms in the above relations do not depend on the control value u, we can write: for $x \geq 1$,

$$V_{k+1}(x) = cx + \mu\, I_{x=0} + \frac{\alpha\mu}{\lambda+\mu} V_k(x-1) + \alpha \min_{u \in [0,1]} \left\{ \frac{\lambda(1-u)}{\lambda+\mu} V_k(x) + \frac{\lambda u}{\lambda+\mu} V_k(x+1) \right\},$$

and for $x = 0$,

$$V_{k+1}(x) = cx + \mu\, I_{x=0} + \alpha \min_{u\in[0,1]} \left\{ \frac{\mu + \lambda(1-u)}{\lambda+\mu} V_k(x) + \frac{\lambda u}{\lambda+\mu} V_k(x+1) \right\}.$$

By introducing the notation $(x-1)+$ *(which is equal to* 0 *if* $x - 1 \leq 0$ *and equal to* $x - 1$ *otherwise, we obtain*

$$\begin{aligned} V_{k+1}(x) &= cx + \mu\, I_{x=0} + \frac{\alpha\mu}{\lambda+\mu} V_k((x-1)^+) + \alpha \min_{u\in[0,1]} \left\{ \frac{\lambda(1-u)}{\lambda+\mu} V_k(x) + \frac{\lambda u}{\lambda+\mu} V_k(x+1) \right\} \\ &= cx + \mu\, I_{x=0} + \frac{\alpha\mu}{\lambda+\mu} V_k((x-1)^+) + \frac{\alpha\lambda}{\lambda+\mu} \left\{ V_k(x) + \min_{u\in[0,1]} u\,(V_k(x+1) - V_k(x)) \right\}. \end{aligned}$$

Thus, the optimal control μ_k^* *is given by*

$$\mu_k^*(x) = \begin{cases} 0 & \text{if } V_k(x+1) > V_k(x) \\ 1 & \text{if } V_k(x+1) < V_k(x) \\ \text{any } u \in [0,1] & \text{if } V_k(x+1) = V_k(x). \end{cases}$$

Furthermore, since

$$\left\{ V_k(x) + \min_{u\in[0,1]} u\,(V_k(x+1) - V_k(x)) \right\} = \min\{V_k(x), V_{k+1}(x)\},$$

we have

$$V_{k+1}(x) = cx + \mu\, I_{x=0} + \frac{\alpha\mu}{\lambda+\mu} V_k((x-1)^+) + \frac{\alpha\lambda}{\lambda+\mu} \min\{V_k(x), V_{k+1}(x)\}.$$

The preceding relation and the expression for the optimal control μ_k^* *can be further analyzed to determine the optimal policy. In particular, by induction on* k*, it can be seen that* $V_k(x+1) - V_k(x)$ *is nondecreasing in* x *and* $V_k(1) - V_k(0) \geq -\frac{\lambda+\mu}{\alpha}$ *(see [17] Section 9.3).*

5.1.2 Infinite Horizon Problems

Infinite horizon problems with stationary description have some special properties. In particular, optimal policies for an infinite horizon problems are typically stationary, i.e., an optimal policy is specified by a function $\mu : x_k \mapsto u_k$, where $u_k \in U(x_k)$, and the function μ does not depend on time.

In the following sections, we study three types of infinite horizon problems:

(1) *Discounted Cost Problems*, which have a discount factor α per stage and uniformly bounded absolute cost value $g(x, u, \xi)$.

(2) *Stochastic Shortest Path Problems*, which have a special cost-free termination state. In fact, these problems have a finite horizon but the length of the horizon is random depending on the policy that is applied.

(3) *Average Cost Problems*, which cost is defined as the limit as $N \to \infty$ of the averaged cost over N stages.

For an infinite horizon problem, a policy π is an infinite sequence $\pi = (\mu_0, \mu_1 \ldots)$. Furthermore, the optimality principle is captured by the *Bellman's equation*

$$V^*(x) = \min_{u \in U(x)} \mathop{\mathrm{E}}_{\xi} \{g(x, u, \xi) + V^*(F(x, u, \xi))\}. \tag{5.2}$$

This equation takes a special form for each of the above mentioned infinite horizon problems.

DP Mapping

We introduce the DP mapping $T : V \mapsto TJ$ that to each function $V : S \to \mathbb{R}$ assigns another function, denoted by TV, i.e., $TV : S \to \mathbb{R}$. The function TV is defined by

$$(TV)(x) = \min_{u \in U(x)} \mathop{\mathrm{E}}_{\xi} \{g(x, u, \xi) + \alpha V(F(x, u, \xi))\} \qquad \text{for all } x \in S.$$

Any function satisfying

$$TV = V$$

is *a fixed point of the mapping* T. Note that the function TV is defined as the function resulting from applying one DP iteration (5.1) to the function V. Furthermore, if $V = V^*$, then the preceding relation and Bellman's equation (5.2) imply that $TV^* = V^*$. We establish these relations formally in the subsequent development.

A stationary policy (for an infinite horizon problem) is a policy π such that $\pi = (\mu, \mu, \ldots)$. For this reason we will identify a stationary policy with the map μ defining the policy.

Given a stationary policy μ, we define a mapping $T_\mu V$ that to each function V assigns a function, denoted by $T_\mu V$, given by

$$(T_\mu V)(x) = \mathop{\mathrm{E}}_{\xi} \{g(x, \mu(x), \xi) + \alpha V(F(x, \mu(x), \xi))\} \qquad \text{for all } x \in S.$$

The k successive applications of the mapping T is denoted by T^k, i.e.,

$$T^k V = T(T^{k-1} V), \qquad \text{with } \; T^0 V = V.$$

Similarly, given a stationary policy μ, the k successive applications of the mapping T_μ is denoted by T_μ^k, i.e.,

$$T_\mu^k V = T_\mu(T_\mu^{k-1})V, \qquad \text{with } \; T_\mu^0 V = V.$$

The maps T and T_μ have some interesting property, namely, they preserve the "ordering of functions" in a sense, as seen in the following lemma.

Lemma 16 *Let functions $V : S \to \mathbb{R}$ and $\tilde{V} : S \to \mathbb{R}$ be such that*

$$V(x) \leq \tilde{V}(x) \qquad \textit{for all } x \in S.$$

Then, for any $k \geq 0$, we have

$$(T^k V)(x) \leq (T^k \tilde{V})(x) \qquad \textit{for all } x \in S,$$

$$(T_\mu^k V)(x) \leq (T_\mu^k \tilde{V})(x) \qquad \textit{for all } x \in S,$$

where μ is any stationary policy.

The proof of this lemma is straightforward from the definitions of the maps T and T_μ.

5.2 Discounted Cost Problem

Here, we focus on the DP problem on infinite horizon with discounted cost [by a factor $\alpha \in (0,1)$] i.e. determining the optimal cost function

$$V^*(x) = \min_{\pi \in \Pi} V_\pi(x), \qquad V_\pi(x) = \mathrm{E}\left\{ \sum_{k=0}^{\infty} \alpha^k g(x_k, \mu_k(x_k), \xi_k) \,\middle|\, x_0 = x \right\}$$

and the optimal policy π^* attaining the optimal cost. In order to have the infinite sum well defined, we assume that the cost $g(x,u,\xi)$ is bounded for all states $x \in S$, all controls u and uncertainties ξ, i.e.,

$$|g(x,u,\xi)| \le G \qquad \text{for all } x,u,\xi.$$

This roundedness assumption is in force throughout this section.

5.2.1 Basic Results

We here establish some basic properties of the DP mapping T, leading to Bellman's equation and a characterization of the optimal stationary policies.

We first show that if the DP algorithm starts with some bounded function V and the DP iterations are executed indefinitely (the size N of horizon goes to infinity), then the algorithm converges to the optimal cost function V^*. By representing each DP iteration with a DP mapping T operation, we summarize the preceding statement in the following.

Theorem 53 *Let $V : S \to \mathbb{R}$ be any bounded function. Then,*

$$\lim_{N\to\infty} (T^N V)(x) = V^*(x) \qquad \textit{for all } x \in S.$$

Proof. Let $\pi = (\mu_0, \mu_1, \ldots)$ be any policy. The cost function of the policy is

$$V_\pi(x) = \mathrm{E}\left\{ \sum_{k=0}^{\infty} \alpha^k g(x_k, \mu_k(x_k), \xi_k) \,\middle|\, x_0 = x \right\} \qquad \text{for all } x \in S.$$

By truncating the horizon at an arbitrary time N, we can write

$$V_\pi(x) = \mathrm{E}\left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), \xi_k) \,\middle|\, x_0 = x \right\} + \mathrm{E}\left\{ \sum_{k=N}^{\infty} \alpha^k g(x_k, \mu_k(x_k), \xi_k) \,\middle|\, x_0 = x \right\},$$

implying that

$$\mathrm{E}\left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), \xi_k) \,\middle|\, x_0 = x \right\} = V_\pi(x) - \mathrm{E}\left\{ \sum_{k=N}^{\infty} \alpha^k g(x_k, \mu_k(x_k), \xi_k) \,\middle|\, x_0 = x \right\}.$$

Since the cost per stage is bounded, we have

$$\left| \mathrm{E}\left\{ \sum_{k=N}^{\infty} \alpha^k g(x_k, \mu_k(x_k), \xi_k) \,\middle|\, x_0 = x \right\} \right| \le G \sum_{k=N}^{\infty} \alpha^k = \frac{G\alpha^N}{1-\alpha}.$$

Combining the preceding two relations, we see that

$$V_\pi(x) - \frac{G\alpha^N}{1-\alpha} \le \mathrm{E}\left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), \xi_k) \,\Big|\, x_0 = x \right\} \le V_\pi(x) + \frac{G\alpha^N}{1-\alpha}.$$

By adding the value $\alpha^N V(x_N)$, and noting that by the boundedness of V we have $V(x_N) \le \max_{x\in S} |V(x)|$, we have

$$V_\pi(x) - \frac{G\alpha^N}{1-\alpha} - \alpha^N \max_{x\in S} |V(x)| \le \mathrm{E}\left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), \xi_k) + \alpha^N V(x_N) \,\Big|\, x_0 = x \right\}, \quad (5.3)$$

$$\mathrm{E}\left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), \xi_k) + \alpha^N V(x_N) \,\Big|\, x_0 = x \right\} \le V_\pi(x) + \frac{G\alpha^N}{1-\alpha} + \alpha^N \max_{x\in S} |V(x)|. \quad (5.4)$$

By taking the minimum over all policies, and noting that

$$\min_{\pi\in\Pi} \mathrm{E}\left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), \xi_k) + \alpha^N V(x_N) \,\Big|\, x_0 = x \right\} = (T^N V)(x),$$

we obtain

$$V^*(x) - \frac{G\alpha^N}{1-\alpha} - \alpha^N \max_{x\in S} |V(x)| \le (T^N V)(x) \le V^*(x) + \frac{G\alpha^N}{1-\alpha} + \alpha^N \max_{x\in S} |V(x)|. \quad (5.5)$$

By letting $N \to \infty$, we see that

$$V^*(x) \le \liminf_{N\to\infty} (T^N V)(x) \le \limsup_{N\to\infty} (T^N V)(x) \le V^*(x),$$

thus showing that $\lim_{N\to\infty} (T^N V)(x) = V^*(x)$. ■

Observe that Theorem 53 indicates that the DP algorithm can be used to approximate the optimal cost function $V^*(x)$, while the proof of the theorem indicates the error bounds. We discuss this later in this section.

We have an analogous result to that of Theorem 53 for the stationary policy value function $V_\mu(x)$.

Theorem 54 *Let $V : S \to \mathbb{R}$ be any bounded function and let μ be a stationary policy. Then,*

$$\lim_{N\to\infty} (T_\mu^N V)(x) = V_\mu(x) \qquad \textit{for all } x \in S.$$

Proof. Since the policy is stationary, we have $\mu_k(x) = \mu(x)$ for all $x \in S$. By using this in the relations (5.3) and (5.4), we obtain

$$V_\pi(x) - \frac{G\alpha^N}{1-\alpha} - \alpha^N \max_{x\in S} |V(x)| \le \mathrm{E}\left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu(x_k), \xi_k) + \alpha^N V(x_N) \,\Big|\, x_0 = x \right\},$$

$$\mathrm{E}\left\{\sum_{k=0}^{N-1}\alpha^k g(x_k,\mu(x_k),\xi_k)+\alpha^N V(x_N)\,\Big|\,x_0=x\right\}\le V_\pi(x)+\frac{G\alpha^N}{1-\alpha}+\alpha^N\max_{x\in S}|V(x)|.$$

Noting that

$$\mathrm{E}\left\{\sum_{k=0}^{N-1}\alpha^k g(x_k,\mu(x_k),\xi_k)+\alpha^N V(x_N)\,\Big|\,x_0=x\right\}=(T_\mu^N V)(x),$$

we obtain

$$V_\mu(x)-\frac{G\alpha^N}{1-\alpha}-\alpha^N\max_{x\in S}|V(x)|\le(T_\mu^N V)(x)\le V_\mu(x)+\frac{G\alpha^N}{1-\alpha}+\alpha^N\max_{x\in S}|V(x)|. \quad (5.6)$$

By letting $N\to\infty$, we see that

$$V_\mu(x)\le\liminf_{N\to\infty}(T_\mu^N V)(x)\le\limsup_{N\to\infty}(T_\mu^N V)(x)\le V_\mu(x),$$

thus showing that $\lim_{N\to\infty}(T_\mu^N V)(x)=V_\mu(x)$. ■

We next establish Bellman's equation.

Theorem 55 *The optimal cost function V^* satisfies Bellman's equation*

$$V^*(x)=\min_{u\in U(x)}\mathrm{E}_\xi\{g(x,u,\xi)+\alpha V^*(F(x,u,\xi))\}\qquad\text{for all } x\in S,$$

or equivalently, the optimal cost function V^* is a fixed point of the DP mapping

$$V^*=TV^*.$$

Furthermore, the optimal cost function V^* is the unique function that satisfies Bellman's equation (unique fixed point of T) within the class of bounded functions.

Proof. We start from Eq. (5.5), i.e., for all $x\in S$,

$$V^*(x)-\frac{G\alpha^N}{1-\alpha}-\alpha^N\max_{x\in S}|V(x)|\le(T^N V)(x)\le V^*(x)+\frac{G\alpha^N}{1-\alpha}+\alpha^N\max_{x\in S}|V(x)|.$$

By choosing V to be the zero function, i.e., $V(x)=0$ for all $x\in S$, from the preceding relation we obtain

$$V^*(x)-\frac{G\alpha^N}{1-\alpha}\le(T^N V)(x)\le V^*(x)+\frac{G\alpha^N}{1-\alpha}.$$

By using the monotonicity property of the mapping T of Lemma 16, we obtain

$$T\left(V^*(x)-\frac{G\alpha^N}{1-\alpha}\right)\le(T^{N+1}V)(x)\le T\left(V^*(x)+\frac{G\alpha^N}{1-\alpha}\right).$$

Since

$$T\left(V^*(x)-\frac{G\alpha^N}{1-\alpha}\right)=TV^*(x)-\frac{G\alpha^{N+1}}{1-\alpha},$$

$$T\left(V^*(x)+\frac{G\alpha^N}{1-\alpha}\right)=TV^*(x)+\frac{G\alpha^{N+1}}{1-\alpha},$$

we further obtain

$$TV^*(x)-\frac{G\alpha^{N+1}}{1-\alpha}\leq (T^{N+1}V)(x)\leq TV^*(x)+\frac{G\alpha^{N+1}}{1-\alpha}.$$

By letting $N\to\infty$ and using Theorem 53, we have

$$TV^*(x)\leq V^*(x)\leq TV^*(x),$$

thus showing that $V^*=TV^*$.

To establish the uniqueness, suppose that V is a bounded function and satisfies $V=TV$. Then, by Theorem 53 we have $V^*=\lim_{N\to\infty}T^NV$, implying that $V^*=V$. ■

We next show that the policy value function V_μ is a fixed point of the mapping T_μ.

Theorem 56 *Let μ be a stationary policy. The policy cost function V_μ satisfies the following relation*

$$V_\mu(x)=\mathop{\mathrm{E}}_{\xi}\{g(x,\mu(x),\xi)+\alpha V_\mu(F(x,\mu(x),\xi))\}\qquad \textit{for all } x\in S,$$

or equivalently, the policy cost function V_μ is a fixed point of the mapping T_μ.

$$V_\mu=T_\mu V_\mu.$$

Furthermore, the policy cost function V_μ is the unique function that satisfies the given relation within the class of bounded functions.

Proof. We use start from Eq. (5.6), i.e., for all $x\in S$,

$$V_\mu(x)-\frac{G\alpha^N}{1-\alpha}-\alpha^N\max_{x\in S}|V(x)|\leq (T_\mu^NV)(x)\leq V_\mu(x)+\frac{G\alpha^N}{1-\alpha}+\alpha^N\max_{x\in S}|V(x)|.$$

By choosing V to be the zero function, i.e., $V(x)=0$ for all $x\in S$, from the preceding relation we obtain

$$V_\mu(x)-\frac{G\alpha^N}{1-\alpha}\leq (T_\mu^NV)(x)\leq V_\mu(x)+\frac{G\alpha^N}{1-\alpha}.$$

By the monotonicity property of the mapping T_μ of Lemma 16, we have

$$T_\mu\left(V_\mu(x)-\frac{G\alpha^N}{1-\alpha}\right)\leq (T_\mu^{N+1}V)(x)\leq T\left(V_\mu(x)+\frac{G\alpha^N}{1-\alpha}\right).$$

Using the relations

$$T_\mu\left(V_\mu(x)-\frac{G\alpha^N}{1-\alpha}\right)=T_\mu V_\mu(x)-\frac{G\alpha^{N+1}}{1-\alpha},$$
$$T_\mu\left(V_\mu(x)+\frac{G\alpha^N}{1-\alpha}\right)=T_\mu V_\mu(x)+\frac{G\alpha^{N+1}}{1-\alpha},$$

we obtain

$$T_\mu V_\mu(x) - \frac{G\alpha^{N+1}}{1-\alpha} \le (T_\mu^{N+1}V)(x) \le T_\mu V_\mu(x) + \frac{G\alpha^{N+1}}{1-\alpha}.$$

By letting $N \to \infty$ and using Theorem 54, we further obtain

$$T_\mu V_\mu(x) \le V_\mu(x) \le T_\mu V_\mu(x),$$

thus showing that $V_\mu = T_\mu V_\mu$.

To show the uniqueness, suppose that V is a bounded function satisfying $V = T_\mu V$. By Theorem 54 we have $V_\mu = \lim_{N\to\infty} T_\mu^N V$, implying that $V_\mu = V$. ■

As a consequence of Theorems 55 and 56, we have the following characterization of an optimal stationary policy.

Theorem 57 *A stationary policy μ is optimal if and only if μ attains the minimum in Bellman's equation, i.e.,*

$$\mu(x) \in \operatorname*{Argmin}_{u\in U(x)} \mathop{\mathrm{E}}_{\xi}\{g(x,u,\xi) + \alpha V^*(F(x,u,\xi))\} \qquad \textit{for all } x \in S,$$

or equivalently

$$TV^* = T_\mu V^*.$$

The next two theorems show that the DP mapping T and any stationary policy mapping T_μ are contractions with respect to the function sup-norm, i.e.,

$$\|V\| = \sup_{x\in S} |V(x)|.$$

Theorem 58 *Let $V : S \to \mathbb{R}$ and $\tilde{V} : S \to \mathbb{R}$ be bounded functions. Then, the DP map is a contraction with factor α, i.e.,*

$$\|TV - T\tilde{V}\| \le \alpha\|V - \tilde{V}\|,$$

where $\|V\|$ is the function sup-norm. Furthermore, for any stationary policy μ, the policy map T_μ is also a contraction with factor α, i.e.,

$$\|T_\mu V - T_\mu \tilde{V}\| \le \alpha\|V - \tilde{V}\|,$$

Proof. We have

$$V(x) = V(x) - \tilde{V}(x) + \tilde{V}(x),$$

implying that

$$V(x) \le |V(x) - \tilde{V}(x)| + \tilde{V}(x), \qquad V(x) \ge -|V(x) - \tilde{V}(x)| + \tilde{V}(x).$$

By letting $s = \max_{x\in S} |V(x) - \tilde{V}(x)| = \|V - \tilde{V}\|$, we can write

$$\tilde{V}(x) - s \le V(x) \le \tilde{V}(x) + s \qquad \text{for all } x \in S.$$

By applying the DP mapping and using the monotonicity property of the DP mapping of Lemma 16, we obtain

$$T\left(\tilde{V}(x) - s \leq TV(x)\right) \leq T\left(\tilde{V}(x) + s\right) \qquad \text{for all } x \in S.$$

Since

$$T\left(\tilde{V}(x) \pm s\right) = T\tilde{V}(x) \pm \alpha s,$$

it follows that

$$T\tilde{V}(x) - \alpha s \leq TV(x) \leq T\tilde{V}(x) + \alpha s \qquad \text{for all } x \in S.$$

The preceding relation implies

$$-\alpha s \leq T\tilde{V}(x) - TV(x) \leq \alpha s \qquad \text{for all } x \in S,$$

thus showing that

$$\max_{x \in S} |T\tilde{V}(x) - TV(x)| \leq s = \max_{x \in S} |V(x) - \tilde{V}(x)|.$$

The same proof works with T_μ instead of T, showing that $\|T_\mu V - T_\mu \tilde{V}\| \leq \alpha \|V - \tilde{V}\|$. ■

As an immediate consequence of Theorem 58, we have for bounded functions V and $\tilde{V}$, and for any $k \geq 1$,

$$\|T^k V - T^k \tilde{V}\| \leq \alpha^k \|V - \tilde{V}\|,$$
$$\|T_\mu^k V - T_\mu^k \tilde{V}\| \leq \alpha^k \|V - \tilde{V}\|.$$

5.2.2 Value Iteration

Based on Theorem 55 we have *value iteration algorithm*, for which by Theorem 58 we also have an error bound, i.e., the term $|T^k V(x) - V^*(x)|$ is at most of the order of α^k.

The value iteration algorithm starts with an arbitrary (bounded) function V, and generates a sequence of functions $T^k V$ by successive applications of the DP algorithm. The function sequence $T^k V$ converges to the optimal cost V^* in the limit as $k \to \infty$, as seen from Theorem 55.

At each iteration k, an upper bound on the error of between $T^k V$ and V^* is available from Theorem 58. The following theorem can also be used for computing the estimates of the errors as the algorithm progresses.

Theorem 59 *For any bounded V, any state $x \in S$, and any $k \geq 0$ we have*

$$T^k V(x) + m_k \leq T^{k+1} V(x) + m_{k+1} \leq V^*(x) \leq T^{k+1} V(x) + M_{k+1} \leq T^k V(x) + M_k,$$

where

$$m_k = \frac{\alpha}{1-\alpha} \min_{x \in S} \{T^k V(x) - T^{k-1} V(x)\},$$
$$M_k = \frac{\alpha}{1-\alpha} \max_{x \in S} \{T^k V(x) - T^{k-1} V(x)\}.$$

The proof of the preceding theorem rests on the monotonicity property of the DP mapping T and Bellman's equation [cf. Lemma 16) and Theorem 55)]. The proof can be found in [7] Section 1.3 (see there Proposition 1.3.1).

In practice, the value iteration algorithm is terminated when the difference $M_k - m_k$ becomes small enough (lower than a prescribed error level). A "median" $\hat{V}_k$ can be used as the final estimate of V^*, i.e.,

$$\hat{V}_k(x) = T^k V(x) + \frac{1}{2}(m_k + M_k) \qquad \text{for all } x \in S,$$

or the "average" $\bar{V}_k$ when the set S of states is finite,

$$\bar{V}_k = T^k V(x) + \frac{\alpha}{|S|(1-\alpha)} \sum_{y \in S} \left(T^k V(y) - T^{k-1} V(y) \right) \qquad \text{for all } x \in S.$$

Given an approximation V_k of the optimal cost function, such as $T^k V$, the "median" $\hat{V}_k$, or the "average" $\bar{V}_k$, we can determine a suboptimal stationary policy and a bound on its error. In particular, given an approximation V_k, we can compute TV_k and determine the policy μ that achieves the minimum in the expression TV_k, i.e., $TV_k = T_m V_k$, or explicitly

$$\mu(x) \in \operatorname*{Argmin}_{u \in U(x)} \mathop{\mathrm{E}}_{\xi} \{ g(x,u,\xi) + \alpha V_k(F(x,u,\xi)) \}.$$

We have the following bound on the suboptimality of the policy μ:

$$\max_{x \in S} |V_\mu(x) - V^*(x)| \leq \frac{\alpha}{1-\alpha} \left(\max_{x \in S} \{ TV_k(x) - V_k(x) \} - \min_{x \in S} \{ TV_k(x) - V_k(x) \} \right).$$

The value iteration algorithm can be slow. This can be related to the second eigenvalue of a Markov transition matrix associated with an optimal solution being close to 1 (see the discussion in [7] Section 1.3, page 28). Furthermore, when the number of states is large, the method may become impractical due to high computational load.

5.2.3 Policy Iteration

While value iteration algorithm generates a sequence of functions approximating the optimal cost function V^*, the policy iteration algorithm generates a sequence of stationary policies approximating the optimal stationary policy.

The idea is rather simple: given a policy μ and its cost function V_μ, determine a policy $\tilde{\mu}$ achieving the minimum in TV_μ, i.e., $\tilde{\mu}$ such that $T_{\tilde{\mu}} V_\mu = TV_\mu$. The hope is that cost $V_{\tilde{\mu}}$ of the new policy $\tilde{\mu}$ is better than the cost V_μ of the policy we started with. In fact, this is the case, as seen in the following theorem that justifies the policy iteration algorithm.

Theorem 60 *Let μ be a stationary policy. Consider a policy $\tilde{\mu}$ satisfying $T_{\tilde{\mu}} V_\mu = TV_\mu$, i.e.,*

$$\tilde{\mu}(x) \in \min_{u \in U(x)} \mathop{\mathrm{E}}_{\xi} \{ g(x,u,\xi) + \alpha V_\mu(F(x,u,\xi)) \} \qquad \textit{for all } x \in S.$$

Then, the policy cost $V_{\tilde{\mu}}$ is dominated by the policy cost V_μ, i.e.,

$$V_{\tilde{\mu}}(x) \leq V_\mu(x) \qquad \text{for all } x \in S.$$

In addition, if μ is not optimal stationary policy, then

$$V_{\tilde{\mu}}(x) < V_\mu(x) \qquad \text{for at least one } x \in S.$$

Proof. By the definition of the policy $\tilde{\mu}$, we have for all $x \in S$,

$$\begin{aligned} T_{\tilde{\mu}} V_\mu(x) &= \mathop{\mathrm{E}}_{\xi}\{g(x, \tilde{\mu}(x), \xi) + \alpha V_\mu(F(x, \tilde{\mu}(x), \xi))\} \\ &= \min_{u \in U(x)} \mathop{\mathrm{E}}_{\xi}\{g(x, u, \xi) + \alpha V_\mu(F(x, u, \xi))\} \\ &\leq \mathop{\mathrm{E}}_{\xi}\{g(x, \mu(x), \xi) + \alpha V_\mu(F(x, \mu(x), \xi))\} \\ &= V_\mu(x), \end{aligned}$$

where the last equality follows from the definition of the policy cost V_μ, i.e., $T_\mu V_\mu = V_\mu$.

By applying the mapping $T_{\tilde{\mu}}$ and using the monotonicity property of $T_{\tilde{\mu}}$, we obtain for all $x \in S$,

$$T_{\tilde{\mu}}^2 V_\mu(x) \leq T_{\tilde{\mu}} V_\mu(x) \leq V_\mu(x).$$

By repeating this process indefinitely, we see that

$$T_{\tilde{\mu}}^N V_\mu(x) \leq V_\mu(x) \qquad \text{for all } x \in S \text{ and any } N \geq 1.$$

Taking the limit as $N \to \infty$ and using Theorem 54, we conclude that

$$V_{\tilde{\mu}}(x) = \lim_{N \to \infty} T_{\tilde{\mu}}^N V_\mu(x) \leq V_\mu(x) \qquad \text{for all } x \in S,$$

thus showing that the policy cost $V_{\tilde{\mu}}$ is dominated by the policy cost V_μ.

Assume that μ is not optimal. To arrive at a contradiction, suppose there is no state x for which the inequality $V_{\tilde{\mu}}(x) < V_\mu(x)$ holds. Then, we must have $V_{\tilde{\mu}}(x) = V_\mu(x)$ for all states x, or equivalently $V_{\tilde{\mu}} = V_\mu$. Consider the function TV_μ. By the definition of $\tilde{\mu}$, we have $TV_\mu = T_{\tilde{\mu}} V_\mu$, implying by $V_\mu = V_{\tilde{\mu}}$ that

$$TV_\mu = T_{\tilde{\mu}} V_{\tilde{\mu}}.$$

By Theorem 56, we have $T_{\tilde{\mu}} V_{\tilde{\mu}} = V_{\tilde{\mu}}$, and since $V_{\tilde{\mu}} = V_\mu$, it follows that

$$TV_\mu = V_\mu.$$

Hence V_μ is a fixed point of T. However, by Theorem 56, the optimal cost V^* is the unique fixed point of the DP mapping T, implying that $V_\mu = V^*$. By Theorem 57 it follows that μ is an optimal policy - a contradiction. Therefore, we must have

$$V_{\tilde{\mu}}(x) < V_\mu(x) \qquad \text{for some } x \in S.$$

■

The preceding theorem justifies the policy iteration method that generates a sequence $\{\mu^k\}$ of policies by the following rule:

$$T_{\mu^{k+1}} V_{\mu^k} = T V_{\mu^k},$$

starting with some initial policy μ^0. In particular, *the policy iteration method* proceeds as follows.

Policy Iteration Start with an initial policy μ^0.

1. At iteration k, having the current policy μ^k, determine the policy value function V_{μ^k}.

2. Generate a new policy μ^{k+1} satisfying $T_{\mu^{k+1}} V_{\mu^k} = T V_{\mu^k}$, or more explicitly

$$\mu^{k+1}(x) \in \operatorname*{Argmin}_{u \in U(x)} \mathop{\mathrm{E}}_{\xi} \{ g(x,u,\xi) + \alpha V_{\mu^k}(F(x,u,\xi)) \} \qquad \text{for all } x \in S.$$

   If $V_{\mu^k} = T V_{\mu^k}$ terminate, the policy μ^k is optimal. Otherwise, go to the policy evaluation step 1.

First note that if the set of all policies is finite, the policy iteration method will terminate with an optimal policy in a finite number of steps. However, note that both steps of the algorithm can be computationally intensive when the number of states is large. In particular, to evaluate a policy μ^k at step 1, we need to solve the functional equation $T_\mu V_\mu = V_m$, which is a system of linear equations with one equation per state x. Similarly, in step 2, we have to evaluate the function $T V_{\mu^k}$ and then find the optimal choice $\mu^k(x)$ for each $x \in S$. In both steps, the size of the corresponding problem we need to solve is the same as the size of the state set S (the cardinality of S). Thus, the applications of policy iteration method are limited in the same way as the value iteration method, by the number of the states. This is known as *the curse of dimensionality*, a term introduced by Richard Bellman.

5.3 Stochastic Shortest Path Problem

In Section 4.3 of Chapter 4, we have considered a deterministic shortest path problem for all origin-single destination case. Here, we consider a stochastic version of that problem for a graph with nodes $1, \ldots, n$ and a single terminal (destination) node t. At each node i, we select a control u which influences the probability distribution $p_{ij}(u)$ on moving from node i to any other node j. The cost of each link is also random. The overall goal is to find the shortest paths from each node to the terminal node, where the length of the path is measured by the total expected cost incurred along the path.

We formulate the stochastic shortest path as a DP problem as follows:

1. The set S of system states is $S = \{1, \ldots, n, t\}$ with t being a special terminal state.

2. For each state $i \in S$, a set $U(i)$ specifies the controls u available at that state. The set $U(i)$ is finite for all i.

3. At each state i, every control u specifies a probability distribution $p_{ij}(u)$ of moving from state i to state $j \in S$, i.e.,

$$p_{ij}(u) = \text{Prob}\{x_{k+1} = j \mid x_k = i, u_k = u\} \qquad \text{for all } i \in S.$$

The terminal state t is absorbing,

$$p_{tt}(u) = 1 \qquad \text{for all } u \in U(t).$$

4. A cost $g(i, u, j)$ is incurred when moving from state i to state j under the control u. The terminal state is cost free,

$$g(t, u, j) = 0 \qquad \text{for all } u \in U(t) \text{ and all } j \in S.$$

5. The objective is to minimize the total expected policy cost $V_\pi(i)$ over all policies $\pi = (\mu_0, \mu_1, \ldots)$, where the policy cost $V_\pi(i)$ starting from state i is given by:

$$V_\pi(i) = \limsup_{N\to\infty} \text{E}\left\{ \sum_{k=0}^{N-1} g(x_k, \mu_k(x_k), x_{k+1}) \,\middle|\, x_0 = i \right\}.$$

Due to the special structure of the system evolution, we can associate a deterministic cost $\bar{g}(i, u)$ for choosing a control u at state i, by letting

$$\bar{g}(i, u) = \text{E}[g(i, u, j)|(i, u)] = \sum_{j\in S} p_{ij}(u)\, g(i, u, j) \qquad \text{for all } i \in S \text{ and all } u \in U(i).$$

Note that for the terminal state, we have

$$\bar{g}(t, u) = \text{E}[g(t, u, j)|(t, u)] = \sum_{j\in S} p_{tj}(u)\, g(i, u, j) = p_{tt} g(t, u, t) = 0 \qquad \text{for all } u \in U(t).$$

Since the terminal state is absorbing and cost-free, the cost of any policy starting from state t is zero, i.e., $V_\pi(t) = 0$ for any policy π. Thus, we can ignore the policy cost values for state t. The problem is now to determine the policy π^* such that for all $i = 1, \ldots, n$,

$$V_{\pi^*}(i) = \min_{\pi\in\Pi} V_\pi(i),$$

where $\pi = (\mu_0, \mu_1, \ldots)$ and

$$\begin{aligned} V_\pi(i) &= \limsup_{N\to\infty} \left\{ \sum_{k=0}^{N-1} \sum_{j\in S} p_{i_k j}(\mu_k(i_k))\, g(i_k, \mu_k(i_k), j) \,\middle|\, i_0 = i \right\} \\ &= \limsup_{N\to\infty} \left\{ \bar{g}(i, \mu_0(i)) + \sum_{k=1}^{N-1} \bar{g}(i_k, \mu_k(i_k)) \right\}. \end{aligned}$$

Note that there is no discount factor ($\alpha = 1$).

As an example of the use of stochastic shortest path model, consider the following.

Example 35 *(G/G/1 First come first serve queue, see [16], Example 2 in Chapter 4.) Suppose that for an FCFS G/G/1 queue with maximum length n, we want to find the expected sum of the times that all customers spend in the system, starting from some time with i customers in the system and ending at the time when the system is empty.*

We can model the system as a stochastic shortest path problem where a state i corresponds to i customers being in the queue. Thus, the states are 0 and $1, \ldots, n$, with 0 being the terminal state. The transition probabilities p_{ij} are given in terms of the customer arrival rate and the service rate, with 0 modeled as an absorbing state. We assign reward $g(i) = i$ when the system is in state i. The expected sum of the times that all customers spend in the system from some time with i customers in the system is equal to the expected reward collected from state i.

We can view a function $V : \{1, \ldots, n\} \to \mathbb{R}$ as a vector in $\mathbb{R}^n$. With this view, the DP mapping T is from $\mathbb{R}^n$ to $\mathbb{R}^n$, with TV defined by

$$TV(i) = \min_{u \in U(i)} \left\{ \bar{g}(i,u) + \sum_{j=1}^{n} p_{ij}(u)\, V(j) \right\} \qquad \text{for all } i = 1, \ldots, n. \tag{5.7}$$

Similarly, a (stationary) policy mapping T_μ is from $\mathbb{R}^n$ to $\mathbb{R}^n$, with $T_\mu V$ defined by

$$T_\mu V(i) = \bar{g}(i, \mu(i)) + \sum_{j=1}^{n} p_{ij}(\mu(i))\, V(j) \qquad \text{for all } i = 1, \ldots, n. \tag{5.8}$$

From now on, we focus on the stationary policies. To further simplify the notation, we introduce the transition matrix P_μ corresponding to a stationary policy μ,

$$P_\mu = \begin{bmatrix} p_{11}(\mu(1)) & \cdots & p_{1n}(\mu(1)) \\ \vdots & \ddots & \vdots \\ p_{n1}(\mu(n)) & \ldots & p_{nn}(\mu(n)) \end{bmatrix}.$$

Note that the matrix P_μ is not stochastic, since for the states i with $p_{it}(\mu(i)) > 0$ we have

$$\sum_{j=1}^{n} p_{ij}(\mu(i)) = 1 - p_{it}(\mu(i)) < 1.$$

Also, we introduce the cost vector $\bar{g}_\mu$ associated with a stationary policy μ,

$$g_\mu = \begin{bmatrix} \bar{g}(1, (\mu(1)) \\ \vdots \\ \bar{g}(n, (\mu(n)) \end{bmatrix}.$$

With this notation, the vector $T_\mu V$ resulting from the mapping T_μ is given by

$$T_\mu V = \bar{g}_\mu + P_\mu V. \tag{5.9}$$

In the absence of additional assumptions, the optimal cost value for the stochastic shortest path problem may not be finite and/or an optimal stationary policy may not exist. To see this, consider the following example.

Example 36 *(Blackmail Problem, see [34] Chapter 25 Section 5, or [7] Section 2.3.) A blackmailer wants to optimize his policy. If he makes a demand of u millions of euros, where $0 < u \leq 1$, he receives an immediate reward u and terminates with probability u^2.*

This situation can be modeled as a DP with a single non-termination state 1 with the control $u \in (0, 1]$. There is no optimal stationary policy that maximizes the black-mailer's profit. To see this note that such a policy should satisfy

$$V_\mu(1) = u + (1 - u^2)V_\mu,$$

implying that

$$V_\mu = \frac{1}{u} \qquad \text{for } u \in (0, 1].$$

By taking $u \to 0$, we see that the optimal value at state 1 is $V^*(1) = +\infty$, but no stationary policy can achieve this value.

It can be seen that the optimal policy $\pi^* = (\mu_0^*, \mu_1^*, \ldots)$ is given by

$$\mu_k^*(1) = \frac{c}{k+1} \qquad \text{for all } k \geq 0,$$

where c is a scalar with $0 < c < \frac{1}{2}$.

To avoid the problem of having the optimal cost infinite at some states, we introduce the notion of proper policies. *A stationary policy is proper* when, under the policy, there is a positive probability that the terminal state t will be reached after at most n stages for any initial state. In other words, the probability that the state t will not be reached at stage n is not 1 for any initial stage, or formally

$$\max_{1 \leq i \leq n} \text{Prob}\{x_n \neq t \mid x_0 = i, \mu\} < 1.$$

A stationary policy is improper when it is not proper.

Throughout the rest of this section, for the stochastic shortest path problem, we assume that *there is at least one proper stationary policy*. We also assume that *for any improper stationary policy μ, there is a state i such that $V_\mu(i) = -\infty$.*

For a deterministic shortest path problem, the former assumption simply states that there is a directed path from each node i to the terminal node t. The later assumption ensures that there are no negative cost cycles (see Section 4.3).

5.3.1 Basic Relations

Here, we provide basic insights into the properties of the DP mapping T and the stationary policy mapping T_μ for the stochastic shortest path problem [cf. Eqs. (5.7) and (5.8)]. In particular, we provide Bellman's equation and optimality conditions amongst others. The proofs of these results can be found, for example, in [7] Section 2.2.

Theorem 61 *For the stochastic shortest path problem, we have*

(a) Let $V \in \mathbb{R}^n$ be arbitrary. For the DP mapping T of Eq. (5.7), we have

$$\lim_{N\to\infty}(T^N V) = V^*,$$

where $V^* \in \mathbb{R}^n$ is the optimal cost vector with components $V^*(i)$ for $i = 1, \ldots, n$.

(b) The optimal cost vector V^* satisfies Bellman's equation

$$V^*(i) = \min_{u\in U(i)} \left\{ \bar{g}(i,u) + \sum_{j=1}^{n} p_{ij}(u)\, V^*(j) \right\} \qquad \text{for all } i = 1, \ldots, n,$$

or equivalently

$$TV^* = V^*.$$

Moreover, V^* is the unique fixed point of the DP mapping T.

We have an analogous result for the policy mapping T_μ.

Theorem 62 *For the stochastic shortest path problem, we have*

(a) Let $V \in \mathbb{R}^n$ be arbitrary and let μ be a proper stationary policy. For the policy mapping T_μ of Eq. (5.8), we have

$$\lim_{N\to\infty}(T_\mu^N V) = V_\mu,$$

where $V_\mu \in \mathbb{R}^n$ is the policy cost vector with components $V_\mu(i)$ for $i = 1, \ldots, n$.

(b) The policy cost vector V_μ satisfies the following relation

$$V_\mu(i) = \left[\bar{g}(i,\mu(i)) + \sum_{j=1}^{n} p_{ij}(\mu(i))\, V^*(j) \right] \qquad \text{for all } i = 1, \ldots, n,$$

or equivalently

$$T_\mu V_\mu = V_\mu.$$

Moreover, V_μ is the unique fixed point of the policy mapping T_μ.

The following result provides a necessary and sufficient condition for optimality of a stationary policy for a stochastic shortest path problem. The result parallels that of Theorem 57 for the discounted cost problem.

Theorem 63 *A proper stationary policy μ is optimal if and only if μ attains the minimum in Bellman's equation, i.e.,*

$$\mu(i) \in \operatorname*{Argmin}_{u\in U(i)} \{\bar{g}(i,u) + \sum_{j=1}^{n} p_{ij}(u) V^*(j)\} \qquad \text{for all } i = 1, \ldots, n,$$

or equivalently

$$TV^* = T_\mu V^*.$$

The value and policy iteration algorithms discussed for discounted cost problems in Sections 5.2.2 and 5.2.3 apply to stochastic shortest path problems with appropriate interpretations of the DP mapping T and the policy mapping T_μ. We discuss these algorithms in the following two sections.

5.3.2 Value Iteration

Recall that for the stochastic shortest path problem the DP mapping $T : \mathbb{R}^n \to \mathbb{R}^n$ is

$$TV(i) = \min_{u \in U(i)} \left\{ \bar{g}(i,u) + \sum_{j=1}^{n} p_{ij}(u) V(j) \right\} \qquad \text{for all } i = 1, \ldots, n$$

[cf. Eq. (5.7)]. Note that this is a reduced mapping that always assumes $V(t) = 0$, since $V^*(t) = 0$. The value iteration algorithm for stochastic shortest path problem generates a sequence of vectors

$$V_k = T^k V,$$

starting with some initial vector $V \in \mathbb{R}^n$. According to Theorem 61, we have

$$\lim_{k \to \infty} (T^k V) = V^*,$$

where $V^* \in \mathbb{R}^n$ is the optimal cost vector. Thus, the value algorithm converges to the optimal cost value.

In general, the value algorithm for the stochastic shortest path problem does not terminate with the optimal cost in a finite number of steps. However, there are some special conditions under which a finite termination may occur. One such case is when a (proper) stationary policy exists for which the corresponding transition probability graph has no directed cycle. In particular, the transition probability graph corresponding to a given stationary policy μ is the graph induced by the Markov chain with transition probabilities P_μ and absorbing state t. Specifically, the transition probability graph of a given stationary policy μ is the graph $G_\mu = (\mathcal{N}, \mathcal{L}_\mu)$ with nodes $\mathcal{N} = 1, \ldots, n, t$ and the set $\mathcal{L}_\mu$ of links defined by

$$\mathcal{L}_\mu = \{(i,j) \mid p_{ij}(\mu(i)) > 0,\ i, j \in \mathcal{N}\}.$$

We now state the result for finite termination under acyclic optimal policy assumption.

Theorem 64 *Assume that the stochastic shortest path problem has an optimal stationary policy μ^* whose corresponding transition probability graph does not have any directed cycle. Then, the value iteration algorithm terminates in at most n iterations for an initial vector $V \in \mathbb{R}^n$ with components $V(i)$ large enough.*

Proof. The formal proof uses

$$V(i) = \infty \qquad \text{for all } i = 1, \ldots, n$$

and recall that $V(t) = 0 = V^*(0)$. Define the sets S_k of states as follows

$$S_{k+1} = \{i \mid p_{ij}(\mu(i)) = 0 \text{ when } j \notin \cup_{r=0}^{k} S_r\},$$

with $S_0 = \{t\}$. The set S_{k+1} is the set of all nodes that have no ingoing links to the set $\cup_{r=0}^{k} S_r$ in the graph G_μ.

By induction on the number k of iterations, we show that

$$V^*(i) = T^k V(i) \qquad \text{for all } i \in \cup_{r=0}^{k} S_r. \tag{5.10}$$

Note that for $k = 0$, we have $T^0 V = V$, and the result holds. Suppose now that for some k, equality (5.10) is valid, and consider the vector $T^{k+1}V$ obtained after the next iteration. Since V^* is optimal, we have

$$V^*(i) \leq T^{k+1} V(i) \qquad \text{for all } i = 1, \ldots, n.$$

Hence, for all $i = 1, \ldots, n$,

$$T^{k+1} V(i) \leq \bar{g}(i, \mu^*(i) + \sum_{j=1}^{n} p_{ij}(\mu^*(i)) T^k V(i).$$

By the induction hypothesis, the relation (5.10) holds, implying that

$$T^{k+1} V(i) \leq \bar{g}(i, \mu^*(i) + \sum_{j=1}^{n} p_{ij}(\mu^*(i)) V^*(j)$$

for all $i \in \cup_{r=0}^{k} S_r$. Furthermore, note that the preceding relation also holds for all $i \in S_{k+1}$. Hence, equation (5.10) holds for $k + 1$, thus completing the induction.

Since the graph G_μ is connected, the set sets S_k are nonempty and disjoint. Eventually, their union must be equal to the set $\{1, \ldots, n\}$. Hence, there could be at most n iterations. ■

As another special case consider the (deterministic) shortest path problem in graph from all nodes $i = 1, \ldots, n-1$ to the destination node n of Section 4.3 of Chapter 4. We assume that there are no negative cycles in the directed graph $G = (\mathcal{N}, \mathcal{L})$ with the node set $\mathcal{N} = \{1, \ldots, n\}$. Consider the dual of the shortest path problem

$$\begin{aligned} &\text{maximize} && \sum_{i=1}^{n-1} p_i \\ &\text{subject to} && p_i \leq c_{ij} + p_j \qquad \text{for all } (i,j) \in \mathcal{L}, \end{aligned}$$

where p_i for $i = 1, \ldots, n$ and c_{ij} is the cost of the link $(i, j) \in \mathcal{L}$.

If all components p_j are fixed to some value, except for p_i, then to optimize the objective, we would set p_i to take the largest possible value. This largest value is $\min_{j \in O(i)} \{c_{ij} + p_j\}$, where $O(i)$ is the set of all end-nodes j of the outgoing links (i, j) from node i. Thus, the dual optimal variables p_i^* satisfy such relations, i.e.,

$$p_i^* = \min_{j \in O(i)} \left\{ c_{ij} + p_j^* \right\} \qquad \text{for all } i = 1, \ldots, n-1, \tag{5.11}$$

with p_n^*.

We can view the deterministic shortest path problem as a stochastic problem with $U(i) = O(i)$, $\bar{g}(i, j) = c_{ij}$, and the probability $p_{ij}(u)$ of moving from i to j equal to 1

(when $u = j$) for each $j \in O(i)$. Thus, that the relation (5.11) for optimal prices p^* is equivalent to saying that the optimal price vector p^* is a fixed point of the DP mapping T. Hence, the deterministic shortest path problem can be solved by applying the value iteration algorithm. When all the link costs c_{ij} are nonnegative, the value algorithm can be implemented very efficiently by generating the sets S_k similar to those in the proof of Theorem 64. This implementation as known as *Dijkstra's algorithm*, which we describe next assuming that node n is the terminal node.

Dijkstra's Algorithm

Let $\mathcal{M} = \{n\}$, $p_i = +\infty$ for all $i \neq n$, and $p_n = 0$.

1. At iteration k, we have available the set $\mathcal{M}$ (of permanently marked nodes) and the price vector p. For each node $i \notin \mathcal{M}$ re-compute the values
$$p_i = \min_{j \in O(i)} \{c_{ij} + p_j\}.$$

2. Determine a node $i^* \notin \mathcal{M}$ attaining the minimum of p_i, i.e., a node i^* such that
$$i^* \in \operatorname*{Argmin}_{i \notin \mathcal{M}} p_i.$$
Include this node i^* in the set $\mathcal{M}$, and set $p_i^* = \min_{i \notin \mathcal{M}} p_i$. Go to step 1.

The values p obtained at the end of the algorithm are optimal dual values, with p_i representing the shortest path length from node i to the destination n. Note that the set $\mathcal{M}$ represents the set of (marked) nodes whose dual optimal prices have been computed.

The Dijkstra's algorithm is illustrated in the following example.

Example 37 *Consider the network given in Figure 5.1. Initially, we have only the terminal node 4 marked, i.e.,* $\mathcal{M} = \{4\}$ *and* $\mathbf{p_4 = 0}$*, while* $p_i = +\infty$ *for* $i = 1, 2, 3$.

In the first iteration, the values p_i *for* $i = 1, 2, 3$ *are updated, and we have*

$$p_1 = \min\{1 + p_2, 3 + p_3\} = \min\{+\infty, \infty\} = +\infty,$$
$$p_2 = \min\{1 + p_3, 2 + p_4\} = \min\{+\infty, 2\} = 2,$$
$$p_3 = 0 + p_4 = 0.$$

The smallest value of p_i *is attained at node 3. This node is permanently marked, its label is* $\mathbf{p_3 = 0}$*, and the node 3 enters the set* $\mathcal{M}$*, i.e.,* $\mathcal{M} = \{3, 4\}$ *at the end of the first iteration.*

In the second iteration, we update p_i *for* $i = 1, 2$ *and obtain*

$$p_1 = \min\{1 + p_2, 3 + p_3\} = \min\{1 + 2, 3 + 0\} = 3,$$
$$p_2 = \min\{1 + p_3, 2 + p_4\} = \min\{1 + 0, 2 + 0\} = 2 = 1.$$

Here, the smallest value of p_i *is attained at node 2, and this node is permanently marked as it enters the set* $\mathcal{M}$*, i.e., we have* $\mathcal{M} = \{2, 3, 4\}$ *at the end of this iteration with* $\mathbf{p_2 = 1}$.

In the third iteration, we update only p_1 *and obtain*

$$p_1 = \min\{1 + p_2, 3 + p_3\} = \min\{1 + 1, 3 + 0\} = 2.$$

Hence, we have $\mathbf{p_1 = 2}$ *and node 1 enters the set* $\mathcal{M}$*. At this point all nodes are permanently marked, and the algorithm terminates with* $\mathbf{p} = [\mathbf{2}, \mathbf{1}, \mathbf{0}, \mathbf{0}]^{\mathbf{T}}$*, which is the vector with shortest path length values for all nodes.*

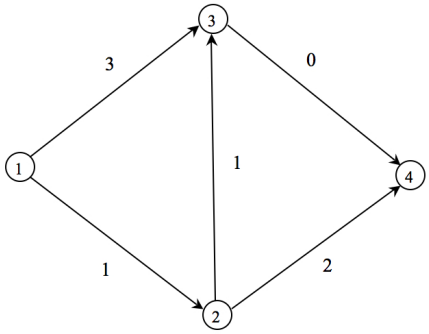

Figure 5.1: A directed graph with nonnegative link costs. The shortest path path from node 1 to node 4 is $1 \to 2 \to 3 \to 4$ with length 2. The shortest path from node 2 to 4 is $2 \to 3 \to 4$ with length 2. The shortest path from 3 to 4 has length 0.

5.3.3 Policy Iteration

The policy iteration can also be used to solve the stochastic shortest path problem. We here consider a policy iteration algorithm that uses an approximation V_k instead of the exact value of the policy cost V_{μ^k}.

In particular, for the policy mapping T_μ,

$$T_\mu V(i) = \bar{g}(i, \mu(i)) + \sum_{j=1}^{n} p_{ij}(\mu(i)) V(i) \qquad \text{for all } i = 1, \ldots, n$$

[cf. Eq. (5.8)] we consider an *approximate policy iteration algorithm*, where an approximation V_k of V_{μ^k} is used in the policy improvement step to determine a new policy μ^{k+1}. Formally, the algorithm is described as follows.

Approximate Policy Iteration

Let δ be a given error tolerance level, and let μ^0 be an initial (proper) policy.

1. At iteration k, having the current policy μ^k, determine a δ-approximation V_k of the policy cost function V_{μ^k}, i.e., V_k such that

$$\max_{1 \le i \le n} \left| V_k(i) - V_{\mu^k}(i) \right| \le \delta.$$

2. Generate a new policy μ^{k+1} satisfying $T_{\mu^{k+1}} V_k = T V_k$, or explicitly

$$\mu^{k+1}(i) \in \operatorname*{Argmin}_{u \in U(i)} \left\{ \bar{g}(i, u) + \sum_{j=1}^{n} p_{ij}(u) V_k(j) \right\} \qquad \text{for all } i = 1, \ldots, n.$$

Furthermore, the new policy at step 2 need not be defined by exactly applying the DP mapping T. In particular, when the number of state is large, we may use μ^{k+1} such that

$$\max_{1 \le i \le n} \left| T_{\mu^{k+1}}(i) - T V_k(i) \right| \le \epsilon.$$

When both the policy value V_{μ^k} and the DP map are approximated, we have the following error bound for the approximate policy iteration algorithm (see for example [7], Section 2.4, Proposition 2.4.1).

Theorem 65 *When all the policies μ^k in the (δ, ϵ)-approximate policy iteration algorithm are proper, the following relation holds:*

$$\limsup_{k\to\infty} \max_{1\le i\le n} \left|V_{\mu^k}(i) - V^*(i)\right| \le n(\epsilon + 2\delta)\frac{n+1-\rho}{(1-\rho)^2},$$

with $1-\rho$ being the smallest probability that the state t is reached in n steps from any state i under any policy μ, or equivalently

$$\rho = \max_{\substack{1\le i\le n \\ \text{proper } \mu}} \text{Prob}\,\{x_n \neq t \mid x_0 = i,\ \mu\}.$$

5.4 Average Cost Problem

Here, we study another DP model with stationary description. This model does not use a discount factor α and does not have a termination state. We will consider the case where the state space and control space is finite. In particular, the structure of the average cost DP problem is as follows:

1. The set S of system states is $S = \{1, \ldots, n\}$.
2. For each state $i \in S$, a set $U(i)$ specifies the controls u available at that state. The set $U(i)$ is finite for all i.
3. At each state i, every control u specifies a probability distribution $p_{ij}(u)$ of moving from state i to state $j \in S$, i.e.,
$$p_{ij}(u) = \text{Prob}\,\{x_{k+1} = j \mid x_k = i, u_k = u\} \qquad \text{for all } i \in S.$$
4. A cost $g(i, u, j)$ is incurred when moving from state i to state j under the control u.
5. The objective is to minimize the long-term average cost $V_\pi(i)$ over all policies $\pi = (\mu_0, \mu_1, \ldots)$, where the average cost $V_\pi(i)$ starting from state i is given by:
$$V_\pi(i) = \limsup_{N\to\infty} \frac{1}{N}\,\text{E}\left\{\sum_{k=0}^{N-1} g(x_k, \mu_k(x_k), x_{k+1}) \,\middle|\, x_0 = i\right\}.$$

Similarly as for the stochastic shortest path, we can associate a deterministic cost $\bar{g}(i, u)$ for choosing a control u at state i, by letting

$$\bar{g}(i, u) = \text{E}[g(i, u, j)|(i, u)] = \sum_{j=1}^{n} p_{ij}(u)\, g(i, u, j) \qquad \text{for all } i \text{ and all } u \in U(i).$$

The problem is now to determine the policy π^* such that for all $i = 1, \ldots, n$,

$$V_{\pi^*}(i) = \min_{\pi \in \Pi} V_\pi(i),$$

where $\pi = (\mu_0, \mu_1, \ldots)$ and

$$V_\pi(i) \;=\; \limsup_{N\to\infty} \frac{1}{N} \left\{ \sum_{k=0}^{N-1} \sum_{j=1}^{n} p_{i_k j}(\mu_k(i_k))\, g(i_k, \mu_k(i_k), j) \;\Big|\; i_0 = i \right\}.$$

For a stationary policy $\pi = (\mu, \mu \ldots)$, we view the policy cost V_μ as a vector with components $V_\mu(i)$, and similarly we view the optimal average cost $V^* = V_{\pi^*}$. Also, we introduce the transition matrix P_μ corresponding to a stationary policy μ,

$$P_\mu = \begin{bmatrix} p_{11}(\mu(1)) & \cdots & p_{1n}(\mu(1)) \\ \vdots & \ddots & \vdots \\ p_{n1}(\mu(n)) & \ldots & p_{nn}(\mu(n)) \end{bmatrix}.$$

Note that the matrix P_μ *is stochastic.* We introduce the cost vector $\bar{g}_\mu$ associated with a stationary policy μ,

$$g_\mu = \begin{bmatrix} \bar{g}(1, (\mu(1)) \\ \vdots \\ \bar{g}(n, (\mu(n)) \end{bmatrix}.$$

Then, for the policy cost vector V_μ, we can write

$$V_\mu = \limsup_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} P_\mu^k\, \bar{g}_\mu.$$

Some applications of the average cost DP model are illustrated in the following examples.

Example 38 *(Transmission Scheduling [33])*
Consider a communication network with a receiver and a transmitter. The network topology state is $S(t)$, which is assumed to evolve according to some irreducible Markov chain with a finite set of states.

At time t, the transmission attempts are denoted by a scalar $R(t)$ with $R(t) = 1$ if the transmitter attempts a transmission at time t, and otherwise $R(t) = 0$. The transmissions are not always successful. Specifically, at each time t, the transmission is successful with probability $P(t)$, which depends on the network state $S(t)$ and the transmission attempt $R(t)$ at time t, i.e.,

$$P(t) = P(S(t), R(t)) \qquad \text{for all } t \geq 1.$$

The transmission control policy is the collection of transmission scalars $\{R(t)\}$. Due to the uncertainties in the transmissions, some of these transmissions are successful and some are not. Let $D(t)$ be the indicator variable of successful transmissions at time t, i.e., with $D(t) = 1$ if the transmission is successful at time t, and otherwise $D(t) = 0$.

The time average throughput of the system is

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} D(t).$$

The problem is to maximize the average throughput over all policies $\{R(t)\}$ for which the limiting value exists.

Example 39 *(Multiclass Queueing Network [15]) Consider a queueing network system with n queues and a single server with variable service rate. Each queue i has an arrival rate λ_i and a service rate μ_i. The goal is to assign the jobs from the queues to the server so as to minimize the long term average of the queue lengths.*

The DP formulation is as follows: the time is slotted $t = 0, 1, \ldots$. It is assumed that, at each time, either a new job arrives to a queue i, or a job from a queue i that is currently being served is completed. Depending on the network system, a completed job from any queue may exit the system or enter another queue.

The state of the system is the vector $x(t)$ with components $x_i(t)$ representing the number of jobs in queue i (queue length). At each time and state, the set of controls is $U = \{e_1, \ldots, e_n\}$, where a decision $u = e_i$ corresponds to a job from queue i being selected to be served. The evolution of the system is given by

$$x_i(t+1) = x_i(t) + a_i(t+1) - d_i(t+1) \qquad \text{for all } i,$$

where $a_i(t)$ is the arrival and $d_i(t)$ is the departure indicator variable for queue i. The objective is to select the jobs from queues to be served, i.e., decide on the sequence $\{u_k\}$, so as to minimize the average cost given by

$$\limsup_{T\to\infty} \frac{1}{T} \, \mathrm{E}\left[\sum_{t=0}^{T-1} \sum_{i=1}^{n} x_i(t) \,\middle|\, x_0 = x \right].$$

Example 40 *(Power Allocation and Routing in Wireless Networks [23])*
Consider a wireless network with n power constrained nodes. At each time t, multiple random streams $a_i^j(t)$ of bits that enter the system at node i destined to node j. Data packets are dynamically routed from node to node through the paths using wireless data links. The link conditions are randomly changing in time, and the time is slotted $t = 0, 1, \ldots$.

Nodes transmit data over multiple links depending on the power $P(t)$ distribution at time t, $P(t) = [p_{ab}(t),\, a, b = 1 \ldots, n]$, which satisfies the power constraint at each node. At time t, transmission rates $r_{ab}(t)$ over all links (a, b) depend on the power matrix $P(t)$ and the state of the network links $S(t) = [s_{ab}(t),\, a, b = 1 \ldots, n]$, i.e., $r_{ab}(t) = r_{ab}(P(t), S(t))$. Each node has $n - 1$ queues for storing data according to its destination.

At each time t, a controller allocates the power $P(t)$. Based on the power-rate curve, the link rate matrix is determined $R(t) = [r_{ab}(t),\, a, b = 1 \ldots, n]$. Each resulting link rate $r_{ab}(t)$ has to be split into rates $r_{ab}^j(t)$ for traffic destined to each node j, resulting into the rate assignment matrix $R^j(t) = [r_{ab}^j(t)\, a, b = 1 \ldots, n]$ for the traffic destined to every node

j. The objective is to determine the power allocation matrices $P(t)$ and rate allocations $R^j(t)$,$j = 1, \ldots, n$, so as to maximize the network throughput while keeping the delay low.

The corresponding DP model is as follows: At time t, the system state consists of the network state $S(t)$ and the current system backlog $W(t) = [w_i^j(t),\, i, j = 1, \ldots, n]$ where $w_i^j(t)$ is the number of bits at node i destined for node j at time t.

At each state $(S(t), W(t))$, the set of controls depends only on $S(t)$, and consists of the set of all power allocation matrices $P(t)$ satisfying the power constraints of the nodes and the rate splitting decisions. In particular, the power allocation $P(t)$ is selected so that

$$\sum_{b \neq i} p_{ib}(t) \leq p_i^{\text{cap}} \qquad \text{for all } i = 1, \ldots, n.$$

Based on the power-rate curve and the system state, the link rates $r_{ab}(t) = r_{ab}(S(t), P(t))$ are determined corresponding to the chosen power $P(t)$. Then, the rate splitting $r_{ab}^j(t)$ is decided such that

$$\sum_j r_{ab}(t)^j \leq r_{ab}(t) \qquad \text{for all } (a, b).$$

The network $S(t)$ evolves according to an ergodic Markov chain with finite set of states. The system backlog $W(t)$ evolves as a function of the state $S(t)$, the selected power allocation $P(t)$ and rate allocation $R(t)$, as follows:

$$w_i^j(t+1) \leq \max \left\{ w_i^j(t) - \sum_b r_{ib}^j(t),\, 0 \right\} + \sum_a r_{ai}^j(t) + A_i^j(t).$$

The objective is to minimize the following average cost

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \left\{ \sum_{i,j} \mathrm{E}\left[w_i^j(t) \right] \right\}.$$

5.4.1 Basic Relations

Here, we provide some basic relations and optimality criteria for the average cost problem. We start by showing that this problem can be viewed as the limit of a sequence of related discounted cost problems with discount factor α approaching value 1.

We note that given any stochastic matrix P, the limiting matrix

$$\bar{P} = \lim_{N \to \infty} \frac{1}{N} \sum_{k=0}^{N-1} P^k$$

always exists. The (i, j)th entry of $\bar{P}$ represents the long term frequency of visits to state j given that the initial state is i. Intuitively, such frequencies are well defined. A sketch of the formal proof is as follows: for a matrix P with eigenvalues on a unit circle (holds for stochastic matrices) we have for any $0 < \alpha < 1$,

$$\sum_{k=0}^{\infty} \alpha^k P^k = (I - \alpha P)^{-1}.$$

Hence,

$$(1-\alpha)(I-\alpha P)^{-1} = \lim_{N\to\infty} \frac{\sum_{k=0}^{N-1} \alpha^k P^k}{\sum_{k=0}^{N-1} \alpha^k}.$$

Letting $\alpha \to 1$ (and assuming we can exchange the limits), we have

$$\bar{P} = \lim_{\alpha\to 1}(1-\alpha)(I-\alpha P)^{-1} = \lim_{N\to\infty}\lim_{\alpha\to 1} \frac{\sum_{k=0}^{N-1} \alpha^k P^k}{\sum_{k=0}^{N-1} \alpha^k} = \lim_{N\to\infty} \frac{\sum_{k=0}^{N-1} P^k}{N}.$$

Hence, the limiting average transition matrix $\bar{P}_\mu = \lim_{N\to\infty} \frac{\sum_{k=0}^{N-1} P_\mu^k}{N}$ of the policy μ can be viewed as the limit of the corresponding transition matrix P_μ of the stationary policy μ for the discounted problem as the discount factor α approaches 1, i.e.,

$$\bar{P}_\mu = \lim_{\alpha\to 1}(1-\alpha)(I-\alpha P_\mu)^{-1}. \tag{5.12}$$

In view of the preceding, the *limsup* in the stationary policy cost can be replaced with the limit, so that

$$V_\mu = \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} P_\mu^k\, \bar{g}_\mu = \bar{P}_\mu\, \bar{g}_\mu. \tag{5.13}$$

By viewing $\bar{P}_\mu$ as the limiting (average) steady state frequency associated with the policy μ, the vector V_μ is the steady state cost corresponding to the policy μ.

Consider now the cost that would be accumulated under a given policy μ, if the cost $\bar{g}_\mu$ per stage were discounted by a factor $\alpha \in (0,1)$. Let $V_{\mu,\alpha}$ denote the expected discounted cost for the policy μ, i.e.,

$$V_{\mu,\alpha} = \sum_{k=0}^{\infty} \alpha^k P_\mu^k\, \bar{g}_\mu.$$

By noting that $\sum_{k=0}^{\infty} \alpha^k P_\mu^k = (I-\alpha P_\mu)^{-1}$, we have

$$V_{\mu,\alpha} = (I-\alpha P_\mu)^{-1}\, \bar{g}_\mu. \tag{5.14}$$

By post-multiplying with $\bar{g}_\mu$ both sides in equation (5.12), and by using relations (5.13) and (5.14), we obtain

$$V_\mu = \lim_{\alpha\to 1}(1-\alpha) \sum_{k=0}^{\infty} \alpha^k P_\mu^k\, \bar{g}_\mu = \lim_{\alpha\to 1} (1-\alpha) V_{\mu,\alpha}, \tag{5.15}$$

Thus, the (averaged) policy cost V_μ is related to the limiting discounted cost $V_{\mu,\alpha}$ as the discount factor approaches 1.

The formal proof of equations (5.12) and (5.15) can be found in [7] Section 4.1. Relation (5.12) suggest that we can approximate the average transition matrix $\bar{P}_\mu$ with matrices $(1-\alpha)^{-1}(I-\alpha P_\mu)$ by taking the discount factor α close to 1, where $(I-\alpha P_m)$ is the matrix associated with a corresponding discounted problem. Similarly, relation (5.15) indicates that we can approximate the average cost V_μ of a stationary policy μ with the corresponding

discounted cost $V_{\mu,\alpha}$ by taking the discount factor α close to 1. Formally, using equations (5.12) and (5.15), we can write for α close to 1,

$$(I - \alpha P_\mu) = (1-\alpha)^{-1}\bar{P}_\mu + Z_\mu + O(1-\alpha),$$

$$V_{\mu,\alpha} = (1-\alpha)^{-1}V_\mu + z_\mu + O(1-\alpha),$$

where $O(\beta)$ is a matrix in the first relation and it is a vector in the second relation, but in each case we have

$$\lim_{\beta \to 0} O(\beta) = 0.$$

The preceding relations are formally stated in the following theorem (with precise forms for the matrix Z_μ and the vector z_μ).

Theorem 66 *Let μ be a stationary policy with the corresponding probability transition matrix P_μ and per-stage cost g_μ.*

(a) Let $\bar{P}_\mu$ be the limiting average probability transition matrix, $\bar{P}_\mu = \lim_{N\to\infty} \frac{1}{N}\sum_{k=0}^{N-1} P_\mu^k$. Then, for any $\alpha \in (0,1)$, we have

$$(I - \alpha P_\mu) = (1-\alpha)^{-1}\bar{P}_\mu + Z_\mu + O(1-\alpha),$$

where $O(\beta)$ is a matrix such that $\lim_{\beta\to 0} O(\beta) = 0$ and Z_μ is a matrix given by

$$Z_\mu = (I - P_\mu + \bar{P}_\mu)^{-1} - \bar{P}_\mu.$$

Furthermore, the matrices $\bar{P}_m$ and Z_μ satisfy the following relations

$$P_\mu \bar{P}_\mu = \bar{P}_\mu P_\mu = \bar{P}_\mu^2, \qquad \bar{P}_\mu^2 = \bar{P}_\mu,$$

$$\bar{P}_\mu Z_\mu = 0, \qquad Z_\mu \bar{P}_\mu = 0, \qquad \bar{P}_\mu + Z_\mu = I + P_\mu Z_\mu.$$

(b) Let V_μ be the average cost for the policy μ, i.e., $V_\mu = \bar{P}_\mu \bar{g}_\mu$. Let $V_{\mu,\alpha}$ be the expected discounted long term cost for the policy μ and the discount factor $\alpha \in (0,1)$, i.e., $V_{\mu,\alpha} = (I - \alpha P_\mu)^{-1}\bar{g}_\mu$. Then, for any $\alpha \in (0,1)$, we have

$$V_{\mu,\alpha} = (1-\alpha)^{-1}V_\mu + z_\mu + O(1-\alpha),$$

where $O(\beta)$ is a vector such that $\lim_{\beta\to 0} O(\beta) = 0$ and z_μ is a vector given by

$$z_\mu = Z_\mu \bar{g}_\mu.$$

The average cost V_μ is such that

$$V_\mu = P_\mu V_\mu, \qquad V_\mu + z_\mu = \bar{g}_\mu + P_\mu z_\mu.$$

The vector z_μ is referred to as *the bias* of the policy μ.

We next discuss a special stationary policies for the average cost problem, namely, Blackwell optimal stationary policies. A stationary policy μ is *Blackwell optimal* for the average cost problem if there is an interval $(\tilde{\alpha}, 1)$ such that the policy μ is (uniformly) optimal for all corresponding discounted problems with the discount factor $\alpha \in (\tilde{\alpha}, 1)$. The finite state set and the finite control set assumptions in the average cost model ensure the existence of a Blackwell optimal stationary policy.

Based on the uniformity property of a Blackwell optimal policy and the properties of V_μ established in Theorem 66, it can be seen that all Blackwell optimal policies have the same cost V_μ and the same bias z_μ,

$$V_\mu = V_{\tilde{\mu}}, \qquad z_\mu = z_{\tilde{\mu}} \qquad \text{for all Blackwell optimal stationary policies } \mu \text{ and } \tilde{\mu}.$$

Denote the common cost vector and the common gain vector of all Blackwell optimal policies by V^* and z^* respectively, i.e.,

$$V_b = V_\mu, \qquad z^* = z_\mu \qquad \text{for all Blackwell optimal stationary policies } \mu.$$

The vector V^* and the gain z^* satisfy some special relations, which play role of Bellman's equation for the average cost problem and lead to the optimality conditions.

The following theorem provides the relations for the cost V^* and the gain z^*.

Theorem 67 *The vector V^* corresponding to the cost of Blackwell optimal policies is such that*

$$V^*(i) = \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u)\, V^*(j) \qquad \textit{for all } i = 1, \ldots, n.$$

For each i, let $U^*(i)$ be the set of controls u attaining the minimum in the right hand side of the preceding relation. The bias z^* of Blackwell optimal policies is such that

$$V^*(i) + z^*(i) = \min_{u \in U^*(i)} \left\{ \bar{g}(i,u) + \sum_{j=1}^{n} p_{ij}(u)\, z^*(j) \right\}.$$

The relations of Theorem 67 play the role of Bellman's equation for the average cost problem. If a policy μ is Blackwell optimal, then its cost and the gain satisfy

$$V^*(i) = \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u)\, V^*(j) \qquad \text{for all } i = 1, \ldots, n, \tag{5.16}$$

$$V^*(i) + z^*(i) = \min_{u \in U^*(i)} \left\{ \bar{g}(i,u) + \sum_{j=1}^{n} p_{ij}(u)\, z^*(j) \right\}. \tag{5.17}$$

where $U^*(i)$ is the set of controls u attaining the above minimum. The converse is also true, as seen from the following.

Theorem 68 *Suppose that the vectors V^* and z^* satisfy relations in Eqs. (5.16)–(5.17). Then, V^* is the optimal cost vector for the average cost problem.*

Let μ^* be a policy such that $\mu^*(i)$ attains the minimum in Eq. (5.17). Then, the policy μ^* is an optimal stationary policy for the average cost problem.

In general, we cannot replace $U^*(i)$ by $U(i)$ in Eq. (5.17). However, a modification of Eq. (5.17) holds for some policies, as seen in the following.

Theorem 69 *Suppose that the vectors V^* and z^* satisfy relations in Eqs. (5.16)–(5.17). Then, there is a scalar $\bar{\delta} > 0$ such that for all $\delta \geq \bar{\delta}$, the vector*

$$z_\delta = z^* + \delta V^*$$

satisfies

$$V^*(i) + z_\delta(i) = \min_{u \in U(i)} \left\{ \bar{g}(i,u) + \sum_{j=1}^{n} p_{ij}(u)\, z_\delta(j) \right\} \qquad \text{for all } i = 1, \ldots, n. \tag{5.18}$$

In addition, if a policy μ attains the minimum in the optimality relation (5.17), then it also attains the minimum in the perturbed optimality relation (5.18).

The preceding theorem will be useful in the subsequent development of the algorithms for average cost problems.

A special case when the optimality relations (5.16)–(5.17) take a particularly simplified form is when the optimal average cost $V^*(i)$ is the same for all states i. This is the case for example, when every stationary policy induces an ergodic[1] Markov chain (with transition probabilities P_μ). The reason for this is the fact that for an ergodic Markov chain, the long term behavior is the same for any initial state of the chain.

From now on, we assume that *every stationary policy* induces an ergodic Markov chain. Under this assumption, the optimality relations (5.16)–(5.17) are more tractable. Relation (5.16) becomes redundant and implies $U^*(i) = U(i)$ for all i. Relation (5.17) reduces to

$$\theta + z^*(i) = \min_{u \in U(i)} \left\{ \bar{g}(i,u) + \sum_{j=1}^{n} p_{ij}(u)\, z^*(j) \right\} \qquad \text{for all } i = 1, \ldots, n, \tag{5.19}$$

where $\theta = V^*(i)$ [the same value for all i]. We refer to this relation as *Bellman's equation* for average cost problem. We define the DP mapping $T : \mathbb{R}^n \to \mathbb{R}^n$ as: for all $z \in \mathbb{R}^n$,

$$Tz(i) = \min_{u \in U(i)} \left\{ \bar{g}(i,u) + \sum_{j=1}^{n} p_{ij}(u)\, z(j) \right\} \qquad \text{for all } i = 1, \ldots, n. \tag{5.20}$$

Then, Bellman's equation (5.19) can be written as

$$\theta e + z^* = Tz^*.$$

[1] An aperiodic Markov chain with a single recurrent class.

Note that, a stationary policy that attains the minimum in the right hand side of Bellman's equation (5.19) is optimal.

Given a stationary policy, under ergodicity assumption, the policy cost $V_\mu(i)$ has the same values for all states i. Denoting this value by θ_μ, from Bellman equation we have

$$\theta_\mu + z(i) = \bar{g}(i, \mu(i)) + \sum_{j=1}^{n} p_{ij}(\mu_(i))\, z(j).$$

Define the policy mapping $T_\mu : \mathbb{R}^n \to \mathbb{R}^n$ by: for all $z \in \mathbb{R}^n$,

$$T_\mu z(i) = \bar{g}(i, \mu(i)) + \sum_{j=1}^{n} p_{ij}(\mu(i))\, z(j) \qquad \text{for all } i = 1, \ldots, n. \tag{5.21}$$

Then, Bellman's equation (5.19) for a policy μ can be written as

$$\theta_\mu e + z = T_\mu z.$$

We next summarize the optimality conditions for the ergodic case. The following result is an immediate consequence of Theorem 68.

Theorem 70 *Suppose that the vectors θ and z are such that $\theta e + z = Tz$, or explicitly*

$$\theta + z(i) = \min_{u \in U(i)} \left\{ \bar{g}(i, u) + \sum_{j=1}^{n} p_{ij}(u)\, z(j) \right\} \qquad \textit{for all } i = 1, \ldots, n.$$

Then, θ is the optimal average cost for all states and a stationary policy μ attaining the minimum in the preceding relation is optimal, i.e.,

$$\theta = V^*(i) \qquad \textit{for all } i = 1, \ldots, n,$$

$$V_\mu(i) = \theta \qquad \textit{for all } i = 1, \ldots, n.$$

We provide a similar result for the average cost V_μ of a stationary policy μ.

Theorem 71 *Suppose that the vectors θ and z are such that $\theta e + z = T_\mu z$, or explicitly*

$$\theta + z(i) = \bar{g}(i, \mu(i)) + \sum_{j=1}^{n} p_{ij}(\mu(i))\, z(j) \qquad \textit{for all } i = 1, \ldots, n.$$

Then, θ is the average cost of policy μ for all states, i.e.,

$$\theta = V_\mu(i) \qquad \textit{for all } i = 1, \ldots, n.$$

5.4.2 Value Iteration

We consider a value iteration algorithm under ergodicity assumption on stationary policies. Basically, the algorithm produces a vector sequence $\{T^k z\}$ starting with an arbitrary vector $z \in \mathbb{R}^n$, and the vector

$$V_k = \frac{1}{k} T^k z,$$

is an approximation of the optimal cost V^*.

The justification of the value iteration algorithm is provided by the following result.

Theorem 72 *Let $h \in \mathbb{R}^n$ be arbitrary. We have*

$$V^* = \lim_{k\to\infty} \frac{1}{k} T^k z,$$

where T is the DP mapping associated with the average cost problem [cf. Eq. (5.20)].

Proof. (Sketch) Consider the vector z^* satisfying Bellman's equation,

$$Tz^* = V^* + z^*.$$

From this relation, it follows that

$$T^2 z^* = TV^* + Tz^* = V^* + (V^* + z^*) = 2V^* + z^*.$$

Continuing in this manner, we see that

$$T^k z^* = kV^* + z^* \qquad \text{for all } k \geq 1.$$

Next step is to estimate the difference $T^k z - T^k z^*$. Consider $Tz - Th$ for arbitrary $z, h \in \mathbb{R}^n$. We have for all i,

$$Tz(i) - Th(i) = \min_{u\in U(i)} \left\{ \bar{g}(i,u) + \sum_{j=1}^n p_{ij}(u)\, z(j) \right\} - \min_{u\in U(i)} \left\{ \bar{g}(i,u) + \sum_{j=1}^n p_{ij}(u)\, h(j) \right\}.$$

Let μ_z be a stationary policy attaining the minimum in $\min_{u\in U(i)} \left\{ \bar{g}(i,u) + \sum_{j=1}^n p_{ij}(u)\, z(j) \right\}$, and μ_h be a stationary policy attaining the minimum in $\min_{u\in U(i)} \left\{ \bar{g}(i,u) + \sum_{j=1}^n p_{ij}(u)\, h(j) \right\}$. By optimality of μ_h for the term involving h, we have for all i,

$$\begin{aligned} Tz(i) - Th(i) &\leq \bar{g}(i,\mu_h(i)) + \sum_{j=1}^n p_{ij}(\mu_h(i))\, z(j) - \left(\bar{g}(i,\mu_h(i)) + \sum_{j=1}^n p_{ij}(\mu_h(i))\, h(j) \right) \\ &= \sum_{j=1}^n p_{ij}(\mu_h(i))\ (z(j) - h(j)) \\ &\leq \max_{1\leq j\leq n} (z(j) - h(j)) \sum_{j=1}^n p_{ij}(\mu_h(i)). \end{aligned}$$

Hence, for all i,

$$Tz(i) - Th(i) \leq \max_{1\leq j\leq n} (z(j) - h(j)) . \tag{5.22}$$

Similarly, by optimality of μ_z for the term involving z, we have for all i,

$$\begin{aligned} Tz(i) - Th(i) &\geq \bar{g}(i, \mu_z(i)) + \sum_{j=1}^{n} p_{ij}(\mu_z(i))\, z(j) - \left(\bar{g}(i, \mu_z(i)) + \sum_{j=1}^{n} p_{ij}(\mu_z(i))\, h(j) \right) \\ &= \sum_{j=1}^{n} p_{ij}(\mu_z(i))\ (z(j) - h(j)) \\ &\geq \min_{1\leq j\leq n} (z(j) - h(j)) \sum_{j=1}^{n} p_{ij}(\mu_z(i)). \end{aligned}$$

Therefore, for all i,

$$Tz(i) - Th(i) \geq \min_{1\leq j\leq n} (z(j) - h(j)) . \tag{5.23}$$

By combining relations (5.22) and (5.23), we have for all $z, h \in \mathbb{R}^n$ and all $i = 1 \ldots, n$,

$$\min_{1\leq j\leq n} (z(j) - h(j)) \leq \min_{1\leq i\leq n} (Tz(i) - Th(i)) \leq Tz(i) - Th(i),$$

$$Tz(i) - Th(i) \leq \max_{1\leq i\leq n} (Tz(i) - Th(i)) \leq \max_{1\leq j\leq n} (z(j) - h(j)) .$$

By repeated application of the preceding relations to $T^k z(i) - T^k z^*(i)$, we can see that

$$\min_{1\leq j\leq n} (z(j) - z^*(j)) \leq T^k z(i) - T^k z^*(i) \leq \max_{1\leq j\leq n} (z(j) - z^*(j)) \qquad \text{for all } i.$$

Using the relation $T^k z^* = kV^* + z^*$ in the preceding relation, we see that

$$\min_{1\leq j\leq n} (z(j) - z^*(j)) + kV^* + z^*(i) \leq T^k z(i) \leq \max_{1\leq j\leq n} (z(j) - z^*(j)) + kV^* + z^*(i).$$

Dividing by k, we obtain

$$V^* + \frac{1}{k} \left(\min_{1\leq j\leq n} (z(j) - z^*(j)) + z^*(i) \right) \leq \frac{1}{k} T^k z(i) \leq V^* + \frac{1}{k} \left(\max_{1\leq j\leq n} (z(j) - z^*(j)) + z^*(i) \right),$$

and the result follows by letting $k \to \infty$. ■

We note that the result of *Theorem 72 is valid even without the assumption of ergodicity* on stationary policies. The proof however has to be modified by using z_δ instead of z^*, where z_δ is as in Theorem 69.

Theorem 72 shows that the value iteration algorithm can be used to compute the optimal cost V^*. But this does not say how we find the optimal bias z^* and an optimal stationary policy. Define

$$\rho_k = T^k z - kV^*.$$

Under ergodicity assumption, it turns out that

$$T^k z - T^k z^* \to 0 \qquad \text{for any } z \in \mathbb{R}^n.$$

Since

$$\rho_k = T^k z - kV^* = T^k z - T^k z^* + z^*,$$

where we use $T^k z^* = kV^* + z^*$, it follows that ρ_k converges to z^*. Then, from

$$T^{k+1}z - T^k z = \rho_{k+1} - \rho_k + V^* \qquad \text{for all } k,$$

we obtain

$$\lim_{k\to\infty} \left(T^{k+1}z - T^k z\right) = V^* \qquad \text{for any } z \in \mathbb{R}^n.$$

Thus, under the ergodicity assumption, we have

$$T^{k+1}z - T^k z \to V^* \qquad \text{and} \qquad \rho_k \to z^*,$$

where V^* is optimal average cost and z^* is the optimal bias satisfying Bellman's equation $V^* + z^* = Tz^*$.

As one can observe, the analysis of the value iteration algorithm for the average cost problem is much more complex than for the discounted cost and shortest path problems. The complexity stems from the structure of the problem and the impact of Markov chains associated with policies. The situation is the same for the policy iteration algorithm, which is discussed next.

5.4.3 Policy Iteration

We consider a *policy iteration algorithm* generating a sequence of policies $\{\mu^k\}$ using the policy mapping T_μ [cf. Eq. (5.21)],

$$T_\mu z(i) = \bar{g}(i, \mu(i)) + \sum_{j=1}^{n} p_{ij}(\mu(i)) z(i) \qquad \text{for all } i = 1, \ldots, n.$$

Policy Iteration
Let μ^0 be an initial (ergodic) policy.

1. At iteration k, having the current policy μ^k, determine the average policy value V_k and its bias z_k, i.e., V_k and z_k such that

$$V_k + z_k(i) = \bar{g}(i, \mu_k(i)) + \sum_{j=1}^{n} p_{ij}(\mu_k(i))\, z_k(j) \qquad \text{for all } i = 1, \ldots, n.$$

2. Generate a new policy μ^{k+1} satisfying $T_{\mu^{k+1}} z_k = Tz_k$, or explicitly

$$\mu^{k+1}(i) \in \operatorname*{Argmin}_{u\in U(i)} \left\{ \bar{g}(i,u) + \sum_{j=1}^{n} p_{ij}(u) z_k(j) \right\} \qquad \text{for all } i = 1, \ldots, n.$$

If $\mu^{k+1} = \mu^k$, then stop; the policy μ^k is optimal. Otherwise, go to step 1.

Under the ergodicity assumption on stationary policies, the system of equations at step 1 has n equations and $n+1$ unknowns. Thus, the solution pair (V_k, z_k) is not unique. To ensure the uniqueness, typically, another equation is included

$$z_k(r) = 0 \qquad \text{for some arbitary reference state } r.$$

At step 2, if the policy $\mu^k(i)$ attains the minimum, then the new policy μ^{k+1} is set so that $\mu^{k+1}(i) = \mu^k(i)$, even when there are other controls u attaining the minimum aside from $\mu^k(i)$. This is known as *policy normalization*.

The following result holds for the policy iteration algorithm.

Theorem 73 *When all the policies μ^k generated by the algorithm are ergodic, the policy iteration algorithm using policy normalization terminates with an optimal stationary policy in a finite number of iterations.*

Bibliography

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms and Applications*, Prentice-Hall, Inc., N. J., 1993.

[2] A. Auslender and M. Teboulle, *Asymptotic Cones and Functions in Optimization and Variational Inequalities*, Springer, Berlin, 2002.

[3] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, SIAM, Philadelphia, 2001.

[4] D. P. Bertsekas, *Network optimization: continuous and discrete models*, Athena Scientific, Belmont, MA, 1998.

[5] ——, *Nonlinear Programming*, 2nd Edition, Athena Scientific, Belmont, MA, 1999.

[6] ——, *Dynamic Programming and Optimal Control*, Athena Scientific, Belmont, MA, Volume 1, 2nd ed., 2000.

[7] ——, *Dynamic Programming and Optimal Control*, Athena Scientific, Belmont, MA, Volume 2, 3rd ed., 2007.

[8] D. P. Bertsekas and R. G. Gallager, *Data Networks*, 2nd ed., Prentice-Hall, Englewood Cliffs, N. J., 1992.

[9] D. P. Bertsekas, A. Nedić, and A. Ozdaglar, *Convex Analysis and Optimization*, Athena Scientific, Belmont, MA, 2003.

[10] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation*, Prentice-Hall, Englewood Cliffs, N. J., 1989.

[11] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, Belmont, MA, 1997.

[12] J. M. Borwein and A. S. Lewis, *Convex analysis and nonlinear optimization*, Springer-Verlag, New York Inc., 2000.

[13] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, U. K., 2004.

[14] U. Bränlund, *On Relaxation Methods for Nonsmooth Convex Optimization*, Doctoral Thesis, Royal Institute of Technology, Stockholm, Sweden, 1993.

[15] R. R. Chen and S. Meyn, *Value iteration and optimization of multiclass queueing networks*, Queueing Systems, 32 (1999), pp. 65–97.

[16] R. G. Gallager, *Discrete Stochastic Processes*, Kluwer Academic Publishers, Norwell, MA, 2nd ed., 1998.

[17] B. Hajek, *Communication Network Analysis*, Lecture Notes for ECE 467, University of Illinois, Urbana-Champaign, 2006.

[18] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, N. Y., 1985.

[19] F. P. Kelly, A. K. Maulloo, and D. K. Tan, *Rate control for communication networks: shadow prices, proportional fairness, and stability*, Journal of the Operational Research Society, 49 (1998), pp. 237–252.

[20] A. N. Kolmogorov and S. V. Fomin, *Introductory Real Analysis*, Dover Publications, Inc., N. Y., 1970.

[21] S. H. Low and D. E. Lapsley, *Optimization flow control I: basic algorithm and convergence*, IEEE/ACM Transactions on Networking, 7 (1999), pp. 861–874.

[22] A. Nedić, *Subgradient Methods for Convex Minimization*, MIT Thesis, May 2002.

[23] M. J. Neely, E. Modiano, and C. E. Rohrs, *Dynamic power allocation and routing for time-varying wireless networks*, IEEE Journal on Selected Areas in Communications, 23 (2005), pp. 89–103.

[24] B. T. Polyak, *Minimisation of unsmooth functionals*, Z. Vychisl. Mat. i Mat. Fiz., 9 (1969), pp. 509–521.

[25] ——, *Introduction to optimization*, Optimization Software Inc., 1987.

[26] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, N. J., 1970.

[27] W. Rudin, *Principles of Mathematical Analysis*, McGraw-Hill, N. Y., 1976.

[28] S. Shakkottai and R. Srikant, *Network optimization and control*, Foundations and Trends in Networking, 2 (2007), pp. 271–379.

[29] N. Z. Shor, *Minimization Methods for Nondifferentiable Functions*, Springer-Verlag, Berlin, 1985.

[30] ——, *Nondifferentiable Optimisation and Polynomial Problems*, Kluwer Academic Publishers, 1998.

[31] R. Srikant, *Mathematics of Internet congestion control*, Birkhauser, 2004.

[32] G. Strang, *Linear Algebra and its Applications*, Academic Press, N. Y., 1976.

[33] L. Tassiulas, *Scheduling and performance limits of networks with constantly changing topology*, IEEE Transactions on Information Theory, 43 (1997), pp. 1067–1073.

[34] P. Whittle, *Optimization Over Time*, Wiley, N. Y., Volume 1, 1982, Volume 2, 1983.

[35] V. A. Zorich, *Mathematical Analysis*, Vols. I and II, Springer-Verlag, Berlin Heidelberg, Germany, 2004.