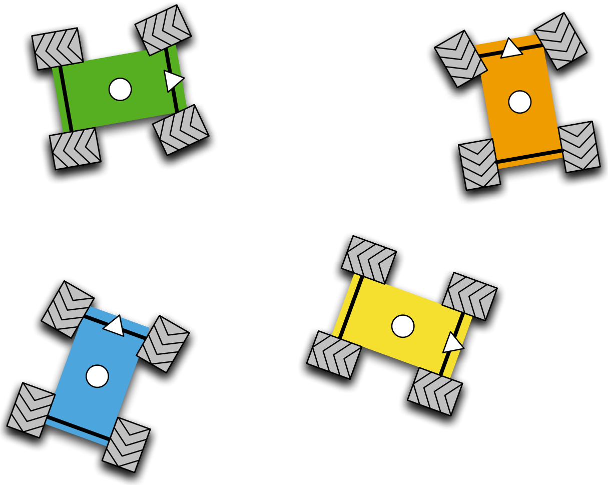
Distributed Control of Robotic Networks

A Mathematical Approach to Motion Coordination Algorithms

Chapter 1: An introduction to distributed algorithms

Francesco Bullo
Jorge Cortés
Sonia Martínez

May 20, 2009



PRINCETON UNIVERSITY PRESS
PRINCETON AND OXFORD

This document is a complete free online version of the following book.

> *Distributed Control of Robotic Networks*, by Francesco Bullo, Jorge Cortés and Sonia Martínez, Applied Mathematics Series, Princeton University Press, 2009, ISBN 978-0-691-14195-4.

The book is available online at

http://coordinationbook.info

(iii) We thank you for any feedback information, including suggestions, evaluations, error descriptions, or comments about teaching or research uses.

Contents

Chapter One

An introduction to distributed algorithms

Graph theory, distributed algorithms, and linear distributed algorithms are a fascinating scientific subject. In this chapter we provide a broad introduction to distributed algorithms by reviewing some preliminary graphical concepts and by studying some simple algorithms. We begin the chapter with one section introducing some basic notation and another section stating a few useful facts from matrix theory, dynamical systems, and convergence theorems based on invariance principles. In the third section of the chapter, we provide a primer on graph theory with a particular emphasis on algebraic aspects, such as the properties of adjacency and Laplacian matrices associated to a weighted digraph. In the next section of the chapter, we introduce the notion of synchronous network and of distributed algorithm. We state various complexity notions and study them in simple example problems such as the broadcast problem, the tree computation problem, and the leader election problem. In the fifth section of the chapter, we discuss linear distributed algorithms. We focus on linear algorithms defined by sequences of stochastic matrices and review the results on their convergence properties. We end the chapter with three sections on, respectively, bibliographical notes, proofs of the results presented in the chapter, and exercises.

1.1 ELEMENTARY CONCEPTS AND NOTATION

1.1.1 Sets and maps

We assume that the reader is familiar with basic notions from topology, such as the notions of open, closed, bounded, and compact sets. In this section, we just introduce some basic notation. We let $x \in S$ denote a point x belonging to a set S. If S is finite, we let $|S|$ denote the number of its elements. For a set S, we let $\mathbb{P}(S)$ and $\mathbb{F}(S)$ denote the collection of subsets of S and the collection of finite subsets of S, respectively. The empty set is denoted by $\emptyset$. The interior and the boundary of a set S are denoted by $\operatorname{int}(S)$ and ∂S, respectively. If R is a subset of or equal to S, then we write $R \subset S$. If R is a strict subset of S, then we write $R \subsetneq S$. We describe subsets of S defined by specific conditions via the notation

$$\{x \in S \mid \text{condition(s) on } x\}.$$

Given two sets S_1 and S_2, we let $S_1 \cup S_2$, $S_1 \cap S_2$, and $S_1 \times S_2$ denote the union, intersection, and Cartesian product of S_1 and S_2, respectively. Given a collection of sets $\{S_a\}_{a \in A}$ indexed by a set A, we interchangeably denote their Cartesian product by $\prod_{a \in A} S_a$ or by $\prod\{S_a \mid a \in A\}$. We adopt analogous notations for union and intersection. We denote by S^n the Cartesian product of n copies of the same S. The *diagonal set* $\operatorname{diag}(S^n)$ of S^n is given by $\operatorname{diag}(S^n) = \{(s, \dots, s) \in S^n \mid s \in S\}$. The set $S_1 \setminus S_2$ contains all points in S_1 that do not belong to S_2.

We let $\mathbb{N}$ and $\mathbb{Z}_{\geq 0}$ denote the set of natural numbers and of non-negative integers, respectively. We let $\mathbb{R}$, $\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{C}$ denote the set of real numbers, strictly positive real numbers, non-negative real numbers, and complex numbers, respectively. The sets $\mathbb{R}^d$, $\mathbb{C}^d$, and $\mathbb{S}^d \subset \mathbb{R}^{d+1}$ are the d-dimensional Euclidean space, the d-dimensional complex space, and the d-dimensional sphere, respectively. The tangent space of $\mathbb{R}^d$, denoted by $T\mathbb{R}^d$, is the set of all vectors tangent to $\mathbb{R}^d$. Note that $T\mathbb{R}^d$ can be identified with $\mathbb{R}^d \times \mathbb{R}^d$ by mapping a vector v tangent to $\mathbb{R}^d$ at $x \in \mathbb{R}^d$ to the pair (x, v). Likewise, $T\mathbb{S}^d$ is the set of all vectors tangent to $\mathbb{S}^d$, and can be identified with $\mathbb{S}^d \times \mathbb{R}^d$. The Euclidean space $\mathbb{R}^d$ contains the vectors $\mathbf{0}_d = (0, \dots, 0)$, $\mathbf{1}_d = (1, \dots, 1)$, and the standard basis $\boldsymbol{e}_1 = (1, 0, \dots, 0), \dots, \boldsymbol{e}_d = (0, \dots, 0, 1)$. Given $a < b$, we let $[a, b]$ and $]a, b[$ denote the closed interval and the open interval between a and b, respectively.

Given two sets S and T, we let $f : S \to T$ denote a map from S to T, that is, a unique way of associating an element of T to an element of S. The *image* of the map $f : S \to T$ is the set $\operatorname{image}(f) = \{f(s) \in T \mid s \in S\}$. Given the map $f : S \to T$ and a set $S_1 \subset S$, we let $f(S_1) = \{f(s) \mid s \in S_1\}$ denote the image of the set S_1 under the map f. Given $f : S \to T$ and $g : U \to S$, we let $f \circ g : U \to T$, defined by $f \circ g\,(u) = f(g(u))$, denote the composition of f and g. The map $\operatorname{id}_S : S \to S$ is the identity map on S. Given $f : S \to \mathbb{R}$, the *support* of f is the set of elements s such that $f(s) \neq 0$. Given a subset $R \subsetneq S$, the indicator map $1_R : S \to \mathbb{R}$ associated with R is given by $1_R(q) = 1$ if $q \in R$, and $1_R(q) = 0$ if $q \notin R$. Given two sets S and T, a *set-valued map*, denoted by $h : S \rightrightarrows T$, associates to an element of S a subset of T. Given a map $f : S \to T$, the *inverse map* $f^{-1} : T \rightrightarrows S$ is defined by

$$f^{-1}(t) = \{s \in S \mid f(s) = t\}.$$

If f is a real-valued function, that is, a function of the form $f : S \to \mathbb{R}$, then $f^{-1}(x) \subset S$, for any $x \in \mathbb{R}$, is a *level set* of f. In what follows, we require the reader to be familiar with some basic smoothness notions for functions. Specifically, we will use the notions of locally and globally Lipschitz functions, differentiable, piecewise differentiable and continuously differentiable functions, and functions that are multiple times differentiable.

Finally, we introduce the so-called *Bachmann–Landau symbols*. For $f, g : \mathbb{N} \to \mathbb{R}_{\geq 0}$, we say that $f \in O(g)$ (resp., $f \in \Omega(g)$) if there exist $n_0 \in \mathbb{N}$ and $K \in \mathbb{R}_{>0}$ (resp., $k \in \mathbb{R}_{>0}$) such that $f(n) \leq Kg(n)$ for all $n \geq n_0$ (resp., $f(n) \geq kg(n)$ for all $n \geq n_0$). If $f \in O(g)$ and $f \in \Omega(g)$, then we use the notation $f \in \Theta(g)$.

1.1.2 Distance functions

A function $\mathrm{dist} : S \times S \to \mathbb{R}_{\geq 0}$ defines a *distance* on a set S if it satisfies: (i) $\mathrm{dist}(x, y) = 0$ if and only if $x = y$; (ii) $\mathrm{dist}(x, y) = \mathrm{dist}(y, x)$, for all $x, y \in S$; and (iii) $\mathrm{dist}(x, y) \leq \mathrm{dist}(x, z) + \mathrm{dist}(z, y)$, for all $x, y, z \in S$. The pair (S, dist) is usually called a *metric space*.

Some relevant examples of distance functions include the following:

L^p-distance on $\mathbb{R}^d$. For $p \in [1, +\infty[$, consider the L^p-norm on $\mathbb{R}^d$ defined by $\|x\|_p = (\sum_{i=1}^{d} |x_i|^p)^{1/p}$. For $p = +\infty$, consider the L^∞-norm on $\mathbb{R}^d$ defined by $\|x\|_\infty = \max_{i \in \{1,\dots,d\}} |x_i|$. Any of these norms defines naturally a L^p*-distance* in $\mathbb{R}^d$ by $\mathrm{dist}_p(x, y) = \|y - x\|_p$. In particular, the most widely used is the Euclidean distance, corresponding to $p = 2$. Unless otherwise noted, it is always understood that $\mathbb{R}^d$ is endowed with this notion of distance. We will also use the L^1- and the L^∞-distances. Finally, it is convenient to define the norm $\|z\|_{\mathbb{C}}$ of a complex number $z \in \mathbb{C}$ to be the Euclidean norm of z regarded as a vector in $\mathbb{R}^2$.

Geodesic distance on $\mathbb{S}^d$. Another example is the notion of *geodesic distance* on $\mathbb{S}^d$. This is defined as follows. For $x, y \in \mathbb{S}^d$, $\mathrm{dist}_g(x, y)$ is the length of the shortest curve in $\mathbb{S}^d$ connecting x and y. We will use this notion of distance in dimensions $d = 1$ and $d = 2$. On the unit circle $\mathbb{S}^1$, by convention, let us define positions as angles measured counterclockwise from the positive horizontal axis. Then, the geodesic distance can be expressed as

$$\mathrm{dist}_g(x, y) = \min\{\mathrm{dist}_{\mathsf{c}}(x, y), \mathrm{dist}_{\mathsf{cc}}(x, y)\}, \quad x, y \in \mathbb{S}^1,$$

where $\mathrm{dist}_{\mathsf{c}}(x, y) = (x - y) \mod 2\pi$ is the clockwise distance and $\mathrm{dist}_{\mathsf{cc}}(x, y) = (y - x) \mod 2\pi$ is the counterclockwise distance. Here the clockwise distance between two angles is the path length from an angle to the other traveling clockwise, and $x \mod 2\pi$ is the remainder

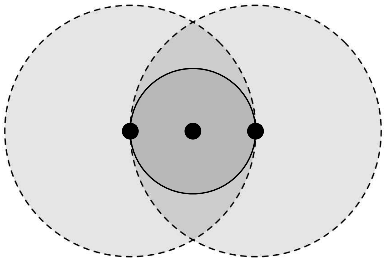


Figure 1.1 Open balls (dashed lines), a closed ball (solid line), and an open lune for the Euclidean distance on the plane.

of the division of x by 2π. On the sphere $\mathbb{S}^2$, the geodesic distance can be computed as follows. Given $x, y \in \mathbb{S}^2$, one considers the great circle determined by x and y. Then, the geodesic distance between x and y is exactly the length of the shortest arc in the great circle connecting x and y.

Cartesian product distance on $\mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$. Consider $\mathbb{R}^{d_1}$ endowed with an L^p-distance, $p \in [1, +\infty]$, and $\mathbb{S}^{d_2}$ endowed with the geodesic distance. Given $(x_1, y_1), (x_2, y_2) \in \mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$, their *Cartesian product distance* is given by $\operatorname{dist}_p(x_1, x_2) + \operatorname{dist}_g(y_1, y_2)$. Unless otherwise noted, it is understood that $\mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$ is endowed with the Cartesian product distance $(\operatorname{dist}_2, \operatorname{dist}_g)$.

Given a metric space (S, dist), the *open* and *closed balls* of center $x \in S$ and radius $\varepsilon \in \mathbb{R}_{>0}$ are defined by, respectively,

$$B(x, \varepsilon) = \{y \in S \mid \operatorname{dist}(x, y) < \varepsilon\},$$
$$\overline{B}(x, \varepsilon) = \{y \in S \mid \operatorname{dist}(x, y) \leq \varepsilon\}.$$

Consider a point $x \in X$ and a set $S \subset X$. A *neighborhood* of a point $x \in X$ is a subset of X that contains an open ball centered at x. A *neighborhood* of a set $Y \subset X$ is a subset of X that, for each point $y \in Y$, contains an open ball centered at y. The *open lune* associated to $x, y \in S$ is $B(x, \operatorname{dist}(x, y)) \cap B(y, \operatorname{dist}(x, y))$. These notions are illustrated in Figure 1.1 for the plane equipped with the Euclidean distance.

Given a metric space (S, dist), the distance between a point $x \in S$ and a set $W \subset S$ is the infimum of all distances between x and each of the points in W. Formally, we set

$$\operatorname{dist}(x, W) = \inf\{\operatorname{dist}(x, y) \mid y \in W\}.$$

The projection of a point $x \in S$ onto a set $W \subset S$ is the set-valued map

$\text{proj}_W : S \rightrightarrows W$ defined by

$$\text{proj}_W(x) = \{y \in W \mid \text{dist}(x,y) = \text{dist}(x,W)\}.$$

If W is a closed set, then $\text{proj}_W(x) \neq \emptyset$ for any $x \in S$. The *diameter* of a set is the maximum distance between any two points in the set; formally, we set $\text{diam}(S) = \sup\{\text{dist}(x,y) \mid x,y \in S\}$. With a slight abuse of notation, we often use $\text{diam}(P)$ to denote $\text{diam}(\{p_1,\dots,p_n\})$ for $P = (p_1,\dots,p_n)$.

1.1.3 Curves

A curve is the image of a continuous map $\gamma : [a,b] \to \mathbb{R}^d$. The map γ is called a *parameterization* of the curve. We usually identify a parameterization with the curve it defines. Without loss of generality, any curve can be given a parametrization with $a = 0$ and $b = 1$. A curve connects the two points p and q if $\gamma(0) = p$ and $\gamma(1) = q$. A curve $\gamma : [0,1] \to \mathbb{R}^d$ is *not self-intersecting* if γ is injective on $(0,1)$. A curve is *closed* if $\gamma(0) = \gamma(1)$.

A set $S \subset \mathbb{R}^d$ is *path connected* if any two points in S can be joined by a curve. A set $S \subset X$ is *simply connected* if it is path connected and any not self-intersecting closed curve can be continuously deformed to a point in the set; that is, for any injective continuous map $\gamma : [0,1] \to S$ that satisfies $\gamma(0) = \gamma(1)$, there exist $p \in S$ and a continuous map $H : [0,1] \times [0,1] \to S$ such that $H(t,0) = \gamma(t)$ and $H(t,1) = p$ for all $t \in [0,1]$. Informally, a simply connected set is a set that consists of a single piece and does not have any holes.

Next, consider a piecewise continuously differentiable curve $\gamma : [0,1] \to \mathbb{R}^d$; the *length* of γ is

$$\text{length}(\gamma) = \int_0^1 \|\dot{\gamma}(s)\|_2 ds,$$

and its *arc-length parameter* is

$$s_{\text{arc}}(s) = \int_0^s \|\dot{\gamma}(t)\|_2 dt.$$

Note that as the parameter t varies in $[0,1]$, the arc-length parameter $s_{\text{arc}}(t)$ varies in $[0, \text{length}(\gamma)]$. The *arc-length parameterization* of the curve is the map $\gamma_{\text{arc}} : [0, \text{length}(\gamma)] \to \mathbb{R}^d$ defined by the equation $\gamma_{\text{arc}}(s_{\text{arc}}(s)) = \gamma(s)$. With a slight abuse of notation, we will often drop the subindex arc and denote the arc-length parameterization by γ too.

For closed, not self-intersecting curves in the plane, we introduce the notion of signed and absolute curvatures as follows. Let $\gamma : [0, \text{length}(\gamma)] \to$

$\mathbb{R}^2$ be the counterclockwise arc-length parameterization of a curve. Assume γ is closed, not self-intersecting and twice continuously differentiable. Define the *tangent vector* $\gamma' : [0, \text{length}(\gamma)] \to \mathbb{R}^2$ by $\gamma'(s) = \frac{d\gamma}{ds}$. Note that the tangent vector has unit length, that is, $\|\gamma'(s)\|_2 = 1$ for all s. Additionally, define the *outward normal vector* $\mathrm{n}_{\text{out}} : [0, \text{length}(\gamma)] \to \mathbb{R}^2$ to be the unit-length vector that is point-wise orthogonal to the tangent vector and directed outside the set enclosed by the closed curve γ. With these notations, the *signed curvature* $\kappa_{\text{signed}} : [0, \text{length}(\gamma)] \to \mathbb{R}$ is defined by requiring that it satisfies

$$\gamma''(s) = -\kappa_{\text{signed}}(s)\, \mathrm{n}_{\text{out}}(s), \quad \text{and} \quad \mathrm{n}'_{\text{out}}(s) = \kappa_{\text{signed}}(s)\gamma'(s).$$

If the set enclosed by the closed curve γ is strictly convex, then the signed curvature of γ is strictly positive. In general, the *(absolute) curvature* $\kappa_{\text{abs}} : [0, \text{length}(\gamma)] \to \mathbb{R}_{\geq 0}$ and the *radius of curvature* $\rho : [0, \text{length}(\gamma)] \to \mathbb{R}_{\geq 0}$ of the curve γ are defined by, respectively,

$$\kappa_{\text{abs}}(s) = |\kappa_{\text{signed}}(s)|, \quad \text{and} \quad \rho(s) = |\kappa_{\text{signed}}(s)|^{-1}.$$

1.2 MATRIX THEORY

Here, we present basic notions and results about matrix theory, following the treatments in Horn and Johnson (1985) and Meyer (2001). We let $\mathbb{R}^{n\times m}$ and $\mathbb{C}^{n\times m}$ denote the set of $n\times m$ real and complex matrices. Given a real matrix A and a complex matrix U, we let A^T and U^* denote the transpose of A and the conjugate transpose matrix of U, respectively. We let I_n denote the $n \times n$ identity matrix. For a square matrix A, we write $A > 0$, resp. $A \geq 0$, if A is symmetric positive definite, resp. symmetric positive semidefinite. For a real matrix A, we let $\text{kernel}(A)$ and $\text{rank}(A)$ denote the kernel and rank of A, respectively. Given a vector v, we let $\text{diag}(v)$ denote the square matrix whose diagonal elements are equal to the component v and whose off-diagonal elements are zero.

1.2.1 Matrix sets

A matrix $A \in \mathbb{R}^{n\times n}$ with entries a_{ij}, $i, j \in \{1, \dots, n\}$, is

(i) *Orthogonal* if $AA^T = I_n$, and is *special orthogonal* if it is orthogonal with $\det(A) = +1$. The set of orthogonal matrices is a group.[1]

[1] A set G with a binary operation, denoted by $G \times G \ni (a, b) \mapsto a \star b \in G$, is a *group* if: (i) $a \star (b \star c) = (a \star b) \star c$ for all $a, b, c \in G$ (associativity property); (ii) there exists $e \in G$ such that $a \star e = e \star a = a$ for all $a \in G$ (existence of an identity element); and (iii) there exists $a^{-1} \in G$ such that $a \star a^{-1} = a^{-1} \star a = e$ for all $a \in G$ (existence of inverse elements).

(ii) *Nonnegative* (resp., *positive*) if all its entries are nonnegative (resp., positive).

(iii) *Row-stochastic* (or *stochastic* for brevity) if it is nonnegative and $\sum_{j=1}^{n} a_{ij} = 1$, for all $i \in \{1, \dots, n\}$; in other words, A is row-stochastic if

$$A\mathbf{1}_n = \mathbf{1}_n.$$

(iv) *Column-stochastic* if it is nonnegative and $\sum_{i=1}^{n} a_{ij} = 1$, for all $j \in \{1, \dots, n\}$.

(v) *Doubly stochastic* if A is row-stochastic and column-stochastic.

(vi) *Normal* if $A^T A = AA^T$.

(vii) A *permutation matrix* if A has precisely one entry equal to one in each row, one entry equal to one in each column, and all other entries equal to zero. The set of permutation matrices is a group.

The scalars $\mu_1, \dots, \mu_k$ are *convex combination coefficients* if $\mu_i \geq 0$, for $i \in \{1, \dots, k\}$, and $\sum_{i=1}^{k} \mu_i = 1$. (Each row of a row-stochastic matrix contains convex combination coefficients.) A *convex combination* of vectors is a linear combination of the vectors with convex combination coefficients. A subset U of a vector space V is *convex* if the convex combination of any two elements of U takes value in U. For example, the set of stochastic matrices and the set of doubly stochastic matrices are convex.

Theorem 1.1 (Birkhoff–von Neumann). *A square matrix is doubly stochastic if and only if it is a convex combination of permutation matrices.*

Next, we review two families of relevant matrices with useful properties. *Toeplitz matrices* are square matrices with equal entries along each diagonal parallel to the main diagonal. In other words, a Toeplitz matrix is a matrix of the form

$$\begin{bmatrix}
t_0 & t_1 & \ddots & \ddots & \ddots & t_{n-2} & t_{n-1} \\
t_{-1} & t_0 & t_1 & \ddots & \ddots & \ddots & t_{n-2} \\
\ddots & t_{-1} & t_0 & t_1 & \ddots & \ddots & \ddots \\
\ddots & \ddots & t_{-1} & t_0 & t_1 & \ddots & \ddots \\
\ddots & \ddots & \ddots & t_{-1} & t_0 & t_1 & \ddots \\
t_{-n+2} & \ddots & \ddots & \ddots & t_{-1} & t_0 & t_1 \\
t_{-n+1} & t_{-n+2} & \ddots & \ddots & \ddots & t_{-1} & t_0
\end{bmatrix}.$$

An $n \times n$ Toeplitz matrix is determined by its first row and column, and hence by $2n-1$ scalars.

Circulant matrices are square Toeplitz matrices where each two subsequent row vectors v_i and v_{i+1} have the following two properties: the last entry of v_i is the first entry of v_{i+1} and the first $(n-1)$ entries of v_i are the second $(n-1)$ entries of v_{i+1}. In other words, a circulant matrix is a matrix of the form

$$\begin{bmatrix} c_0 & c_1 & \ddots & \ddots & \ddots & c_{n-2} & c_{n-1} \\ c_{n-1} & c_0 & c_1 & \ddots & \ddots & \ddots & c_{n-2} \\ \ddots & c_{n-1} & c_0 & c_1 & \ddots & \ddots & \ddots \\ \ddots & \ddots & c_{n-1} & c_0 & c_1 & \ddots & \ddots \\ \ddots & \ddots & \ddots & c_{n-1} & c_0 & c_1 & \ddots \\ c_2 & \ddots & \ddots & \ddots & c_{n-1} & c_0 & c_1 \\ c_1 & c_2 & \ddots & \ddots & \ddots & c_{n-1} & c_0 \end{bmatrix},$$

and, therefore, it is determined by its first row.

1.2.2 Eigenvalues, singular values, and induced norms

We require the reader to be familiar with the notion of eigenvalue and of simple eigenvalue, that is, an eigenvalue with algebraic and geometric multiplicity[2] equal to 1. The set of eigenvalues of a matrix $A \in \mathbb{R}^{n\times n}$ is called its *spectrum* and is denoted by $\operatorname{spec}(A) \subset \mathbb{C}$. The *singular values* of the matrix $A \in \mathbb{R}^{n\times n}$ are the positive square roots of the eigenvalues of $A^T A$.

We begin with a well-known property of the spectrum of a matrix.

Theorem 1.2 (Geršgorin disks). *Let A be an $n \times n$ matrix. Then*

$$\operatorname{spec}(A) \subset \bigcup_{i\in\{1,\dots,n\}} \Big\{ z \in \mathbb{C} \mid \|z - a_{ii}\|_{\mathbb{C}} \leq \sum_{j=1, j\neq i}^{n} |a_{ij}| \Big\}.$$

Next, we review a few facts about normal matrices, their eigenvectors and their singular values.

[2] The algebraic multiplicity of an eigenvalue is the multiplicity of the corresponding root of the characteristic equation. The geometric multiplicity of an eigenvalue is the number of linearly independent eigenvectors corresponding to the eigenvalue. The algebraic multiplicity is greater than or equal to the geometric multiplicity.

Lemma 1.3 (Normal matrices). *For a matrix $A \in \mathbb{R}^{n\times n}$, the following statements are equivalent:*

(i) A is normal;

(ii) A has a complete orthonormal set of eigenvectors; and

(iii) A is unitarily similar to a diagonal matrix, that is, there exists a unitary[3] matrix U such that U^*AU is diagonal.

Lemma 1.4 (Singular values of a normal matrix). *If a normal matrix has eigenvalues $\{\lambda_1, \dots, \lambda_n\}$, then its singular values are $\{|\lambda_1|, \dots, |\lambda_n|\}$.*

It is well known that real symmetric matrices are normal, are diagonalizable by orthogonal matrices, and have real eigenvalues. Additionally, circulant matrices are normal.

We conclude by defining the notion of induced norm of a matrix. For $p \in \mathbb{N}\cup\{\infty\}$, the *p-induced norm* of $A \in \mathbb{R}^{n\times n}$ is

$$\|A\|_p = \max\{\|Ax\|_p \mid \|x\|_p = 1\}.$$

One can see that

$$\|A\|_1 = \max_{j\in\{1,\dots,n\}} \sum_{i=1}^n |a_{ij}|, \quad \|A\|_\infty = \max_{i\in\{1,\dots,n\}} \sum_{j=1}^n |a_{ij}|,$$
$$\|A\|_2 = \max\{\sigma \mid \sigma \text{ is a singular value of } A\}.$$

1.2.3 Spectral radius and convergent matrices

The *spectral radius* of a matrix $A \in \mathbb{R}^{n\times n}$ is

$$\rho(A) = \max\{\|\lambda\|_{\mathbb{C}} \mid \lambda \in \operatorname{spec}(A)\}.$$

In other words, $\rho(A)$ is the radius of the smallest disk centered at the origin that contains the spectrum of A.

Lemma 1.5 (Induced norms and spectral radius). *For any square matrix A and in any norm $p \in \mathbb{N}\cup\{\infty\}$, $\rho(A) \leq \|A\|_p$.*

We will often deal with matrices with an eigenvalue equal to 1 and all other eigenvalues strictly inside the unit disk. Accordingly, we generalize the notion of spectral radius as follows. For a square matrix A with $\rho(A) = 1$, we define the *essential spectral radius*

$$\rho_{\text{ess}}(A) = \max\{\|\lambda\|_{\mathbb{C}} \mid \lambda \in \operatorname{spec}(A) \setminus \{1\}\}. \tag{1.2.1}$$

[3] A complex matrix $U \in \mathbb{C}^{n\times n}$ is *unitary* if $U^{-1} = U^*$.

Next, we will consider matrices with useful convergence properties.

Definition 1.6 (Convergent and semi-convergent matrices). A matrix $A \in \mathbb{R}^{n \times n}$ is

(i) *semi-convergent* if $\lim_{\ell \to +\infty} A^\ell$ exists; and

(ii) *convergent* if it is semi-convergent and $\lim_{\ell \to +\infty} A^\ell = 0$. •

These two notions are characterized as follows.

Lemma 1.7 (Convergent and semi-convergent matrices). *The square matrix A is convergent if and only if $\rho(A) < 1$. Furthermore, A is semi-convergent if and only if the following three properties hold:*

(i) $\rho(A) \leq 1$*;*

(ii) $\rho_{\mathrm{ess}}(A) < 1$*, that is,* 1 *is an eigenvalue and* 1 *is the only eigenvalue on the unit circle; and*

(iii) the eigenvalue 1 *is* semisimple*, that is, it has equal algebraic and geometric multiplicity (possibly larger than one).*

In other words, A is semi-convergent if and only if there exists a nonsingular matrix T such that

$$A = T \begin{bmatrix} I_k & 0 \\ 0 & B \end{bmatrix} T^{-1},$$

where $B \in \mathbb{R}^{(n-k) \times (n-k)}$ is convergent, that is, $\rho(B) < 1$. With this notation, we have $\rho_{\mathrm{ess}}(A) = \rho(B)$ and the algebraic and geometric multiplicity of the eigenvalue 1 is k.

1.2.4 Perron–Frobenius theory

Positive and nonnegative matrices have useful spectral properties. In what follows, the first theorem amounts to the original Perron's Theorem for positive matrices and the following theorems are the extension due to Frobenius for certain nonnegative matrices. We refer to (Horn and Johnson, 1985, Chapter 8) for a detailed treatment.

Theorem 1.8 (Perron-Frobenius for positive matrices). *If the square matrix A is positive, then*

(i) $\rho(A) > 0$*;*

(ii) $\rho(A)$ *is an eigenvalue, it is simple, and* $\rho(A)$ *is strictly larger than the magnitude of any other eigenvalue; and*

(iii) $\rho(A)$ *has an eigenvector with positive components.*

Requiring the matrix to be strictly positive is a key assumption that limits the applicability of this theorem. It turns out that it is possible to obtain the same results of the theorem under weaker assumptions.

Definition 1.9 (Irreducible matrix). A nonnegative matrix $A \in \mathbb{R}^{n \times n}$ is *irreducible* if, for any nontrivial partition $J \cup K$ of the index set $\{1, \dots, n\}$, there exist $j \in J$ and $k \in K$ such that $a_{jk} \neq 0$.

Remark 1.10 (Properties of irreducible matrices). An equivalent definition of irreducibility is given as follows. A matrix $A \in \mathbb{R}^{n \times n}$ is *irreducible* if it is not reducible, and is *reducible* if either:

(i) $n = 1$ and $A = 0$; or

(ii) there exists a permutation matrix $P \in \mathbb{R}^{n \times n}$ and a number $r \in \{1, \dots, n-1\}$ such that $P^T A P$ is block upper triangular with diagonal blocks of dimensions $r \times r$ and $(n-r) \times (n-r)$.

It is an immediate consequence that the property of irreducibility depends upon only the patterns of zeros and nonzero elements of the matrix. •

We can now weaken the assumption in Theorem 1.8 and obtain a comparable, but weaker, result for irreducible matrices.

Theorem 1.11 (Perron–Frobenius for irreducible matrices). *If the nonnegative square matrix A is irreducible, then*

(i) $\rho(A) > 0$*;*

(ii) $\rho(A)$ *is an eigenvalue, and it is simple; and*

(iii) $\rho(A)$ *has an eigenvector with positive components.*

In general, the spectral radius of a nonnegative irreducible matrix does not need to be the only eigenvalue of maximum magnitude. For example, the matrix $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ has eigenvalues $\{1, -1\}$. In other words, irreducible matrices do indeed have weaker spectral properties than positive matrices. Therefore, it remains unclear which nonnegative matrices have the same properties as those stated for positive matrices in Theorem 1.8.

Definition 1.12 (Primitive matrix). A nonnegative square matrix A is *primitive* if there exists $k \in \mathbb{N}$ such that A^k is positive. •

It is easy to see that if a nonnegative square matrix is primitive, then it is irreducible. In later sections we will provide a graph-theoretical characterization of primitive matrices; for now, we are finally in a position to sharpen the results of Theorem 1.11.

Theorem 1.13 (Perron–Frobenius for primitive matrices). *If the nonnegative square matrix A is primitive, then*

(i) $\rho(A) > 0$*;*

(ii) $\rho(A)$ *is an eigenvalue, it is simple, and* $\rho(A)$ *is strictly larger than the magnitude of any other eigenvalue; and*

(iii) $\rho(A)$ *has an eigenvector with positive components.*

We conclude this section by noting the following convergence property that is an immediate corollary to Lemma 1.7 and to Theorem 1.13.

Corollary 1.14. *If the nonnegative square matrix A is primitive, then the matrix $\rho(A)^{-1}A$ is semi-convergent.*

1.3 DYNAMICAL SYSTEMS AND STABILITY THEORY

In this section, we introduce some basic concepts about dynamical and control systems; see, for example Sontag (1998) and Khalil (2002). We discuss stability and attractivity notions as well as the invariance principle. We conclude with a treatment of set-valued systems and time-dependent systems.

1.3.1 State machines and dynamical systems

Here, we introduce three classes of dynamical and control systems: (i) state machines or discrete-time discrete-space dynamical systems; (ii) discrete-time continuous-space control systems; and (iii) continuous-time continuous-space control systems.

We begin with our specific definition of state machine. A *(deterministic, finite) state machine* is a tuple (X, U, X_0, f), where X is a finite set called the *state space*, U is a finite set called the *input space*, $X_0 \subset X$ is the set of *allowable initial states*, and $f : X \times U \to X$ is the *evolution map*. Given an input sequence $u : \mathbb{Z}_{\geq 0} \to U$, the state machine evolution $x : \mathbb{Z}_{\geq 0} \to X$ starting from $x(0) \in X_0$ is given by

$$x(\ell + 1) = f(x(\ell), u(\ell)), \quad \ell \in \mathbb{Z}_{\geq 0}.$$

We will often refer to a state machine as a *processor*. Note that, in a state machine, both the state and the input spaces are finite or *discrete*. Often times, we will find it useful to consider systems that evolve in continuous space and that are time dependent. Let us then provide two additional definitions in the following paragraphs.

A *(time-dependent) discrete-time continuous-space control system* is a tuple (X, U, X_0, f), where X is a d-dimensional space chosen among $\mathbb{R}^d$, $\mathbb{S}^d$, and the Cartesian products $\mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$, for some $d_1 + d_2 = d$, U is a compact subset of $\mathbb{R}^m$ containing $\mathbf{0}_m$, $X_0 \subset X$, and $f : \mathbb{Z}_{\geq 0} \times X \times U \to X$ is a continuous map. As before, the individual objects X, U, X_0, and f are termed the *state space*, *input space*, *allowable initial states*, and *evolution map*, respectively. Given an input sequence $u : \mathbb{Z}_{\geq 0} \to U$, the evolution $x : \mathbb{Z}_{\geq 0} \to X$ of the dynamical system starting from $x(0) \in X_0$ is given by

$$x(\ell + 1) = f(\ell, x(\ell), u(\ell)), \quad \ell \in \mathbb{Z}_{\geq 0}.$$

A *(time-dependent) continuous-time continuous-space control system* is a tuple (X, U, X_0, f), where X is a d-dimensional space chosen among $\mathbb{R}^d$, $\mathbb{S}^d$, and the Cartesian products $\mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$, for some $d_1 + d_2 = d$, U is a compact subset of $\mathbb{R}^m$ containing $\mathbf{0}_m$, $X_0 \subset X$, and $f : \mathbb{R}_{\geq 0} \times X \times U \to TX$ is a continuously differentiable map. The individual objects X, U, X_0, and f are termed the *state space*, *input space*, *allowable initial states*, and *control vector field*, respectively. Given an input function $u : \mathbb{R}_{\geq 0} \to U$, the evolution $x : \mathbb{R}_{\geq 0} \to X$ of the dynamical system starting from $x(0) \in X_0$ is given by

$$\dot{x}(t) = f(t, x(t), u(t)), \quad t \in \mathbb{R}_{\geq 0}.$$

We often consider the case when the control vector field can be written as $f(t, x, u) = f_0(t, x) + \sum_{a=1}^{m} f_a(t, x)\, u_a$, for some continuously differentiable maps $f_0, f_1, \dots, f_m : \mathbb{R}_{\geq 0} \times X \to TX$. Each of these individual maps is called a *(time-dependent) vector field*, and f is said to be a *control-affine* vector field. The control vector field f is *driftless* if $f(t, x, \mathbf{0}_m) = 0$ for all $x \in X$ and $t \in \mathbb{R}_{\geq 0}$.

Finally, the term *dynamical system* denotes a control system that is not subject to any external control action; this terminology is applicable both in discrete and continuous time. Furthermore, we will sometimes neglect to define a specific set of allowable initial states; in this case we mean that any point in the state space is allowable as initial condition.

1.3.2 Stability and attractivity notions

In this section, we consider a continuous-space dynamical system (X, f). We first consider the discrete-time case and later we briefly present the analogous continuous-time case. We study dynamical systems that are *time-invariant*. In discrete time, a time-invariant system is simply described by an evolution map of the form $f : X \to X$.

Definition 1.15 (Equilibrium point). A point $x_* \in X$ is an *equilibrium point* for the time-invariant dynamical system (X, f) if the constant curve $x : \mathbb{Z}_{\geq 0} \to X$, defined by $x(\ell) = x_*$ for all $\ell \in \mathbb{Z}_{\geq 0}$, is an evolution of the system. •

It can immediately be seen that a point x_* is an equilibrium point if and only if $f(x_*) = x_*$. We denote the set of equilibrium points of the dynamical system by $\operatorname{Equil}(X, f)$.

Definition 1.16 (Trajectories and sets). Let (X, f) be a time-invariant dynamical system and let W be a subset of X. Then:

(i) The set W is *positively invariant* for (X, f) if each evolution with initial condition in W remains in W for all subsequent times.

(ii) A trajectory $x : \mathbb{Z}_{\geq 0} \to X$ *approaches* a set $W \subset X$ if, for every neighborhood Y of W, there exists a time $\ell_0 > 0$ such that $x(\ell)$ takes values in Y for all subsequent times $\ell \geq \ell_0$. In such a case, we write $x(\ell) \to W$ as $\ell \to +\infty$. •

In formal terms, W is positively invariant if $x(0) \in W$ implies $x(\ell) \in W$ for all $\ell \in \mathbb{Z}_{\geq 0}$, where $x : \mathbb{Z}_{\geq 0} \to X$ is the evolution of (X, f) starting from $x(0)$.

Definition 1.17 (Stability and attractivity). For a time-invariant dynamical system (X, f), a set S is:

(i) *stable* if, for any neighborhood Y of S, there exists a neighborhood W of S such that every evolution of (X, f) with initial condition in W remains in Y for all subsequent times;

(ii) *unstable* if it is not stable;

(iii) *locally attractive* if there exists a neighborhood Y of S such that every evolution with initial condition in Y approaches the set S; and

(iv) *locally asymptotically stable* if it is stable and locally attractive.

Additionally, the set S is *globally attractive* if every evolution of the dynamical system approaches it and it is *globally asymptotically stable* if it is stable and globally attractive. •

Remark 1.18 (Continuous-time dynamical systems). It is straightforward to extend the previous definitions to the setting of continuous-time continuous-space dynamical systems. These notions are illustrated in Figure 1.2. •

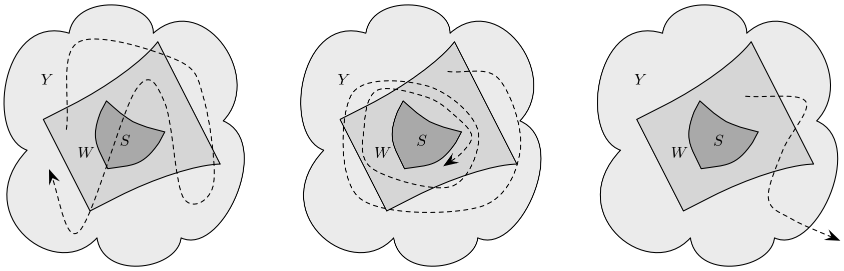


Figure 1.2 Illustrations of stability, asymptotic stability, and instability.

1.3.3 Invariance principles

Before discussing various versions of the invariance principle, we begin with a useful notion. Given a discrete-time time-invariant continuous-space dynamical system (X, f) and a set $W \subset X$, a function $V : X \to \mathbb{R}$ is *non-increasing along f in W* if $V(f(x)) \leq V(x)$ for all $x \in W$. (Such functions are often referred to as *Lyapunov functions.*) In other words, if a function V is non-increasing along f, then the composite function $\ell \mapsto V(y(\ell))$ is non-increasing for each evolution y of the dynamical system (X, f). The following theorem exploits this fact to establish useful properties of the evolutions of (X, f).

Theorem 1.19 (LaSalle Invariance Principle for discrete-time dynamical systems). *Let (X, f) be a discrete-time time-independent dynamical system. Assume that:*

(i) there exists a closed set $W \subset X$ that is positively invariant for (X, f);

(ii) there exists a function $V : X \to \mathbb{R}$ that is non-increasing along f on W;

(iii) all evolutions of (X, f) with initial conditions in W are bounded; and

(iv) f and V are continuous on W.

Then each evolution with initial condition in W approaches a set of the form $V^{-1}(c) \cap S$, where c is a real constant and S is the largest positively invariant set contained in $\{w \in W \mid V(f(w)) = V(w)\}$.

We refer to Section 1.8.1 for a discussion about the proof of this result. Next, we present the continuous-time version of the invariance principle. In other words, we now assume that (X, f) is a continuous-time time-invariant continuous-space dynamical system.

We begin by revisiting the notion of non-increasing function. Given a continuously differentiable function $V : X \to \mathbb{R}$, the *Lie derivative* of V along f, denoted by $\mathcal{L}_f V : X \to \mathbb{R}$, is defined by

$$\mathcal{L}_f V(x) = \frac{\mathrm{d}}{\mathrm{d}t} V(\gamma(t))\Big|_{t=0},$$

where the trajectory $\gamma :]-\varepsilon, \varepsilon[\to X$ satisfies $\dot{\gamma}(t) = f(\gamma(t))$ and $\gamma(0) = x$. If $X = \mathbb{R}^d$, then we can write x in components $(x_1, \dots, x_d)$ and we can give the following explicit formula for the Lie derivative:

$$\mathcal{L}_f V(x) = \sum_{i=1}^{d} \frac{\partial V}{\partial x_i}(x) f_i(x).$$

Similar formulas can be obtained for more general state spaces. Note that, given a set $W \subset X$, a function $V : X \to \mathbb{R}$ is non-increasing along f in W if $\mathcal{L}_f V(x) \leq 0$ for all $x \in W$.

Finally, we state the invariance principle for continuous-time systems.

Theorem 1.20 (LaSalle Invariance Principle for continuous-time dynamical systems). *Let (X, f) be a continuous-time time-independent dynamical system. Assume that:*

(i) there exists a closed set $W \subset X$ that is positively invariant for (X, f);

(ii) there exists a function $V : X \to \mathbb{R}$ that is non-increasing along f on W;

(iii) all evolutions of (X, f) with initial conditions in W are bounded; and

(iv) f and V are continuously differentiable[4] on W.

Then, each evolution with initial condition in W approaches a set of the form $V^{-1}(c) \cap S$, where c is a real constant and S is the largest positively invariant set contained in $\{w \in W \mid \mathcal{L}_f V(w) = 0\}$.

[4] It suffices that f be locally Lipschitz and V be continuously differentiable; see Cortés (2008a).

1.3.4 Notions and results for set-valued systems

Next, we focus on a more sophisticated version of the LaSalle Invariance Principle for more general dynamical systems, that is, dynamical systems described by set-valued maps that allow for non-deterministic evolutions. To do so, we need to present numerous notions, including set-valued dynamical systems, closedness properties, and weak positive invariance.

Specifically, a *discrete-time continuous-space set-valued dynamical system* (in short, *set-valued dynamical system*) is determined by a tuple (X, X_0, T), where X is a d-dimensional space chosen among $\mathbb{R}^d$, $\mathbb{S}^d$, and the Cartesian products $\mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$, for some $d_1 + d_2 = d$, $X_0 \subset X$, and $T : X \rightrightarrows X$ is a set-valued map. We assume that T assigns to each point $x \in X$ a nonempty set $T(x) \subset X$. The individual objects X, X_0, and T are termed the *state space*, *allowable initial states*, and *evolution map*, respectively. An *evolution* of the dynamical system (X, X_0, T) is any trajectory $x : \mathbb{Z}_{\geq 0} \to X$ satisfying

$$x(\ell + 1) \in T(x(\ell)), \quad \ell \in \mathbb{Z}_{\geq 0}.$$

Figure 1.3 illustrates this notion. In particular, a (time-invariant) discrete-time continuous-space dynamical system (X, X_0, f) can be seen as a discrete-time continuous-space set-valued dynamical system (X, X_0, T), where the evolution set-valued map is just the singleton-valued map $x \mapsto T(x) = \{f(x)\}$.

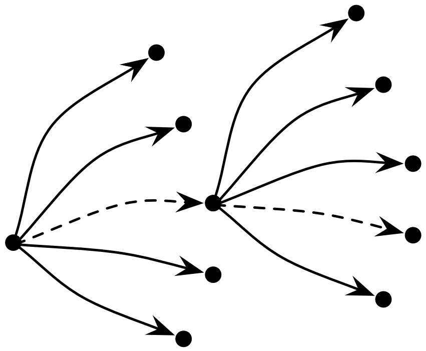


Figure 1.3 A discrete-time continuous-space set-valued dynamical system. A sample evolution is shown dashed.

Next, we introduce a notion of continuity for set-valued maps. The evolution map T is said to be *closed at* $x \in X$ if, for any sequences $\{x_k \mid k \in \mathbb{Z}_{\geq 0}\}$

and $\{y_k \mid k \in \mathbb{Z}_{\geq 0}\}$ such that

$$\lim_{k \to +\infty} x_k = x, \quad \lim_{k \to +\infty} y_k = y, \quad \text{and} \quad y_k \in T(x_k),$$

it holds that $y \in T(x)$. The evolution set-valued map T is *closed at* $W \subset X$ if for any $x \in W$, T is closed at x. Note that a continuous map $f : X \to X$ is closed when viewed as a singleton-valued map.

(i) A set $C \subset X$ is *weakly positively invariant with respect to* T if, for any $x \in C$, there exists $y \in C$ such that $y \in T(x)$.

(ii) A set $C \subset X$ is *strongly positively invariant with respect to* T if $T(x) \subset C$ for any $x \in C$.

A point x_0 is said to be a *fixed point of* T if $x_0 \in T(x_0)$. A continuous function $V : X \to \mathbb{R}$ is *non-increasing along* T *in* $W \subset X$ if $V(y) \leq V(x)$ for all $x \in W$ and $y \in T(x)$.

We finally state and prove a general version of the invariance principle, whose proof is presented in Section 1.8.1.

Theorem 1.21 (LaSalle Invariance Principle for set-valued discrete-time dynamical systems). *Let (X, X_0, T) be a discrete-time set-valued dynamical system. Assume that:*

(i) there exists a closed set $W \subset X$ that is strongly positively invariant for (X, X_0, T);

(ii) there exists a function $V : X \to \mathbb{R}$ that is non-increasing along T on W;

(iii) all evolutions of (X, X_0, T) with initial conditions in W are bounded; and

(iv) T is nonempty and closed at W and V is continuous on W.

Then, each evolutions with initial condition in W approaches a set of the form $V^{-1}(c) \cap S$, where c is a real constant and S is the largest weakly positively invariant set contained in $\{w \in W \mid \exists w' \in T(w)$ such that $V(w') = V(w)\}$.

1.3.5 Notions and results for time-dependent systems

In this final subsection, we consider time-dependent discrete-time dynamical systems and discuss *uniform* stability and convergence notions. We begin with some uniform boundedness, stability, and attractivity definitions.

In what follows, given a time-dependent discrete-time dynamical system (X, X_0, f), an evolution with initial condition in W at time $\ell_0 \in \mathbb{Z}_{\geq 0}$ is a trajectory $x : [\ell_0, +\infty[\rightarrow X$ of the dynamical system (X, X_0, f) defined by the initial condition $x(\ell_0) = x_0$, for some $x_0 \in W$. In other words, for time-dependent systems we will often consider trajectories that begin at time ℓ_0 not necessarily equal to zero.

Definition 1.22 (Uniformly bounded evolutions). A time-dependent discrete-time dynamical system (X, X_0, f) has *uniformly bounded evolutions* if, given any bounded set Y, there exists a bounded set W such that every evolution with initial condition in Y at any time $\ell_0 \in \mathbb{Z}_{\geq 0}$ remains in W for all subsequent times $\ell \geq \ell_0$. •

Definition 1.23 (Uniform stability and attractivity notions). For a time-dependent discrete-time dynamical system (X, X_0, f), the set S is:

(i) *uniformly stable* if, for any neighborhood Y of S, there exists a neighborhood W of S such that every evolution with initial condition in W at any time $\ell_0 \in \mathbb{Z}_{\geq 0}$ remains in Y for all subsequent times $\ell \geq \ell_0$;

(ii) *uniformly locally attractive* if there exists a neighborhood Y of S such that every evolution with initial condition in Y at any time ℓ_0 approaches the set S in the following *time-uniform* manner:

> for all $\ell_0 \in \mathbb{Z}_{\geq 0}$, for all $x_0 \in Y$, and for all neighborhoods W of S, there exists a single $\tau_0 \in \mathbb{Z}_{\geq 0}$ such that the evolution $x : [\ell_0, +\infty[\rightarrow X$ defined by $x(\ell_0) = x_0$ takes value in W for all times $\ell \geq \ell_0 + \tau_0$; and

(iii) *uniformly locally asymptotically stable* if it is uniformly stable and uniformly locally attractive.

Additionally, the set S is *uniformly globally attractive* if every evolution of the dynamical system approaches the set in a time-uniform manner, and it is *uniformly globally asymptotically stable* if it is uniformly stable and uniformly globally attractive. •

With the same notation in the definition, the set S is *(non-uniformly) locally attractive* if for all $\ell_0 \in \mathbb{Z}_{\geq 0}$, $x_0 \in Y$, and neighborhoods W of S, the evolution $x : [\ell_0, +\infty[\rightarrow X$ defined by $x(\ell_0) = x_0$, takes value in W for all times $\ell \geq \ell_0 + \tau_0(\ell_0)$, for some $\tau_0(\ell_0) \in \mathbb{Z}_{\geq 0}$.

To establish uniform stability and attractivity results we will overapproximate the evolution of the time-dependent dynamical system by considering the larger set of evolutions of an appropriate set-valued dynamical system. Given a time-dependent evolution map $f : \mathbb{Z}_{\geq 0} \times X \rightarrow X$, define a set-valued *overapproximation map* $T_f : X \rightrightarrows X$ by

$$T_f(x) = \{f(\ell, x) \mid \ell \in \mathbb{Z}_{\geq 0}\}.$$

With this notion we can state a useful result, whose proof is left to the reader as an exercise.

Lemma 1.24 (Overapproximation Lemma). *Consider a discrete-time time-dependent dynamical system* (X, X_0, f)*:*

(i) If $x : [\ell_0, +\infty[\to X$ *is an evolution of the dynamical system* (X, f)*, then* $y : \mathbb{Z}_{\geq 0} \to X$ *defined by* $y(\ell) = x(\ell + \ell_0)$ *is an evolution of the set-valued* overapproximation system (X, T_f)*.*

(ii) If the set S *is locally attractive for the set-valued overapproximation system* (X, T_f)*, then it is uniformly locally attractive for* (X, f)*.*

In other words, every evolution of the time-dependent dynamical system from any initial time is an evolution of the set-valued overapproximation system and, therefore, the set of trajectories of the set-valued overapproximation system contains the set of trajectories of the original time-dependent system. Uniform attractivity is a consequence of attractivity for the time-invariant set-valued overapproximation.

1.4 GRAPH THEORY

Here we present basic definitions about graph theory, following the treatments in the literature; see, for example Biggs (1994), Godsil and Royle (2001), and Diestel (2005).

A *directed graph*—in short, *digraph*—of order n is a pair $G = (V, E)$, where V is a set with n elements called *vertices* (or *nodes*) and E is a set of ordered pair of vertices called *edges*. In other words, $E \subseteq V \times V$. We call V and E the *vertex set* and *edge set*, respectively. When convenient, we let $V(G)$ and $E(G)$ denote the vertices and edges of G, respectively. For $u, v \in V$, the ordered pair (u, v) denotes an edge *from* u *to* v.

An *undirected graph*—in short, *graph*—consists of a vertex set V and of a set E of unordered pairs of vertices. For $u, v \in V$ and $u \neq v$, the set $\{u, v\}$ denotes an unordered edge. A digraph is *undirected* if $(v, u) \in E$ anytime $(u, v) \in E$. It is possible and convenient to identify an undirected digraph with the corresponding graph; vice versa, the *directed version* of a graph (V, E) is the digraph (V', E') with the property that $(u, v) \in E'$ if and only if $\{u, v\} \in E$. In what follows, our convention is to allow self-loops in both graphs and digraphs.

A digraph (V', E') is a *subgraph* of a digraph (V, E) if $V' \subset V$ and $E' \subset E$; additionally, a digraph (V', E') is a *spanning subgraph* if it is a subgraph and $V' = V$. The subgraph of (V, E) *induced by* $V' \subset V$ is the digraph (V', E'), where E' contains all edges in E between two vertices in V'. For two digraphs $G = (V, E)$ and $G' = (V', E')$, the *intersection* and *union* of G and G' are defined by

$$G \cap G' = (V \cap V', E \cap E'),$$
$$G \cup G' = (V \cup V', E \cup E').$$

Analogous definitions may be given for graphs.

In a digraph G with an edge $(u, v) \in E$, u is called an *in-neighbor* of v, and v is called an *out-neighbor* of u. We let $\mathcal{N}_G^{\text{in}}(v)$ (resp., $\mathcal{N}_G^{\text{out}}(v)$) denote the set of in-neighbors, (resp. the set of out-neighbors) of v in the digraph G. We will drop the subscript when the graph G is clear from the context. The *in-degree* and *out-degree* of v are the cardinality of $\mathcal{N}^{\text{in}}(v)$ and $\mathcal{N}^{\text{out}}(v)$, respectively. A digraph is *topologically balanced* if each vertex has the same in- and out-degrees (even if distinct vertices have distinct degrees). Likewise, in an undirected graph G, the vertices u and v are *neighbors* if $\{u, v\}$ is an undirected edge. We let $\mathcal{N}_G(v)$ denote the set of neighbors of v in the undirected graph G. As in the directed case, we will drop the subscript when the graph G is clear from the context. The *degree* of v is the cardinality of $\mathcal{N}(v)$.

Remark 1.25 (Additional notions). For a digraph $G = (V, E)$, the *reverse digraph* $\text{rev}(G)$ has vertex set V and edge set $\text{rev}(E)$ composed of all edges in E with reversed direction. A digraph $G = (V, E)$ is *complete* if $E = V \times V$. A *clique* (V', E') of a digraph (V, E) is a subgraph of (V, E) which is complete, that is, such that $E' = V' \times V'$. Note that a clique is fully determined by its set of vertices, and hence there is no loss of precision in denoting it by V'. A *maximal clique V' of an edge* of a digraph is a clique of the digraph with the following two properties: it contains the edge, and any other subgraph of the digraph that strictly contains $(V', V' \times V')$ is not a clique. •

1.4.1 Connectivity notions

Let us now review some basic connectivity notions for digraphs and graphs. We begin with the setting of undirected graphs because of its simplicity.

A *path* in a graph is an ordered sequence of vertices such that any pair of consecutive vertices in the sequence is an edge of the graph. A graph is *connected* if there exists a path between any two vertices. If a graph is not connected, then it is composed of multiple *connected components*, that is, multiple connected subgraphs. A path is *simple* if no vertices appear more than once in it, except possibly for initial and final vertex. A *cycle* is a simple path that starts and ends at the same vertex. A graph is *acyclic* if it contains no cycles. A connected acyclic graph is a *tree*. A *forest* is a graph that can be written as the disjoint union of trees. Trees have interesting properties: for example, $G = (V, E)$ is a tree if and only if G is connected and $|E| = |V| - 1$. Alternatively, $G = (V, E)$ is a tree if and only if G is acyclic and $|E| = |V| - 1$. Figure 1.4 illustrates these notions.

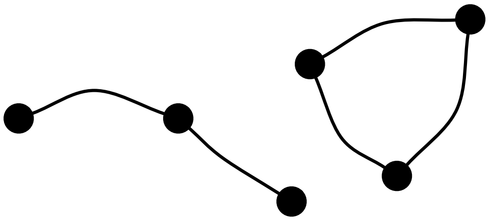


Figure 1.4 An illustration of connectivity notions on a graph. The graph has two connected components. The leftmost connected component is a tree, while the rightmost connected component is a cycle.

Next, we generalize these notions to the case of digraphs. A *directed path* in a digraph is an ordered sequence of vertices such that any ordered pair of vertices appearing consecutively in the sequence is an edge of the digraph. A *cycle* in a digraph is a directed path that starts and ends at the same vertex and that contains no repeated vertex except for the initial and the final vertex. A digraph is *acyclic* if it contains no cycles. In an acyclic graph, every vertex of in-degree 0 is named a *source*, and every vertex of out-degree 0 is named a *sink*. Every acyclic digraph has at least one source and at least one sink. Figure 1.5 illustrates these notions.

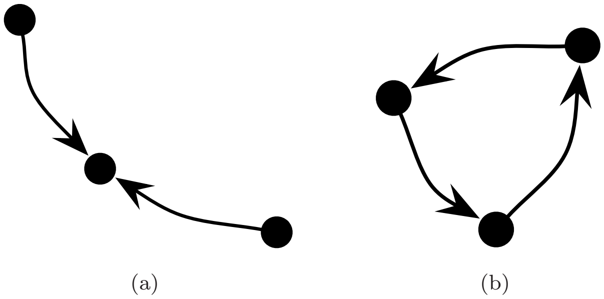


Figure 1.5 Illustrations of connectivity notions on a digraph: (a) shows an acyclic digraph with one sink and two sources; (b) shows a directed path which is also a cycle.

The set of cycles of a directed graph is finite. A directed graph is *aperiodic* if there exists no $k > 1$ that divides the length of every cycle of the graph.

In other words, a digraph is aperiodic if the greatest common divisor of the lengths of its cycles is one. A digraph is *periodic* if it is not aperiodic. Figure 1.6 shows examples of a periodic and an aperiodic digraph.

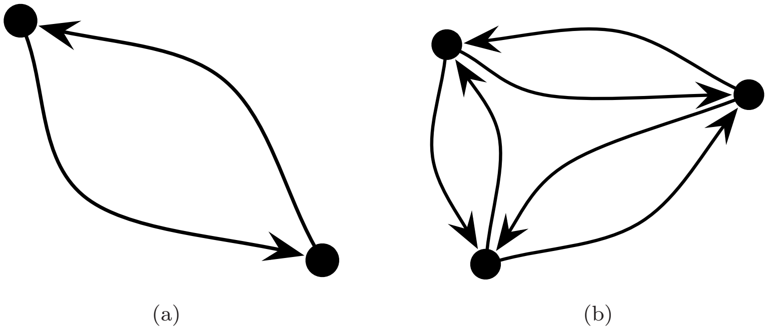


Figure 1.6 (a) A periodic digraph. (b) An aperiodic digraph with cycles of length 2 and 3.

A vertex of a digraph is *globally reachable* if it can be reached from any other vertex by traversing a directed path. A digraph is *strongly connected* if every vertex is globally reachable. The decomposition of a digraph into its strongly connected components and the notion of condensation digraph are discussed in Exercise E1.13.

A *directed tree* (sometimes called a rooted tree) is an acyclic digraph with the following property: there exists a vertex, called the *root*, such that any other vertex of the digraph can be reached by one and only one directed path starting at the root. In a directed tree, every in-neighbor of a vertex is called a *parent* and every out-neighbor is called a *child*. Two vertices with the same parent are called *siblings*. A *successor* of a vertex u is any other node that can be reached with a directed path starting at u. A *predecessor* of a vertex v is any other node such that a directed path exists starting at it and reaching v. A *directed spanning tree*, or simply a *spanning tree*, of a digraph is a spanning subgraph that is a directed tree. Clearly, a digraph contains a spanning tree if and only if the reverse digraph contains a globally reachable vertex. A *(directed) chain* is a directed tree with exactly one source and one sink. A *(directed) ring digraph* is the cycle obtained by adding to the edge set of a chain a new edge from its sink to its source. Figure 1.7 illustrates some of these notions.

The proof of the following result is given in Section 1.8.2.

Lemma 1.26 (Connectivity in topologically balanced digraphs). *Let G be a digraph. The following statements hold:*

(i) if G is strongly connected, then it contains a globally reachable vertex

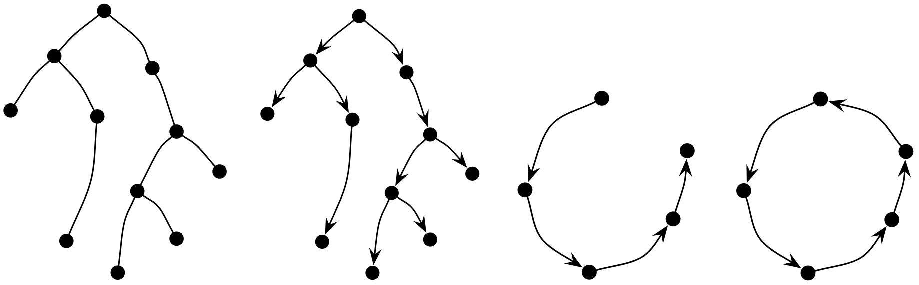


Figure 1.7 From left to right, tree, directed tree, chain, and ring digraphs.

and a spanning tree; and

(ii) if G is topologically balanced and contains either a globally reachable vertex or a spanning tree, then G is strongly connected and is Eulerian.[5]

Given a digraph $G = (V, E)$, an *in-neighbor* of a nonempty set of nodes U is a node $v \in V \setminus U$ for which there exists an edge $(v, u) \in E$ for some $u \in U$.

Lemma 1.27 (Disjoint subsets and spanning trees). *Given a digraph G with at least two nodes, the following two properties are equivalent:*

(i) G has a spanning tree; and

(ii) for any pair of nonempty disjoint subsets U_1, $U_2 \subset V$, either U_1 has an in-neighbor or U_2 has an in-neighbor.

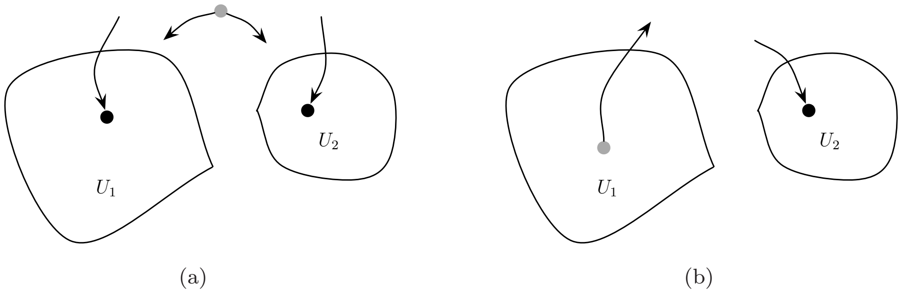


Figure 1.8 An illustration of Lemma 1.27. The root of the spanning tree is plotted in gray. In (a), the root is outside the sets U_1 and U_2. Because these sets are non-empty, there exists a directed path from the root to a vertex in each one of these sets. Therefore, both U_1 and U_2 have in-neighbors. In (b), the root is contained in U_1. Because U_2 is non-empty, there exists a directed path from the root to a vertex in U_2, and, therefore, U_2 has in-neighbors. The case when the root belongs to U_2 is treated analogously.

[5] A graph is Eulerian if it has a cycle that visits all the graph edges exactly once.

We will postpone the proof to Section 1.8.2. The result is illustrated in Figure 1.8. We can also state the result in terms of global reachability: G has a globally reachable node if and only if, for any pair of nonempty disjoint subsets U_1, $U_2 \subset V$, either U_1 has an out-neighbor or U_2 has an out-neighbor. We let the reader give a proper definition of the out-neighbor of a set.

1.4.2 Weighted digraphs

A *weighted digraph* is a triplet $G = (V, E, A)$, where the pair (V, E) is a digraph with nodes $V = \{v_1, \dots, v_n\}$, and where the nonnegative matrix $A \in \mathbb{R}^{n \times n}_{\geq 0}$ is a *weighted adjacency matrix* with the following property: for $i, j \in \{1, \dots, n\}$, the entry $a_{ij} > 0$ if (v_i, v_j) is an edge of G, and $a_{ij} = 0$ otherwise. In other words, the scalars a_{ij}, for all $(v_i, v_j) \in E$, are a set of weights for the edges of G. Note that the edge set is uniquely determined by the weighted adjacency matrix and it can therefore be omitted. When convenient, we denote the adjacency matrix of a weighted digraph G by $A(G)$. Figure 1.9 shows an example of a weighted digraph.

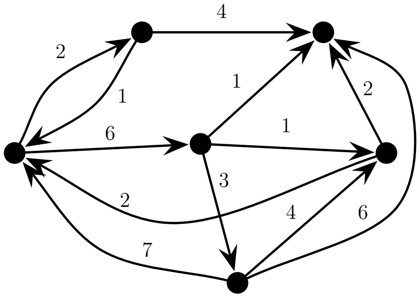


Figure 1.9 A weighted digraph with natural weights.

A digraph $G = (V, E)$ can be naturally thought of as a weighted digraph by defining the weighted adjacency matrix $A \in \{0, 1\}^{n \times n}$ as

$$a_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E, \\ 0, & \text{otherwise,} \end{cases} \tag{1.4.1}$$

where $V = \{v_1, \dots, v_n\}$. The adjacency matrix of a graph is the adjacency matrix of the directed version of the graph. Reciprocally, given a weighted digraph $G = (V, E, A)$, we refer to the digraph (V, E) as the *unweighted version* of G and to its associated adjacency matrix as the *unweighted adjacency matrix*. A weighted digraph is *undirected* if $a_{ij} = a_{ji}$ for all $i, j \in \{1, \dots, n\}$. Clearly, G is undirected if and only if $A(G)$ is symmetric.

Numerous concepts introduced for digraphs remain equally valid for the case of weighted digraphs, including the connectivity notions and the definitions of in- and out-neighbors.

Finally, we generalize the notions of in- and out-degree to weighted digraphs. In a weighted digraph $G = (V, E, A)$ with $V = \{v_1, \dots, v_n\}$, the *weighted out-degree* and the *weighted in-degree* of vertex v_i are defined by, respectively,

$$d_{\text{out}}(v_i) = \sum_{j=1}^{n} a_{ij}, \quad \text{and} \quad d_{\text{in}}(v_i) = \sum_{j=1}^{n} a_{ji}.$$

The weighted digraph G is *weight-balanced* if $d_{\text{out}}(v_i) = d_{\text{in}}(v_i)$ for all $v_i \in V$. The *weighted out-degree matrix* $D_{\text{out}}(G)$ and the *weighted in-degree matrix* $D_{\text{in}}(G)$ are the diagonal matrices defined by

$$D_{\text{out}}(G) = \operatorname{diag}(A\mathbf{1}_n), \quad \text{and} \quad D_{\text{in}}(G) = \operatorname{diag}(A^T\mathbf{1}_n).$$

That is, $(D_{\text{out}}(G))_{ii} = d_{\text{out}}(v_i)$ and $(D_{\text{in}}(G))_{ii} = d_{\text{in}}(v_i)$, respectively.

1.4.3 Distances on digraphs and weighted digraphs

We first present a few definitions for unweighted digraphs. Given a digraph G, the *(topological) length* of a directed path is the number of the edges composing it. Given two vertices u and v in the digraph G, the *distance* from u to v, denoted $\operatorname{dist}_G(u, v)$, is the smallest length of any directed path from u to v, or $+\infty$ if there is no directed path from u to v. That is,

$$\operatorname{dist}_G(u, v) = \min\big(\{\operatorname{length}(p) \mid p \text{ is a directed path from } u \text{ to } v\} \cup \{+\infty\}\big).$$

Given a vertex v of a digraph G, the *radius* of v in G is the maximum of all the distances from v to any other vertex in G. That is,

$$\operatorname{radius}(v, G) = \max\{\operatorname{dist}_G(v, u) \mid u \in V(G)\}.$$

If T is a directed tree and v is its root, then the *depth* of T is $\operatorname{radius}(v, T)$. Finally, the *diameter* of the digraph G is

$$\operatorname{diam}(G) = \max\{\operatorname{dist}_G(u, v) \mid u, v \in V(G)\}.$$

These definitions lead to the following simple results:

(i) $\operatorname{radius}(v, G) \leq \operatorname{diam}(G)$ for all vertices v of G;

(ii) G contains a spanning tree rooted at v if and only if $\operatorname{radius}(v, G) < +\infty$; and

(iii) G is strongly connected if and only if $\operatorname{diam}(G) < +\infty$.

The definitions of path length, distance between vertices, radius of a vertex, and diameter of a digraph can be easily applied to undirected graphs.

Next, we consider weighted digraphs. Given two vertices u and v in the weighted digraph G, the *weighted distance* from u to v, denoted $\operatorname{wdist}_G(u, v)$, is the smallest weight of any directed path from u to v, or $+\infty$ if there is no directed path from u to v. That is,

$$\operatorname{wdist}_G(u, v) = \min\big(\{\operatorname{weight}(p) \mid p \text{ is a directed path from } u \text{ to } v\} \cup \{+\infty\}\big).$$

Here, the weight of a subgraph of a weighted digraph is the sum of the weights of all the edges of the subgraph. Note that when a digraph is thought of as a weighted digraph (with the unweighted adjacency matrix (1.4.1)), the notions of weight and weighted distance correspond to the usual notions of length and distance, respectively. We leave it the reader to provide the definitions of weighted radius, weighted depth, and weighted diameter.

1.4.4 Graph algorithms

In this section, we present a few algorithms defined on graphs. We present only high-level descriptions and we refer to Cormen et al. (2001) for a comprehensive discussion including a detailed treatment of computationally efficient data structures and algorithmic implementations.

1.4.4.1 Breadth-first spanning tree

Let v be a vertex of a digraph G with $\operatorname{radius}(v, G) < +\infty$. A *breadth-first spanning (BFS) tree* of G with respect to v, denoted T_{BFS}, is a spanning directed tree rooted at v that contains a shortest path from v to every other vertex of G. (Here, a shortest path is one with the shortest topological length.) Let us provide the BFS ALGORITHM that, given a digraph G of order n and a vertex v with $\operatorname{radius}(v, G) < +\infty$, computes a BFS tree T_{BFS} rooted at v:

> *[Informal description]* Initialize a subgraph to contain only the root v. Repeat $\operatorname{radius}(v, G)$ times the following instructions: attach to the subgraph all out-neighbors of the subgraph as well as a single connecting edge for each out-neighbor. The final subgraph is the desired directed tree.

The algorithm is formally stated as follows:

```
function BFS($G, v$)
  $(V_1, E_1) := (\{v\}, \emptyset)$
  for $k = 2$ to radius$(v, G)$ do
      find all vertices $w_1, \dots, w_m$ not in $V_{k-1}$ that are out-neighbors of
      some vertex in $V_{k-1}$ and, for $j \in \{1, \dots, m\}$, let $e_j$ be an edge
      connecting a vertex in $V_{k-1}$ to $w_j$
      $V_k := V_{k-1} \cup \{w_1, \dots, w_m\}$
      $E_k := E_{k-1} \cup \{e_1, \dots, e_m\}$
  return $(V_n, E_n)$
```

Note that the output of this algorithm is not necessarily unique, since the choice of edges at step 3: in the algorithm is not unique. Figure 1.10 shows an execution of the BFS ALGORITHM.

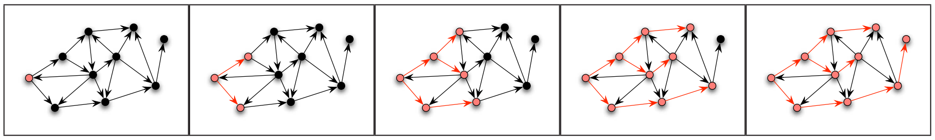


Figure 1.10 Execution of the BFS ALGORITHM. In the leftmost frame, vertex v is colored in red. The other frames correspond to incremental additions of vertices and edges as specified by the function BFS. The output of the algorithm is a BFS tree of the digraph. The BFS tree is represented in the last frame with vertices and edges colored in red.

Some properties of the BFS ALGORITHM are characterized as follows.

Lemma 1.28 (BFS tree). *For a digraph G with a vertex v, any digraph T computed by the* BFS ALGORITHM*, $T \in \text{BFS}(G, v)$, has the following properties:*

(i) T is a directed tree with root v;

(ii) T contains a shortest path from v to any other vertex reachable from v inside G, that is, if there is a path in G from v to w, then $w \in T$ and $\text{dist}_G(v, w) = \text{dist}_T(v, w)$; and

(iii) if G contains a spanning tree rooted at v, then T is spanning too and therefore, T is a BFS tree of G.

We leave the proof to the reader. The key property of the algorithm is that (V_k, E_k), $k \in \{1, \dots, n\}$, is a sequence of directed trees with the property that $(V_k, E_k) \subset (V_{k+1}, E_{k+1})$, for $k \in \{1, \dots, n-1\}$.

1.4.4.2 The depth-first spanning tree

Next, we define the DFS ALGORITHM that, given a digraph G and a vertex v with $\operatorname{radius}(v, G) < +\infty$, computes what we term a *depth-first spanning (DFS) tree* T_{DFS} rooted at v:

> *[Informal description]* Visit all nodes of the graph recording the traveled edges to form the desired tree. Visit the nodes in the following recursive way: (1) as long as a node has an unvisited child, visit it; (2) when the node has no more unvisited children, then return to its parent (and recursively attempt to visit its unvisited children).

The algorithm is formally stated as a recursive procedure, as follows:

```
function DFS($G, v$)
1: $(V_{\mathrm{visited}}, E_{\mathrm{visited}}) := (\{v\}, \emptyset)$
2: DFS-Visit($G, v$)
3: return $(V_{\mathrm{visited}}, E_{\mathrm{visited}})$

function DFS-Visit($G, w$)
1: for $u$ out-neighbor of $w$ do
2:   if $u$ does not belong to $V_{\mathrm{visited}}$ then
3:     $V_{\mathrm{visited}} := V_{\mathrm{visited}} \cup \{u\}$
4:     $E_{\mathrm{visited}} := E_{\mathrm{visited}} \cup \{(w, u)\}$
5:     DFS-Visit($G, u$)
```

Note that the output of this algorithm is not necessarily unique, since the order in which the vertices are chosen in step `1:` of DFS-VISIT is not unique. Any digraph T computed by the DFS ALGORITHM, $T \in \mathrm{DFS}(G, v)$, is a directed spanning tree with root v. Figure 1.11 shows an execution of the algorithm.

Some properties of the DFS ALGORITHM are characterized as follows.

Lemma 1.29 (DFS tree). *For a digraph G with a vertex v, any digraph T computed by the* DFS ALGORITHM, *$T \in \mathrm{DFS}(G, v)$, has the following properties:*

(i) T is a directed tree with root v; and

(ii) if G contains a spanning tree rooted at v, then T is spanning too.

Note that both BFS and DFS trees are uniquely defined once a lexicographic order is introduced for the children of a node.

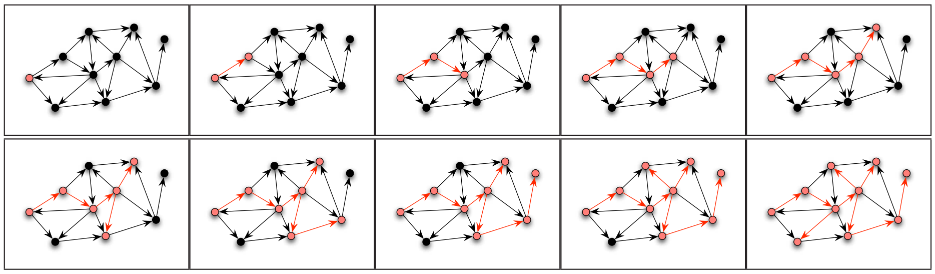


Figure 1.11 Execution of the DFS ALGORITHM. In the top leftmost frame, vertex v is colored in red. The other frames correspond to incremental additions of vertices and edges as specified by the function DFS. The output of the algorithm is a DFS tree of the digraph. The DFS tree is represented in the last frame, with vertices and edges in red.

1.4.4.3 The shortest-paths tree in weighted digraphs via the Dijkstra algorithm

Finally, we focus on weighted digraphs and on the notion of weighted path length. Given a weighted digraph G of order n with weighted adjacency matrix A and a vertex v with $\text{radius}(v, G) < +\infty$, a *shortest-paths tree* of G with respect to v, denoted $T_{\text{shortest-paths}}$, is a spanning directed tree rooted at v that contains a (weighted) shortest path from v to every other vertex of G. This tree differs from the BFS tree defined above because here the path length is measured using the digraph weights.

We now provide the DIJKSTRA ALGORITHM that, given a digraph G of order n and a vertex v with $\text{radius}(v, G) < +\infty$, computes a shortest-paths tree $T_{\text{shortest-paths}}$ rooted at v:

> *[Informal description]* Incrementally construct a tree that contains only shortest paths. In each round, add to the tree (1) the node that is closest to the source and is not yet in the tree, and (2) the edge corresponding to the shortest path. The weighted distance to the source (required to perform step (1)) is computed via an array of distance estimates that is updated as follows: when a node is added to the tree, the distance estimates of all its out-neighbors are updated.

The algorithm is formally stated as follows:

```
function DIJKSTRA$((V, E, A), v)$
  $T_{\text{shortest-paths}} := \emptyset$
  % Initialize estimated distances and estimated parent nodes
  for $u \in V$ do
```

```
 3:    $\texttt{dist}(u) := \begin{cases} 0, & u = v, \\ +\infty, & \text{otherwise.} \end{cases}$
 4:    $\texttt{parent}(u) := u$
    % Main loop to grow the tree and update estimates
 5: while ($T_{\text{shortest-paths}}$ does not contain all vertices) do
 6:    find vertex $u$ outside $T_{\text{shortest-paths}}$ with smallest $\texttt{dist}(u)$
 7:    add to $T_{\text{shortest-paths}}$ the vertex $u$
 8:    if $u \neq v$, add to $T_{\text{shortest-paths}}$ the edge $(\texttt{parent}(u), u)$
 9:    for each node $w$ that is an out-neighbor of $u$ in $(V, E, A)$ do
10:       if $\texttt{dist}(w) > \texttt{dist}(u) + a_{uw}$ then
11:          $\texttt{dist}(w) := \texttt{dist}(u) + a_{uw}$
12:          $\texttt{parent}(w) := u$
13: return $T_{\text{shortest-paths}}$
```

Note that the output of this algorithm is not necessarily unique, since the choice of vertex at step `6:` in the algorithm is not unique. Figure 1.12 shows an execution of the the Dijkstra algorithm.

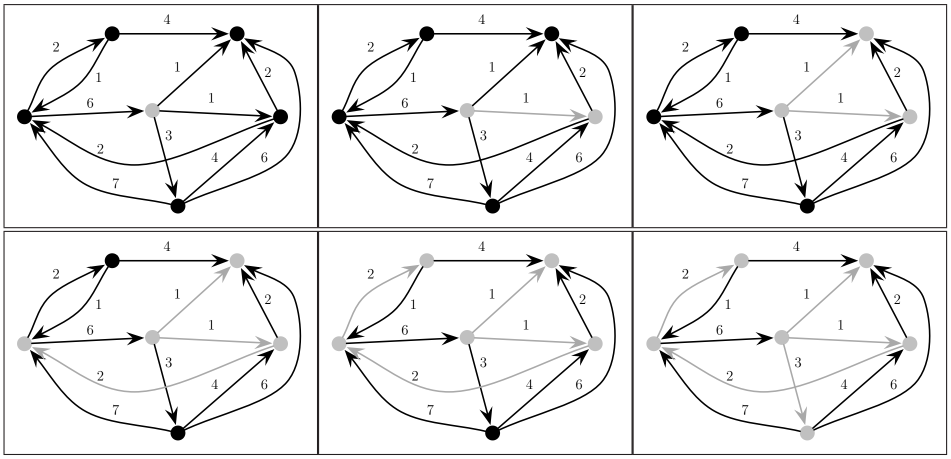


Figure 1.12 Execution of the Dijkstra algorithm on the weighted digraph plotted in Figure 1.9. In the top leftmost frame, vertex v is colored in gray. The other frames correspond to incremental additions of vertices and edges as specified by the function Dijkstra. The output of the algorithm is a shortest-paths tree of the digraph rooted at v. This tree is represented in the last frame with vertices and edges colored in gray.

The following properties of the Dijkstra algorithm mirror those of the BFS algorithm in Lemma 1.28.

Lemma 1.30 (Dijkstra algorithm). *For a weighted digraph G with a vertex v, any digraph T computed by the* Dijkstra algorithm, $T \in$ Dijkstra(G, v), *has the following properties:*

(i) *T is a directed tree with root v;*

(ii) *T contains a shortest path from v to any other vertex reachable from v inside G, that is, if there is a path in G from v to w, then $w \in T$ and $\mathrm{wdist}_G(v, w) = \mathrm{wdist}_T(v, w)$; and*

(iii) *if G contains a spanning tree rooted at v, then T is spanning too, and therefore, T is a shortest-paths tree of G.*

1.4.4.4 On combinatorial optimization problems

We conclude this section on graph algorithms with a brief mention of classic optimization problems defined on graphs. Standard references on combinatorial optimization include Vazirani (2001) and Korte and Vygen (2005). Given a weighted directed graph G, classical combinatorial optimization problems include the following:

Minimum-weight spanning tree. A *minimum-weight spanning tree* of G, denoted MST, is a spanning tree with the minimum possible weight. In order for the MST to exist, G must contain a spanning tree. If all the weights of the individual edges are different, then the MST is unique.

Traveling salesperson problem. A *traveling salesperson tour* of G, denoted TSP, is a cycle that passes through all the nodes of the digraph and has the minimum possible weight. In order for the TSP to exist, G must contain a cycle through all nodes.

Multicenter optimization problems. Given a weighted digraph G with vertices $V = \{v_1, \dots, v_n\}$ and a set $U = \{u_1, \dots, u_k\} \subset V$, the weighted distance from $v \in V$ to the set U is the smallest weighted distance from v to any vertex in $\{u_1, \dots, u_k\}$. We now consider the cost functions $\mathcal{H}_{\max}, \mathcal{H}_{\Sigma} : V^k \to \mathbb{R}$ defined by

$$\mathcal{H}_{\max}(u_1, \dots, u_k) = \max_{i \in \{1,\dots,n\}} \min_{h \in \{1,\dots,k\}} \mathrm{wdist}_G(v_i, u_h),$$

$$\mathcal{H}_{\Sigma}(u_1, \dots, u_k) = \sum_{i=1}^{n} \min_{h \in \{1,\dots,k\}} \mathrm{wdist}_G(v_i, u_h).$$

The *k-center problem* and the *k-median problem* consist of finding a set of vertices $\{u_1, \dots, u_k\}$ that minimizes the *k-center function* $\mathcal{H}_{\max}$ and the *k-median function* $\mathcal{H}_{\Sigma}$, respectively. We refer to Vazirani (2001) for a discussion of the k-center and k-median problems (as well as the more general *uncapacitated facility location* problem) over complete undirected graphs with edge costs satisfying the triangle inequality.

The Euclidean versions of these combinatorial optimization problems refer to the situation where one considers a weighted complete digraph whose vertex set is a point set in $\mathbb{R}^d$, $d \in \mathbb{N}$, and whose weight map assigns to each edge the Euclidean distance between the two nodes connected by the edge.

1.4.5 Algebraic graph theory

Algebraic graph theory (Biggs, 1994; Godsil and Royle, 2001) is the study of matrices defined by digraphs: in this section, we expose two topics. First, we study the equivalence between properties of graphs and of their associated adjacency matrices. We also specify how to associate a digraph to a nonnegative matrix. Second, we introduce and characterize the Laplacian matrix of a weighted digraph.

We begin by studying adjacency matrices. Note that the adjacency matrix of a weighted digraph is nonnegative and, in general, not stochastic. The following lemma expands on this point.

Lemma 1.31 (Weight-balanced digraphs and doubly stochastic adjacency matrices). *Let G be a weighted digraph of order n with weighted adjacency matrix A and weighted out-degree matrix D_{out}. Define the matrix*

$$F = \begin{cases} D_{\text{out}}^{-1} A, & \textit{if each out-degree is strictly positive,} \\ (I_n + D_{\text{out}})^{-1}(I_n + A), & \textit{otherwise.} \end{cases}$$

Then

(i) F is row-stochastic; and

(ii) F is doubly stochastic if G is weight-balanced and the weighted degree is constant for all vertices.

Proof. Consider first the case when each vertex has an outgoing edge so that D_{out} is invertible. We first note that $\operatorname{diag}(v)^{-1}v = \mathbf{1}_n$, for each $v \in (\mathbb{R}\setminus\{0\})^n$. Therefore

$$\big(D_{\text{out}}^{-1}A\big)\mathbf{1}_n = \operatorname{diag}(A\mathbf{1}_n)^{-1}\big(A\mathbf{1}_n\big) = \mathbf{1}_n,$$

which proves (i). Furthermore, if $D_{\text{out}} = D_{\text{in}} = dI_n$ for some $d \in \mathbb{R}_{>0}$, then

$$\big(D_{\text{out}}^{-1}A\big)^T\mathbf{1}_n = \frac{1}{d}\big(A^T\mathbf{1}_n\big) = D_{\text{in}}^{-1}\big(A^T\mathbf{1}_n\big) = \operatorname{diag}(A^T\mathbf{1}_n)^{-1}\big(A^T\mathbf{1}_n\big) = \mathbf{1}_n,$$

which proves (ii). Finally, if (V, E, A) does not have outgoing edges at each vertex, then apply the statement to the weighted digraph $(V, E \cup \{(i,i) \mid i \in \{1, \dots, n\}\}, A + I_n)$. ■

The next result characterizes the relationship between the adjacency matrix and directed paths in the digraph.

Lemma 1.32 (Directed paths and powers of the adjacency matrix). *Let G be a weighted digraph of order n with weighted adjacency matrix A, with unweighted adjacency matrix $A_{0,1} \in \{0,1\}^{n \times n}$, and possibly with self-loops. For all $i, j, k \in \{1, \dots, n\}$*

(i) the (i,j) entry of $A_{0,1}^k$ equals the number of directed paths of length k (including paths with self-loops) from node i to node j; and

(ii) the (i,j) entry of A^k is positive if and only if there exists a directed path of length k (including paths with self-loops) from node i to node j.

Proof. The second statement is a direct consequence of the first. The first statement is proved by induction. The statement is clearly true for $k = 1$. Next, we assume the statement is true for $k \geq 1$ and we prove it for $k + 1$. By assumption, the entry $(A^k)_{ij}$ equals the number of directed paths from i to j of length k. Note that each path from i to j of length $k+1$ identifies (1) a unique node ℓ such that (i, ℓ) is an edge of G and (2) a unique path from ℓ to j of length k. We write $A^{k+1} = AA^k$ in components as

$$(A^{k+1})_{ij} = \sum_{\ell=1}^{n} A_{i\ell}(A^k)_{\ell j}.$$

Therefore, it is true that the entry $(A^{k+1})_{ij}$ equals the number of directed paths from i to j of length $k+1$. This concludes the induction argument. ■

The following proposition characterizes in detail the relationship between various connectivity properties of the digraph and algebraic properties of the adjacency matrix. The result is illustrated in Figure 1.13 and its proof is postponed until Section 1.8.3.

Proposition 1.33 (Connectivity properties of the digraph and positive powers of the adjacency matrix). *Let G be a weighted digraph of order n with weighted adjacency matrix A. The following statements are equivalent:*

(i) G is strongly connected;

(ii) A is irreducible; and

(iii) $\sum_{k=0}^{n-1} A^k$ is positive.

For any $j \in \{1, \dots, n\}$, the following two statements are equivalent:

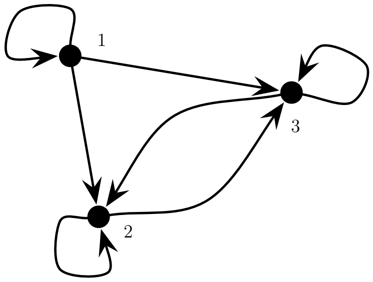


Figure 1.13 An illustration of Proposition 1.33. Even though vertices 2 and 3 are globally reachable, the digraph is not strongly connected because vertex 1 has no in-neighbor other than itself. Therefore, the associated adjacency matrix $A = (a_{ij})$ with $(a_{1j}) = \mathbf{1}_3$, $(a_{2j}) = (a_{3j}) = (0, 1, 1)$, is reducible.

(iv) the jth node of G is globally reachable; and

(v) the jth column of $\sum_{k=0}^{n-1} A^k$ has positive entries.

Stronger statements can be given for digraphs with self-loops.

Proposition 1.34 (Connectivity properties of the digraph and positive powers of the adjacency matrix: cont'd). *Let G be a weighted digraph of order n with weighted adjacency matrix A and with self-loops at each node. The following statements are equivalent:*

(iv) G is strongly connected; and

(v) A^{n-1} is positive.

For any $j \in \{1, \dots, n\}$, the following two statements are equivalent:

(iv) the jth node of G is globally reachable; and

(v) the jth column of A^{n-1} has positive entries.

Next, we characterize the relationship between irreducible aperiodic digraphs and primitive matrices (recall Definition 1.12). We will postpone the proof to Section 1.8.3.

Proposition 1.35 (Strongly connected and aperiodic digraph and primitive adjacency matrix). *Let G be a weighted digraph of order n with weighted adjacency matrix A. The following two statements are equivalent:*

(i) G is strongly connected and aperiodic; and

(ii) A is primitive, that is, there exists $k \in \mathbb{N}$ such that A^k is positive.

This concludes our study of adjacency matrices associated to weighted digraphs. Next, we emphasize how all results obtained so far have analogs that hold when the original object is a nonnegative matrix, instead of a weighted digraph.

Remark 1.36 (From a nonnegative matrix to its associated digraphs). Given a nonnegative $n \times n$ matrix A, its *associated weighted digraph* is the weighted digraph with nodes $\{1, \dots, n\}$, and weighted adjacency matrix A. The unweighted version of this weighted digraph is called the *associated digraph.* The following statements are analogs of the previous lemmas:

(i) if A is stochastic, then its associated digraph has weighted out-degree matrix equal to I_n;

(ii) if A is doubly stochastic, then its associated weighted digraph is weight-balanced and, additionally, both in-degree and out-degree matrices are equal to I_n; and

(iii) A is irreducible if and only if its associated weighted digraph is strongly connected. •

So far, we have analyzed in detail the properties of adjacency matrices. We conclude this section by studying a second relevant matrix associated to a digraph, called the Laplacian matrix. The *Laplacian matrix* of the weighted digraph G is

$$L(G) = D_{\text{out}}(G) - A(G).$$

Some immediate consequences of this definition are the following:

(i) $L(G)\mathbf{1}_n = \mathbf{0}_n$, that is, 0 is an eigenvalue of $L(G)$ with eigenvector $\mathbf{1}_n$;

(ii) G is undirected if and only if $L(G)$ is symmetric; and

(iii) $L(G)$ equals the Laplacian matrix of the digraph obtained by adding to or removing from G any self-loop with arbitrary weight.

Further properties are established as follows.

Theorem 1.37 (Properties of the Laplacian matrix). *Let G be a weighted digraph of order n. The following statements hold:*

(i) all eigenvalues of $L(G)$ have nonnegative real part (thus, if G is undirected, then $L(G)$ is symmetric positive semidefinite);

(ii) if G is strongly connected, then $\operatorname{rank}(L(G)) = n - 1$, *that is,* 0 *is a simple eigenvalue of $L(G)$;*

(iii) *G contains a globally reachable vertex if and only if* $\operatorname{rank}(L(G)) = n-1$*;*

(iv) *the following three statements are equivalent:*

(a) *G is weight-balanced;*

(b) $\mathbf{1}_n^T L(G) = \mathbf{0}_n^T$*; and*

(c) $L(G) + L(G)^T$ *is positive semidefinite.*

1.5 DISTRIBUTED ALGORITHMS ON SYNCHRONOUS NETWORKS

Here, we introduce a synchronous network as a group of processors with the ability to exchange messages and perform local computations. What we present is a basic classic model studied extensively in the distributed algorithms literature. Our treatment is directly adopted with minor variations, from the texts by Lynch (1997) and Peleg (2000).

1.5.1 Physical components and computational models

Loosely speaking, a synchronous network is a group of processors, or nodes, that possess a local state, exchange messages along the edges of a digraph, and compute an update to their local state based on the received messages. Each processor alternates the two tasks of exchanging messages with its neighboring processors and of performing a computation step. Let us begin by describing what constitutes a network.

Definition 1.38 (Network). The physical component of a *synchronous network* $\mathcal{S}$ is a digraph (I, E_{cmm}), where:

(i) $I = \{1, \dots, n\}$ is called the *set of unique identifiers (UIDs)*; and

(ii) E_{cmm} is a set of directed edges over the vertices $\{1, \dots, n\}$, called the communication links. •

In general, the set of unique identifiers does not need to be n consecutive natural numbers, but we adopt this convention for simplicity. The set E_{cmm} models the topology of the communication service among the nodes: for $i, j \in \{1, \dots, n\}$, processor i can send a message to processor j if the directed edge (i, j) is present in E_{cmm}. Note that, unlike the standard treatments in Lynch (1997) and Peleg (2000), we do not assume the digraph to be strongly connected; the required connectivity assumption will be specified on a case-by-case basis.

Next, we discuss the state and the algorithms that each processor possesses and executes, respectively. By convention, we let the superscript $[i]$ denote any quantity associated with the node i.

Definition 1.39 (Distributed algorithm). A *distributed algorithm* $\mathcal{DA}$ for a network $\mathcal{S}$ consists of the sets

(i) $\mathbb{A}$, a set containing the `null` element, called the *communication alphabet*—elements of $\mathbb{A}$ are called *messages*;

(ii) $W^{[i]}$, $i \in I$, called the *processor state sets*; and

(iii) $W_0^{[i]} \subseteq W^{[i]}$, $i \in I$, sets of *allowable initial values*;

and of the maps

(i) $\mathrm{msg}^{[i]} : W^{[i]} \times I \to \mathbb{A}$, $i \in I$, called *message-generation functions*; and

(ii) $\mathrm{stf}^{[i]} : W^{[i]} \times \mathbb{A}^n \to W^{[i]}$, $i \in I$, called *state-transition functions.*

If $W^{[i]} = W$, $\mathrm{msg}^{[i]} = \mathrm{msg}$, and $\mathrm{stf}^{[i]} = \mathrm{stf}$ for all $i \in I$, then $\mathcal{DA}$ is said to be *uniform* and is described by a tuple $(\mathbb{A}, W, \{W_0^{[i]}\}_{i\in I}, \mathrm{msg}, \mathrm{stf})$. •

Now, with all elements in place, we can explain in more detail how a synchronous network executes a distributed algorithm (see Figure 1.14). The *state* of processor i is a variable $w^{[i]} \in W^{[i]}$, initially set equal to an allowable value in $W_0^{[i]}$. At each time instant $\ell \in \mathbb{Z}_{\geq 0}$, processor i sends to each of its out-neighbors j in the communication digraph (I, E_{cmm}) a message (possibly the `null` message) computed by applying the message-generation function $\mathrm{msg}^{[i]}$ to the current values of its state $w^{[i]}$ and to the identity j. Subsequently, but still at time instant $\ell \in \mathbb{Z}_{\geq 0}$, processor i updates the value of its state $w^{[i]}$ by applying the state-transition function $\mathrm{stf}^{[i]}$ to the current value of $w^{[i]}$ and to the messages it receives from its in-neighbors. At each round, the first step is transmission and the second one is computation. These notions are formalized in the following definition.

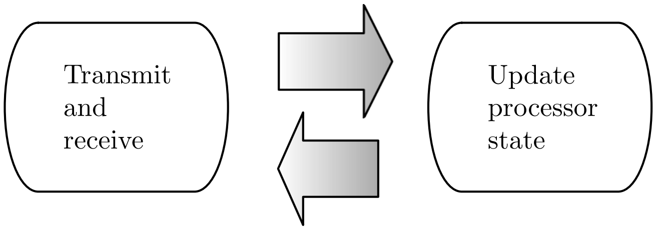


Figure 1.14 The execution of a distributed algorithm by a synchronous network.

Definition 1.40 (Network evolution). Let $\mathcal{DA}$ be a distributed algorithm for the network $\mathcal{S}$. The *evolution* of $(\mathcal{S}, \mathcal{DA})$ from initial conditions $w_0^{[i]} \in W_0^{[i]}$, $i \in I$, is the collection of trajectories $w^{[i]} : \mathbb{Z}_{\geq 0} \to W^{[i]}$, $i \in I$, satisfying

$$w^{[i]}(\ell) = \operatorname{stf}^{[i]}(w^{[i]}(\ell - 1), y^{[i]}(\ell)),$$

where $w^{[i]}(-1) = w_0^{[i]}$, $i \in I$, and where the trajectory $y^{[i]} : \mathbb{Z}_{\geq 0} \to \mathbb{A}^n$ (describing the messages received by processor i) has components $y_j^{[i]}(\ell)$, for $j \in I$, given by

$$y_j^{[i]}(\ell) = \begin{cases} \operatorname{msg}^{[j]}(w^{[j]}(\ell - 1), i), & \text{if } (j, i) \in E_{\text{cmm}}, \\ \texttt{null}, & \text{otherwise.} \end{cases}$$

Let $\ell \mapsto w(\ell) = (w^{[1]}(\ell), \dots, w^{[n]}(\ell))$ denote the collection of trajectories. •

We conclude this section with two sets of remarks. We first discuss some aspects of our communication model that have a large impact on the subsequent development. We then collect a few general comments about control structures and failure modes relevant in the study of distributed algorithms on networks.

Remarks 1.41 (Aspects of the communication model).

(i) The network $\mathcal{S}$ and the algorithm $\mathcal{DA}$ are referred to as *synchronous* because the communications between all processors takes place at the same time for all processors.

(ii) Communication is modeled as a so-called "point-to-point" service: a processor can specify different messages for different out-neighbors and knows the processor identity corresponding to any incoming message.

(iii) Information is exchanged between processors as messages, that is, elements of the alphabet $\mathbb{A}$; the message `null` indicates no communication. Messages might encode logical expressions such as `true` and `false`, or finite-resolution quantized representations of integer and real numbers.

(iv) In some uniform algorithms, the messages between processors are the processors' states. In such cases, the corresponding communication alphabet is $\mathbb{A} = W \cup \{\texttt{null}\}$ and the message-generation function $\operatorname{msg}_{\text{std}}(w, j) = w$ is referred to as the *standard message-generation function.* •

Remarks 1.42 (Advanced topics: Control structures and failures).

(i) Processors in a network have only partial information about the network topology. In general, each processor only knows its own UID, and the UID of its in- and out-neighbors. Sometimes we will assume that the processor knows the network diameter. In some cases (Peleg, 2000), actively running networks might depend upon "control structures," that is, structures that are computed at initial time and are exploited in subsequent algorithms. For example, routing tables might be computed for routing problems, "leader" processors might be elected, and tree structures might be computed and represented in a distributed manner for various tasks; for example, coloring or maximal independent set problems. We present some sample algorithms to compute these structures below.

(ii) A key issue in the study of distributed algorithms is the possible occurrence of failures. A network might experience intermittent or permanent communication failures: along given edges, a `null` message or an arbitrary message might be delivered instead of the intended value. Alternatively, a network might experience various types of processor failures: a processor might transmit only `null` messages (i.e., the msg function always returns `null`), a processor might quit updating its state (i.e., the stf function neglects incoming messages and returns the current state value), or a processor might implement arbitrarily modified msg and stf functions. The latter situation, in which completely arbitrary and possibly malicious behavior is adopted by faulty nodes, is referred to as a Byzantine failure in the distributed algorithms literature. •

1.5.2 Complexity notions

Here, we begin our analysis of the performance of distributed algorithms. We introduce a notion of algorithm completion and, in turn, we introduce the classic notions of time, space, and communication complexity.

Definition 1.43 (Algorithm completion). We say that an algorithm *terminates* when only `null` messages are transmitted and all processors' states become constants. •

Remarks 1.44 (Alternative termination notions).

(i) In the interest of simplicity, we have defined evolutions to be unbounded in time and we do not explicitly require algorithms to actually have termination conditions, that is, to be able to detect when termination takes place.

(ii) It is also possible to define the termination time as the first instant when a given problem or task is achieved, independently of the fact that the algorithm might continue to transmit data subsequently. •

Definition 1.45 (Time complexity). The *(worst-case) time complexity* of a distributed algorithm $\mathcal{DA}$ on a network $\mathcal{S}$, denoted $\mathrm{TC}(\mathcal{DA})$, is the maximum number of rounds required by the execution of $\mathcal{DA}$ on $\mathcal{S}$ among all allowable initial states until termination. •

Next, it is of interest to quantify the memory and communication requirements of distributed algorithms. From an information theory viewpoint (Gallager, 1968), the information content of a memory variable or of a message is properly measured in bits. On the other hand, it is convenient to use the alternative notions of "basic memory unit" and "basic message." It is customary (Peleg, 2000) to assume that a "basic memory unit" or a "basic message" contains $\log(n)$ bits; so that, for example, the information content of a robot identifier $i \in \{1, \dots, n\}$ is $\log(n)$ bits or, equivalently, one "basic memory unit." Note that elements of the processor state set W or of the alphabet set $\mathbb{A}$ might amount to multiple basic memory units or basic messages; the `null` message has zero cost. Unless specified otherwise, the following definitions and examples are stated in terms of basic memory units and basic messages.

Definition 1.46 (Space complexity). The *(worst-case) space complexity* of a distributed algorithm $\mathcal{DA}$ on a network $\mathcal{S}$, denoted by $\mathrm{SC}(\mathcal{DA})$, is the maximum number of basic memory units required by a processor executing $\mathcal{DA}$ on $\mathcal{S}$ among all processors and among all allowable initial states until termination. •

Remark 1.47 (Space complexity conventions). By convention, each processor knows its identity, that is, it requires $\log(n)$ bits to represent its unique identifier in a set with n distinct elements. We do not count this cost in the space complexity of an algorithm. •

Next, we introduce a notion of communication complexity.

Definition 1.48 (Communication complexity). The *(worst-case) communication complexity* of a distributed algorithm $\mathcal{DA}$ on a network $\mathcal{S}$, denoted by $\mathrm{CC}(\mathcal{DA})$, is the maximum number of basic messages transmitted over the entire network during the execution of $\mathcal{DA}$ among all allowable initial states until termination. •

We conclude this section by discussing ways of quantifying time, space and communication complexity. The idea, borrowed from combinatorial optimization, is to adopt asymptotic "order of magnitude" measures. Formally, complexity bounds will be expressed with respect to the Bachmann–Landau symbols O, Ω and Θ defined in Section 1.1. Let us be more specific:

(i) we will say that an algorithm has time complexity *of order* $\Omega(f(n))$ *over some network* if, for all n, there exists a network of order n and initial processor values such that the time complexity of the algorithm is greater than a constant factor times $f(n)$;

(ii) we will say that an algorithm has time complexity *of order* $O(f(n))$ *over arbitrary networks* if, for all n, for all networks of order n and for all initial processor values, the time complexity of the algorithm is lower than a constant factor times $f(n)$; and

(iii) we will say that an algorithm has time complexity *of order* $\Theta(f(n))$ if its time complexity is of order $\Omega(f(n))$ over some network and $O(f(n))$ over arbitrary networks at the same time.

Similar conventions will be used for space and communication complexity.

In many cases, the complexity of an algorithm will typically depend upon the number of vertices of the network. It is therefore useful to present a few simple facts about these functions now. Over arbitrary digraphs $\mathcal{S} = (I, E_{\text{cmm}})$ of order n, we have

$$\operatorname{diam}(\mathcal{S}) \in \Theta(n), \quad |E_{\text{cmm}}(\mathcal{S})| \in \Theta(n^2) \quad \text{and} \quad \operatorname{radius}(v, \mathcal{S}) \in \Theta(\operatorname{diam}(\mathcal{S})),$$

where v is any vertex of $\mathcal{S}$.

Remark 1.49 (Additional complexity notions). Numerous variations of the proposed complexity notions are possible and may be of interest.

Global lower bounds. In the definition of lower bound, consider the logic quantifier describing the role of the network. The lower bound statement is "existential" rather than "global," in the sense that the bound does not hold for all graphs. As discussed in Peleg (2000), it is possible to define also "global" lower bounds, that is, lower bounds over all graphs, or lower bounds over specified classes of graphs.

Average complexity notions. The proposed complexity notions focus on the worst-case situation. It is possible to define *expected* or *average* complexity notions, where one is interested in characterizing, for example, the average number of rounds required or the average number of basic messages transmitted over the entire network during the algorithm execution among all allowable initial states until termination.

Problem complexity. It is possible to define complexity notions for problems, rather than algorithms, by considering, for example, the worst-case optimal performance among all algorithms that solve the given problem, or over classes of algorithms or classes of graphs. •

1.5.3 Broadcast and BFS tree computation

In the following, we consider some basic algorithmic problems such as the simple one-to-all communication task—that is, broadcasting—and the establishment of some "control structures" (see Remarks 1.42), such as the construction of a BFS spanning tree and the election of a leader.

Problem 1.50 (Broadcast). Assume that a processor, called the *source*, has a message, called the *token*. Transmit the token to all others processors in the network. •

Note that existence of a spanning tree rooted at the source is a necessary requirement for the broadcast problem to be solvable. We begin by establishing some analysis results for the broadcast problem.

Lemma 1.51 (Complexity lower bounds for the broadcast problem). *Let $\mathcal{S}$ be a network containing a spanning tree rooted at v. The broadcast problem for $\mathcal{S}$ from the source v has communication complexity lower bounded by $n-1$ and time complexity lower bounded by* $\operatorname{radius}(v, \mathcal{S})$.

In what follows, we shall solve the broadcast problem while simultaneously also considering the following problem.

Problem 1.52 (BFS tree computation). Let $\mathcal{S}$ be a network containing a spanning tree rooted at v. Compute a distributed representation for a BFS tree rooted at v. •

We add two remarks on the BFS tree computation problem:

(i) By a distributed representation of a directed tree with bounded memory at each node, we mean the following: each child vertex knows the identity of its parent and the root vertex knows that it has no parents. A more informative structure would require each parent to know the identity of its children; this is easy to achieve on undirected digraphs.

(ii) The BFS tree computation has the same lower bounds as the broadcast problem.

An elegant and classic solution to the broadcast and BFS tree computation problems is given by the FLOODING ALGORITHM. This algorithm implements the same "breadth-first search" mechanism of the (centralized) BFS ALGORITHM characterized in Lemma 1.28:

> *[Informal description]* The source broadcasts the token to its out-neighbors. In each communication round, each node determines whether it has received a non-`null` message from one of its in-neighbors. When a non-`null` message is received—that is, the token is received—the node performs two actions. First, the node stores the token in the variable `data` (this solves the Broadcast problem). Second, the node stores the identity of one of the transmitting in-neighbors in the variable `parent` (this solves the BFS tree computation problem). Specifically, if the message is received simultaneously from multiple in-neighbors, then the node stores the smallest among the identities of the transmitting in-neighbors. In the subsequent communication round, the node broadcasts the token to its out-neighbors.

To formally describe the algorithm, we assume that the node with the message to be broadcast is $v = 1$. Also, we assume that the token is a letter of the Greek alphabet $\{\alpha, \dots, \omega\}$:

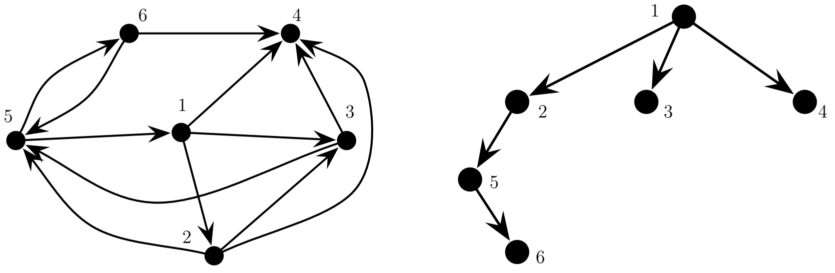
```
Synchronous Network: $\mathcal{S} = (\{1, \dots, n\}, E_{\mathrm{cmm}})$
Distributed Algorithm: FLOODING
Alphabet: $\mathbb{A} = \{\alpha, \dots, \omega\} \cup \texttt{null}$
Processor State: $w = (\texttt{parent}, \texttt{data}, \texttt{snd-flag})$, where
  parent    $\in \{0, \dots, n\}$,          initially: $\texttt{parent}^{[1]} = 1$,
                                                   $\texttt{parent}^{[j]} = 0$ for all $j \neq 1$
  data      $\in \mathbb{A}$,                 initially: $\texttt{data}^{[1]} = \mu$,
                                                   $\texttt{data}^{[j]} = \texttt{null}$ for all $j \neq 1$
  snd-flag  $\in \{\texttt{false}, \texttt{true}\}$, initially: $\texttt{snd-flag}^{[1]} = \texttt{true}$,
                                                   $\texttt{snd-flag}^{[j]} = \texttt{false}$ for $j \neq 1$

function msg$(w, i)$
  if $(\texttt{parent} \neq i)$ AND $(\texttt{snd-flag} = \texttt{true})$ then
    return data
  else
    return null

function stf$(w, y)$
```

```
case
  (data = null) AND ($y$ contains only null messages):
   % The node has not yet received the token
     new-parent := null
     new-data := null
     new-snd-flag := false
  (data = null) AND ($y$ contains a non-null message):
   % The node has just received the token
     new-parent := smallest UID among transmitting in-neighbors
     new-data := a non-null message
     new-snd-flag := true
  (data ≠ null):
   % If the node already has the token, then do not re-broadcast it
     new-parent := parent
     new-data := data
     new-snd-flag := false
return (new-parent, new-data, new-snd-flag)
```

An execution of the FLOODING ALGORITHM is shown in Figure 1.15.



(a)

(b)

Figure 1.15 An example execution of the FLOODING ALGORITHM. The source is vertex 1: (a) shows the network and (b) shows the BFS tree that results from the execution.

This algorithm can analyzed by induction: one can show that, for $d \in \{1, \dots, \text{radius}(v, \mathcal{S})\}$, every node at a distance d from the root receives a non-null message at round d. A summary of the results is given as follows.

Lemma 1.53 (Complexity upper bounds for the flooding algorithm). *For a network $\mathcal{S}$ containing a spanning tree rooted at v, the* FLOODING ALGORITHM *has communication complexity in $\Theta(|E_{\text{cmm}}|)$, time complexity in $\Theta(\text{radius}(v, \mathcal{S}))$, and space complexity in $\Theta(1)$.*

We conclude the section with a final remark.

Remark 1.54 (Termination condition for the flooding algorithm). As presented, the FLOODING ALGORITHM does not include a termination condition, that is, the processors do not have a mechanism to detect when the broadcast and tree computation are complete. If an upper bound on the graph diameter is known, then it is easy to design a termination condition based on this information; we do this in the next subsection. If no *a priori* knowledge is available, then one can design more sophisticated algorithms for networks with stronger connectivity properties. We refer to Lynch (1997) and Peleg (2000) for a complete discussion about this. •

1.5.4 Leader election

Next, we formulate another interesting problem for a network.

Problem 1.55 (Leader election). Assume that all processors of a network have a state variable, say `leader`, initially set to `unknwn`. We say that a leader is elected when one and only one processor has the state variable set to `true` and all others have it set to `false`. Elect a leader. •

This task that is a bit more global in nature. We display here a solution that requires individual processors to know the diameter of the network, denoted by $\operatorname{diam}(\mathcal{S})$, or an upper bound on it:

> *[Informal description]* In each communication round, each agent sends to its out-neighbors the maximum UID it has received up to that time. This is repeated for $\operatorname{diam}(\mathcal{S})$ rounds. At the last round, each agent compares the maximum received UID with its own, and declares itself a leader if they coincide, or a non-leader otherwise.

The algorithm is called the FLOODMAX ALGORITHM: the maximum UID in the network is transmitted to other agents in an incremental fashion. At the first communication round, agents that are neighbors of the agent with the maximum UID receive the message from it. At the next communication round, the neighbors of these agents receive the message with the maximum UID. This process goes on for $\operatorname{diam}(\mathcal{S})$ rounds, to ensure that every agent receives the maximum UID. Note that there are networks for which all agents receive the message with the maximum UID in fewer communication rounds than $\operatorname{diam}(\mathcal{S})$. The algorithm is formally stated as follows:

```
Synchronous Network: $\mathcal{S} = (\{1, \dots, n\}, E_{\text{cmm}})$

Distributed Algorithm: FLOODMAX
```

Alphabet: $\mathbb{A} = \{1, \dots, n\} \cup \{\texttt{null}\}$

Processor State: $w = (\texttt{my-id}, \texttt{max-id}, \texttt{leader}, \texttt{round})$, where

```
my-id  ∈ $\{1,\dots,n\}$,                          initially: my-id$^{[i]} = i$ for all $i$
max-id ∈ $\{1,\dots,n\}$,                          initially: max-id$^{[i]} = i$ for all $i$
leader ∈ $\{$false, true, unknwn$\}$,              initially: leader$^{[i]} =$ unknwn for all $i$
round  ∈ $\{0, 1, \dots, \operatorname{diam}(\mathcal{S})\}$,   initially: round$^{[i]} = 0$ for all $i$
```

```
function msg$(w, i)$
  if round $< \operatorname{diam}(\mathcal{S})$ then
    return max-id
  else
    return null
```

```
function stf$(w, y)$
  new-id:= $\max\{$max-id, largest identifier in $y\}$
  case
    round $< \operatorname{diam}(\mathcal{S})$:    new-lead := unknwn
    round $= \operatorname{diam}(\mathcal{S})$ AND max-id $=$ my-id:    new-lead := true
    round $= \operatorname{diam}(\mathcal{S})$ AND max-id $>$ my-id:    new-lead := false
  return (my-id, new-id, new-lead, round $+1$)
```

Figure 1.16 shows an execution of the FLOODMAX ALGORITHM. Some properties of this algorithm are characterized in the following lemma. A complete analysis of this algorithm, including modifications to improve the communication complexity, is discussed in Lynch (1997, Section 4.1).

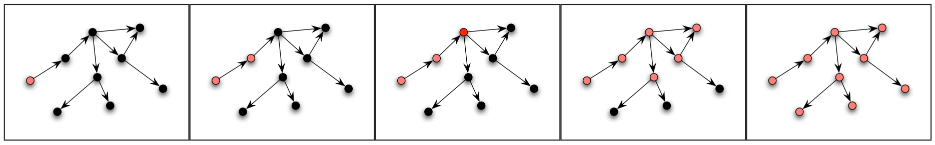


Figure 1.16 Execution of the FLOODMAX ALGORITHM. The diameter of the network is 4. In the leftmost frame, the agent with the maximum UID is colored in red. After four communication rounds, its message has been received by all agents.

Lemma 1.56 (Complexity upper bounds for the floodmax algorithm). *For a network $\mathcal{S}$ containing a spanning tree, the* FLOODMAX ALGORITHM *has communication complexity in $O(\operatorname{diam}(\mathcal{S})|E_{\mathrm{cmm}}|)$, time complexity equal to $\operatorname{diam}(\mathcal{S})$, and space complexity in $\Theta(1)$.*

A simplification of the FLOODMAX ALGORITHM leads to the Le Lann–Chang–Roberts algorithm (or LCR ALGORITHM in short) for leader election in rings, see (Lynch, 1997, Chapter 3.3), which we describe next. The

LCR ALGORITHM runs on a ring digraph and does not require the agents to know the diameter of the network. We provide an informal and a formal description of the algorithm.

> *[Informal description]* In each communication round, each agent sends to its neighbors the maximum UID it has received up to that time. (Agents do not record the number of communication rounds.) When the agent with the maximum UID receives its own UID from an in-neighbor, it declares itself the leader.

`Synchronous Network:` ring digraph

`Distributed Algorithm:` LCR

`Alphabet`: $\mathbb{A} = \{1, \dots, n\} \cup \{\texttt{null}\}$

`Processor State`: $w = (\texttt{my-id}, \texttt{max-id}, \texttt{leader}, \texttt{snd-flag})$, where

```
my-id    ∈ {1,...,n},                initially: my-id^[i] = i for all i
max-id   ∈ {1,...,n},                initially: max-id^[i] = i for all i
leader   ∈ {true,false,unknwn},      initially: leader^[i] = unknwn for all i
snd-flag ∈ {true,false},             initially: snd-flag^[i] = true for all i
```

```
function msg(w, i)
if snd-flag = true then
   return max-id
else
   return null
```

```
function stf(w, y)
case
   (y contains only null msgs) OR (largest identifier in y < my-id):
      new-id := max-id
      new-lead := leader
      new-snd-flag := false
   (largest identifier in y = my-id):
      new-id := max-id
      new-lead := true
      new-snd-flag := false
   (largest identifier in y > my-id):
      new-id := largest identifier in y
      new-lead := false
      new-snd-flag := true
return (my-id, new-id, new-lead, new-snd-flag)
```

Figure 1.17 shows an execution of the LCR ALGORITHM. The properties of the LCR ALGORITHM can be characterized as follows.

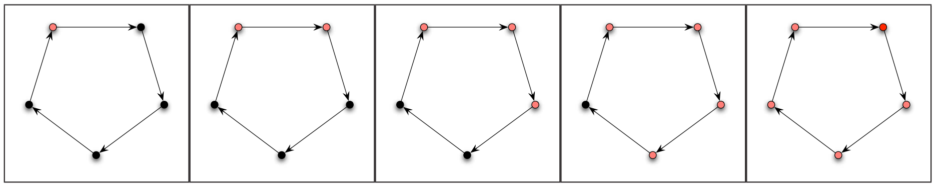


Figure 1.17 Execution of the LCR ALGORITHM. In the leftmost frame, the agent with the maximum UID is colored in red. After five communication rounds, this agent receives its own UID from its in-neighbor and declares itself the leader.

Lemma 1.57 (Complexity upper bounds for the LCR algorithm). *For a ring network $\mathcal{S}$ of order n, the* LCR ALGORITHM *has communication complexity in $\Theta(n^2)$, time complexity equal to n, and space complexity in $\Theta(1)$.*

1.5.5 Shortest-paths tree computation

Finally, we consider the shortest-paths tree problem in a weighted digraph: in Section 1.4.4 we presented the DIJKSTRA ALGORITHM to solve this problem in a centralized setting; we present here the BELLMAN-FORD ALGORITHM for the distributed setting. We consider a synchronous network associated to a weighted digraph, that is, we assume that a strictly positive weight is associated to each communication edge. We aim to compute a tree containing shortest paths from a source, say node 1, to all other nodes. As for the computation of a BFS tree, we aim to obtain a distributed representation of a directed tree with bounded memory at each node:

> *[Informal description]* Each agent maintains in its memory an estimate `dist` of its weighted distance from the source, and an estimate `parent` of the in-neighbor corresponding to the (weighted) shortest path from the source. The `dist` estimate is initialized to 0 for the source and to $+\infty$ for all other nodes. In each communication round, each agent performs the following tasks: (1) it transmits its `dist` value estimate to its out-neighbors, (2) it computes the smallest quantity among "the `dist` value received from an in-neighbor summed with the edge weight corresponding to that in-neighbor," and (3) if the agent's estimate `dist` is larger than this quantity, then the agent updates its `dist` and its estimate `parent`.

The algorithm is formally stated as follows:

`Synchronous Network with Weights:` $\mathcal{S} = (\{1, \dots, n\}, E_{\text{cmm}}, A)$

`Distributed Algorithm:` Distributed Bellman-Ford

`Alphabet:` $\mathbb{A} = \mathbb{R}_{>0} \cup \texttt{null} \cup \{+\infty\}$

`Processor State:` $w = (\texttt{parent}, \texttt{dist})$, where

```
parent ∈ $\{1, \dots, n\}$,    initially: $\texttt{parent}^{[j]} = j$ for all $j$
dist   ∈ $\mathbb{A}$,         initially: $\texttt{data}^{[1]} = 0$,
                               $\texttt{data}^{[j]} = +\infty$ for all $j \neq 1$
```

```
function msg($w, i$)
if round $< n$ then
    return dist
else
    return null
```

```
function stf($w, y$)
$i :=$ processor UID
$k := \operatorname{arginf}\{y_j + a_{ji} \mid \text{for all } y_j \neq \texttt{null}\}$
if (dist $< k$) then
    return (parent, dist)
else
    return $(k, y_k + a_{ki})$
```

In other words, if we let $d_i \in \mathbb{R}_{\geq 0} \cup \{+\infty\}$ denote the `dist` variable for each processor i, then the Bellman-Ford algorithm is equivalent to the following discrete-time dynamical system:

$$d_i(\ell + 1) = \inf \big\{ d_i(\ell) \,,\, \inf\{d_j(\ell) + a_{ji} \mid (j, i) \in E_{\text{cmm}}\} \big\},$$

with initial conditions $d(0) = (1, +\infty, \dots, +\infty)$. (Recall that E_{cmm} is the edge set and that the weights a_{ij} are strictly positive for all $(i, j) \in E_{\text{cmm}}$.)

The following formal statements may be made about the evolution of this algorithm. If there exists a directed spanning tree rooted at vertex 1, then all variables d_i will take a final value in time equal to their topological distance from vertex 1. After k communication rounds, the estimated distance at node i equals the shortest path of topological length at most k from the source to node i. Therefore, after $n - 1$ communication rounds, all possible distinct topological paths connecting source to node i have been investigated.

The complexity properties of the distributed Bellman-Ford algorithm are described as follows.

Lemma 1.58 (Complexity upper bounds for the distributed Bellman-Ford algorithm). *For a network $\mathcal{S}$ of order n containing a spanning tree rooted at v, the* DISTRIBUTED BELLMAN-FORD ALGORITHM *has communication complexity in $\Theta(n|E_{\text{cmm}}|)$, time complexity equal to $n-1$, and space complexity in $\Theta(1)$.*

Figure 1.18 shows an execution of the DISTRIBUTED BELLMAN-FORD ALGORITHM in a weighted digraph with four nodes and six edges.

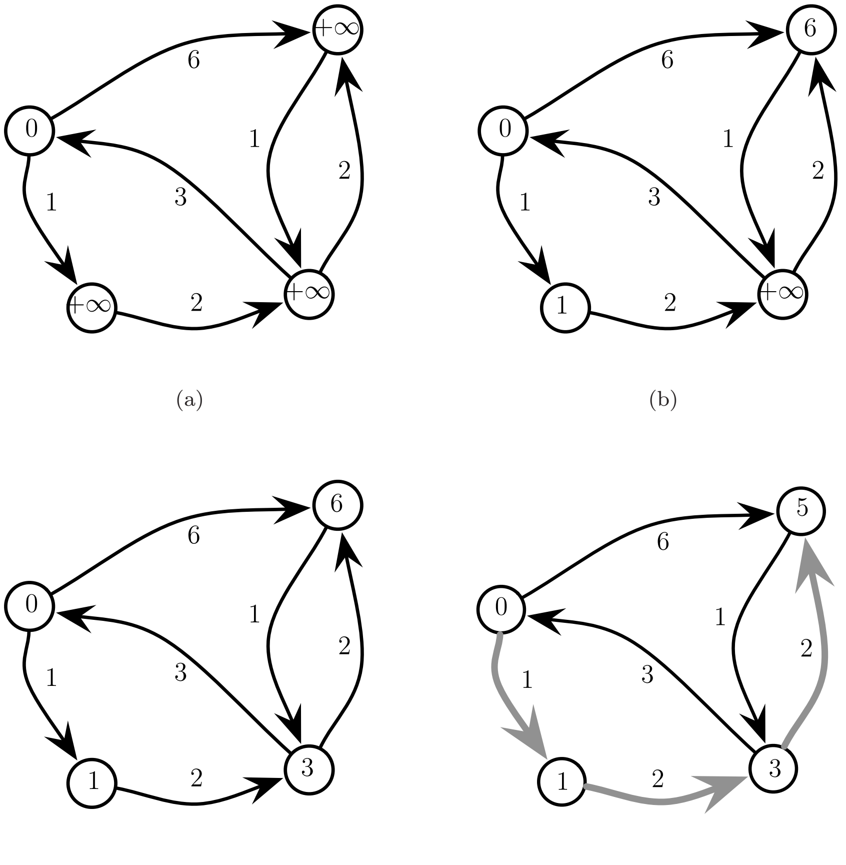


Figure 1.18 Execution of the DISTRIBUTED BELLMAN-FORD ALGORITHM. (a) The processor state initialization. The vertex 1 is the only one whose variable `dist` is 0. After three iterations, as guaranteed by Lemma 1.58, (d) depicts the resulting shortest-paths tree of the digraph rooted at vertex 1. This tree is represented in the last frame, with edges colored in gray.

1.6 LINEAR DISTRIBUTED ALGORITHMS

Computing a linear combination of the initial states of the processors is one of the most basic computation that we might be interested in implementing on a synchronous network. More accurately, linear distributed algorithms on synchronous networks are discrete-time linear dynamical systems whose evolution map is linear and has a sparsity structure related to the network. These algorithms represent an important class of iterative algorithms that find applications in optimization, in the solution of systems of equations, and in distributed decision making; see, for instance Bertsekas and Tsitsiklis (1997). In this section, we present some relevant results on distributed linear algorithms.

1.6.1 Linear iterations on synchronous networks

Given a synchronous network $\mathcal{S} = (\{1, \dots, n\}, E_{\text{cmm}})$, assign a scalar $f_{ji} \neq 0$ to each directed edge $(i, j) \in E_{\text{cmm}}$. Given such scalars f_{ji} for $(i, j) \in E_{\text{cmm}}$, the LINEAR COMBINATION ALGORITHM over $\mathcal{S}$ is defined as follows:

```
Distributed Algorithm: LINEAR COMBINATION
Alphabet: $\mathbb{A} = \mathbb{R} \cup \texttt{null}$
Processor state: $w \in \mathbb{R}$

function msg$(w, i)$ = msg$_{\text{std}}(w, i)$

function stf$(w, y)$
  $i :=$ processor UID
  return $f_{ii} w + \sum_{j \in \mathcal{N}^{\text{in}}(i)} f_{ij} y_j$
```

We assume that each processor $i \in \{1, \dots, n\}$ knows the scalars f_{ij}, for $j \in \mathcal{N}^{\text{in}}(i) \cup \{i\}$, so that it can evaluate the state-transition function. Also, we assume that real numbers may be transmitted through a communication channel, that is, we neglect quantization issues in the message-generation function.

In the language of Section 1.3, one can regard the LINEAR COMBINATION ALGORITHM over $\mathcal{S}$ as the discrete-time continuous-space dynamical system (X, X_0, f), with $X = X_0 = \mathbb{R}^n$ and an evolution map defined by $f(w) = F \cdot w$, where we define a matrix $F \in \mathbb{R}^{n \times n}$ with vanishing entries except for f_{ji}, for $(i, j) \in E_{\text{cmm}}$. Note that, if $A(\mathcal{S})$ denotes the adjacency matrix of the digraph $\mathcal{S}$, then the entries of F vanish precisely when the entries of

$A(\mathcal{S})^T$ vanish. With this notation, the evolution $w : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$ with initial condition $w_0 \in \mathbb{R}^n$ is given by

$$w(0) = w_0, \quad w(\ell+1) = F \cdot w(\ell), \quad \ell \in \mathbb{Z}_{\geq 0}. \tag{1.6.1}$$

Conversely, any linear algorithm of the form (1.6.1) can easily be cast as a Linear combination algorithm over a suitable synchronous network. We do this bookkeeping carefully, in order to be consistent with the notion of associated weighted digraph from Remark 1.36. Given $F \in \mathbb{R}^{n \times n}$, let $\mathcal{S}_F$ be the synchronous network with node set $\{1, \dots, n\}$ and with edge set $E_{\text{cmm}}(F)$, defined by any of the equivalent statements:

(i) $(i, j) \in E_{\text{cmm}}(F)$ if and only if $f_{ji} \neq 0$; or

(ii) $\mathcal{S}_F$ is the reversed and unweighted version of the digraph associated to F.

1.6.2 Averaging algorithms

Here, we study linear combination algorithms over time-dependent weighted directed graphs; we restrict our analysis to nonnegative weights.

Definition 1.59 (Averaging algorithms). The *averaging algorithm* associated to a sequence of stochastic matrices $\{F(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\} \subset \mathbb{R}^{n \times n}$ is the discrete-time dynamical system

$$w(\ell+1) = F(\ell) \cdot w(\ell), \quad \ell \in \mathbb{Z}_{\geq 0}. \tag{1.6.2}$$

•

In the literature, such algorithms are often referred to as agreement algorithms, or as consensus algorithms.

There are useful ways to compute a stochastic matrix, and therefore, a time-independent averaging algorithm, from a weighted digraph; see Exercise E1.15.

Definition 1.60 (Adjacency- and Laplacian-based averaging). Let G be a weighted digraph with node set $\{1, \dots, n\}$, weighted adjacency matrix A, weighted out-degree matrix D_{out}, and weighted Laplacian L. Then

(i) the *adjacency-based averaging algorithm* is defined by the stochastic matrix $(I_n + D_{\text{out}})^{-1}(I_n + A)$ and reads in components

$$w_i(\ell+1) = \frac{1}{1 + d_{\text{out}}(i)} \big(w_i(\ell) + \sum_{j=1}^{n} a_{ij} w_j(\ell) \big); \tag{1.6.3}$$

(ii) given a positive scalar ε upper bounded by $\min\{1/d_{\text{out}}(i) \mid i \in \{1,\dots,n\}\}$, the *Laplacian-based averaging algorithm* is defined by the stochastic matrix $I_n - \varepsilon L(G)$ and reads in components

$$w_i(\ell+1) = \Big(1 - \varepsilon \sum_{j=1, j\neq i}^{n} a_{ij}\Big) w_i(\ell) + \varepsilon \sum_{j=1, j\neq i}^{n} a_{ij} w_j(\ell). \tag{1.6.4}$$

These notions are immediately extended to sequences of stochastic matrices arising from sequences of weighted digraphs. •

Adjacency-based averaging algorithms arising from unweighted undirected graphs without self-loops are also known as *equal-neighbor averaging rule* or the *Vicsek's model* (see Vicsek et al., 1995). Specifically, if G is an unweighted graph with vertices $\{1,\dots,n\}$ and without self-loops, then the equal-neighbor averaging rule is

$$w_i(\ell+1) = \operatorname{avrg}\Big(\{w_i(\ell)\} \cup \{w_j(\ell) \mid j \in \mathcal{N}_G(i)\}\Big), \tag{1.6.5}$$

where we adopt the shorthand $\operatorname{avrg}(\{x_1,\dots,x_k\}) = (x_1 + \dots + x_k)/k$.

Remark 1.61 (Sensing versus communication interpretation of directed edges). In the definition of averaging algorithms arising from digraphs, the digraph edges play the role of "sensing edges," not that of "communication edges." In other words, a nonzero entry a_{ij}, corresponding to the digraph edge (i,j), implies that the ith component of the state is updated with the jth component of the state. It is as if node i could sense the state of node j, rather than node i transmitting to node j its own state. •

Next, we present the main stability and convergence results for averaging algorithms associated to a sequence of stochastic matrices. We start by discussing equilibrium points and their stability. Recall that $\mathbf{1}_n$ is an eigenvector of any stochastic matrix with eigenvalue 1 and that the diagonal set $\operatorname{diag}(\mathbb{R}^n)$ is the vector subspace generated by $\mathbf{1}_n$. Therefore, any point in $\operatorname{diag}(\mathbb{R}^n)$ is an equilibrium for any averaging algorithm. We refer to the points of the $\operatorname{diag}(\mathbb{R}^n)$ as *agreement configurations*, since all the components of an element in $\operatorname{diag}(\mathbb{R}^n)$ are equal to the same value. We will informally say that an algorithm *achieves agreement* if it steers the network state toward the set of agreement configurations.

Lemma 1.62 (Stability of agreement configurations). *Any averaging algorithm in $\mathbb{R}^n$ is uniformly stable and uniformly bounded with respect to* $\operatorname{diag}(\mathbb{R}^n)$.

Regarding convergence results, we need to introduce a useful property of collections of stochastic matrices. Given $\alpha \in]0,1]$, the set of *non-degenerate matrices with respect to* α consists of all stochastic matrices F with entries f_{ij}, for $i, j \in \{1, \dots, n\}$, satisfying

$$f_{ii} \in [\alpha, 1], \quad \text{and} \quad f_{ij} \in \{0\} \cup [\alpha, 1] \text{ for } j \neq i.$$

Additionally, the sequence of stochastic matrices $\{F(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\}$ is *non-degenerate* if there exists $\alpha \in]0,1]$ such that $F(\ell)$ is non-degenerate with respect to α for all $\ell \in \mathbb{Z}_{\geq 0}$. We now state the main convergence result and postpone its proof to Section 1.8.5.

Theorem 1.63 (Convergence for time-dependent stochastic matrices). *Let $\{F(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\} \subset \mathbb{R}^{n \times n}$ be a non-degenerate sequence of stochastic matrices. For $\ell \in \mathbb{Z}_{\geq 0}$, let $G(\ell)$ be the unweighted digraph associated to $F(\ell)$, according to Remark 1.36. The following statements are equivalent:*

(i) the set $\operatorname{diag}(\mathbb{R}^n)$ is uniformly globally attractive for the averaging algorithm associated to $\{F(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\}$; and

(ii) there exists a duration $\delta \in \mathbb{N}$ such that, for all $\ell \in \mathbb{Z}_{\geq 0}$, the digraph

$$G(\ell + 1) \cup \dots \cup G(\ell + \delta)$$

contains a globally reachable vertex.

We collect a few observations about this result.

Remarks 1.64 (Discussion of Theorem 1.63).

(i) The statement in Theorem 1.63(i) means that each solution to the time-dependent linear dynamical system (1.6.2) converges uniformly and asymptotically to the vector subspace generated by $\mathbf{1}_n$.

(ii) The necessary and sufficient condition in Theorem 1.63(ii) amounts to the existence of a uniformly bounded time duration δ with the property that a weak connectivity assumption holds over each collection of δ consecutive digraphs. We refer to Blondel et al. (2005) for a counterexample showing that if the duration in Theorem 1.63 is not uniformly bounded, then there exist algorithms that do not converge.

(iii) According to Definition 1.23, uniform convergence is a property of all solutions to system (1.6.2) starting at *any arbitrary time*, and not only at time equal to zero. If we restrict our attention to solutions that only start at time zero, then Theorem 1.63 should be modified as follows: the statement in Theorem 1.63(i) implies, but is not implied by, the statement in Theorem 1.63(ii).

(iv) The theorem applies only to sequences of non-degenerate matrices. Indeed, there exist sequences of degenerate stochastic matrices whose associated averaging algorithms converge. Furthermore, one does not even need to consider sequences, because it is possible to define converging algorithms by just considering a single stochastic matrix. Precisely when the stochastic matrix is primitive, we already know that the associated averaging algorithm will converge (see Theorem 1.13). Examples of degenerate primitive stochastic matrices (with converging associated averaging algorithms) are given in Exercise E1.23. We discuss time-invariant averaging algorithms in Proposition 1.68 below. •

Theorem 1.63 gives a general result about non-degenerate stochastic matrices that are not necessarily symmetric. The following theorem presents a convergence result for the case of symmetric matrices (i.e., undirected digraphs) under connectivity requirements that are weaker (i.e., the duration does not need to be uniformly bounded) than those expressed in statement (ii) of Theorem 1.63.

Theorem 1.65 (Convergence for time-dependent stochastic symmetric matrices). *Let $\{F(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\} \subset \mathbb{R}^{n\times n}$ be a non-degenerate sequence of symmetric, stochastic matrices. For $\ell \in \mathbb{Z}_{\geq 0}$, let $G(\ell)$ be the unweighted graph associated to $F(\ell)$, according to Remark 1.36. The following statements are equivalent:*

(i) the set $\operatorname{diag}(\mathbb{R}^n)$ is globally attractive for the averaging algorithm associated to $\{F(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\}$; and

(ii) for all $\ell \in \mathbb{Z}_{\geq 0}$, the graph

$$\bigcup_{\tau \geq \ell} G(\tau)$$

is connected.

Let us particularize our discussion here on adjacency- and Laplacian-based averaging algorithms.

Corollary 1.66 (Convergence of adjacency- and Laplacian-based averaging algorithms). *Let $\{G(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\} \subset \mathbb{R}^{n\times n}$ be a sequence of weighted digraphs. The following statements are equivalent:*

(i) there exists $\delta \in \mathbb{N}$ such that, for all $\ell \in \mathbb{Z}_{\geq 0}$, the digraph

$$G(\ell+1) \cup \cdots \cup G(\ell+\delta)$$

contains a globally reachable vertex;

(ii) the set $\operatorname{diag}(\mathbb{R}^n)$ is uniformly globally attractive for the adjacency-

based averaging algorithm (1.6.3) *associated to* $\{G(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\}$*; and*

(iii) the set $\operatorname{diag}(\mathbb{R}^n)$ *is uniformly globally attractive for the Laplacian-based averaging algorithm* (1.6.4) *(defined with* $\varepsilon < 1/n$*) associated to* $\{G(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\}$*.*

Finally, we refine the results presented thus far by discussing some further aspects.

Proposition 1.67 (Convergence to a point in the invariant set). *Under the assumptions in Theorem 1.63 and assuming that* $\operatorname{diag}(\mathbb{R}^n)$ *is uniformly globally attractive for the averaging algorithm, each individual evolution converges to a specific point of* $\operatorname{diag}(\mathbb{R}^n)$*.*

In general, the final value upon which all w_i, $i \in \{1, \dots, n\}$, agree in the limit is unknown. This final value depends on the initial condition and the specific sequence of matrices defining the time-dependent linear algorithm. In some cases, however, one can compute the final value by restricting the class of allowable matrices. We consider two settings: time-independent averaging algorithms and doubly stochastic averaging algorithms.

First, we specialize the main convergence result to the case of time-independent averaging algorithms. Note that, given a stochastic matrix F, convergence of the averaging algorithm associated to F for all initial conditions is equivalent to the matrix F being semi-convergent (see Definition 1.6).

Proposition 1.68 (Time-independent averaging algorithm). *Consider the linear dynamical system on* $\mathbb{R}^n$

$$w(\ell + 1) = F w(\ell), \quad \ell \in \mathbb{Z}_{\geq 0}. \tag{1.6.6}$$

Assume that $F \in \mathbb{R}^{n \times n}$ *is stochastic, let* $G(F)$ *denote its associated weighted digraph, and let* $v \in \mathbb{R}^n$ *be a left eigenvector of* F *with eigenvalue* 1*. Assume either one of the two following properties:*

(i) F *is primitive (i.e.,* $G(F)$ *is strongly connected and aperiodic); or*

(ii) F *has non-zero diagonal terms and a column of* F^{n-1} *has positive entries (i.e.,* $G(F)$ *has self-loops at each node and has a globally reachable node).*

Then every trajectory w *of system* (1.6.6) *converges to* $(v^T w(0)/v^T \mathbf{1}_n)\mathbf{1}_n$*.*

Proof. From Theorem 1.63 we know that the dynamical system (1.6.6) converges if property (ii) holds. The same conclusion follows if F satisfies property (i) because of the Perron–Frobenius Theorem 1.13 and Lemma 1.7. To computing the limiting value, note that

$$v^T w(\ell+1) = v^T F w(\ell) = v^T w(\ell),$$

that is, the quantity $\ell \mapsto v^T w(\ell)$ is constant. Because F is semi-convergent and stochastic, we know that $\lim_{\ell\to+\infty} w(\ell) = \alpha \mathbf{1}_n$ for some α. To conclude, we compute α from the relationship $\alpha(v^T\mathbf{1}_n) = \lim_{\ell\to+\infty} v^T w(\ell) = v^T w(0)$. ■

Remarks 1.69 (Alternative conditions for time-independent averaging).

(i) The following necessary and sufficient condition generalizes and is weaker than the two sufficient conditions given in Proposition 1.68: every trajectory of system (1.6.6) is asymptotically convergent if and only if all sinks of the condensation digraph of $G(F)$ are aperiodic subgraphs of $G(F)$. We refer the interested reader to Meyer (2001, Chapter 8) for the proof of this statement and for the related notion of ergodic classes of a Markov chain. Also, we refer the interested reader to Exercise E1.13 for the notion of condensation digraph.

(ii) Without introducing any trajectory w, the result of the proposition can be equivalently stated by saying that

$$\lim_{\ell\to+\infty} F^\ell = (v^T\mathbf{1}_n)^{-1}\mathbf{1}_n v^T. \qquad \bullet$$

Second, we focus on the case of doubly stochastic averaging algorithms.

Corollary 1.70 (Average consensus). *Let $\{F(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\}$ be a sequence of stochastic matrices as in Theorem 1.63. If all matrices $F(\ell)$, $\ell \in \mathbb{Z}_{\geq 0}$, are doubly stochastic, then every trajectory w of the averaging algorithms satisfies*

$$\sum_{i=1}^n w_i(\ell) = \sum_{i=1}^n w_i(0), \quad \textit{for all } \ell,$$

that is, the sum of the initial conditions is a conserved quantity. Therefore, if $\{F(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\}$ is non-degenerate and satisfies property (ii) in Theorem 1.63, then

$$\lim_{\ell\to+\infty} w_j(\ell) = \frac{1}{n}\sum_{i=1}^n w_i(0), \quad j \in \{1,\dots,n\}.$$

Proof. The proof of the first fact is an immediate consequence of

$$\sum_{i=1}^{n} w_i(\ell+1) = \mathbf{1}_n^T w(\ell+1) = \mathbf{1}_n^T F(\ell) w(\ell) = \mathbf{1}_n^T w(\ell) = \sum_{i=1}^{n} w_i(\ell).$$

The second fact is an immediate consequence of the first fact. ■

In other words, if the matrices are doubly stochastic, then each component of the trajectories will converge to the average of the initial condition. We therefore adopt the following definition: an *average-consensus averaging algorithm* is an averaging algorithm whose sequence of stochastic matrices are all doubly stochastic.

1.6.3 The convergence speed of averaging algorithms

We know that any trajectory of the associated averaging algorithm converges to the diagonal set $\mathrm{diag}(\mathbb{R}^n)$; in what follows we characterize how fast this convergence takes place. We begin with some general definitions for semi-convergent matrices (recall the discussion culminating in Lemma 1.7).

Definition 1.71 (Convergence time and exponential convergence factor). Let $A \in \mathbb{R}^{n\times n}$ be semi-convergent with limit $\lim_{\ell\to+\infty} A^\ell = A^*$.

(i) For $\varepsilon \in \,]0,1[$, the *ε-convergence time* of A is the smallest time $T_\varepsilon(A) \in \mathbb{Z}_{\geq 0}$ such that, for all $x_0 \in \mathbb{R}^n$ and $\ell \geq T_\varepsilon(A)$,

$$\left\| A^\ell x_0 - A^* x_0 \right\|_2 \leq \varepsilon \|x_0 - A^* x_0\|_2.$$

(ii) The *exponential convergence factor* of A, denoted by $r_{\mathrm{exp}}(A) \in [0,1[$, is

$$r_{\mathrm{exp}}(A) = \sup_{x_0 \neq A^* x_0} \ \limsup_{\ell\to+\infty} \Big(\frac{\|A^\ell x_0 - A^* x_0\|_2}{\|x_0 - A^* x_0\|_2} \Big)^{1/\ell}. \qquad \bullet$$

The exponential convergence factor has the following interpretation: If the trajectory $x(\ell) = A^\ell x_0$ maximizing the sup operator has the form $x(\ell) = \rho^\ell (x_0 - x^*) + x^*$, for $\rho < 1$, then it is immediate to see that $r_{\mathrm{exp}}(A) = \rho$.

Lemma 1.72 (Exponential convergence factor of a convergent matrix). *If A is a convergent matrix, then $r_{\mathrm{exp}}(A) = \rho(A)$.*

In what follows, we are interested in studying how the convergence time and the exponential convergence factor of a matrix depend upon ε and upon the dimension of the matrix itself.

Remark 1.73 (Complexity notions). Analogously to the treatment in Section 1.5.2, we introduce some complexity notions. Let $A_n \in \mathbb{R}^{n\times n}$, $n \in \mathbb{N}$, be a sequence of semi-convergent matrices with limit $\lim_{\ell\to+\infty} A_n^\ell = A_n^*$, and let $\varepsilon \in \,]0,1]$. We say that:

(i) $T_\varepsilon(A_n)$ is *of order* $\Omega(f(n,\varepsilon))$ if, for all n and all ε, there exists an initial condition $x_0 \in \mathbb{R}^n$ such that $\|A_n^\ell x_0 - A^* x_0\|_2 > \varepsilon\big\|x_0 - A^* x_0\big\|_2$ for all times ℓ greater than a constant factor times $f(n,\varepsilon)$;

(ii) $T_\varepsilon(A_n)$ is *of order* $O(f(n,\varepsilon))$ if, for all n and all ε, $T_\varepsilon(A_n)$ is less than or equal to a constant factor times $f(n,\varepsilon)$; and

(iii) $T_\varepsilon(A_n)$ is *of order* $\Theta(f(n,\varepsilon))$ if it is both of order $\Omega(f(n,\varepsilon))$ and of order $O(f(n,\varepsilon))$. •

Lemma 1.74 (Asymptotic relationship). *Let $A_n \in \mathbb{R}^{n\times n}$, $n \in \mathbb{N}$, be a sequence of semi-convergent matrices and let $\varepsilon \in \,]0,1]$. In the limit as $\varepsilon \to 0^+$ and as $n \to +\infty$,*

$$T_\varepsilon(A_n) \in O\Big(\frac{1}{1 - r_{\exp}(A_n)} \log \varepsilon^{-1}\Big).$$

Proof. By the definition of the exponential convergence factor and of lim sup, we know that for all $\eta > 0$, there exists N such that, for all $\ell > N$,

$$\big\|A^\ell x_0 - A^* x_0\big\|_2 \leq (r_{\exp}(A_n) + \eta)^\ell \|x_0 - A^* x_0\|_2.$$

The ε-convergence time is upper bounded by any ℓ such that $(r_{\exp}(A_n) + \eta)^\ell \leq \varepsilon$. Selecting $\eta = (1 - r_{\exp}(A_n))/2$, simple manipulations lead to

$$\ell \geq \frac{1}{-\log((r_{\exp}(A_n) + 1)/2)} \log \varepsilon^{-1}.$$

It is also immediate to note that $\frac{2}{1-r} \geq \frac{1}{-\log((r+1)/2)}$, for all $r \in \,]0,1[$. This establishes the bound in the statement above. ■

Next, we apply the notion of convergence time and exponential convergence factor to any non-degenerate stochastic matrix whose associated digraph has a globally reachable node.

Lemma 1.75 (Exponential convergence factor of stochastic matrices). *Let F be a stochastic matrix with strictly positive diagonal entries and whose associated digraph has a globally reachable node. Then*

$$r_{\exp}(F) = \rho_{\text{ess}}(F).$$

(From equation (1.2.1)*, recall that* $\rho_{\mathrm{ess}}(F) = \max\{\|\lambda\|_{\mathbb{C}} \mid \lambda \in \operatorname{spec}(F) \setminus \{1\}\}$*.)*

Proof. If $v \in \mathbb{R}^n$ is a left eigenvector of F, then, as in Proposition 1.68,

$$\lim_{\ell \to +\infty} F^{\ell} = F^* = (v^T \mathbf{1}_n)^{-1} \mathbf{1}_n v^T.$$

Relying upon $v^T F = v^T$ and $F\mathbf{1}_n = \mathbf{1}_n$, straightforward manipulations show that $F^* = F^*F = FF^* = F^*F^*$ and in turn

$$F^{\ell+1} - F^* = (F - F^*)(F^{\ell} - F^*).$$

For any $w_0 \in \mathbb{R}^n$ such that $w_0 \neq F^* w_0$, define the error variable $e(\ell) := F^{\ell} w_0 - F^* w_0$. Note that the error variable evolves according to $e(\ell+1) = (F - F^*)e(\ell)$ and converges to zero. Additionally, the rate at which $w(\ell) = F^{\ell} w_0$ converges to $F^* w_0$ is the same at which $e(\ell)$ converges to zero, that is,

$$r_{\mathrm{exp}}(F - F^*) = r_{\mathrm{exp}}(F).$$

Therefore,

$$r_{\mathrm{exp}}(F) = r_{\mathrm{exp}}(F - F^*) = \rho(F - F^*) = \rho_{\mathrm{ess}}(F).$$

■

The following result establishes bounds on convergence factors and convergence times for stochastic matrices arising from the equal-neighbor averaging rule in equation (1.6.5).

Theorem 1.76 (Bounds on the convergence factor and the convergence time). *Let G be an undirected unweighted connected graph of order n and let $\varepsilon \in \,]0,1]$. Define the stochastic matrix $F = (I_n + D(G))^{-1}(I_n + A(G))$. There exists $\gamma > 0$ (independent of n) such that the exponential convergence factor and convergence time of F satisfy*

$$r_{\mathrm{exp}}(F) \leq 1 - \gamma n^{-3}, \quad \textit{and} \quad T_{\varepsilon}(F) \in O(n^3 \log \varepsilon^{-1}),$$

as $\varepsilon \to 0^+$ and $n \to +\infty$.

1.6.4 Algorithms defined by tridiagonal Toeplitz and tridiagonal circulant matrices

This section presents a detailed analysis of the convergence rates of linear distributed algorithms defined by tridiagonal Toeplitz matrices and by certain circulant matrices. Let us start by introducing the family of matrices under study. For $n \geq 2$ and $a, b, c \in \mathbb{R}$, define the $n \times n$ matrices $\operatorname{Trid}_n(a, b, c)$ and $\operatorname{Circ}_n(a, b, c)$ by

$$\operatorname{Trid}_n(a,b,c) = \begin{bmatrix} b & c & 0 & \dots & 0 \\ a & b & c & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & a & b & c \\ 0 & \dots & 0 & a & b \end{bmatrix},$$

and

$$\operatorname{Circ}_n(a,b,c) = \operatorname{Trid}_n(a,b,c) + \begin{bmatrix} 0 & \dots & \dots & 0 & a \\ 0 & \dots & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ c & 0 & \dots & 0 & 0 \end{bmatrix}.$$

We call the matrices Trid_n and Circ_n *tridiagonal Toeplitz* and *tridiagonal circulant*, respectively. The two matrices only differ in their $(1, n)$ and $(n, 1)$ entries. Note our convention that

$$\operatorname{Circ}_2(a,b,c) = \begin{bmatrix} b & a+c \\ a+c & b \end{bmatrix}.$$

Note that, for $a = 0$ and $c \neq 0$ (alternatively, $a \neq 0$ and $c = 0$), the synchronous networks defined by $\operatorname{Trid}(a, b, c)$ and $\operatorname{Circ}(a, b, c)$ are, respectively, the chain and the ring digraphs introduced in Section 1.4. If both a and c are non-vanishing, then the synchronous networks are, respectively, the undirected versions of the chain and the ring digraphs.

Now, we characterize the eigenvalues and eigenvectors of Trid_n and Circ_n.

Lemma 1.77 (Eigenvalues and eigenvectors of tridiagonal Toeplitz and tridiagonal circulant matrices). *For $n \geq 2$ and $a, b, c \in \mathbb{R}$, the following statements hold:*

(i) for $ac \neq 0$, the eigenvalues and eigenvectors of $\operatorname{Trid}_n(a, b, c)$ are, respectively, for $i \in \{1, \dots, n\}$,

$$b + 2c\sqrt{\frac{a}{c}} \cos\left(\frac{i\pi}{n+1}\right) \in \mathbb{C}, \qquad \begin{pmatrix} \left(\frac{a}{c}\right)^{1/2} \sin\left(\frac{i\pi}{n+1}\right) \\ \left(\frac{a}{c}\right)^{2/2} \sin\left(\frac{2i\pi}{n+1}\right) \\ \vdots \\ \left(\frac{a}{c}\right)^{n/2} \sin\left(\frac{ni\pi}{n+1}\right) \end{pmatrix} \in \mathbb{C}^n;$$

(ii) the eigenvalues and eigenvectors of $\operatorname{Circ}_n(a, b, c)$ are, respectively,

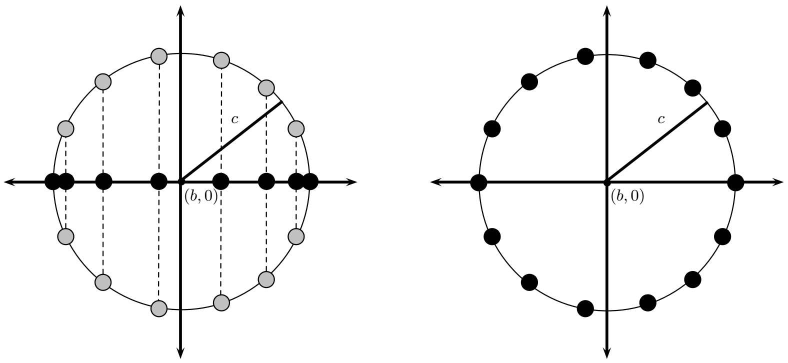
for $i \in \{1, \dots, n\}$ *and* $\omega = \exp(\frac{2\pi\sqrt{-1}}{n})$,

$$b + (a+c)\cos\left(\frac{i2\pi}{n}\right) + \sqrt{-1}(c-a)\sin\left(\frac{i2\pi}{n}\right) \in \mathbb{C},$$

and $(1, \omega^i, \dots, \omega^{(n-1)i})^T \in \mathbb{C}^n$.

Proof. Both facts are discussed, for example, in Meyer (2001, Example 7.2.5 and Exercise 7.2.20). Fact (ii) requires some straightforward algebraic manipulations. ■

Figure 1.19 illustrates the location of the eigenvalues of these matrices in the complex plane.



(a)

(b)

Figure 1.19 The eigenvalues of Toeplitz and circulant matrices (cf., Lemma 1.77) are closely related to the roots of unity. Plotted in the complex plane, the black disks correspond in (a) to the eigenvalues of $\mathrm{Trid}_{13}(a, b, c)$, and in (b) to the eigenvalues of $\mathrm{Circ}_{14}(0, b, c)$.

Remarks 1.78 (Inclusion relationships for eigenvalues of tridiagonal Toeplitz and tridiagonal circulant matrices).

(i) The set of eigenvalues of $\mathrm{Trid}_n(a, b, c)$ is contained in the real interval $[b - 2\sqrt{ac}, b + 2\sqrt{ac}]$, if $ac \geq 0$, and in the interval in the complex plane $[b - 2\sqrt{-1}\sqrt{|ac|}, b + 2\sqrt{-1}\sqrt{|ac|}]$, if $ac \leq 0$.

(ii) The set of eigenvalues of $\mathrm{Circ}_n(a, b, c)$ is contained in the ellipse on the complex plane with center b, horizontal axis $2|a+c|$, and vertical axis $2|c - a|$. •

Next, we characterize the convergence rate of linear algorithms defined by tridiagonal Toeplitz and tridiagonal circulant matrices. As in the previous section, we are interested in asymptotic results as the system dimension $n \to +\infty$ and as the accuracy parameter ε goes to 0^+.

Theorem 1.79 (Linear algorithms defined by tridiagonal Toeplitz and tridiagonal circulant matrices). *Let $n \geq 2$, $\varepsilon \in]0,1[$, and $a, b, c \in \mathbb{R}$. Let $x : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$ and $y : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$ be solutions to*

$$x(\ell+1) = \mathrm{Trid}_n(a,b,c)\,x(\ell), \qquad y(\ell+1) = \mathrm{Circ}_n(a,b,c)\,y(\ell),$$

with initial conditions $x(0) = x_0$ and $y(0) = y_0$, respectively. The following statements hold:

(i) if $a = c \neq 0$ and $|b| + 2|a| = 1$, then $\lim_{\ell\to+\infty} x(\ell) = \mathbf{0}_n$ with ε-convergence time in $\Theta\big(n^2 \log \varepsilon^{-1}\big)$;

(ii) if $a \neq 0$, $c = 0$ and $0 < |b| < 1$, then $\lim_{\ell\to+\infty} x(\ell) = \mathbf{0}_n$ with ε-convergence time in $O\big(n \log n + \log \varepsilon^{-1}\big)$; and

(iii) if $a \geq 0$, $c \geq 0$, $1 > b > 0$ and $a + b + c = 1$, then $\lim_{\ell\to+\infty} y(\ell) = \big(\frac{1}{n}\mathbf{1}_n^T y_0\big)\mathbf{1}_n$ with ε-convergence time in $\Theta\big(n^2 \log \varepsilon^{-1}\big)$.

The proof of this result is reported in Section 1.8.6. Next, we extend these results to another interesting set of tridiagonal matrices. For $n \geq 2$ and $a, b \in \mathbb{R}$, define the $n \times n$ matrices $\mathrm{ATrid}_n^+(a,b)$ and $\mathrm{ATrid}_n^-(a,b)$ by

$$\mathrm{ATrid}_n^{\pm}(a,b) = \mathrm{Trid}_n(a,b,a) \pm \begin{bmatrix} a & 0 & \dots & \dots & 0 \\ 0 & 0 & \dots & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 0 & 0 \\ 0 & \dots & \dots & 0 & a \end{bmatrix}.$$

We refer to these matrices as *augmented tridiagonal* matrices. If we define

$$P_+ = \begin{bmatrix} 1 & 1 & 0 & 0 & \dots & 0 \\ 1 & -1 & 1 & 0 & \dots & 0 \\ 1 & 0 & -1 & 1 & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \\ 1 & 0 & \dots & 0 & -1 & 1 \\ 1 & 0 & \dots & 0 & 0 & -1 \end{bmatrix},$$

and

$$P_- = \begin{bmatrix} 1 & 1 & 0 & 0 & \dots & 0 \\ -1 & 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & 1 & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \\ (-1)^{n-2} & 0 & \dots & 0 & 1 & 1 \\ (-1)^{n-1} & 0 & \dots & 0 & 0 & 1 \end{bmatrix},$$

then the following similarity transforms are satisfied:

$$\mathrm{ATrid}_n^{\pm}(a,b) = P_{\pm} \begin{bmatrix} b \pm 2a & 0 \\ 0 & \mathrm{Trid}_{n-1}(a,b,a) \end{bmatrix} P_{\pm}^{-1}. \tag{1.6.7}$$

To analyze the convergence properties of the linear algorithms determined by $\mathrm{ATrid}_n^{+}(a,b)$ and $\mathrm{ATrid}_n^{-}(a,b)$, we will find it useful to consider the vector

$$\mathbf{1}_{n-}^T = (1, -1, 1, \dots, (-1)^{n-2}, (-1)^{n-1})^T \in \mathbb{R}^n.$$

In the following theorem, we will not assume that the matrices of interest are semi-convergent. We will establish convergence to a trajectory, rather than to a fixed point. For $\varepsilon \in]0,1[$, we say that a trajectory $x : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$ converges to $x_{\text{final}} : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$ with convergence time $T_\varepsilon \in \mathbb{Z}_{\geq 0}$ if

(i) $\|x(\ell) - x_{\text{final}}(\ell)\|_2 \to 0$ as $\ell \to +\infty$; and

(ii) T_ε is the smallest time such that $\|x(\ell) - x_{\text{final}}(\ell)\|_2 \leq \varepsilon \|x(0) - x_{\text{final}}(0)\|_2$, for all $\ell \geq T_\varepsilon$.

Theorem 1.80 (Linear algorithms defined by augmented tridiagonal matrices). *Let $n \geq 2$, $\varepsilon \in]0,1[$, and $a, b \in \mathbb{R}$ with $a \neq 0$ and $|b| + 2|a| = 1$. Let $x : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$ and $z : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$ be solutions to*

$$x(\ell+1) = \mathrm{ATrid}_n^{+}(a,b)\, x(\ell), \qquad z(\ell+1) = \mathrm{ATrid}_n^{-}(a,b)\, z(\ell),$$

with initial conditions $x(0) = x_0$ and $z(0) = z_0$, respectively. The following statements hold:

(i) $\lim_{\ell \to +\infty} \big(x(\ell) - x_{\text{ave}}(\ell)\mathbf{1}_n\big) = \mathbf{0}_n$, where $x_{\text{ave}}(\ell) = (\frac{1}{n}\mathbf{1}_n^T x_0)(b+2a)^\ell$, with ε-convergence time in $\Theta\big(n^2 \log \varepsilon^{-1}\big)$; and

(ii) $\lim_{\ell \to +\infty} \big(z(\ell) - z_{\text{ave}}(\ell)\mathbf{1}_{n-}\big) = \mathbf{0}_n$, where $z_{\text{ave}}(\ell) = (\frac{1}{n}\mathbf{1}_{n-}^T z_0)(b - 2a)^\ell$, with ε-convergence time in $\Theta\big(n^2 \log \varepsilon^{-1}\big)$.

The proof of this result is reported in Section 1.8.6.

Remark 1.81 (From Toeplitz to stochastic matrices). A tridiagonal Toeplitz matrix is not stochastic unless its off-diagonal elements are zero. The tridiagonal circulant matrices Circ_n and augmented tridiagonal matrices ATrid_n^{+} studied in Theorem 1.79(iii) and Theorem 1.80(i) are slight modifications of tridiagonal Toeplitz matrices and are doubly stochastic. Indeed, the evolutions converge to the average consensus value, as predicted by Corollary 1.70. Note that convergence times obtained for Circ_n and ATrid_n^+ are consistent with the upper bound predicted by Theorem 1.76. •

We conclude this section with some useful bounds.

Lemma 1.82 (Bounds on vector norms). *Assume that $x \in \mathbb{R}^n$, $y \in \mathbb{R}^{n-1}$, and $z \in \mathbb{R}^{n-1}$ jointly satisfy*

$$x = P_+ \begin{bmatrix} 0 \\ y \end{bmatrix}, \qquad x = P_- \begin{bmatrix} 0 \\ z \end{bmatrix}.$$

Then $\frac{1}{2}\|x\|_2 \leq \|y\|_2 \leq (n-1)\|x\|_2$ and $\frac{1}{2}\|x\|_2 \leq \|z\|_2 \leq (n-1)\|x\|_2$.

The proof of this result is based on spelling out the coordinate expressions for x, y, and z, and is left to the reader as Exercise E1.29.

1.7 NOTES

Dynamical systems and stability theory

Our definition of a state machine is very basic; more general definitions of state machines can be found in the literature (see Sipser, 2005), but the one presented in this chapter is sufficient for our purposes.

The literature on dynamical and control systems is vast. The main tool that we use in later chapters is the LaSalle Invariance Principle, obtained by LaSalle (1960) and discussed in LaSalle (1986); see also the earlier works by Barbašin and Krasovskiĭ (1952) and Krasovskiĭ (1963) for related versions. Relevant sample references include modern texts on dynamical systems (Guckenheimer and Holmes, 1990), linear control systems (Chen, 1984), nonlinear control systems (Khalil, 2002), robust control (Dullerud and Paganini, 2000), and discrete-event systems (Cassandras and Lafortune, 2007).

Graph theory

The basic definitions of graph theory are standard in the literature; see, for example, Biggs (1994), Godsil and Royle (2001), and Diestel (2005). The discussion about graph algorithms is taken from Cormen et al. (2001), which also contains detailed discussion on implementation and complexity issues.

Regarding Section 1.4.4.4, standard references on combinatorial optimization include Vazirani (2001) and Korte and Vygen (2005).

In Section 1.4.5, all statements about powers of the adjacency matrix are standard results in algebraic graph theory; see, for example Biggs (1994) and Godsil and Royle (2001). Lemma 1.27 is a recent result from Lin et al. (2005) andMoreau (2005). Proposition 1.35, on the fact that a weighted digraph is aperiodic and irreducible if and only if its adjacency matrix is primitive, is related to standard results in the theory of Markov chains; see, for example Seneta (1981) and Meyn and Tweedie (1999). Our proof adopts the approach in Lin (2005). Laplacian matrices have numerous remarkable properties; two elegant surveys are Mohar (1991) and Merris (1994). Theorem 1.37, characterizing the properties of the Laplacian matrix, contains some recent results. A proof of statement (ii) is given in Olfati-Saber and Murray (2004); in our proof, we follow the approach in Francis (2006). Statement (iii) is proved by Lin et al. (2005) and Francis (2006); the following equivalent version is proved in Ren and Beard (2005): a weighted digraph G contains a spanning tree if and only if $\operatorname{rank}(L(\operatorname{rev}(G))) = n-1$. Regarding statement (iv), the equivalence between (iv)a and (iv)b is proved by Olfati-Saber and Murray (2004) and the equivalence between (iv)b and (iv)c is proved by Moreau (2005).

Distributed algorithms

Our discussion of distributed algorithms is extremely incomplete. We have only presented a few token ideas and we refer to the textbooks by Lynch (1997) and Peleg (2000) for detailed treatments. Let us mention briefly that many more efficient algorithms are available in the literature—for example, the GHS algorithm (Gallager et al., 1983) for minimum spanning tree computation and consensus algorithms with communication and processors faults; much attention is dedicated to fault tolerance in asynchronous systems with shared memory and in asynchronous network systems.

Linear distributed algorithms

Distributed linear algorithms—and, in particular, averaging iterations that achieve consensus among processors—have a long and rich history. The richness comes from the vivid analogies with physical processes of diffusion, with Markov chain models, and with the sharp theory of positive matrices developed by Perron and Frobenius. What follows is a necessarily incomplete list. An early reference on averaging opinions and achieving consensus is DeGroot (1974). An early reference on the connection between averaging algorithms, the products of stochastic matrices, and ergodicity in inhomogeneous Markov chains is Chatterjee and Seneta (1977) – the history of inhomogeneous Markov chains being a classic topic since the early twentieth century. The stochastic setting was investigated in Cogburn (1984). Load balancing with divisible tasks in parallel computers is discussed in Cybenko (1989). A comprehensive theory of asynchronous parallel processors implementing distributed gradient methods and time-dependent averaging algorithms is developed in the series of works Tsitsiklis (1984), Tsitsiklis et al. (1986), and Bertsekas and Tsitsiklis (1997). Much interest for averaging algorithms arose from the influential work on flocking by Jadbabaie et al. (2003). Sharp conditions for convergence for the time-dependent setting were obtained in Moreau (2005). Finally, proper attention was given to the average consensus problem in Olfati-Saber and Murray (2004).

Regarding Theorem 1.63, characterizing the convergence of averaging algorithms defined by sequences of stochastic matrices, we note that: (1) the PhD thesis Tsitsiklis (1984) established convergence under a strong-connectivity assumption; (2) a sufficient condition was independently rediscovered in Jadbabaie et al. (2003), adopting a result from Wolfowitz (1963); and (3) Moreau (2003, 2005) obtained the necessary and sufficient condition (for uniform convergence in non-degenerate sequences) involving the existence of a uniformly globally reachable node. The work in Moreau (2003, 2005) is an early reference also for Theorem 1.65; additional related results and a historical discussion appeared in Blondel et al. (2005) and Hendrickx (2008). The estimates of the convergence factor given in Theorem 1.76 in Section 1.6.3 were proved by Landau and Odlyzko (1981). Our treatment in Section 1.6.4 follows Martínez et al. (2007).

Among the numerous recent directions of research on consensus and averaging, we would like to mention the following: continuous-time consensus algorithms (Olfati-Saber and Murray, 2004; Moreau, 2004; Lin et al., 2004; Ren and Beard, 2005; Lin et al., 2005, 2007), consensus over random networks (Hatano and Mesbahi, 2005; Wu, 2006; Patterson et al., 2007; Picci and Taylor, 2007; Porfiri and Stilwell, 2007; Tahbaz-Salehi and Jadbabaie, 2008; Fagnani and Zampieri, 2009), consensus in finite time (Cortés, 2006; Sundaram and Hadjicostis, 2008), consensus in small-world networks (Olfati-Saber, 2005; Durrett, 2006; Tahbaz-Salehi and Jadbabaie, 2007), consensus algorithms for general functions (Bauso et al., 2006; Cortés, 2008b; Lorenz and Lorenz, 2008; Sundaram and Hadjicostis, 2008), connections with the heat equation and partial difference equation (Ferrari-Trecate et al., 2006), spatially decaying interactions (Cucker and Smale, 2007), convergence in time-delayed and asynchronous settings (Blondel et al., 2005; Angeli and Bliman, 2006; Fang and Antsaklis, 2008), quantized consensus problems (Savkin, 2004; Kashyap et al., 2007; Carli et al., 2009; Zhu and Martínez, 2008b), consensus on manifolds (Scardovi et al., 2007; Sarlette and Sepulchre, 2009; Igarashi et al., 2007), applications to distributed signal processing (Spanos et al., 2005; Xiao et al., 2005; Olfati-Saber et al., 2006; Zhu and Martínez, 2008a), characterization of convergence rates and time complexity (Landau and Odlyzko, 1981; Olshevsky and Tsitsiklis, 2009; Cao et al., 2008; Carli et al., 2008). Numerous interesting results are reported in recent PhD theses (Lin, 2005; Cao, 2007; Lorenz, 2007; Barooah, 2007; Carli, 2008; Hendrickx, 2008; Sarlette, 2009). Finally, we would like to point out two recent surveys (Olfati-Saber et al., 2007; Ren et al., 2007) and the text by (Ren and Beard, 2008).

Synchronization is a fascinating topic related to averaging algorithms. A very early reference is the work by Huygens (1673) on coupled pendula. The synchronization of oscillators in dynamical systems has received increasing attention, and key references include Wiener (1958), Kuramoto (1975), Winfree (1980), Kuramoto (1984), Strogatz (2000), and Nijmeijer (2001); see also the widely accessible Strogatz (2003). Under all-to-all interactions, Mirollo and Strogatz (1990) prove synchronization of a collection of "integrate and fire" biological oscillators. Recent works on the Kuramoto and other synchronized oscillator models include Jadbabaie et al. (2004), Chopra and Spong (2009), Triplett et al. (2006), Papachristodoulou and Jadbabaie (2006), Wang and Slotine (2006).

1.8 PROOFS

This section gathers the proofs of the main results presented in the chapter.

1.8.1 Proof of Theorem 1.21

Here we provide the proof of the LaSalle Invariance Principle for set-valued discrete-time dynamical systems. We remark that Theorem 1.19 is an immediate consequence of Theorem 1.21 and that Theorem 1.20 is proved in a similar way (for details, we refer to (Khalil, 2002)).

Proof of Theorem 1.21. Let γ be any evolution of (X, X_0, T) starting from W. Let $\Omega(\gamma)$ denote the ω-limit set[6] of the sequence $\gamma = \{\gamma(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\}$; since W is closed, it follows that $\Omega(\gamma) \subset W$. Next, we prove that $\Omega(\gamma)$ is weakly positively invariant. Let $z \in \Omega(\gamma)$. Then there exists a subsequence

[6] The ω-limit set of a sequence $\gamma = \{\gamma(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\}$ is the set of points y for which there exists a subsequence $\{\gamma(\ell_m) \mid m \in \mathbb{Z}_{\geq 0}\}$ of γ such that $\lim_{m \to +\infty} \gamma(\ell_m) = y$.

$\{\gamma(\ell_m) \mid m \in \mathbb{Z}_{\geq 0}\}$ of γ such that $\lim_{m\to+\infty} \gamma(\ell_m) = z$. Consider the sequence $\{\gamma(\ell_m + 1) \mid m \in \mathbb{Z}_{\geq 0}\}$. Since this sequence is bounded, it has a convergent subsequence. For ease of notation, we use the same notation to refer to it, that is, there exists y such that $\lim_{m\to+\infty} \gamma(\ell_m + 1) = y$. By definition, $y \in \Omega(\gamma)$. Moreover, using the fact that T is closed, we deduce that $y \in T(z)$. Therefore, $\Omega(\gamma)$ is weakly positively invariant.

Now, consider the sequence $V \circ \gamma = \{V(\gamma(\ell)) \mid \ell \in \mathbb{Z}_{\geq 0}\}$. Since γ is bounded and V is non-increasing along T on W, the sequence $V \circ \gamma$ is decreasing and bounded from below, and therefore, convergent. Let $c \in \mathbb{R}$ satisfy $\lim_{\ell\to+\infty} V(\gamma(\ell)) = c$. Next, we prove that the value of V on $\Omega(\gamma)$ is constant and equal to c. Take any $z \in \Omega(\gamma)$. Accordingly, there exists a subsequence $\{\gamma(\ell_m) \mid m \in \mathbb{Z}_{\geq 0}\}$ such that $\lim_{m\to+\infty} \gamma(\ell_m) = z$. Since V is continuous, $\lim_{m\to+\infty} V(\gamma(\ell_m)) = V(z)$. From $\lim_{\ell\to+\infty} V(\gamma(\ell)) = c$, we conclude that $V(z) = c$.

Finally, the fact that $\Omega(\gamma)$ is weakly positively invariant and V being constant on $\Omega(\gamma)$ implies that

$$\Omega(\gamma) \subset \{x \in X \mid \exists y \in T(x) \text{ such that } V(y) = V(x)\}.$$

Therefore, we conclude that $\lim_{\ell\to+\infty} \operatorname{dist}(\gamma(\ell), S \cap V^{-1}(c)) = 0$, where S is the largest weakly positively invariant set contained in $\{x \in X \mid \exists y \in T(x) \text{ such that } V(y) = V(x)\}$. ■

1.8.2 Proofs of Lemmas 1.26 and 1.27

Proof of Lemma 1.26. The first statement is obvious. Regarding the second statement, we prove that a topologically balanced digraph with a globally reachable node is strongly connected, and leave the proof of the other case to the reader. We reason by contradiction. Assume that G is not strongly connected. Let $S \subset V$ be the set of all nodes of G that are globally reachable. By hypothesis, $S \neq \emptyset$. Since G is not strongly connected, we have $S \subsetneq V$. Note that any outgoing edge with origin in a globally reachable node automatically makes the destination a globally reachable node too. This implies that there cannot be any outgoing edges from a node in S to a node in $V \setminus S$. Let $v \in V \setminus S$ such that v has an out-neighbor in S (such a node must exist, since otherwise the nodes in S cannot be globally reachable). Since by hypothesis G is balanced, there must exist an edge of the form $(w, v) \in E$. Clearly, $w \notin S$, since otherwise v would be globally reachable too, which is a contradiction. Therefore, $w \in V \setminus S$. Again, using the fact that G is topologically balanced, there must exist an edge of the form $(z, w) \in E$. As before, $z \in V \setminus S$ (note that $z = v$ is a possibility). Since $V \setminus S$ is finite and so is the number of possible edges between its nodes, applying this argument repeatedly, we find that there exists a vertex whose out-degree is strictly larger than its in-degree, which is a contradiction with the fact that G is topologically balanced. We refer to Cortés (2008b) for the proof that G is Eulerian. ■

Proof of Lemma 1.27. *(i)* $\implies$ *(ii)* Assume that $i \in V$ is the root of the spanning tree and take an arbitrary pair of nonempty, disjoint subsets $U_1, U_2 \subset V$. If $i \in U_1$, then there must exist a path from $i \in U_1$ to a node in U_2. Therefore, U_2 must have an in-neighbor. Analogously, if $i \in U_2$, then U_1 must have an in-neighbor. Finally, it is possible that $i \notin U_1 \cup U_2$. In this case, there exist paths from i to both U_1 and U_2, that is, both sets have in-neighbors.

(ii) $\implies$ *(i)* This is proved by finding a node from which there exists a path to all others. We do this in an algorithmic manner using induction. At each induction step k, except the last one, four sets of nodes are considered, $U_1(k) \subset W_1(k) \subset V$, $U_2(k) \subset W_2(k) \subset V$, with the following properties:

(a) the sets $W_1(k)$ and $W_2(k)$ are disjoint; and

(b) from each node of $U_s(k)$ there exists a path to each other node in $W_s(k) \setminus U_s(k)$, $s \in \{1, 2\}$.

Induction Step k=1: Set $U_1 = W_1 = \{i_1\}$ and $U_2 = W_2 = \{i_2\}$, where i_1, i_2 are two arbitrary different nodes of the graph that satisfy the properties (a) and (b).

Induction Step $k > 1$: Suppose that for $k-1$ we found sets $U_1(k-1) \subset W_1(k-1)$ and $U_2(k-1) \subset W(k-1)$ as in (a) and (b). Since $U_1(k-1)$ and $U_2(k-1)$ are disjoint, then there exists either an edge (i_k, j_1) with $j_1 \in U_1(k-1)$, $i_k \in V \setminus U_1(k-1)$, or an edge (i_k, j_2) with $j_2 \in U_2(k-1)$ and $i_k \in V \setminus U_2(k-1)$. Suppose that an edge (i_k, j_2) exists (the case of a edge (i_k, j_1) can be treated in a similar way). Only four cases are possible.

(A) If $i_k \in W_1(k-1)$ and $W_1(k-1) \cup W_2(k-1) = V$, then we can terminate the algorithm and conclude that from any node $h \in U_1(k-1)$ there exists a path to all other nodes in the graph and thus there is a spanning tree.

(B) If $i_k \in W_1(k-1)$ and $W_1(k-1) \cup W_2(k-1) \neq V$, then set:

$$\begin{aligned} U_1(k) &= U_1(k-1), \\ W_1(k) &= W_1(k-1) \cup W_2(k-2), \\ U_2(k) &= W_2(k) = \{h_k\}, \end{aligned}$$

where h_k is an arbitrary node not belonging to $W_1(k-1) \cup W_2(k-1)$.

(C) If $i_k \notin W_1(k-1) \cup W_2(k-1)$, then set

$$\begin{aligned} U_1(k) &= U_1(k-1), \\ W_1(k) &= W_1(k-1), \\ U_2(k) &= \{i_k\}, \\ W_2(k) &= W_2(k-1) \cup \{i_k\}. \end{aligned}$$

(D) If $i_k \in W_2(k-1) \setminus U_2(k-1)$ then

$$\begin{aligned} U_1(k) &= U_1(k-1), \\ W_1(k) &= W_1(k-1), \\ U_2(k) &= U_2(k-1) \cup \{i_k\}, \\ W_2(k) &= W_2(k-1). \end{aligned}$$

The algorithm terminates in a finite number of induction steps because at each step, except when finally case (A) holds true, either the number of nodes in $W_1 \cup W_2$ increases, or the number of nodes in $W_1 \cup W_2$ remains constant and the number of nodes in $U_1 \cup U_2$ increases. ■

1.8.3 Proofs of Propositions 1.33 and 1.35

Proof of Proposition 1.33. (ii) $\Longrightarrow$ *(i)* We aim to show that there exist directed paths from any node to any other node. Fix $i \in \{1, \dots, n\}$ and let $R_i \subset \{1, \dots, n\}$ be the set of nodes that belong to directed paths originating from node i. Denote the unreachable nodes by $U_i = \{1, \dots, n\} \setminus R_i$. We argue that U_i cannot contain any element, because if it does, then $R_i \cup U_i$ is a nontrivial partition of the index set $\{1, \dots, n\}$ and irreducibility implies the existence of a non-zero entry a_{jk} with $j \in R_i$ and $k \in U_i$. Therefore, $U_i = \emptyset$, and all nodes are reachable from i. The converse statement *(i)* $\Longrightarrow$ *(ii)* is proved similarly.

(i) $\Longrightarrow$ *(iii)* If G is strongly connected, then there exists a directed path of length $k \leq n-1$ connecting any node i to any other node j. Hence, by Lemma 1.32(ii), the entry $(A^k)_{ij}$ is strictly positive. This immediately implies the statement (iii). The converse statement *(iii)* $\Longrightarrow$ *(i)* is proved similarly. ■

Next, we present a useful number theory result. This states that relatively co-prime numbers generate all sufficiently large natural numbers.

Lemma 1.83 (Natural number combination). *Let $a_1, \dots, a_N \in \mathbb{N}$ have greatest common divisor 1. There exists $k \in \mathbb{N}$ such that every number $m > k$ can be written as*

$$m = \alpha_1 a_1 + \cdots + \alpha_N a_N,$$

for appropriate numbers $\alpha_1, \dots, \alpha_N \in \mathbb{N}$.

Proof. Assume that $a_1 \leq \cdots \leq a_N$ without loss of generality. From the generalized Bezout identity we know that, for any numbers $a_1, \dots, a_N$ with greatest common divisor 1, there exist integers $\gamma_1, \dots, \gamma_N \in \mathbb{Z}$ such that

$$1 = \gamma_1 a_1 + \cdots + \gamma_N a_N. \tag{1.8.1}$$

Pick $k = |\gamma_1| a_1^2 + \cdots + |\gamma_N| a_N^2 \in \mathbb{N}$. Every number $m > k$ can be written as

$$m = k + m_{\text{qtnt}} a_1 + m_{\text{rmndr}},$$

for appropriate numbers $m_{\text{qtnt}} \geq 0$ and $1 \leq m_{\text{rmndr}} < a_1$. Using the definition of k and equation (1.8.1), we write

$$\begin{aligned} m &= \big(|\gamma_1| a_1^2 + \cdots + |\gamma_N| a_N^2\big) + m_{\text{qtnt}} a_1 + m_{\text{rmndr}}(\gamma_1 a_1 + \cdots + \gamma_N a_N) \\ &= m_{\text{qtnt}} a_1 + (|\gamma_1| a_1 + m_{\text{rmndr}}\gamma_1) a_1 + \cdots + (|\gamma_N| a_N + m_{\text{rmndr}}\gamma_N) a_N. \end{aligned}$$

The proof is now completed by noting that each integer number $(|\gamma_1| a_1 + m_{\text{rmndr}}\gamma_1), \dots, (|\gamma_N| a_N + m_{\text{rmndr}}\gamma_N)$ is strictly positive, because $m_{\text{rmndr}} < a_1 \leq \cdots \leq a_N$. ■

Proof of Proposition 1.35. (i) $\Longrightarrow$ (ii) Pick any i. We claim that there exists a number $k(i)$ with the property that, for all $m > k(i)$, we have that $(A^m)_{ii}$ is positive, that is, there exists a directed path from i to i of length m for all m larger than a number $k(i)$. To show this claim, let $\{c_1, \dots, c_N\}$ be the set of the cycles of G and let $\{\ell_1, \dots, \ell_N\}$ be their lengths. Because G is aperiodic, Lemma 1.83 implies the existence of a number $h(\ell_1, \dots, \ell_N)$ such that any number larger than $h(\ell_1, \dots, \ell_N)$ is a linear combination of $\ell_1, \dots, \ell_N$ with natural numbers as coefficients. Because G is strongly connected, there exists a path γ of arbitrary length $\Gamma(i)$ that starts at i, contains a vertex of each of the cycles $c_1, \dots, c_N$, and terminates at i. Now, we claim that $k(i) = \Gamma(i) + h(\ell_1, \dots, \ell_N)$ has the desired property. Indeed, pick any number $m > k(i)$ and write it as $k = \Gamma(i) + \beta_1 \ell_1 + \cdots + \beta_N \ell_N$ for appropriate numbers $\beta_1, \dots, \beta_N \in \mathbb{N}$. A directed path from i to i of length m is constructed by attaching to the path γ the following cycles: β_1 times the cycle c_1, β_2 times the cycle c_2, ..., β_N times the cycle c_N. Finally, having proved the existence of $k(i)$ with the desired property, let K be the maximum $k(i)$

over all nodes i, and recall that $\operatorname{diam}(G)$ is the maximum pairwise distance between nodes. Clearly, A^M is positive for all $M > K + \operatorname{diam}(G)$.

(ii) $\implies$ *(i)* From Lemma 1.32 we know that $A^k > 0$ means that there are paths from every node to every other node of length k. Hence, the digraph G is strongly connected. Next, we prove aperiodicity. Because G is strongly connected, each node of G has at least one outgoing edge, that is, for all i, there exists at least one index j such that $a_{ij} > 0$. This fact implies that the matrix $A^{k+1} = AA^k$ is positive via the following simple calculation: $(A^{k+1})_{il} = \sum_{h=1}^{n} a_{ih}(A^k)_{hl} \geq a_{ij}(A^k)_{jl} > 0$. In summary, we have shown that, if A^k is positive for some k, then A^m is positive for all subsequent $m \geq k$. Therefore, there are cycles in G of any length greater than or equal to k, which means that G is aperiodic. ■

1.8.4 Proof of Theorem 1.37

Proof. We begin with statement (i). Let l_{ij}, for $i, j \in \{1, \dots, n\}$, be the entries of $L(G)$. Note that $l_{ii} = \sum_{j=1, j\neq i}^{n} a_{ij} \geq 0$ and $l_{ij} = -a_{ij} \leq 0$ for $i \neq j$. By the Geršgorin disks Theorem 1.2, we know that each eigenvalue of $L(G)$ belongs to at least one of the disks

$$\Big\{ z \in \mathbb{C} \;\Big|\; \|z - l_{ii}\|_{\mathbb{C}} \leq \sum_{j=1, j\neq i}^{n} |l_{ij}| \Big\} = \big\{ z \in \mathbb{C} \mid \|z - l_{ii}\|_{\mathbb{C}} \leq l_{ii} \big\}.$$

These disks contain the origin $\mathbf{0}_n$ and complex numbers with a positive real part. This concludes the proof of statement (i).

Regarding statement (ii), note that $D_{\text{out}}(G)$ is invertible because G is strongly connected. Define the two matrices $\tilde{A} = D_{\text{out}}(G)^{-1}A(G)$ and $\tilde{L} = D_{\text{out}}(G)^{-1}L(G)$, and note that they satisfy $\tilde{L} = I_n - \tilde{A}$. Since $D_{\text{out}}(G)$ is diagonal, the matrices $A(G)$ and $\tilde{A}$ have the same pattern of zeros and positive entries. This observation and the assumption that G is strongly connected imply that $\tilde{A}$ is nonnegative and irreducible. By the Perron–Frobenius Theorem 1.11, the spectral radius $\rho(\tilde{A})$ is a simple eigenvalue. Furthermore, one can verify that $\tilde{A}$ is row-stochastic (see Lemma 1.31), and therefore, its spectral radius is 1 (see Exercise E1.4). In summary, we conclude that 1 is a simple eigenvalue of $\tilde{A}$, that 0 is a simple eigenvalue of $\tilde{L}$, that $\tilde{L}$ has rank $n-1$, and that $L(G)$ has rank $n-1$.

Regarding statement (iii), we first prove that $\operatorname{rank}(L(G)) = n - 1$ implies the existence of a globally reachable vertex. By contradiction, let G contain no globally reachable vertex. Then, by Lemma 1.27, there exist two nonempty disjoint subsets $U_1, U_2 \subset V(G)$ without any out-neighbor. After a permutation of the vertices, the adjacency matrix can be partitioned into the blocks

$$A(G) = \begin{bmatrix} A_{11} & 0 & 0 \\ 0 & A_{22} & 0 \\ A_{31} & A_{32} & A_{33} \end{bmatrix}.$$

Here, A_{12} and A_{13} vanish because U_1 does not have any out-neighbor, and A_{21} and A_{23} vanish because U_2 does not have any out-neighbor. Note that $D_{11} - A_{11}$ and $D_{22} - A_{22}$ are the Laplacian matrices of the graphs defined by restricting G to the vertices in U_1 and in U_2, respectively. Therefore, the eigenvalue 0 has geometric multiplicity at least 2 for the matrix $D_{\text{out}}(G) - A(G)$. This contradicts the assumption that $\operatorname{rank}(L(G)) = n - 1$.

Next, still regarding statement (iii), we prove that the existence of a globally reachable vertex implies $\operatorname{rank}(L(G)) = n-1$. Without loss of generality, we assume that G contains self-loops at each node (so that D_{out} is invertible). Let R be the set of globally reachable vertices; let $r \in \{1, \dots, n\}$ be its cardinality. If $r = n$, then the graph is strongly connected and statement (ii) implies that $\operatorname{rank}(L(G)) = n - 1$. Therefore, assume that $r < n$. Renumber the vertices so that R is the set of the first r vertices. After this permutation, the adjacency matrix and the Laplacian matrix can be partitioned into the blocks

$$A(G) = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix}, \quad \text{and} \quad L(G) = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}.$$

Here, $A_{12} \in \mathbb{R}^{r\times(n-r)}$ vanishes, because there can be no out-neighbor of R; otherwise that out-neighbor would be a globally reachable vertex in $V \setminus R$. Note that the rank of $L_{11} \in \mathbb{R}^{r\times r}$ is exactly $r-1$, since the digraph associated to A_{11} is strongly connected. To complete the proof it suffices to show that the rank of $L_{22} \in \mathbb{R}^{(n-r)\times(n-r)}$ is full. Note that the same block partition applies to the matrices $\tilde{A} = D_{\text{out}}^{-1} A$ and $\tilde{L} = D_{\text{out}}^{-1} L$ considered in the proof of statement (ii) above. With this block decomposition, we compute

$$\tilde{A}^{n-1} = \begin{bmatrix} \tilde{A}_{11}^{n-1} & 0 \\ \tilde{A}_{21}(n-1) & \tilde{A}_{22}^{n-1} \end{bmatrix},$$

for some matrix $\tilde{A}_{21}(n-1)$ that depends upon $\tilde{A}_{11}$, $\tilde{A}_{21}$ and $\tilde{A}_{22}$. Because a globally reachable node in G is globally reachable also in the digraph associated to $\tilde{A}$, Proposition 1.33(v) implies that $\tilde{A}_{21}(n-1)$ is positive. This fact, combined with the fact that $\tilde{A}$ and hence $\tilde{A}^{n-1}$ are row-stochastic, implies that $\tilde{A}_{22}^{n-1}$ has maximal row sum (that is, ∞-induced norm) strictly less than 1. Hence, the spectral radii of $\tilde{A}_{22}^{n-1}$ and of $\tilde{A}_{22}$ are strictly less than 1. Since $\tilde{A}_{22}$ has spectral radius strictly less than 1, the matrix $\tilde{L}_{22} = I_{n-r} - \tilde{A}_{22}$, and in turn the matrix L_{22}, have full rank.

Regarding statement (iv), the equivalence between (iv)a and (iv)b is proved as follows. Because $\sum_{j=1}^{n} l_{ij} = d_{\text{out}}(v_i) - d_{\text{in}}(v_i)$ for all $i \in \{1, \dots, n\}$, it follows that $\mathbf{1}_n^T L(G) = \mathbf{0}_n^T$ if and only if $D_{\text{out}}(G) = D_{\text{in}}(G)$. Next, we prove that (iv)b implies (iv)c. Suppose that $L(G)^T \mathbf{1}_n = \mathbf{0}_n^T$ and consider the system $\dot{\gamma}(t) = -L(G)\gamma(t)$, $\gamma(0) = x_0$, together with the positive definite function $V : \mathbb{R}^n \to \mathbb{R}$ defined by $V(x) = x^T x$. We compute the Lie derivative of the function V along the vector field $x \mapsto -L(G)x$ as $\dot{V}(x) = -2x^T L(G)x$. Note that $\dot{V}(x) \leq 0$, for all $x \in \mathbb{R}^n$, is equivalent to $L(G) + L(G)^T \geq 0$. Because $\mathbf{1}_n^T L(G) = \mathbf{0}_n^T$ and $L(G)\mathbf{1}_n = \mathbf{0}_n$, it can immediately be established that $\exp(-L(G)t)$, $t \in \mathbb{R}$, is a doubly stochastic matrix. From Theorem 1.1, we know that if we let $\{P_\alpha\}$ be the set of $n \times n$ permutation matrices, then there exist time-dependent convex combination coefficients $\sum_\alpha \lambda_\alpha(t) = 1$, $\lambda_\alpha(t) \geq 0$, such that $\exp(-L(G)t) = \sum_\alpha \lambda_\alpha(t) P_\alpha$. By the convexity of V and its invariance under coordinate permutations, for any $x \in \mathbb{R}^n$, we have

$$\begin{aligned} V(\exp(-L(G)t)x) &= V(\sum_\alpha \lambda_\alpha(t) P_\alpha x) \\ &\leq \sum_\alpha \lambda_\alpha(t) V(P_\alpha x) = \sum_\alpha \lambda_\alpha(t) V(x) = V(x)\,. \end{aligned}$$

In other words, $V(\exp(-L(G)t)x) \leq V(x)$ for all $x \in \mathbb{R}^n$, which implies $\dot{V}(x) \leq 0$, for all $x \in \mathbb{R}^n$. Finally, we prove that (iv)c implies (iv)b. By assumption, $-x^T(L(G) + L(G)^T)x = -2x^T L(G)x \leq 0$ for all $x \in \mathbb{R}^n$. In particular, for any small $\varepsilon > 0$ and $x = \mathbf{1}_n - \varepsilon L(G)^T \mathbf{1}_n$,

$$-(\mathbf{1}_n^T - \varepsilon \mathbf{1}_n^T L(G))L(G)(\mathbf{1}_n - \varepsilon L(G)^T \mathbf{1}_n) = \varepsilon \|L(G)^T \mathbf{1}_n\|_2^2 + O(\varepsilon^2) \leq 0,$$

which is possible only if $L(G)^T \mathbf{1}_n = \mathbf{0}_n^T$. ■

1.8.5 Proofs of Theorem 1.63 and Proposition 1.67

In this section, we prove Theorem 1.63. The exposition follows along the main lines of the original proof by Moreau (2005), with the variation of using the LaSalle Invariance Principle for set-valued dynamical systems, presented as Theorem 1.21. We begin with some preliminary results.

Lemma 1.84 (Union of digraphs and sums of adjacency matrices). *Let $G_1, \dots, G_\delta$ be unweighted digraphs with common node set $\{1, \dots, n\}$ and adjacency matrices $A_1, \dots, A_\delta$. The unweighted digraph*

$$G_1 \cup \dots \cup G_\delta = (\{1, \dots, n\}, E(A_1) \cup \dots \cup E(A_\delta))$$

is equal to the unweighted digraph associated to the nonnegative matrix $\sum_{k \in \{1,\dots,\delta\}} A_k$, that is, the unweighted digraph $(\{1, \dots, n\}, E(A_1 + \dots + A_\delta))$.

Proof. If $(i, j) \in \cup_{k \in \{1,\dots,\delta\}} E(G_k)$, then there exists $k_0 \in \{1, \dots, \delta\}$ such that $(i,j) \in E(G_{k_0})$. Denoting the entries of the matrix A_k by $a_{ij}(k)$, this implies that $a_{ij}(k_0) > 0$, that $a_{ij}(1) + \cdots + a_{ij}(\delta) > 0$, and that (i,j) is an edge in $E(A_1 + \cdots + A_\delta)$. The converse statement is easily proved with an analogous reasoning. ■

In what follows, for $\alpha \in \,]0,1]$, let $\mathcal{F}(\alpha)$ denote the set of $n \times n$ stochastic matrices that are non-degenerate with respect to α. Given $\alpha \in \,]0,1]$ and $\delta \in \mathbb{N}$, define the sets $\mathcal{F}_\delta(\alpha) \subset \mathbb{R}^{n\times n}$ by

$$\begin{aligned}\mathcal{F}_\delta(\alpha) = \big\{F \in \mathcal{F}(\alpha^\delta) \mid \exists\, F_1,\dots,F_\delta \in \mathcal{F}(\alpha) \text{ such that } F = F_\delta \cdots F_1 \\ \text{and } G(F_1) \cup \cdots \cup G(F_\delta) \text{ contains a globally reachable node}\big\},\end{aligned}$$

or, equivalently by Proposition 1.33,

$$\begin{aligned}\mathcal{F}_\delta(\alpha) = \big\{F \in \mathcal{F}(\alpha^\delta) \mid \exists\, F_1,\dots,F_\delta \in \mathcal{F}(\alpha) \text{ such that } F = F_\delta \dots F_1 \\ \text{and a column of } (F_1 + \cdots + F_\delta)^n \text{ has positive entries}\big\}.\end{aligned}$$

Lemma 1.85 (Compact sets of stochastic matrices). *For $\alpha \in \,]0,1]$, the sets $\mathcal{F}(\alpha)$ and $\mathcal{F}_\delta(\alpha)$, $\delta \in \mathbb{N}$, are compact.*

Proof. All sets are clearly bounded. In Exercise E1.24, we invite the reader to prove that $\mathcal{F}(\alpha)$ is closed. Let us now prove that $\mathcal{F}_\delta(\alpha)$ is closed. Consider a matrix sequence $\{F(k) \mid k \in \mathbb{N}\} \subset \mathcal{F}_\delta(\alpha)$ convergent to some matrix F. Because $\mathcal{F}(\alpha^\delta)$ is closed, we establish that $F \in \mathcal{F}(\alpha^\delta)$. Because each matrix $F(k)$ belongs to $\mathcal{F}_\delta(\alpha)$, there exist matrices $F_1(k),\dots,F_\delta(k) \in \mathcal{F}(\alpha)$ such that $F(k) = F_\delta(k)\cdots F_1(k)$. We claim that there exists a sequence $k_l \in \mathbb{N}$, for $l \in \mathbb{N}$, such that, for all $s \in \{1,\dots,\delta\}$, the matrix sequences $F_s(k_l)$, $l \in \mathbb{N}$, are convergent. (To see this, note that $F_1(k)$ takes value in a compact set; hence it must have a convergent subsequence. Restrict $F_2(k)$ to the instants of time in the convergent subsequence for $F_1(k)$ and observe that it takes value in a compact set, etc.) Therefore, there exist matrices F_s, to which the matrix sequences $F_s(k_l)$, $l \in \mathbb{N}$, converge. Taking the limit as $l \to +\infty$ in the equality $F(k_l) = F_\delta(k_l)\cdots F_1(k_l)$, we establish that $F = F_\delta \cdots F_1$. Finally, it remains to be shown that a column of $B := (F_1 + \cdots + F_\delta)^n$ has positive entries. For $k \in \mathbb{N}$, define $B(k) = (F_1(k) + \cdots + F_\delta(k))^n$. Clearly, $B(k) \to B$ as $k \to +\infty$. By the definition of the sequence $F(k)$, each $B(k) = (b_{ij}(k))$ has the property that there exists $j_k \in \{1,\dots,n\}$ such that $b_{ij_k}(k) > 0$ for all $i \in \{1,\dots,n\}$. Since $\{1,\dots,n\}$ is a finite set, there exists $j_0 \in \{1,\dots,n\}$ that satisfies this property for an infinite subsequence of matrices $B(k_l)$, $l \in \mathbb{N}$. With some straightforward bookkeeping, we write

$$(B(k_l))_{ij_0} = \sum_{a_1,\dots,a_n=1}^{\delta} \sum_{h_1=1}^{n} \cdots \sum_{h_{n-1}=1}^{n} (F_{a_1}(k_l))_{ih_1} \cdots (F_{a_n}(k_l))_{h_{n-1}j_0}.$$

Note that, because $F_s(k) \in \mathcal{F}(\alpha)$, for $s \in \{1, \dots, \delta\}$, each nonzero entry $F_s(k)$ is lower bounded by $\alpha > 0$. Furthermore, each entry $(B(k_l))_{ij_0}$ is the sum of nonnegative terms, each of which is the product of n factors, each of which is lower bounded by α. Hence, because $(B(k_l))_{ij_0}$ is positive, it is also lower bounded by α^n. Since $\lim_{l \to +\infty} B(k_l) = B$, by the compactness of $[\alpha^n, 1] \cup \{0\}$, it must be that $B = (b_{ij})$ satisfies $b_{ji_0} \geq \alpha^n > 0$ for all $j \in \{1, \dots, n\}$. In particular, this implies that $F \in \mathcal{F}_\delta(\alpha)$ and then $\mathcal{F}_\delta(\alpha)$ is closed. ■

Finally, we are able to prove the equivalences in Theorem 1.63.

Proof of Theorem 1.63. First, we prove that (i) implies (ii). Suppose that for all durations $\delta \in \mathbb{N}$, there exists some $\ell_0 \in \mathbb{N}$ such that the digraph with edges $\cup_{s \in [\ell_0, \ell_0 + \delta]} E(F(s))$ does not contain a globally reachable node. By Lemma 1.27, there must exist a set of nodes $U_1, U_2 \subset \{1, \dots, n\}$ such that there are no out-going edges (i_1, j_1), with $i_1 \in U_1$, $i_1 \in \{1, \dots, n\} \setminus U_1$ or (i_2, j_2), with $j_2 \in U_2$, $i_2 \in \{1, \dots, n\} \setminus U_2$. Take any values $a, b \in \mathbb{R}$, $a \neq b$, and consider the initial conditions:

$$w_i(\ell_0) = \begin{cases} a, & i \in U_1, \\ b, & i \in U_2, \\ c \in \operatorname{co}(a, b), & i \in \{1, \dots, n\} \setminus (U_1 \cup U_2). \end{cases}$$

Because of the properties of U_1 and U_2, for all $\delta \in \mathbb{N}$, we must have

$$w_j(\ell_0 + \delta + 1) = \begin{cases} a, & j \in U_1, \\ b, & j \in U_2, \\ c \in \operatorname{co}(a, b), & j \in \{1, \dots, n\} \setminus (U_1 \cup U_2). \end{cases}$$

Because δ can be chosen arbitrarily large, one can easily construct a contradiction with the fact that $\operatorname{diag}(\mathbb{R}^n)$ is supposed to be uniformly globally attractive.

Next, we show that (ii) implies (i). Let $\alpha \in \,]0, 1]$ to be the scalar with respect to which the sequence is non-degenerate. Consider the set-valued discrete-time dynamical system $(\mathbb{R}^n, \mathbb{R}^n, T_{\alpha,\delta})$, with evolution map $T_{\alpha,\delta} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ defined by

$$T_{\alpha,\delta}(w) = \{Fw \mid F \in \mathcal{F}_\delta(\alpha)\}.$$

Because of this definition, any trajectory $w : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$ of the averaging algorithm (1.6.2) satisfies

$$w((k+1)\delta) \in T_{\alpha,\delta}(w(k\delta)), \quad k \in \mathbb{Z}_{\geq 0}.$$

In what follows, we intend to use the LaSalle Invariance Principle for set-valued discrete dynamical systems, presented as Theorem 1.21, to prove that $\lim_{\ell\to+\infty} \operatorname{dist}(w(k\ell), \operatorname{diag}(\mathbb{R}^n)) = 0$. This will then imply, by Lemma 1.24, the uniform attractivity statement in the theorem. In the following, we check the conditions of the theorem.

Closedness of the set-valued dynamical system. Consider a pair of vector sequences $\{x_k \mid k \in \mathbb{N}\}$ and $\{y_k \mid k \in \mathbb{N}\}$ in $\mathbb{R}^n$ such that $\lim_{k\to+\infty} x_k = x$, $\lim_{k\to+\infty} y_k = y$, and $y_k \in T_{\alpha,\delta}(x_k)$, for all $k \in \mathbb{N}$. We need to show that $y \in T_{\alpha,\delta}(x)$. By definition of $T_{\alpha,\delta}$ and because $y_k \in T_{\alpha,\delta}(x_k)$, there exists a sequence $\{F(k) \mid k \in \mathbb{N}\} \subseteq \mathcal{F}_\delta(\alpha)$ such that $F(k)x_k = y_k$, for all $k \in \mathbb{N}$. Furthermore, since $\mathcal{F}_\delta(\alpha)$ is compact by Lemma 1.85, there exists a subsequence $\{F(k_l) \mid l \in \mathbb{N}\}$ that is convergent to some $F \in \mathcal{F}_\delta(\alpha)$. The desired conclusion follows from

$$y = \lim_{l\to+\infty} y_{k_\ell} = \lim_{l\to+\infty} F(k_\ell) x_{k_\ell} = Fx.$$

Non-increasing Lyapunov function. Define the function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ by

$$V(x) = \max_{i\in\{1,\dots,n\}} x_i - \min_{i\in\{1,\dots,n\}} x_i.$$

Note that V is continuous. Pick any $x \in \mathbb{R}^n$ and any stochastic matrix $F \in \mathcal{F}_\delta(\alpha)$. Recall that $\|x\|_\infty = \max_{i\in\{1,\dots,n\}} |x_i|$, and that $\|F\|_\infty = 1$. Therefore, by the definition of the induced norm, $\|Fx\|_\infty \leq \|x\|_\infty$. Similarly, in components,

$$(Fx)_i = \sum_{j\in\{1,\dots,n\}} f_{ij} x_j \geq \Big(\sum_{j\in\{1,\dots,n\}} f_{ij} \Big) \min_{k\in\{1,\dots,n\}} x_k,$$

which implies $\min_{i\in\{1,\dots,n\}} (Fx)_i \geq \min_{k\in\{1,\dots,n\}} x_k$. Therefore, we have that $V(Fx) \leq V(x)$ for all $x \in \mathbb{R}^n$ and $F \in \mathcal{F}_\delta(\alpha)$. In other words, the function V is non-increasing along $T_{\alpha,\delta}$ in $\mathbb{R}^n$.

Boundedness. It can immediately be seen that, since $\|Fx\|_\infty \leq \|x\|_\infty$ for all stochastic matrices F and vectors x, the trajectory $k \mapsto w(k\delta)$ is bounded.

Invariant set. By Theorem 1.21, any trajectory of $T_{\alpha,\delta}$, and hence also the trajectory $w : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$ of the averaging algorithm (1.6.2), will converge to the largest weakly positively invariant set contained in a level set of the Lyapunov function V and in a set where the Lyapunov function does not decrease along T. In the following, we determine that this set must be contained in $\operatorname{diag}(\mathbb{R}^n)$.

For $k \in \mathbb{N}$ fixed, assume that $w(k\delta)$ satisfies $V(w(k\delta)) > 0$. Given the averaging algorithm (1.6.2) defined by the sequence $\{F(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\} \subset \mathcal{F}(\alpha)$, define $F_1(k) = F(k+1), \dots, F_\delta(k) = F(k+\delta)$. Additionally, define $F(k) = F_\delta(k) \cdots F_1(k)$ and note that $F(k) \in \mathcal{F}_\delta(\alpha)$, by construction. With this notation, note that $w(k\delta + s) = F_s(k) \cdots F_1(k) w(k\delta)$ for $s \in \{1, \dots, \delta\}$. Define $w_M = \max_{i \in \{1,\dots,n\}} w_i(k\delta)$ and $w_m = \min_{i \in \{1,\dots,n\}} w_i(k\delta)$; by hypothesis we know $w_M > w_m$. Define $U_M = \{i \in \{1, \dots, n\} \mid w(k\delta) = w_M\}$ and $U_m = \{i \in \{1, \dots, n\} \mid w_j(k\delta) = w_m\}$; by hypothesis we know $U_M \cap U_m = \emptyset$. Now, we are ready to use property (ii) in the theorem statement. Since $(\{1, \dots, n\}, \cup_{s \in \{1,\dots,\delta\}} E(F_s(k))$ contains a globally reachable node and since U_M and U_m are nonempty and disjoint, then Lemma 1.27 implies that there exists either

- (an out-neighbor of U_M) an edge $(i_M, j_M) \in E(F_s(k\delta))$ with $i_M \in U_M$, $j_M \in \{1, \dots, n\} \setminus U_M$, and $s \in \{1, \dots, \delta\}$; or
- (an out-neighbor of U_m) an edge $(i_m, j_m) \in E(F_s(k\delta))$ with $i_m \in U_m$, $j_m \in \{1, \dots, n\} \setminus U_m$, and $s \in \{1, \dots, \delta\}$.

Without loss of generality, suppose that an edge (i_M, j_M) exists and let $s_0 \in \{1, \dots, \delta\}$ be the first time index for which this happens. We have the following two facts.

First, for every $s \in \{1, \dots, s_0 - 1\}$, there does not exist any edge (i, h) with $i \in U_M$ and $h \notin U_M$, and, thus, for all $i \in U_M$;

$$\begin{aligned} w_i(k\delta + 1) &= \sum_{j=1}^{n} (F_1(k))_{ij} w_j(k\delta) = \sum_{h \in U_M} (F_1(k))_{ih} w_h(k\delta) \\ &= \Big(\sum_{h \in U_M} (F_1(k))_{ih}(k) \Big) w_M = w_M. \end{aligned}$$

The same argument can be repeated for $F_2(k), \dots, F_s(k)$, so that $w_i(k\delta + s) = w_M$ for all $i \in U_M$.

Second, if $i \notin U_M$ at time $k\delta$, then $w_i(k\delta+s) < w_M$ for all $s \in \{1, \dots, s_0 - 1\}$. To see this, we compute

$$\begin{aligned} w_i(k\delta + 1) = \sum_{j=1}^{n} (F_1(k))_{ij} w_j(k\delta) &= (F_1(k))_{ii} w_i(k\delta) + \sum_{j=1, j\neq i}^{n} (F_1(k))_{ij} w_j(k\delta) \\ &\leq (F_1(k))_{ii} w_i(k\delta) + \Big(\sum_{j=1, j \neq i}^{n} (F_1(k))_{ij} \Big) w_M \\ &\leq \alpha w_i(k\delta) + (1 - \alpha) w_M < w_M, \end{aligned}$$

where we used the assumption of non-degeneracy with parameter $\alpha \in \,]0, 1]$.

The same argument can be repeated for the subsequent multiplications by the matrices $F_2(k), \dots, F_s(k)$.

We finally reach time s_0 and compute

$$
\begin{aligned}
w_{i_M}(k\delta + s_0) &= \sum_{j=1}^{n} (F_{s_0}(k))_{i_M j} w_j(k\delta + s_0 - 1) \\
&= (F_{s_0}(k))_{i_M j_M} w_{j_M}(k\delta + s_0 - 1) + \sum_{j=1, j \neq j_M}^{n} (F_{s_0}(k))_{i_M j} w_j(k\delta + s_0 - 1) \\
&< (F_{s_0}(k))_{i_M j_M} w_M + \sum_{j=1, j \neq j_M}^{n} (F_{s_0}(k))_{i_M j} w_j(k\delta + s_0 - 1) \leq w_M \,.
\end{aligned}
$$

This implies that $w_{i_M}((k+1)\delta) < w_M$, so that i_M does not belong to U_M at time $(k+1)\delta$. That is, the cardinality of U_M decreases at least by one after $(k+1)\delta$. Since $\{1, \dots, n\}$ is finite, after repeating this argument at most $n-1$ times, we have that U_M becomes empty at time $(k+n-1)\delta$. (Here we are assuming that the out-neighbor always exists for U_M; an analogous argument can be made for the general case.) This is enough to guarantee that $V(w((k+n)\delta)) < w_M - w_m = V(w(k\delta))$. This is what we need to conclude that $\lim_{k \to +\infty} \operatorname{dist}(w(k\delta), \operatorname{diag}(\mathbb{R})) = 0$. In summary, this concludes the proof of Theorem 1.63. ■

We conclude this section by establishing convergence to an individual point, rather than a set of points.

Proof of Proposition 1.67. We adopt the same notation as above, that is, as in the proof of Theorem 1.63. Since $F(k) \in \mathcal{F}_\delta(\alpha)$, the set of sequence points $\{w(k\delta) \mid k \in \mathbb{N}\}$ belongs to the convex hull of all the components of the initial condition, that is, $[\min_i w_i(0), \max_i w_i(0)]^n$. Since $[\min_i w_i(0), \max_i w_i(0)]^n$ is compact, there exists a convergent subsequence $\{w(k_l\delta) \mid l \in \mathbb{N}\}$ to a point $c\mathbf{1}_n$. We also notice that for any $k_l \in \mathbb{N}$, we have $w_i((k_l + k)\delta) \in [\min_i w_i(k_l\delta), \max_i w_i(k_l\delta)]^n$, for all $i \in \{1, \dots, n\}$ and $k \in \mathbb{N}$. Because $\lim_{l \to +\infty} w(k_l\delta) = c\mathbf{1}_n$ we know that $\lim_{l \to +\infty} [\min_i w_i(k_l\delta), \max_i w_i(k_l\delta)]^n = c\mathbf{1}_n$. Therefore, any sequence $\{w((k_l + k)\delta) \mid k \in \mathbb{N}\}$, for $l \in \mathbb{N}$, must converge to $c\mathbf{1}_n$. This implies that $\lim_{k \to +\infty} w(k\delta) = c\mathbf{1}_n$. ■

1.8.6 Proofs of Theorems 1.79 and 1.80

Proof of Theorem 1.79. Let us prove fact (i). Because $\operatorname{Trid}_n(a,b,a)$ is a real symmetric matrix, $\operatorname{Trid}_n(a,b,a)$ is normal and its 2-induced norm—that is, its largest singular value—is equal to the magnitude of its eigenvalue with the largest magnitude. Based on this information and on the eigenvalue computation in Lemma 1.77, we compute

$$\begin{aligned}\|\operatorname{Trid}_n(a,b,a)\|_2 &= \max_{i\in\{1,\dots,n\}} \left| b + 2a\cos\left(\frac{i\pi}{n+1}\right)\right| \\ &\leq |b| + 2|a| \max_{i\in\{1,\dots,n\}} \left|\cos\left(\frac{i\pi}{n+1}\right)\right| \leq |b| + 2|a|\cos\left(\frac{\pi}{n+1}\right).\end{aligned}$$

Because we assumed that $|b|+2|a| = 1$ and because $\cos(\frac{\pi}{n+1}) < 1$ for any $n \geq 2$, the 2-induced norm of $\operatorname{Trid}_n(a,b,a)$ is strictly less than 1. Additionally, for $\ell > 0$, we bound from above the magnitude of the curve x, as

$$\|x(\ell)\|_2 = \|\operatorname{Trid}_n(a,b,a)^\ell x_0\|_2 \leq \left(|b| + 2|a|\cos\left(\frac{\pi}{n+1}\right)\right)^\ell \|x_0\|_2.$$

In order to have $\|x(\ell)\|_2 < \varepsilon\|x_0\|_2$, it is sufficient to require that $\log\varepsilon > \ell \log\left(|b| + 2|a|\cos\left(\frac{\pi}{n+1}\right)\right)$, that is,

$$\ell > \frac{\log \varepsilon^{-1}}{-\log\left(|b| + 2|a|\cos\left(\frac{\pi}{n+1}\right)\right)}. \tag{1.8.2}$$

The upper bound now follows by noting that, as $t \to 0$, we have

$$-\frac{1}{\log(1 - 2|a|(1-\cos t))} = \frac{1}{|a|t^2} + O(1).$$

Let us now show the lower bound. Assume, without loss of generality, that $ab > 0$ and consider the eigenvalue $b + 2a\cos(\frac{\pi}{n+1})$ of $\operatorname{Trid}_n(a,b,a)$. Note that $|b + 2a\cos(\frac{\pi}{n+1})| = |b| + 2|a|\cos(\frac{\pi}{n+1})$. (If $ab < 0$, then consider the eigenvalue $b + 2a\cos(\frac{n\pi}{n+1})$.) For $n > 2$, define the unit-length vector

$$\mathbf{v}_n = \sqrt{\frac{2}{n+1}} \begin{bmatrix} \sin\frac{\pi}{n+1} \\ \vdots \\ \sin\frac{n\pi}{n+1}\end{bmatrix} \in \mathbb{R}^n, \tag{1.8.3}$$

and note that, by Lemma 1.77(i), $\mathbf{v}_n$ is an eigenvector of $\operatorname{Trid}_n(a,b,a)$ with eigenvalue $b + 2a\cos(\frac{\pi}{n+1})$. The trajectory x with initial condition $\mathbf{v}_n$ satisfies $\|x(\ell)\|_2 = \left(|b| + 2|a|\cos\left(\frac{\pi}{n+1}\right)\right)^\ell \|\mathbf{v}_n\|_2$, and therefore, it will enter $B(\mathbf{1}_n, \varepsilon\|\mathbf{v}_n\|_2)$ only when ℓ satisfies equation (1.8.2). This completes the proof of fact (i).

Next, we prove fact (ii). Clearly, all eigenvalues of the matrix $\mathrm{Trid}_n(a,b,0)$ are strictly inside the unit disk. For $\ell > 0$, we compute

$$\begin{aligned}&\mathrm{Trid}_n(a,b,0)^\ell\\ &\quad= b^\ell\Big(I_n + \frac{a}{b}\,\mathrm{Trid}_n(1,0,0)\Big)^\ell = b^\ell \sum_{j=0}^{n-1} \frac{\ell!}{j!(\ell-j)!}\Big(\frac{a}{b}\Big)^j\,\mathrm{Trid}_n(1,0,0)^j,\end{aligned}$$

because of the nilpotency of $\mathrm{Trid}_n(1,0,0)$. Now, we can bound from above the magnitude of the curve x, as

$$\begin{aligned}&\|x(\ell)\|_2 = \|\,\mathrm{Trid}_n(a,b,0)^\ell x_0\|_2\\ &\quad\le |b|^\ell \sum_{j=0}^{n-1} \frac{\ell!}{j!(\ell-j)!}\Big(\frac{a}{b}\Big)^j \big\|\,\mathrm{Trid}_n(1,0,0)^j x_0\big\|_2 \le \mathrm{e}^{a/b}\ell^{n-1}\,|b|^\ell\,\|x_0\|_2.\end{aligned}$$

Here, we used $\|\,\mathrm{Trid}_n(1,0,0)^j x_0\|_2 \le \|x_0\|_2$ and $\max\{\frac{\ell!}{(\ell-j)!} \mid j \in \{0,\dots,n-1\}\} \le \ell^{n-1}$. Therefore, in order to have $\|x(\ell)\|_2 < \varepsilon\|x_0\|_2$, it suffices that $\log(\mathrm{e}^{a/b}) + (n-1)\log\ell + \ell\log|b| \le \log\varepsilon$, that is,

$$\ell - \frac{n-1}{-\log|b|}\log\ell > \frac{\frac{a}{b} - \log\varepsilon}{-\log|b|}.$$

A sufficient condition for $\ell - \alpha\log\ell > \beta$, for $\alpha,\beta > 0$, is that $\ell \ge 2\beta + 2\alpha\max\{1,\log\alpha\}$. For, if $\ell \ge 2\alpha$, then $\log\ell$ is bounded from above by the line $\ell/2\alpha + \log\alpha$. Furthermore, the line $\ell/2\alpha + \log\alpha$ is a lower bound for the line $(\ell-\beta)/\alpha$ if $\ell \ge 2\beta + 2\alpha\log\alpha$. In summary, it is true that $\|x(\ell)\|_2 \le \varepsilon\|x(0)\|_2$ whenever

$$\ell \ge 2\frac{\frac{a}{b} - \log\varepsilon}{-\log|b|} + 2\frac{n-1}{-\log|b|}\max\Big\{1, \log\frac{n-1}{-\log|b|}\Big\}.$$

This completes the proof of the upper bound, that is, fact (ii).

The proof of fact (iii) is similar to that of fact (i). Because $\mathrm{Circ}_n(a,b,c)$ is circulant, it is also normal and each of its singular values corresponds to an eigenvector–eigenvalue pair. From Lemma 1.77(ii) and from the assumption $a+b+c=1$, it is clear that the eigenvalue corresponding to $i=n$ is equal to 1; this is the largest singular value of $\mathrm{Circ}_n(a,b,c)$ and the corresponding eigenvector is $\mathbf{1}_n$. We now compute the second largest singular value:

$$\begin{aligned}&\max_{i\in\{1,\dots,n-1\}} \Big\| b + (a+c)\cos\Big(\frac{i2\pi}{n}\Big) + \sqrt{-1}(c-a)\sin\Big(\frac{i2\pi}{n}\Big)\Big\|_{\mathbb{C}}\\ &\qquad = \Big\| 1 - (a+c)\Big(1 - \cos\big(\frac{2\pi}{n}\big)\Big) + \sqrt{-1}(c-a)\sin\Big(\frac{2\pi}{n}\Big)\Big\|_{\mathbb{C}}.\end{aligned}$$

Here, $\|\cdot\|_{\mathbb{C}}$ is the norm in $\mathbb{C}$. Because of the assumptions on a,b,c, the second largest singular value is strictly less than 1. In the orthogonal decomposition induced by the eigenvectors of $\mathrm{Circ}_n(a,b,c)$, we assume that the vector y_0 has a component y_{ave} along the eigenvector $\mathbf{1}_n$. For $\ell > 0$, we bound the distance of the curve $y(\ell)$ from $y_{\mathrm{ave}}\mathbf{1}_n$ as

$$\begin{aligned}
&\|y(\ell) - y_{\mathrm{ave}}\mathbf{1}_n\|_2 \\
&\quad = \| \,\mathrm{Circ}_n(a,b,c)^{\ell} y_0 - y_{\mathrm{ave}}\mathbf{1}_n\|_2 = \| \,\mathrm{Circ}_n(a,b,c)^{\ell}\big(y_0 - y_{\mathrm{ave}}\mathbf{1}_n\big)\|_2 \\
&\quad \leq \Big\| 1 - (a+c)\Big(1 - \cos\big(\frac{2\pi}{n}\big)\Big) + \sqrt{-1}(c-a)\sin\Big(\frac{2\pi}{n}\Big)\Big\|_{\mathbb{C}}^{\ell} \|y_0 - y_{\mathrm{ave}}\mathbf{1}_n\|_2 .
\end{aligned}$$

This proves that $\lim_{\ell\to+\infty} y(\ell) = y_{\mathrm{ave}}\mathbf{1}_n$. Also, for $\alpha = a + c, \beta = c - a$ and as $t \to 0$, we have

$$-\frac{1}{\log\Big(\big(1-\alpha(1-\cos t)\big)^2 + \beta^2\sin^2 t\Big)^{1/2}} = \frac{2}{(\alpha-\beta^2)t^2} + O(1).$$

Here, $\beta^2 < \alpha$ because $a, c \in \,]0,1[$. From this, one deduces the upper bound in (iii).

Now, consider the eigenvalues $\lambda_n = b+(a+c)\cos\big(\frac{2\pi}{n}\big)+\sqrt{-1}(c-a)\sin\big(\frac{2\pi}{n}\big)$ and $\overline{\lambda}_n = b+(a+c)\cos\Big(\frac{(n-1)2\pi}{n}\Big)+\sqrt{-1}(c-a)\sin\Big(\frac{(n-1)2\pi}{n}\Big)$ of $\mathrm{Circ}_n(a,b,c)$, and its associated eigenvectors (cf. Lemma 1.77(ii))

$$\mathbf{v}_n = \begin{bmatrix} 1 \\ \omega \\ \vdots \\ \omega^{n-1} \end{bmatrix} \in \mathbb{C}^n, \quad \overline{\mathbf{v}}_n = \begin{bmatrix} 1 \\ \omega^{n-1} \\ \vdots \\ \omega \end{bmatrix} \in \mathbb{C}^n. \tag{1.8.4}$$

Note that the vector $\mathbf{v}_n + \overline{\mathbf{v}}_n$ belongs to $\mathbb{R}^n$. Moreover, its component y_{ave} along the eigenvector $\mathbf{1}_n$ is zero. The trajectory y with initial condition $\mathbf{v}_n+\overline{\mathbf{v}}_n$ satisfies $\|y(\ell)\|_2 = \|\lambda_n^{\ell}\mathbf{v}_n + \overline{\lambda}_n^{\ell}\overline{\mathbf{v}}_n\|_2 = |\lambda_n|^{\ell}\|\mathbf{v}_n+\overline{\mathbf{v}}_n\|_2$, and therefore it will enter $B(\mathbf{0}_n, \varepsilon\|\mathbf{v}_n+\overline{\mathbf{v}}_n\|_2)$ only when

$$\ell > \frac{\log\varepsilon^{-1}}{-\log\Big\|1-(a+c)\Big(1-\cos\big(\frac{2\pi}{n}\big)\Big)+\sqrt{-1}(c-a)\sin\big(\frac{2\pi}{n}\big)\Big\|_{\mathbb{C}}}.$$

This completes the proof of fact (iii). ■

Proof of Theorem 1.80. We prove fact (i) and observe that the proof of fact (ii) is analogous. Consider the change of coordinates

$$x(\ell) = P_+\begin{bmatrix} x'_{\mathrm{ave}}(\ell) \\ y(\ell) \end{bmatrix} = x'_{\mathrm{ave}}(\ell)\mathbf{1}_n + P_+\begin{bmatrix} 0 \\ y(\ell) \end{bmatrix},$$

where $x'_{\mathrm{ave}}(\ell) \in \mathbb{R}$ and $y(\ell) \in \mathbb{R}^{n-1}$. A quick calculation shows that $x'_{\mathrm{ave}}(\ell) = \frac{1}{n}\mathbf{1}_n^T x(\ell)$, and the similarity transformation described in equation (1.6.7) implies

$$y(\ell+1) = \mathrm{Trid}_{n-1}(a,b,a)\,y(\ell), \quad \text{and} \quad x'_{\text{ave}}(\ell+1) = (b+2a)x'_{\text{ave}}(\ell).$$

Therefore, $x_{\text{ave}} = x'_{\text{ave}}$. It is also clear that

$$\begin{aligned} &x(\ell+1) - x_{\text{ave}}(\ell+1)\mathbf{1}_n \\ &\quad = P_+ \begin{bmatrix} 0 \\ y(\ell+1) \end{bmatrix} = \Big(P_+ \begin{bmatrix} 0 & 0 \\ 0 & \mathrm{Trid}_{n-1}(a,b,a) \end{bmatrix} P_+^{-1} \Big) (x(\ell) - x_{\text{ave}}(\ell)\mathbf{1}_n). \end{aligned}$$

Consider the matrix in parentheses determining the trajectory $\ell \mapsto (x(\ell) - x_{\text{ave}}(\ell)\mathbf{1}_n)$. This matrix is symmetric, its singular values are 0 and the singular values of $\mathrm{Trid}_{n-1}(a,b,a)$, and its eigenvectors are $\mathbf{1}_n$ and the eigenvectors of $\mathrm{Trid}_{n-1}(a,b,a)$ (padded with an extra zero). These facts are sufficient to duplicate, step by step, the proof of fact (i) in Theorem 1.79. Therefore, the trajectory $\ell \mapsto (x(\ell) - x_{\text{ave}}(\ell)\mathbf{1}_n)$ satisfies the stated properties. ■

1.9 EXERCISES

E1.1 **(Orthogonal and permutation matrices).** Prove that

(i) the set of orthogonal matrices is a group;

(ii) the set of permutation matrices is a group; and

(iii) each permutation matrix is orthogonal.

E1.2 **(Doubly stochastic matrices).** Show that the set of doubly stochastic matrices is convex and that it contains the set of permutation matrices. Find in the literature as many distinct proofs of Theorem 1.1 as possible.
Hint: *A proof is contained in Horn and Johnson (1985). A second proof method is based on methods from combinatorics.*

E1.3 **(Circulant matrices).** Given two $n \times n$ circulant matrices C_1 and C_2, show that the following hold:

(i) C_1^T, $C_1 + C_2$ and C_1C_2 are circulant; and

(ii) $C_1C_2 = C_2C_1$.

E1.4 **(Spectral radius and ∞-induced norm of a row-stochastic matrix).** Show that the spectral radius and the ∞-induced norm of a row-stochastic matrix are 1.
Hint: *Let $A \in \mathbb{R}^{d\times d}$ be stochastic. First, show $\|A\|_\infty \leq 1$ by direct algebraic manipulation. Second, use the bound in Lemma 1.5 to show that $\rho(A) \leq 1$. Finally, conclude the proof by noting that 1 is an eigenvalue of A.*
Hint: *An alternative proof that $\rho(A) = 1$ is as follows. First, use Geršgorin disks Theorem 1.2 to show that $\mathrm{spec}(A)$ is contained in the unit disk centered at the origin. Second, note that $\rho(A) \geq 1$, since 1 is an eigenvalue of A.*

E1.5 **(Positive semidefinite matrix defined by a doubly stochastic and irreducible matrix).** Let $A \in \mathbb{R}^{n\times n}$ be doubly stochastic and irreducible. Show that the matrix

$$I_n - A^T A$$

is positive semidefinite and that its eigenvalue 0 is simple.

E1.6 **(M-matrices).** This exercise summarizes some properties of the so-called M-matrices (see Fiedler, 1986). A matrix $A \in \mathbb{R}^{n\times n}$ is an *M-matrix* (resp. an *M_0-matrix*) if

(i) all the off-diagonal elements of A are zero or negative; and

(ii) there exist a nonnegative matrix $C \in \mathbb{R}^{n\times n}$ and $k > \rho(C)$ (resp. $k \geq \rho(C)$) such that $A = kI_n - C$.

Show that:

(i) the matrix $B \in \mathbb{R}^{n\times n}$ is an M-matrix if

(a) all the off-diagonal elements of B are zero or negative; and

(b) there exists a vector $v \in \mathbb{R}^n$ with positive entries such that Bv has positive entries;

(ii) if A is an M_0-matrix, irreducible and singular, then there exists $x \in \mathbb{R}^n$ with positive entries such that $Ax = 0$ and $\operatorname{rank}(A) = n - 1$; and

(iii) if A is an M-matrix, then all eigenvalues of A have positive real parts.

E1.7 **(Decomposition of a stochastic matrix).** Consider the matrix

$$T = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & -1 \\ \frac{1}{n} & \frac{1}{n} & \dots & \frac{1}{n} & \frac{1}{n} \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

Show that:

(i) T is invertible.

(ii) For a stochastic matrix $F \in \mathbb{R}^{n\times n}$, there exist $F_{\text{err}} \in \mathbb{R}^{(n-1)\times(n-1)}$ and $c_{\text{err}} \in \mathbb{R}^{1\times(n-1)}$ such that

$$TFT^{-1} = \begin{bmatrix} F_{\text{err}} & 0_{(n-1)\times 1} \\ c_{\text{err}} & 1 \end{bmatrix}.$$

Moreover, if F is symmetric, then $c_{\text{err}} = 0_{1\times(n-1)}$.

E1.8 This exercise establishes two extensions of the LaSalle Invariance Principle. Consider the same setup and assumptions as in Theorem 1.19, and remove the assumption that the set W is closed. Prove the following two conclusions.

(i) Each evolution with initial condition in W approaches a set of the form $V^{-1}(c) \cap (S \cup (\partial W \setminus W))$, where c is a real constant and S is the largest positively invariant set contained in $\{w \in W \mid V(f(w)) = V(w)\}$.

(ii) Each evolution $\gamma : \mathbb{Z}_{\geq 0} \to W$ with $\overline{\text{image}(\gamma)} \subset W$ approaches a set of the form $V^{-1}(c) \cap S$, where c is a real constant and S is the largest positively invariant set contained in $\{w \in W \mid V(f(w)) = V(w)\}$.

Hint: *Regarding part (i), follow the same steps as in the proof of Theorem 1.21 in Section 1.8.1 with the following difference: even though the set $\Omega(\gamma)$ is not a subset of W in general, the set $\Omega(\gamma) \cap W$ is a subset of W and is positively invariant.*

E1.9 **(The closed map defined by a finite collection of continuous maps).** Let $f_1, \dots, f_m : X \to X$ be continuous functions, where X is a d-dimensional space chosen among $\mathbb{R}^d$, $\mathbb{S}^d$, and the Cartesian products $\mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$, for some $d_1 + d_2 = d$. Define the set-valued map $T : X \rightrightarrows X$ by

$$T(x) = \{f_1(x), \dots, f_m(x)\}.$$

Show that T is closed on X.

E1.10 **(Overapproximation Lemma).** Prove Lemma 1.24.

E1.11 **(Acyclic digraphs).** Let G be an acyclic digraph. Show that:

(i) G contains at least one sink, that is, a vertex without out-neighbors;

(ii) G contains at least one source, that is, a vertex without in-neighbors; and

(iii) in an appropriate ordering of the vertices of G, the adjacency matrix A is lower-triangular, that is, all its entries above the main diagonal vanish.
Hint: *Order the vertices of G according to their distance to a sink.*

E1.12 **(A sufficient condition for a matrix to be primitive).** Show that if $A \in \mathbb{R}^{n \times n}$ is nonnegative, irreducible, and has a positive element on the diagonal, then A is primitive. Give an example that shows that this condition is sufficient but not necessary, that is, find a primitive matrix with no positive element on the diagonal.
Hint: *See Exercise E1.23 below for a candidate matrix.*

E1.13 **(Condensation digraph).** This exercise studies the decomposition of a digraph G in its strongly connected components. A subgraph H is a *strongly connected component* of G if H is strongly connected and any other subgraph of G strictly containing H is not strongly connected. The *condensation digraph* of G, denoted $C(G)$, is defined as follows: the nodes of $C(G)$ are the strongly connected components of G, and there exists a directed edge in $C(G)$ from node H_1 to node H_2 if and only if there exists a directed edge in G from a node of H_1 to a node of H_2. Show that:

(i) every condensation digraph is acyclic;

(ii) a digraph contains a globally reachable node if and only if its condensation digraph contains a globally reachable node; and

(iii) a digraph contains a directed spanning tree if and only if its condensation digraph contains a directed spanning tree.

E1.14 **(Incidence matrix).** Given a weighted digraph G of order n, choose an arbitrary ordering of its edges. Define the *incidence matrix* $H(G) \in \mathbb{R}^{|E| \times n}$ of G by specifying that the row of $H(G)$ corresponding to edge (i, j) has an entry 1 in column i, an entry -1 in column j, and all other entries equal to zero. Show that

$$H(G)^T W H(G) = L(G) + L(\mathrm{rev}(G)),$$

where $W \in \mathbb{R}^{|E| \times |E|}$ is the diagonal matrix with a_{ij} in the entry corresponding to edge (i, j).

E1.15 **(From digraphs to stochastic matrices and back).** Let G be a weighted digraph of order n with adjacency matrix A, out-degree matrix D_{out}, and Laplacian matrix L. Define the following matrices:

$$\begin{aligned} F_1 &= (\kappa I_n + D_{\text{out}})^{-1}(\kappa I_n + A), && \text{for } \kappa \in \mathbb{R}_{>0}, \\ F_2 &= I_n - \varepsilon L, && \text{for } \varepsilon \in [0, \min\{(D_{\text{out}})_{ii}^{-1} \mid i \in \{1, \dots, n\}\}[. \end{aligned}$$

Perform the following tasks:

(i) compute the entries of F_1 and F_2 as a function of the entries of $A(G)$;

(ii) show that the matrices F_1 and F_2 are row-stochastic;

(iii) identify the least restrictive conditions on G such that the matrices F_1 and F_2 are doubly stochastic; and

(iv) determine under what conditions a row-stochastic matrix can be written in the form F_1, or F_2 for some appropriate digraph (and for some appropriate scalars κ and ε).

E1.16 **(Metropolis–Hastings weights from the theory of Markov chains).** Given an undirected graph G of order n, define a weighted adjacency matrix A with entries

$$a_{ij} = \frac{1}{1 + \max\{|\mathcal{N}(i)|, |\mathcal{N}(j)|\}},$$

for $(i, j) \in E$. Perform the following tasks:

(i) show that the weighted degree of any vertex is strictly smaller than 1;

(ii) use (i) to justify that $\varepsilon = 1$ can be chosen in Exercise E1.15 for the construction of the matrix F_2; and

(iii) express the exponential convergence factor $r_{\exp}(F_2)$ as a function of the eigenvalues of the Laplacian of G.

E1.17 **(Some properties of products of stochastic matrices).** Show the following holds:

(i) If the matrices $A_1, \dots, A_k$ are nonnegative, row-stochastic, or doubly stochastic, respectively, then their product $A_1 \cdots A_k$ is non-negative, row-stochastic, or doubly stochastic, respectively.

(ii) If the nonnegative matrices $A_1, \dots, A_k$ have strictly positive diagonal elements, then their product $A_1 \cdots A_k$ has strictly positive diagonal elements.

(iii) Assume that $G_1, \dots, G_k$ are digraphs associated with the nonnegative matrices $A_1, \dots, A_k$ and that these matrices have strictly positive diagonal elements. If the digraph $G_1 \cup \ldots \cup G_k$ is strongly connected, then the matrix $A_1 \cdots A_k$ is irreducible.

E1.18 **(Disagreement function).** The *quadratic form* associated with a symmetric matrix $B \in \mathbb{R}^{n \times n}$ is the function $x \mapsto x^T B x$. Given a digraph G of order n, the *disagreement function* $\Phi_G : \mathbb{R}^n \to \mathbb{R}$ is defined by

$$\Phi_G(x) = \frac{1}{2} \sum_{i,j=1}^{n} a_{ij}(x_j - x_i)^2. \qquad \text{(E1.1)}$$

Show that the following are true:

(i) the disagreement function is the quadratic form associated with the symmetric positive-semidefinite matrix

$$P(G) = \frac{1}{2}(D_{\text{out}}(G) + D_{\text{in}}(G) - A(G) - A(G)^T);$$

(ii) $P(G) = \frac{1}{2}\big(L(G) + L(\text{rev}(G))\big)$.

E1.19 **(Weight-balanced graphs and connectivity).** Let G be a weighted digraph and let A be a nonnegative $n \times n$ matrix. Show the following statements:

(i) if G is weight-balanced and contains a globally reachable node, then it is strongly connected;

(ii) if A is doubly stochastic and its associated weighted digraph contains a globally reachable node, then its associated weighted digraph is strongly connected; and

(iii) if A is doubly stochastic and a column of $\sum_{k=0}^{n-1} A^k$ is positive, then $\sum_{k=0}^{n-1} A^k$ is positive.

E1.20 **(The Laplacian matrix is positive semidefinite).** Without relying on the Geršgorin disks Theorem 1.2, show that if the weighted digraph G is undirected, then the matrix $L(G)$ is symmetric positive semidefinite. (Note that the proof of statement (i) in Theorem 1.37 relies on Geršgorin disks Theorem 1.2).

E1.21 **(Properties of the BFS algorithm).** Prove Lemma 1.28.

E1.22 **(LCR algorithm).** Consider the following LCR algorithm for leader election:

(i) Give a UID assignment to each processor for which $\Omega(n^2)$ messages are sent; and

(ii) give a UID assignment to each processor for which only $O(n)$ messages are sent.

(iii) Show that the average number of messages sent is $O(n \log n)$, where the average is taken over all possible ordering of the processors on the ring, each ordering assumed to be equally likely.

E1.23 **(Properties of a stochastic matrix and its associated digraph).** Consider the stochastic matrices

$$A_1 = \frac{1}{2}\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad \text{and} \quad A_2 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

Define and draw the associated digraphs G_1 and G_2. Without relying on the characterization in Propositions 1.33 and 1.35, perform the following tasks:

(i) show that the matrices A_1 and A_2 are irreducible and that the associated digraphs G_1 and G_2 are strongly connected;

(ii) show that the matrices A_1 and A_2 are primitive and that the associated digraphs G_1 and G_2 are strongly connected and aperiodic; and

(iii) show that the averaging algorithm associated with A_2 converges in a finite number of steps.

E1.24 **(Compactness of the set of non-degenerate matrices with respect to a parameter).** Show that, for any $\alpha \in]0,1]$, the set of non-degenerate matrices with respect to α is compact.

E1.25 **(Laplacian flow: Olfati-Saber and Murray, 2004).** Let G be a weighted directed graph with a globally reachable node. Define the *Laplacian flow* on $\mathbb{R}^n$ by

$$\dot{x} = -L(G)x,$$

or, equivalently in components,

$$\dot{x}_i = \sum_{j \in \mathcal{N}^{\text{out}}(i)} a_{ij}(x_j - x_i), \quad i \in \{1, \dots, n\}.$$

Perform the following tasks:

(i) Find the equilibrium points of the Laplacian flow.

(ii) Show that, if G is undirected, then the disagreement function (see Exercise E1.18) is monotonically non-increasing along the Laplacian flow.

(iii) Given $x_0 = ((x_0)_1, \dots, (x_0)_n) \in \mathbb{R}^n$, show that the solution $t \mapsto x(t)$ of the Laplacian flow starting at x_0 verifies

$$\min\{(x_0)_1, \dots, (x_0)_n\} \leq x_i(t) \leq \max\{(x_0)_1, \dots, (x_0)_n\},$$

for all $t \in \mathbb{R}_{\geq 0}$. Use this fact to deduce that the solution $t \mapsto x(t)$ is bounded.

(iv) For G undirected, use (i)-(iii) to apply the LaSalle Invariance Principle in Theorem 1.20 and show that the solutions of the Laplacian flow converge to $\text{diag}(\mathbb{R}^n)$.

(v) Find an example G such that, with the notation in Exercise E1.18, the symmetric matrix $L(G)^T P(G) + P(G)L(G)$ is indefinite.
Hint: *To show that the matrix is indefinite, it suffices to find* $x_1, x_2 \in \mathbb{R}^n$ *such that* $x_1(L(G)^T P(G) + P(G)L(G))x_1 < 0$ *and* $x_2(L(G)^T P(G) + P(G)L(G))x_2 > 0$.

(vi) Show that the Euler discretization of the Laplacian flow is the Laplacian-based averaging algorithm.

E1.26 **(Log–Sum–Exp consensus: Tahbaz-Salehi and Jadbabaie, 2006).** Pick $\alpha \in \mathbb{R} \setminus \{0\}$ and define the function $f_\alpha : \mathbb{R}^n \to \mathbb{R}$ by

$$f_\alpha(x) = \alpha \log\Big(\frac{1}{n}\sum_{i=1}^{n} \mathrm{e}^{x_i/\alpha}\Big).$$

Show that:

(i) $\lim_{\alpha \to 0^-} f_\alpha(x) = \min\{x_1, \dots, x_n\}$ and $\lim_{\alpha \to 0^+} f_\alpha(x) = \max\{x_1, \dots, x_n\}$; and

(ii) $\lim_{\alpha \to +\infty} f_\alpha(x) = \lim_{\alpha \to -\infty} f_\alpha(x) = \dfrac{1}{n}(x_1 + \dots + x_n)$.

Next, let $A \in \mathbb{R}^{n \times n}$ be a non-degenerate, doubly stochastic matrix whose associated digraph contains a globally reachable node. Given such a matrix A, consider the discrete-time dynamical system

$$w_i(\ell+1) = \alpha \log\Big(\sum_{j=1}^{n} a_{ij}\, \mathrm{e}^{w_j(\ell)/\alpha}\Big).$$

(iii) Show that $w(\ell) \to f_\alpha(w(0))\mathbf{1}_n$ as $\ell \to +\infty$.

E1.27 **(The theory of Markov chains and random walks on graphs).** List as many connections as possible between the theory of averaging algorithms discussed in Section 1.6.2 and the theory of Markov chains. Some relevant references on Markov chains include Seneta (1981) and Lovász (1993).
Hint: *There is a one-to-one correspondence between averaging algorithms and Markov chains. A homogeneous Markov chains corresponds precisely to a time-independent averaging algorithm. A reversible Markov chain corresponds precisely to a symmetric stochastic matrix.*

E1.28 **(Distributed hypothesis testing: Rao and Durrant-Whyte, 1993; Olfati-Saber et al., 2006).** Let h_γ, for $\gamma \in \Gamma$ in a finite set Γ, be a set of alternative hypotheses about an uncertain event. Suppose that n nodes take measurements z_i, for $i \in \{1,\dots,n\}$, related to the event. Assume that each observation is conditionally independent of all other observations, given any hypothesis.

(i) Using Bayes' Theorem and the independence assumption, show that the *a posteriori probabilities* satisfy

$$p(h_\gamma|z_1,\dots,z_n) = \frac{p(h_\gamma)}{p(z_1,\dots,z_n)} \prod_{i=1}^{n} p(z_i|h_\gamma).$$

(ii) Suppose that the nodes form a undirected unweighted connected synchronous network with adjacency matrix A. Consider the discrete-time dynamical system

$$\pi_i(\ell+1) = \Big(\pi_i(\ell) \prod_{j=1}^{n} \pi_j^{a_{ij}}(\ell)\Big)^{1/(1+d_{\text{out}}(i))}.$$

Fix $\gamma \in \Gamma$, set $\pi_i(0) = p(z_i|h_\gamma)$, and show that $\pi(\ell) \to \sqrt[n]{\prod_{i=1}^{n} p(z_i|h_\gamma)}\mathbf{1}_n$ as $\ell \to +\infty$.

(iii) What information does each node need in order to compute the *maximum a posteriori estimate*, that is, to estimate the most likely hypothesis?
Hint: *Can you compute $p(z_1,\dots,z_n)$, given knowledge of $p(h_\gamma)$ and of $\prod_{i=1}^{n} p(z_i|h_\gamma)$?*

As a bibliographic note, the variable π_i is referred to as the *belief* in the seminal work by Pearl (1988).

E1.29 **(Bounds on vector norms).** Prove Lemma 1.82.

E1.30 **(The "n-bugs problem" and cyclic interactions).** The "n-bugs problem" related to the *pursuit curves* from mathematics, inquires about what the paths of n bugs, not aligned initially, are when they chase one another. Simple versions of the problem (e.g., for three bugs starting at the vertices of an equilateral triangle) were studied as early as the nineteenth century. It was in Watton and Kydon

(1969) that a general solution for the general n-bugs problem for non-collinear initial positions was given. The bugs trace out logarithmic spirals that eventually meet at the same point, and it is not necessary that they move with constant velocity. Surveys about cyclic pursuit problems are given in the papers in Watton and Kydon (1969) and Marshall et al. (2004). Cyclic pursuit, has also been studied recently in the multi-agent and control literature; see, for example Bruckstein et al. (1991), Marshall et al. (2004), and Smith et al. (2005). In particular, the paper Marshall et al. (2004) extends the n-bugs problem to the case of n kinematic unicycles evolving in continuous time.

Consider the simplified scenario of the n-bugs problem placed on a circle of radius r and suppose that the bugs' motion is constrained to be on that circle. Assume that agents are ordered counterclockwise with identities $i \in \{1, \dots, n\}$, where, for convenience, we identify $n+1$ with 1. Denote by $p_i(\ell) = (r, \theta_i(\ell))$ the sequence of positions of bug i, initially at $p_i(0) = (r, \theta_i(0))$. We illustrate two scenarios of interest in Figure E1.1 and we describe them in some detail below.

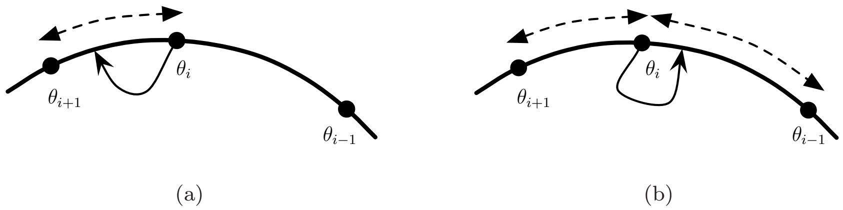


Figure E1.1 An illustration of the n-bugs problem. In (a), agent i looks at the position of agent $i+1$ and moves toward it by an amount proportional to their distance. In (b), agent i looks at the position of agents $i+1$ and $i-1$ and moves toward the one which is furthest by an amount proportional to the difference between the two distances. In both cases, the proportionality constant is k.

Cyclic pursuit. Suppose that each bug is chasing the closest counterclockwise neighbor (according to the order we have given them on the circle), see Figure E1.1(a). In other words, each bug feels an attraction toward the closest counterclockwise neighbor that can be described by the equation

$$\theta_i(\ell+1) = (1-k)\theta_i(\ell) + k\theta_{i+1}(\ell), \quad \ell \in \mathbb{Z}_{\geq 0},$$

where $k \in [0,1]$. Determine for which values of k the bugs converge to a configuration for which $\operatorname{dist}_{\mathsf{c}}(\theta_{i+1}, \theta_i) = \operatorname{dist}_{\mathsf{c}}(\theta_i, \theta_{i-1})$ for all $i \in \{1, \dots, n\}$. Observe that the bugs will approach this equally spaced configuration while moving around the circle indefinitely.

Cyclic balancing. Suppose that each bug makes a compromise between chasing its closest counterclockwise neighbor and the closest clockwise neighbor, see Figure E1.1(b). In other words, each bug feels an attraction towards the closest counterclockwise and clockwise neighbors that can be described by the equation

$$\theta_i(\ell+1) = k\theta_{i+1}(\ell) + (1-2k)\theta_i(\ell) + k\theta_{i-1}(\ell), \quad \ell \in \mathbb{Z}_{\geq 0},$$

where $k \in [0,1]$. Perform the following two tasks:

(i) Determine for which values of k the bugs converge to a configuration for which $\mathrm{dist}_{\mathsf{c}}(\theta_{i+1}, \theta_i) = \mathrm{dist}_{\mathsf{c}}(\theta_i, \theta_{i-1})$ for all $i \in \{1, \dots, n\}$.

(ii) Show that the bugs will approach this equally spaced configuration while each of them converges to a stationary position on the circle.

Hint: *Rewrite the cyclic pursuit and cyclic balancing systems in terms of the inter-bug distances, that is, in terms of $d_i(\ell) = \mathrm{dist}_{\mathsf{c}}(\theta_{i+1}(\ell), \theta_i(\ell))$, $i \in \{1, \dots, n\}$, $\ell \in \mathbb{Z}_{\geq 0}$. Find the matrices that describe the linear iterations in these new coordinates. Show that the agreement space, that is, the diagonal set in $\mathbb{R}^n$, is invariant under the dynamical systems. Finally, determine which values of k make each system converge to the agreement space. Lemma 1.77 might be of use in this regard. Regarding part (ii)b), recall that an exponentially decaying sequence is summable.*

Bibliography

Angeli, D. and Bliman, P.-A. [2006] *Stability of leaderless discrete-time multi-agent systems*, Mathematics of Control, Signals and Systems, **18**(4), 293–322.

Barbašin, E. A. and Krasovskiĭ, N. N. [1952] *On stability of motion in the large*, Doklady Akad. Nauk SSSR, **86**, 453–456, (In Russian).

Barooah, P. [2007] *Estimation and Control with Relative Measurements: Algorithms and Scaling Laws*, Ph.D. thesis, University of California at Santa Barbara.

Bauso, D., Giarré, L., and Pesenti, R. [2006] *Nonlinear protocols for optimal distributed consensus in networks of dynamic agents*, Systems & Control Letters, **55**(11), 918–928.

Bertsekas, D. P. and Tsitsiklis, J. N. [1997] *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, ISBN 1886529019.

Biggs, N. [1994] *Algebraic Graph Theory*, second edition, Cambridge University Press, ISBN 0521458978.

Blondel, V. D., Hendrickx, J. M., Olshevsky, A., and Tsitsiklis, J. N. [2005] *Convergence in multiagent coordination, consensus, and flocking*, in *IEEE Conf. on Decision and Control and European Control Conference*, pages 2996–3000, Seville, Spain.

Bruckstein, A. M., Cohen, N., and Efrat, A. [1991] *Ants, crickets, and frogs in cyclic pursuit*, Technical Report CIS 9105, Center for Intelligent Systems, Technion, Haifa, Israel, available at `http://www.cs.technion.ac.il/tech-reports`.

Cao, M. [2007] *Multi-Agent Formations and Sensor Networks*, Ph.D. thesis, Yale University.

Cao, M., Morse, A. S., and Anderson, B. D. O. [2008] *Reaching a consensus in a dynamically changing environment - convergence rates, measurement delays and asynchronous events*, SIAM Journal on Control and Optimization, **47**(2), 601–623.

Carli, R. [2008] *Topics in the Average Consensus Problems*, Ph.D. thesis, Universitá di Padova, Italy.

Carli, R., Bullo, F., and Zampieri, S. [2009] *Quantized average consensus via dynamic coding/decoding schemes*, International Journal on Robust and Nonlinear Control, (Submitted May 2008) to appear.

Carli, R., Fagnani, F., Speranzon, A., and Zampieri, S. [2008] *Communication constraints in the average consensus problem*, Automatica, **44**(3), 671–684.

Cassandras, C. G. and Lafortune, S. [2007] *Introduction to Discrete-Event Systems*, second edition, Springer, ISBN 0387333320.

Chatterjee, S. and Seneta, E. [1977] *Towards consensus: Some convergence theorems on repeated averaging*, Journal of Applied Probability, **14**(1), 89–97.

Chen, C.-T. [1984] *Linear System Theory and Design*, Holt, Rinehart, and Winston, ISBN 0030602890.

Chopra, N. and Spong, M. W. [2009] *On exponential synchronization of Kuramoto oscillators*, IEEE Transactions on Automatic Control, **54**(2), 353–357.

Cogburn, R. [1984] *The ergodic theory of Markov chains in random environments*, Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete, **66**(1), 109–128.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. [2001] *Introduction to Algorithms*, second edition, MIT Press, ISBN 0262032937.

Cortés, J. [2006] *Finite-time convergent gradient flows with applications to network consensus*, Automatica, **42**(11), 1993–2000.

— [2008a] *Discontinuous dynamical systems – a tutorial on solutions, nonsmooth analysis, and stability*, IEEE Control Systems Magazine, **28**(3), 36–73.

— [2008b] *Distributed algorithms for reaching consensus on general functions*, Automatica, **44**(3), 726–737.

Cucker, F. and Smale, S. [2007] *Emergent behavior in flocks*, IEEE Transactions on Automatic Control, **52**(5), 852–862.

Cybenko, G. [1989] *Dynamic load balancing for distributed memory multiprocessors*, Journal of Parallel and Distributed Computing, **7**(2), 279–301.

DeGroot, M. H. [1974] *Reaching a consensus*, Journal of the American Statistical Association, **69**(345), 118–121.

Diestel, R. [2005] *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*, second edition, Springer, ISBN 978-3-540-26182-7.

Dullerud, G. E. and Paganini, F. [2000] *A Course in Robust Control Theory*, number 36 in Texts in Applied Mathematics, Springer, ISBN 978-0-387-98945-7.

Durrett, R. [2006] *Random Graph Dynamics*, Series in Statistical and Probabilistic Mathematics, Cambridge University Press, ISBN 0521866561.

Fagnani, F. and Zampieri, S. [2009] *Average consensus with packet drop communication*, SIAM Journal on Control and Optimization, **48**(1), 102–133.

Fang, L. and Antsaklis, P. J. [2008] *Asynchronous consensus protocols using nonlinear paracontractions theory*, IEEE Transactions on Automatic Control, **53**, 2351–2355.

Ferrari-Trecate, G., Buffa, A., and Gati, M. [2006] *Analysis of coordination in multi-agent systems through partial difference equations*, IEEE Transactions on Automatic Control, **51**(6), 1058–1063.

Fiedler, M. [1986] *Special Matrices and their Applications in Numerical Mathematics*, Martinus Nijhoff Publishers, ISBN 9024729572.

Francis, B. A. [2006] *Distributed control of autonomous mobile robots*, course Notes, Version 1.01, University of Toronto, Canada.

Gallager, R. G. [1968] *Information Theory and Reliable Communication*, John Wiley, ISBN 0471290483.

Gallager, R. G., Humblet, P. A., and Spira, P. M. [1983] *A distributed algorithm for minimum-weight spanning trees*, ACM Transactions on Programming Languages and Systems, **5**(1), 66–77.

Godsil, C. D. and Royle, G. F. [2001] *Algebraic Graph Theory*, volume 207 of *Graduate Texts in Mathematics*, Springer, ISBN 0387952411.

Guckenheimer, J. and Holmes, P. [1990] *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, volume 42 of *Applied Mathematical Sciences*, Springer, ISBN 0387908196.

Hatano, Y. and Mesbahi, M. [2005] *Agreement over random networks*, IEEE Transactions on Automatic Control, **50**(11), 1867–1872.

Hendrickx, J. M. [2008] *Graphs and Networks for the Analysis of Autonomous Agent Systems*, Ph.D. thesis, Université Catholique de Louvain, Belgium.

Horn, R. A. and Johnson, C. R. [1985] *Matrix Analysis*, Cambridge University Press, ISBN 0521386322.

Huygens, C. [1673] *Horologium Oscillatorium*, Paris, France.

Igarashi, Y., Hatanaka, T., Fujita, M., and Spong, M. W. [2007] *Passivity-based 3D attitude coordination: Convergence and connectivity*, in *IEEE Conf. on Decision and Control*, pages 2558–2565, New Orleans, LA.

Jadbabaie, A., Lin, J., and Morse, A. S. [2003] *Coordination of groups of mobile autonomous agents using nearest neighbor rules*, IEEE Transactions on Automatic Control, **48**(6), 988–1001.

Jadbabaie, A., Motee, N., and Barahona, M. [2004] *On the stability of the Kuramoto model of coupled nonlinear oscillators*, in *American Control Conference*, pages 4296–4301, Boston, MA.

Kashyap, A., Başar, T., and Srikant, R. [2007] *Quantized consensus*, Automatica, **43**(7), 1192–1203.

Khalil, H. K. [2002] *Nonlinear Systems*, third edition, Prentice Hall, ISBN 0130673897.

Korte, B. and Vygen, J. [2005] *Combinatorial Optimization: Theory and Algorithms*, volume 21 of *Algorithmics and Combinatorics*, third edition, Springer, ISBN 3540256849.

Krasovskiĭ, N. N. [1963] *Stability of motion. Applications of Lyapunov's second method to differential systems and equations with delay*, Stanford University Press, translated by J. L. Brenner.

Kuramoto, Y. [1975] *Self-entrainment of a population of coupled non-linear oscillators*, in *International Symposium on Mathematical Problems in Theoretical Physics*, H. Araki, editor, volume 39 of *Lecture Notes in Physics*, pages 420–422, Springer, ISBN 978-3-540-07174-7.

— [1984] *Chemical oscillations, waves, and turbulence*, Springer, ISBN 0387133224.

Landau, H. J. and Odlyzko, A. M. [1981] *Bounds for eigenvalues of certain stochastic matrices*, Linear Algebra and its Applications, **38**, 5–15.

LaSalle, J. P. [1960] *Some extensions of Liapunov's second method*, IRE Trans. Circuit Theory, **CT-7**, 520–527.

— [1986] *The Stability and Control of Discrete Processes*, volume 62 of *Applied Mathematical Sciences*, Springer, ISBN 0387964118.

Lin, Z. [2005] *Coupled Dynamic Systems: From Structure Towards Stability and Stabilizability*, Ph.D. thesis, University of Toronto, Canada.

Lin, Z., Broucke, M., and Francis, B. [2004] *Local control strategies for groups of mobile autonomous agents*, IEEE Transactions on Automatic Control, **49**(4), 622–629.

Lin, Z., Francis, B., and Maggiore, M. [2005] *Necessary and sufficient graphical conditions for formation control of unicycles*, IEEE Transactions on Automatic Control, **50**(1), 121–127.

— [2007] *State agreement for continuous-time coupled nonlinear systems*, SIAM Journal on Control and Optimization, **46**(1), 288–307.

Lorenz, J. [2007] *Repeated Averaging and Bounded Confidence – Modeling, Analysis and Simulation of Continuous Opinion Dynamics*, Ph.D. thesis, University of Bremen, Germany, available at `http://nbn-resolving.de/urn:nbn:de:gbv:46-diss000106688`.

Lorenz, J. and Lorenz, D. A. [2008] *On conditions for convergence to consensus*, available at `http://arxiv.org/abs/0803.2211`.

Lovász, L. [1993] *Random walks on graphs: A survey*, in *Combinatorics: Paul Erdös is Eighty*, T. S. D. Miklós, V. T. Sós, editor, volume 2, pages 353–398, János Bolyai Mathematical Society, ISBN 9638022744.

Lynch, N. A. [1997] *Distributed Algorithms*, Morgan Kaufmann, ISBN 1558603484.

Marshall, J. A., Broucke, M. E., and Francis, B. A. [2004] *Formations of vehicles in cyclic pursuit*, IEEE Transactions on Automatic Control, **49**(11), 1963–1974.

Martínez, S., Bullo, F., Cortés, J., and Frazzoli, E. [2007] *On synchronous robotic networks – Part I: Models, tasks and complexity*, IEEE Transactions on Automatic Control, **52**(12), 2199–2213.

Merris, R. [1994] *Laplacian matrices of a graph: A survey*, Linear Algebra its Applications, **197**, 143–176.

Meyer, C. D. [2001] *Matrix Analysis and Applied Linear Algebra*, SIAM, ISBN 0898714540.

Meyn, S. and Tweedie, R. [1999] *Markov Chains and Stochastic Stability*, Springer, ISBN 3540198326.

Mirollo, R. E. and Strogatz, S. H. [1990] *Synchronization of pulse-coupled biological oscillators*, SIAM Journal on Applied Mathematics, **50**(6), 1645–1662.

Mohar, B. [1991] *The Laplacian spectrum of graphs*, in *Graph Theory, Combinatorics, and Applications*, Y. Alavi, G. Chartrand, O. R. Oellermann, and A. J. Schwenk, editors, volume 2, pages 871–898, John Wiley, ISBN 0471532452.

Moreau, L. [2003] *Time-dependent unidirectional communication in multi-agent systems*, available at http://arxiv.org/abs/math/0306426.

— [2004] *Stability of continuous-time distributed consensus algorithms*, available at http://arxiv.org/abs/math/0409010.

— [2005] *Stability of multiagent systems with time-dependent communication links*, IEEE Transactions on Automatic Control, **50**(2), 169–182.

Nijmeijer, H. [2001] *A dynamical control view on synchronization*, Physica D, **154**(3-4), 219–228.

Olfati-Saber, R. [2005] *Ultrafast consensus in small world networks*, in *American Control Conference*, pages 2371–2378, Portland, OR.

Olfati-Saber, R., Fax, J. A., and Murray, R. M. [2007] *Consensus and cooperation in networked multi-agent systems*, Proceedings of the IEEE, **95**(1), 215–233.

Olfati-Saber, R., Franco, E., Frazzoli, E., and Shamma, J. S. [2006] *Belief consensus and distributed hypothesis testing in sensor networks*, in *Network Embedded Sensing and Control. (Proceedings of NESC'05 Worskhop)*, P. Antsaklis and P. Tabuada, editors, volume 331 of *Lecture Notes in Control and Information Sciences*, pages 169–182, Springer, ISBN 3540327940.

Olfati-Saber, R. and Murray, R. M. [2004] *Consensus problems in networks of agents with switching topology and time-delays*, IEEE Transactions on Automatic Control, **49**(9), 1520–1533.

Olshevsky, A. and Tsitsiklis, J. N. [2009] *Convergence speed in distributed consensus and averaging*, SIAM Journal on Control and Optimization, **48**(1), 33–55.

Papachristodoulou, A. and Jadbabaie, A. [2006] *Synchronization in oscillator networks with heterogeneous delays, switching topologies and nonlinear dynamics*, in *IEEE Conf. on Decision and Control*, pages 4307–4312, San Diego, CA.

Patterson, S., Bamieh, B., and Abbadi, A. E. [2007] *Distributed average consensus with stochastic communication failures*, in *IEEE Conf. on Decision and Control*, pages 4215–4220, New Orleans, LA.

Pearl, J. [1988] *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, ISBN 1558604790.

Peleg, D. [2000] *Distributed Computing. A Locality-Sensitive Approach*, Monographs on Discrete Mathematics and Applications, SIAM, ISBN 0898714648.

Picci, G. and Taylor, T. [2007] *Almost sure convergence of random gossip algorithms*, in *IEEE Conf. on Decision and Control*, pages 282–287, New Orleans, LA.

Porfiri, M. and Stilwell, D. J. [2007] *Consensus seeking over random weighted directed graphs*, IEEE Transactions on Automatic Control, **52**(9), 1767–1773.

Rao, B. S. Y. and Durrant-Whyte, H. F. [1993] *A decentralized Bayesian algorithm for identification of tracked targets*, IEEE Transactions on Systems, Man, & Cybernetics, **23**(6), 1683–1698.

Ren, W. and Beard, R. W. [2005] *Consensus seeking in multi-agent systems under dynamically changing interaction topologies*, IEEE Transactions on Automatic Control, **50**(5), 655–661.

— [2008] *Distributed Consensus in Multi-vehicle Cooperative Control*, Communications and Control Engineering, Springer, ISBN 978-1-84800-014-8.

Ren, W., Beard, R. W., and Atkins, E. M. [2007] *Information consensus in multivehicle cooperative control: Collective group behavior through local interaction*, IEEE Control Systems Magazine, **27**(2), 71–82.

Sarlette, A. [2009] *Geometry and Symmetries in Coordination Control*, Ph.D. thesis, University of Liège, Belgium.

Sarlette, A. and Sepulchre, R. [2009] *Consensus optimization on manifolds*, SIAM Journal on Control and Optimization, **48**(1), 56–76.

Savkin, A. [2004] *Coordinated collective motion of groups of autonomous mobile robots: Analysis of Vicsek's model*, IEEE Transactions on Automatic Control, **49**(6), 981–982.

Scardovi, L., Sarlette, A., and Sepulchre, R. [2007] *Synchronization and balancing on the N-torus*, Systems & Control Letters, **56**(5), 335–341.

Seneta, E. [1981] *Non-negative Matrices and Markov Chains*, second edition, Springer, ISBN 0387297650.

Sipser, M. [2005] *Introduction to the Theory of Computation*, second edition, Course Technology, ISBN 0534950973.

Smith, S. L., Broucke, M. E., and Francis, B. A. [2005] *A hierarchical cyclic pursuit scheme for vehicle networks*, Automatica, **41**(6), 1045–1053.

Sontag, E. D. [1998] *Mathematical Control Theory: Deterministic Finite Dimensional Systems*, volume 6 of *TAM*, second edition, Springer, ISBN 0387984895.

Spanos, D. P., Olfati-Saber, R., and Murray, R. M. [2005] *Approximate distributed Kalman filtering in sensor networks with quantifiable performance*, in *Symposium on Information Processing of Sensor Networks*, pages 133–139, Los Angeles, CA.

Strogatz, S. H. [2000] *From Kuramoto to Crawford: Exploring the onset of synchronization in populations of coupled oscillators*, Physica D, **143**(1), 1–20.

— [2003] *SYNC: The emerging science of spontaneous order*, Hyperion, ISBN 0786868449.

Sundaram, S. and Hadjicostis, C. N. [2008] *Distributed function calculation and consensus using linear iterative strategies*, IEEE Journal on Selected Areas in Communications, **26**(4), 650–660.

Tahbaz-Salehi, A. and Jadbabaie, A. [2006] *A one-parameter family of distributed consensus algorithms with boundary: From shortest paths to mean hitting times*, in *IEEE Conf. on Decision and Control*, pages 4664–4669, San Diego, CA.

— [2007] *Small world phenomenon, rapidly mixing markov chains, and average consensus algorithms*, in *IEEE Conf. on Decision and Control*, pages 276–281, New Orleans, LA.

— [2008] *Consensus over random networks*, IEEE Transactions on Automatic Control, **53**(3), 791–795.

Triplett, B. I., Klein, D. J., and Morgansen, K. A. [2006] *Discrete time Kuramoto models with delay*, in *Network Embedded Sensing and Control. (Proceedings of NESC'05 Worskhop)*, P. J. Antsaklis and P. Tabuada, editors, volume 331 of *Lecture Notes in Control and Information Sciences*, pages 9–24, Springer, ISBN 3540327940.

Tsitsiklis, J. N. [1984] *Problems in Decentralized Decision Making and Computation*, Ph.D. thesis, Massachusetts Institute of Technology, available at `http://web.mit.edu/jnt/www/Papers/PhD-84-jnt.pdf`.

Tsitsiklis, J. N., Bertsekas, D. P., and Athans, M. [1986] *Distributed asynchronous deterministic and stochastic gradient optimization algorithms*, IEEE Transactions on Automatic Control, **31**(9), 803–812.

Vazirani, V. V. [2001] *Approximation Algorithms*, Springer, ISBN 3540653678.

Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., and Shochet, O. [1995] *Novel type of phase transition in a system of self-driven particles*, Physical Review Letters, **75**(6-7), 1226–1229.

Wang, W. and Slotine, J.-J. E. [2006] *A theoretical study of different leader roles in networks*, IEEE Transactions on Automatic Control, **51**(7), 1156–1161.

Watton, A. and Kydon, D. W. [1969] *Analytical aspects of the N-bug problem*, American Journal of Physics, **37**(2), 220–221.

Wiener, N. [1958] *Nonlinear Problems in Random Theory*, MIT Press.

Winfree, A. T. [1980] *The Geometry of Biological Time*, Springer, ISBN 0387525289.

Wolfowitz, J. [1963] *Product of indecomposable, aperiodic, stochastic matrices*, Proceedings of American Mathematical Society, **14**(5), 733–737.

Wu, C. W. [2006] *Synchronization and convergence of linear dynamics in random directed networks*, IEEE Transactions on Automatic Control, **51**(7), 1207–1210.

Xiao, L., Boyd, S., and Lall, S. [2005] *A scheme for robust distributed sensor fusion based on average consensus*, in *Symposium on Information Processing of Sensor Networks*, pages 63–70, Los Angeles, CA.

Zhu, M. and Martínez, S. [2008a] *Dynamic average consensus on synchronous communication networks*, in *American Control Conference*, pages 4382–4387, Seattle, WA.

— [2008b] *On the convergence time of distributed quantized averaging algorithms*, in *IEEE Conf. on Decision and Control*, pages 3971–3976, Cancún, México.

Algorithm Index

Subject Index

Symbol Index

$\mathbb{R}_{>0}$	:	set of positive real numbers, 6
$\mathrm{radius}(v, G)$	:	radius of v in G, 30
ρ	:	radius of curvature, 10
$\mathrm{rank}(A)$	:	rank of a matrix A, 10
$\mathrm{SC}(\mathcal{DA})$	:	space complexity of a distributed algorithm $\mathcal{DA}$, 45
$\rho(A)$	:	spectral radius of a matrix A, 13
$\rho_{\mathrm{ess}}(A)$	:	essential spectral radius of a matrix A, 14
$\mathrm{spec}(A)$	:	spectrum of a matrix A, 12
$\mathbb{S}^d$	:	sphere of dimension d, 6
stf	:	state-transition function, 42
$T_{\varepsilon}(A)$	:	ε-convergence time of $A \in \mathbb{R}^{n \times n}$, 63
$T\mathbb{R}^d$	:	tangent space of $\mathbb{R}^d$, 6
$T\mathbb{S}^d$	:	tangent space of $\mathbb{S}^d$, 6
$\mathrm{TC}(\mathcal{DA})$	:	time complexity of a distributed algorithm $\mathcal{DA}$, 45
TSP	:	traveling salesperson tour, 36
T_{BFS}	:	breadth-first spanning (BFS) tree, 31
T_{DFS}	:	depth-first spanning (DFS) tree, 33
$T_{\text{shortest-paths}}$	:	shortest-paths tree, 34
$\mathrm{Trid}_n(a, b, c)$	:	tridiagonal Toeplitz matrix, 66
$V(G)$	:	vertices of G, 24
$\mathbb{Z}_{\geq 0}$	:	set of non-negative integer numbers, 6