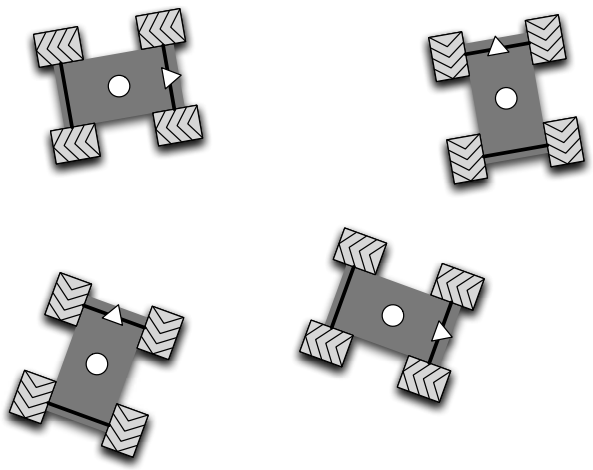
Francesco Bullo
Jorge Cortés
Sonia Martínez

Distributed Control of Robotic Networks

A Mathematical Approach to Motion Coordination Algorithms

June 4, 2008



To Lily, the third author, the third author's parents

Preface

Objectives

The last years have seen a thriving research activity on cooperative control and motion coordination. This interest is motivated by the growing possibilities enabled by robotic networks in the monitoring of natural phenomena and the enhancement of human capabilities in hazardous and unknown environments.

Our first objective with this book is to present a coherent introduction to basic distributed algorithms for robotic networks. This emerging discipline sits at the intersection of different areas such as distributed algorithms, parallel processing, control, and estimation. Our second objective is to provide a self-contained, broad exposition of the notions and tools from these areas that are relevant in cooperative control problems. These concepts include graph-theoretic notions (connectivity, adjacency and Laplacian matrices), distributed algorithms from computer science (leader election, basic tree computations) and from parallel processing (averaging algorithms, convergence rates), and geometric models and optimization (Voronoi partitions, proximity graphs). Our third objective is to put forth a model for robotic networks that helps to rigorously formalize coordination algorithms running on them. We illustrate how computational geometry plays an important role in modeling the interconnection topology of robotic networks. We draw on classical notions from distributed algorithms to provide complexity measures that characterize the execution of coordination algorithms. Such measures allow us to quantify the algorithm performance and implementation costs. Our fourth and last objective is to present various algorithms for coordination tasks such as connectivity maintenance, rendezvous, and deployment. We put special emphasis on analyzing the correctness of the algorithms and providing measures of their complexity.

The thematic variety of the exposition is also present in the proofs of the main results of the book. The technical treatment combines control-theoretic tools such as Lyapunov functions and invariance principles with techniques

from computer science and parallel processing such as induction and message counting.

Intended audience

The intended audience of this book are first- and second-year graduate students in control and robotics from Computer Science, Electrical Engineering, Mechanical Engineering, and Aerospace Engineering. A familiarity with basic concepts from analysis, linear algebra, dynamical systems, and control theory is assumed. The writing style of the book is mathematical: we have aimed at being precise in the introduction of the notions, the statement of the results, and the formal description of the algorithms. This is complemented by numerous examples, exercises, and a special effort carried throughout the book at motivating the introduction of concepts and giving intuitive explanations behind the results.

Researchers in the fields of control theory and robotics who are not aware of the literature on distributed algorithms will also benefit from the book. The book uses notions with a clear computer-science flavor such as synchronous networks, complexity measures, basic tree computations, and linear distributed iterations, and integrates them into the study of robotic networks. Likewise, researchers in the fields of distributed algorithms and automata theory who are not aware of robotic networks and distributed control will also find the book useful. The numerous connections that can be drawn between the classical study of distributed algorithms and the present book provide a friendly roadmap to step into the field of controlled coordination of robotic networks.

Book outline

Chapter 1 presents a broad introduction to distributed algorithms on synchronous networks. We start by presenting basic matrix notions and a primer on graph theory that makes special emphasis on linear algebraic aspects such as adjacency and Laplacian matrices. After this, we introduce the notion of synchronous networks, and present time, communication, and space complexity notions. We examine these notions in basic algorithms such as broadcast, tree computation, and leader election. The chapter ends with a thorough treatment of linear iterations and averaging algorithms.

Chapter 2 presents basic geometric notions that are relevant in motion coordination. Robotic networks have a spatial component which is not always present in synchronous networks as studied in computer science. Geometric objects such as polytopes, Voronoi partitions, and geometric centers play an important role in modeling the interaction of robotic networks with the physical environment. Proximity graphs allow us to rigorously formalize the

interconnection topology of a network of robotic agents, and characterize the spatially-distributed character of coordination algorithms. This notion is a natural translation of the notion of distributed algorithms treated in the previous chapter. The chapter concludes with a detailed discussion on concepts from geometric optimization and multicenter functions.

Chapter 3 introduces a model for a group of robots that synchronously communicate/sense locally, process information, and move. We describe the physical components of the robotic network and introduce a formal notion of motion coordination algorithm as a control and communication law. Generalizing the notions introduced in Chapter 1, we introduce the notion of task and of time, communication, and space complexity. We illustrate these concepts by means of a simple and insightful example of a group of robots moving on a circle.

Chapter 4 analyzes in detail two coordination tasks: connectivity maintenance and rendezvous. The objective of "connectivity maintenance" is to establish local rules that allow agents to move without losing the connectivity of the overall networks. The objective of "rendezvous" is to establish local rules that allow agents to agree on a common spatial location. We present coordination algorithms that achieve these tasks, making use of the geometric concepts introduced in the previous chapters. Furthermore, we provide results on the correctness and complexity of these algorithms.

Chapter 5 considers deployment problems. The "deployment problem" objective is to establish local rules that allow agents to achieve optimal network configurations in an environment of interest. Here, optimality is defined using the multicenter functions from geometric optimization introduced in Chapter 2. We present coordination algorithms that achieve these tasks, characterizing their correctness and complexity.

The reader will note that, as the discussion progresses, the selection of topics emphasizes problems in which we have been directly involved. There are exciting topics that have been considered in the literature and are not presented here in depth, albeit we briefly discuss a number of them throughout the exposition. In this, our first effort, we decided to tackle the problems we knew better, postponing the rest for the future. We hope the reader will appreciate the result and share, while reading it, some of the fun we had writing it.

How to use this book as a text

Our experience and opinion is that this text can be used for a quarter- or semester-long course on "Distributed Control" or on "Robotic Networks." Such a course could be taught in an Engineering or in a Computer Science department. We taught such a course at our respective institutions over a 10 weeks, 3 hours a week, period, skipping some material and some proofs

(e.g., skipping combinatorial optimization in Chapter 1, some of the multi-center functions and the nonconvex geometry treatment in Chapter 2, and the relative-sensing model in Chapter 3). With proofs and more complete treatment, we estimate the material might require 45 hours of lecture time.

Finally, the complete latest version of the manuscript, as well as supplementary material such as slides and software, is freely avaiable on the internet at:

http://coordinationbook.info

Acknowledgments

We would like to thank Emilio Frazzoli and Anurag Ganguli for their contributions to joint work that is contained in this text. Specifically, Chapter 3 contains joint work with Emilio Frazzoli on robotic networks, and Chapter 4 contains joint work with Anurag Ganguli on nonconvex rendezvous developed as part of his Ph.D. thesis. We would like to thank all our collaborators who helped shape our perspective on cooperative control over the last few years, including Giuseppe Notarstefano, Sara Susca and Ketan Savla. We would like to thank the students and colleagues who followed courses based on early versions of this text and gave constructive feedback, including Ruggero Carli, Rishi Graham, Karl J. Obermeyer, Michael Schuresko, and Stephen L. Smith. The first author would like to thank Andrew D. Lewis for his positive influence on this text. The first author is grateful for support from the National Science Foundation and the Department of Defense. The second and third authors are grateful for support from the National Science Foundation. We would like to thank the anonymous reviewers and editors who helped us improve the document. Finally, we thank our families for their tremendous patience and ongoing support.

Santa Barbara, California — *Francesco Bullo*
San Diego, California — *Jorge Cortés*
San Diego, California — *Sonia Martínez*
June 29, 2006 — June 4, 2008

Contents

1

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. In Section 1.1, we review some basic notation and state a few useful facts from matrix theory. In Section 1.2, we review basic notions of dynamical systems and convergence theorems based on invariance principles. In Section 1.3 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 Section 1.4 we finally introduce the notion of synchronous network and of distributed algorithm. We introduce various complexity notions and study them in simple example problems such as the broadcast problem, the tree computation problem, and the leader election problem. Finally, in Section 1.5 we discuss linear distributed algorithms. We focus on linear algorithms defined by sequences of stochastic matrices and review the results on their convergence properties.

1.1 Elementary concepts and 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 set of subsets of S and the set 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 $\text{int}(S)$ and ∂S, respectively. If A is a subset of or equal to S, then we write $A \subset S$. If A is a strict subset of S, then we write $A \subsetneq S$. 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. For convenience,

we will interchangeably denote the intersection of a collection of sets S_a, with $a \in A$ by $\cap_{a\in A} S_a$ or by $\cap\{S_a \mid a \in A\}$. We let $\prod_{i\in\{1,\ldots,n\}} S_i$ denote the Cartesian product of sets $S_1, \ldots, S_n$ and 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* $\mathrm{diag}(S^n)$ of S^n is given by $\mathrm{diag}(S^n) = \{(s, \ldots, s) \in S^n \mid s \in S\}$. 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 $\mathbf{0}_d = (0, \ldots, 0)$, $\mathbf{1}_d = (1, \ldots, 1)$ and the standard basis $\boldsymbol{e}_1 = (1, 0, \ldots, 0), \ldots, \boldsymbol{e}_d = (0, \ldots, 0, 1)$. 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, i.e., a unique way of associating an element of T to an element of S. Given $f : S \to T$ and $S_1 \subset S$, we let $f(S_1)$ denote the image set $\{f(s) \mid s \in S_1\}$. Given $f : S \to T$ and $g : T \to U$, we let $f \circ g : S \to U$, $f \circ g\,(s) = f(g(s))$, denote the composition of f and g. The map $\mathrm{id}_S : S \to S$ is the identity map on S. Given $f : S \to \mathbb{R}$, the *support* of f is the set of elements s such that $f(s) \neq 0$. The indicator function $1_S : S \to \mathbb{R}$ associated with S is given by $1_S(q) = 1$ if $q \in S$, and $1_S(q) = 0$ if $q \notin S$. 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, then $f^{-1}(x)$, for any $x \in \mathbb{R}$, is a *level set* of f.

In a topological space X, a (continuous) curve C is the image of a continuous map $\gamma : [a, b] \to X$. The map γ is called a *parameterization* of C. We usually identify a parameterization with the curve it defines. Note that, without loss of generality, we can take $a = 0$ and $b = 1$. A curve connects two points p and q if $\gamma(0) = p$ and $\gamma(1) = q$. The length of a continuous and piecewise continuously differentiable curve γ is

$$\mathrm{length}(\gamma) = \int_0^1 \|\dot{\gamma}(t)\|_2 dt.$$

A curve $\gamma : [0, 1] \to X$ is *not self-intersecting* if γ is injective on $(0, 1)$. A curve is *closed* if $\gamma(0) = \gamma(1)$. A set $S \subset X$ 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 non self-intersecting closed curve can be continuously deformed to a point in the set, i.e., 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.

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

1.1.1 Distance functions

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

Some relevant examples of distance functions include

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 $\|x\|_\infty = \max_{i \in \{1,\ldots,d\}} |x_i|$. Any of these norms defines naturally a *L^p-distance* in $\mathbb{R}^d$ by $\text{dist}_p(x,y) = \|y - x\|_p$. 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 $x, y \in \mathbb{S}^d$, $\text{dist}_g(x,y)$ 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

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

where $\text{dist}_{\mathsf{c}}(x,y) = (x-y) \mod 2\pi$ is the clockwise distance and $\text{dist}_{\mathsf{cc}}(x,y) = (y-x) \mod 2\pi$ 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 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. Then, one can define the *Cartesian product distance* on $\mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$ by $(\text{dist}_p, \text{dist}_g)((x_1, y_1), (x_2, y_2)) = \text{dist}_p(x_1, x_2) + \text{dist}_g(y_1, y_2)$ for (x_1, y_1), $(x_2, y_2) \in \mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$. Unless otherwise noted, it is always understood that $\mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$ is endowed with the Cartesian product distance $(\text{dist}_2, \text{dist}_g)$.

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

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

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 $B(x, \text{dist}(x, y)) \cap B(y, \text{dist}(x, y))$. These notions are illustrated in Figure 1.1 for the Euclidean distance on the plane.

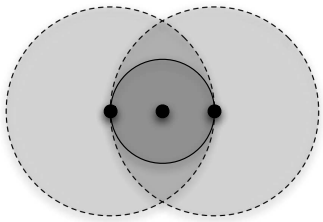


Fig. 1.1. Open balls (dashed lines), closed ball (solid line), and open lune for the Euclidean distance on the plane.

The distance between a point $x \in S$ and a set $W \in \mathbb{P}(S)$ is the infimum of all distances between x and each of the points in W. Formally, we set

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

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 $\text{diam}(\{p_1, \ldots, p_n\})$ for $P = (p_1, \ldots, p_n) \in (\mathbb{R}^d)^n$.

1.1.2 Matrix theory

For ease of reference we present here some basic results from 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 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. 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 kernel(A) and rank(A) denote the kernel and rank of A, respectively. Given a vector v, we let diag(v) denote the square matrix whose diagonal elements are equal to the component v and whose off-diagonal elements are zero.

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]

(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) *doubly stochastic* if it is row-stochastic and column-stochastic, where we say that A is *column-stochastic* if $\mathbf{1}_n^T A = \mathbf{1}_n^T$;

(v) 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;

(vi) *irreducible* if, for any nontrivial partition $J \cup K$ of the index set $\{1, \dots, n\}$, there exists $j \in J$ and $k \in K$ such that $a_{jk} \neq 0$.

Remark 1.1 (Properties of irreducible matrices). The property of irreducibility depends only upon the patterns of zeros and nonzero elements of the matrix. Also, note the following equivalent definition of irreducibility. 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

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

The scalars $\mu_1, \ldots, \mu_k$ are *convex combination coefficients* if $\mu_i \geq 0$, for $i \in \{1, \ldots, k\}$, and $\sum_{i=1}^{k} \mu_i = 1$. 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 1.2 (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, 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

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

A circulant matrix is determined by its first row, hence by n scalars. The transpose of a circulant matrix is circulant. Given two $n \times n$ circulant matrices C_1 and C_2, one can show that $C_1 + C_2$ and $C_1 C_2$ are circulant and that $C_1 C_2 = C_2 C_1$.

Eigenvalues, singular values and induced norms

We require the reader to be familiar with the notion of eigenvalue and of simple eigenvalue (i.e., an eigenvalue with algebraic and geometric multiplicity 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 $\text{spec}(A) \subset \mathbb{C}$. 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 1.3 (Geršgorin Disks). *Let A be an $n \times n$ matrix. Then*

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

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

Lemma 1.4 (Normal matrices). *For a matrix $A \in \mathbb{R}^{n \times n}$, the following statements are equivalent:*

(i) A has a complete orthonormal set of eigenvectors,
(ii) $A^T A = A A^T$, and
(iii) A is unitarily similar to a diagonal matrix, i.e., there exists a unitary[2] matrix U such that $U^* A U$ is diagonal.

If these properties hold, then the matrix A is called normal.

Lemma 1.5 (The 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 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

$$\|A\|_1 = \max_{j \in \{1,\dots,n\}} \sum_{i=1}^{n} |a_{ij}|, \quad \|A\|_\infty = \max_{i \in \{1,\dots,n\}} \sum_{j=1}^{n} |a_{ij}|,$$

$$\|A\|_2 = \max\{\sigma \mid \sigma \text{ is a singular value of } A\}.$$

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

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 1.6 (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*

$$\rho_{\text{ess}}(A) = \max\{\|\lambda\|_{\mathbb{C}} \mid \lambda \in \operatorname{spec}(A) \setminus \{1\}\}. \tag{1.1}$$

Next, we consider matrices with useful convergence properties.

Definition 1.7 (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 1.8 (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) $\rho_{\text{ess}}(A) < 1$, *that is,* 1 *is an eigenvalue and* 1 *is the only eigenvalue on the unit circle, and*
(iii) the eigenvalue 1 *is* semisimple*, i.e., 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 I_k is the $k \times k$ identity matrix, $k \in \{1, \dots, n\}$, and $B \in \mathbb{R}^{(n-k)\times(n-k)}$ is convergent, that is, $\rho(B) < 1$. With this notation, we have $\rho_{\text{ess}}(A) = \rho(B)$ and the algebraic and geometric multiplicity of the eigenvalue 1 is k.

Perron-Frobenius theory

Positive and nonnegative matrices have useful spectral properties. The first statement in the following theorem amounts to the original Perron's Theorem for positive matrices; the second statement is the extension due to Frobenius for nonnegative matrices.

Theorem 1.9 (Perron-Frobenius Theorem). *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 larger than the magnitude of any other eigenvalue, and*
(iii) $\rho(A)$ *has an eigenvector with positive components.*

Furthermore, if the square matrix A is nonnegative and 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.*

Note that, 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 $\{1, -1\}$. It is useful, therefore, to introduce a sharper characterization of nonnegative irreducible matrices.

Definition 1.10 (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 matrix is reducible, then it cannot be primitive; or in other words, if A is primitive, then it must be irreducible. The second part of the Perron-Frobenius Theorem 1.9 can now be sharpened as follows.

Theorem 1.11 (Perron-Frobenius for primitive matrices). *If a nonnegative matrix is primitive, then its spectral radius is its only eigenvalue of maximum magnitude.*

We conclude this section by noting the following convergence properties. If A is positive or A is nonnegative and primitive, then Lemma 1.8 guarantees that $\rho(A)^{-1}A$ is semi-convergent.

1.2 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

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

More general definitions of state machines can be found in the literature, e.g., see [Sipser, 2005], but the one presented here will be enough of our purposes. 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 next paragraphs.

A *(time-dependent) discrete-time continuous-space control system* is a tuple (X, U, X_0, f), where X is 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 *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 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 smooth map. The individual objects X, U, X_0 and f are termed *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 smooth maps $f_0, f_1, \ldots, f_m : \mathbb{R}_{\geq 0} \times X \to TX$. 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.

1.2.1 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 1.12 (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 is immediate to see 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 $\text{Equil}(X, f)$.

Definition 1.13 (Trajectories and sets). Let (X, f) be a time-invariant dynamical system and let W be a subset of X.

(i) 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 1.14 (Stability and attractivity notions). 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.

Remark 1.15 (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. •

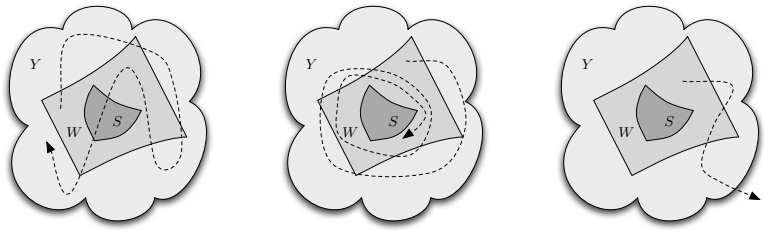


Fig. 1.2. Illustrations of stability, asymptotic stability, and instability.

1.2.2 Invariance principles

In this section we present various versions of the LaSalle-Invariance Principle, see [Khalil, 2002, LaSalle, 1986].

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 $x \in W$.

Theorem 1.16 (LaSalle Invariance Principle for discrete-time dynamical systems). *Let (X, f) be a (discrete-time continuous-space) time-independent dynamical system. Assume that*

(i) there exists a 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.

Let M denote the largest positively invariant set contained in $\{p \in \overline{W} \mid V(f(p)) = V(p)\}$. Then there exists $c \in \mathbb{R}$ such that all evolutions with initial conditions in W approach the set $M \cap V^{-1}(c)$.

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 $(x_1, \ldots, x_d)$ 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. Next, 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 are ready to state the invariance principle for continuous-time systems.

Theorem 1.17 (LaSalle Invariance Principle for continuous-time dynamical systems). *Let (X, f) be a (continuous-time continuous-space) time-independent dynamical system. Assume that*

(i) there exists a 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[3] on W.

Let M denote the largest positively invariant set contained in $\{p \in \overline{W} \mid \mathcal{L}_f V(p) = 0\}$. Then there exists $c \in \mathbb{R}$ such that all evolutions with initial conditions in W approach the set $M \cap V^{-1}(c)$.

1.2.3 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 *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

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

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 $x \mapsto T(x) = \{f(x)\}$.

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

[3] It suffices that f be locally Lipschitz and V be continuously differentiable.

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

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.

Theorem 1.18 (LaSalle Invariance Principle for set-valued discrete-time dynamical systems). *Let (X, X_0, T) be a (discrete-time continuous-space) set-valued dynamical system. Assume that*

(i) there exists a 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.

Let M denote the largest weakly positively invariant set contained in $\{p \in \overline{W} \mid \exists p' \in T(p) \text{ such that } V(p') = V(p)\}$. Then there exists $c \in \mathbb{R}$ such that all evolutions with initial conditions in W approach the set $M \cap V^{-1}(c)$.

Proof. Let γ be any evolution of (X, X_0, T) starting from W. Let $\Omega(\gamma) \subset \overline{W}$ denote the ω-limit set[4] of the sequence $\gamma = \{\gamma(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\}$. First, let us prove that $\Omega(\gamma)$ is weakly positively invariant. Let $z \in \Omega(\gamma)$. Then there exists a subsequence $\{\gamma(\ell_m) \mid m \in \mathbb{Z}_{\geq 0}\}$ of γ such that $\lim_{m\to+\infty} \gamma(\ell_m) = z$. 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, i.e., there exits 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 subse-

[4] 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 $\lim_{m\to+\infty} \gamma(\ell_m) = y$.

quence $\{\gamma(\ell_m) \mid m \in \mathbb{Z}_{\geq 0}\}$ 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, $\Omega(\gamma)$ being weakly positively invariant and V being constant on $\Omega(\gamma)$ imply that

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

Therefore, we conclude that $\lim_{\ell \to +\infty} \text{dist}(\gamma(\ell), M \cap V^{-1}(c)) = 0$, where M is the largest weakly positively invariant set contained in $\{p \in \overline{W} \mid \exists p' \in T(p) \text{ such that } V(p') = V(p)\}$. ■

1.2.4 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 $x : [\ell_0, +\infty[\to X$ 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 0.

Definition 1.19 (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 1.20 (Uniform stability and attractivity notions). For a time-dependent discrete-time dynamical system (X, X_0, f), the set S is

(i) *uniformly stable* for (X, X_0, f) 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* for (X, X_0, f) 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 $x : [\ell_0, +\infty[\to X$ 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.

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 $x : [\ell_0, +\infty[\to X$ 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 $f : \mathbb{Z}_{\geq 0} \times X \to X$, define a set-valued *overapproximation map* $T_f : X \rightrightarrows X$ by

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

Lemma 1.21 (Overapproximation Lemma). *Consider a time-dependent discrete-time 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 trajectory of the set-valued overapproximation system contains the set of trajectory of the original time-dependent system. Uniform attractivity is a consequence of attractivity for the time-invariant set-valued overapproximation.

1.3 Graph theory

We take the following basic definitions from [Cormen et al., 2001, Godsil and Royle, 2001, Biggs, 1994].

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*. For a digraph $G = (V, E)$, the *reverse digraph* $\mathrm{rev}(G)$ has vertex set V and edge set $\mathrm{rev}(E)$ composed of all edges in E with reversed direction. A digraph $G = (V, E)$ is *complete* if $E = V \times V$.

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'. A *clique* (V', E') of a digraph (V, E) is a subgraph of (V, E) which is complete, i.e., 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. 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 $\mathcal{N}_G^{\mathrm{in}}(v)$, respectively $\mathcal{N}_G^{\mathrm{out}}(v)$, denote the set of in-neighbors, respectively 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 $\mathcal{N}^{\mathrm{in}}(v)$ and $\mathcal{N}^{\mathrm{out}}(v)$, respectively. A digraph is *topologically balanced* if each vertex has the same in- and out-degrees.

Likewise, u and v are *neighbors* in a graph G 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 $\mathcal{N}(v)$.

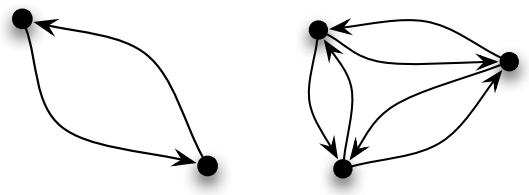
1.3.1 Connectivity notions

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 two consecutive vertices in the sequence are 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*, i.e., multiple connected subgraphs. A *cycle* is a non-trivial 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*. 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 $|E| = |V| - 1$.

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 two consecutive vertices in the sequence are a directed edge of the digraph. A *cycle* in a digraph is a non-trivial directed path that starts and ends at the same vertex. A digraph is *acyclic* if it contains no cycles. In an acyclic graph, every vertex of in-degree 0 is named *source*, and every vertex of out-degree 0 is named *sink*. Every acyclic digraph has at least one source and at least one sink.

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. Figure 1.3 shows an example of a periodic and an aperiodic digraph.



(a) (b)

Fig. 1.3. (a) Periodic and (b) aperiodic digraphs

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 E1.6.

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 *forest* is a graph that can be written as the disjoint union of trees. 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 1.4 illustrates some of these notions.

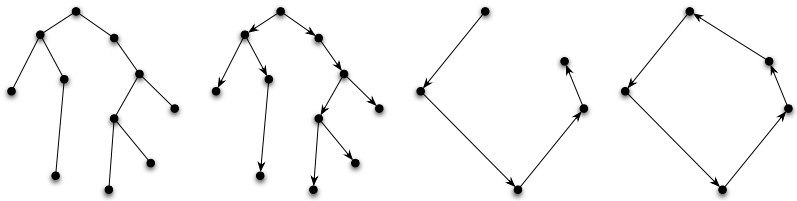


Fig. 1.4. From left to right, tree, directed tree, chain, and ring digraphs.

The proof of the following result is given in Section 1.7.1.

Lemma 1.22 (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 and a spanning tree; and

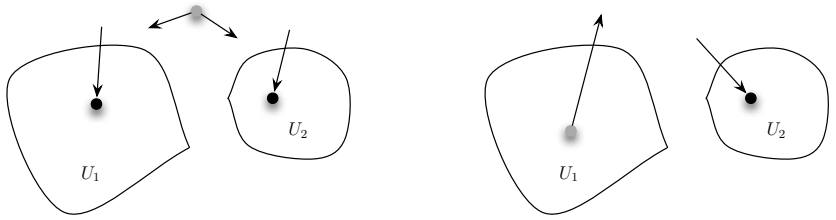
(ii) if G is topologically balanced and contains either a globally reachable vertex or a spanning tree, then G is strongly connected and is Eulerian.[5]

We conclude this section with a result from [Moreau, 2005, Lin et al., 2005]. 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 1.23 (Disjoint subsets and spanning trees). *Given a digraph G with at least two nodes, the following two properties are equivalent:*

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

(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.



(a) (b)

Fig. 1.5. Illustration of Lemma 1.23. 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.

We postpone the proof to Section 1.7.1. The result is illustrated in Figure 1.5. 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 out-neighbor of a set.

1.3.2 Weighted digraphs

A *weighted digraph* is a triplet $G = (V, E, A)$ where $V = \{v_1, \dots, v_n\}$ and E are a digraph and where $A \in \mathbb{R}^{n \times n}_{\geq 0}$ is a *weighted adjacency matrix* with the following properties: 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 edge set is uniquely determined by the weighted adjacency matrix and it can be therefore omitted. When convenient, we denote the adjacency matrix of a weighted digraph G by $A(G)$. Figure 1.6 shows an example of a weighted digraph.

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

$$a_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E, \\ 0, & \text{otherwise,} \end{cases} \tag{1.2}$$

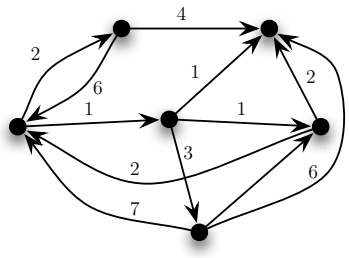


Fig. 1.6. A weighted digraph.

where $V = \{v_1, \ldots, v_n\}$. 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 $i, j \in \{1, \ldots, n\}$. Clearly, G is undirected if and only if $A(G)$ is symmetric.

A number of the 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 $V = \{v_1, \ldots, v_n\}$, 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

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

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

1.3.3 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 $\text{dist}_G(u, v)$, 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,

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

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,

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

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

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

These definitions lead to the following simple results:

(i) $\text{radius}(v, G) \leq \text{diam}(G)$ for all vertices v of G;
(ii) G contains a spanning tree rooted at v if and only if $\text{radius}(v, G) < +\infty$;
(iii) G is strongly connected if and only if $\text{diam}(G) < +\infty$.

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 $\text{wdist}_G(u, v)$, 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,

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

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 (1.2)), 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.

1.3.4 Graph algorithms

In this section we present a few algorithms defined on graphs. We only present high-level descriptions and we refer to [Cormen et al., 2001] for detailed discussion on implementation and efficiency issues.

Breadth-first spanning tree

Let v be a vertex of a digraph G with $\text{radius}(v, G) < +\infty$. 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 shortest topological length.) Let us provide the algorithm BFS that, given a digraph G of order n and a vertex v with $\text{radius}(v, G) < +\infty$, computes a BFS tree T_{BFS} rooted at v.

[Informal description] Initialize a subgraph to contain only the root v. Repeat 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)

1: $(V_1, E_1) := (\{v\}, \emptyset)$
2: **for** $k = 2$ to radius(v, G) **do**
3: find all vertices $w_1, \ldots, w_m$ not in V_{k-1} that are out-neighbors of some vertex in V_{k-1} and, for $j \in \{1, \ldots, m\}$, let e_j be an edge connecting a vertex in V_{k-1} to w_j
4: $V_k := V_{k-1} \cup \{w_1, \ldots, w_m\}$
5: $E_k := E_{k-1} \cup \{e_1, \ldots, e_m\}$
6: **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 1.7 shows an execution of the BFS algorithm.

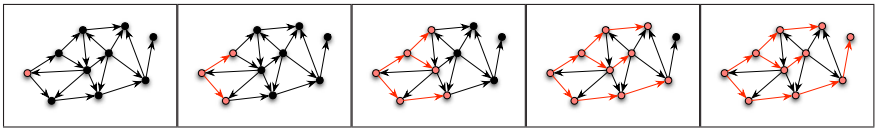


Fig. 1.7. Execution of the BFS algorithm. In the leftmost frame, vertex v is colored in red. The other frames correspond to incremental additions of vertices and edges as specified by the function BFS. The output of the algorithm is a BFS tree of the digraph. The BFS tree is represented in the last frame with vertices and edges colored in red.

Some properties of the BFS algorithm are characterized as follows.

Lemma 1.24 (BFS tree). *For a digraph G with a vertex v, any digraph T computed by the* BFS *algorithm, $T \in$ 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 $\text{dist}_G(v, w) = \text{dist}_T(v, w)$*;*
(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), $k \in \{1, \ldots, n\}$, is a sequence of directed trees with the property that $(V_k, E_k) \subset (V_{k+1}, E_{k+1})$, for $k \in \{1, \ldots, n-1\}$.

Depth-first spanning tree

Next, we define the algorithm DFS that, given a digraph G and a vertex v with $\text{radius}(v, G) < +\infty$, 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: $(V_{\text{visited}}, E_{\text{visited}}) := (\{v\}, \emptyset)$
2: DFS-VISIT(G, v)
3: **return** $(V_{\text{visited}}, E_{\text{visited}})$

`function` DFS-VISIT(G, w)

1: **for** u out-neighbor of w **do**
2: **if** u does not belong to V_{visited} **then**
3: $V_{\text{visited}} := V_{\text{visited}} \cup \{u\}$
4: $E_{\text{visited}} := E_{\text{visited}} \cup \{(w, u)\}$
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, $T \in \text{DFS}(G, v)$, is a directed spanning tree with root v. Figure 1.8 shows an execution of the algorithm.

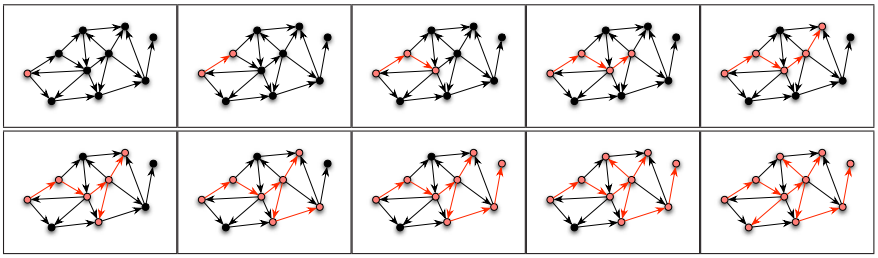


Fig. 1.8. Execution of the DFS algorithm. In the top leftmost frame, vertex v is colored in red. The other frames correspond to incremental additions of vertices and edges as specified by the function DFS. The output of the algorithm is a DFS tree of the digraph. The DFS tree is represented in the last frame with vertices and edges colored in red.

Some properties of the DFS algorithm are characterized as follows.

Lemma 1.25 (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;
(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.

Shortest-paths tree in weighted digraphs via Dijkstra's 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 $\mathrm{radius}(v, G) < +\infty$, 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 algorithm DIJKSTRA that, given a digraph G of order n and a vertex v with $\mathrm{radius}(v, G) < +\infty$, computes a shortest-paths tree $T_{\text{shortest-paths}}$ rooted at v.

> *[Informal description]* Incrementally construct a tree that contains only shortest paths. At 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$\big((V, E, A), v\big)$

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

12: `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 1.9 shows an execution of the Dijkstra's algorithm.

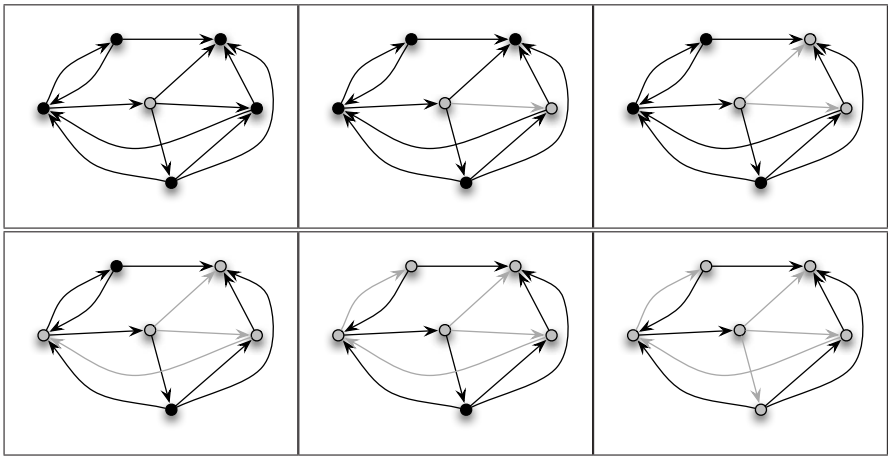


Fig. 1.9. Execution of the Dijkstra's algorithm on the weighted digraph plotted in Figure 1.6. In the top leftmost frame, vertex v is colored in gray. The other frames correspond to incremental additions of vertices and edges as specified by the function DIJKSTRA. The output of the algorithm is a shortest-paths 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's algorithm mirrors those of the BFS algorithm in Lemma 1.24.

Lemma 1.26 (Dijkstra's 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 $\text{wdist}_G(v, w) = \text{wdist}_T(v, w)$*;*
(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.

On combinatorial optimization problems

We conclude this section on graph algorithms with a brief mention of classic problems from combinatorial optimization. Standard references on combina-

torial optimization include [Korte and Vygen, 2005, Vazirani, 2001]. Given a weighted directed graph G, classical combinatorial optimization problems include:

Minimum-weight spanning tree: A *minimum-weight spanning tree (MST)* of G, denoted T_{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 (TSP)* of G, denoted T_{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 with vertices $V = \{v_1, \ldots, v_n\}$ and a set $U = \{u_1, \ldots, u_k\} \subset V$, the weighted distance from $v \in V$ to the set U is the smallest weighted distance from v to any vertex in $\{u_1, \ldots, u_k\}$. We now consider the cost functions $\mathcal{H}_{\max}, \mathcal{H}_{\Sigma} : V^k \to \mathbb{R}$ defined by

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

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

The *k-center problem* and the *k-median problem* consist of finding a set of vertices $\{u_1, \ldots, u_k\}$ 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 *uncapacited 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.

1.3.5 Algebraic graph theory

The study of matrices defined by digraphs is called algebraic graph theory; e.g., see [Godsil and Royle, 2001, Biggs, 1994]. 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 1.27 (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*

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

Then

(i) F is row-stochastic, and
(ii) F is doubly stochastic if G is weight-balanced and the 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 $\text{diag}(v)^{-1}v = \mathbf{1}_n$, for each $v \in (\mathbb{R} \setminus \{0\})^n$. Therefore

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

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

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

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 1.28 (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 1.10.

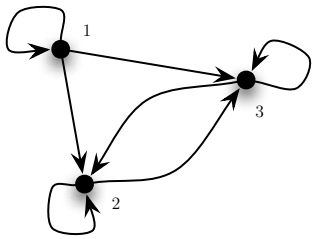


Fig. 1.10. Illustration of Proposition 1.29. Even though vertices 2 and 3 are globally reachable, the digraph is not strongly connected because vertex 1 has no in-neighbor other than itself. Therefore, the associated adjacency matrix is reducible.

Proposition 1.29 (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.

Furthermore, let $j \in \{1, \dots, n\}$. The following two statements are equivalent:

(iv) the jth node of G is globally reachable, and
(v) the jth column of $\sum_{k=0}^{n-1} A^k$ has positive entries.

Similarly, if each node of G has a self-loop, then the following two statements are equivalent:

(vi) the jth node of G is globally reachable, and
(vii) the jth column of A^{n-1} has positive entries.

Proof. (ii) $\implies$ (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)* $\implies$ *(ii)* is proved similarly.

(i) $\implies$ *(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 1.28(ii), the entry $(A^k)_{ij}$ is strictly positive. This immediately implies the statement (iii). The converse statement *(iii)* $\implies$ *(i)* is proved similarly.

(vii) $\implies$ *(vi)* Suppose there exists $j \in \{1, \dots, n\}$ such that $(A^{n-1})_{ij} > 0$, for all $i \in \{1, \dots, n\}$. Pick $i \in \{1, \dots, n\}$ and, by definition of matrix multiplication, write

$$(A^{n-1})_{ij} = \sum_{j_1, \dots, j_{n-2} \in \{1, \dots, n\}} a_{ij_1} a_{j_1 j_2} \cdots a_{j_{n-2} j}. \tag{1.3}$$

Because $(A^{n-1})_{ij}$ is strictly positive and because all terms in the summation are nonnegative, at least one of the terms in the summation is strictly positive. That is, there exist $k_1, \dots, k_{n-2} \in \{1, \dots, n\}$ such that all a_{ik_1}, $a_{k_1 k_2}$, $\dots$, $a_{k_{n-2} j}$ are strictly positive. Therefore, (i, k_1), (k_1, k_2), $\dots$, $(k_{n-2} j)$ is a directed path and so node j is globally reachable.

(vi) $\implies$ *(vii)* Suppose node j is globally reachable. Pick $i \in \{1, \dots, n\}$ and let (i, k_1), (k_1, k_2), $\dots$, (k_m, j) be a directed path from i to j of length $m \leq n-1$. Therefore, we know a_{ik_1}, $a_{k_1 k_2}$, $\dots$, $a_{k_m j}$ are strictly positive. Additionally, we know a_{jj} is strictly positive because G has self loops at each node. Finally, for arbitrary $i \in \{1, \dots, n\}$ we compute

$$\begin{aligned}(A^{n-1})_{ij} &= \sum_{j_1, \dots, j_{n-2} \in \{1, \dots, n\}} a_{ij_1} a_{j_1 j_2} \cdots a_{j_{n-2} j} \\ &\geq a_{ik_1} a_{k_1 k_2} \cdots a_{k_m j} \underbrace{a_{jj} \cdots a_{jj}}_{n-m-1 \text{ times}} > 0.\end{aligned}$$

This concludes our proof that fact *(vii)* is equivalent to fact *(vi)*. In the interest of brevity we do not include the analogous proof of the equivalence between fact *(iv)* and fact *(v)*. ■

Next, we characterize the relationship between irreducible aperiodic digraphs and primitive matrices (recall Definition 1.10).

Proposition 1.30 (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;
(ii) A is primitive, i.e., 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 analogues that hold when the original object is a nonnegative matrix, instead of a weighted digraph.

Remark 1.31 (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 analogues 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;
(iii) A is irreducible if and only if its associated weighted digraph is strongly connected. •

Finally, to conclude this section, we study a third relevant matrix associated to a digraph, called the Laplacian matrix. The *Laplacian matrix* of the weighted digraph G is

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

Laplacian matrices have numerous remarkable properties; two elegant surveys are [Mohar, 1991, Merris, 1994]. Here we present only a few basic properties directly relevant to later developments. We begin with some immediate consequences of the definitions above:

(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 1.32 (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.

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 1.3, we know that each eigenvalue of $L(G)$ belongs to at least one of the disks

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

These disks contain the origin $\mathbf{0}_n$ and complex numbers with 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 $\bar{A} = D_{\text{out}}(G)^{-1} A(G)$ and $\bar{L} = D_{\text{out}}(G)^{-1} L(G)$ and note that they satisfy $\bar{L} = I_n - \bar{A}$. Since $D_{\text{out}}(G)$ is diagonal, the matrices $A(G)$ and $\bar{A}$ have the same pattern of zeros and positive entries. This observation and the assumption that G is strongly connected imply that $\bar{A}$ is nonnegative and irreducible. By the Perron-Frobenius Theorem 1.9, the spectral radius $\rho(\bar{A})$ is a simple eigenvalue. Furthermore, one can verify that $\bar{A}$ is row-stochastic (see Lemma 1.27) and that, therefore, its spectral radius is 1 (see Exercise E1.2). In summary, we conclude that 1 is a simple eigenvalue of $\bar{A}$, that 0 is a simple eigenvalue of $\bar{L}$, that $\bar{L}$ has rank $n-1$, and that $L(G)$ has rank $n-1$.

Regarding statement (iii), we first prove that $\text{rank}(L(G)) = n-1$ implies the existence of a globally reachable vertex. By contradiction, let G contain no globally reachable vertex. Then, by Lemma 1.23, 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 in the following 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 $\text{rank}(L(G)) = n-1$.

Next, still regarding statement (iii), we prove that the existence of a globally reachable vertex implies $\text{rank}(L(G)) = n-1$. 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 $\text{rank}(L(G)) = n-1$. Assume therefore $r < n$. Renumber the vertices so that R is the set of the first r vertices. After this permutation, the adjacency matrix and Laplacian matrix can be partitioned in the following 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 $\bar{A} = D_{\text{out}}^{-1}A$ and $\bar{L} = D_{\text{out}}^{-1}L$ considered in the proof of statement (ii) above. With this block decomposition, we compute

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

for some matrix $\bar{A}_{21}(n-1)$ that depends upon $\bar{A}_{11}$, $\bar{A}_{21}$ and $\bar{A}_{22}$. Because a globally reachable node in G is globally reachable also in the digraph associated to $\bar{A}$, Proposition 1.29(vii) implies that $\bar{A}_{21}(n-1)$ is positive. This fact, combined with the fact that $\bar{A}$ and hence $\bar{A}^{n-1}$ are row-stochastic, implies that $\bar{A}_{22}^{n-1}$ has maximal row sum (that is, ∞-induced norm) strictly less than 1. Hence, the spectral radius of $\bar{A}_{22}^{n-1}$ and of $\bar{A}_{22}$ are strictly less than 1. Since $\bar{A}_{22}$ has spectral radius strictly less than 1, the matrix $\bar{L}_{22} = I_{n-r} - \bar{A}_{22}$, 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 $\dot{V}(x) \le 0$, for all $x \in \mathbb{R}^n$, is equivalent to $L(G) + L(G)^T \ge 0$. Because $\mathbf{1}_n^T L(G) = \mathbf{0}_n^T$ and $L(G)\mathbf{1}_n = \mathbf{0}_n$, it is immediate to establish that $\exp(-L(G)t)$, $t \in \mathbb{R}$, is a doubly stochastic matrix. From Theorem 1.2 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 $\sum_\alpha \lambda_\alpha(t) = 1$, $\lambda_\alpha(t) \ge 0$, 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

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

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

1.4 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].

1.4.1 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 1.33 (Network). The physical component of a synchronous network $\mathcal{S}$ is a digraph (I, E_{cmm}), where

(i) $I = \{1, \ldots, n\}$ is called the *set of unique identifiers (UIDs)*, and
(ii) E_{cmm} is a set of directed edges over the vertices $\{1, \ldots, n\}$, called the communication links. •

In general, the set of unique identifiers does not need to be n consecutive natural numbers, but we take this convention for simplicity. The set E_{cmm} models the topology of the communication service among the nodes: for $i, j \in \{1, \ldots, n\}$, 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 1.34 (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*;

(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*;
(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. 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 its state $w^{[i]}$ and to the messages it receives from its in-neighbors. Note that, at each round, the first step is transmission and the second one is computation.

Definition 1.35 (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

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

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

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

We let $\ell \mapsto w(\ell) = (w^{[1]}(\ell), \ldots, w^{[n]}(\ell))$ 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 distributed algorithms on networks.

Remarks 1.36 (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, i.e., 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 $\mathrm{msg}_{\mathrm{std}}(w, j) = w$ is referred to as the *standard message-generation function*. •

Remarks 1.37 (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," i.e., 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, e.g., 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 returns `null` always), 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. •

1.4.2 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 1.38 (Algorithm completion). We say that an algorithm *terminates* when only `null` messages are transmitted and all processors states become constants. •

Remark 1.39 (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, i.e., 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 1.40 (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, we quantify 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 unit and basic messages.

Definition 1.41 (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 1.42 (Space complexity conventions). By convention, each processor knows its identity, i.e., 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 compute the communication complexity by counting basic messages.

Definition 1.43 (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 Bachman-Laundau 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 $f(n)$;
(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

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

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

Remark 1.44 (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, i.e., 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 also possible to define *expected* or *average* complexity notions, where one is be 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 execution of an algorithm 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. •

1.4.3 Broadcast and BFS tree computation

In the following, we consider some basic algorithmic problems such as the simple one-to-all communication task, i.e., broadcasting, and the establishment of some "control structures," see Remarks 1.37, such as the construction of a BFS spanning tree and the election of a leader.

Problem 1.45 (Broadcast). Assume that a processor, called the *source*, has a message, called the *token*. Transmit the token to all others processors. •

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 1.46 (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* $\mathrm{radius}(v, \mathcal{S})$.

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

Problem 1.47 (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 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 1.24.

> *[Informal description]* The source broadcasts the token to its out-neighbors. At 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, i.e., 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. At 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 $\{\alpha, \ldots, \omega\}$.

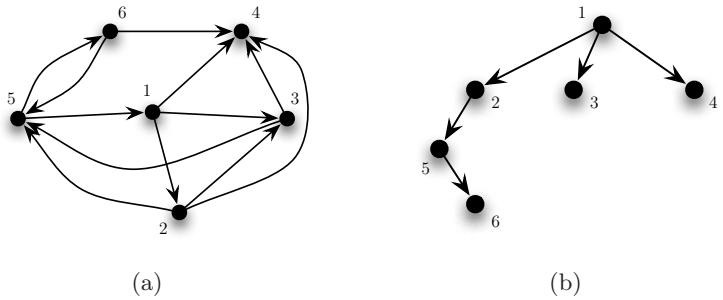
`Synchronous Network:` $\mathcal{S} = (\{1, \ldots, n\}, E_{\text{cmm}})$
`Distributed Algorithm:` FLOODING
`Alphabet:` $\mathbb{A} = \{\alpha, \ldots, \omega\} \cup$ `null`
`Processor State:` $w = ($`parent`, `data`, `snd-flag`$)$, where

`parent`	$\in \{1, \ldots, n\} \cup$ `null`,	initially: $\texttt{parent}^{[1]} = 1$, $\texttt{parent}^{[j]} =$ `null` for all $j \neq 1$
`data`	$\in \mathbb{A}$,	initially: $\texttt{data}^{[1]} = \mu$, $\texttt{data}^{[j]} =$ `null` for all $j \neq 1$
`snd-flag`	$\in \{$`false`, `true`$\}$,	initially: $\texttt{snd-flag}^{[1]} =$ `true`, $\texttt{snd-flag}^{[j]} =$ `false` for all $j \neq 1$

`function` msg(w, i)
1: **if** (`parent` $\neq i$) AND (`snd-flag` = `true`) **then**
2: **return** `data`
3: **else**
4: **return** `null`

`function` stf(w, y)
1: **case**
2: (`data` = `null`) AND (y contains only `null` messages):
% The node has not yet received the token
3: `new-parent` := `null`
4: `new-data` := `null`
5: `new-snd-flag` := `false`
6: (`data` = `null`) AND (y contains a non-`null` message):
% The node has just received the token
7: `new-parent` := smallest UID among transmitting in-neighbors
8: `new-data` := a non-`null` message
9: `new-snd-flag` := `true`
10: (`data` $\neq$ `null`):
% If the node already has the token, then do not re-broadcast it
11: `new-parent` := `parent`
12: `new-data` := `data`
13: `new-snd-flag` := `false`
14: **return** (`new-parent`, `new-data`, `new-snd-flag`)

An execution of the FLOODING algorithm is illustrated in Figure 1.11.



(a) (b)

Fig. 1.11. 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 $d \in \{1, \ldots, \text{radius}(v, \mathcal{S})\}$, every node at a distance d from the root receives a non-`null` message at round d. A summary of results is given as follows.

Lemma 1.48 (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 $\Theta(1)$.*

Remark 1.49. As presented, the flooding algorithm does not include a termination condition, i.e., 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, Peleg, 2000] for a complete discussion about this. •

1.4.4 Leader election

Next, we formulate another interesting problem for a network.

Problem 1.50 (Leader election). Assume that all processors of a network have a state variable, say `leader`, initially set to `unknown`. 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 is a 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 $\text{diam}(\mathcal{S})$, or an upper bound on it.

[Informal description] At each communication round, each agent sends to all its neighbors the maximum UID it has received up to that time. This is repeated for $\mathrm{diam}(\mathcal{S})$ 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 FLOODMAX: 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 $\mathrm{diam}(\mathcal{S})$ 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 $\mathrm{diam}(\mathcal{S})$. The algorithm is formally stated as follows.

`Synchronous Network:` $\mathcal{S} = (\{1, \ldots, n\}, E_{\mathrm{cmm}})$
`Distributed Algorithm:` FLOODMAX
`Alphabet:` $\mathbb{A} = \{1, \ldots, n\} \cup \{\texttt{null}\}$
`Processor State:` $w = (\texttt{my-id}, \texttt{max-id}, \texttt{leader}, \texttt{round})$, where

$\texttt{my-id} \in \{1, \ldots, n\}$,	initially:	$\texttt{my-id}^{[i]} = i$ for all i
$\texttt{max-id} \in \{1, \ldots, n\}$,	initially:	$\texttt{max-id}^{[i]} = i$ for all i
$\texttt{leader} \in \{\texttt{false}, \texttt{true}, \texttt{unknown}\}$,	initially:	$\texttt{leader}^{[i]} = \texttt{unknown}$ for all i
$\texttt{round} \in \{0, 1, \ldots, \mathrm{diam}(\mathcal{S})\}$,	initially:	$\texttt{round}^{[i]} = 0$ for all i

`function` $\mathrm{msg}(w, i)$

```
1: if round < diam(S) then
2:    return max-id
3: else
4:    return null
```

`function` $\mathrm{stf}(w, y)$

```
1: new-id:= max{max-id, largest identifier in y}
2: case
3:    round < diam(S):    new-lead := unknown
4:    round = diam(S) AND max-id = my-id:    new-lead := true
5:    round = diam(S) AND max-id > my-id:    new-lead := false
6: return (my-id, new-id, new-lead, round +1)
```

Figure 1.12 shows an execution of the FLOODMAX algorithm. This algorithm's properties 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].

Lemma 1.51 (Complexity upper bounds for the floodmax algorithm). *For a network $\mathcal{S}$ containing a spanning tree, the* FLOODMAX *algorithm has*

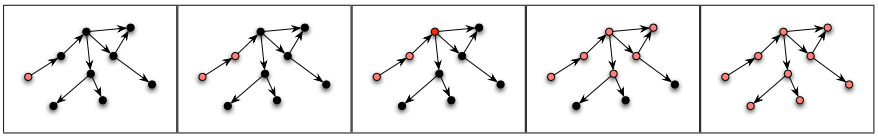


Fig. 1.12. Execution of the FLOODMAX algorithm. The diameter of the network is 4. In the leftmost frame, the agent with the maximum UID is colored in red. After 4 communication rounds, its message has been received by all agents.

communication complexity in $O(\text{diam}(\mathcal{S})|E_{\text{cmm}}|)$*, time complexity equal to* $\text{diam}(\mathcal{S})$*, and space complexity in* $\Theta(1)$*.*

A simplification of the FLOODMAX algorithm leads to the Le Lann-Chang-Roberts (LCR) algorithm for leader election in rings, see [Lynch, 1997, Chapter 3.3], that we describe next. The LCR algorithm runs on a ring digraph and does not require the agents to know the diameter of the network.

> *[Informal description]* At each communication round, if the agent receives from its in-neighbor a UID that is larger than the UIDs received earlier, then the agent records the received UID and forwards it to the out-neighbor during the following communication round. (Agents do not record the number of communication rounds.) When the agent with the maximum UID receives its own UID from a neighbor, it declares itself the leader.

The algorithm is formally stated as follows.

`Synchronous Network:` ring digraph
`Distributed Algorithm:` LCR
`Alphabet`: $\mathbb{A} = \{1, \ldots, n\} \cup \{\texttt{null}\}$
`Processor State`: $w = (\texttt{my-id}, \texttt{max-id}, \texttt{leader}, \texttt{snd-flag})$, where

`my-id`	$\in \{1, \ldots, n\}$,	initially: $\texttt{my-id}^{[i]} = i$ for all i
`max-id`	$\in \{1, \ldots, n\}$,	initially: $\texttt{max-id}^{[i]} = i$ for all i
`leader`	$\in \{\texttt{true}, \texttt{false}, \texttt{unknown}\}$,	initially: $\texttt{leader}^{[i]} = \texttt{unknown}$ for all i
`snd-flag`	$\in \{\texttt{true}, \texttt{false}\}$,	initially: $\texttt{snd-flag}^{[i]} = \texttt{true}$ for all i

`function` msg(w, i)

```
1: if snd-flag = true then
2:     return max-id
3: else
4:     return null
```

`function` stf(w, y)

```
1: case
2:   (y contains only null msgs) OR (largest identifier in y < my-id):
3:       new-id := max-id
4:       new-lead := leader
```

```
5:       new-snd-flag := false
6:   (largest identifier in y = my-id):
7:       new-id := max-id
8:       new-lead := true
9:       new-snd-flag := false
10:  (largest identifier in y > my-id):
11:      new-id := largest identifier in y
12:      new-lead := false
13:      new-snd-flag := true
14: return (my-id, new-id, new-lead, new-snd-flag)
```

Figure 1.13 shows an execution of the LCR algorithm. The properties of

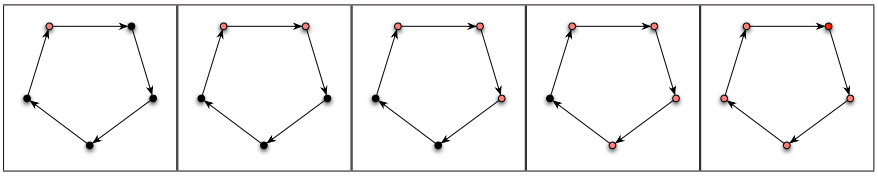


Fig. 1.13. Execution of the LCR algorithm. In the leftmost frame, the agent with the maximum UID is colored in red. After 5 communication rounds, this agent receives its own UID from its in-neighbor and declares itself the leader.

the LCR algorithm can be characterized as follows.

Lemma 1.52 (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 $\Theta(1)$.*

1.4.5 Shortest-paths tree computation

Finally, we consider the shortest-paths tree problem in a weighted digraph: in Section 1.3.4 we presented Dijkstra's 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, i.e., we assume that a strictly positive weight is associated to each communication edge. We assume that the source is vertex 1 and we aim to compute a tree containing shortest paths from 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 (1) an estimate `dist` of its weighted distance from the source, and (2) 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. At each communication round, each agent performs the following tasks: (1) it transmits its `dist` to its out-neighbors, (2) it computes the smallest quantity among "the `dist` received from an in-neighbor summed with the edge weight corresponding to that same 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

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

`function` msg(w, i)

```
1: if round < n then
2:     return dist
3: else
4:     return null
```

`function` stf(w, y)

```
1: i := processor UID
2: k := arginf{y_j + a_ji | for all y_j ≠ null}
3: if (dist < k) then
4:     return (parent, dist)
5: else
6:     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:

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

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

This algorithm's key property enabling its analysis is that, after k communication rounds, the estimated distance at node i equals the shortest path of topological length 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 properties of the DISTRIBUTED BELLMAN-FORD algorithm as follows.

Lemma 1.53 (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 $\Theta(1)$.*

1.5 Linear distributed algorithms

Computing 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.

1.5.1 Linear iterations on synchronous networks

Given a synchronous network $\mathcal{S} = (\{1, \ldots, n\}, E_{\text{cmm}})$, 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 $\text{msg}(w, i) = \text{msg}_{\text{std}}(w, i)$

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

We assume that each processor $i \in \{1, \ldots, n\}$ 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, i.e., we neglect quantization issues in the message-generation function.

In the language of Section 1.2, 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 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

$$w(0) = w_0, \quad w(\ell + 1) = F \cdot w(\ell), \quad \ell \in \mathbb{Z}_{\geq 0}. \tag{1.4}$$

Conversely, any linear algorithm of the form (1.4) can easily be casted 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 1.31. 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.

1.5.2 Averaging algorithms

In what follows we consider linear combination algorithms over time-dependent weighted directed graphs; we restrict our analysis to nonnegative weights.

Definition 1.54 (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

$$w(\ell + 1) = F(\ell) \cdot w(\ell), \quad \ell \in \mathbb{Z}_{\geq 0}. \tag{1.5}$$

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 E1.8.

Definition 1.55 (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

$$w_i(\ell + 1) = \frac{1}{1 + d_{\text{out}}(i)} \big(w_i(\ell) + \sum_{j=1}^{n} a_{ij} w_j(\ell) \big); \tag{1.6}$$

(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

$$w_i(\ell + 1) = \Big(1 - \varepsilon \sum_{j=1, j \neq i}^{n} a_{ij} \Big) w_i(\ell) + \varepsilon \sum_{j=1, j \neq i}^{n} a_{ij} w_j(\ell). \tag{1.7}$$

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 the *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

$$w_i(\ell + 1) = \text{avrg}\Big(\{w_i(\ell)\} \cup \{w_j(\ell) \mid j \in \mathcal{N}_G(i)\}\Big), \tag{1.8}$$

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

Remark 1.56 (Sensing versus communication interpretation of directed edges). In the definition of averaging algorithms arising from digraphs, the edges of the digraph play the role of "sensing edges," not that of "communication edges." In other words, a nonzero entry a_{ij}, corresponding to the edge (i, j) in the digraph, 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 $\text{diag}(\mathbb{R}^n)$ is the vector subspace generated by $\mathbf{1}_n$. Therefore, any point in $\text{diag}(\mathbb{R}^n)$ is an equilibrium for any averaging algorithm. We refer to the points of the $\text{diag}(\mathbb{R}^n)$ as *agreeement configurations*, since all the components of an element in $\text{diag}(\mathbb{R}^n)$ are equal to the same value. We will informally say that an algorithm *achieves agreement* if it steers the network state towards the set of agreement configurations.

Lemma 1.57 (Stability of agreement configurations). *Any averaging algorithm in $\mathbb{R}^n$ is uniformly stable and uniformly bounded with respect to* $\text{diag}(\mathbb{R}^n)$.

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 (in the sharpest version given by Moreau [2005]) and postpone its proof to Section 1.7.2.

Theorem 1.58 (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 1.31. The following statements are equivalent:*

(i) the set $\mathrm{diag}(\mathbb{R}^n)$ *is uniformly globally attractive for the associated averaging algorithm;*

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

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

contains a globally reachable vertex.

We collect a few observations about this result.

Remarks 1.59. (i) The statement in Theorem 1.58(i) means that each solution to the time-dependent linear dynamical system (1.5) converges uniformly and asymptotically to the vector subspace generated by $\mathbf{1}_n$.

(ii) The necessary and sufficient condition in Theorem 1.58(ii) amounts to the existence of a uniformly-bounded time duration δ with the property that a weak connectivity assumptions holds over each collection of δ consecutive digraphs. We refer to [Blondel et al., 2005] for a counterexample showing is the duration in Theorem 1.58 is not uniformly bounded, then there exist algorithms that do not converge.

(iii) According to Definition 1.20, uniform convergence is a property of all solutions to system (1.5) starting at *any arbitrary time*, and not only at time equal to 0. If we restrict our attention to solutions that only start at time 0, then Theorem 1.58 should be modified as follows: the statement in Theorem 1.58(i) implies, but is not implied by, the statement in Theorem 1.58(ii).

(iv) The theorem applies only to sequences of non-degenerate matrices. Indeed, there exist sequences of degenerate stochastic matrices whose associated averaging algorithms converges. 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 1.11). Examples of degenerate primitive stochastic matrices (with converging associated averaging algorithms) are given in Exercise E1.15. We discuss time-invariant averaging algorithms in Proposition 1.63 below. •

Theorem 1.58 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 the ones expressed in statement (ii) of Theorem 1.58.

Theorem 1.60 (Convergence for time-dependent symmetric stochastic 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 1.31. The following statements are equivalent:*

(i) the set $\mathrm{diag}(\mathbb{R}^n)$ *is globally attractive for the associated averaging algorithm;*
(ii) for all $\ell \in \mathbb{Z}_{\geq 0}$, the graph

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

is connected.

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

Corollary 1.61 (Convergence of adjancency- 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 $\mathrm{diag}(\mathbb{R}^n)$ *is uniformly globally attractive for the the time-dependent adjancency-based averaging algorithm (1.6);*
(iii) the set $\mathrm{diag}(\mathbb{R}^n)$ *is uniformly globally attractive for the the time-dependent Laplacian-based averaging algorithm (1.7) (defined with $\varepsilon < 1/n$).*

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

Proposition 1.62 (Convergence to a point in the invariant set). *Under the assumptions in Theorem 1.58 and assuming that* $\mathrm{diag}(\mathbb{R}^n)$ *is uniformly globally attractive for the averaging algorithm, each individual evolution converges to a specific point of* $\mathrm{diag}(\mathbb{R}^n)$*, rather than converging to the whole set.*

In general, the specific value upon which all w_i, $i \in \{1, \ldots, n\}$, agree in the limit is unknown. Clearly, this agreement value depends on the initial condition and the specific sequence of matrices defining the time-dependent linear algorithm. In some cases, however, by restricting the class of allowable matrices, we can elucidate the common limit value. We consider two important 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 1.7).

Proposition 1.63 (Time-independent averaging algorithm). *Consider the linear dynamical system on* $\mathbb{R}^n$

$$w(\ell+1) = Fw(\ell), \quad \ell \in \mathbb{Z}_{\geq 0}. \tag{1.9}$$

Assume $F \in \mathbb{R}^{n\times n}$ *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* (1.9) *converges to* $(v^T w(0)/v^T \mathbf{1}_n)\mathbf{1}_n$.

Proof. From Theorem 1.58 we know that the dynamical system (1.9) converges if property (ii) holds. The same conclusion follows if F satisfies property (i) because of Perron-Frobenius Theorem 1.9 and Lemma 1.8. 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 $\alpha(v^T\mathbf{1}_n) = \lim_{\ell\to+\infty} v^T w(\ell) = v^T w(0)$. ■

Remark 1.64. (i) The following necessary and sufficient condition generalizes and is weaker than the two sufficient conditions given in Proposition 1.63: every trajectory of system (1.9) 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 Exercise E1.6 for the notion of condensation digraph and to Chapter 8 of [Meyer, 2001] for the proof of this statement and for the related notion of ergodic classes of a Markov chain.

(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 specialize the main convergence result to the case of doubly stochastic averaging algorithms.

Corollary 1.65 (Average consensus). *Let $\{F(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\}$ be a sequence of stochastic matrices as in Theorem 1.58. If all matrices $F(\ell)$, $\ell \in \mathbb{Z}_{\geq 0}$, are doubly stochastic, then every trajectory w of the averaging algorithms satisfies*

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

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 1.58, 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 double 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.

1.5.3 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 1.8).

Definition 1.66 (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 $r_{\exp}(A) \in [0, 1[$, is

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

The exponential factor of convergence 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 $r_{\exp}(A) = \rho$.

Lemma 1.67. *If A is a convergent matrix, then $r_{\exp}(A) = \rho(A)$.*

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

Remark 1.68 (Complexity notions). Analogously to the treatment in Section 1.4.2, 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 $f(n, \varepsilon)$;
(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 1.69 (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 definition of exponential convergence factor and of limsup, 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 $\eta = (1 - r_{\exp}(A_n))/2$, straightforward 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 1.70 (Exponential convergence factor for stochastic matrices). *Let F be a non-degenerate stochastic matrix whose associated digraph has a globally reachable node. Then*

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

(From equation (1.1) recall $\rho_{\text{ess}}(F) = \max\{\|\lambda\|_{\mathbb{C}} \mid \lambda \in \operatorname{spec}(F) \setminus \{1\}\}$.)

Proof. If $v \in \mathbb{R}^n$ is a left eigenvector of F, then, as in Proposition 1.63,

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

Relaying 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,

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

Therefore

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

The following result establishes bounds on convergence factors and convergence times for stochastic matrices arising from the equal-neighbor averaging rule in equation (1.8).

Theorem 1.71 ([Landau and Odlyzko, 1981]). *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*

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

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

1.5.4 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. Our presentation follows Martínez et al. [2007a]. 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 $\mathrm{Trid}_n(a, b, c)$ and $\mathrm{Circ}_n(a, b, c)$ by

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

and

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

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

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

Note that, for $a = 0$ and $c \neq 0$ (alternatively, $a \neq 0$ and $c = 0$), the synchronous networks defined by $\mathrm{Trid}(a, b, c)$ and $\mathrm{Circ}(a, b, c)$ are, respectively, the chain and the ring digraphs introduced in Section 1.3. 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 1.72 (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 $\mathrm{Trid}_n(a, b, c)$ are, for $i \in \{1, \dots, n\}$,

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

(ii) the eigenvalues and eigenvectors of $\mathrm{Circ}_n(a,b,c)$ *are, for* $\omega = \exp(\frac{2\pi\sqrt{-1}}{n})$ *and for* $i \in \{1,\dots,n\}$,

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

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. ■

Remarks 1.73. (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 1.74 (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* $O\big(n \log n + \log \varepsilon^{-1}\big)$;

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

The proof of this result is reported in Section 1.7.3. 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

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

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

then the following similarity transforms are satisfied

$$\mathrm{ATrid}_n^{\pm}(a,b) = P_{\pm} \begin{bmatrix} b \pm 2a & 0 \\ 0 & \mathrm{Trid}_{n-1}(a,b,a) \end{bmatrix} P_{\pm}^{-1}, \tag{1.10}$$

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 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 $x_{\mathrm{final}} : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$ with convergence time $T_\varepsilon \in \mathbb{Z}_{\geq 0}$ if

(i) $\|x(\ell) - x_{\mathrm{final}}(\ell)\|_2 \to 0$ as $\ell \to +\infty$, and
(ii) T_ε is the smallest time such that $\|x(\ell) - x_{\mathrm{final}}(\ell)\|_2 \leq \varepsilon \|x(0) - x_{\mathrm{final}}(0)\|_2$, for all $\ell \geq T_\varepsilon$.

Theorem 1.75 (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) $\lim_{\ell \to +\infty} \big(x(\ell) - x_{\mathrm{ave}}(\ell)\mathbf{1}_n\big) = \mathbf{0}_n$, *where* $x_{\mathrm{ave}}(\ell) = (\frac{1}{n}\mathbf{1}_n^T x_0)(b+2a)^\ell$, *with ε-convergence time in* $\Theta\big(n^2 \log \varepsilon^{-1}\big)$;

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

The proof of this result is reported in Section 1.7.3.

Remark 1.76 (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 1.74(iii) and Theorem 1.75(i) are slight modifications of tridiagonal Toeplitz matrices and are doubly stochastic. Indeed, note that the evolutions converge to the average consensus value, as predicted by Corollary 1.65. Note that convergence times obtained for Circ_n and ATrid_n^+ are consistent with the upper bound predicted by Theorem 1.71. •

We conclude this section with some useful bounds.

Lemma 1.77. *Assume $x \in \mathbb{R}^n$, $y \in \mathbb{R}^{n-1}$, 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 E1.21.

1.6 Notes

State machines and dynamical systems

The literature on dynamical and control systems is vast. The main tool we use in later sections is the LaSalle Invariance Principle, obtained in [LaSalle, 1960]; see also the works by [Barbašin and Krasovskiĭ, 1952] and [Krasovskiĭ, 1963] for related earlier versions. Example relevant 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

In Section 1.3.5, all statements about powers of the adjacency matrix are standard results in algebraic graph theory, e.g., see [Godsil and Royle, 2001, Biggs, 1994].

Proposition 1.30, on the fact that a weighted digraph is aperiodic and irreducible if and only if its adjacency matrix is primitive, is a standard result in the theory of Markov chains, e.g., see [Seneta, 1981, Meyer, 2001].

Theorem 1.32 characterizing the properties of the Laplacian matrix contains recent results. Statement (ii) is proved 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], 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 $\text{rank}(L(\text{rev}(G))) = n - 1$. 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 only presented a few token ideas and we refer to the textbooks by Lynch [1997] and Peleg [2000] for wonderful 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 process 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 20th 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, 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 to the average consensus problem was given in [Olfati-Saber and Murray, 2004].

Regarding Theorem 1.58 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 re-discovered 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 1.60; additional related results and a historical discussion appeared in [Blondel et al., 2005, Hendrickx, 2008].

Among the numerous recent directions of research on consensus and averaging we mention: continuous-time consensus algorithms [Olfati-Saber and Murray, 2004, Moreau, 2004, Lin et al., 2004, Ren and Beard, 2005, Lin et al., 2005, 2007c], consensus over random networks [Hatano and Mesbahi, 2005, Wu, 2006, Porfiri and Stilwell, 2007, Tahbaz-Salehi and Jadbabaie, 2008, Picci and Taylor, 2007, Fagnani and Zampieri, 2008, Patterson et al., 2007], consensus in finite-time [Cortés, 2006, Sundaram and Hadjicostis, 2008], consensus algorithms for general functions [Bauso et al., 2006, Cortés, 2008, 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., 2008a, Zhu and Martínez, 2008b], consensus on manifolds [Sarlette and Sepulchre, 2007, Scardovi et al., 2007, 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 the convergence rates and time complexity [Landau and Odlyzko, 1981, Olshevsky and Tsitsiklis, 2007, Carli et al., 2008b, Cao et al., 2008]. Numerous interesting results are reported in the recent PhD theses [Lin, 2005, Lorenz, 2007, Cao, 2007, Hendrickx, 2008, Carli, 2008]. Finally, we point out two recent surveys [Olfati-Saber et al., 2007, Ren et al., 2007] and the text [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], 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, 2008, Triplett et al., 2006, Papachristodoulou and Jadbabaie, 2006, Wang and Slotine, 2006].

1.7 Proofs

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

1.7.1 Proofs of Lemmas 1.22 and 1.23

Proof (Lemma 1.22). 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 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 [2008] for the proof that G is Eulerian. ■

Proof (Lemma 1.23). (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 exists 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,
(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 requirements (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 $\bar{i} \in U_1(k-1)$ 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:

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

where $\bar{i}_k$ is an arbitrary node which does not belong 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. ■

1.7.2 Proofs of Theorem 1.58 and Proposition 1.62

In this section we prove Theorem 1.58. 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 1.18. We begin with some preliminary results.

Lemma 1.78 (Union of graphs 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*

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

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

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

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

or, equivalently by Proposition 1.29,

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

Lemma 1.79. *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. We invite the reader in Exercise E1.16 to prove that $\mathcal{F}(\alpha)$ is closed. Let us prove now 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 show 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 1.58.

Proof (Theorem 1.58). First, we prove that (i) implies (ii). Suppose that for all $\delta \in \mathbb{N}$, 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 1.23 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 condition:

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

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

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

Because δ can be chose arbitrarily large, one can easily construct a contradiction with the fact that $\mathrm{diag}(\mathbb{R}^n)$ 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 (1.5) satisfies

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

Next, we intend to use the LaSalle Invariance Principle for set-valued discrete systems, presented as Theorem 1.18, to prove that $\lim_{\ell\to+\infty} \mathrm{dist}(w(k\ell), \mathrm{diag}(\mathbb{R}^n)) = 0$. This will then imply, by Lemma 1.21, the uniform attractivity statement in 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 1.79, 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 definition of induced norm, $\|Fx\|_\infty \leq \|x\|_\infty$. Similarly, in components,

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

that 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 is immediate to see 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 1.18, any trajectory of $T_{\alpha,\delta}$, and hence also the trajectory $w : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$ of the averaging algorithm (1.5), 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 $\mathrm{diag}(\mathbb{R}^n)$.

For $k \in \mathbb{N}$ fixed, assume $w(k\delta)$ satisfies $V(w(k\delta)) > 0$. Given the averaging algorithm (1.5) 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 the 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 1.23 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 that:

- 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 $i \in U_M$,

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

  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 $i \in U_M$.
- 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 $\alpha \in]\alpha, 1]$. 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 either 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 1.58. ■

We conclude this section by establishing convergence to an individual point.

Proof (Proposition 1.62). We adopt the same notation as above, i.e., as in the proof of Theorem 1.58. Since $F(k) \in \mathcal{F}_\delta(\alpha)$, the set of sequence points $\{w(k\delta) \mid k \in \mathbb{N}\}$ belong 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 $k_l \in \mathbb{N}$, $w_i((k_l+k)\delta) \in [\min_i w_i(k_l\delta), \max_i w_i(k_l\delta)]^n$, $i \in \{1,\dots,n\}$ and $k \in \mathbb{N}$. Since $\lim_{l\to+\infty} w(k_l\delta) = c\mathbf{1}_n$, then $\lim_{l\to+\infty} [\min_i w_i(k_l\delta), \max_i w_i(k_l\delta)]^n = c\mathbf{1}_n$. 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 $\lim_{k\to+\infty} w(k\delta) = c\mathbf{1}_n$.

1.7.3 Proofs of Theorems 1.74 and 1.75

Proof (Theorem 1.74). 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, i.e., its largest singular value, is equal to the magnitude of its eigenvalue with largest magnitude. Based on this information and on the eigenvalue computation in Lemma 1.72, we compute

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

Because we assumed $|b| + 2|a| = 1$ and because $\cos(\frac{\pi}{n+1}) < 1$ for any $n \ge 2$, 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:

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

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

$$\ell > \frac{\log \varepsilon^{-1}}{-\log\left(|b| + 2|a|\cos\left(\frac{\pi}{n+1}\right)\right)}. \tag{1.11}$$

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 $\mathrm{Trid}_n(a,b,a)$. 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

$$\mathbf{v}_n = \sqrt{\frac{2}{n+1}} \begin{bmatrix} \sin\frac{\pi}{n+1} \\ \vdots \\ \sin\frac{n\pi}{n+1} \end{bmatrix} \in \mathbb{R}^n, \tag{1.12}$$

and note that, by Lemma 1.72(i), $\mathbf{v}_n$ is an eigenvector of $\mathrm{Trid}_n(a,b,a)$ 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 (1.11). This completes the proof of fact (i).

Next, we consider 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

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

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 $\ell \geq 2\beta + 2\alpha \max\{1, \log \alpha\}$. For, if $\ell \geq 2\alpha$, 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 $\ell \geq 2\beta + 2\alpha \log \alpha$. In summary, it is true that $\|x(\ell)\|_2 \leq \varepsilon \|x(0)\|_2$ whenever

$$\ell \geq 2\frac{\frac{a}{b} - \log \varepsilon}{-\log |b|} + 2\frac{n-1}{-\log |b|} \max\left\{1, \log \frac{n-1}{-\log |b|}\right\}.$$

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 1.72(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:

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

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

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

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 $\overline{\lambda}_n = b + (a+c)\cos\left(\frac{(n-1)2\pi}{n}\right) + \sqrt{-1}(c-a)\sin\left(\frac{(n-1)2\pi}{n}\right)$ of $\mathrm{Circ}_n(a, b, c)$, and its associated eigenvectors (cf. Lemma 1.72(ii))

$$\mathbf{v}_n = \begin{bmatrix} 1 \\ \omega \\ \vdots \\ \omega^{n-1} \end{bmatrix} \in \mathbb{C}^n, \quad \overline{\mathbf{v}}_n = \begin{bmatrix} 1 \\ \omega^{n-1} \\ \vdots \\ \omega \end{bmatrix} \in \mathbb{C}^n. \tag{1.13}$$

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 0. 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

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

This completes the proof of fact (iii). ■

Proof (Theorem 1.75). We prove fact (i) and observe that the proof of fact (ii) is analogous. Consider the change of coordinates

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

where $x'_{\text{ave}}(\ell) \in \mathbb{R}$ and $y(\ell) \in \mathbb{R}^{n-1}$. A quick calculation shows that $x'_{\text{ave}}(\ell) = \frac{1}{n}\mathbf{1}_n^T x(\ell)$, and the similarity transformation described in equation (1.10) implies

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

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

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

Consider the matrix in parenthesis 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 $\text{Trid}_{n-1}(a,b,a)$, and its eigenvectors are $\mathbf{1}_n$ and the eigenvectors of $\text{Trid}_{n-1}(a,b,a)$ (padded with an extra zero). These facts are sufficient to duplicate, step by step, the proof of fact (i) in Theorem 1.74. Therefore, the trajectory $\ell \mapsto (x(\ell) - x_{\text{ave}}(\ell)\mathbf{1}_n)$ satisfies the stated properties. ■

1.8 Exercises

E1.1 Show that the following sets of matrices are compact:
(i) the set of stochastic matrices,

(ii) the set of permutation matrices.

E1.2 Show that the spectral radius and the ∞-induced norm of a row-stochastic matrix is 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 1.6 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 1.3 to show that* 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.*

E1.3 Show that 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 1.2 as possible.
Hint: *A proof is contained in [Horn and Johnson, 1985]. A second proof method is based on combinatorics methods.*

E1.4 Let $f_1, \ldots, f_m : X \to X$ 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

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

Show that T is closed on X.
Hint: *Reason by contradiction.*

E1.5 **(Acyclic digraphs).** Let G be an acyclic digraph. Show that:
(i) G contains at least one source, i.e., a vertex without in-neighbors,
(ii) G contains at least one sink, i.e., a vertex without out-neighbors, and
(iii) in an appropriate ordering of the vertices of G, the adjacency matrix A is lower-triangular, i.e., all its entries above the main diagonal vanish.
Hint: *Order the vertices of G according to their distance to a sink.*

E1.6 **(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,
(iii) a digraph contains a directed spanning tree if and only if its condensation digraph contains a directed spanning tree.

E1.7 **(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).

E1.8 **(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:

$$F_1 = (\kappa I_n + D_{\mathrm{out}})^{-1}(\kappa I_n + A), \quad \text{for } \kappa \in \mathbb{R}_{>0},$$
$$F_2 = I_n - \varepsilon L, \quad \text{for } \varepsilon \in [0, \min\{(D_{\mathrm{out}})_{ii}^{-1} \mid i \in \{1,\dots,n\}\}[.$$

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,
(iv) under what conditions can a row-stochastic matrix be written in the form F_1, or F_2 for some appropriate digraph (and for some appropriate scalars κ and ε)?

E1.9 **(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 E1.8 for the construction of the matrix F_2,
(iii) express the exponential convergence factor $r_{\exp}(F_2)$ as a function of the eigenvalues of the Laplacian of G.

E1.10 **(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

$$\Phi_G(x) = \frac{1}{2}\sum_{i,j=1}^{n} a_{ij}(x_j - x_i)^2. \tag{E1.1}$$

Show that:
(i) the disagreement function is the quadratic form associated with the symmetric positive-semidefinite matrix

$$P(G) = \frac{1}{2}(D_{\mathrm{out}}(G) + D_{\mathrm{in}}(G) - A(G) - A(G)^T),$$

(ii) $P(G) = \frac{1}{2}\big(L(G) + L(\mathrm{rev}(G))\big)$.
Hint: *A sample proof is provided in [Gao et al., 2008].*

E1.11 **(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,
(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.

E1.12 Without relying upon the Geršgorin Disks Theorem 1.3, 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 1.32 relies upon Geršgorin Disks Theorem 1.3.)

E1.13 **(Properties of the BFS algorithm).** Prove Lemma 1.24.

E1.14 **(LCR algorithm).** Consider the LCR algorithm for leader election:
(i) give a UID assignment to each processor for which $\Omega(n^2)$ messages are sent
(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.

E1.15 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 upon the characterization in Propositions 1.29 and 1.30, 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,
(iii) show that the averaging algorithm associated with A_2 converges in a finite number of steps.

E1.16 Show that, for any $\alpha \in]0, 1]$, the set of non-degenerate matrices with respect to α is compact.

E1.17 **(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

$$\dot{x} = -L(G)x.$$

(i) what are the equilibrium points?
(ii) show that, if G is undirected, then the disagreement function (see Exercise E1.10) 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 1.17 and show that the solutions of the Laplacian flow converge to $\mathrm{diag}(\mathbb{R}^n)$,

(v) find an example G such that, with the notation in Exercise E1.10, 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.

E1.18 **(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 $\lim_{\alpha \to 0^+} f_\alpha(x) = \max\{x_1, \dots, x_n\}$,

(ii) $\lim_{\alpha \to +\infty} f_\alpha(x) = \lim_{\alpha \to -\infty} f_\alpha(x) = \frac{1}{n}(x_1 + \cdots + x_n)$.

Let $A \in \mathbb{R}^{n \times n}$ be a non-degenerate, doubly stochastic matrix whose associated digraph contains a globally reachable node. 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 $\ell \to +\infty$.

E1.19 **(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 1.5.2 and the theory of Markov chains. Some relevant references on Markov chains include [Seneta, 1981, 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.*

E1.20 **(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 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 bibliographical note, the variable π_i is referred to as the *belief* in the seminal work by Pearl [1988].

E1.21 Prove Lemma 1.77.

E1.22 **(Cyclic pursuit [Watton and Kydon, 1969]).** The "n-bugs problem" related to the *pursuit curves* from mathematics, inquires about what are the paths of n bugs, not aligned initially, 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 19th century. It was in [Watton and Kydon, 1969] when 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, Marshall et al., 2004]. Cyclic pursuit, has also been recently studied in the multi-agent and control literature, e.g., see [Bruckstein et al., 1991, Marshall et al., 2004, 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 we identify for convenience $n+1$ with 1. Denote by $p_i(\ell) = (r, \theta_i(\ell))$ the sequence of positions of bug i, initially at $p_i(0) = (r, \theta_i(0))$.

(i) 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 towards 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 $\mathrm{dist}_c(\theta_{i+1},\theta_i) = \mathrm{dist}_c(\theta_i,\theta_{i-1})$ for all $i \in \{1,\dots,n\}$. Observe that the bugs will approach this equally-spaced configuration while moving around the circle indefinitely.

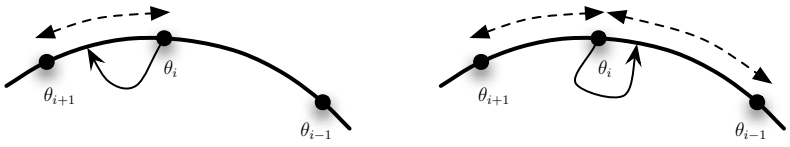
(ii) Suppose that each bug makes a compromise between chasing its closest counterclockwise neighbor and the closest clockwise neighbor, see Figure E1.1(b). In other words, each bug feels an attraction towards the closest counterclockwise and clockwise neighbors that can be described by the equation

$$\theta_i(\ell+1) = k\theta_{i+1}(\ell) + (1-2k)\theta_i(\ell) + k\theta_{i-1}(\ell), \quad \ell \in \mathbb{Z}_{\geq 0},$$

where $k \in [0,1]$.

a) Determine for which values of k the bugs converge to a configuration for which $\mathrm{dist}_c(\theta_{i+1},\theta_i) = \mathrm{dist}_c(\theta_i,\theta_{i-1})$ for all $i \in \{1,\dots,n\}$.

b) Show that the bugs will approach this equally-spaced configuration while each of them converges to a stationary position on the circle.



(a) (b)

Fig. E1.1. 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 constant of proportionality is k.

Hint: *Rewrite the systems in (i) and (ii) in terms of the inter-bug distances; that is, in terms of* $d_i(\ell) = \mathrm{dist}_c(\theta_{i+1}(\ell),\theta_i(\ell))$, $i \in \{1,\dots,n\}$, $\ell \in \mathbb{Z}_{\geq 0}$. *Find the matrices that describe the linear iterations in these new coordinates. Show that the agreement space, i.e., 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 1.72 might be of use in this regard. Regarding part (ii)b), recall that an exponentially decaying sequence is summable.*

2

Geometric models and optimization

This chapter presents various geometric objects and geometric optimization problems that have strong connections with motion coordination. Basic geometric notions such as polytopes, centers, partitions, and distances are ubiquitous in cooperative strategies, coordination tasks, and the interaction of robotic networks with the physical environment. The notion of Voronoi partition finds application in diverse areas such as wireless communications, signal compression, facility location, and mesh optimization. Proximity graphs provide a natural way to mathematically model the network interconnection topology resulting from the agents' sensing and/or communication capabilities. Finally, multicenter functions play the role of aggregate objective functions in geometric optimization problems. We introduce these concepts here in preparation for the later chapters.

The chapter is organized as follows. Section 2.1 introduces basic geometric constructions. Section 2.2 introduces the notion of proximity graphs and presents numerous examples. Finally, Section 2.3 presents geometric optimization problems and multicenter functions, paying special attention to the characterization of their smoothness properties and critical points.

2.1 Basic geometric notions

In this section, we gather some classical geometric constructions that will be invoked profusely throughout the book.

2.1.1 Polygons and polytopes

For $p, q \in \mathbb{R}^d$, we let $]p, q[= \{\lambda p + (1-\lambda)q \mid \lambda \in]0,1[\}$ and $[p, q] = \{\lambda p + (1-\lambda)q \mid \lambda \in [0,1]\}$ denote the *open* and *closed segment* with extreme points p and q, respectively. We let $H_{p,q} = \{x \in \mathbb{R}^d \mid \|x-p\|_2 \leq \|x-q\|_2\}$ denote the *closed halfspace* of $\mathbb{R}^d$ of points closer (in Euclidean distance) to p than to q. In the plane, we often refer to a halfspace as a *halfplane*.

As seen in Section 1.1.2, a set $S \subset \mathbb{R}^d$ is *convex* if, for any two points p, q in S, the closed segment $[p, q]$ is contained in S. The *convex hull* of a set is the smallest (with respect to the inclusion) convex set that contains it. We denote the convex hull of S by $\mathrm{co}(S)$. For $S = \{p_1, \ldots, p_n\}$ finite, the convex hull can be explicitly described as

$$\mathrm{co}(S) = \big\{\lambda_1 p_1 + \cdots + \lambda_n p_n \mid \lambda_i \geq 0 \text{ and } \sum_{i=1}^{n} \lambda_i = 1\big\}.$$

Given p and q in $\mathbb{R}^d$ and a convex closed set $Q \subset \mathbb{R}^d$ with $p \in Q$ (see Figure 2.1), define the *from-to-inside* function by

$$\mathrm{fti}(p, q, Q) = \begin{cases} q, & \text{if } q \in Q, \\ [p, q] \cap \partial Q, & \text{if } q \notin Q. \end{cases}$$

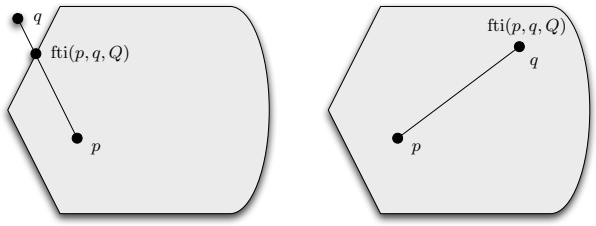


Fig. 2.1. Illustration of the from-to-inside function fti.

The function fti selects the point in the closed segment $[p, q]$ which is at the same time closest to q and inside Q. Note that fti depends continuously on its arguments.

A *polygon* is a set in $\mathbb{R}^2$ whose boundary is the union of a finite number of closed segments. A polygon is *simple* if its boundary, regarded as a curve, is non self-intersecting. We will only consider simple polygons. The closed segments composing the boundary of a polygon are called *edges*, and points resulting from the pairwise intersection between consecutive edges are called *vertices*. A convex polygon can be written as

(i) the convex hull of its set of vertices, or
(ii) the intersection of halfplanes defined by its edges.

Two vertices whose open segment is contained in the interior of the polygon define a *diagonal*. To each vertex of a polygon we associate an *interior* and an *exterior angle*. A vertex is *strictly convex* (resp. *strictly nonconvex*) if its interior angle is strictly smaller (resp. greater) than π radians. A polygon is

nonconvex if it has at least one strictly concave vertex. The *perimeter* of a polygon is the length of its boundary, i.e., the sum of the lengths of its edges.

A polytope is the generalization of the notion of polygon to $\mathbb{R}^d$, for $d \geq 3$. In this book, we will not consider nonconvex polytopes in dimension larger than 2. As for convex polygons, a *(convex) polytope* in $\mathbb{R}^d$ can be defined as either the convex hull of a finite set of points in $\mathbb{R}^d$ or the bounded intersection of finite set of halfspaces. A *face* of a polytope is the intersection between the polytope and the boundary of a closed halfspace that defines the polytope. The faces of dimensions 0, 1, and $d-1$ are called, respectively, *vertices*, *edges*, and *facets*. For a convex polytope Q, we will refer to them, respectively, by $\mathrm{Ve}(Q)$, $\mathrm{Ed}(Q)$, and $\mathrm{Fa}(Q)$.

2.1.2 Nonconvex geometry

In this section, we gather some basic notions on nonconvex geometry. We consider environments that include nonconvex polygons as a particular case.

We begin with some visibility notions. Given $S \subset \mathbb{R}^d$, two points $p, q \in S$ are *visible* to each other if the closed segment $[p, q]$ is contained in S. The *visibility set* $\mathrm{Vi}(p; S)$ is the set of all points in S visible from p. Given $r > 0$, the *range-limited visibility set* $\mathrm{Vi}_{\mathrm{disk}}(p; S) = \mathrm{Vi}(p; S) \cap \overline{B}(p, r)$ is the set of all points in S within a distance r and visible from p. The set S is *star-shaped* if there exists $p \in S$ such that $\mathrm{Vi}(p; S) = S$. The *kernel set* of S is comprised of all the points with this property, i.e., $\mathrm{kernel}(S) = \{p \in S \mid \mathrm{Vi}(p; S) = S\}$. Trivially, any convex set is star-shaped. Given $\delta \in \mathbb{R}_{>0}$, the δ-contraction of S is the set $S_\delta = \{p \in S \mid \mathrm{dist}(p, \partial S) \geq \delta\}$. Note that, if two points $p, q \in S$ are visible to each other in S_δ, then any point within distance δ of p and any point within distance δ of q are visible to each other. Figure 2.2 illustrates these visibility notions.

Next, we introduce various concavity notions. Given $S \subset \mathbb{R}^d$ connected and closed, $p \in \partial S$ is *strictly concave* if for any $\varepsilon \in \mathbb{R}_{>0}$, there exist $q_1, q_2 \in B(p, \varepsilon) \cap \partial S$ such that $[q_1, q_2] \not\subset S$. This definition coincides with the notion of strictly concave vertex when the set S is a polygon. A *strict concavity* of S is either an isolated strictly concave point or a concave arc, i.e., a connected set of strictly concave points. An *allowable environment* $S \subset \mathbb{R}^2$ is a set that satisfies the following properties: it is closed, simply connected, has a finite number of strict concavities and its boundary can be described by a continuous and piecewise continuously differentiable curve which is not differentiable at most at a finite number of points. Figure 2.3 shows a sample allowable environment. Given an allowable environment S, let v belong to a concave arc and such that the boundary of S is continuously differentiable at v. The *internal tangent halfplane* $H_S(v)$ is the closed halfplane whose boundary is tangent to ∂S at v and whose interior does not contain any points of the strict concavity; see Figure 2.3.

The following result presents an interesting property of allowable environments.

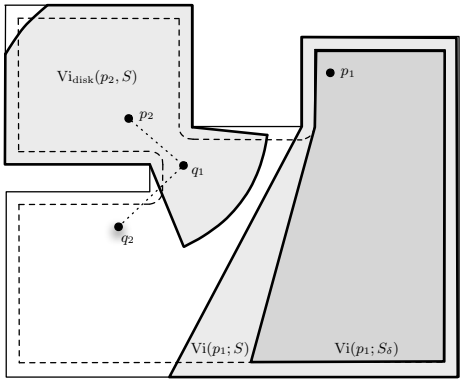


Fig. 2.2. Illustration of various visibility notions. The visibility set $\mathrm{Vi}(p_1; S)$ from p_1 in S, the visibility set $\mathrm{Vi}(p_1; S_\delta)$ from p_1 in S_δ, and the range-limited visibility set $\mathrm{Vi}_{\mathrm{disk}}(p_2; S)$ from p_2 in S are depicted in light gray. The dashed curve in the interior of S corresponds to the boundary of the δ-contraction of S. The points p_2 and q_1 are visible to each other in S_δ. The points q_1 and q_2 are visible to each other in S, but they are not visible to each other in S_δ.

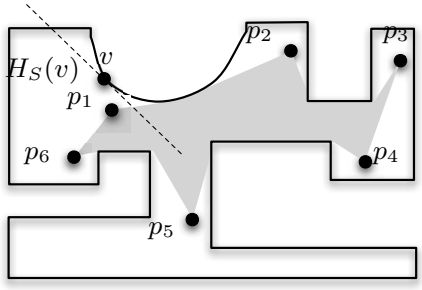


Fig. 2.3. An allowable environment S. The curved portion of the boundary is a concave arc. The vertices whose interior angle is $3\pi/2$ radians are isolated strictly concave points. The relative convex hull of $\{p_1, \ldots, p_6\}$ in S is depicted in light gray. Finally, the dashed line represents the boundary of the internal tangent halfplane $H_S(v)$ tangent to ∂S at v.

Lemma 2.1. *Given an allowable environment S, the δ-contraction S_δ is also allowable for sufficiently small $\delta \in \mathbb{R}_{>0}$ and does not have isolated strictly concave points. Furthermore, the boundary of S_δ is continuously differentiable at the concavities.*

Lemma 2.1 implies that the internal tangent halfplane is well-defined at any strict concavity of the δ-contraction S_δ.

A set $S \subset X$ is *relatively convex* in $X \subset \mathbb{R}^d$ if, for any two points p, q in S, the shortest curve in X that connects p and q is contained in S. Relatively

convex sets in $\mathbb{R}^d$ are just convex sets. The *relative convex hull* of a set S in X is the smallest (with respect to the operation of inclusion) relatively convex set in X that contains S, see Figure 2.3. We denote the relative convex hull of S in X by $\mathrm{rco}(S; X)$. The *(relative) perimeter* of S in X is the length of the shortest measurable closed curve contained in X that encloses S.

2.1.3 Geometric centers

Let $X = \mathbb{R}^d$, $X = \mathbb{S}^d$ or $X = \mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$, $d = d_1 + d_2$. Recall our convention (cf. Section 1.1.1) that, unless otherwise noted, $\mathbb{R}^d$ is endowed with the Euclidean distance, $\mathbb{S}^d$ is endowed with the geodesic distance, and $\mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$ is endowed with the Cartesian product distance $(\mathrm{dist}_2, \mathrm{dist}_g)$.

The *circumcenter* of a bounded set $S \subset X$, denoted by $\mathrm{CC}(S)$, is the center of the closed ball of minimum radius that contains S. The *circumradius* of S, denoted by $\mathrm{CR}(S)$, is the radius of this ball[1]. The circumcenter is always unique.

The computation of the circumcenter and the circumradius of a polytope $Q \subset \mathbb{R}^d$ is a strictly convex problem and in particular a quadratically constrained linear program in p (the center) and r (the radius). It consists of minimizing the radius r of the ball centered at p subject to the constraints that the distance between q and each of the polygon vertices is smaller than or equal to r. Formally, the problem can be expressed as follows.

$$\begin{aligned} &\text{minimize } \ r\,, \\ &\text{subject to } \ \|q - p\|_2^2 \leq r^2, \ \text{for all } q \in \mathrm{Ve}(Q). \end{aligned} \tag{2.1}$$

Next, we summarize some useful properties of the circumcenter in Euclidean space, see Exercise E2.1. In the following result, for $S \in \mathbb{F}(\mathbb{R}^d)$ with $d = 1$, we let $\mathrm{Ve}(\mathrm{co}(S))$ denote the set of extreme points of the interval $\mathrm{co}(S)$.

Lemma 2.2 (Properties of the circumcenter in Euclidean space). *Let $S = \{p_1, \ldots, p_n\} \in \mathbb{F}(\mathbb{R}^d)$ with $n \geq 2$. The following properties hold:*

(i) $\mathrm{CC}(S) \in \mathrm{co}(S) \setminus \mathrm{Ve}(\mathrm{co}(S))$*;*

(ii) if $p \in \mathrm{co}(S) \setminus \{\mathrm{CC}(S)\}$ and $r \in \mathbb{R}_{>0}$ are such that $S \subset \overline{B}(p, r)$, then $]p, \mathrm{CC}(S)[$ has a nonempty intersection with $\overline{B}(\frac{p+q}{2}, \frac{r}{2})$ for all $q \in \mathrm{co}(S)$.

Given $X = \mathbb{R}^d$, $X = \mathbb{S}^d$ or $X = \mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$, $d = d_1 + d_2$, the *incenter or Chebyshev center* of a compact set $S \subset X$, denoted by $\mathrm{IC}(S)$, is the set containing the centers of all closed balls of maximum radius contained in S. The *inradius* of S, denoted by $\mathrm{IR}(S)$, is the common radius of any of these balls.

[1] Note that the definition of circumcenter given here is in general different from the classical notion of circumcenter of a triangle, i.e., the center of the circle passing through the three vertices of the triangle.

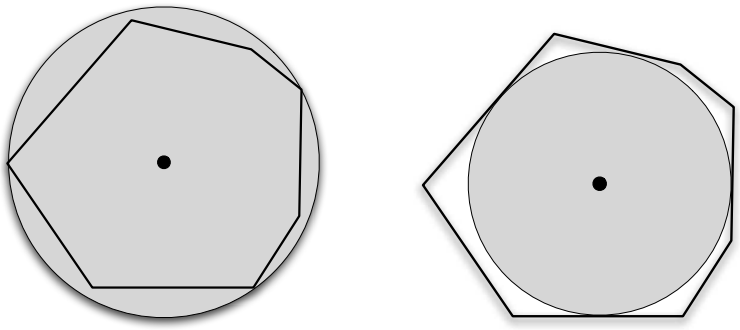


Fig. 2.4. Circumcenter and circumradius (left), and incenter and inradius (right) of a convex polygon.

The computation of the incenter and the inradius of a polytope $Q \subset \mathbb{R}^d$ is a convex problem, and in particular a linear program in p and r. It consists of maximizing the radius r of the ball centered at p subject to the constraints that the distance between p and each of the polytope facets is greater than or equal to r. Formally, the problem can be expressed as follows. For each $f \in \mathrm{Fa}(Q)$, select a point $q_f \in Q$ belonging to f. Then, we set

$$\begin{aligned} &\text{maximize } r\,, \\ &\text{subject to } (p - q_f) \cdot \mathrm{n}_{\mathrm{in},f} \geq r\,, \text{ for all } f \in \mathrm{Fa}(Q), \end{aligned} \tag{2.2}$$

where $\mathrm{n}_{\mathrm{in},f}$ denotes the normal to the face f pointing toward the interior of the polytope. The incenter of a polytope is not necessarily unique (consider, for instance, the case of a rectangle).

In Euclidean space, $X = \mathbb{R}^d$, we refer to a bounded measurable function $\phi : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ as a *density* on $\mathbb{R}^d$. The *(generalized) area* and the *centroid* of a bounded measurable set $S \subset \mathbb{R}^d$ with respect to ϕ, denoted respectively by $\mathrm{A}_\phi(S)$ and $\mathrm{CM}_\phi(S)$, are given by

$$\mathrm{A}_\phi(S) = \int_S \phi(q)dq, \qquad \mathrm{CM}_\phi(S) = \frac{1}{\mathrm{A}_\phi(S)} \int_S q\phi(q)dq.$$

When the function ϕ being used is clear from the context, we simply refer to the area and the centroid of S. The centroid can alternatively be defined as follows. Define the *polar moment of inertia* of S about $p \in S$ by

$$\mathrm{J}_\phi(S, p) = \int_S \|q - p\|_2^2 \phi(q)dq.$$

Then, the centroid of S is precisely the point $p \in S$ that minimizes the polar moment of inertia of S about p. This can be easily seen from the Parallel Axis Theorem [Hibbeler, 2003], that states

$$\mathrm{J}_\phi(S, p) = \mathrm{J}_\phi(S, \mathrm{CM}_\phi(S)) + \mathrm{A}_\phi(S)\|p - \mathrm{CM}_\phi(S)\|_2^2.$$

Remark 2.3 (Computation of geometric centers in the plane). The circumcenter, incenter, and centroid of a polygon can be computed in several ways. A simple procedure to compute the circumcenter consists of enumerating all pairs and triplets of vertices of the polygon, compute the centers and radiuses of the balls passing through them, and selecting the ball with the smallest radius that encloses the polygon. An alternative, more efficient way of computing the circumcenter, is to use the formulation (2.1). A convex quadratically constrained linear program is a particular case of a semidefinite-quadratic-linear program (SQLP). Several freely-available numerical packages exist to solve SQLP problems, e.g., SDPT3 [Tutuncu et al., 2003]. The computation of the incenter set of a polygon can be performed via linear programming using the formulation (2.2). Finally, the centroid of a polygon can be computed with any numerical routine that accurately approximates the integral of a function over a planar domain. •

2.1.4 Voronoi and range-limited Voronoi partitions

A *partition* of a set S is a subdivision of S into connected subsets that are disjoint except for their boundary. Formally, a partition of S is a collection of closed connected sets $\{W_1, \ldots, W_m\} \subset \mathbb{P}(S)$ that verify

$$S = \cup_{i=1}^{m} W_i \quad \text{and} \quad \operatorname{int}(W_j) \cap \operatorname{int}(W_k) = \emptyset,$$

for all $j, k \in \{1, \ldots, m\}$.

Definition 2.4 (Voronoi partition). Given a distance function $\operatorname{dist} : X \times X \to \mathbb{R}_{\geq 0}$, a set $S \subset X$ and n distinct points $\mathcal{P} = \{p_1, \ldots, p_n\}$ in S, the *Voronoi partition* of S generated by $\mathcal{P}$ is the collection of sets $\mathcal{V}(\mathcal{P}) = \{V_1(\mathcal{P}), \ldots, V_n(\mathcal{P})\} \subset \mathbb{P}(S)$ defined by, for each $i \in \{1, \ldots, n\}$,

$$V_i(\mathcal{P}) = \{q \in S \mid \operatorname{dist}(q, p_i) \leq \operatorname{dist}(q, p_j), \text{ for all } p_j \in \mathcal{P}\}. \quad \bullet$$

In other words, $V_i(\mathcal{P})$ is the set of the points of S that are closer to p_i than to any of the other points in $\mathcal{P}$. We will refer to $V_i(\mathcal{P})$ as the *Voronoi cell* of p_i.

Unless explicitly noted otherwise, we compute the Voronoi partition according to the following conventions:

- for $X = \mathbb{R}^d$, with respect to the Euclidean distance;
- for $X = \mathbb{S}^d$, with respect to the geodesic distance;
- for $X = \mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$, $d_1 + d_2 = d$, with respect to the Cartesian product distance determined by dist_2 on $\mathbb{R}^{d_1}$ and dist_g on $\mathbb{S}^{d_2}$.

In the Euclidean case, the Voronoi cell of p_i is equal to the intersection of half-spaces determined by p_i and the other locations in $\mathcal{P}$, and as such it is a convex polytope. The left plot in Figure 2.5 shows an example of the Voronoi partition of a convex polygon generated by 40 points. Figure 2.6 shows an example of the Voronoi partition of the circle generated by 5 points.

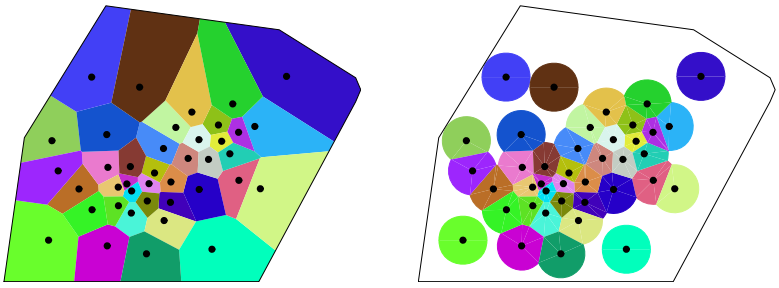


Fig. 2.5. Voronoi partition of a convex polygon (left) and r-limited Voronoi partition inside a convex polygon (right) generated by 40 points. The colored regions correspond to the Voronoi cells and the r-limited Voronoi cells, respectively, of each individual point.

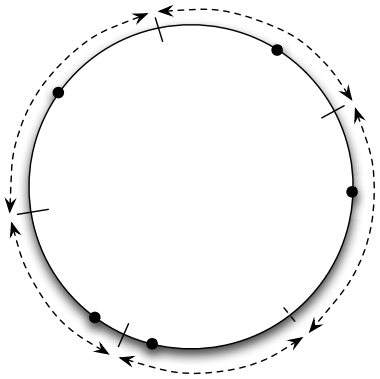


Fig. 2.6. Voronoi partition of the circle generated by 5 points. The dashed segments correspond to the Voronoi cells of each individual point.

Definition 2.5 (r-limited Voronoi partition). Given a distance function $\text{dist} : X \times X \to \mathbb{R}_{\geq 0}$, a set $S \subset X$, n distinct points $\mathcal{P} = \{p_1, \ldots, p_n\}$ in S, and a positive real number $r \in \mathbb{R}_{>0}$, the *r-limited Voronoi partition* inside S generated by $\mathcal{P}$ is the collection of sets $\mathcal{V}_r(\mathcal{P}) = \{V_{1,r}(\mathcal{P}), \ldots, V_{n,r}(\mathcal{P})\} \subset \mathbb{P}(S)$ defined by

$$V_{i,r}(\mathcal{P}) = V_i(\mathcal{P}) \cap \overline{B}(p_i, r), \quad i \in \{1, \ldots, n\}. \qquad \bullet$$

Note that the r-limited Voronoi partition inside S is precisely the Voronoi partition of the set $\cup_{i=1}^n \overline{B}(p_i, r) \cap S$. We will refer to $V_{i,r}(\mathcal{P})$ as the *r-limited Voronoi cell* of p_i. The right plot in Figure 2.5 shows an example of the r-limited Voronoi partition inside a convex polygon generated by 40 points.

Let $X = \mathbb{R}^d$, $X = \mathbb{S}^d$ or $X = \mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$, $d = d_1 + d_2$. Given a density ϕ on X, a set of n distinct points $\mathcal{P} = \{p_1, \ldots, p_n\}$ in $S \subset X$ is a

(i) *centroidal Voronoi configuration* if each point is the centroid of its own Voronoi cell, i.e., $p_i = \mathrm{CM}_\phi(V_i(\mathcal{P}))$;
(ii) *r-limited centroidal Voronoi configuration*, for $r \in \mathbb{R}_{>0}$, if each point is the centroid of its own r-limited Voronoi cell, that is, $p_i = \mathrm{CM}_\phi(V_{i,r}(\mathcal{P}))$. If $r \geq \mathrm{diam}(S)$, then an r-limited centroidal Voronoi configuration is a centroidal Voronoi configuration;
(iii) *circumcenter Voronoi configuration* if each point is the circumcenter of its own Voronoi cell, i.e., $p_i = \mathrm{CC}(V_i(\mathcal{P}))$;
(iv) *incenter Voronoi configuration* if each point is an incenter of its own Voronoi cell, i.e., $p_i \in \mathrm{IC}(V_i(\mathcal{P}))$.

Figure 2.7 presents an illustration of the various notions of center Voronoi configurations.

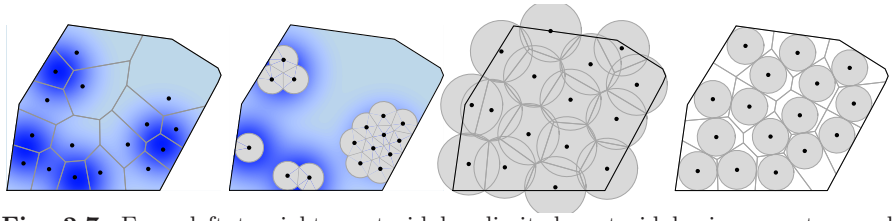


Fig. 2.7. From left to right, centroidal, r-limited centroidal, circumcenter, and incenter Voronoi configurations composed by 16 points in a convex polygon. Darker blue-colored areas correspond to higher values of the density ϕ.

2.2 Proximity graphs

Roughly speaking, a proximity graph is a graph whose vertex set is a set of distinct points and whose edge set is a function of the relative locations of the point set. Proximity graphs appear in computational geometry. In this section we study this important notion in detail following the presentation by Cortés et al. [2005].

Definition 2.6 (Proximity graph). Let $S \subset X$, 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 $d_1 + d_2 = d$. Let $\mathbb{G}(S)$ be the set of all undirected graphs whose vertex set is an element of $\mathbb{F}(S)$. A *proximity graph* $\mathcal{G} : \mathbb{F}(S) \to \mathbb{G}(S)$ associates to a set of distinct points $\mathcal{P} = \{p_1, \ldots, p_n\} \subset S$ an undirected graph with vertex set $\mathcal{P}$ and whose edge set is $\mathcal{E}_\mathcal{G}(\mathcal{P}) \subseteq \{(p, q) \in \mathcal{P} \times \mathcal{P} \mid p \neq q\}$. •

Note that in a proximity graph a point cannot be its own neighbor. From this definition, we observe that the distinguishing feature of proximity graphs is that their edge sets change with the location of their vertices. It is also

possible to define proximity graphs that associate to each point set a digraph, but we will not consider them here.

Examples of proximity graphs on X, where recall that $\text{dist} = \text{dist}_2$ if $X = \mathbb{R}^d$, $\text{dist} = \text{dist}_g$ if $X = \mathbb{S}^d$, and $\text{dist} = (\text{dist}_2, \text{dist}_g)$ if $X = \mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$, include the following:

(i) the *complete graph* $\mathcal{G}_{\text{cmplt}}$ where any two points are neighbors. When convenient, we may view the complete graph as weighted by assigning the weight $\text{dist}(p_i, p_j)$ to the edge $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{cmplt}}}(\mathcal{P})$;

(ii) the *r-disk graph* $\mathcal{G}_{\text{disk}}(r)$, for $r \in \mathbb{R}_{>0}$, where two points are neighbors if they are located within a distance r, that is, $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{disk}}(r)}(\mathcal{P})$ if $\text{dist}(p_i, p_j) \leq r$;

(iii) the *Delaunay graph* $\mathcal{G}_{\text{D}}$, where two points are neighbors if their corresponding Voronoi cells intersect, that is, $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{D}}}(\mathcal{P})$ if $V_i(\mathcal{P}) \cap V_j(\mathcal{P}) \neq \emptyset$;

(iv) the *r-limited Delaunay graph* $\mathcal{G}_{\text{LD}}(r)$, for $r \in \mathbb{R}_{>0}$, where two points are neighbors if their corresponding $\frac{r}{2}$-limited Voronoi cells intersect, that is, $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{LD}}(r)}(\mathcal{P})$ if $V_{i,\frac{r}{2}}(\mathcal{P}) \cap V_{j,\frac{r}{2}}(\mathcal{P}) \neq \emptyset$;

(v) the *relative neighborhood graph* $\mathcal{G}_{\text{RN}}$, where two points are neighbors if their associated open lune (cf. Section 1.1.1) does not contain any point in $\mathcal{P}$, that is, $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{RN}}}(\mathcal{P})$ if $p_k \notin B(p_i, \text{dist}(p_i, p_j)) \cap B(p_j, \text{dist}(p_i, p_j))$ for all $p_k \in \mathcal{P}$, $k \notin \{i, j\}$;

Figure 2.8 shows examples of these proximity graphs in the plane.

Additional examples of proximity graphs in the Euclidean space include:

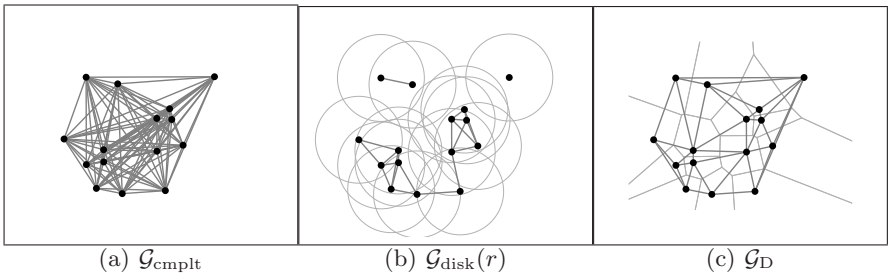
(vi) the *Gabriel graph* $\mathcal{G}_{\text{G}}$, where two points are neighbors if the ball centered at their midpoint and passing through both of them does not contain any point in $\mathcal{P}$, that is, $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{G}}}(\mathcal{P})$ if $p_k \notin B\big(\frac{p_i+p_j}{2}, \frac{\text{dist}(p_i,p_j)}{2}\big)$ for all $p_k \in \mathcal{P}$;

(vii) the *r-∞-disk graph* $\mathcal{G}_{\infty\text{-disk}}(r)$, for $r \in \mathbb{R}_{>0}$, where two points are neighbors if they are located within a L^∞-distance r, that is, $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\infty\text{-disk}}(r)}(\mathcal{P})$ if $\text{dist}_\infty(p_i, p_j) \leq r$;

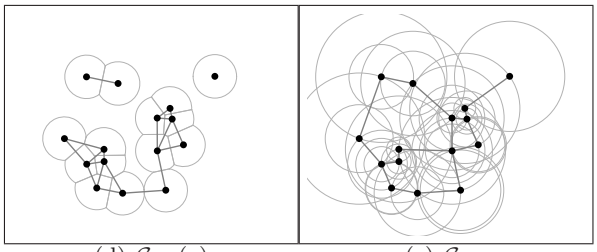
(viii) the *Euclidean minimum spanning tree* $\mathcal{G}_{\text{EMST},\mathcal{G}}$ of a proximity graph $\mathcal{G}$, that assigns to each $\mathcal{P}$ a minimum-weight spanning tree (cf. Section 1.3.4) of $\mathcal{G}(\mathcal{P})$ with weighted adjacency matrix $a_{ij} = \|p_i - p_j\|_2$, for $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}}(\mathcal{P})$. If $\mathcal{G}(\mathcal{P})$ is not connected, then $\mathcal{G}_{\text{EMST},\mathcal{G}}(\mathcal{P})$ is the union of Euclidean minimum spanning trees of its connected components. When $\mathcal{G}$ is the complete graph, we simply denote the Euclidean minimum spanning tree by $\mathcal{G}_{\text{EMST}}$;

(ix) the *visibility graph* $\mathcal{G}_{\text{vis},Q}$ in an allowable environment Q in $\mathbb{R}^2$, where two points are neighbors if they are visible to each other, that is, $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{vis},Q}}(\mathcal{P})$ if the closed segment $[p_i, p_j]$ from p_i to p_j is contained in Q.

(x) the *range-limited visibility graph* $\mathcal{G}_{\text{vis-disk},Q}$ in an allowable environment Q in $\mathbb{R}^2$, where two points are neighbors if they are visible to each other and their distance is no more than r, that is, $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{vis-disk},Q}}(\mathcal{P})$ if and only if $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{vis},Q}}(\mathcal{P})$ and $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{disk}}(r)}(\mathcal{P})$.



(a) $\mathcal{G}_{\text{cmplt}}$ (b) $\mathcal{G}_{\text{disk}}(r)$ (c) $\mathcal{G}_{\text{D}}$



(d) $\mathcal{G}_{\text{LD}}(r)$ (e) $\mathcal{G}_{\text{RN}}$

Fig. 2.8. Proximity graphs in $\mathbb{R}^2$. From left to right, in the first row, complete, r-disk, and Delaunay, and in the second row, r-limited Delaunay and relative neighborhood for a set of 15 points. When appropriate, the geometric objects determining the edge relationship are plotted in lighter gray.

Figure 2.9 shows examples of these proximity graphs in the plane.

Figure 2.10 shows some examples of proximity graphs in three-dimensional space.

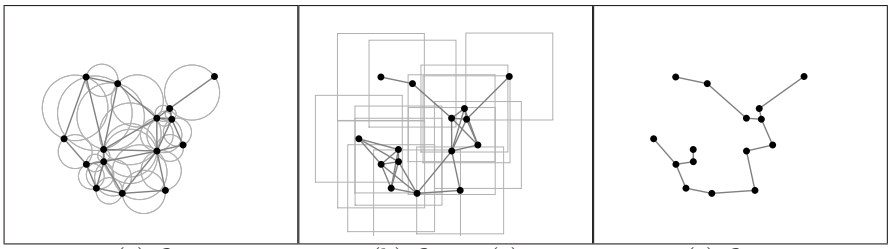
As for standard graphs, let us alternatively describe the edge set by means of the sets of neighbors of the individual graph vertices. To each proximity graph $\mathcal{G}$, each $p \in X$ and each $\mathcal{P} = \{p_1, \ldots, p_n\} \in \mathbb{F}(X)$, we associate the *set of neighbors* map $\mathcal{N}_{\mathcal{G}} : X \times \mathbb{F}(X) \to \mathbb{F}(X)$ defined by

$$\mathcal{N}_{\mathcal{G}}(p, \mathcal{P}) = \{q \in \mathcal{P} \mid (p, q) \in \mathcal{E}_{\mathcal{G}}(\mathcal{P} \cup \{p\})\}.$$

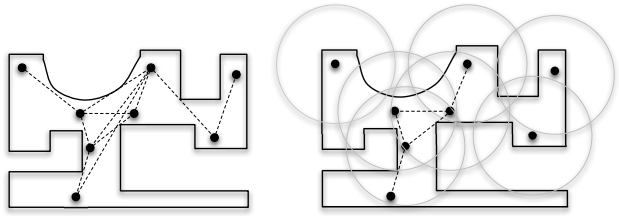
Typically, p is a point in $\mathcal{P}$, but the definition is well-posed for any $p \in X$. Under the assumption that $\mathcal{P}$ does not contain repeated elements, the definition will not lead to counterintuitive interpretations later. Given $p \in X$, it is convenient to define the map $\mathcal{N}_{\mathcal{G},p} : \mathbb{F}(X) \to \mathbb{F}(X)$ by $\mathcal{N}_{\mathcal{G},p}(\mathcal{P}) = \mathcal{N}_{\mathcal{G}}(p, \mathcal{P})$.

A proximity graph $\mathcal{G}_1$ is a *subgraph* of a proximity graph $\mathcal{G}_2$, denoted $\mathcal{G}_1 \subset \mathcal{G}_2$, if $\mathcal{G}_1(\mathcal{P})$ is a subgraph of $\mathcal{G}_2(\mathcal{P})$ for all $\mathcal{P} \in \mathbb{F}(X)$. The following result summarizes the subgraph relationships in the Euclidean case among the various proximity graphs introduced above.

Theorem 2.7 (Subgraph relationships among some standard proximity graphs on $\mathbb{R}^d$). *For $r \in \mathbb{R}_{>0}$, the following statements hold:*

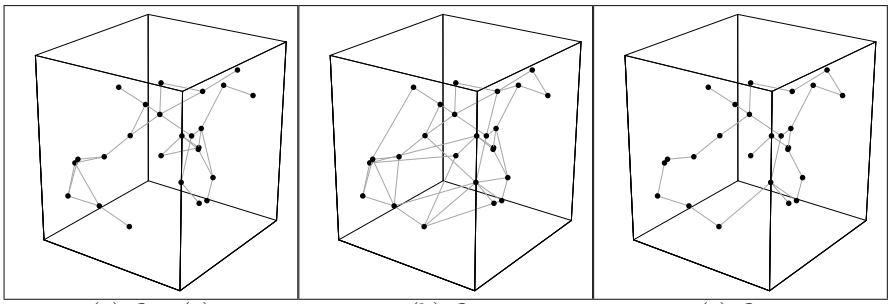


(a) $\mathcal{G}_{\mathrm{G}}$ (b) $\mathcal{G}_{\infty\text{-disk}}(r)$ (c) $\mathcal{G}_{\mathrm{EMST}}$



(d) $\mathcal{G}_{\mathrm{vis},Q}$ (e) $\mathcal{G}_{\text{vis-disk},Q}$

Fig. 2.9. Proximity graphs in $\mathbb{R}^2$. From left to right, in the first row, Gabriel, r-∞-disk, and Euclidean minimum spanning tree graphs for a set of 15 points, and in the second row, visibility and range-limited visibility graphs for a set of 8 agents in an allowable environment. When appropriate, the geometric objects determining the edge relationship are plotted in lighter gray.



(a) $\mathcal{G}_{\mathrm{disk}}(r)$ (b) $\mathcal{G}_{\mathrm{G}}$ (c) $\mathcal{G}_{\mathrm{RN}}$

Fig. 2.10. Proximity graphs in $\mathbb{R}^3$. From left to right, r-disk, relative neighborhood, and Gabriel graphs for a set of 25 points.

(i) $\mathcal{G}_{\mathrm{EMST}} \subset \mathcal{G}_{\mathrm{RN}} \subset \mathcal{G}_{\mathrm{G}} \subset \mathcal{G}_{\mathrm{D}}$;
(ii) $\mathcal{G}_{\mathrm{G}} \cap \mathcal{G}_{\mathrm{disk}}(r) \subset \mathcal{G}_{\mathrm{LD}}(r) \subset \mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r)$.

Remark 2.8. The inclusion $\mathcal{G}_{\mathrm{LD}}(r) \subset \mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r)$ is in general strict; this counterintuitive fact is discussed in Exercise E2.2. •

Note that, since $\mathcal{G}_{\mathrm{EMST}}$ is by definition connected, Theorem 2.7(i) implies that $\mathcal{G}_{\mathrm{RN}}$, $\mathcal{G}_{\mathrm{G}}$ and $\mathcal{G}_{\mathrm{D}}$ are connected. Regarding the connectivity properties of $\mathcal{G}_{\mathrm{disk}}(r)$, we have the following result, see [Cortés et al., 2005, 2006].

Theorem 2.9 (Connectivity properties of some standard proximity graphs on $\mathbb{R}^d$). *For $r \in \mathbb{R}_{>0}$, the following statements hold:*

(i) $\mathcal{G}_{\mathrm{EMST}} \subset \mathcal{G}_{\mathrm{disk}}(r)$ *if and only if* $\mathcal{G}_{\mathrm{disk}}(r)$ *is connected;*
(ii) $\mathcal{G}_{\mathrm{EMST}} \cap \mathcal{G}_{\mathrm{disk}}(r)$, $\mathcal{G}_{\mathrm{RN}} \cap \mathcal{G}_{\mathrm{disk}}(r)$, $\mathcal{G}_{\mathrm{G}} \cap \mathcal{G}_{\mathrm{disk}}(r)$ *and* $\mathcal{G}_{\mathrm{LD}}(r)$ *have the same connected components as* $\mathcal{G}_{\mathrm{disk}}(r)$ *(i.e., for all point sets* $\mathcal{P} \in \mathbb{F}(\mathbb{R}^d)$*, all graphs have the same number of connected components consisting of the same vertices).*

Note that in Theorem 2.9, fact (ii) implies (i). However, the proof of fact (ii) requires fact (i), see Section 2.5.2.

2.2.1 Spatially distributed proximity graphs

We now consider the following loosely-stated question: when a does a given proximity graph encode sufficient information to compute another proximity graph. For instance, if a node knows the position of its neighbors in the complete graph (that is, of every other node in the graph), then it is clear that the node can compute its neighbors with respect to any proximity graph. Let us formalize this idea. A proximity graph $\mathcal{G}_1$ is *spatially distributed over* a proximity graph $\mathcal{G}_2$ if, for all $p \in \mathcal{P}$,

$$\mathcal{N}_{\mathcal{G}_1,p}(\mathcal{P}) = \mathcal{N}_{\mathcal{G}_1,p}\big(\mathcal{N}_{\mathcal{G}_2,p}(\mathcal{P})\big),$$

that is, any node informed about the location of its neighbors with respect to $\mathcal{G}_2$ can compute its set of neighbors with respect to $\mathcal{G}_1$.

Clearly, any proximity graph is spatially distributed over the complete graph. It is straightforward to deduce that if $\mathcal{G}_1$ is spatially distributed over $\mathcal{G}_2$, then $\mathcal{G}_1$ is a subgraph of $\mathcal{G}_2$. The converse is in general not true. For instance, $\mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r)$ is a subgraph of $\mathcal{G}_{\mathrm{disk}}(r)$, but $\mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r)$ is not spatially distributed over $\mathcal{G}_{\mathrm{disk}}(r)$, see Exercise E2.3.

The following result identifies proximity graphs which are spatially distributed over $\mathcal{G}_{\mathrm{disk}}(r)$.

Proposition 2.10. *The proximity graphs* $\mathcal{G}_{\mathrm{RN}} \cap \mathcal{G}_{\mathrm{disk}}(r)$, $\mathcal{G}_{\mathrm{G}} \cap \mathcal{G}_{\mathrm{disk}}(r)$, *and* $\mathcal{G}_{\mathrm{LD}}(r)$ *are spatially distributed over* $\mathcal{G}_{\mathrm{disk}}(r)$.

Remark 2.11 (Computation of the Delaunay graph over the r-disk graph). In general, for a fixed $r \in \mathbb{R}_{>0}$, $\mathcal{G}_{\mathrm{D}}$ is not spatially distributed over $\mathcal{G}_{\mathrm{disk}}(r)$. However, for a given $\mathcal{P} \in \mathbb{F}(X)$, it is always possible find r such that $\mathcal{G}_{\mathrm{D}}(P)$ is spatially distributed over $\mathcal{G}_{\mathrm{disk}}(r)(\mathcal{P})$. This is a consequence of the following observations. Given $\mathcal{P} \in \mathbb{F}(X)$, define the convex sets

$$W(p_i, r) = \overline{B}(p_i, r) \cap \big(\cap_{\|p_i - p_j\| \leq r} H_{p_i, p_j} \big), \quad i \in \{1, \ldots, n\},$$

where recall that $H_{p,x}$ is the half-space of points q in $\mathbb{R}^d$ with the property that $\|q - p\|_2 \leq \|q - x\|_2$. Note that the intersection $\overline{B}(p_i, r) \cap V_i$ is a subset of $W(p_i, r)$. Provided r is twice as large as the maximum distance between p_i and the vertices of $W(p_i, r)$, then all Delaunay neighbors of p_i are within distance r from p_i. Equivalently, the half-space $H_{p_i,p}$ determined by p_i and a point p outside $\overline{B}(p_i, r)$ does not intersect $W(p_i, r)$. Therefore, the equality $V_i = W(p_i, r)$ holds. For node $i \in \{1, \ldots, n\}$, the minimum adequate radius is then

$$r_{i,\min} = 2 \max\{\|p_i - q\|_2 \mid q \in W(p_i, r_{i,\min})\}.$$

The minimum adequate radius across the overall network is then $r_{\min} = \max_{i \in \{1,\ldots,n\}} r_{i,\min}$. The algorithm presented in [Cortés et al., 2004] builds on these observations to compute the Voronoi partition of a bounded set generated by a pointset in a distributed way. •

The locally-cliqueless graph of a proximity graph

Given a proximity graph, it is sometimes useful to construct another proximity graph that has fewer edges and the same number of connected components. This is certainly the case when optimizing multi-agent cost functions in which the proximity graph edges describe pairwise constraints between agents. Additionally, the construction of the new proximity graph should be spatially distributed over the original proximity graph. Here, we present the notion of locally-cliqueless graph of a proximity graph.

Let $\mathcal{G}$ be a proximity graph in the Euclidean space. The *locally-cliqueless graph* $\mathcal{G}_{\mathrm{lc},\mathcal{G}}$ *of* $\mathcal{G}$ is the proximity graph defined by: $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\mathrm{lc},\mathcal{G}}}(\mathcal{P})$ if $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}}(\mathcal{P})$ and

$$(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\mathrm{EMST}}}(\mathcal{P}'),$$

for any maximal clique $\mathcal{P}'$ of (p_i, p_j) in $\mathcal{G}$. Figure 2.11 shows an illustration of this notion. The properties of this construction are summarized in the following result, see [Ganguli et al., 2007b].

Theorem 2.12 (Properties of the locally-cliqueless graph). *Let $\mathcal{G}$ be a proximity graph in the Euclidean space. Then, the following statements hold:*

(i) $\mathcal{G}_{\mathrm{EMST},\mathcal{G}} \subseteq \mathcal{G}_{\mathrm{lc},\mathcal{G}} \subseteq \mathcal{G}$;

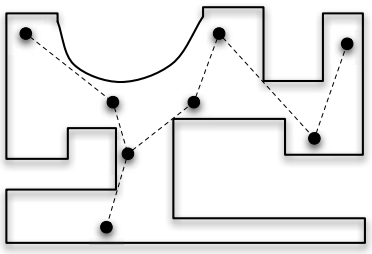


Fig. 2.11. Locally-cliqueless graph $\mathcal{G}_{\mathrm{lc},\mathcal{G}_{\mathrm{vis},Q}}$ of the visibility graph $\mathcal{G}_{\mathrm{vis},Q}$ for the node configuration shown in Figure 2.9(d).

(ii) $\mathcal{G}_{\mathrm{lc},\mathcal{G}}$ *has the same connected components as* $\mathcal{G}$*;*
(iii) For $\mathcal{G} = \mathcal{G}_{\mathrm{disk}}(r)$*,* $\mathcal{G}_{\mathrm{vis},Q}$*, and* $\mathcal{G}_{\text{vis-disk},Q}$*, where* $r \in \mathbb{R}_{>0}$ *and* Q *is an allowable environment,* $\mathcal{G}_{\mathrm{lc},\mathcal{G}}$ *is spatially distributed over* $\mathcal{G}$*.*

In general, the inclusions in Theorem 2.12(i) are strict.

2.2.2 Proximity graphs over tuples of points

The notion of proximity graph is defined for sets of distinct points $\mathcal{P} = \{p_1, \dots, p_n\}$. However, we will be interested in considering tuples of elements of X of the form $P = (p_1, \dots, p_n)$, where p_i corresponds to the position of an agent i of a robotic network. In principle, note that the tuple P might contain coincident points. In order to reconcile this mismatch between sets and tuples, we will do the following.

Let $i_{\mathbb{F}} : X^n \to \mathbb{F}(X)$ be the natural immersion of X^n into $\mathbb{F}(X)$, that is, $i_{\mathbb{F}}(P)$ is the point set that contains only the distinct points in $P = (p_1, \dots, p_n)$. Note that $i_{\mathbb{F}}$ is invariant under permutations of its arguments and that the cardinality of $i_{\mathbb{F}}(p_1, \dots, p_n)$ is in general less than or equal to n. In what follows, $\mathcal{P} = i_{\mathbb{F}}(P)$ will always denote the point set associated to $P \in X^n$. Using the natural inclusion, the notion of proximity graphs can be naturally extended as follows: given $\mathcal{G}$, we define (with a slight abuse of notation)

$$\mathcal{G} = \mathcal{G} \circ i_{\mathbb{F}} : X^n \to \mathbb{G}(X).$$

Additionally, we define the set of neighbors map $\mathcal{N}_{\mathcal{G}} : X \times X^n \to \mathbb{F}(X)$ by

$$\mathcal{N}_{\mathcal{G}}(p, (p_1, \dots, p_n)) = \mathcal{N}_{\mathcal{G}}(p, i_{\mathbb{F}}(p_1, \dots, p_n)).$$

Note that, according to this definition, coincident points in the tuple $(p_1, \dots, p_n)$ will have the same set of neighbors. As before, it is convenient to define the shorthand notation $\mathcal{N}_{\mathcal{G},p} : X^n \to \mathbb{F}(X)$, $\mathcal{N}_{\mathcal{G},p}(P) = \mathcal{N}_{\mathcal{G}}(p, P)$ for $p \in X$.

2.2.3 Spatially distributed maps

Given a set Y and a proximity graph $\mathcal{G}$, a map $T : X^n \to Y^n$ is *spatially distributed over* $\mathcal{G}$ if there exists a map $\tilde{T} : X \times \mathbb{F}(X) \to Y$, with the property that, for all $(p_1, \ldots, p_n) \in X^n$ and for all $j \in \{1, \ldots, n\}$,

$$T_j(p_1, \ldots, p_n) = \tilde{T}(p_j, \mathcal{N}_{\mathcal{G},p_j}(p_1, \ldots, p_n)),$$

where T_j denotes the jth-component of T. In other words, the jth component of a spatially distributed map at $(p_1, \ldots, p_n)$ can be computed with only the knowledge of the vertex p_j and the neighboring vertices in the undirected graph $\mathcal{G}(P)$.

When studying coordination tasks and coordination algorithms, it will be relevant to characterize the spatially distributed features of functions, vector fields and set-valued maps with respect to suitable proximity graphs.

Remark 2.13. Note that the proximity graph $\mathcal{G}_1$ is spatially distributed over the proximity graph $\mathcal{G}_2$ if and only if the map

$$P \in X^n \mapsto (\mathcal{N}_{\mathcal{G}_1,p_1}(P), \ldots, \mathcal{N}_{\mathcal{G}_1,p_n}(P)) \in \mathbb{F}(X)^n$$

is spatially distributed over $\mathcal{G}_2$. •

2.3 Geometric optimization problems and multicenter functions

In this section we consider various interesting geometric optimization problems. By geometric optimization, we mean an optimization problem induced by a collection of geometric objects, see [Boltyanski et al., 1999]. We shall pay particular attention to facility location problems, in which service sites are spatially allocated to fulfill a particular request.

2.3.1 Expected-value multicenter functions

Let $S \subset \mathbb{R}^d$ be a bounded environment of interest, and consider a density function $\phi : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$. For the discussion of this section, only the value of ϕ restricted to S is of interest. One can regard ϕ as a function measuring the probability that some event takes place over the environment. The larger the value of $\phi(q)$, the more important the location q is. We refer to a non-increasing and piecewise continuously differentiable function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}$, possibly with finite jump discontinuities, as a *performance*. Performance functions describe the utility of placing a node at a certain distance from a location in the environment. The smaller the distance, the larger the value of f, i.e., the better the performance. For instance, in servicing problems, performance functions can encode the travel time or the energy expenditure required to service a

specific destination. In sensing problems, performance functions can encode the signal-to-noise-ratio between a source with unknown location and a sensor attempting to locate it.

Given a bounded measurable set $S \subset \mathbb{R}^d$, a density function ϕ, and a performance function f, let us consider the expected value of the coverage over any point in S provided by a set of points $p_1, \dots, p_n$. Formally, we define the *expected-value multicenter* function $\mathcal{H}_{\text{exp}} : S^n \to \mathbb{R}$ by

$$\mathcal{H}_{\text{exp}}(p_1, \dots, p_n) = \int_S \max_{i \in \{1,\dots,n\}} f(\|q - p_i\|_2)\phi(q)dq. \tag{2.3}$$

The definition of $\mathcal{H}_{\text{exp}}$ can be read as follows: for each location $q \in S$, consider the best coverage of q among those provided by each of the nodes $p_1, \dots, p_n$, which corresponds to the value $\max_{i\in\{1,\dots,n\}} f(\|q-p_i\|_2)$. Then, weigh the performance by the importance $\phi(q)$ of the location q. Finally, sum the resulting quantity over all the locations of the environment S to obtain $\mathcal{H}_{\text{exp}}(p_1, \dots, p_n)$ as a measure of the overall coverage provided by $p_1, \dots, p_n$.

Given the meaning of $\mathcal{H}_{\text{exp}}$, we seek to solve the following geometric optimization problem

$$\text{maximize } \mathcal{H}_{\text{exp}}(p_1, \dots, p_n), \tag{2.4}$$

that is, we seek to determine a set of configurations $p_1, \dots, p_n$ that maximize the value of the multicenter function $\mathcal{H}_{\text{exp}}$. An equivalent formulation of this problem is referred to as a *continuous p-median problem* in the literature on facility location, e.g., see [Drezner, 1995]. In our discussion, we will pay special attention to the case when $n = 1$, that we term the *1-center* problem. For the purpose of solving (2.4), note that we can assume that the performance function satisfies $f(0) = 0$. This can be done without loss of generality, since for any $c \in \mathbb{R}$, one has

$$\int_S \max_{i \in \{1,\dots,n\}} (f(\|q - p_i\|_2) + c)\phi(q)dq = \mathcal{H}_{\text{exp}}(p_1, \dots, p_n) + c\,\mathrm{A}_\phi(S).$$

The expected-value multicenter function can be alternatively described in terms of the Voronoi partition of S generated by $\mathcal{P} = \{p_1, \dots, p_n\}$. Let us define the set

$$\mathcal{S}_{\text{coinc}} = \{(p_1, \dots, p_n) \in (\mathbb{R}^d)^n \mid p_i = p_j \text{ for some } i \neq j\},$$

consisting of tuples of n points where some of them are repeated. Then, for $(p_1, \dots, p_n) \in S^n \setminus \mathcal{S}_{\text{coinc}}$, one has

$$\mathcal{H}_{\text{exp}}(p_1, \dots, p_n) = \sum_{i=1}^{n} \int_{V_i(\mathcal{P})} f(\|q - p_i\|_2)\phi(q)dq. \tag{2.5}$$

This expression of $\mathcal{H}_{\text{exp}}$ is appealing because it clearly shows the result of the overall coverage of the environment as the aggregate contribution of all

individual nodes. If $(p_1, \ldots, p_n) \in \mathcal{S}_{\text{coinc}}$, then a similar decomposition of $\mathcal{H}_{\text{exp}}$ can be written in terms of the distinct points $\mathcal{P} = i_{\mathbb{F}}(p_1, \ldots, p_n)$.

Inspired by the expression (2.5), let us define a more general version of the expected-value multicenter function. Given $(p_1, \ldots, p_n) \in S^n$ and a partition $\{W_1, \ldots, W_n\} \subset \mathbb{P}(S)$ of S, let

$$\mathcal{H}_{\text{exp}}(p_1, \ldots, p_n, W_1, \ldots, W_n) = \sum_{i=1}^{n} \int_{W_i} f(\|q - p_i\|_2)\phi(q)dq. \tag{2.6}$$

Notice that $\mathcal{H}_{\text{exp}}(p_1, \ldots, p_n) = \mathcal{H}_{\text{exp}}(p_1, \ldots, p_n, V_1(\mathcal{P}), \ldots, V_n(\mathcal{P}))$, for all $(p_1, \ldots, p_n) \in S^n \setminus \mathcal{S}_{\text{coinc}}$. Moreover, one can establish the following optimality result, see [Du et al., 1999].

Proposition 2.14 ($\mathcal{H}_{\text{exp}}$-optimality of the Voronoi partition). *Let $\mathcal{P} = \{p_1, \ldots, p_n\} \in \mathbb{F}(S)$. For any performance function f and for any partition $\{W_1, \ldots, W_n\} \subset \mathbb{P}(S)$ of S,*

$$\mathcal{H}_{\text{exp}}(p_1, \ldots, p_n, V_1(\mathcal{P}), \ldots, V_n(\mathcal{P})) \geq \mathcal{H}_{\text{exp}}(p_1, \ldots, p_n, W_1, \ldots, W_n),$$

that is, the Voronoi partition $\mathcal{V}(\mathcal{P})$ is optimal for $\mathcal{H}_{\text{exp}}$ among all partitions of S.

Proof. Assume that, for some $i \neq j \in \{1, \ldots, n\}$, the set $\text{int}(W_i) \cap \text{int}(V_j)$ has strictly positive measure. For all $q \in \text{int}(W_i) \cap \text{int}(V_j)$, we know $\|q - p_i\|_2 > \|q - p_j\|_2$. Because f is non-increasing, $f(\|q - p_i\|_2) < f(\|q - p_j\|_2)$ and, since $\text{int}(W_i) \cap \text{int}(V_j)$ has strictly positive measure,

$$\int_{\text{int}(W_i) \cap \text{int}(V_j)} f(\|q - p_i\|_2)\phi(q)dq < \int_{\text{int}(W_i) \cap \text{int}(V_j)} f(\|q - p_j\|_2)\phi(q)dq.$$

Therefore, we deduce

$$\int_{W_i} f(\|q - p_i\|_2)\phi(q)dq < \sum_{j=1}^{n} \int_{W_i \cap V_j} f(\|q - p_j\|_2)\phi(q)dq,$$

and the statement follows. ■

Different choices of performance function give rise to different expected-value multicenter functions with particular features. Let us examine some important cases.

Distortion problem: Consider as performance function $f(x) = -x^2$. Then, on $S \setminus \mathcal{S}_{\text{coinc}}$, the expected-value multicenter function takes the form

$$\mathcal{H}_{\text{distor}}(p_1, \ldots, p_n) = -\sum_{i=1}^{n} \int_{V_i(P)} \|q - p_i\|_2^2 \phi(q)dq = -\sum_{i=1}^{n} \text{J}_\phi(V_i(\mathcal{P}), p_i),$$

where recall that $\mathrm{J}_\phi(W,p)$ denotes the polar moment of inertia of the set W about the point p. In signal compression $-\mathcal{H}_{\mathrm{distor}}$ is referred to as the *distortion function* and is relevant in many disciplines including vector quantization, signal compression, and numerical integration, see [Gray and Neuhoff, 1998, Du et al., 1999]. Here, distortion refers to the average deformation (weighted by the density ϕ) caused by reproducing $q \in S$ with the location p_i in $\mathcal{P} = \{p_1, \ldots, p_n\}$ such that $q \in V_i(\mathcal{P})$. It is interesting to note that

$$\begin{aligned}\mathcal{H}_{\mathrm{distor}}(p_1,\ldots,p_n,W_1,\ldots,W_n) &= -\sum_{i=1}^n \mathrm{J}_\phi(W_i,p_i)\\ &= -\sum_{i=1}^n \mathrm{J}_\phi(W_i,\mathrm{CM}_\phi(W_i)) - \sum_{i=1}^n \mathrm{A}_\phi(W_i)\|p_i-\mathrm{CM}_\phi(W_i)\|_2^2, \quad (2.7)\end{aligned}$$

where in the last equality we have used the Parallel Axis Theorem [Hibbeler, 2003]. Note that the first term only depends on the partition of S, whereas the second term also depends on the location of the points. The following result is a consequence of this observation.

Proposition 2.15 ($\mathcal{H}_{\mathrm{distor}}$-optimality of centroid locations). *Let $\{W_1,\ldots,W_n\} \subset \mathbb{P}(S)$ be a partition of S. Then, for any $\mathcal{P} = \{p_1,\ldots,p_n\} \in \mathbb{F}(S)$,*

$$\begin{aligned}\mathcal{H}_{\mathrm{distor}}\big(\mathrm{CM}_\phi(W_1),\ldots,\mathrm{CM}_\phi(W_n),W_1,\ldots,W_n\big)&\\ \geq \mathcal{H}_{\mathrm{distor}}(p_1,\ldots,p_n,W_1,\ldots,W_n),&\end{aligned}$$

that is, the centroid locations $\mathrm{CM}_\phi(W_1),\ldots,\mathrm{CM}_\phi(W_n)$ are optimal for $\mathcal{H}_{\mathrm{distor}}$ among all configurations in S. Moreover, if all the sets $\{W_1,\ldots,W_n\}$ have non-vanishing areas, the inequality is strict unless $p_i = \mathrm{CM}_\phi(W_i)$ for all $i \in \{1,\ldots,n\}$.

A consequence of this result is that for the 1-center problem, i.e., when $n = 1$, the node location that optimizes $p \mapsto \mathcal{H}_{\mathrm{distor}}(p) = -\mathrm{J}_\phi(S,p)$ is the centroid of the set S, denoted by $\mathrm{CM}_\phi(S)$;

Area problem: Consider as performance function $f(x) = 1_{[0,a]}(x)$, $a \in \mathbb{R}_{>0}$ the indicator function of the closed interval $[0,a]$. Then, the expected-value multicenter function takes the form

$$\begin{aligned}\mathcal{H}_{\mathrm{area},a}(p_1,\ldots,p_n) &= \sum_{i=1}^n \int_{V_i(\mathcal{P})} 1_{[0,a]}(\|q-p_i\|_2)\phi(q)dq\\ &= \sum_{i=1}^n \int_{V_i(\mathcal{P})\cap\overline{B}(p_i,a)} \phi(q)dq\\ &= \sum_{i=1}^n \mathrm{A}_\phi(V_i(\mathcal{P})\cap\overline{B}(p_i,a)) = \mathrm{A}_\phi(\cup_{i=1}^n \overline{B}(p_i,a)),\end{aligned}$$

that is, it corresponds to the area, measured according to ϕ, covered by the union of the n balls $\overline{B}(p_1, a), \ldots, \overline{B}(p_n, a)$. Exercise E2.4 discusses the 1-center area problem;

Mixed distortion-area problem: Consider as performance function $f(x) = -x^2\ 1_{[0,a]}(x) + b \cdot 1_{]a,+\infty[}(x)$, with $a \in \mathbb{R}_{>0}$ and $b \leq -a^2$. Then, on $S \backslash \mathcal{S}_{\text{coinc}}$, the expected-value multicenter function takes the form

$$\mathcal{H}_{\text{distor-area},a,b}(p_1, \ldots, p_n) = -\sum_{i=1}^{n} \mathrm{J}_\phi(V_{i,a}(\mathcal{P}), p_i) + b\, \mathrm{A}_\phi(Q \setminus \cup_{i=1}^{n} \overline{B}(p_i, a)),$$

that is, it is a combination of the multicenter functions corresponding to the distortion problem and the area problem. Of special interest to us is the multicenter function that results from the choice $b = -a^2$. In this case, the performance function f is continuous, and we simply write $\mathcal{H}_{\text{distor-area},a}$. The extension of this function to sets of points and partitions of the space reads as

$$\begin{aligned}\mathcal{H}_{\text{distor-area},a}&(p_1, \ldots, p_n, W_1, \ldots, W_n) \\ &= -\sum_{i=1}^{n} \Big(\mathrm{J}_\phi(W_i \cap \overline{B}(p_i, a), p_i) + a^2\, \mathrm{A}_\phi(W_i \cap (S \setminus \overline{B}(p_i, a))) \Big).\end{aligned}$$

The following optimality result can be established (see Exercise E2.7).

Proposition 2.16 ($\mathcal{H}_{\text{distor-area},a}$-optimality of centroid locations). *Let $\{W_1, \ldots, W_n\} \subset \mathbb{P}(S)$ be a partition of S. Then, for any $\mathcal{P} = \{p_1, \ldots, p_n\} \in \mathbb{F}(S)$,*

$$\begin{aligned}\mathcal{H}_{\text{distor-area},a}\big(\mathrm{CM}_\phi(W_1 \cap \overline{B}(p_1, a)), \ldots, \mathrm{CM}_\phi(W_n \cap \overline{B}(p_n, a)), W_1, \ldots, W_n\big) \\ \geq \mathcal{H}_{\text{distor}}(p_1, \ldots, p_n, W_1, \ldots, W_n).\end{aligned}$$

Moreover, if all the sets $\{W_1, \ldots, W_n\}$ have non-vanishing areas, the inequality is strict unless $p_i = \mathrm{CM}_\phi(W_i \cap \overline{B}(p_i, a))$, for all $i \in \{1, \ldots, n\}$.

A consequence of this result is that for the 1-center problem, i.e., when $n = 1$, the node location that optimizes $p \mapsto \mathcal{H}_{\text{distor-area},a}(p) = \mathrm{J}_\phi(S \cap \overline{B}(p, a), p) + a^2\, \mathrm{A}_\phi(S \backslash \overline{B}(p, a))$ is the centroid of the set $S \cap \overline{B}(p, a)$, denoted by $\mathrm{CM}_\phi(S \cap \overline{B}(p, a))$.

Next, we characterize the smoothness of the expected-value multicenter function. Before stating the precise result, let us introduce some useful notation. For a performance function f, let $\mathrm{Dscn}(f)$ denote the (finite) set of points where f is discontinuous. For each $a \in \mathrm{Dscn}(f)$, define the limiting values from the left and from the right, respectively, as

$$f_-(a) = \lim_{x \to a^-} f(x), \qquad f_+(a) = \lim_{x \to a^+} f(x).$$

We are now ready to characterize the smoothness of $\mathcal{H}_{\text{exp}}$, see [Cortés et al., 2005]. Before stating the result, recall that the line integral of a function $g : \mathbb{R}^2 \to \mathbb{R}$ over a curve C parameterized by a continuous and piecewise continuously differentiable map $\gamma : [0, 1] \to \mathbb{R}^2$ is defined by

$$\int_C g = \int_C g(\gamma) d\gamma := \int_0^1 g(\gamma(t)) \, \|\dot{\gamma}(t)\|_2 \, dt,$$

and is independent of the selected parameterization.

Theorem 2.17 (Smoothness properties of $\mathcal{H}_{\text{exp}}$). *Given $S \subset \mathbb{R}^d$ bounded and measurable, a density $\phi : \mathbb{R} \to \mathbb{R}_{\geq 0}$ and a performance function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}$, the expected-value multicenter function $\mathcal{H}_{\text{exp}} : S^n \to \mathbb{R}$ is*

(i) globally Lipschitz on S^n, and
(ii) continuously differentiable on $S^n \setminus \mathcal{S}_{\text{coinc}}$, where for each $i \in \{1, \dots, n\}$

$$\frac{\partial \mathcal{H}_{\text{exp}}}{\partial p_i}(P) = \int_{V_i(\mathcal{P})} \frac{\partial}{\partial p_i} f(\|q - p_i\|_2) \phi(q) dq + \sum_{a \in \text{Dscn}(f)} \big(f_-(a) - f_+(a)\big) \int_{V_i(\mathcal{P}) \cap \partial \overline{B}(p_i, a)} \text{n}_{\text{out}, \overline{B}(p_i, a)}(q) \phi(q) dq. \tag{2.8}$$

Therefore, the gradient of $\mathcal{H}_{\text{exp}}$, interpreted as a map from S^n to $\mathbb{R}^n$, is spatially distributed (in the sense defined in Section 2.2.3) over the Delaunay graph $\mathcal{G}_{\text{D}}$.

Let us discuss how Theorem 2.17 particularizes to the distortion, area, and mixed distortion-area problems.

Distortion problem: In this case, the performance function does not have any discontinuities and, therefore, the second term in (2.8) vanishes. The gradient of $\mathcal{H}_{\text{distor}}$ on $S^n \setminus \mathcal{S}_{\text{coinc}}$ then takes the form, for each $i \in \{1, \dots, n\}$,

$$\frac{\partial \mathcal{H}_{\text{distor}}}{\partial p_i}(P) = 2 \, \text{A}_\phi(V_i(\mathcal{P}))(\text{CM}_\phi(V_i(\mathcal{P})) - p_i),$$

that is, the ith component of the gradient points in the direction of the vector going from p_i to the centroid of its Voronoi cell. The critical points of $\mathcal{H}_{\text{distor}}$ are therefore the set of centroidal Voronoi configurations in S (cf. Section 2.1.4). This is a natural generalization of the result for the 1-center case, where the optimal node location is the centroid $\text{CM}_\phi(S)$;

Area problem: In this case, the performance function is differentiable everywhere except at a single discontinuity, and its derivative is identically zero. Therefore the first term in (2.8) vanishes. The gradient of $\mathcal{H}_{\text{area},a}$ on $S^n \setminus \mathcal{S}_{\text{coinc}}$ then takes the form, for each $i \in \{1, \dots, n\}$,

$$\frac{\partial \mathcal{H}_{\text{area},a}}{\partial p_i}(P) = \int_{V_i(\mathcal{P}) \cap \partial \overline{B}(p_i,a)} \mathrm{n}_{\text{out},\overline{B}(p_i,a)}(q)\phi(q)dq,$$

that is, the ith component of the gradient is an average of the normal at each point of $V_i(\mathcal{P}) \cap \partial\overline{B}(p_i, a)$, see Figure 2.12 for an illustration. The

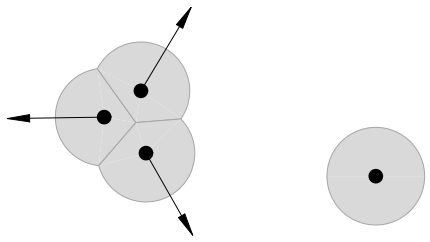


Fig. 2.12. Gradient of the area function in an environment with constant density function. The component of the gradient corresponding to the rightmost node is zero – roughly speaking, because there is no incentive for this node to move in any particular direction. However, the component of the gradient for each of the three leftmost nodes is non-zero – roughly speaking, because by moving in the direction of the gradient, these agents decrease the overlapping among the disk and cover new regions of the space.

critical points of $\mathcal{H}_{\text{area},a}$ correspond to configurations with the property that each p_i is a local maximum for the area of $V_{i,a}(P) = V_i(P) \cap \overline{B}(p_i, a)$ at fixed $V_i(P)$. We refer to these configurations as *a-limited area-centered* Voronoi configurations. This is a natural generalization of the result for the 1-center case, where the optimal node location maximizes $\mathrm{A}_\phi(S \cap \overline{B}(p, a))$ (cf. Exercise E2.4);

Mixed distortion-area problem: In this case, the gradient of $\mathcal{H}_{\text{distor-area},a,b}$ is a combination of the gradients of $\mathcal{H}_{\text{distor}}$ and $\mathcal{H}_{\text{area},a}$. Specifically, one has for each $i \in \{1, \dots, n\}$,

$$\begin{aligned}\frac{\partial \mathcal{H}_{\text{distor-area},a,b}}{\partial p_i}(P) = {} & 2\,\mathrm{A}_\phi(V_{i,a}(\mathcal{P}))(\mathrm{CM}_\phi(V_{i,a}(\mathcal{P})) - p_i) \\ & - (a^2 + b) \int_{V_i(\mathcal{P}) \cap \partial \overline{B}(p_i,a)} \mathrm{n}_{\text{out},\overline{B}(p_i,a)}(q)\phi(q)dq.\end{aligned}$$

For the particular case when $b = -a^2$, the performance function is continuous, and the gradient of $\mathcal{H}_{\text{distor-area},a}$ takes the simpler form

$$\frac{\partial \mathcal{H}_{\text{distor-area},a}}{\partial p_i}(P) = 2\,\mathrm{A}_\phi(V_{i,a}(\mathcal{P}))(\mathrm{CM}_\phi(V_{i,a}(\mathcal{P})) - p_i),$$

which points in the direction of the vector from p_i to the centroid of its a-limited Voronoi cell. In this case, the critical points of $\mathcal{H}_{\text{distor-area},a}$ are

therefore the set of a-limited centroidal Voronoi configurations in S (cf. Section 2.1.4). This is a natural generalization of the result for the 1-center case, where the optimal node location is the centroid $\mathrm{CM}_\phi(S \cap \overline{B}(p, a))$.

We refer to $\mathcal{H}_{\text{distor}}$, $\mathcal{H}_{\text{area},a}$, and $\mathcal{H}_{\text{distor-area},a}$ as multicenter functions because, as the above discussion shows, their critical points correspond to various notions of center Voronoi configurations.

It is important to note that the gradients of $\mathcal{H}_{\text{area},a}$ and $\mathcal{H}_{\text{distor-area},a,b}$ are spatially distributed over the $2a$-limited Delaunay graph $\mathcal{G}_{\text{LD}}(2a)$. This observation is important for practical considerations: robotic agents with limited-range interactions cannot in general compute the gradient of $\mathcal{H}_{\text{distor}}$ because, as we noted in Remark 2.11, for a given $r \in \mathbb{R}_{>0}$, $\mathcal{G}_{\text{D}}$ is not in general spatially distributed over $\mathcal{G}_{\text{disk}}(r)$. However, robotic agents with limited-range interactions can compute the gradients of $\mathcal{H}_{\text{area},a}$ and $\mathcal{H}_{\text{distor-area},a,b}$ as long as $r \geq 2a$ because, from Theorem 2.7(iii), $\mathcal{G}_{\text{LD}}(r)$ is spatially distributed over $\mathcal{G}_{\text{disk}}(r)$. The relevance of this fact is further justified by the following result.

Proposition 2.18 (Constant-factor approximation of $\mathcal{H}_{\text{distor}}$). *Let $S \subset \mathbb{R}^d$ be bounded and measurable. Consider the mixed distortion-area problem with $a \in]0, \operatorname{diam} S]$ and $b = -\operatorname{diam}(S)^2$. Then, for all $P \in S^n$,*

$$\mathcal{H}_{\text{distor-area},a,b}(P) \leq \mathcal{H}_{\text{distor}}(P) \leq \beta^2\, \mathcal{H}_{\text{distor-area},a,b}(P) < 0, \tag{2.9}$$

where $\beta = \frac{a}{\operatorname{diam}(S)} \in [0, 1]$.

In fact, similar constant-factor approximations of the expected-value multicenter function $\mathcal{H}_{\text{exp}}$ can also be established, see [Cortés et al., 2005].

2.3.2 Worst-case and disk-covering multicenter functions

Given a compact set $S \subset \mathbb{R}^d$ and a performance function f, let us consider the point in S that is worst covered by a set of points $p_1, \ldots, p_n$. Formally, we define the *worst-case multicenter* function $\mathcal{H}_{\text{worst}} : S^n \to \mathbb{R}$ by

$$\mathcal{H}_{\text{worst}}(p_1, \ldots, p_n) = \min_{q \in S} \max_{i \in \{1,\ldots,n\}} f(\|q - p_i\|_2). \tag{2.10}$$

The definition of $\mathcal{H}_{\text{worst}}$ can be read as follows: for each location $q \in S$, consider the best coverage of q among those provided by each of the nodes $p_1, \ldots, p_n$, which corresponds to the value $\max_{i \in \{1,\ldots,n\}} f(\|q - p_i\|_2)$. Then, compute the worst coverage $\mathcal{H}_{\text{worst}}(p_1, \ldots, p_n)$ by comparing the performance at all locations in S.

Given the interpretation of $\mathcal{H}_{\text{worst}}$, we seek to solve the following geometric optimization problem

$$\text{maximize } \mathcal{H}_{\text{worst}}(p_1, \ldots, p_n), \tag{2.11}$$

that is, we seek to determine configurations $p_1, \ldots, p_n$ that maximize the value of $\mathcal{H}_{\text{worst}}$. An equivalent formulation of this problem is referred to as a *continuous p-center problem* in the literature on facility location, see e.g., [Drezner, 1995].

In the present context, also relevant is the *disk-covering multicenter* function $\mathcal{H}_{\text{dc}} : S^n \to \mathbb{R}$, defined by

$$\mathcal{H}_{\text{dc}}(p_1, \ldots, p_n) = \max_{q \in S} \min_{i \in \{1,\ldots,n\}} \|q - p_i\|_2. \tag{2.12}$$

The value of $\mathcal{H}_{\text{dc}}$ can be interpreted as the largest possible distance from a point in S to one of the locations $p_1, \ldots, p_n$. Note that, by definition, the environment S is contained in the union of n closed balls centered at $p_1, \ldots, p_n$ with radius $\mathcal{H}_{\text{dc}}(p_1, \ldots, p_n)$.

The following result establishes the relationship between the worst-case and the disk-covering multicenter functions, and as byproduct, provides an elegant reformulation of the geometric optimization problem (2.11). Its proof is left to the reader.

Lemma 2.19 (Relationship between $\mathcal{H}_{\text{worst}}$ and $\mathcal{H}_{\text{dc}}$). *Given $S \subset \mathbb{R}^d$ compact and a performance function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}$, one has $\mathcal{H}_{\text{worst}} = f \circ \mathcal{H}_{\text{dc}}$.*

Using Lemma 2.19 and the fact that f is non-increasing, we can reformulate the geometric optimization problem (2.11) as

$$\text{minimize } \mathcal{H}_{\text{dc}}(p_1, \ldots, p_n), \tag{2.13}$$

that is, find the minimum radius r such that the environment S is covered by n closed balls center at $p_1, \ldots, p_n$ with equal radius r. Note the connection between this formulation and the classical disk-covering problem: how to cover a region with (possibly overlapping) disks of minimum radius. We shall comment more on this connection later.

Given the equivalence between the geometric optimization problems (2.11) and (2.13), we focus our attention on $\mathcal{H}_{\text{dc}}$. The disk-covering multicenter function can be alternatively described in terms of the Voronoi partition of S generated by $\mathcal{P} = \{p_1, \ldots, p_n\}$. For $(p_1, \ldots, p_n) \in S^n \setminus \mathcal{S}_{\text{coinc}}$, one has

$$\begin{aligned} \mathcal{H}_{\text{dc}}(p_1, \ldots, p_n) &= \max_{i \in \{1,\ldots,n\}} \max_{q \in V_i(\mathcal{P})} \|q - p_i\|_2 \\ &= \max_{i \in \{1,\ldots,n\}} \max_{q \in \partial V_i(\mathcal{P})} \|q - p_i\|_2. \end{aligned} \tag{2.14}$$

This expression of $\mathcal{H}_{\text{dc}}$ is appealing because it clearly shows the value of the function as the result of the aggregate contribution of all individual nodes. If $(p_1, \ldots, p_n) \in \mathcal{S}_{\text{coinc}}$, then a similar decomposition of $\mathcal{H}_{\text{dc}}$ can be written in terms of the distinct points $\mathcal{P} = i_{\mathbb{F}}(p_1, \ldots, p_n)$.

Inspired by the expression (2.14), let us define a more general version of the worst-case multicenter function. Given $(p_1, \ldots, p_n) \in S^n$ and a partition $\{W_1, \ldots, W_n\} \subset \mathbb{P}(S)$ of S, let

$$\mathcal{H}_{\mathrm{dc}}(p_1,\dots,p_n,W_1,\dots,W_n) = \max_{i\in\{1,\dots,n\}} \max_{q\in\partial W_i} \|q-p_i\|_2.$$

Notice that $\mathcal{H}_{\mathrm{dc}}(p_1,\dots,p_n) = \mathcal{H}_{\mathrm{dc}}(p_1,\dots,p_n,V_1(\mathcal{P}),\dots,V_n(\mathcal{P}))$, for all $(p_1,\dots,p_n) \in S^n \setminus \mathcal{S}_{\mathrm{coinc}}$. Moreover, one can establish the following optimality result.

Proposition 2.20 ($\mathcal{H}_{\mathrm{dc}}$-optimality of the Voronoi partition and circumcenter locations). *For any $\mathcal{P} = \{p_1,\dots,p_n\} \in \mathbb{F}(S)$ and any partition $\{W_1,\dots,W_n\} \subset \mathbb{P}(S)$ of S,*

$$\mathcal{H}_{\mathrm{dc}}(p_1,\dots,p_n,V_1(\mathcal{P}),\dots,V_n(\mathcal{P})) \leq \mathcal{H}_{\mathrm{dc}}(p_1,\dots,p_n,W_1,\dots,W_n),$$

that is, the Voronoi partition $\mathcal{V}(\mathcal{P})$ is optimal for $\mathcal{H}_{\mathrm{dc}}$ among all partitions of S, and

$$\mathcal{H}_{\mathrm{dc}}(\mathrm{CC}(W_1),\dots,\mathrm{CC}(W_n),W_1,\dots,W_n) \leq \mathcal{H}_{\mathrm{dc}}(p_1,\dots,p_n,W_1,\dots,W_n),$$

that is, the circumcenter locations $\mathrm{CC}(W_1),\dots,\mathrm{CC}(W_n)$ are optimal for $\mathcal{H}_{\mathrm{dc}}$ among all configurations in S.

As a corollary of this result, we have that the circumcenter of S is a global optimum of $\mathcal{H}_{\mathrm{dc}}$ for the 1-center problem, i.e., when $n = 1$. This comes at no surprise since, in this case, the value $\mathcal{H}_{\mathrm{dc}}(p)$ corresponds to the radius of the minimum-radius sphere centered at p that encloses S.

The following result characterizes the smoothness properties of the disk-covering multicenter function, see [Cortés and Bullo, 2005].

Theorem 2.21 (Smoothness properties of $\mathcal{H}_{\mathrm{dc}}$). *Given $S \subset \mathbb{R}^d$ compact, the disk-covering multicenter function $\mathcal{H}_{\mathrm{dc}} : S^n \to \mathbb{R}$ is globally Lipschitz on S^n.*

The generalized gradient and the critical points of $\mathcal{H}_{\mathrm{sp}}$ can be characterized, but require a careful study based on nonsmooth analysis [Clarke, 1983]. In particular, two facts taken from [Cortés and Bullo, 2005] are of interest here. First, under certain technical conditions, one can show that the critical points of $\mathcal{H}_{\mathrm{dc}}$ are circumcenter Voronoi configurations. This is why we refer to $\mathcal{H}_{\mathrm{dc}}$ as a multicenter function. Second, the generalized gradient of $\mathcal{H}_{\mathrm{dc}}$ is not spatially distributed over $\mathcal{G}_{\mathrm{D}}$. This is essentially due to the inherent comparison among all agents that is embedded in the definition of $\mathcal{H}_{\mathrm{dc}}$ (via the max function).

2.3.3 Sphere-packing multicenter functions

Given a compact, connected set $S \subset \mathbb{R}^d$, consider the situation where one seeks to cover S as much as possible by means of open balls centered at a set of points $p_1,\dots,p_n$ with equal radius in such a way that these regions do not intersect or leave the environment. Formally, we define the *sphere-packing multicenter* function $\mathcal{H}_{\mathrm{sp}} : S^n \to \mathbb{R}$ by

$$\mathcal{H}_{\rm sp}(p_1,\dots,p_n) = \min_{i\neq j\in\{1,\dots,n\}} \Big\{ \frac{1}{2}\|p_i - p_j\|_2, \operatorname{dist}(p_i, \partial S) \Big\}. \tag{2.15}$$

The definition of $\mathcal{H}_{\rm sp}$ can be read as follows: consider the pairwise distances between any two points p_i, p_j (multiplied by a factor $1/2$ so that each point can fit a ball of equal radius and these balls do not intersect), and the individual distances from each point to the boundary of the environment. The value of $\mathcal{H}_{\rm sp}$ is then the smallest of all distances, guaranteeing that the union of n open balls centered at $p_1,\dots,p_n$ with radius $\mathcal{H}_{\rm sp}(p_1,\dots,p_n)$ is disjoint and contained in S.

Given the interpretation of $\mathcal{H}_{\rm sp}$, we seek to solve the following geometric optimization problem

$$\text{maximize } \mathcal{H}_{\rm sp}(p_1,\dots,p_n), \tag{2.16}$$

that is, we seek to determine configurations $p_1,\dots,p_n$ that maximize the value of $\mathcal{H}_{\rm sp}$. Note the connection of this formulation with the classical sphere-packing problem: how to maximize the coverage of a region with non-overlapping disks (contained in the region) of maximum radius. We shall comment more on this connection later.

The sphere-packing multicenter function can be alternatively described in terms of the Voronoi partition of S generated by $\mathcal{P} = \{p_1,\dots,p_n\}$. For $(p_1,\dots,p_n) \in S^n \setminus \mathcal{S}_{\rm coinc}$, one has

$$\mathcal{H}_{\rm sp}(p_1,\dots,p_n) = \min_{i\in\{1,\dots,n\}} \min_{q\in\partial V_i(\mathcal{P})} \|q - p_i\|_2. \tag{2.17}$$

As for the previous multicenter functions, this expression of $\mathcal{H}_{\rm sp}$ is appealing because it clearly shows the value of the function as the result of the aggregate contribution of all individual nodes. If $(p_1,\dots,p_n) \in \mathcal{S}_{\rm coinc}$, then a similar decomposition of $\mathcal{H}_{\rm sp}$ exists in terms of the distinct points $\mathcal{P} = i_{\mathbb{F}}(p_1,\dots,p_n)$.

Inspired by the expression (2.17), let us define a more general version of the worst-case multicenter function. Given $(p_1,\dots,p_n) \in S^n$ and a partition $\{W_1,\dots,W_n\} \subset \mathbb{P}(S)$ of S, let

$$\mathcal{H}_{\rm sp}(p_1,\dots,p_n,W_1,\dots,W_n) = \min_{i\in\{1,\dots,n\}} \min_{q\in\partial W_i} \|q - p_i\|_2.$$

Notice that $\mathcal{H}_{\rm sp}(p_1,\dots,p_n) = \mathcal{H}_{\rm sp}(p_1,\dots,p_n,V_1(\mathcal{P}),\dots,V_n(\mathcal{P}))$, for all $(p_1,\dots,p_n) \in S^n \setminus \mathcal{S}_{\rm coinc}$. Also note that the value $\mathcal{H}_{\rm sp}(q_1,\dots,q_n,W_1,\dots,W_n)$ is the same for any $q_i \in \mathrm{IC}(W_i)$, $i \in \{1,\dots,n\}$. With a slight abuse of notation, we refer to this common value as $\mathcal{H}_{\rm sp}(\mathrm{IC}(W_1),\dots,\mathrm{IC}(W_n),W_1,\dots,W_n)$. Moreover, one can establish the following optimality result.

Proposition 2.22 ($\mathcal{H}_{\rm sp}$-optimality of the Voronoi partition and in-center locations). *For any $\mathcal{P} = \{p_1,\dots,p_n\} \in \mathbb{F}(S)$ and any partition $\{W_1,\dots,W_n\} \subset \mathbb{P}(S)$ of S,*

$$\mathcal{H}_{\rm sp}(p_1,\dots,p_n,V_1(\mathcal{P}),\dots,V_n(\mathcal{P})) \geq \mathcal{H}_{\rm sp}(p_1,\dots,p_n,W_1,\dots,W_n),$$

that is, the Voronoi partition $\mathcal{V}(\mathcal{P})$ is optimal for $\mathcal{H}_{\text{sp}}$ among all partitions of S, and

$$\mathcal{H}_{\text{sp}}(\text{IC}(W_1), \ldots, \text{IC}(W_n), W_1, \ldots, W_n) \geq \mathcal{H}_{\text{sp}}(p_1, \ldots, p_n, W_1, \ldots, W_n),$$

that is, the incenter locations $\text{IC}(W_1), \ldots, \text{IC}(W_n)$ are optimal for $\mathcal{H}_{\text{sp}}$ among all configurations in S.

As a corollary of this result, we have that the incenter set of S is composed of global optima of $\mathcal{H}_{\text{sp}}$ for the 1-center problem, i.e., when $n = 1$. This comes at no surprise since, in this case, the value $\mathcal{H}_{\text{sp}}(p)$ corresponds to the radius of the maximum-radius sphere centered at p enclosed in S.

The following result characterizes the smoothness properties of the sphere-packing multicenter function, see [Cortés and Bullo, 2005].

Theorem 2.23 (Smoothness properties of $\mathcal{H}_{\text{sp}}$). *Given $S \subset \mathbb{R}^d$ compact, the sphere-packing multicenter function $\mathcal{H}_{\text{sp}} : S^n \to \mathbb{R}$ is globally Lipschitz on S^n.*

We conclude the section with some remark that are analogous to the ones for the function $\mathcal{H}_{\text{dc}}$. The generalized gradient and the critical points of $\mathcal{H}_{\text{sp}}$ can be characterized, but require a careful study based on nonsmooth analysis [Clarke, 1983]. In particular, two facts taken from [Cortés and Bullo, 2005] are of interest here. First, under certain technical conditions, one can show that the critical points of $\mathcal{H}_{\text{sp}}$ are incenter Voronoi configurations. This is why we refer to $\mathcal{H}_{\text{sp}}$ as a multicenter function. Second, the generalized gradient of $\mathcal{H}_{\text{sp}}$ is not spatially distributed over $\mathcal{G}_{\text{D}}$. This is essentially due to the inherent comparison among all agents that is embedded in the definition of $\mathcal{H}_{\text{sp}}$ (via the min function).

2.4 Notes

A thorough introduction to computational geometric concepts can be found in [Preparata and Shamos, 1993, de Berg et al., 2000, O'Rourke, 2000]. The handbooks [Goodman and O'Rourke, 2004, Sack and Urrutia, 2000] present a comprehensive overview of computational geometric problems and their applications. Among the numerous topics we do not discuss in this chapter, we mention distance geometry and rigidity theory [Whiteley, 1997] that are notable for their applications to network localization and formation control.

The notion of Voronoi partition, and generalizations of it, have been applied in a numerous areas, including spatial interpolation, pattern analysis, spatial processes modeling, and optimization, to name a few. The survey [Aurenhammer, 1991] and the book [Okabe et al., 2000] discuss the history, properties, and applications of Voronoi partitions. The nearest neighbor and natural neighbor interpolations based on Voronoi partitions, e.g., see [Sibson,

1981, Boissonnat and Cazals, 2002], are of particular interest to the treatment of this chapter because of their spatially-distributed computation character. Spatially-distributed maps for motion coordination are discussed in [Martínez et al., 2007c] and adopted in later chapters.

Proximity graphs [Jaromczyk and Toussaint, 1992] are a powerful tool to capture the structure and shape of geometric objects, and therefore have applications in multiple areas, including topology control of wireless networks [Santi, 2005], computer graphics [Langetepe and Zachmann, 2006], and geographic analysis [Radke, 1988]. In cooperative control, a closely related notion is that of state-dependent graph [Mesbahi, 2005]. Random geometric graphs [Penrose, 2003] and percolation theory [Bollobás and Riordan, 2006, Meester and Roy, 2008] study the properties of proximity graphs associated to the random deployment of points according to some specified density function.

Locational optimization problems [Drezner, 1995, Drezner and Hamacher, 2001] are spatial resource-allocation problems (e.g., where to place mailboxes in a city, where to place cache serves on the internet) that pervade a broad spectrum of scientific disciplines. Computational geometry plays an important role in locational optimization [Robert and Toussaint, 1990, Okabe et al., 2000]. The field of geometric optimization [Mitchell, 1997, Agarwal and Sharir, 1998, Boltyanski et al., 1999] blends the geometric and locational optimization aspects to study a wide variety of optimization problems induced by geometric objects.

2.5 Proofs

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

2.5.1 Proof of Theorem 2.7

Proof. The inclusions in fact (i) are taken from Jaromczyk and Toussaint [1992], de Berg et al. [2000].

The proof of the first inclusion in fact (ii) is as follows. Let $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\mathrm{G}} \cap \mathcal{G}_{\mathrm{disk}}(r)}(\mathcal{P})$. From the definition of the Gabriel graph, we deduce that $\|\frac{p_i+p_j}{2} - p_i\|_2 = \|\frac{p_i+p_j}{2} - p_j\|_2 \leq \|\frac{p_i+p_j}{2} - p_k\|_2$, for all $k \in \{1, \dots, n\} \setminus \{i, j\}$, and therefore, $\frac{p_i+p_j}{2} \in V_i(\mathcal{P}) \cap V_j(\mathcal{P})$. Since $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\mathrm{disk}}(r)}(\mathcal{P})$, we deduce that $\frac{p_i+p_j}{2} \in \overline{B}(p_i, \frac{r}{2}) \cap \overline{B}(p_j, \frac{r}{2})$, and hence $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\mathrm{LD}}(r)}(\mathcal{P})$. The second inclusion in (ii) is straightforward: if $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\mathrm{LD}}(r)}(\mathcal{P})$, then $V_i(\mathcal{P}) \cap V_j(\mathcal{P}) \neq \emptyset$, i.e., $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\mathrm{D}}}(\mathcal{P})$. Since clearly $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\mathrm{disk}}(r)}(\mathcal{P})$, we conclude (ii). ■

2.5.2 Proof of Theorem 2.9

Proof. The proof of fact (i) is as follows. Let $\mathcal{P} \in \mathbb{F}(\mathbb{R}^d)$. If $\mathcal{G}_{\mathrm{EMST}}(\mathcal{P}) \subseteq \mathcal{G}_{\mathrm{disk}}(r)(\mathcal{P})$, then clearly $\mathcal{G}_{\mathrm{disk}}(r)(\mathcal{P})$ is connected. To prove the other implication, we reason by contradiction. Assume $\mathcal{G}_{\mathrm{disk}}(r)(\mathcal{P})$ is connected and let

$\mathcal{G}_{\text{EMST}}(\mathcal{P}) \not\subseteq \mathcal{G}_{\text{disk}}(r)(\mathcal{P})$, i.e., there exists p_i and p_j with $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{EMST}}}(\mathcal{P})$ and $\|p_i - p_j\|_2 > r$. If we remove this edge from $\mathcal{E}_{\mathcal{G}_{\text{EMST}}}(\mathcal{P})$, then the tree becomes disconnected into two connected components T_1 and T_2, with $p_i \in T_1$ and $p_j \in T_2$. Now, since by hypothesis $\mathcal{G}_{\text{disk}}(r)(\mathcal{P})$ is connected, there must exist $k, l \in \{1, \dots, n\}$ such that $p_k \in T_1$, $p_l \in T_2$ and $\|p_k - p_l\|_2 \leq r$. If we add the edge (p_k, p_l) to the set of edges of $T_1 \cup T_2$, then the resulting graph G is acyclic, connected, and contains all the vertices $\mathcal{P}$, i.e., G is a spanning tree. Moreover, since $\|p_k - p_l\|_2 \leq r < \|p_i - p_j\|_2$ and T_1 and T_2 are induced subgraphs of $\mathcal{G}_{\text{EMST}}(\mathcal{P})$, we conclude that G has smaller length than $\mathcal{G}_{\text{EMST}}(\mathcal{P})$, which is a contradiction with the definition of the Euclidean minimum spanning tree.

Let us prove fact (ii). For $r \in \mathbb{R}_+$, it is enough to show that $\mathcal{G}_{\text{EMST}} \cap \mathcal{G}_{\text{disk}}(r)$ has the same connected components as $\mathcal{G}_{\text{disk}}(r)$, since this implies that the same result holds for $\mathcal{G}_{\text{RN}} \cap \mathcal{G}_{\text{disk}}(r)$, $\mathcal{G}_{\text{G}} \cap \mathcal{G}_{\text{disk}}(r)$, and $\mathcal{G}_{\text{LD}}(r)$. Since $\mathcal{G}_{\text{EMST}} \cap \mathcal{G}_{\text{disk}}(r)$ is a subgraph of $\mathcal{G}_{\text{disk}}(r)$, it is clear that vertices belonging to the same connected component of $\mathcal{G}_{\text{EMST}} \cap \mathcal{G}_{\text{disk}}(r)$ must also belong to the same connected component of $\mathcal{G}_{\text{disk}}(r)$. To prove the converse, let $\mathcal{P} \in \mathbb{F}(\mathbb{R}^d)$, and assume p_i and p_j in $\mathcal{P}$ verify $\|p_i - p_j\|_2 \leq r$. Let C be the connected component of $\mathcal{G}_{\text{disk}}(r)(\mathcal{P})$ to which they belong. With a slight abuse of notation, we also denote by C the vertices of the connected component. Since C is connected, then $\mathcal{G}_{\text{EMST}}(C) \subset C$ by fact (i). Moreover, since all the nodes in $\mathcal{P} \setminus C$ are at a distance strictly larger than r from any node of C, we deduce from the definition of the Euclidean minimum spanning tree that $\mathcal{G}_{\text{EMST}}(C)$ is equal to the subgraph of $\mathcal{G}_{\text{EMST}}(\mathcal{P})$ induced by C. Therefore, $\mathcal{G}_{\text{EMST}}(C) \subset \mathcal{G}_{\text{EMST}} \cap \mathcal{G}_{\text{disk}}(r)(\mathcal{P})$, and p_i and p_j belong to the same component of $\mathcal{G}_{\text{EMST}} \cap \mathcal{G}_{\text{disk}}(r)(\mathcal{P})$. This implies the result. ■

2.5.3 Proof of Proposition 2.10

Proof. Regarding the statement on $\mathcal{G}_{\text{RN}} \cap \mathcal{G}_{\text{disk}}(r)$, note that

$$B(p_i, \|p_i - p_j\|_2) \cap B(p_j, \|p_i - p_j\|_2) \subset B(p_i, \|p_i - p_j\|_2).$$

Therefore, if $\|p_i - p_j\|_2 \leq r$, then any node contained in the intersection $B(p_i, \|p_i - p_j\|_2) \cap B(p_j, \|p_i - p_j\|_2)$ must necessarily be within a distance r of p_i. From here, we deduce that $\mathcal{G}_{\text{RN}} \cap \mathcal{G}_{\text{disk}}(r)$ is spatially distributed over $\mathcal{G}_{\text{disk}}(r)$. Regarding the statement on $\mathcal{G}_{\text{G}} \cap \mathcal{G}_{\text{disk}}(r)$, note that

$$B\Big(\frac{p_i + p_j}{2}, \frac{\|p_i - p_j\|_2}{2}\Big) \subset B(p_i, \|p_i - p_j\|_2).$$

Therefore, if $\|p_i - p_j\|_2 \leq r$, then any node contained in $B\big(\frac{p_i+p_j}{2}, \frac{\|p_i-p_j\|_2}{2}\big)$ must necessarily be within a distance r of p_i. From here, we deduce that $\mathcal{G}_{\text{G}} \cap \mathcal{G}_{\text{disk}}(r)$ is spatially distributed over $\mathcal{G}_{\text{disk}}(r)$. Finally, note that if $\|p_i - p_j\|_2 > r$, then the half plane $\{q \in \mathbb{R}^2 \mid \|q - p_i\|_2 \leq \|q - p_j\|_2\}$ contains the ball $\overline{B}(p_i, \frac{r}{2})$. Accordingly,

$$
\begin{aligned}
V_{i,\frac{r}{2}}(\mathcal{P}) &= V_i(\mathcal{P}) \cap \overline{B}(p_i, \tfrac{r}{2}) \\
&= \{q \in \mathbb{R}^2 \mid \|q - p_i\|_2 \leq \|q - p_j\|_2, \text{ for all } p_j \in \mathcal{P}\} \cap \overline{B}(p_i, \tfrac{r}{2}) \\
&= \{q \in \mathbb{R}^2 \mid \|q - p_i\|_2 \leq \|q - p_j\|_2, \text{ for all } p_j \in \mathcal{N}_{\mathcal{G}_{\text{disk}}(r),p_i}(\mathcal{P})\} \cap \overline{B}(p_i, \tfrac{r}{2}),
\end{aligned}
$$

from where we deduce that $\mathcal{G}_{\text{LD}}(r)$ is spatially distributed over $\mathcal{G}_{\text{disk}}(r)$. ■

2.5.4 Proof of Theorem 2.17

We begin with some preliminary notions. In the following, a set $\Omega \subset \mathbb{R}^2$ is *piecewise smooth* if its boundary, $\partial\Omega$, is a not self-intersecting closed curve that admits a continuous and piecewise continuously differentiable parameterization $\gamma : [0,1] \to \mathbb{R}^2$. Likewise, a collection of sets $\{\Omega(x) \subset \mathbb{R}^2 \mid x \in (a,b)\}$ is a *piecewise smooth family* if $\Omega(x)$ is piecewise smooth for all $x \in (a,b)$, and there exists a continuous function $\gamma : [0,1] \times (a,b) \to \mathbb{R}^2$, $(t,x) \mapsto \gamma(t,x)$, continuously differentiable with respect to its second argument, such that for each $x \in (a,b)$, the map $t \mapsto \gamma_x(t) = \gamma(t,x)$ is a continuous and piecewise smooth parameterization of $\partial\Omega(x)$. We refer to γ as a *parameterization for the family* $\{\Omega(x) \subset \mathbb{R}^2 \mid x \in (a,b)\}$.

The following result is an extension of the integral form of the Conservation-of-Mass Law in fluid mechanics [Chorin and Marsden, 1994] and of the classic divergence theorem in differential geometry [Chavel, 1984].

Proposition 2.24 (Generalized conservation of mass). *Let $\{\Omega(x) \subset \mathbb{R}^2 \mid x \in (a,b)\}$ be a piecewise smooth family such that $\Omega(x)$ is star-shaped for all $x \in (a,b)$. Let the function $\phi : \mathbb{R}^2 \times (a,b) \to \mathbb{R}$ be continuous on $\mathbb{R}^2 \times (a,b)$, continuously differentiable with respect to its second argument for all $x \in (a,b)$ and almost all $q \in \Omega(x)$, and such that for each $x \in (a,b)$, the maps $q \mapsto \phi(q,x)$ and $q \mapsto \frac{\partial\phi}{\partial x}(q,x)$ are measurable, and integrable on $\Omega(x)$. Then, the function*

$$
(a,b) \ni x \mapsto \int_{\Omega(x)} \phi(q,x)dq \tag{2.18}
$$

is continuously differentiable and

$$
\frac{d}{dx}\int_{\Omega(x)} \phi(q,x)dq = \int_{\Omega(x)} \frac{\partial\phi}{\partial x}(q,x)dq + \int_{\partial\Omega(x)} \phi(\gamma,x)\Big(n(\gamma)\cdot\frac{\partial\gamma}{\partial x}\Big)d\gamma\,,
$$

where $n : \partial\Omega(x) \to \mathbb{R}^2$, $q \mapsto n(q)$, denotes the unit outward normal to $\partial\Omega(x)$ at $q \in \partial\Omega(x)$, and $\gamma : [0,1] \times (a,b) \to \mathbb{R}^2$ is a parameterization for the family $\{\Omega(x) \subset \mathbb{R}^2 \mid x \in (a,b)\}$.

We interpret the proposition as follows: in the fluid mechanics interpretation, as the parameter x changes, the total mass variation inside the region can be decomposed into two terms. The first term is the amount of mass created inside the region, whereas the second term is the amount of mass that crosses the moving boundary of the region.

Proof (Proposition 2.24). Let $x_0 \in (a,b)$. Using the fact that the map γ is continuous and that $\Omega(x_0)$ is star-shaped, one can show that there exist an interval around x_0 of the form $(x_0-\varepsilon, x_0+\varepsilon)$, a continuously differentiable function $u_{x_0} : [0,1] \times \mathbb{R}_{\geq 0} \to \mathbb{R}^2$ and a function $r_{x_0} : [0,1] \times (x_0-\varepsilon, x_0+\varepsilon) \to \mathbb{R}_{\geq 0}$ continuously differentiable in its second argument and piecewise continuously differentiable in its first argument, such that for all $x \in (x_0-\varepsilon, x_0+\varepsilon)$, one has

$$\begin{aligned} \Omega(x) &= \cup_{t\in[0,1]}\{u_{x_0}(t,s) \mid 0 \leq s \leq r_{x_0}(t,x)\}, \\ \gamma(t,x) &= u_{x_0}(t, r_{x_0}(t,x)), \quad \text{for all } t \in [0,1]. \end{aligned}$$

For simplicity, we denote by r and u the functions r_{x_0} and u_{x_0}, respectively. By definition, the function in (2.18) is continuously differentiable at x_0 if the following limit exists

$$\lim_{h\to 0} \frac{1}{h}\Big(\int_{\Omega(x_0+h)} \phi(q, x_0+h)dq - \int_{\Omega(x_0)} \phi(q,x_0)dq \Big),$$

and depends continuously on x_0. Now, we can rewrite the previous limit as

$$\begin{aligned} \lim_{h\to 0} \frac{1}{h} \int_0^1 \Big(\int_0^{r(t,x_0+h)} & \phi(u(t,s), x_0+h) \Big\| \frac{\partial u}{\partial t} \times \frac{\partial u}{\partial s} \Big\|_2 ds \\ & - \int_0^{r(t,x_0)} \phi(u(t,s),x_0) \Big\| \frac{\partial u}{\partial t} \times \frac{\partial u}{\partial s} \Big\|_2 ds \Big) dt \\ = \lim_{h\to 0} \frac{1}{h} \int_0^1 \Big(\int_{r(t,x_0)}^{r(t,x_0+h)} & \phi(u(t,s), x_0+h) \Big\| \frac{\partial u}{\partial t} \times \frac{\partial u}{\partial s} \Big\|_2 ds \\ + \int_0^{r(t,x_0)} & (\phi(u(t,s),x_0+h) - \phi(u(t,s),x_0)) \Big\| \frac{\partial u}{\partial t} \times \frac{\partial u}{\partial s} \Big\|_2 ds \Big) dt, \end{aligned} \quad (2.19)$$

where $\times$ denotes the vector product and for brevity we omit that the partial derivatives $\frac{\partial u}{\partial t}$ and $\frac{\partial u}{\partial s}$ are evaluated at (t,s) in the integrals. Regarding the second integral in the last equality of (2.19), since

$$\begin{aligned} \lim_{h\to 0} \frac{1}{h} \Big((\phi(u(t,s),x_0+h) - \phi(u(t,s),x_0)) \Big\| \frac{\partial u}{\partial t} \times \frac{\partial u}{\partial s} \Big\|_2 \Big) & \\ = \frac{\partial \phi}{\partial x_0}(u(t,s),x_0) \Big\| \frac{\partial u}{\partial t} \times \frac{\partial u}{\partial s} \Big\|_2, & \end{aligned}$$

almost everywhere and this function is measurable and its integral over the bounded set $\Omega(x_0)$ is finite by hypothesis, the Lebesgue Dominated Convergence Theorem [Bartle, 1995] implies that

$$\lim_{h\to 0}\frac{1}{h}\int_0^1\int_0^{r(t,x_0)}(\phi(u(t,s),x_0+h)-\phi(u(t,s),x_0))\left\|\frac{\partial u}{\partial t}\times\frac{\partial u}{\partial s}\right\|_2 dsdt$$
$$=\int_0^1\int_0^{r(t,x_0)}\frac{\partial\phi}{\partial x}(u(t,s),x_0)\left\|\frac{\partial u}{\partial t}\times\frac{\partial u}{\partial s}\right\|_2 dsdt$$
$$=\int_{\Omega(x_0)}\frac{\partial\phi}{\partial x}(q,x_0)dq. \tag{2.20}$$

On the other hand, regarding the first integral in the last equality of (2.19), using the continuity of ϕ, one can deduce that

$$\lim_{h\to 0}\frac{1}{h}\int_0^1\int_{r(t,x_0)}^{r(t,x_0+h)}\phi(u(t,s),x_0+h)\left\|\frac{\partial u}{\partial t}(t,s)\times\frac{\partial u}{\partial s}(t,s)\right\|_2 ds\,dt$$
$$=\lim_{h\to 0}\frac{1}{h}\int_0^1\int_{x_0}^{x_0+h}\phi(u(t,r(t,z)),x_0+h)$$
$$\cdot\left\|\frac{\partial u}{\partial t}(t,r(t,z))\times\frac{\partial u}{\partial s}(t,r(t,z))\right\|_2\frac{\partial r}{\partial x}(t,z)\,dz\,dt$$
$$=\int_0^1\phi(u(t,r(t,x_0)),x_0)\left\|\frac{\partial u}{\partial t}(t,r(t,x_0))\times\frac{\partial u}{\partial s}(t,r(t,x_0))\right\|_2\frac{\partial r}{\partial x_0}(t,x_0)\,dt.$$

Since $\gamma(t,x)=u(t,r(t,x))$ for all $t\in[0,1]$ and $x\in(x_0-\varepsilon,x_0+\varepsilon)$, one has

$$\frac{\partial\gamma}{\partial t}(t,x_0)=\frac{\partial u}{\partial t}(t,r(t,x_0))+\frac{\partial u}{\partial s}(t,r(t,x_0))\frac{\partial r}{\partial t}(t,x_0)\,,$$
$$\frac{\partial\gamma}{\partial x}(t,x_0)=\frac{\partial u}{\partial s}(t,r(t,x_0))\frac{\partial r}{\partial x}(t,x_0).$$

Let χ denote the angle formed by $\frac{\partial\gamma}{\partial t}(t,x_0)$ and $\frac{\partial u}{\partial s}(t,r(t,x_0))$. Then (omitting the expression $(t,r(t,x))$ for brevity),

$$\left\|\frac{\partial u}{\partial t}\times\frac{\partial u}{\partial s}\right\|_2=\left\|\left(\frac{\partial u}{\partial t}+\frac{\partial u}{\partial s}\frac{\partial r}{\partial t}\right)\times\frac{\partial u}{\partial s}\right\|_2$$
$$=\left\|\frac{d\gamma}{dt}\right\|_2\left\|\frac{\partial u}{\partial s}\right\|_2\sin\chi=\left\|\frac{\partial\gamma}{\partial t}\right\|_2 n^T(\gamma)\frac{\partial u}{\partial s},$$

where in the last inequality we have used the fact that, since γ_{x_0} is a parameterization of $\partial\Omega(x_0)$, then $\sin\chi=\cos\psi$, where ψ is the angle formed by n, the outward normal to $\partial\Omega(x_0)$, and $\frac{\partial u}{\partial s}$. Therefore, we finally arrive at

$$\int_0^1\phi(\gamma(t),x_0)\left\|\frac{\partial u}{\partial t}(t,r(t,x_0))\times\frac{\partial u}{\partial s}(t,r(t,x_0))\right\|_2\frac{\partial r}{\partial x}(t,x_0)dt$$
$$=\int_0^1\phi(\gamma(t),x_0)\left\|\frac{\partial\gamma}{\partial t}(t,x_0)\right\|_2 n^T(\gamma(t,x_0))\frac{\partial\gamma}{\partial x}(t,x_0)dt$$
$$=\int_{\partial\Omega(x_0)}\phi(\gamma,x_0)n^T(\gamma)\frac{\partial\gamma}{\partial x}d\gamma. \tag{2.21}$$

Given the hypothesis of Proposition 2.24, both terms in (2.20) and (2.21) have a continuous dependence on $x_0 \in (a, b)$. This concludes the proof. ■

We are finally ready to state the proof of the main result of Section 2.3.

Proof (Theorem 2.17). We prove the theorem statement when the performance function is continuously differentiable and we refer to [Cortés et al., 2005] for the complete proof for the case when the performance function is piecewise continuously differentiable. Specifically, we show that if f is continuously differentiable, then for $P \in S^n \setminus \mathcal{S}_{\text{coinc}}$,

$$\frac{\partial \mathcal{H}_{\text{exp}}}{\partial p_i}(P) = \int_{V_i(\mathcal{P})} \frac{\partial}{\partial p_i} f(\|q - p_i\|_2)\phi(q)dq.$$

From Proposition 2.24, we have

$$\frac{\partial}{\partial p_i}\Big(\sum_{j=1}^{n} \int_{V_j(\mathcal{P})} f(\|q - p_j\|_2)\phi(q)dq\Big) = \int_{V_i(\mathcal{P})} \frac{\partial}{\partial p_i} f(\|q - p_i\|_2)\phi(q)dq + \sum_{j=1}^{n} \int_{\partial V_j(\mathcal{P})} \varphi(p_j, q)\Big(n(\gamma_j) \cdot \frac{\partial \gamma_j}{\partial p_i}\Big)d\gamma_j,$$

where γ_j is a parametrization of $V_j(\mathcal{P})$ and where we abbreviate $\varphi(p_j, q) = f(\|q - p_j\|_2)\phi(q)$. Next, we show that the second term vanishes. Note that the motion of p_i affects the Voronoi cell $V_i(\mathcal{P})$ and the cells of all its neighbors in $\mathcal{N}_{\mathcal{G}_{\text{D}}, p_i}(\mathcal{P})$. Therefore, the second term equals

$$\int_{\partial V_i(\mathcal{P})} \varphi(p_i, q)\Big(n(\gamma_i) \cdot \frac{\partial \gamma_i}{\partial p_i}\Big)d\gamma_i + \sum_{p_j \in \mathcal{N}_{\mathcal{G}_{\text{D}}, p_i}(\mathcal{P})} \int_{\partial V_j(\mathcal{P})} \varphi(p_j, q)\Big(n(\gamma_j) \cdot \frac{\partial \gamma_j}{\partial p_i}\Big)d\gamma_j.$$

Without loss of generality assume $V_i(\mathcal{P})$ does not share any face with ∂S. Since the boundary of $V_i(\mathcal{P})$ satisfies $\partial V_i(\mathcal{P}) = \bigcup_j \Delta_{ij}$, where $\Delta_{ij} = \Delta_{ji}$ is the edge between $V_i(\mathcal{P})$ and $V_j(\mathcal{P})$, for all neighbors p_j, we compute

$$\int_{\partial V_i(\mathcal{P})} \varphi(p_i, q)\Big(n(\gamma_i) \cdot \frac{\partial \gamma_i}{\partial p_i}\Big)d\gamma_i = \sum_{p_j \in \mathcal{N}_{\mathcal{G}_{\text{D}}, p_i}(\mathcal{P})} \int_{\Delta_{ij}} \varphi(p_i, q)\Big(n_{ij}(\gamma_j) \cdot \frac{\partial \gamma_j}{\partial p_i}\Big)d\gamma_j,$$

$$\int_{\partial V_j(\mathcal{P})} \varphi(p_j, q)\Big(n(\gamma_j) \cdot \frac{\partial \gamma_j}{\partial p_i}\Big)d\gamma_j = \int_{\Delta_{ji}} \varphi(p_j, q)\Big(n_{ji}(\gamma_j) \cdot \frac{\partial \gamma_j}{\partial p_i}\Big)d\gamma_j,$$

where n_{ij} denotes the unit normal along Δ_{ij} outward of $V_i(P)$. Noting that $n_{ji} = -n_{ij}$ and collecting the results obtained so far, we write

$$\sum_{j=1}^{n} \int_{\partial V_j(\mathcal{P})} \varphi(p_j, q)\Big(n(\gamma_j) \cdot \frac{\partial \gamma_j}{\partial p_i}\Big)d\gamma_j = \sum_{p_j \in \mathcal{N}_{\mathcal{G}_{\text{D}}, p_i}(\mathcal{P})} \int_{\Delta_{ij}} \Big(\varphi(p_i, q) - \varphi(p_j, q)\Big)\Big(n_{ij}(\gamma_j) \cdot \frac{\partial \gamma_j}{\partial p_i}\Big)d\gamma_j.$$

This quantity vanishes because $f(\|q-p_i\|_2) = f(\|q-p_j\|_2)$ and, therefore, $\varphi(p_i,q) = \varphi(p_j,q)$ for any q belonging to the edge Δ_{ij}. ■

2.5.5 Proof of Proposition 2.20

Proof. Recall that $\mathcal{H}_{\rm dc}(p_1,\dots,p_n) = \mathcal{H}_{\rm dc}(p_1,\dots,p_n,V_1(\mathcal{P}),\dots,V_n(\mathcal{P}))$. To show the first inequality, let $j \in \{1,\dots,n\}$ and $q_* \in V_j(\mathcal{P})$ be such that $\mathcal{H}_{\rm dc}(p_1,\dots,p_n) = \|q_*-p_j\|_2$. By definition, given a partition $\{W_1,\dots,W_n\}$ of S, there exists k such that $q_* \in W_k$. Therefore,

$$\begin{aligned}\mathcal{H}_{\rm dc}(p_1,\dots,p_n) &= \|q_*-p_j\|_2 \leq \|q_*-p_k\|_2 \\ &\leq \max_{q\in W_k} \|q-p_j\|_2 \leq \mathcal{H}_{\rm dc}(p_1,\dots,p_n,W_1,\dots,W_n).\end{aligned}$$

To show the second inequality, note that the definition of circumcenter implies that, for each $i \in \{1,\dots,n\}$,

$$\max_{q\in\partial W_i} \|q-{\rm CC}(W_i)\|_2 \leq \max_{q\in\partial W_i} \|q-p_i\|_2.$$

Taking the maximum over all nodes, we deduce

$$\mathcal{H}_{\rm dc}({\rm CC}(W_1),\dots,{\rm CC}(W_n),W_1,\dots,W_n) \leq \mathcal{H}_{\rm dc}(p_1,\dots,p_n,W_1,\dots,W_n),$$

as claimed. ■

2.5.6 Proof of Proposition 2.22

Proof. Recall that $\mathcal{H}_{\rm sp}(p_1,\dots,p_n) = \mathcal{H}_{\rm sp}(p_1,\dots,p_n,V_1(\mathcal{P}),\dots,V_n(\mathcal{P}))$. To show the first inequality, let $j \in \{1,\dots,n\}$ and $q_* \notin {\rm int}(V_j(\mathcal{P}))$ be such that $\mathcal{H}_{\rm sp}(p_1,\dots,p_n) = \|q_*-p_j\|_2$. Since $q_* \notin {\rm int}(V_j(\mathcal{P}))$, there exists $i \in \{1,\dots,n\}$ such that $\|q_*-p_j\|_2 \geq \|q_*-p_i\|_2$. On the other hand, there must exist $k \in \{1,\dots,n\}$ such that $q_* \in W_k$. Now, if $k=j$, then $q_* \notin {\rm int}(W_i)$. Therefore

$$\begin{aligned}\mathcal{H}_{\rm sp}(p_1,\dots,p_n) &= \|q_*-p_j\|_2 \geq \|q_*-p_i\|_2 \\ &\geq \min_{q\notin{\rm int}(W_i)} \|q-p_i\|_2 \geq \mathcal{H}_{\rm sp}(p_1,\dots,p_n,W_1,\dots,W_n).\end{aligned}$$

Now, if $k=i$, then $q_* \notin {\rm int}(W_j)$. Therefore

$$\mathcal{H}_{\rm sp}(P) = \|q_*-p_j\|_2 \geq \min_{q\notin{\rm int}(W_j)} \|q-p_i\|_2 \geq \mathcal{H}_{\rm sp}(p_1,\dots,p_n,W_1,\dots,W_n).$$

Finally, if $k \neq i,j$, then $q_* \notin {\rm int}(W_i) \cup {\rm int}(W_j)$, and a similar argument guarantees $\mathcal{H}_{\rm sp}(p_1,\dots,p_n) \geq \mathcal{H}_{\rm sp}(p_1,\dots,p_n,W_1,\dots,W_n)$.

To show the second inequality, let $i \in \{1,\dots,n\}$ and select $q_i \in {\rm IC}(W_i)$. The definition of incenter set implies that,

$$\min_{q\in\partial W_i} \|q-q_i\|_2 \geq \min_{q\in\partial W_i} \|q-p_i\|_2.$$

The expression on the left does not depend on the specific point selected in the incenter set. Taking the minimum over all nodes, we deduce

$$\mathcal{H}_{\mathrm{sp}}(\mathrm{IC}(W_1),\dots,\mathrm{IC}(W_n),W_1,\dots,W_n) \geq \mathcal{H}_{\mathrm{sp}}(p_1,\dots,p_n,W_1,\dots,W_n),$$

as claimed. ■

2.6 Exercises

E2.1 **(Proof of Lemma 2.2).** For $S = \{p_1,\dots,p_n\} \in \mathbb{F}(\mathbb{R}^d)$ with $n \geq 2$, prove the following statements:
(i) $\mathrm{CC}(S) \in \mathrm{co}(S) \setminus \mathrm{Ve}(\mathrm{co}(S))$;
(ii) if $p \in \mathrm{co}(S) \setminus \{\mathrm{CC}(S)\}$ and $r \in \mathbb{R}_{>0}$ are such that $S \subset \overline{B}(p,r)$, then $]p,\mathrm{CC}(S)[$ has a nonempty intersection with $\overline{B}(\frac{p+q}{2},\frac{r}{2})$ for all $q \in \mathrm{co}(S)$.
Hint: *To show (i), invoke the definition of circumcenter. To show (ii), distinguish between the case when $\|p-q\|_2 < r$ and $\|p-q\|_2 = r$. A proof is contained in [Cortés et al., 2006].*

E2.2 **(The inclusion $\mathcal{G}_{\mathbf{LD}}(r) \subset \mathcal{G}_{\mathbf{D}} \cap \mathcal{G}_{\mathbf{disk}}(r)$ is in general strict).** Consider the nodes $p_1 = (0,0)$, $p_2 = (1,0)$, and $p_3 = (2, \frac{1}{10})$. Pick $r = 3$ and perform the following tasks:
(i) draw the three points, their Voronoi partitions and the disks centered at the points with radius r, and
(ii) show that p_1 and p_3 are neighbors in the graph $\mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r)$, but not in the graph $\mathcal{G}_{\mathrm{LD}}(r)$.

E2.3 **(The proximity graph $\mathcal{G}_{\mathbf{D}} \cap \mathcal{G}_{\mathbf{disk}}(r)$ is not spatially distributed over $\mathcal{G}_{\mathbf{disk}}(r)$).** Consider the nodes $p_1 = (0,0)$, $p_2 = (1,0)$, $p_3 = (2, \frac{1}{10})$, and $p_4 = (0, \frac{31}{10})$. Compute the Voronoi partitions of the plane generated by $\{p_1,p_2.p_3\}$ and $\{p_1,p_2,p_3,p_4\}$. For $r = 3$, show that p_1 and p_3 are neighbors in the graph $\mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r)(\{p_1,p_2,p_3\})$ but not in the graph $\mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r)(\{p_1,p_2,p_3,p_4\})$. Why does this exercise illustrate that $\mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r)$ is not spatially distributed over $\mathcal{G}_{\mathrm{disk}}(r)$?

E2.4 **(1-center area problem).** Let $W \subset \mathbb{R}^2$ be a polygon, ϕ a density function on $\mathbb{R}^2$ and $a \in \mathbb{R}_{>0}$. Assume that the a-contraction of W is non-empty. Consider the area function $\mathcal{H}_1 : W \to \mathbb{R}$ defined by

$$\mathcal{H}_1(p) = \int_{W \cap \overline{B}(p,a)} \phi(q)dq = \mathrm{A}_\phi(W \cap \overline{B}(p,a)).$$

Justify informally why, at points in the boundary of a convex polygon W, the gradient of $\mathcal{H}_1$ is non-vanishing, and points toward the interior of the polygon. (Note that it is not known whether the function $\mathcal{H}_1$ is concave and how to characterize critical points of $\mathcal{H}_1$ in geometric terms.)

E2.5 **(Proof of Proposition 2.15).** This exercise asks you to prove a statement slightly more general than Proposition 2.15. Let $\{W_1,\dots,W_n\} \subset \mathbb{P}(S)$ be a partition of $S \subset \mathbb{R}^d$ and ϕ a density function on $\mathbb{R}^d$. For any $\{p_1,\dots,p_n\},\{\overline{p}_1,\dots,\overline{p}_n\} \in \mathbb{F}(S)$ with the property that, for all $i \in \{1,\dots,n\}$,

$$\|\overline{p}_i - \mathrm{CM}_\phi(W_i)\|_2 \leq \|p_i - \mathrm{CM}_\phi(W_i)\|_2$$

show that

$$\mathcal{H}_{\mathrm{distor}}(\overline{p}_1, \ldots, \overline{p}_n, W_1, \ldots, W_n) \geq \mathcal{H}_{\mathrm{distor}}(p_1, \ldots, p_n, W_1, \ldots, W_n),$$

Moreover, if all the sets $\{W_1, \ldots, W_n\}$ have non-vanishing areas, the inequality is strict if there exists $i \in \{1, \ldots, n\}$ such that $\|\overline{p}_i - \mathrm{CM}_\phi(W_i)\|_2 < \|p_i - \mathrm{CM}_\phi(W_i)\|_2$

Hint: *Use the expression of* $\mathcal{H}_{\mathrm{distor}}$ *in* (2.7).

E2.6 **(Mixed distortion-area multicenter function).** Show that the expected multicenter function $\mathcal{H}_{\mathrm{exp}}$ takes the form of $\mathcal{H}_{\text{distor-area},a}$ stated in Section 2.3.1 when the performance function is

$$f(x) = -x^2 \; 1_{[0,a]}(x) + b \cdot 1_{]a,+\infty[}(x),$$

with $a \in \mathbb{R}_{>0}$ and $b \leq -a^2$.

Hint: *As an intermediate step, show that for* $P = (p_1, \ldots, p_n) \in S^n$, *one has* $V_i(P) \cap (S \setminus \overline{B}(p_i, a)) = V_i(P) \cap \left(S \setminus \cup_{k=1}^n \overline{B}(p_k, a)\right)$ *for all* $i \in \{1, \ldots, n\}$.

E2.7 **(Proof of Proposition 2.16).** This exercise is a guided proof of Proposition 2.16. Let $W \subset \mathbb{R}^d$ be a connected set, ϕ a density function on $\mathbb{R}^d$ and $a \in \mathbb{R}_{>0}$. For $p \in W$ and $\overline{B}$ a closed ball centered at a point in W with radius a, define $(p, \overline{B}) \mapsto \mathcal{H}_W(p, \overline{B})$ by

$$\mathcal{H}_W(p, \overline{B}) = -\int_{W \cap \overline{B}} \|q - p\|_2^2 \phi(q) dq - \int_{W \cap (S \setminus \overline{B})} a^2 \phi(q) dq.$$

Do the following:

(i) Show that the multicenter function $\mathcal{H}_{\text{distor-area},a}$ admits the expression

$$\mathcal{H}_{\text{distor-area},a}(p_1, \ldots, p_n, W_1, \ldots, W_n) = \sum_{i=1}^n \mathcal{H}_{W_i}(p_i, \overline{B}(p_i, a));$$

(ii) Given a closed ball $\overline{B}$ centered at a point in W with radius a, show that for any $p \in W$,

$$\mathcal{H}_W(\mathrm{CM}_\phi(W \cap \overline{B}), \overline{B}) \geq \mathcal{H}_W(p, \overline{B}),$$

with strict inequality unless $p = \mathrm{CM}_\phi(W \cap \overline{B})$;

Hint: *Use the Parallel Axis Theorem [Hibbeler, 2003].*

(iii) Given $p \in W$, show that for any closed ball $\overline{B}$ centered at a point in W with radius a,

$$\mathcal{H}_W(p, \overline{B}(p, a)) \geq \mathcal{H}_W(p, \overline{B});$$

Hint: *Consider the decomposition of* W *given by the union of the disjoint sets* $\overline{B}(p, a) \cap \overline{B}$, $\overline{B}(p, a) \cap (W \setminus \overline{B})$, $(W \setminus \overline{B}(p, a)) \cap \overline{B}$ *and* $(W \setminus \overline{B}(p, a)) \cap (W \setminus \overline{B})$, *and compare the integrals over each set.*

(iv) Deduce, using (ii) and (iii), that

$$\mathcal{H}_W(\mathrm{CM}_\phi(W \cap \overline{B}(p, a)), \overline{B}(\mathrm{CM}_\phi(W \cap \overline{B}(p, a)), a)) \geq \mathcal{H}_W(p, \overline{B}(p, a)),$$

with strict inequality unless $p = \mathrm{CM}_\phi(W \cap \overline{B})$;

(v) Combine (i) and (iv) to prove Proposition 2.16.

E2.8 **(Locally-cliqueless proximity graph).** Give an example of an allowable environment Q and a configuration of points such that the inclusions of Theorem 2.12(i),

$$\mathcal{G}_{\mathrm{EMST},\mathcal{G}} \subseteq \mathcal{G}_{\mathrm{lc},\mathcal{G}} \subseteq \mathcal{G}$$

are strict for $\mathcal{G} = \mathcal{G}_{\mathrm{vis},Q}$.

E2.9 Prove Theorem 2.12.
Hint: *This exercise has considerable theoretical content. To prove Theorem 2.12(i), use an argument by contradiction to show that the first inclusion holds, and the definition of locally-cliqueless graph to show that the second inclusion holds.*

E2.10 Assume $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is continuously differentiable in its both arguments and let $\partial_1 f$ its partial derivative with respect to its first argument. Assume the function $y^* : \mathbb{R} \to \mathbb{R}$ satisfies, for each $x \in \mathbb{R}$,

$$f(x, y^*(x)) = \max\{f(x, z) \mid z \in \mathbb{R}\},$$

and is continuously differentiable. Show that

$$\frac{d}{dx} f(x, y^*(x)) = \partial_1 f(x, y^*(x)).$$

Explain how this result gives insight into the expression of the gradient of $\mathcal{H}_{\mathrm{exp}}$ in Theorem 2.17(ii) for smooth performance functions.

E2.11 **(Distortion gradient ascent flow).** Given a (convex) polytope $S \subset \mathbb{R}^d$ and a density function ϕ, consider n nodes $p_1, \ldots, p_n$ evolving under the continuous-time gradient ascent flow of the multicenter function $\mathcal{H}_{\mathrm{distor}}$,

$$\dot{p}_i = 2\,\mathrm{A}_\phi(V_i(\mathcal{P}))(\mathrm{CM}_\phi(V_i(\mathcal{P})) - p_i), \quad i \in \{1, \ldots, n\}.$$

(i) What are the equilibrium points?
(ii) Show that $\mathcal{H}_{\mathrm{distor}}$ is monotonically non-decreasing along the flow.
(iii) Justify that the polytope S is invariant, i.e., that the trajectories of the system remain in S.
(iv) Use (i)-(iii) to apply the LaSalle Invariance Principle and show that the solutions of the flow converge to the set of centroidal Voronoi configurations in S.
(v) Implement numerically the flow in the software of your choice. Select the unit square $S = [0, 1] \times [0, 1]$ and the density function

$$\phi = \exp\Big(-\big(x - \frac{1}{8}\big)^2 - \big(y - \frac{1}{8}\big)^2\Big) + \exp\Big(-\big(x - \frac{7}{8}\big)^2 - \big(y - \frac{7}{8}\big)^2\Big).$$

Run simulations from different initial conditions and with different numbers of nodes. Show by illustration that multiple local maxima exist.

E2.12 **(Area gradient ascent flow).** Given a (convex) polytope $S \subset \mathbb{R}^d$, a density function ϕ, and a radius $a \in \mathbb{R}_{>0}$, consider n nodes $p_1, \ldots, p_n$ evolving under the continuous-time gradient ascent flow of the multicenter function $\mathcal{H}_{\mathrm{area},a}$,

$$\dot{p}_i = \int_{V_i(\mathcal{P}) \cap \partial\overline{B}(p_i,a)} \mathrm{n}_{\mathrm{out},\overline{B}(p_i,a)}(q)\phi(q)dq, \quad i \in \{1, \ldots, n\}.$$

(i) What are the equilibrium points?
(ii) Show that $\mathcal{H}_{\text{area},a}$ is monotonically non-decreasing along the flow.
(iii) Justify that the polytope S is invariant, i.e., that the trajectories of the system remain in S.
(iv) Use (i)-(iii) to apply the LaSalle Invariance Principle and show that the solutions of the flow converge to the set of a-limited area-centered Voronoi configurations in S.
(v) Implement numerically the flow in the software of your choice. Select the unit square $S = [0,1] \times [0,1]$, the density function

$$\phi = \exp\Big(- \big(x - \frac{1}{8}\big)^2 - \big(y - \frac{1}{8}\big)^2 \Big) + \exp\Big(- \big(x - \frac{7}{8}\big)^2 - \big(y - \frac{7}{8}\big)^2 \Big),$$

and the parameter $a = \frac{1}{8}$. Run simulations from different initial conditions and with different numbers of nodes. Show by illustration that multiple local maxima exist.

3

Robotic network models and complexity notions

This chapter introduces the main subject of study of this book, namely a model for groups of robots that can sense their own position, communicate messages according to a geometric communication topology, process information, and control their motion. We refer to such systems as robotic networks. The exposition of this chapter has evolved from [Martínez et al., 2007a].

The chapter is organized as follows. The first section of the chapter contains the formal model. We begin by presenting the physical components of a network, that is, the mobile robots and the communication service connecting them. We then present the notion of control and communication law, and how a law is executed by a robotic network. These notions subsume the notions of synchronous network and distributed algorithm described in Section 1.4. As an example of these notions, we introduce a simple law, called the agree and pursue law, which combines ideas from leader election algorithms and from cyclic pursuit (i.e., a game in which robots chase each other in a circular environment). In the second section of the chapter, we propose a model of groups of robots that interact through sensing, rather than communication. The third section of the chapter discusses time, space, and communication complexity notions for robotic networks as extensions of the corresponding notions for distributed algorithms. The complexity notions rely on the basic concept of coordination task and task achievement. The fourth and last section of the chapter establishes the time, space, and communication required by the agree and pursue law to steer a group of robots to a uniformly-spaced rotating configuration.

3.1 A model for synchronous robotic networks

Here we introduce a model for a synchronous robotic network. This model is an extension of the synchronous network model in Section 1.4.1. We start by detailing the physical components of the network, which include the robots themselves as well as the communicate service among them.

3.1.1 Physical components

Let us start by providing a basic definition of robot and a model for how each robot moves in space.

A *mobile robot* is a continuous-time continuous-space dynamical system as defined in Section 1.2; that is, a tuple (X, U, X_0, f), where

(i) X is 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 $d_1 + d_2 = d$, called the *state space*;
(ii) U is a compact subset of $\mathbb{R}^m$ containing $\mathbf{0}_m$, called the *input space*;
(iii) X_0 is a subset of X, called the *set of allowable initial states*;
(iv) $f : X \times U \to \mathbb{R}^d$ is a smooth control vector field on X; that is, f determines the robot motion $x : \mathbb{R}_{\geq 0} \to X$ via the differential equation, or control system,

$$\dot{x}(t) = f(x(t), u(t)), \tag{3.1}$$

subject to the control $u : \mathbb{R}_{\geq 0} \to U$.

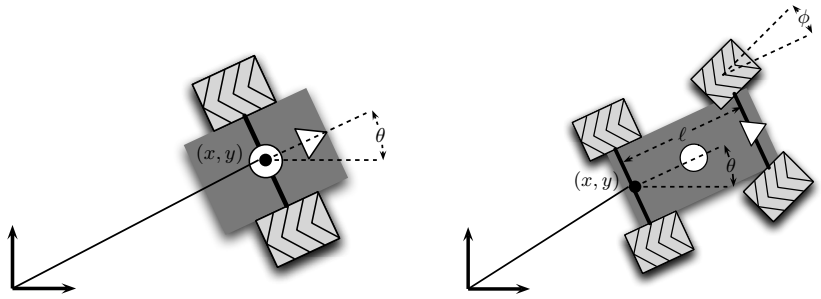
We will use the terms robot and agent interchangeably. We refer to $x \in X$ and $u \in U$ as a *physical state* and an *input* of the mobile robot, respectively. Most often the physical state will have the interpretation of a location, or a location and velocity. We will often consider control-affine vector fields. In such a case, we represent f as the ordered family of smooth vector fields $(f_0, f_1, \ldots, f_m)$ on X. In general the control signal u will not depend only on time but also on x and possible other variables in the system. Note that there is no additional difficulty in modeling mobile robots using dynamical systems defined on manifolds [Bullo and Lewis, 2004], but we avoid it here in the interest of simplicity.

Example 3.1 (Planar vehicle models). The following models of control systems are commonly used in robotics, beginning with the early works in [Dubins, 1957, Reeds and Shepp, 1990]. Figure 3.1(a) and (b) show a two-wheeled vehicle and a four-wheeled vehicle, respectively. The two-wheeled planar vehicle is described by the dynamical system:

$$\dot{x} = v \cos\theta, \quad \dot{y} = v \sin\theta, \quad \dot{\theta} = \omega, \tag{3.2}$$

with state variables $x \in \mathbb{R}$, $y \in \mathbb{R}$ and $\theta \in \mathbb{S}^1$, describing the planar position and orientation of the vehicle, and with controls v and ω, describing the forward linear velocity and the angular velocity of the vehicle. Depending on what set the controls are restricted to, we define the models:

The unicycle: The controls v and ω take value in $[-1, 1]$ and $[-1, 1]$, respectively.
The differential drive robot: Set $v = (\omega_{\text{right}} + \omega_{\text{left}})/2$ and $\omega = (\omega_{\text{right}} - \omega_{\text{left}})/2$ and assume both ω_{right} and ω_{left} take value in $[-1, 1]$.
The Reeds–Shepp car: The control v takes values in $\{-1, 0, 1\}$ and the control ω takes values in $[-1, 1]$.



(a) (b)

Fig. 3.1. (a) Two-wheeled vehicle and (b) four-wheeled vehicle. In each case, the orientation of the vehicle is indicated by the small triangle.

The Dubins vehicle: The control v is set equal to 1 and the control ω takes value in $[-1, 1]$.

Finally, the four-wheeled planar vehicle, composed of a rear and a front axle separated by a distance ℓ, is described by the same dynamical system (3.2) with the following distinctions: $(x, y) \in \mathbb{R}^2$ is the position of the midpoint of the rear axle, $\theta \in \mathbb{S}^1$ is the orientation of the rear axle, the control v is the forward linear velocity of the rear axle and the angular velocity satisfies $\omega = \frac{v}{\ell} \tan \phi$, where the control ϕ is the steering angle of the vehicle. •

The following definition is a generalization of the concept of synchronous network introduced in Definition 1.33.

Definition 3.2 (Robotic network). The physical components of a *robotic network* $\mathcal{S}$ consist of a tuple $(I, \mathcal{R}, E_{\text{cmm}})$, where

(i) $I = \{1, \ldots, n\}$; I is called the *set of unique identifiers (UIDs)*;
(ii) $\mathcal{R} = \{R^{[i]}\}_{i \in I} = \{(X^{[i]}, U^{[i]}, X_0^{[i]}, f^{[i]})\}_{i \in I}$ is a set of mobile robots;
(iii) E_{cmm} is a map from $\prod_{i \in I} X^{[i]}$ to the subsets of $I \times I$; this map is called the *communication edge map*.

If $R^{[i]} = (X, U, X_0, f)$ for all i, then the robotic network is called *uniform*. •

Remarks 3.3. (i) Following the convention established in Section 1.4, we let the superscript $[i]$ denote the variables and spaces which correspond to the robot with unique identifier i; for instance, $x^{[i]} \in X^{[i]}$ and $x_0^{[i]} \in X_0^{[i]}$ denote the physical state and the initial physical state of robot $R^{[i]}$, respectively. We refer to $x = (x^{[1]}, \ldots, x^{[n]}) \in \prod_{i \in I} X^{[i]}$ as a *state* of the network.
(ii) The map $x \mapsto (I, E_{\text{cmm}}(x))$ models the topology of the communication service among the robots: at a physical state $x = (x^{[1]}, \ldots, x^{[n]})$, two

robots at locations $x^{[i]}$ and $x^{[j]}$ can communicate if and only if the pair (i, j) is an edge in $E_{\text{cmm}}(x) = E_{\text{cmm}}(x^{[1]}, \ldots, x^{[n]})$. Accordingly, we refer to $(I, E_{\text{cmm}}(x))$ as the *communication graph* at x. When and what robots communicate is discussed in Section 3.1.2. As communication graphs, we will often adopt one of the proximity graphs discussed in Section 2.2 and in particular the (undirected) disk graph. •

To make things concrete, let us present some examples of robotic networks that will be commonly used later.

Example 3.4 (First-order robots with range-limited communication). Consider a group of robots moving in $\mathbb{R}^d$, $d \geq 1$. As in Chapter 2, we let p denote a point in $\mathbb{R}^d$ and we let $\{p^{[1]}, \ldots, p^{[n]}\}$ denote the robot locations. Assume the robots move according to

$$\dot{p}^{[i]}(t) = u^{[i]}(t), \tag{3.3}$$

with $u^{[i]} \in [-u_{\max}, u_{\max}]$. See Figure 3.2 for an illustration. According to our

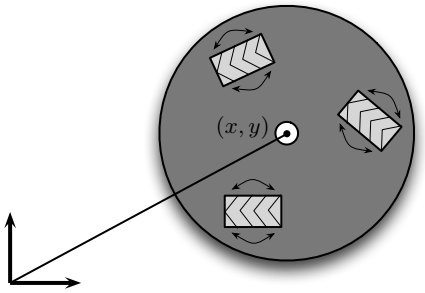


Fig. 3.2. Omnidirectional vehicle. In addition to controlling the rotation speed of the wheels, the vehicle can also actuate the direction in which they point. This allows the vehicle to move in any direction according to the first-order dynamics (3.3).

mobile robot notation, these are identical robots of the form

$$(\mathbb{R}^d, [-u_{\max}, u_{\max}]^d, \mathbb{R}^d, (\mathbf{0}_d, \boldsymbol{e}_1, \ldots, \boldsymbol{e}_d)).$$

We assume that each robot can sense its own position and can communicate to any other robot within distance r, that is, we adopt the r-disk graph $\mathcal{G}_{\text{disk}}(r)$ defined in Section 2.2 as communication graph. These data define the uniform robotic network $\mathcal{S}_{\text{disk}}$.

It shall also be interesting to consider first-order robots with communication graphs other than the disk graph; important examples include the Delaunay graph $\mathcal{G}_{\text{D}}$, the limited Delaunay graph $\mathcal{G}_{\text{LD}}(r)$, and the ∞-disk graph

$\mathcal{G}_{\infty\text{-disk}}(r)$, discussed in Section 2.2. These three graphs, adopted as communication models, give rise to three robotic networks denoted $\mathcal{S}_{\text{D}}$, $\mathcal{S}_{\text{LD}}$, $\mathcal{S}_{\infty\text{-disk}}$, respectively. •

Example 3.5 (Planar vehicle robots with Delaunay communication). We consider a group of vehicle robots moving in an allowable environment $Q \subset \mathbb{R}^2$ according to the planar vehicle dynamics introduced in Example 3.1. We let $\{(p^{[1]}, \theta^{[1]}), \ldots, (p^{[n]}, \theta^{[n]})\}$ denote the robot physical states, where $p^{[i]} = (x^{[i]}, y^{[i]}) \in Q$ corresponds to the position and $\theta^{[i]} \in \mathbb{S}^1$ corresponds to the orientation of the robot $i \in I$. As communication graph, we adopt the Delaunay graph $\mathcal{G}_{\text{D}}$ on Q introduced in Section 2.2. These data define the uniform robotic network $\mathcal{S}_{\text{vehicles}}$. •

Example 3.6 (First-order robots with line-of-sight communication). We consider a group of robots moving in an allowable environment $Q \subset \mathbb{R}^2$. As in Example 3.4, we let $\{p^{[1]}, \ldots, p^{[n]}\}$ denote the robot locations and assume the robots move according to the motion model (3.3). Each robot can sense its own position, the boundary of ∂Q, and can communicate to any other robot within distance r and within line of sight, that is, we adopt the range-limited visibility graph $\mathcal{G}_{\text{vis-disk},Q}$ in Q defined in Section 2.2 as the communication graph. These data define the uniform robotic network $\mathcal{S}_{\text{vis-disk}}$. •

Example 3.7 (First-order robots in $\mathbb{S}^1$). Consider n robots $\{\theta^{[1]}, \ldots, \theta^{[n]}\}$ in $\mathbb{S}^1$, moving along on the unit circle with angular velocity equal to the control input. Each robot is described by the tuple $(\mathbb{S}^1, [-u_{\max}, u_{\max}], \mathbb{S}^1, (0, \boldsymbol{e}))$, where $\boldsymbol{e}$ is the vector field on $\mathbb{S}^1$ describing unit-speed counterclockwise rotation. As in the previous examples, we assume that each robot can sense its own position and can communicate to any other robot within distance r along the circle, that is, we adopt the r-disk graph $\mathcal{G}_{\text{disk}}(r)$ on $\mathbb{S}^1$ defined in Section 2.2 as the communication graph. These data define the uniform robotic network $\mathcal{S}_{\text{circle}}$. •

We conclude this section with a remark.

Remark 3.8 (Congestion models in robotic networks). The behavior of a robotic network might be affected by communication and physical congestion problems.

Communication congestion: Omnidirectional wireless transmissions interfere. Clear reception of a signal requires that no other signals are present at the same point in time and space. In an ad hoc network, node i receives a message transmitted by node j only if all other neighbors of i are silent. In other words, the transmission medium is shared among the agents. As the density of agents increases, so does wireless communication congestion. Asymptotic and optimization results are known on this regard.

First, for ad hoc networks with n uniformly randomly placed nodes, it is known [Gupta and Kumar, 2000] that the maximum-throughput

communication range $r(n)$ of each node decreases as the density of nodes increases; in d dimensions the appropriate scaling law is $r(n) \in \Theta\big((\log(n)/n)^{1/d}\big)$. This is referred to as the *connectivity regime* in percolation theory and statistical mechanics. Using the k-nearest-neighbor graph over uniformly placed nodes, the analysis in Xue and Kumar [2004] suggests that the minimal number of neighbors in a connected network grows with $\log(n)$.

Second, a growing body of literature [Santi, 2005, Lloyd et al., 2005] is available on *topology control*, i.e., on how to compute transmission power values in an ad hoc network so as to minimize energy consumption and interference (due to multiple sources), while achieving various graph topological properties, such as connectivity or low network diameter.

Physical congestion: Robots can collide: it is clearly important to avoid "simultaneous access to the same physical area" by multiple robots. It is reasonable to assume that, as the number of robots increase, so should the area available for their motion. An alternative convenient approach is the one taken by Sharma et al. [2007], where robots' safety zones decrease with decreasing robots' speed. This suggests that, in a fixed environment, individual nodes of a large ensemble have to move at a speed decreasing with n, and in particular, at a speed proportional to $n^{-1/d}$. Roughly speaking, if the overall volume V where the groups of agents move is constant, and there are n robots, then the speed v they can move at goes approximately as $v^d \approx \frac{V}{n}$.

In summary, one way to incorporate congestion effects into the robotic network model is to assume that the parameters of the physical components of the network depend upon the number of robots n. In the limit as $n \to +\infty$ we will sometimes assume that r and $u_{\max}$, the communication range and the velocity upper bound in Examples 3.4 and 3.7, are of order $n^{-1/d}$. •

3.1.2 Control and communication laws

Here we present a discrete-time communication, continuous-time motion model for the evolution of a robotic network subject to a communication and control law. In our model, each robot evolves in the physical domain in continuous time, senses its position in continuous time, and, in discrete time, exchanges information with other robots and executes a state machine, which we shall refer to as a processor. The following definition is a generalization of the concept of distributed algorithm introduced in Definition 1.34 and of the classical notion of dynamical feedback controller.

Definition 3.9 (Control and communication law). A *control and communication law* $\mathcal{CC}$ for a robotic 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*;
(iii) $W_0^{[i]} \subseteq W^{[i]}$, $i \in I$, sets of *allowable initial values*;

and of the maps:

(i) $\mathrm{msg}^{[i]} : X^{[i]} \times W^{[i]} \times I \to \mathbb{A}$, $i \in I$, called *message-generation functions*;
(ii) $\mathrm{stf}^{[i]} : X^{[i]} \times W^{[i]} \times \mathbb{A}^n \to W^{[i]}$, $i \in I$, called *(processor) state-transition functions*;
(iii) $\mathrm{ctl}^{[i]} : X^{[i]} \times X^{[i]} \times W^{[i]} \times \mathbb{A}^n \to U^{[i]}$, $i \in I$, called *(motion) control functions.*

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

We will sometimes refer to a control and communication law as a *distributed motion coordination algorithm*. Roughly speaking, the rationale behind Definition 3.9 is the following: The state of robot i includes both the physical state $x^{[i]} \in X^{[i]}$ and the *processor state* $w^{[i]} \in W^{[i]}$ of the state machine that robot i implements. These states are initialized with values in their corresponding allowable initial sets $X_0^{[i]}$ and $W_0^{[i]}$. We assume the robot can sense it own physical position $x^{[i]}$. At each time instant $\ell \in \mathbb{Z}_{\geq 0}$, robot i sends to each of its out-neighbors j in the communication digraph $(I, E_{\mathrm{cmm}}(x))$ a message (possibly the `null` message) computed by applying the message-generation function $\mathrm{msg}^{[i]}$ to the current values of its physical state $x^{[i]}$, processor state $w^{[i]}$ and to the identity j. Subsequently, but still at the time instant $\ell \in \mathbb{Z}_{\geq 0}$, robot i updates the value of its processor state $w^{[i]}$ by applying the state-transition function $\mathrm{stf}^{[i]}$ to the current value of its physical state $x^{[i]}$, processor state $w^{[i]}$ and to the messages it receives from its in-neighbors. Between communication instants, i.e., for $t \in [\ell, \ell + 1)$ for some $\ell \in \mathbb{Z}_{\geq 0}$, the motion of the ith robot is determined by applying the control function to the current value of $x^{[i]}$, the value of $x^{[i]}$ at time ℓ, the current value of $w^{[i]}$, and the messages received at time ℓ. This evolution model is very similar to the one we introduced for synchronous networks in Definition 1.35: at each communication round, the first step is transmission and the second one is computation and, except for the dependence on the physical state x, the communication and state transition processes are identical.

These ideas are formalized in the following definition.

Definition 3.10 (Evolution of a robotic network). Let $\mathcal{CC}$ be a control and communication law for the robotic network $\mathcal{S}$. The *evolution* of $(\mathcal{S}, \mathcal{CC})$ from initial conditions $x_0^{[i]} \in X_0^{[i]}$ and $w_0^{[i]} \in W_0^{[i]}$, $i \in I$, is the collection of curves $x^{[i]} : \mathbb{R}_{\geq 0} \to X^{[i]}$ and $w^{[i]} : \mathbb{Z}_{\geq 0} \to W^{[i]}$, $i \in I$, defined by

$$\dot{x}^{[i]}(t) = f\Big(x^{[i]}(t),\, \mathrm{ctl}^{[i]}\big(x^{[i]}(t), x^{[i]}(\lfloor t \rfloor), w^{[i]}(\lfloor t \rfloor), y^{[i]}(\lfloor t \rfloor)\big)\Big),$$

where $\lfloor t \rfloor = \max\{\ell \in \mathbb{Z}_{\geq 0} \mid \ell < t\}$, and

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

with $x^{[i]}(0) = x_0^{[i]}$, and $w^{[i]}(-1) = w_0^{[i]}$, $i \in I$. In the previous equations, $y^{[i]} : \mathbb{Z}_{\geq 0} \to \mathbb{A}^n$ (describing the messages received by processor i) has components $y_j^{[i]}(\ell)$, for $j \in I$, defined by

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

•

For convenience, we define $w(t) = w(\lfloor t \rfloor)$ for all $t \in \mathbb{R}_{\geq 0}$, and let $\mathbb{R}_{\geq 0} \ni t \mapsto (x(t), w(t))$ denote the curves $x^{[i]}$ and $w^{[i]}$, for $i \in \{1, \ldots, n\}$.

Remark 3.11 (Simplifications of control and communication laws).

(i) A control and communication law $\mathcal{CC}$ is *static* if the processor state set $W^{[i]}$ is a singleton for all $i \in I$. This means that there is no meaningful evolution of the processor state. In this case $\mathcal{CC}$ can be described by a tuple $(\mathbb{A}, \{\mathrm{msg}^{[i]}\}_{i\in I}, \{\mathrm{ctl}^{[i]}\}_{i\in I})$, with $\mathrm{msg}^{[i]} : X^{[i]} \times I \to \mathbb{A}$, and $\mathrm{ctl}^{[i]} : X^{[i]} \times X^{[i]} \times \mathbb{A}^n \to U^{[i]}$, for $i \in I$;

(ii) A control and communication law $\mathcal{CC}$ is *data-sampled* if the control functions are independent of the current position of the robot and depend only upon the robot position at last sample time. Specifically, the control functions have the following property: given a processor state $w^{[i]} \in W^{[i]}$, an array of messages $y^{[i]} \in \mathbb{A}^n$, a current state $x^{[i]}$, and a state at last sample time $x^{[i]}_{\mathrm{smpld}}$, the control input $\mathrm{ctl}^{[i]}(x^{[i]}, x^{[i]}_{\mathrm{smpld}}, w^{[i]}, y^{[i]})$ is independent of $x^{[i]}$, for all $i \in I$. In this case the control functions can be described by maps of the form $\mathrm{ctl}^{[i]} : X^{[i]} \times W^{[i]} \times \mathbb{A}^n \to U^{[i]}$, for $i \in I$.

(iii) In many control and communication laws, the robots exchange their states, including both their processor and their physical states. For such laws, we identify the communication alphabet with $\mathbb{A} = (X \times W) \cup \{\texttt{null}\}$ and we refer to the corresponding message generation function $\mathrm{msg}_{\mathrm{std}}(x, w, j) = (x, w)$ as the *standard message-generation function*.

Note that we allow the processor state set and the communication alphabet to contain an infinite number of symbols. In other words, we assume that a robot can store and transmit a (finite number of) integer and real numbers, among other things. This is equivalent to assuming that we neglect any inaccuracies due to quantization, as we did in Section 1.5. •

Remark 3.12 (Extensions of control and communication laws). Here we briefly discuss alternative models and extensions of the proposed models.

Asynchronous sensor-based interactions: In the model proposed in Suzuki and Yamashita [1999], robots are referred to as "anonymous" and "oblivious" in precisely the same way in which we defined control and communication laws to be uniform and static, respectively. As compared with our

notion of robotic network, the model in [Suzuki and Yamashita, 1999] is more general in that the robots' activations schedules do not necessarily coincide (i.e., this model is asynchronous), and at the same time it is less general in that (1) robots cannot communicate any information other than their respective positions, and (2) each robot observes every other robot's position (i.e., the complete communication graph is adopted). In the next Section 3.2 we present a model where robots rely on sensing rather than communication for their interaction.

Discrete and continuous time in motion and communication: In some cases it will be convenient to consider discrete-time motion models; for example, we present discrete-time motion models for first order agents in Section 4.1. In some other cases, it will be convenient to consider dynamical interactions between agents taking place in continuous time.

Stochastic link models: Although we do not present any result on this topic in this notes, it is possible to develop robotic networks models over random graphs and random geometric graphs, as studied in Bollobás [2001], Penrose [2003]. Furthermore, it is of interest to consider communication links with time-varying rates. •

3.1.3 Agree and pursuit control and communication law

We conclude this section with an example of a dynamic control and communication law. The problem is described as follows: a collection of robots with range-limited communication are placed on the unit circle; the robots move and communicate with the objectives of (1) agreeing on a direction of motion (clockwise or counterclockwise) and (2) achieving an equidistant configuration where all robots are equally angularly-spaced. To achieve these two objectives, we combine ideas from leader election algorithms for synchronous networks (see Section 1.4.4) and from cyclic pursuit problems (see Exercise E1.22): the robots move a distance proportional to an appropriate inter-robot separation, and they repeatedly compare their identifiers to discover the direction of motion of the robot with the largest identifier. In other words, the robots run a leader election task in their processor states and a uniform robotic deployment task in their physical state — these are among the most basic tasks in distributed algorithms and cooperative control. We present the algorithm here and characterize its correctness and performance later in the chapter.

From Example 3.7, we consider the uniform network $\mathcal{S}_{\text{circle}}$ of locally-connected first-order robots on $\mathbb{S}^1$. For $r, u_{\max}, k_{\text{prop}} \in]0, \frac{1}{2}[$ with $k_{\text{prop}} r \leq u_{\max}$, we define the AGREE & PURSUE law, denoted by $\mathcal{CC}_{\text{AGREE \& PURSUE}}$, as the *uniform data-sampled* law loosely described as follows:

> *[Informal description]* The processor state consists of `dir` (the robot's direction of motion) taking values in $\{\texttt{c}, \texttt{cc}\}$ (meaning clockwise and counterclockwise) and `max-id` (the largest UID received by the robot, initially set to the robot's UID) taking values in I. At each communication round, each robot transmits its position and its processor state.

Among the messages received from agents moving towards its position, each agent picks the message with the largest value of `max-id`. If this value is larger than its own value, the agent resets its processor state with the selected message. Between communication rounds, each robot moves in the clockwise or counterclockwise direction depending on whether its processor state `dir` is `c` or `cc`. Each robot moves k_{prop} times the distance to the immediately next neighbor in the chosen direction, or, if no neighbors are detected, k_{prop} times the communication range r.

Note that the processor state with the largest UID will propagate throughout the network as in the FLOODMAX algorithm for leader election. Also, note that the assumption $k_{\text{prop}} r \leq u_{\max}$ guarantees that the desired control is always within the allowable range $[-u_{\max}, u_{\max}]$. Next, we define the law formally.

`Robotic Network:` $\mathcal{S}_{\text{circle}}$, first-order agents in $\mathbb{S}^1$
with absolute sensing of own position, and
with communication range r

`Distributed Algorithm:` AGREE & PURSUE

`Alphabet`: $\mathbb{A} = \mathbb{S}^1 \times \{\texttt{c}, \texttt{cc}\} \times I \cup \{\texttt{null}\}$

`Processor State`: $w = (\texttt{dir}, \texttt{max-id})$, where

`dir` $\in \{\texttt{c}, \texttt{cc}\}$, initially: $\texttt{dir}^{[i]}$ unspecified
`max-id` $\in I$, initially: $\texttt{max-id}^{[i]} = i$ for all i

`function` msg(θ, w, i) *% Standard message generation function*

1: **return** (θ, w)

`function` stf(θ, w, y)

1: **for** each non-`null` message $(\theta_{\text{rcvd}}, (\texttt{dir}_{\text{rcvd}}, \texttt{max-id}_{\text{rcvd}}))$ in y **do**
2: **if** $(\texttt{max-id}_{\text{rcvd}} > \texttt{max-id})$ AND $(\text{dist}_{\texttt{cc}}(\theta, \theta_{\text{rcvd}}) \leq r$ AND $\texttt{dir}_{\text{rcvd}} = \texttt{c})$ OR $(\text{dist}_{\texttt{c}}(\theta, \theta_{\text{rcvd}}) \leq r$ AND $\texttt{dir}_{\text{rcvd}} = \texttt{cc})$ **then**
3: $\texttt{new-dir} := \texttt{dir}_{\text{rcvd}}$
4: $\texttt{new-id} := \texttt{max-id}_{\text{rcvd}}$
5: **return** $(\texttt{new-dir}, \texttt{new-id})$

`function` ctl$(\theta_{\text{smpld}}, w, y)$

1: $d_{\text{tmp}} := r$
2: **for** each non-`null` message $(\theta_{\text{rcvd}}, (\texttt{dir}_{\text{rcvd}}, \texttt{max-id}_{\text{rcvd}}))$ in y **do**
3: **if** $(\texttt{dir} = \texttt{cc})$ AND $(\text{dist}_{\texttt{cc}}(\theta_{\text{smpld}}, \theta_{\text{rcvd}}) < d_{\text{tmp}})$ **then**
4: $d_{\text{tmp}} := \text{dist}_{\texttt{cc}}(\theta_{\text{smpld}}, \theta_{\text{rcvd}})$
5: $u_{\text{tmp}} := k_{\text{prop}} d_{\text{tmp}}$
6: **if** $(\texttt{dir} = \texttt{c})$ AND $(\text{dist}_{\texttt{c}}(\theta_{\text{smpld}}, \theta_{\text{rcvd}}) < d_{\text{tmp}})$ **then**
7: $d_{\text{tmp}} := \text{dist}_{\texttt{c}}(\theta_{\text{smpld}}, \theta_{\text{rcvd}})$
8: $u_{\text{tmp}} := -k_{\text{prop}} d_{\text{tmp}}$
9: **return** u_{tmp}

An implementation of this control and communication law is shown in Figure 3.3. As parameters we select $n = 45$, $r = 2\pi/40$, $u_{\max} = 1/4$ and $k_{\text{prop}} = 7/16$. Along the evolution, all robots agree upon a common direction of motion and, after suitable time, they reach a uniform distribution.

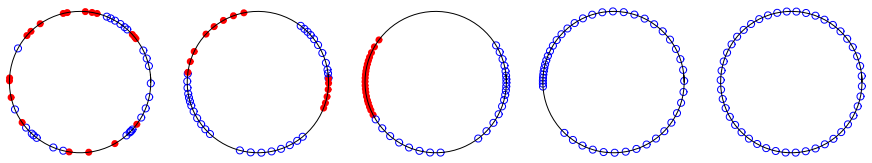


Fig. 3.3. The AGREE & PURSUE law. Disks and circles correspond to robots moving counterclockwise and clockwise, respectively. The initial positions and the initial directions of motion are randomly generated. The five pictures depect the network state at times 0, 9, 20, 100, and 800.

3.2 Robotic networks with relative sensing

The model presented above assumes the ability of each robot to know its own absolute position. Here, we treat the alternative setting in which the robots do not communicate amongst themselves, but instead detect and measure each other's relative position through appropriate sensors. Additionally, we assume that the robots will perform measurements of the environment without having any a priory knowledge of it. We assume that robots do not have the ability to perform measurements expressed in a common reference frame. An early reference where relative information is adopted is Lin et al. [2005].

3.2.1 Kinematics notions

Because robots do not have a common reference frame, all the measurements generated by their on-board sensors are expressed in a local reference frame. To formalize this fact, it is useful to review some basic kinematics conventions. We let $\Sigma^{\text{fixed}} = (p^{\text{fixed}}, \{\boldsymbol{x}^{\text{fixed}}, \boldsymbol{y}^{\text{fixed}}, \boldsymbol{z}^{\text{fixed}}\})$ be a fixed reference frame in $\mathbb{R}^3$. A point q, a vector v, and a set of points S expressed with respect to the frame Σ^{fixed} are denoted by q_{fixed}, v_{fixed} and S_{fixed}, respectively. Next, let $\Sigma^{\text{b}} = (p^{\text{b}}, \{\boldsymbol{x}^{\text{b}}, \boldsymbol{y}^{\text{b}}, \boldsymbol{z}^{\text{b}}\})$ be a reference frame fixed with a moving body. The origin of Σ^{b} is the point p^{b}, denoted by $p^{\text{b}}_{\text{fixed}}$ when expressed with respect to Σ^{fixed}. The orientation of Σ^{b} is characterized by the d-dimensional rotation matrix $R^{\text{b}}_{\text{fixed}}$, whose columns are the frame vectors $\{\boldsymbol{x}^{\text{b}}, \boldsymbol{y}^{\text{b}}, \boldsymbol{z}^{\text{b}}\}$ of Σ^{b} expressed with respect to Σ^{fixed}. We recall here the definition of the group of rotation matrices in d-dimensions:

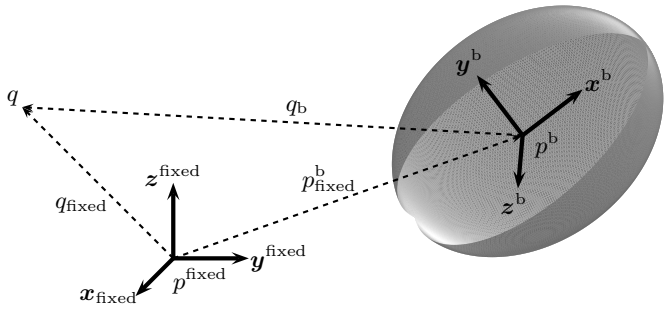


Fig. 3.4. Inertially-fixed and body-fixed frames in $\mathbb{R}^3$

$$\mathrm{SO}(d) = \{R \in \mathbb{R}^{d \times d} \mid RR^T = I_d,\ \det(R) = +1\}.$$

With these notations, changes of reference frames are described by

$$\begin{aligned} q_{\mathrm{fixed}} &= R^{\mathrm{b}}_{\mathrm{fixed}} q_{\mathrm{b}} + p^{\mathrm{b}}_{\mathrm{fixed}}, \\ v_{\mathrm{fixed}} &= R^{\mathrm{b}}_{\mathrm{fixed}} v_{\mathrm{b}}, \\ S_{\mathrm{fixed}} &= R^{\mathrm{b}}_{\mathrm{fixed}} S_{\mathrm{b}} + p^{\mathrm{b}}_{\mathrm{fixed}}. \end{aligned} \tag{3.4}$$

Note that these change of frames formulas also hold in the planar case with the corresponding definition of rotation matrix in SO(2).

Remark 3.13 (Comparison with literature). In our notation the subscript denotes the frame with respect to which the quantity is expressed. Some other references, e.g., [Spong et al., 2006] adopt the opposite convention in which the superscript denotes the frame with respect to which the quantity is expressed. •

3.2.2 The physical components

In what follows we describe our notion of *mobile robots equipped with relative sensors*. We consider a group of n robots moving in an allowable environment $Q \subset \mathbb{R}^d$, for $d \in \{2, 3\}$, and we assume that a reference frame $\Sigma^{[i]}$, for $i \in \{1, \ldots, n\}$, is attached to each robot, see Figure 3.5. Expressed with respect to the fixed frame Σ^{fixed}, the ith frame $\Sigma^{[i]}$ is described by a position $p^{[i]}_{\text{fixed}} \in \mathbb{R}^d$ and an orientation $R^{[i]}_{\text{fixed}} \in \text{SO}(d)$. The continuous-time motion and discrete-time sensing models are described as follows.

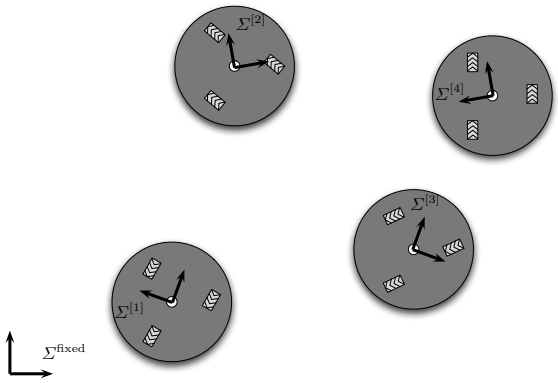


Fig. 3.5. Robotic network with relative sensing. A group of 4 robots moves in $\mathbb{R}^2$. Each robot $i \in \{1, \ldots, 4\}$ has its own reference frame $\Sigma^{[i]}$.

Motion model: We select a simple motion model: for all $t \in \mathbb{R}_{\geq 0}$, the orientation $R^{[i]}_{\text{fixed}}$ is constant in time and robot i translates according to

$$\dot{p}^{[i]}_{\text{fixed}}(t) = R^{[i]}_{\text{fixed}} u^{[i]}_i, \tag{3.5}$$

that is, the ith control input $u^{[i]}_i$ is known and applied in the robot frame. Each control input $u^{[i]}_i$, $i \in \{1, \ldots, n\}$, takes values in a compact input space U. Clearly, it would be possible to consider a motion model with time-varying orientation and we refer the reader to Exercise E3.1 where we do so.

Sensing model: At each discrete time instant, robot i activates a sensor that detects the presence and returns a measurement about the relative position of any object (robots or environment boundary) inside a given "sensor footprint." We describe the model in two steps. First, each robot measures other robots' positions and the environment as follows:

Sensing other robots' positions: there exists a set $\mathbb{A}_{\text{rbt}}$ containing the `null` element, called the *sensing alphabet*, and a map rbt-sns : $\mathbb{R}^d \to \mathbb{A}_{\text{rbt}}$,

called the *sensing function*, with the interpretation that robot i acquires the symbol rbt-sns$(p_i^{[j]}) \in \mathbb{A}_{\text{rbt}}$ for each robot $j \in \{1, \dots, n\} \setminus \{i\}$.

Sensing the environment: there exists a set $\mathbb{A}_{\text{env}}$ containing the `null` element, called the *environment sensing alphabet*, and a map env-sns : $\mathbb{P}(\mathbb{R}^d) \to \mathbb{A}_{\text{env}}$, called the *environment sensing function*, with the interpretation that robot i acquires the symbol env-sns$(Q_i) \in \mathbb{A}_{\text{env}}$.

Second, we let $S^{[i]} \subset \mathbb{R}^d$ be the *sensor footprint* of robot i and $S_i^{[i]}$ be its expression in the frame $\Sigma^{[i]}$. For simplicity, we assume that all robot sensors are equal, so that we write $S_i^{[i]} = S$. We require both sensing functions to provide no information about robots and boundaries that are outside S in the following two meanings: (i) if p is any point outside S, then rbt-sns$(p) =$ `null`, and (ii) if W is any subset of $\mathbb{R}^d$, env-sns$(W) =$ env-sns$(W \cap S)$.

We summarize this discussion with the following definition.

Definition 3.14 (Network with relative sensing). The physical components of a *network with relative sensing* consist of n mobile robots with identifiers $\{1, \dots, n\}$, with configurations in $Q \times \mathrm{SO}(d)$, for an allowable environment $Q \subset \mathbb{R}^d$, with dynamics described by equation (3.5), and with relative sensors described by the sensor footprint S, sensing alphabets $\mathbb{A}_{\text{rbt}}$ and $\mathbb{A}_{\text{env}}$, and sensing functions rbt-sns and env-sns.

To make things concrete, let us present two examples of robotic networks with relative sensing that are analogs of the "communication-based" robotic networks $\mathcal{S}_{\text{disk}}$ and $\mathcal{S}_{\text{vis-disk}}$ in Examples 3.4 and 3.6.

Example 3.15 (Disk sensor and corresponding relative-sensing network). Given a *sensing range* $r \in \mathbb{R}_{>0}$, the *disk sensor* has sensor footprint $\overline{B}(\mathbf{0}_d, r)$, i.e., a disk sensor measures any object (robot and environment boundary) within distance r. Regarding sensing of other robots, we assume that the alphabet is $\mathbb{A}_{\text{rbt}} = \mathbb{R}^d \cup \{\texttt{null}\}$ and that the sensing function is rbt-sns$(p_i^{[j]}) = p_i^{[j]}$ for each robot $j \in \{1, \dots, n\} \setminus \{i\}$, inside the sensor footprint $\overline{B}(\mathbf{0}_d, r)$, and rbt-sns$(p_i^{[j]}) =$ `null`, otherwise. Regarding sensing of the environment, we assume that the alphabet is $\mathbb{A}_{\text{env}} = \mathbb{P}(\mathbb{R}^d)$ and that the sensing function is env-sns$(Q_i) = Q_i \cap \overline{B}(\mathbf{0}_d, r)$. A group of robots with disk sensors defines the robotic network with relative sensing $\mathcal{S}_{\text{disk}}^{\text{rs}}$. •

Example 3.16 (Range-limited visibility sensor and corresponding relative-sensing network). Given a *sensing range* $r \in \mathbb{R}_{>0}$, the *range-limited visibility sensor* has sensor footprint $\overline{B}(\mathbf{0}_d, r)$ and performs measurements only of objects within unobstructed line of sight. Regarding sensing of other robots, we assume that the alphabet is $\mathbb{A}_{\text{rbt}} = \mathbb{R}^d \cup \{\texttt{null}\}$ and that the sensing function is rbt-sns$(p_i^{[j]}) = p_i^{[j]}$ for each robot $j \in \{1, \dots, n\} \setminus \{i\}$, inside the range-limited visibility set $\mathrm{Vi}_{\text{disk}}(\mathbf{0}_2; Q_i)$, and rbt-sns$(p_i^{[j]}) =$ `null`, otherwise. Regarding sensing of the environment, we assume[1] that the alphabet

[1] It would be equivalent to assume that the robot can sense every portion of ∂Q that is within distance r and that is visible from the robot position.

is $\mathbb{A}_{\text{env}} = \mathbb{P}(\mathbb{R}^d)$ and that the environment sensor measures the range-limited visibility set $\text{Vi}_{\text{disk}}(p^{[i]}_{\text{fixed}}; Q)$ expressed with respect to the frame $\Sigma^{[i]}$; see Section 2.1.2 for the definition of range-limited visibility set. In other words, the environment sensing function is $\text{env-sns}(Q_i) = \text{Vi}_{\text{disk}}(\mathbf{0}_2; Q_i)$. This is illustrated in Figure 3.6. A group of robots with range-limited visibility sensors

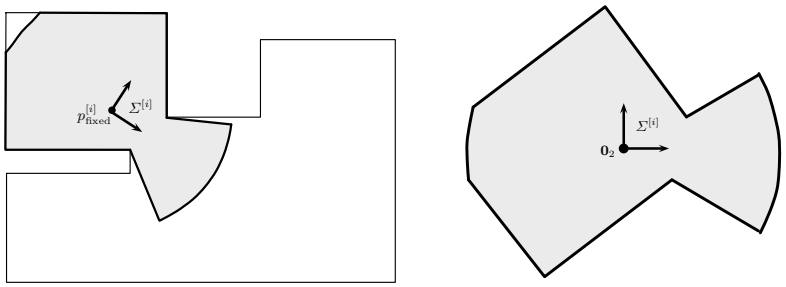


Fig. 3.6. The left plot depicts the range-limited visibility set $\text{Vi}_{\text{disk}}(p^{[i]}_{\text{fixed}}; Q)$ expressed with respect to an inertially fixed frame. The right plot depicts the range-limited visibility set expressed with respect to the body-fixed frame $\Sigma^{[i]}$, that is, $\text{Vi}_{\text{disk}}(\mathbf{0}_2; Q_i)$.

defines the robotic network with relative sensing $\mathcal{S}^{\text{rs}}_{\text{vis-disk}}$. •

Remark 3.17 (Sensing model consequences). The proposed sensing model has the following two consequences:

(i) robots have no information about the absolute position and orientation of themselves, the other robots or any part of the environment; and
(ii) the relative sensing capacity of the robots gives rise to a proximity graph, called the *sensing graph*, whose edges are the collection of robot pairs that are within sensing range. For example, in the network $\mathcal{S}^{\text{rs}}_{\text{disk}}$, the sensing graph is the disk graph $\mathcal{G}_{\text{disk}}(r)$. In general, sensing graphs are directed.•

3.2.3 Relative-sensing control laws

As we did for robotic networks with interactions based on communication, we define here control laws based on relative sensing and describe the closed-loop evolution of robotic networks with relative sensing.

First, we consider a robotic network with relative sensing $\mathcal{S}^{\text{rs}}$ characterized by: identifiers $\{1, \dots, n\}$, configurations in $Q \times \text{SO}(d)$, for an allowable environment $Q \subset \mathbb{R}^d$, dynamics described by equation (3.5), and relative sensors described by the sensor footprint S, sensing alphabets $\mathbb{A}_{\text{rbt}}$ and $\mathbb{A}_{\text{env}}$, and sensing functions rbt-sns and env-sns. A *relative-sensing control law* $\mathcal{RSC}$ for the robotic network with relative sensing $\mathcal{S}^{\text{rs}}$ consists of the tuple:

(i) W, called the *processor state set*, with corresponding set of *allowable initial values* $W_0 \subseteq W$;
(ii) $\mathrm{stf} : W \times \mathbb{A}_{\mathrm{rbt}}^n \times \mathbb{A}_{\mathrm{env}} \to W$, called the *(processor) state-transition function*; and
(iii) $\mathrm{ctl} : W \times \mathbb{A}_{\mathrm{rbt}}^n \times \mathbb{A}_{\mathrm{env}} \to U$, called the *(motion) control function*.

As for robotic networks, we say that $\mathcal{RSC}$ is *static* if W is a singleton for all $i \in \{1, \dots, n\}$; in this case $\mathcal{RSC}$ can be described by a motion control function $\mathrm{ctl} : \mathbb{A}_{\mathrm{rbt}}^n \times \mathbb{A}_{\mathrm{env}} \to U$. Additionally, if the environment $Q = \mathbb{R}^d$, then $\mathcal{RSC}$ can be described by a motion control function $\mathrm{ctl} : W \times \mathbb{A}_{\mathrm{rbt}}^n \to U$.

Second, the *evolution* of $(\mathcal{S}^{\mathrm{rs}}, \mathcal{RSC})$ from initial conditions $(p_0^{[i]}, R_{\mathrm{fixed}}^{[i]}) \in \mathbb{R}^d \times \mathrm{SO}(d)$ and $w_0^{[i]} \in W_0$, $i \in \{1, \dots, n\}$, is the collection of curves $p_{\mathrm{fixed}}^{[i]} : \mathbb{R}_{\geq 0} \to \mathbb{R}^d$ and $w^{[i]} : \mathbb{Z}_{\geq 0} \to W$, $i \in \{1, \dots, n\}$, defined by

$$\begin{aligned} \dot{p}_{\mathrm{fixed}}^{[i]}(t) &= R_{\mathrm{fixed}}^{[i]} \, \mathrm{ctl}\big(w^{[i]}(\lfloor t \rfloor), y^{[i]}(\lfloor t \rfloor), y_{\mathrm{env}}^{[i]}(\lfloor t \rfloor)\big), \\ w^{[i]}(\ell) &= \mathrm{stf}(w^{[i]}(\ell - 1), y^{[i]}(\ell), y_{\mathrm{env}}^{[i]}(\ell)), \end{aligned}$$

with $p_{\mathrm{fixed}}^{[i]}(0) = p_0^{[i]}$, and $w^{[i]}(-1) = w_0^{[i]}$, $i \in \{1, \dots, n\}$. In the previous equations, $y^{[i]} : \mathbb{Z}_{\geq 0} \to \mathbb{A}_{\mathrm{rbt}}^n$ (describing the robot measurements taken by sensor i) with components $y_j^{[i]}(\ell)$, for $j \in \{1, \dots, n\}$, and $y_{\mathrm{env}}^{[i]} : \mathbb{Z}_{\geq 0} \to \mathbb{A}_{\mathrm{env}}$ (describing the environment measurements taken by sensor i) are defined by

$$y_j^{[i]}(\ell) = \text{rbt-sns}(p_i^{[j]}(\ell)), \quad y_{\mathrm{env}}^{[i]}(\ell) = \text{env-sns}(Q_i(\ell)).$$

In the last equation, $p_i^{[j]}$ and $Q_i(\ell)$ denote the position of the j-th robot and the environment Q as expressed with respect to the moving frame $\Sigma^{[i]}$.

3.2.4 Equivalence between communication and relative-sensing laws

Consider a "communication-based" robotic network $\mathcal{S}_1$ with control and communication law $\mathcal{CC}_1$ with the following properties:

(i) regarding $\mathcal{S}_1$: the network is uniform, the state space is $X = \mathbb{R}^d$ with states denoted by $x^{[i]} = p^{[i]}$, the communication graph is the r-disk graph, and the robot dynamics is $\dot{p}^{[i]} = u^{[i]}$; and
(ii) regarding $\mathcal{CC}_1$: the control and communication law is uniform and data-sampled, the communication alphabet is $\mathbb{A} = \mathbb{R}^d \cup \{\texttt{null}\}$, and the message-generation function is $\mathrm{msg}(p, w, j) = p$.

Given a network and a law $(\mathcal{S}_1, \mathcal{CC}_1)$ satisfying (i) and (ii), the control and communication law $\mathcal{CC}_1$ is *invariant* if its state transition and control maps satisfy, for all $p \in \mathbb{R}^d$, $w \in W$, $y \in \mathbb{A}^n$ and $R \in \mathrm{SO}(d)$,

$$\begin{aligned} \mathrm{stf}(p, w, y) &= \mathrm{stf}\big(\mathbf{0}_d, w, R(y - p)\big), \\ \mathrm{ctl}(p, w, y) &= R^T \mathrm{ctl}\big(\mathbf{0}_d, w, R(y - p)\big), \end{aligned}$$

where the ith component of $R(y-p) \in \mathbb{A}^n$ is $R(y_i - p)$ if $y_i \in \mathbb{R}^d$, or $\texttt{null}$ if $y_i = \texttt{null}$.

Next, consider a relative-sensing network $\mathcal{S}_2$ with disk sensors as in Example 3.15 that is, assume the sensing footprint is $\overline{B}(\mathbf{0}_d, r)$, the sensing alphabet is $\mathbb{A}_{\text{rbt}} = \mathbb{R}^d \cup \{\texttt{null}\}$ and the sensing function equals the identity function in $\overline{B}(\mathbf{0}_d, r)$. We assume no environment sensing as we set $Q = \mathbb{R}^d$. The communication and control law $\mathcal{CC}_1$ and the relative-sensing control law $\mathcal{RSC}_2$ for network $\mathcal{S}_2$ are *equivalent* if their processor state sets identical, e.g., denote both by W, and their state transition and control maps satisfy, for all $w \in W$ and $y \in \mathbb{R}^d \cup \{\texttt{null}\} = \mathbb{A}^n = \mathbb{A}^n_{\text{rbt}}$,

$$\text{stf}_1(\mathbf{0}_d, w, y) = \text{stf}_2(w, y), \qquad \text{and} \qquad \text{ctl}_1(\mathbf{0}_d, w, y) = \text{ctl}_2(w, y).$$

Proposition 3.18 (Evolution equivalence). *If $\mathcal{CC}_1$ is invariant and if $\mathcal{CC}_1$ and $\mathcal{RSC}_2$ are equivalent, then the evolutions of $(\mathcal{S}_1, \mathcal{CC}_1)$ and $(\mathcal{S}_2, \mathcal{RSC}_2)$ from identical initial conditions are identical.*

Proof. Assume that the messages and measurements array $y^{[i]}(t)$ received by the i-th robot at time t in the communication-based network and in the relative-sensing networks are equal to, respectively:

$$p^{[j_1]}_{\text{fixed}}, \dots, p^{[j_k]}_{\text{fixed}}, \qquad \text{and} \qquad p^{[j_1]}_i, \dots, p^{[j_k]}_i.$$

Then, the evolution of the communication-based network and of the relative-sensing networks are written as, respectively:

$$\begin{aligned} \dot{p}^{[i]}_{\text{fixed}} &= \text{ctl}_1(p^{[i]}_{\text{fixed}}, w^{[i]}, p^{[j_1]}_{\text{fixed}}, \dots, p^{[j_k]}_{\text{fixed}}), \\ \dot{p}^{[i]}_{\text{fixed}} &= R^{[i]}_{\text{fixed}} \text{ctl}_2(w^{[i]}, p^{[j_1]}_i, \dots, p^{[j_k]}_i). \end{aligned}$$

From equation (3.4) we know that, for all $j \in \{j_1, \dots, j_k\}$,

$$p^{[j]}_{\text{fixed}} = R^{[i]}_{\text{fixed}} p^{[j]}_i + p^{[i]}_{\text{fixed}} \qquad \Longrightarrow \qquad p^{[j]}_i = (R^{[i]}_{\text{fixed}})^T (p^{[j]}_{\text{fixed}} - p^{[i]}_{\text{fixed}}).$$

From this equality and from the fact that $\mathcal{CC}_1$ is invariant, we observe that

$$\text{ctl}_1(p^{[i]}_{\text{fixed}}, w^{[i]}, p^{[j_1]}_{\text{fixed}}, \dots, p^{[j_k]}_{\text{fixed}}) = R^{[i]}_{\text{fixed}} \text{ctl}_1(\mathbf{0}_d, w^{[i]}, p^{[j_1]}_i, \dots, p^{[j_k]}_i).$$

Since $\mathcal{CC}_1$ and $\mathcal{RSC}_2$ are equivalent, the two evolution equations coincide. A similar reasoning shows that also the evolutions of the processor states are identical. ■

Remark 3.19 (Communication-based laws on relative-sensing networks). Proposition 3.18 implies the following fact. Given an invariant control and communication law for a robotic network satisfying all appropriate properties, the control and communication law can be implemented on an appropriate relative-sensing network as a relative-sensing control law. •

3.3 Coordination tasks and complexity notions

In this section we introduce concepts and tools useful to analyze a communication and control law in a robotic network; our treatment is directly generalized to relative-sensing networks. We address the following questions: What is a coordination task for a robotic network? When does a control and communication law achieve a task? And with what time, space, and communication complexity?

3.3.1 Coordination tasks

Our first analysis step is to characterize the correctness properties of a communication and control law. We do so by defining the notion of task and of task achievement by a robotic network.

Definition 3.20 (Coordination task). Let $\mathcal{S}$ be a robotic network and let $\mathcal{W}$ be a set.

(i) A *coordination task* for $\mathcal{S}$ is a map $\mathcal{T}\colon \prod_{i\in I} X^{[i]} \times \mathcal{W}^n \to \{\texttt{true}, \texttt{false}\}$.
(ii) If $\mathcal{W}$ is a singleton, then the coordination task is said to be *static* and can be described by a map $\mathcal{T}\colon \prod_{i\in I} X^{[i]} \to \{\texttt{true}, \texttt{false}\}$.

Additionally, let $\mathcal{CC}$ a control and communication law for $\mathcal{S}$.

(i) The law $\mathcal{CC}$ is *compatible* with the task $\mathcal{T}\colon \prod_{i\in I} X^{[i]} \times \mathcal{W}^n \to \{\texttt{true}, \texttt{false}\}$ if its processor state take values in $\mathcal{W}$, that is, if $W^{[i]} = \mathcal{W}$, for all $i \in I$.
(ii) The law $\mathcal{CC}$ *achieves* the task $\mathcal{T}$ if it is compatible with it and if, for all initial conditions $x_0^{[i]} \in X_0^{[i]}$ and $w_0^{[i]} \in W_0^{[i]}$, $i \in I$, there exists $T \in \mathbb{R}_{>0}$ such that the network evolution $t \mapsto (x(t), w(t))$ has the property that $\mathcal{T}(x(t), w(t)) = \texttt{true}$ for all $t \geq T$. •

Remark 3.21 (Temporal logic). Loosely speaking, achieving a task means obtaining and maintaining a specified pattern in the robot physical or processor state. In other words, the task is achieved if *at some time* and *for all subsequent times* the predicate evaluates to true along system trajectories. It is possible to consider more general tasks based on more expressive predicates on trajectories. Such predicates can be defined through various forms of temporal and propositional logic, e.g., see Emerson [1994]. In particular, (linear) temporal logic contains certain constructs that allow reasoning in terms of time and is hence appropriate for robotic applications, as argued for example in Fainekos et al. [2005]. Network tasks such as periodically visiting a desired set of configurations can be encoded with temporal logic statements. •

Example 3.22 (Direction agreement and equidistance tasks). From Example 3.7, consider the uniform network $\mathcal{S}_{\text{circle}}$ of locally-connected first-order agents in $\mathbb{S}^1$. From Section 3.1.3, recall the AGREE & PURSUE control and communication law $\mathcal{CC}_{\text{AGREE \& PURSUE}}$ with processor state taking values in

$W = \{\texttt{cc}, \texttt{c}\} \times I$. There are two tasks of interest. First, we define the *direction agreement task* $\mathcal{T}_{\texttt{dir}} : (\mathbb{S}^1)^n \times W^n \to \{\texttt{true}, \texttt{false}\}$ by

$$\mathcal{T}_{\texttt{dir}}(\theta, w) = \begin{cases} \texttt{true}, & \text{if } \texttt{dir}^{[1]} = \cdots = \texttt{dir}^{[n]}, \\ \texttt{false}, & \text{otherwise}, \end{cases}$$

where $\theta = (\theta^{[1]}, \ldots, \theta^{[n]})$, $w = (w^{[1]}, \ldots, w^{[n]})$, and $w^{[i]} = (\texttt{dir}^{[i]}, \texttt{max-id}^{[i]})$, for $i \in I$. Furthermore, for $\varepsilon > 0$, we define the static *equidistance task* $\mathcal{T}_{\varepsilon\text{-eqdstnc}} : (\mathbb{S}^1)^n \to \{\texttt{true}, \texttt{false}\}$ to be `true` if and only if

$$\left| \min_{j \neq i} \text{dist}_{\texttt{c}}(\theta^{[i]}, \theta^{[j]}) - \min_{j \neq i} \text{dist}_{\texttt{cc}}(\theta^{[i]}, \theta^{[j]}) \right| < \varepsilon, \quad \text{for all } i \in I.$$

In other words, $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$ is true when, for every agent, the distance to the closest clockwise neighbor and to the closest counterclockwise neighbor are approximately equal. •

3.3.2 Complexity notions

We are finally ready to define the key notions of time, space and communication complexity. These notions describe the cost that a certain control and communication law incurs while completing a certain coordination task. We also define the complexity of a task to be the infimum of the costs incurred by all laws that achieve that task. We begin with a remark highlighting a difference between what follows and the complexity treatment for synchronous networks.

Remark 3.23 (Termination via task completion). As discussed in Remark 1.39 in Section 1.4, it is possible to consider various algorithm termination notions. Here we will establish the completion of an algorithm as the instant when a given task is achieved. •

First, we define the time complexity of an achievable task as the minimum number of communication rounds needed by the agents to achieve the task $\mathcal{T}$.

Definition 3.24 (Time complexity). Let $\mathcal{S}$ be a robotic network and let $\mathcal{T}$ be a coordination task for $\mathcal{S}$. Let $\mathcal{CC}$ be a control and communication law for $\mathcal{S}$ compatible with $\mathcal{T}$.

(i) The *(worst-case) time complexity to achieve $\mathcal{T}$ with $\mathcal{CC}$ from* $(x_0, w_0) \in \prod_{i \in I} X_0^{[i]} \times \prod_{i \in I} W_0^{[i]}$ is

$$\text{TC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) = \inf \{\ell \mid \mathcal{T}(x(k), w(k)) = \texttt{true}\,, \text{ for all } k \geq \ell\}\,,$$

where $t \mapsto (x(t), w(t))$ is the evolution of $(\mathcal{S}, \mathcal{CC})$ from the initial condition (x_0, w_0);

(ii) The *(worst-case) time complexity to achieve $\mathcal{T}$ with $\mathcal{CC}$* is

$$\mathrm{TC}(\mathcal{T},\mathcal{CC}) = \sup\Big\{ \mathrm{TC}(\mathcal{T},\mathcal{CC},x_0,w_0) \mid (x_0,w_0) \in \prod_{i\in I} X_0^{[i]} \times \prod_{i\in I} W_0^{[i]} \Big\}.$$

(iii) The *(worst-case) time complexity of $\mathcal{T}$* is

$$\mathrm{TC}(\mathcal{T}) = \inf\{\mathrm{TC}(\mathcal{T},\mathcal{CC}) \mid \mathcal{CC} \text{ compatible with } \mathcal{T}\}. \qquad \bullet$$

Next, we quantify memory and communication requirements of communication and control laws. We assume 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." We let $|W|_{\text{basic}}$ and $|\mathbb{A}|_{\text{basic}}$ denote the number of basic memory units and basic messages required to represent elements of W and $\mathbb{A}$, respectively. The `null` message has zero cost. To clarify this assumption, we adopt two conventions. First, as in Section 1.4.2, we assume that a "basic memory unit" or a "basic message" contains $\log(n)$ bits. This implies that the $\log(n)$ bits required to store or transmit a robot identifier $i \in \{1,\dots,n\}$ are equivalent to one "basic memory unit." Second, as mentioned in Remark 3.11, we assume that a processor can store and transmit a (finite number of) integer and real numbers and we adopt the convention that any such number is quantized and represented by a constant number of basic memory units or basic messages.

We now quantify memory requirements of algorithms and tasks by counting the required number of basic memory units. Let the network $\mathcal{S}$, the task $\mathcal{T}$ and the control and communication law $\mathcal{CC}$ be as in Definition 3.24.

Definition 3.25 (Space complexity).

(i) The *(worst-case) space complexity to achieve $\mathcal{T}$ with $\mathcal{CC}$*, denoted by $\mathrm{SC}(\mathcal{T},\mathcal{CC})$, is the maximum number of basic memory units required by a robot processor executing the $\mathcal{CC}$ on $\mathcal{S}$ among all robots and among all allowable initial physical and processor states until termination.

(ii) The space complexity of $\mathcal{T}$ is the infimum among the space complexities of all control and communication laws that achieve $\mathcal{T}$. $\bullet$

The set of all non-null messages generated during one communication round from network state (x,w) is denoted by

$$\mathcal{M}(x,w) = \{(i,j) \in E_{\text{cmm}}(x) \mid \mathrm{msg}^{[i]}(x^{[i]},w^{[i]},j) \neq \texttt{null}\}.$$

We now quantify mean and total communication requirements of algorithms and tasks by counting the number of transmitted basic messages.

Definition 3.26 (Mean and Total Communication complexity).

(i) The *(worst-case) mean communication complexity* and the *(worst-case) total communication complexity* to achieve $\mathcal{T}$ with $\mathcal{CC}$ from $(x_0,w_0) \in \prod_{i\in I} X_0^{[i]} \times \prod_{i\in I} W_0^{[i]}$ are, respectively,

$$\mathrm{MCC}(\mathcal{T},\mathcal{CC},x_0,w_0) = \frac{|\mathbb{A}|_{\mathrm{basic}}}{\tau}\sum_{\ell=0}^{\tau-1}|\mathcal{M}(x(\ell),w(\ell))|,$$

$$\mathrm{TCC}(\mathcal{T},\mathcal{CC},x_0,w_0) = |\mathbb{A}|_{\mathrm{basic}}\sum_{\ell=0}^{\tau-1}|\mathcal{M}(x(\ell),w(\ell))|,$$

where $t \mapsto (x(t), w(t))$ is the evolution of $(\mathcal{S},\mathcal{CC})$ from the initial condition (x_0, w_0) and where $\tau = \mathrm{TC}(\mathcal{CC},\mathcal{T},x_0,w_0)$. Here, MCC is defined only for initial conditions (x_0, w_0) with the property that $\mathcal{T}(x_0, w_0) = \texttt{false}$.

(ii) The *(worst-case) mean communication complexity* (resp. the *(worst-case) total communication complexity*) to achieve $\mathcal{T}$ with $\mathcal{CC}$ is the supremum of $\mathrm{MCC}(\mathcal{T},\mathcal{CC},x_0,w_0)$ (resp. $\mathrm{TCC}(\mathcal{T},\mathcal{CC},x_0,w_0)$) over all allowable initial states (x_0, w_0).

(iii) The *(worst-case) mean communication complexity* (resp. the *(worst-case) total communication complexity*) of $\mathcal{T}$ is the infimum among the mean communication complexity (resp. the total communication complexity) of all control and communication laws that achieve $\mathcal{T}$. •

By construction, one can verify that it always happens that

$$\mathrm{TCC}(\mathcal{T},\mathcal{CC}) \leq \mathrm{MCC}(\mathcal{T},\mathcal{CC})\cdot\mathrm{TC}(\mathcal{T},\mathcal{CC}). \tag{3.6}$$

We conclude this section with possible variations and extensions of the complexity definitions.

Remark 3.27 (Infinite-horizon mean communication complexity). The mean communication complexity MCC measures the average cost of the communication rounds required to achieve a task over a finite time horizon; a similar statement holds for the total communication complexity TCC. One might be interested in a notion of mean communication complexity required to maintain true the task for all times. Accordingly, the infinite-horizon mean communication complexity of $\mathcal{CC}$ from initial conditions (x_0, w_0) is

$$\text{IH-MCC}(\mathcal{CC},x_0,w_0) = \lim_{\tau\to+\infty}\frac{|\mathbb{A}|_{\mathrm{basic}}}{\tau}\sum_{\ell=0}^{\tau}|\mathcal{M}(x(\ell),w(\ell))|. \qquad \bullet$$

Remark 3.28 (Communication complexity in omnidirectional networks). In omnidirectional wireless networks the standard operation mode is for all neighbors of a node to receive the signal it transmits. In other words, the transmission is omnidirectional rather than unidirectional. It is straightforward to require the message generation function to have the property that the output it generates be independent of the intended receiver. Under such assumptions, it make sense to count as communication complexity not the number of messages transmitted in the network, but the number of transmissions, i.e., a unit cost per node rather than a unit cost per edge of the network. •

Remark 3.29 (Energy complexity). Given a model for the energy consumed by the robot to move and to transmit a message, one can easily define a notion of energy complexity for a control and communication law. In modern wireless transmitters, the energy consumptions in transmitting a signal at a distance r varies with a power of r. Analogously, energy consumption is an increasing function of distance traveled. We consider this to be a promising avenue for further research. •

3.3.3 Invariance under rescheduling

Here we discuss the invariance properties of time and communication complexity under the *rescheduling* of a control and communication law. The idea behind rescheduling is to "spread" the execution of the law over time without affecting the trajectories described by the robots.

For simplicity we consider the setting of static laws; similar results can be obtained for the general setting. Also, for ease of presentation, we allow our communication and control laws to be time dependent, i.e., we consider message-generation functions and motion control functions of the form $\mathrm{msg}^{[i]} : \mathbb{Z}_{\geq 0} \times X^{[i]} \times I \to \mathbb{A}$ and $\mathrm{ctl}^{[i]} : \mathbb{R}_{\geq 0} \times X^{[i]} \times X^{[i]} \times \mathbb{A}^n \to U^{[i]}$, respectively. The Definition 3.10 of network evolution can be readily extended to this more general time-dependent setup.

Let $\mathcal{S} = (I, \mathcal{R}, E_{\mathrm{cmm}})$ be a robotic network where each mobile robot is a driftless control system, as defined in Section 1.2. Let $\mathcal{CC} = (\mathbb{A}, \{\mathrm{msg}^{[i]}\}_{i\in I}, \{\mathrm{ctl}^{[i]}\}_{i\in I})$ be a static control and communication law. In what follows, we define a new control and communication law by modifying $\mathcal{CC}$; to do so we introduce some notation. Let $s \in \mathbb{N}$, with $s \leq n$, and let $\mathcal{P}_I = \{I_0, \ldots, I_{s-1}\}$ be an *s-partition* of I, that is, $I_0, \ldots, I_{s-1} \subset I$ are disjoint and nonempty and $I = \cup_{k=0}^{s-1} I_k$. For $i \in I$, define the message-generation functions $\mathrm{msg}^{[i]}_{\mathcal{P}_I} : \mathbb{Z}_{\geq 0} \times X^{[i]} \times I \to \mathbb{A}$ by

$$\mathrm{msg}^{[i]}_{\mathcal{P}_I}(\ell, x, j) = \mathrm{msg}^{[i]}(\lfloor \ell/s \rfloor, x, j), \tag{3.7}$$

if $i \in I_k$ and $k = \ell \mod s$, and $\mathrm{msg}^{[i]}_{\mathcal{P}_I}(\ell, x, j) = \texttt{null}$ otherwise. According to this message-generation function, only the agents with unique identifier in I_k will send messages at time ℓ, where $\ell \in \{k + as\}_{a\in\mathbb{Z}_{\geq 0}}$. Equivalently, this can be stated as follows: according to (3.7), the messages originally sent at the time instant ℓ are now rescheduled to be sent at the time instants $F(\ell) - s + 1, \ldots, F(\ell)$, where $F : \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ is defined by $F(\ell) = s(\ell + 1) - 1$. Figure 3.7 illustrates this idea. For $i \in I$, define the control functions $\mathrm{ctl}^{[i]} : \mathbb{R}_{\geq 0} \times X^{[i]} \times X^{[i]} \times \mathbb{A}^n \to U^{[i]}$ by

$$\mathrm{ctl}^{[i]}_{\mathcal{P}_I}(t, x, x_{\mathrm{smpld}}, y) = \mathrm{ctl}^{[i]}\left(t - \ell + F^{-1}(\ell), x, x_{\mathrm{smpld}}, y\right), \tag{3.8}$$

if $t \in [\ell, \ell + 1]$ and $\ell = -1 \mod s$ and $\mathrm{ctl}^{[i]}_{\mathcal{P}_I}(t, x, x_{\mathrm{smpld}}, y) = 0$ otherwise. Here $F^{-1} : \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ is the inverse of F, defined by $F^{-1}(\ell) = \frac{\ell+1}{s} - 1$.

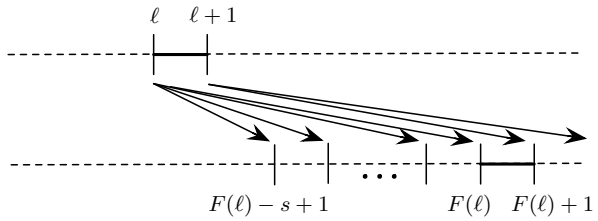


Fig. 3.7. Under the rescheduling, the messages that are sent at the time instant ℓ under the control and communication law $\mathcal{CC}$ are rescheduled to be sent over the time instants $F(\ell)-s+1,\dots,F(\ell)$ under the control and communication law $\mathcal{CC}_{(s,\mathcal{P}_I)}$.

Roughly speaking, the control law $\mathrm{ctl}^{[i]}_{\mathcal{P}_I}$ makes the agent i wait for the time intervals $[\ell,\ell+1]$, with $\ell\in\{as-1\}_{a\in\mathbb{N}}$, to execute any motion. Accordingly, the evolution of the robotic network under the original law $\mathcal{CC}$ during the time interval $[\ell,\ell+1]$ now takes place when all the corresponding messages have been transmitted, i.e., along the time interval $[F(\ell),F(\ell)+1]$. The following definition summarizes this construction.

Definition 3.30 (Rescheduling of control and communication laws). Let $\mathcal{S}=(I,\mathcal{R},E_{\mathrm{cmm}})$ be a robotic network with driftless physical agents, and let $\mathcal{CC}=(\mathbb{Z}_{\geq 0},\mathbb{A},\{\mathrm{msg}^{[i]}\}_{i\in I},\{\mathrm{ctl}^{[i]}\}_{i\in I})$ be a static control and communication law. Let $s\in\mathbb{N}$, with $s\leq n$, and let $\mathcal{P}_I$ be an s-partition of I. The control and communication law $\mathcal{CC}_{(s,\mathcal{P}_I)}=(\mathbb{Z}_{\geq 0},\mathbb{A},\{\mathrm{msg}^{[i]}_{\mathcal{P}_I}\}_{i\in I},\{\mathrm{ctl}^{[i]}_{\mathcal{P}_I}\}_{i\in I})$ defined by equations (3.7) and (3.8) is called a *$\mathcal{P}_I$-rescheduling of $\mathcal{CC}$*. •

The following result whose proof is presented in Section 3.6.1 shows that the total communication complexity of $\mathcal{CC}$ remains invariant under rescheduling.

Proposition 3.31 (Complexity of rescheduled laws). *With the assumptions of Definition 3.30, let $\mathcal{T}:\prod_{i\in I}X^{[i]}\to\{\mathtt{true},\mathtt{false}\}$ be a coordination task for $\mathcal{S}$. Then, for all $x_0\in\prod_{i\in I}X_0^{[i]}$,*

$$\mathrm{TC}(\mathcal{T},\mathcal{CC}_{(s,\mathcal{P}_I)},x_0)=s\cdot\mathrm{TC}(\mathcal{T},\mathcal{CC},x_0)\,.$$

Moreover, if $\mathrm{C}_{\mathrm{rnd}}$ is additive, then, for all $x_0\in\prod_{i\in I}X_0^{[i]}$

$$\mathrm{MCC}(\mathcal{T},\mathcal{CC}_{(s,\mathcal{P}_I)},x_0)=\frac{1}{s}\cdot\mathrm{MCC}(\mathcal{T},\mathcal{CC},x_0)\,,$$

and, therefore, $\mathrm{TCC}(\mathcal{T},\mathcal{CC}_{(s,\mathcal{P}_I)},x_0)=\mathrm{TCC}(\mathcal{T},\mathcal{CC},x_0)$, i.e., the total communication complexity of $\mathcal{CC}$ is invariant under rescheduling. •

Remark 3.32 (Appropriate complexity notions for driftless agents). Given the results in the previous theorem, one should be careful in choosing what notion of communication complexity to evaluate control and communication laws. For driftless physical agents, rather than the *mean* communication complexity MCC, one should really consider the *total* communication complexity TCC, since the latter is invariant with respect to rescheduling. Note that the notion of infinite-horizon mean communication complexity IH-MCC defined in Remark 3.27 satisfies the same relationship as MCC, that is, IH-MCC$(\mathcal{CC}_{(s,\mathcal{P}_I)}, x_0) = \frac{1}{s}$ IH-MCC$(\mathcal{CC}, x_0)$. •

3.4 Complexity of direction agreement and equidistance

From Example 3.7, Section 3.1.3, and Example 3.22, recall the definition of uniform network $\mathcal{S}_{\text{circle}}$ of locally-connected first-order agents in $\mathbb{S}^1$, the AGREE & PURSUE control and communication law $\mathcal{CC}_{\text{AGREE \& PURSUE}}$, and the two coordination tasks $\mathcal{T}_{\texttt{dir}}$ and $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$. In this section, we characterize the complexity to achieve these coordination tasks with $\mathcal{CC}_{\text{AGREE \& PURSUE}}$. Because the number of bits required to represent the variable $\texttt{max-id} \in \{1, \dots, n\}$ is $\log(n)$, Note that the space complexity of $\mathcal{CC}_{\text{AGREE \& PURSUE}}$ is $\log(n)$ bits, that is one basic memory unit in our convention discussed in Section 3.3.2.

Motivated by Remark 3.8, we model wireless communication congestion by assuming that the communication range is a monotone non-increasing function $r : \mathbb{N} \to]0, \pi[$ of the number of agents n. Likewise, we assume that the maximum control amplitude $u_{\max}$ is a non-increasing function $u_{\max} : \mathbb{N} \to]0, 1[$; recall that $u_{\max}$ is the maximum robot speed. Finally, it is convenient to define the function $n \mapsto \delta(n) = nr(n) - 2\pi \in \mathbb{R}$ that compares the sum of the communication ranges of all the robots with the length of the unit circle.

We are now ready to state the main result of this section; proofs are postponed to Section 3.6.2.

Theorem 3.33 (Time complexity of agree-and-pursue law). *For $k_{\text{prop}} \in]0, \frac{1}{2}[$, in the limit as $n \to +\infty$ and $\varepsilon \to 0^+$, the network $\mathcal{S}_{\text{circle}}$ with $u_{\max}(n) \geq k_{\text{prop}} r(n)$, the law $\mathcal{CC}_{\text{AGREE \& PURSUE}}$, and the tasks $\mathcal{T}_{\texttt{dir}}$ and $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$ together satisfy:*

(i) $\text{TC}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Theta(r(n)^{-1})$;
(ii) if $\delta(n)$ is lower bounded by a positive constant as $n \to +\infty$, then

$$\text{TC}(\mathcal{T}_{\varepsilon\text{-eqdstnc}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Omega(n^2 \log(n\varepsilon)^{-1}),$$
$$\text{TC}(\mathcal{T}_{\varepsilon\text{-eqdstnc}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in O(n^2 \log(n\varepsilon^{-1})).$$

If $\delta(n)$ is upper bounded by a negative constant, then $\mathcal{CC}_{\text{AGREE \& PURSUE}}$ does not achieve $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$ in general.

Next, we study the total communication complexity of the agree-and-pursue control and communication law. First, we note that any message in $\mathbb{A} = \mathbb{S}^1 \times \{\texttt{cc}, \texttt{c}\} \times \{1, \dots, n\} \cup \{\texttt{null}\}$ requires only a finite number of basic message to encode, that is, $|\mathbb{A}|_{\text{basic}} \in O(1)$.

Theorem 3.34 (Total communication complexity of agree-and-pursue law). *For $k_{\text{prop}} \in]0, \frac{1}{2}[$, in the limit as $n \to +\infty$ and $\varepsilon \to 0^+$, the network $\mathcal{S}_{\text{circle}}$ with $u_{\max}(n) \geq k_{\text{prop}} r(n)$, the law $\mathcal{CC}_{\text{AGREE \& PURSUE}}$, and the tasks $\mathcal{T}_{\texttt{dir}}$ and $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$ together satisfy:*

(i) if $\delta(n) \geq \pi(1/k_{\text{prop}} - 2)$ as $n \to +\infty$, then

$$\text{TCC}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Theta(n^2 r(n)^{-1}),$$

otherwise if $\delta(n) \leq \pi(1/k_{\text{prop}} - 2)$ as $n \to +\infty$, then

$$\text{TCC}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Omega(n^3 + nr(n)^{-1}),$$
$$\text{TCC}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in O(n^2 r(n)^{-1});$$

(ii) if $\delta(n)$ is lower bounded by a positive constant as $n \to +\infty$, then

$$\text{TCC}(\mathcal{T}_{\varepsilon\text{-eqdstnc}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Omega(n^3 \delta(n) \log(n\varepsilon)^{-1}),$$
$$\text{TCC}(\mathcal{T}_{\varepsilon\text{-eqdstnc}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in O(n^4 \log(n\varepsilon^{-1})).$$

Remark 3.35 (Comparison with leader election). Let us compare the agree-and-pursue control and communication law with the classical Le Lann-Chang-Roberts (LCR) algorithm for leader election discussed in Section 1.4.4. The leader election task consists of electing a unique agent among all agents in the network; it is therefore different from, but closely related to, the coordination task $\mathcal{T}_{\texttt{dir}}$. The LCR algorithm operates on a static network with the ring communication topology, and achieves leader election with time and total communication complexity, respectively, $\Theta(n)$ and $\Theta(n^2)$. The agree-and-pursue law operates on a robotic network with the $r(n)$-disk communication topology, and achieves $\mathcal{T}_{\texttt{dir}}$ with time and total communication complexity, respectively, $\Theta(r(n)^{-1})$ and $O(n^2 r(n)^{-1})$. If wireless communication congestion is modeled by $r(n)$ of order $1/n$ as in Remark 3.8, then the two algorithms have identical time complexity and the LCR algorithm has better communication complexity. Note that computations on a possibly disconnected, dynamic network are more complex than on a static ring topology. •

3.5 Notes

The literature on multi-robot systems is very extensive. Examples include the survey by Cao et al. [1997], the text by Arkin [1998] on behavior-based robotics, and the recent special issue [Arai et al., 2002] of the IEEE Transaction on Robotics and Automation. Together with this literature, the starting

point to develop the material of this chapter are the standard notions of *synchronous and asynchronous networks* in distributed [Lynch, 1997, Peleg, 2000, Tel, 2001] and parallel [Bertsekas and Tsitsiklis, 1997, Parhami, 1999] computation. This established body of knowledge on networks is, however, not applicable to the robotic network setting because of the agents' mobility and the ensuing dynamic communication topology.

An early influential contribution towards a network model of mobile interacting robots is the work by Suzuki and Yamashita [1999]. This model consists of a group of identical "distributed anonymous mobile robots" characterized as follows: no explicit communication takes place between them, and at each time instant of an "activation schedule," each robot senses the relative position of all other robots and moves according to a pre-specified algorithm. An artificial intelligence approach to multi-agent behavior in a shared environment is taken in [Moses and Tennenholtz, 1995]. Santoro [2001] provides, with an emphasis on computer science aspects, a brief survey of models, algorithms, and the need for appropriate complexity notions. Recently, a notion of communication complexity for control and communication algorithms in multi-robot systems is analyzed by Klavins [2003], see also [Klavins and Murray, 2004]. Notions of failures and robustness in robotic networks are discussed by Gupta et al. [2006b]. From a broad hybrid networked systems viewpoint, our robotic network model can be regarded as special cases of the general modeling paradigms discussed in [Lynch et al., 2003, Lygeros et al., 2003, Sanfelice et al., 2007].

A key feature of the synchronous robotic network model proposed in this chapter is the adoption of proximity graphs from computational geometry as a basis for our communication model. This design choice is justified by the vast wireless networking literature, where this assumption is made. The simplest communication model, in which two robots communicate only if they are within a fixed communication range, is a common model adopted studied, for example, in the works [Gupta and Kumar, 2000, Santi, 2005, Lloyd et al., 2005, Li, 2003]. These works study the proximity graph solutions to various communication optimization problems; this discipline is referred to as *topology control*, cf. Remark 3.8. Although we focus our presentation on the topological aspect of the communication service, more realistic communication models would include randomness, packet losses, coding, quantization, and delays, e.g., see [Toh, 2001, Goldsmith, 2005].

Next, we review some literature on emergent and self-organized swarming behaviors in biological groups. Interesting dynamical systems arise in biological networks at multiple levels of resolution, all the way from interactions among molecules and cells [Miller and Bassler, 2001] to the behavioral ecology of animal groups [Okubo, 1986]. Flocks of birds and schools of fish can travel in formation and act as one unit (see [Parrish et al., 2002]), allowing these animals to defend themselves against predators and protect their territories. Wildebeest and other animals exhibit complex collective behaviors when migrating, such as obstacle avoiding, leader election, and formation keeping

(see [Sinclair, 1977, Gueron and Levin, 1993]). Certain foraging behaviors include individual animals partitioning their environment into nonoverlapping zones (see [Barlow, 1974]). Honey bees [Seeley and Buhrman, 1999], gorillas [Stewart and Harcourt, 1994], and whitefaced capuchins [Boinski and Campbell, 1995] exhibit synchronized group activities such as initiation of motion and change of travel direction. These remarkable dynamic capabilities are achieved apparently without following a group leader; see [Okubo, 1986, Parrish et al., 2002, Gueron and Levin, 1993, Barlow, 1974, Seeley and Buhrman, 1999, Stewart and Harcourt, 1994, Boinski and Campbell, 1995] for specific examples of animal species and [Couzin et al., 2005, Conradt and Roper, 2003] for general studies. A comprehensive exposition of bio-inspired optimization and control methods is presented in [Passino, 2004].

With regards to distributed motion coordination algorithms, much progress has been made on collective pattern formation [Suzuki and Yamashita, 1999, Belta and Kumar, 2004, Justh and Krishnaprasad, 2004, Yang et al., 2008, Sepulchre et al., 2007], flocking [Tanner et al., 2007, Olfati-Saber, 2006, Lee and Spong, 2007, Moshtagh and Jadbabaie, 2007], motion feasibility of formations [Tabuada et al., 2005], formation control using rigidity and persistence theory [Olfati-Saber and Murray, 2002, Baillieul and Suri, 2003, Krick, 2007, Yu et al., 2008a, Hendrickx et al., 2007], formation stability [Tanner et al., 2004, Kang et al., 2006, Smith and Hadaegh, 2007, Zheng et al., 2008], motion camouflage [Justh and Krishnaprasad, 2006], self-assembly [Klavins et al., 2006], swarm aggregation [Gazi and Passino, 2003], gradient climbing [Ögren et al., 2004, Cortés, 2007], cyclic pursuit [Bruckstein et al., 1991, Marshall et al., 2004, Martínez and Bullo, 2006, Smith et al., 2005, Pavone and Frazzoli, 2007], vehicle routing [Lumelsky and Harinarayan, 1997, Sharma et al., 2007], motion planning with collision avoidance [Lumelsky and Harinarayan, 1997, Pallottino et al., 2007, Hu et al., 2007], and cooperative boundary estimation [Bertozzi et al., 2004, Clark and Fierro, 2005, Casbeer et al., 2006, Zhang and Leonard, 2005, Susca et al., 2008]. It is worth also mentioning works on network localization, estimation, and tracking, e.g., see [Barooah and Hespanha, 2007, Aspnes et al., 2006, Oh et al., 2007] and the references therein.

Much research has been devoted to distributed task allocation problems. The work in [Gerkey and Mataric, 2004] proposes a taxonomy of task allocation problems. In papers such as [Godwin et al., 2006, Alighanbari and How, 2006, Schumacher et al., 2003, Moore and Passino, 2007, Tang and Özgüner, 2005], advanced heuristic methods are developed, and their effectiveness is demonstrated through analysis, simulation or real world implementation. Distributed auction algorithms are discussed in [Castañón and Wu, 2003, Moore and Passino, 2007] building on the classic works in [Bertsekas and Castañón, 1991, 1993]. A distributed mixed-integer-linear-programming solver is proposed in [Alighanbari and How, 2006]. A spatially distributed receding-horizon scheme is proposed in [Frazzoli and Bullo, 2004, Pavone et al., 2007]. There has also been prior work on target assignment problems [Beard et al., 2002, Ar-

slan et al., 2007, Zavlanos and Pappas, 2007a, Smith and Bullo, 2007]. Target allocation for vehicles with nonholonomic constraints is studied in [Rathinam et al., 2007, Savla et al., 2008, 2007a].

3.6 Proofs

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

3.6.1 Proof of Proposition 3.31

Proof (Proposition 3.31). Let $t \mapsto x(t)$ and $t \mapsto \tilde{x}(t)$ denote the network evolutions starting from $x_0 \in \prod_{i\in I} X_0^{[i]}$ under $\mathcal{CC}$ and $\mathcal{CC}_{(s,\mathcal{P}_I)}$, respectively. From the definition of rescheduling, one can verify that, for all $k \in \mathbb{Z}_{\geq 0}$,

$$\tilde{x}^{[i]}(t) = \begin{cases} \tilde{x}^{[i]}(F(k-1)+1), & \text{for } t \in \bigcup_{\ell=F(k-1)+1}^{F(k)-1} [\ell, \ell+1], \\ x^{[i]}(t - F(k) + k), & \text{for } t \in [F(k), F(k)+1]. \end{cases} \tag{3.9}$$

By definition of $\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0)$, we have $\mathcal{T}(x(k)) = \texttt{true}$, for all $k \geq \mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0)$, and $\mathcal{T}(x(\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0) - 1)) = \texttt{false}$. Let us rewrite these equalities in terms of the trajectories of $\mathcal{CC}_{(s,\mathcal{P}_I)}$. From (3.9), one can write $x^{[i]}(k) = \tilde{x}^{[i]}(F(k))$, for all $i \in I$ and $k \in \mathbb{Z}_{\geq 0}$. Therefore, we have

$$\begin{aligned} &\mathcal{T}(\tilde{x}(F(k))) = \mathcal{T}(x(k)) = \texttt{true}\,, \qquad \text{for all } F(k) \geq F(\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0)), \\ &\mathcal{T}(\tilde{x}(F(\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0) - 1))) = \mathcal{T}(x(\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0) - 1)) = \texttt{false}, \end{aligned}$$

where we have used the rescheduled message-generation function in (3.7). Now, note that by equation (3.9), $\tilde{x}^{[i]}(\ell) = \tilde{x}^{[i]}(F(\lfloor \ell/s \rfloor - 1) + 1)$, for all $\ell \in \mathbb{Z}_{\geq 0}$ and all $i \in I$. Therefore, $\mathcal{T}(\tilde{x}(F(\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0) - 1) + 1)) = \mathcal{T}(\tilde{x}(F(\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0))))$ and we can rewrite the previous identities as

$$\begin{aligned} \mathcal{T}(\tilde{x}(k)) = \texttt{true}, \quad \text{for all } k \geq F(\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0) - 1) + 1, \\ \mathcal{T}(\tilde{x}(F(\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0) - 1))) = \texttt{false}, \end{aligned}$$

which imply that $\mathrm{TC}(\mathcal{T}, \mathcal{CC}_{(s,\mathcal{P}_I)}, x_0) = F(\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0) - 1) + 1 = s\,\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0)$. As for the mean communication complexity, additivity of $\mathrm{C}_{\mathrm{rnd}}$ implies

$$\begin{aligned} &\mathrm{C}_{\mathrm{rnd}} \circ \mathcal{M}(\ell, x(\ell)) \\ &= \mathrm{C}_{\mathrm{rnd}} \circ \mathcal{M}(F(\ell) - s + 1, \tilde{x}(F(\ell) - s + 1)) + \cdots + \mathrm{C}_{\mathrm{rnd}} \circ \mathcal{M}(F(\ell), \tilde{x}(F(\ell))), \end{aligned}$$

where we have used $F(\ell - 1) + 1 = F(\ell) - s + 1$. We conclude the proof by computing

$$\begin{aligned}\sum_{\ell=0}^{\mathrm{TC}(\mathcal{T},\mathcal{CC}_{(s,\mathcal{P}_I)},x_0)-1} & \mathrm{C}_{\mathrm{rnd}} \circ \mathcal{M}(\ell, \tilde{x}(\ell)) \\ &= \sum_{\ell=0}^{F(\mathrm{TC}(\mathcal{T},\mathcal{CC},x_0)-1)} \mathrm{C}_{\mathrm{rnd}} \circ \mathcal{M}(\ell, \tilde{x}(\ell)) \\ &= \sum_{\ell=0}^{\mathrm{TC}(\mathcal{T},\mathcal{CC},x_0)-1} \sum_{k=F(\ell)-s+1}^{F(\ell)} \mathrm{C}_{\mathrm{rnd}} \circ \mathcal{M}(k, \tilde{x}(k)) \\ &= \sum_{\ell=0}^{\mathrm{TC}(\mathcal{T},\mathcal{CC},x_0)-1} \mathrm{C}_{\mathrm{rnd}} \circ \mathcal{M}(\ell, x(\ell)). \end{aligned}$$ ■

3.6.2 Proof of Theorem 3.33

Proof (Theorem 3.33). In the following four *STEPS* we prove the two upper bounds and the two lower bounds.

STEP 1: We start by proving the upper bound in statement (i). We claim that $\mathrm{TC}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \leq 2\pi/(k_{\mathrm{prop}} r(n))$, and we reason by contradiction, i.e., we assume that there exists an initial condition which gives rise to an execution with time complexity strictly larger than $2\pi/(k_{\mathrm{prop}} r(n))$. Without loss of generality, assume $\texttt{dir}^{[n]}(0) = \texttt{c}$. For $\ell \leq 2\pi/(k_{\mathrm{prop}} r(n))$, define

$$k(\ell) = \mathrm{argmin}\{\mathrm{dist}_{\texttt{cc}}(\theta^{[n]}(0), \theta^{[i]}(\ell)) \mid \texttt{dir}^{[i]}(\ell) = \texttt{cc},\ i \in \{1, \dots, n\}\}.$$

In other words, agent $k(\ell)$ is the agent moving counterclockwise that has smallest counterclockwise distance from the initial position of agent n. Note that $k(\ell)$ is well-defined since, by hypothesis of contradiction, $\mathcal{T}_{\texttt{dir}}$ is $\texttt{false}$ for $\ell \leq 2\pi/(k_{\mathrm{prop}} r(n))$. According to the state-transition function of $\mathcal{CC}_{\text{AGREE \& PURSUE}}$ (cf. Section 3.1.3), messages with $\texttt{dir} = \texttt{cc}$ can only travel counterclockwise, while messages with $\texttt{dir} = \texttt{c}$ can only travel clockwise. Therefore, the position of agent $k(\ell)$ at time ℓ can only belong to the counterclockwise interval from the position of agent $k(0)$ at time 0 to the position of agent n at time 0.

Let us examine how fast the message from agent n travels clockwise. To this end, for $\ell \leq 2\pi/(k_{\mathrm{prop}} r(n))$, define

$$j(\ell) = \mathrm{argmax}\{\mathrm{dist}_{\texttt{c}}(\theta^{[n]}(0), \theta^{[i]}(\ell)) \mid \texttt{max-id}^{[i]}(\ell) = n,\ i \in \{1, \dots, n\}\}.$$

In other words, agent $j(\ell)$ has $\texttt{max-id}$ equal to n, is moving clockwise, and is the agent furthest from the initial position of agent n in the clockwise direction with these two properties. Initially, $j(0) = n$. Additionally, for $\ell \leq 2\pi/(k_{\mathrm{prop}} r(n))$, we claim that

$$\mathrm{dist}_{\texttt{c}}(\theta^{[j(\ell)]}(\ell), \theta^{[j(\ell+1)]}(\ell+1)) \geq k_{\mathrm{prop}} r(n).$$

This happens because either (1) there is no agent clockwise-ahead of $\theta^{[j(\ell)]}(\ell)$ within clockwise distance $r(n)$ and, therefore, the claim is obvious, or (2) there

are such agents. In case (2), let m denote the agent whose clockwise distance to agent $j(\ell)$ is maximal within the set of agents with clockwise distance $r(n)$ from $\theta^{[j(\ell)]}(\ell)$. Then,

$$\begin{aligned}
&\mathrm{dist}_{\mathtt{c}}(\theta^{[j(\ell)]}(\ell), \theta^{[j(\ell+1)]}(\ell+1)) \\
&\quad = \mathrm{dist}_{\mathtt{c}}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell+1)) \\
&\quad = \mathrm{dist}_{\mathtt{c}}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell)) + \mathrm{dist}_{\mathtt{c}}(\theta^{[m]}(\ell), \theta^{[m]}(\ell+1)) \\
&\quad \geq \mathrm{dist}_{\mathtt{c}}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell)) + k_{\mathrm{prop}}\big(r(n) - \mathrm{dist}_{\mathtt{c}}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell))\big) \\
&\quad = k_{\mathrm{prop}} r(n) + (1 - k_{\mathrm{prop}})\, \mathrm{dist}_{\mathtt{c}}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell)) \ \geq k_{\mathrm{prop}} r(n),
\end{aligned}$$

where the first inequality follows from the fact that at time ℓ there can be no agent whose clockwise distance to agent m is less than $(r(n) - \mathrm{dist}_{\mathtt{c}}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell)))$. Therefore, after $2\pi/(k_{\mathrm{prop}} r(n))$ communication rounds, the message with $\mathtt{max\text{-}id} = n$ has traveled the whole circle in the clockwise direction, and must therefore have reached agent $k(\ell)$. This is a contradiction.

STEP 2: We now prove the lower bound in statement (i). If $r(n) > \pi$ for all n, then $1/r(n) < 1/\pi$, and the upper bound reads $\mathrm{TC}(\mathcal{T}_{\mathtt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in O(1)$. Obviously, the time complexity of any evolution with an initial configuration where $\mathtt{dir}^{[i]}(0) = \mathtt{cc}$ for $i \in \{1, \ldots, n-1\}$, $\mathtt{dir}^{[n]}(0) = \mathtt{c}$ and $\mathcal{E}_{\mathcal{G}_{\mathrm{disk}}(r)}(\theta^{[1]}(0), \ldots, \theta^{[n]}(0))$ is the complete graph, is lower bounded by 1. Therefore, $\mathrm{TC}(\mathcal{T}_{\mathtt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Omega(1)$. If $r(n) > \pi$ for all n, then we conclude $\mathrm{TC}(\mathcal{T}_{\mathtt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Theta(r(n)^{-1})$. Assume now that $r(n) \leq \pi$ for sufficiently large n. Consider an initial configuration where $\mathtt{dir}^{[i]}(0) = \mathtt{cc}$ for $i \in \{1, \ldots, n-1\}$, $\mathtt{dir}^{[n]}(0) = \mathtt{c}$, and the agents are placed as depicted in Figure 3.8. Note that, after each communication round, agent 1 has moved

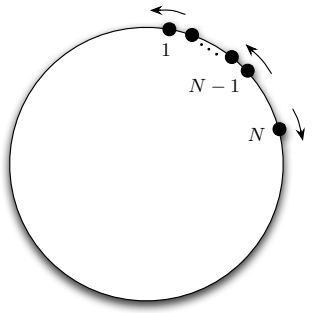


Fig. 3.8. Initial condition for the lower bound of $\mathrm{TC}(\mathcal{T}_{\mathtt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}})$, with $0 < \mathrm{dist}_{\mathtt{c}}(\theta^{[n-1]}(0), \theta^{[n]}(0)) - r(n) < \varepsilon$ and $\mathrm{dist}_{\mathtt{c}}(\theta^{[1]}(0), \theta^{[n-1]}(0)) \leq r(n) - \varepsilon$, for some fixed $\varepsilon > 0$.

$k_{\mathrm{prop}} r(n)$ in the counterclockwise direction, while agent n has moved $k_{\mathrm{prop}} r(n)$ in the clockwise direction. These two agents keep moving at full speed towards

each other until they become neighbors at a time lower bounded by

$$\frac{2\pi - r(n)}{2k_{\text{prop}}r(n)} > \frac{\pi}{k_{\text{prop}}r(n)} - 1.$$

We conclude $\text{TC}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Omega(r(n)^{-1})$.

STEP 3: We now prove the upper bound in (ii). We begin by noting that the lower bound on δ implies $r(n)^{-1} \in O(n)$. Therefore, $\text{TC}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}})$ belongs to $O(n)$ and is negligible as compared with the claimed upper bound estimates for $\text{TC}(\mathcal{T}_{\varepsilon\text{-eqdstnc}}, \mathcal{CC}_{\text{AGREE \& PURSUE}})$. In what follows, we therefore assume that $\mathcal{T}_{\texttt{dir}}$ has been achieved and that, without loss of generality, all agents are moving clockwise. We now prove a fact regarding connectivity. At time $\ell \in \mathbb{Z}_{\geq 0}$, let $H(\ell)$ be the union of all the empty "circular segments" of length at least $r(n)$, that is, let

$$H(\ell) = \{x \in \mathbb{S}^1 \mid \min_{i \in \{1,\dots,n\}} \text{dist}_{\texttt{c}}(x, \theta^{[i]}(\ell)) + \min_{j \in \{1,\dots,n\}} \text{dist}_{\texttt{cc}}(x, \theta^{[j]}(\ell)) > r(n)\}.$$

In other words, $H(\ell)$ does not contain any point between two agents separated by a distance less than $r(n)$, and each connected component of $H(\ell)$ has length at least $r(n)$. Let $n_H(\ell)$ be the number of connected components of $H(\ell)$, if $H(\ell)$ is empty, then we take the convention that $n_H(\ell) = 0$. Clearly, $n_H(\ell) \leq n$. We claim that, if $n_H(\ell) > 0$, then $\tau \mapsto n_H(\ell+\tau)$ is non-increasing. Let $d(\ell) < r(n)$ be the distance between any two consecutive agents at time ℓ. Because both agents move in the same direction, a simple calculation shows that

$$\begin{aligned} d(\ell+1) &\leq d(\ell) + k_{\text{prop}}(r - d(\ell)) = (1 - k_{\text{prop}})d(\ell) + k_{\text{prop}}r(n) \\ &< (1 - k_{\text{prop}})r + k_{\text{prop}}r(n) = r(n). \end{aligned}$$

This means that the two agents remain within distance $r(n)$ and, therefore connected, at the following time instant. Because the number of connected components of $\mathcal{E}_{\mathcal{G}_{\text{disk}(r)}}(\theta^{[1]}, \dots, \theta^{[n]})$ does not increase, it follows that the number of connected components of H cannot increase. Next, we claim that, if $n_H(\ell) > 0$, then there exists $\tau > \ell$ such that $n_H(\tau) < n_H(\ell)$. By contradiction, assume $n_H(\ell) = n_H(\tau)$ for all $\tau \geq \ell$. Without loss of generality, let $\{1, \dots, m\}$ be a set of agents with the properties that $\text{dist}_{\texttt{cc}}\left(\theta^{[i]}(\ell), \theta^{[i+1]}(\ell)\right) \leq r(n)$, for $i \in \{1, \dots, m\}$, that $\theta^{[1]}(\ell)$ and $\theta^{[m]}(\ell)$ belong to the boundary of $H(\ell)$, and that there is no other set with the same properties and more agents. (Note that this implies that the agents $1, \dots, m$ are in counterclockwise order.) One can show that, for $\tau \geq \ell$,

$$\begin{aligned} \theta^{[1]}(\tau+1) &= \theta^{[1]}(\tau) - k_{\text{prop}}r(n), \\ \theta^{[i]}(\tau+1) &= \theta^{[i]}(\tau) - k_{\text{prop}}\,\text{dist}_{\texttt{c}}(\theta^{[i]}(\tau), \theta^{[i-1]}(\tau)), \end{aligned}$$

for $i \in \{2, \dots, m\}$. If we consider the inter-agent distances

$$d(\tau) = \big(\operatorname{dist}_{\mathsf{cc}}(\theta^{[1]}(\tau), \theta^{[2]}(\tau)), \dots, \operatorname{dist}_{\mathsf{cc}}(\theta^{[m-1]}(\tau), \theta^{[m]}(\tau))\big) \in \mathbb{R}^{m-1}_{>0},$$

then the previous equations can be rewritten as

$$d(\tau+1) = \operatorname{Trid}_{m-1}(k_{\text{prop}}, 1-k_{\text{prop}}, 0)\, d(\tau) + r(n) k_{\text{prop}} \boldsymbol{e}_1,$$

where the linear map $(a,b,c) \mapsto \operatorname{Trid}_{m-1}(a,b,c) \in \mathbb{R}^{(m-1)\times(m-1)}$ is defined in Section 1.5.4. This is a discrete-time affine time-invariant dynamical system with unique equilibrium point $r(n)\mathbf{1}_{m-1}$. By construction, the initial condition of this system satisfies $\|d(0) - r(n)\mathbf{1}_{m-1}\|_2 \leq r(n)\sqrt{m-1}$. By Theorem 1.74(ii) in Section 1.5.4, for $\eta_1 \in]0,1[$, the solution $\tau \mapsto d(\tau)$ to this system reaches a ball of radius η_1 centered at the equilibrium point in time $O(m \log m + \log \eta_1^{-1})$. (Here we used the fact that the initial condition of this system is bounded.) In turn, this implies that $\tau \mapsto \sum_{i=1}^m d_i(\tau)$ is larger than $(m-1)(r(n)-\eta_1)$ in time $O(m \log m + \log \eta_1^{-1})$. We are now ready to find the contradiction and show that $n_H(\tau)$ cannot remain equal to $n_H(\ell)$ for all time τ. After time $O(m \log m + \log \eta_1^{-1}) = O(n \log n + \log \eta_1^{-1})$, we have:

$$\begin{aligned} 2\pi &\geq n_H(\ell) r(n) + \sum_{j=1}^{n_H(\ell)} (r(n) - \eta_1)(m_j - 1) \\ &= n_H(\ell) r(n) + (n - n_H(\ell))(r(n) - \eta_1) = n_H(\ell)\eta_1 + n(r(n) - \eta_1). \end{aligned}$$

Here $m_1, \dots, m_{n_H(\ell)}$ are the number of agents in each isolated group, and each connected component of $H(\ell)$ has length at least $r(n)$. Now, take $\eta_1 = (nr(n) - 2\pi)n^{-1} = \delta(n)n^{-1}$, and the contradiction follows from

$$\begin{aligned} 2\pi &\geq n_H(\ell)\eta_1 + nr(n) - n\eta_1 \\ &= n_H(\ell)\eta_1 + nr(n) + 2\pi - nr(n) = n_H(\ell)\eta_1 + 2\pi. \end{aligned}$$

In summary, this shows that the number of connected components of $H(\ell)$ decreases by one in time $O(n \log n + \log \eta_1^{-1}) = O(n \log n + \log(n\delta(n)^{-1}))$. Note that δ being lower bounded implies $n\delta(n)^{-1} = O(n)$ and, therefore, $O(n \log n + \log(n\delta(n)^{-1})) = O(n \log n)$. Iterating this argument n times, in time $O(n^2 \log n)$ the set H will become empty. At that time, the resulting network will obey the discrete-time linear time-invariant dynamical system:

$$d(\tau+1) = \operatorname{Circ}_n(k_{\text{prop}}, 1-k_{\text{prop}}, 0)\, d(\tau), \tag{3.10}$$

where the linear map $(a,b,c) \mapsto \operatorname{Circ}_n(a,b,c) \in \mathbb{R}^{n\times n}$ is defined in Section 1.5.4. Here $d(\tau) = \big(\operatorname{dist}_{\mathsf{cc}}(\theta^{[1]}(\tau), \theta^{[2]}(\tau)), \dots, \operatorname{dist}_{\mathsf{cc}}(\theta^{[n]}(\tau), \theta^{[n+1]}(\tau))\big) \in \mathbb{R}^n_{>0}$, with the convention $\theta^{[n+1]} = \theta^{[1]}$. By Theorem 1.74(iii) in Section 1.5.4, in time $O\big(n^2 \log \varepsilon^{-1}\big)$, the error 2-norm satisfies the contraction inequality $\|d(\tau) - d_*\|_2 \leq \varepsilon \|d(0) - d_*\|_2$, for $d_* = \frac{2\pi}{n}\mathbf{1}_n$. We convert this inequality on 2-norms into an appropriate inequality on ∞-norms as follows. Note that $\|d(0) - d_*\|_\infty = \max_{i\in\{1,\dots,n\}} |d^{[i]}(0) - d_*^{[i]}| \leq 2\pi$. For $\eta_2 \in]0,1[$ and for τ of order $n^2 \log \eta_2^{-1}$,

$$\|d(\tau) - d_*\|_\infty \leq \|d(\tau) - d_*\|_2 \leq \eta_2 \|d(0) - d_*\|_2 \\ \leq \eta_2 \sqrt{n} \|d(0) - d_*\|_\infty \leq \eta_2 2\pi\sqrt{n}.$$

This means that the desired configuration is achieved for $\eta_2 2\pi\sqrt{n} = \varepsilon$, that is, in time $O(n^2 \log \eta_2^{-1}) = O(n^2 \log(n\varepsilon^{-1}))$. In summary, the equidistance task is achieved in time $O(n^2 \log(n\varepsilon^{-1}))$.

STEP 4: Finally, we prove the lower bound in (ii). As we reasoned before, $\mathrm{TC}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}})$ is negligible as compared with the claimed lower bound estimate for $\mathrm{TC}(\mathcal{T}_{\varepsilon\text{-eqdstnc}}, \mathcal{CC}_{\text{AGREE \& PURSUE}})$ and, therefore, we assume that $\mathcal{T}_{\texttt{dir}}$ has been achieved. We consider an initial configuration with the properties that (i) agents are counterclockwise-ordered according to their unique identifier, (ii) the set $H(0)$ is empty, and (iii) the inter-agent distances $d(0) = \big(\mathrm{dist}_{\texttt{cc}}(\theta^{[1]}(0), \theta^{[2]}(0)), \ldots, \mathrm{dist}_{\texttt{cc}}(\theta^{[n]}(0), \theta^{[1]}(0))\big)$ are given by

$$d(0) = \frac{2\pi}{n}\mathbf{1}_n + \frac{\pi - \varepsilon'}{n}(\mathbf{v}_n + \overline{\mathbf{v}}_n),$$

where $\varepsilon' \in]\pi, 0[$ and where $\mathbf{v}_n$ is the eigenvector of $\mathrm{Circ}_n(k_{\mathrm{prop}}, 1 - k_{\mathrm{prop}}, 0)$ corresponding to the eigenvalue $1 - k_{\mathrm{prop}} + k_{\mathrm{prop}} \cos\left(\frac{2\pi}{n}\right) - k_{\mathrm{prop}}\sqrt{-1}\sin\left(\frac{2\pi}{n}\right)$ (see Section 1.5.4). One can verify that $\mathbf{v}_n + \overline{\mathbf{v}}_n = 2(1, \cos(2\pi/n), \ldots, \cos((n-1)2\pi/n))$ and that $\|\mathbf{v}_n + \overline{\mathbf{v}}_n\|_2 = \sqrt{2n}$. In turn, this implies that $d(0) \in \mathbb{R}^n_{>0}$ and that $\|d(0) - \frac{2\pi}{n}\mathbf{1}_n\|_2 \in O(1/\sqrt{n})$. Take $\eta_3 \in]0, 1[$. The argument described in the proof of Theorem 1.74(iii) leads to the following statement: the 2-norm of the difference between $\ell \mapsto d(\ell)$ and the desired configuration $\frac{2\pi}{n}\mathbf{1}_n$ decreases by a factor η_3 in time of order $n^2 \log \eta_3^{-1}$. Given an initial error of order $O(1/\sqrt{n})$ and a final desired error of order ε, we set $\eta_3 = \varepsilon\sqrt{n}$ and obtain the desired result that it takes time of order $n^2 \log(n\varepsilon)^{-1}$ to reduce the 2-norm error, and therefore, the ∞-norm error to size ε. This concludes the proof. ■

3.6.3 Proof of Theorem 3.34

Proof (Theorem 3.34). Note that the number of edges in $\mathcal{S}_{\mathrm{circle}}$ is at most $O(n^2)$ as it is possible that all robots are within distance $r(n)$ of each other. The upper bounds in (i) and (ii) follow then from inequality (3.6) and Theorem 3.33. To prove the lower bounds we follow the steps and notation in the proof of Theorem 3.33. Regarding the lower bounds in (i), we examine the evolution of the initial configuration depicted in Figure 3.8. From *STEP 2:* in the proof of Theorem 3.33, recall that the time it takes agent 1 to receive the message with $\texttt{max-id} = n$ is lower bounded by $\pi/(k_{\mathrm{prop}} r(n)) - 1$. Our proof strategy is to lower bound the number of edges in the graph until this event happens. Note that, at initial time, there are $(n-1)^2$ edges in the communication graph of the network, and therefore, $(n-1)^2$ messages get transmitted. At the next communication round, agent 1 has moved $k_{\mathrm{prop}} r(n)$ counterclockwise and, therefore, the number of edges is lower bounded by $(n-2)^2$. Iterating

this reasoning, we see that after $i < \pi/(k_{\text{prop}} r(n))$ communication rounds, the number of edges is lower bounded by $(n-i)^2$. Now, if $\delta(n) > \pi(1/k_{\text{prop}} - 2)$, then $n > \pi/k_{\text{prop}} r(n))$, and therefore, the total communication complexity is lower bounded by

$$\sum_{i=1}^{\frac{\pi}{k_{\text{prop}} r(n)}} (n-i)^2 \in \Omega(n^2 r(n)^{-1}).$$

On the other hand, if $\delta(n) < \pi(1/k_{\text{prop}} - 2)$, then $n < \pi/k_{\text{prop}} r(n))$, and after n time steps, we lower bound the number of edges in the communication graph by the number of edges in a chain of length n, that is, $n-1$. Therefore, the total communication complexity is lower bounded by

$$\sum_{i=1}^{n} (n-i)^2 + (n-1)\Big(\frac{\pi}{k_{\text{prop}} r(n)} - n\Big) \in \Omega(n^3 + nr(n)^{-1}).$$

The two lower bounds match when $\delta(n) = \pi(1/k_{\text{prop}} - 2)$.

Regarding the lower bound in (ii), we consider first the case when $n_H(0) = 0$. In this case, the network obeys the discrete-time linear time-invariant dynamical system (3.10). Consider the initial condition $d(0)$ that we adopted for *STEP 4:*. We know it takes time of order $n^2 \log(n\varepsilon)^{-1}$ for the appropriate contraction property to hold. At $d(0)$, the maximal inter-agent distance is $(4\pi - \varepsilon')/n$ and it decreases during the evolution. Because each robot can communicate with any other robot within a distance $r(n)$, the number of agents within communication range of a given agent is of order $r(n)n/(4\pi - \varepsilon')$, that is, of order $\delta(n)$. From here we deduce that the total communication complexity belongs to $\Omega(n^3 \delta(n) \log(n\varepsilon)^{-1})$. ■

3.7 Exercises

E3.1 **(Orientation dynamics).** We review some basic kinematic concepts about orientation dynamics; e.g., see [Bullo and Lewis, 2004, Spong et al., 2006]. Define the set of skew symmetric matrices in $\mathbb{R}^{d\times d}$:

$$\mathfrak{so}(d) = \{S \in \mathbb{R}^{d\times d} \mid S = -S^T\}.$$

Let $\times$ denote the cross-product on $\mathbb{R}^3$ and define the linear map $\widehat{\cdot}: \mathbb{R}^3 \to \mathfrak{so}(3)$ by $\widehat{x}y = x \times y$ for all $y \in \mathbb{R}^3$.

(i) Show that, if $x = (x_1, x_2, x_3)$, then

$$\widehat{x} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}.$$

(ii) Given a differentiable curve $R : [0, T] \to \mathrm{SO}(3)$, show that there exists a curve $\omega : [0, T] \to \mathbb{R}^3$ such that

$$\dot{R}(t) = R(t)\widehat{\omega}(t).$$

These two results lead to a motion model of a relative sensing network with time-varying orientation. Generalizing the constant-orientation model in equation (3.5), the complete position and orientation dynamics may be written as:

$$\dot{p}^{[i]}_{\text{fixed}}(t) = R^{[i]}_{\text{fixed}}(t)\, u^{[i]}_i,$$
$$\dot{R}^{[i]}_{\text{fixed}}(t) = R^{[i]}_{\text{fixed}}(t)\, \widehat{\omega}^{[i]}_i,$$

where, for $i \in \{1, \dots, n\}$, $u^{[i]}_i$ and $\omega^{[i]}_i$ are the linear and the body angular velocities of robot i, respectively.

E3.2 **(Variation of agree & pursue control and communication law).** Consider the AGREE & PURSUE control and communication law defined in Section 3.1.3 with the state transition function replaced by the following

`function` $\text{stf}(\theta, w, y)$
1: **for** each non-`null` message $(\theta_{\text{rcvd}}, (\texttt{dir}_{\text{rcvd}}, \texttt{max-id}_{\text{rcvd}}))$ in y **do**
2: **if** $(\texttt{max-id}_{\text{rcvd}} > \texttt{max-id})$ **then**
3: $\texttt{new-dir} := \texttt{dir}_{\text{rcvd}}$
4: $\texttt{new-id} := \texttt{max-id}_{\text{rcvd}}$
5: **return** $(\texttt{new-dir}, \texttt{new-id})$

The only difference between this law and the AGREE & PURSUE law in Section 3.1.3 is that, at each communication round, each agent picks the message with the largest value of max-id among all messages received (instead of among the messages received only from agents moving towards its position). We refer to this law as MOD-AGREE & PURSUE.

Consider the direction agreement task $\mathcal{T}_{\texttt{dir}} : (\mathbb{S}^1)^n \times W^n \to \{\texttt{true}, \texttt{false}\}$ defined in Example 3.22. Assume $\texttt{dir}^{[n]}(0) = \texttt{c}$, and let $k \in \{1, \dots, n-1\}$ be the largest identity such that $\texttt{dir}^{[k]}(0) = \texttt{cc}$. Do the following:

(i) Show that, if the message from agent k gets delivered to agents clockwise-placed with respect to agent k along two consecutive communication rounds, then the message from agent k has traveled at least $(1 - k_{\text{prop}})r(n)$ along the circle in the clockwise direction.

(ii) Show that, if $\text{dist}_{\texttt{cc}}(\theta^{[n]}(0), \theta^{[k]}(0)) < 2\, r(n)$, then

$$\text{TC}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{MOD-AGREE \& PURSUE}}, x_0, w_0) = \Theta(r(n)^{-1}).$$

(iii) Implement the algorithm in your favorite simulation software (e.g., Mathematica, Matlab, or Maple), and compute the time complexity of multiple executions of the algorithm starting from different initial conditions. Does your simulation analysis support the conjecture that

$$\text{TC}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{MOD-AGREE \& PURSUE}}) = \Theta(r(n)^{-1})?$$

For the simulation analysis to be relevant, you should use a large number of randomly generated initial physical positions and processor states.

E3.3 **(Leader-following flocking).** Consider a group of robots moving in $\mathbb{R}^2$ according to the following discrete-time version of the planar vehicle dynamics introduced in Example 3.1:

$$
\begin{aligned}
x(\ell+1) &= x(\ell) + v\cos(\theta(\ell)),\\
y(\ell+1) &= y(\ell) + v\sin(\theta(\ell)),\\
\theta(\ell+1) &= \theta(\ell) + \omega.
\end{aligned}
$$

We let $\{(p^{[1]},\theta^{[1]}),\dots,(p^{[n]},\theta^{[n]})\}$ denote the robot physical states, where $p^{[i]} = (x^{[i]},y^{[i]}) \in \mathbb{R}^2$ corresponds to the position and $\theta^{[i]} \in [0,2\pi)$ corresponds to the orientation of the robot $i \in I$. As communication graph, we adopt the r-disk graph $\mathcal{G}_{\text{disk}}(r)$ introduced in Section 2.2.

Assume all agents move at unit speed, $v = 1$, and update their heading according to the leader-following version of Vicsek's model (see equation (1.8)):

$$
\begin{aligned}
&\theta^{[1]}(\ell+1) = \theta^{[1]}(\ell), \qquad \text{(E3.1)}\\
&\theta^{[i]}(\ell+1) = \operatorname{avrg}\Big(\{\theta^{[i]}(\ell)\} \cup \{\theta^{[j]}(\ell) \mid j \text{ s.t. } \|p^{[j]}(\ell) - p^{[i]}(\ell)\|_2 \le r\}\Big),
\end{aligned}
$$

for $i \in \{2,\dots,n\}$. Do the following:

(i) Write the algorithm formally as a control and communication law as defined in Section 3.1.2.

(ii) Given initial conditions for the position and orientation of the robots, express (E3.1) as the time-dependent linear iteration associated to a sequence of matrices $\{F(\ell) \mid \ell \in \mathbb{Z}_{\ge 0}\}$. Are these matrices stochastic? Are they symmetric? Is the sequence non-degenerate?

(iii) We loosely define the flocking task as achieving agreement on the heading of the agents. Using Theorem 1.58, identify connectivity conditions on the sequence of graphs determined by the evolution of the network that guarantee that agents achieve flocking. What is the final orientation in which the network flocks?

4

Connectivity maintenance and rendezvous

The aims of this chapter are twofold. First, we introduce the rendezvous problem and analyze various coordination algorithms that achieve it, providing upper and lower bounds on their time complexity. Second we introduce the problem of maintaining connectivity among a group of mobile robots and use geometric approaches to preserve this topological property of the network.

Loosely speaking, the *rendezvous objective* is to achieve agreement over the physical location of as many robots as possible, i.e., to steer the robots to a common location. This objective is to be achieved with the limited information flow described in the model of the network. Typically, it will be impossible to solve the rendezvous problem for all robots if the robots are placed in such a way that they do not form a connected communication graph. Therefore, it is reasonable to assume that the network is connected at initial time and that a good property of any rendezvous algorithm is that of maintaining some form of connectivity among robots. This discussion motivates the *connectivity maintenance problem*. Once a model for when two robots can acquire each other's relative position is adopted, this problem is of particular relevance as the inter-robot topology depends on the physical states of the robots. Our exposition here is mainly based on [Ando et al., 1999, Cortés et al., 2006, Ganguli et al., 2007b].

The chapter is organized as follows. In Section 4.1 we formally introduce the two problems. In Section 4.2 we define various connectivity constraint sets to limit the robot motion in order to maintain network connectivity. In Section 4.3 we study various rendezvous algorithms with connectivity maintenance properties. We study the circumcenter algorithm and characterize its complexity. Additionally, we introduce the perimeter minimizing algorithm for nonconvex environments. Finally, in Section 4.5 we present the proofs of the main results of the chapter. Our treatment is based on the LaSalle Invariance Principle and on geometric tools such as proximity graphs and robust visibility.

4.1 Problem statement

We here review the class of networks and the types of problems that will be considered in the chapter.

Networks with discrete-time motion

Along the chapter, we will consider the robotic networks $\mathcal{S}_{\text{disk}}$, $\mathcal{S}_{\text{LD}}$ and $\mathcal{S}_{\infty\text{-disk}}$, and the relative-sensing networks $\mathcal{S}^{\text{rs}}_{\text{disk}}$ and $\mathcal{S}^{\text{rs}}_{\text{vis-disk}}$ presented in Example 3.4 and in Section 3.2.2.

For the robotic networks $\mathcal{S}_{\text{disk}}$, $\mathcal{S}_{\text{LD}}$ and $\mathcal{S}_{\infty\text{-disk}}$, we will, however, assume that the robots move in discrete time, that is, we adopt the discrete-time motion model:

$$p^{[i]}(\ell+1) = p^{[i]}(\ell) + u^{[i]}(\ell), \quad i \in \{1, \dots, n\}. \tag{4.1}$$

Similarly, for the relative-sensing networks $\mathcal{S}^{\text{rs}}_{\text{disk}}$ and $\mathcal{S}^{\text{rs}}_{\text{vis-disk}}$, we adopt the discrete-time motion model:

$$p^{[i]}_{\text{fixed}}(\ell+1) = p^{[i]}_{\text{fixed}}(\ell) + R^{[i]}_{\text{fixed}} u^{[i]}_i(\ell), \quad i \in \{1, \dots, n\}. \tag{4.2}$$

As an aside, if we express the previous equation with respect to frame i at time t, then equation (4.2) reads

$$p^{[i]}_{(\text{frame } i \text{ at time } \ell)}(\ell+1) = u^{[i]}_{(\text{frame } i \text{ at time } \ell)}(\ell), \quad i \in \{1, \dots, n\}.$$

We present the treatment in discrete-time for simplicity. It is easy to show that any control law for the discrete-time motion model can be implemented in the continuous-time networks. We usually assume no bound on the control or $u_{\max}$ as magnitude bound and we explicitly state when we instead introduce the bound.

The rendezvous task

Next, we discuss the rendezvous problem. There are different ways of formulating this objective in terms of task maps. Let $\mathcal{S} = (\{1, \dots, n\}, \mathcal{R}, E_{\text{cmm}})$ be a uniform robotic network. The *(exact) rendezvous task* $\mathcal{T}_{\text{rndzvs}} : X^n \to \{\texttt{true}, \texttt{false}\}$ for $\mathcal{S}$ is the coordination task defined by:

$$\mathcal{T}_{\text{rndzvs}}(x^{[1]}, \dots, x^{[n]}) = \begin{cases} \texttt{true}, & \text{if } x^{[i]} = x^{[j]}, \text{ for all } (i,j) \in E_{\text{cmm}}(x^{[1]}, \dots, x^{[n]}), \\ \texttt{false}, & \text{otherwise.} \end{cases}$$

Next, assume that, for the same network $\mathcal{S} = (\{1, \dots, n\}, \mathcal{R}, E_{\text{cmm}})$, the robots' physical state space is $X \subset \mathbb{R}^d$. It is convenient to review some

basic notation consistent with what we adopted in Chapter 2. We let $\mathcal{P} = \{p^{[1]}, \ldots, p^{[n]}\}$ denote the set of agents location in $X \subset \mathbb{R}^d$ and P be an array of n points in $\mathbb{R}^d$. Furthermore, we let avrg denote the average of a finite point set in $\mathbb{R}^d$, that is,

$$\text{avrg}(\{q_1, \ldots, q_k\}) = \frac{1}{k}(q_1 + \cdots + q_k).$$

For $\varepsilon \in \mathbb{R}_{>0}$, the *$\varepsilon$-rendezvous task* $\mathcal{T}_{\varepsilon\text{-rndzvs}} : (\mathbb{R}^d)^n \to \{\texttt{true}, \texttt{false}\}$ for $\mathcal{S}$ is defined as follows: $\mathcal{T}_{\varepsilon\text{-rndzvs}}$ is `true` at P if each robot position $p^{[i]}$, for $i \in \{1, \ldots, n\}$, is at distance less than ε from the average position of its E_{cmm}-neighbors. Formally,

$$\begin{aligned} &\mathcal{T}_{\varepsilon\text{-rndzvs}}(P) = \texttt{true} \\ &\quad \iff \|p^{[i]} - \text{avrg}\left(\{p^{[j]} \mid (i,j) \in E_{\text{cmm}}(P)\}\right)\|_2 < \varepsilon, \quad i \in \{1, \ldots, n\}. \end{aligned}$$

The connectivity maintenance problem

Assume the communication graph, computed as a function of the robot positions, is connected: how should the robots move in such a way that their communication graph is again connected. Clearly, the problem depends upon (1) how do the robots move, and (2) what proximity graph describes the communication graph or, in the case of relative-sensing networks, what sensor model is available on each robot.

The key idea is to restrict the allowable motion of each agent. Different motion constraint sets correspond to different communication or sensing graphs. We have three objectives in doing so. First, we aim to achieve this objective only based on local measurements or 1-hop communication, i.e., without introducing processor states explicitly dedicated to this task. Second, the constraint sets should depend continuously on the position of the robots. Third, we have the somehow informal objective to design the constraint sets as "large" as possible so as to minimally constrain the motion of the robots.

4.2 Connectivity maintenance algorithms

In this section we present some algorithms that might be used by a robotic network to maintain communication connectivity. The results presented in this section start with the original idea introduced by Ando et al. [1999] for first-order robots communicating along the edges of a disk graph, that is, for the network described in Example 3.4. This idea is then generalized to a number of useful settings. The properties of proximity graphs presented in Section 2.2 play a key role in formulating and solving the connectivity problem.

4.2.1 Enforcing range-limited links

First, we aim to constrain the motion of two first-order agents in order to maintain a communication link between them. We assume that the communication takes place over the disk graph $\mathcal{G}_{\text{disk}}(r)$ with communication range $r > 0$.

Loosely stated, the *pairwise connectivity maintenance problem* is as follows: given two neighbors in the proximity graph $\mathcal{G}_{\text{disk}}(r)$, find a rich set of control inputs for both agents with the property that, after moving, both agents are again within distance r. We provide a solution to this problem as follows.

Definition 4.1 (Pairwise connectivity constraint set). Consider two agents i and j at positions $p^{[i]} \in \mathbb{R}^d$ and $p^{[j]} \in \mathbb{R}^d$ such that $\|p^{[i]} - p^{[j]}\|_2 \leq r$. The *connectivity constraint set* of agent i with respect to agent j is

$$\mathcal{X}_{\text{disk}}\big(p^{[i]}, p^{[j]}\big) = \overline{B}\Big(\frac{p^{[j]} + p^{[i]}}{2}, \frac{r}{2}\Big). \qquad \bullet$$

Note that both robots i and j can independently compute their respective connectivity constraint sets. The proof of the following result is straightforward.

Lemma 4.2 (Maintaining pairwise connectivity). *Assume that at time ℓ the distance between agents $p^{[i]}$ and $p^{[j]}$ is no more than r. If the control $u^{[i]}(\ell)$ takes value in*

$$u^{[i]}(\ell) \in \mathcal{X}_{\text{disk}}\big(p^{[i]}(\ell), p^{[j]}(\ell)\big) - p^{[i]}(\ell) = \overline{B}\Big(\frac{p^{[j]}(\ell) - p^{[i]}(\ell)}{2}, \frac{r}{2}\Big),$$

and, similarly, $u^{[j]}(\ell) \in \mathcal{X}_{\text{disk}}\big(p^{[j]}(\ell), p^{[i]}(\ell)\big) - p^{[j]}(\ell)$, then, according to the discrete-time motion model (4.1),

(i) both agents positions at time $\ell + 1$ are inside $\mathcal{X}_{\text{disk}}\big(p^{[i]}(\ell), p^{[j]}(\ell)\big)$, and
(ii) the distance between the agents positions at time $\ell + 1$ is no more than r.

We illustrate these pairwise connectivity maintenance concepts in Figure 4.1.

Remark 4.3 (Constraints for relative-sensing networks). Let us consider a relative-sensing network with a disk sensor of radius r; see Example 3.15. Recall the following facts about this model. First, agent i measures the position of robot j in its frame $\Sigma^{[i]}$, that is, robot i measures $p_i^{[j]}$. Second, $p_i^{[i]} = \mathbf{0}_d$. Third, if $W \subset \mathbb{R}^d$, then W_i denotes its expression in the frame $\Sigma^{[i]}$. Combining these notions and assuming the inter-agent distance is no more than r, the pairwise connectivity constraint set in Definition 4.1 satisfies:

$$\Big(\mathcal{X}_{\text{disk}}(p^{[i]}, p^{[j]})\Big)_i = \mathcal{X}_{\text{disk}}\big(\mathbf{0}_d, p_i^{[j]}\big) = \overline{B}\Big(\frac{p_i^{[j]}}{2}, \frac{r}{2}\Big). \qquad \bullet$$

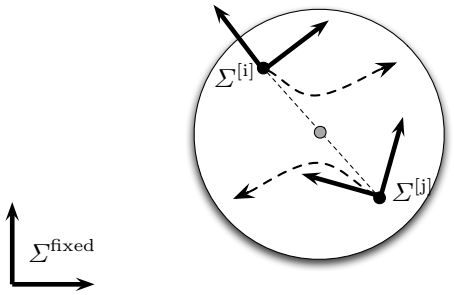


Fig. 4.1. Illustration of the connectivity maintenance constraint. Starting from positions $p^{[i]}$ and $p^{[j]}$, the robots are restricted to move inside the disk centered at $\mathcal{X}_{\text{disk}}(p^{[i]}, p^{[j]}) = \frac{1}{2}\big(p^{[i]} + p^{[j]}\big)$ with radius $\frac{r}{2}$.

4.2.2 Enforcing network connectivity

Here, we focus on how to constrain the mobility of multiple agents in order to maintain connectivity for the entire network that they form. We again consider the case of first-order agents moving according to the discrete-time equation (4.1) and communicating over $\mathcal{G}_{\text{disk}}(r)$.

Loosely stated, the *network connectivity maintenance problem* is as follows: Given n agents at positions $\mathcal{P}(\ell) = \{p^{[1]}(\ell), \dots, p^{[n]}(\ell)\}$ in which they form a connected r-disk graph $\mathcal{G}_{\text{disk}}(r)$, the objective is to find a rich set of control inputs for all agents with the property that, at time $\ell + 1$, the agents' new positions $\mathcal{P}(\ell + 1)$ form again a connected r-disk graph $\mathcal{G}_{\text{disk}}(r)$. We provide a simple, but potentially conservative, solution to this problem as follows.

Definition 4.4 (Connectivity constraint set). Consider a group of agents at positions $\mathcal{P} = \{p^{[1]}, \dots, p^{[n]}\} \subset \mathbb{R}^d$. The *connectivity constraint set* of agent i with respect to $\mathcal{P}$ is

$$\mathcal{X}_{\text{disk}}(p^{[i]}, \mathcal{P}) = \bigcap \big\{ \mathcal{X}_{\text{disk}}(p^{[i]}, q) \mid q \in \mathcal{P} \setminus \{p^{[i]}\} \text{ s.t. } \|q - p^{[i]}\|_2 \leq r \big\}. \qquad \bullet$$

In other words, if $q_1, \dots, q_l$ are agents positions whose distance from $p^{[i]}$ is no more than r, then the connectivity constraint set for agent i is the intersection of the constraint sets $\overline{B}\big(\frac{1}{2}(q_k + p^{[i]}), \frac{r}{2}\big)$ for $k \in \{1, \dots, l\}$; see Figure 4.2.

The following result is a consequence of Lemma 4.2.

Lemma 4.5 (Maintaining network connectivity). *Consider a group of agents at positions $\mathcal{P}(\ell) = \{p^{[1]}(\ell), \dots, p^{[n]}(\ell)\} \subset \mathbb{R}^d$ at time ℓ. If each agent's control $u^{[i]}(\ell)$ takes value in*

$$u^{[i]}(\ell) \in \mathcal{X}_{\text{disk}}\big(p^{[i]}(\ell), \mathcal{P}(\ell)\big) - p^{[i]}(\ell), \quad i \in \{1, \dots, n\},$$

then, according to the discrete-time motion model (4.1),

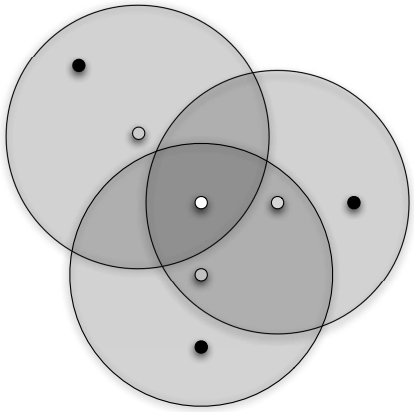


Fig. 4.2. Illustration of network connectivity maintenance. The connectivity $\mathcal{X}_{\text{disk}}$-constraint set of the white-colored agent is the intersection of the individual constraint sets determined by its neighbors.

(i) each agent remains in its connectivity constraint set, that is, $p^{[i]}(\ell+1) \in \mathcal{X}_{\text{disk}}(p^{[i]}(\ell), \mathcal{P}(\ell))$,

(ii) each edge of $\mathcal{G}_{\text{disk}}(r)$ *at* $\mathcal{P}(\ell)$ *is maintained after the motion step, i.e., if* $\|p^{[i]}(\ell) - p^{[j]}(\ell)\|_2 \leq r$, *then also* $\|p^{[i]}(\ell+1) - p^{[j]}(\ell+1)\|_2 \leq r$,

(iii) if $\mathcal{G}_{\text{disk}}(r)$ *at time* ℓ *is connected, then* $\mathcal{G}_{\text{disk}}(r)$ *at time* $\ell+1$ *is connected, and*

(iv) the number of connected components of the graph $\mathcal{G}_{\text{disk}}(r)$ *at time* $\ell+1$ *is equal to or smaller than the number of connected components of the graph* $\mathcal{G}_{\text{disk}}(r)$ *at time* ℓ.

Remark 4.6 (Constraints for relative-sensing networks: cont'd). Following up on Remark 4.3, the connectivity constraint set in Definition 4.4, written in the frame $\Sigma^{[i]}$, is

$$\mathcal{X}_{\text{disk}}(\mathbf{0}_d, \{p_i^{[1]}, \dots, p_i^{[n]}\}) = \bigcap \left\{ \overline{B}\Big(\frac{p_i^{[j]}}{2}, \frac{r}{2}\Big) \mid j \neq i \text{ such that } \|p^{[j]} - p^{[i]}\|_2 \leq r \right\}. \quad \bullet$$

Next, we relax the constraints in Definition 4.4 to provide the network nodes with larger, and therefore less conservative, motion constraint sets. Recall from Section 2.2 the relative neighborhood graph $\mathcal{G}_{\text{RN}}$, the Gabriel graph $\mathcal{G}_{\text{G}}$, and the r-limited Delaunay graph $\mathcal{G}_{\text{LD}}(r)$. These proximity graphs are illustrated in Figure 2.8. From Theorem 2.9 and Proposition 2.10, respectively, recall that the proximity graphs $\mathcal{G}_{\text{RN}} \cap \mathcal{G}_{\text{disk}}(r)$, $\mathcal{G}_{\text{G}} \cap \mathcal{G}_{\text{disk}}(r)$ and $\mathcal{G}_{\text{LD}}(r)$ have the following properties:

(i) they have the same connected components as $\mathcal{G}_{\text{disk}}(r)$, that is, for all point sets $\mathcal{P} \subset \mathbb{R}^d$, all graphs have the same number of connected components consisting of the same vertices, and
(ii) they are spatially distributed over $\mathcal{G}_{\text{disk}}(r)$.

These mathematical facts have two implications. First, to maintain or decrease the number of connected components of a disk graph, it is sufficient to maintain or decrease the number of connected components of any of the three proximity graphs $\mathcal{G}_{\text{RN}} \cap \mathcal{G}_{\text{disk}}(r)$, $\mathcal{G}_{\text{G}} \cap \mathcal{G}_{\text{disk}}(r)$ and $\mathcal{G}_{\text{LD}}(r)$. Because each of these graphs is more sparse than the disk graph, i.e., they are subgraphs of $\mathcal{G}_{\text{disk}}(r)$, fewer connectivity constraints need to be imposed. Second, because these proximity graphs are spatially distributed over the disk graph, it is possible for each agent to determine which of its neighbors in $\mathcal{G}_{\text{disk}}(r)$ are its neighbors also in these subgraphs. We formalize this discussion as follows.

Definition 4.7 ($\mathcal{G}$-connectivity constraint set). Let $\mathcal{G}$ be a proximity graph that is spatially distributed over $\mathcal{G}_{\text{disk}}(r)$ and that has the same connected components as $\mathcal{G}_{\text{disk}}(r)$. Consider a group of agents at positions $\mathcal{P} = \{p^{[1]}, \dots, p^{[n]}\} \subset \mathbb{R}^d$. The *$\mathcal{G}$-connectivity constraint set* of agent i with respect to $\mathcal{P}$ is

$$\mathcal{X}_{\text{disk},\mathcal{G}}(p^{[i]}, \mathcal{P}) = \bigcap \left\{ \mathcal{X}_{\text{disk}}(p^{[i]}, q) \mid q \in \mathcal{P} \text{ s. t. } (q, p^{[i]}) \text{ is an edge of } \mathcal{G}(\mathcal{P}) \right\}.$$

•

Lemma 4.8 (Maintaining connectivity of sparser networks). *Let $\mathcal{G}$ be a proximity graph that is spatially distributed over $\mathcal{G}_{\text{disk}}(r)$ and that has the same connected components as $\mathcal{G}_{\text{disk}}(r)$. Consider a group of agents at positions $\mathcal{P}(\ell) = \{p^{[1]}(\ell), \dots, p^{[n]}(\ell)\} \subset \mathbb{R}^d$ at time ℓ. If each agent's control $u^{[i]}(\ell)$ takes value in*

$$u^{[i]}(\ell) \in \mathcal{X}_{\text{disk},\mathcal{G}}\big(p^{[i]}(\ell), \mathcal{P}(\ell)\big) - p^{[i]}(\ell), \quad i \in \{1, \dots, n\},$$

then, according to the discrete-time motion model (4.1),

(i) each agent remains in its $\mathcal{G}$-connectivity constraint set,
(ii) two agents that are in the same connected component of $\mathcal{G}$ remain at the same connected component after the motion step,
(iii) the number of connected components of the graph $\mathcal{G}$ at $\mathcal{P}(\ell+1)$ is equal to or smaller than the number of connected components of the graph $\mathcal{G}$ at $\mathcal{P}(\ell)$.

The reader is asked to provide a proof of this result in Exercise E4.1.

4.2.3 Enforcing range-limited line-of-sight links and network connectivity

Here, we consider the connectivity maintenance problem for a group of agents with range-limited line-of-sight communication, as described in Example 3.6.

It is convenient to treat directly and only the case of a compact allowable non-convex environment $Q \subset \mathbb{R}^2$ contracted into $Q_\delta = \{q \in Q \mid \text{dist}(q, \partial Q) \geq \delta\}$ for a small positive δ. We present a solution based on designing constraint sets that guarantee that every edge of the range-limited visibility graph $\mathcal{G}_{\text{vis-disk},Q_\delta}$ is preserved.

We begin with a useful observation and a corresponding geometric algorithm. Assume that, at time ℓ, robot j is inside the range-limited visibility set from $p^{[i]}$ in Q_δ, that is, with the notation of Section 2.1.2,

$$p^{[j]}(\ell) \in \text{Vi}_{\text{disk}}(p^{[i]}(\ell); Q_\delta) = \text{Vi}(p^{[i]}(\ell); Q_\delta) \cap \overline{B}(p^{[i]}(\ell), r).$$

This property holds also at time $\ell + 1$ if $\|p^{[i]}(\ell+1) - p^{[j]}(\ell+1)\|_2 \leq r$ and $[p^{[i]}(\ell+1), p^{[j]}(\ell+1)] \subset Q_\delta$. A sufficient condition is therefore that

$$p^{[i]}(\ell+1),\ p^{[j]}(\ell+1) \in \mathcal{X},$$

for some convex subset $\mathcal{X}$ of $Q_\delta \cap \overline{B}\big(\frac{1}{2}(p^{[i]}(\ell) + p^{[j]}(\ell)), \frac{r}{2}\big)$. Intuitively speaking, $\mathcal{X}$ plays the role of $\mathcal{X}$-constraint set for the proximity graph $\mathcal{G}_{\text{vis-disk},Q_\delta}$. The following geometric algorithm, given the positions $p^{[i]}$ and $p^{[j]}$ in an environment Q_δ, computes precisely one such convex subset.

`function` ITERATED TRUNCATION$(p^{[i]}, p^{[j]}; Q_\delta)$
% Executed by robot i for $p^{[j]}$ within range-limited line of sight of $p^{[i]}$

1: $\mathcal{X}_{\text{temp}} := \text{Vi}_{\text{disk}}(p^{[i]}; Q_\delta) \cap \overline{B}\big(\frac{1}{2}(p^{[i]} + p^{[j]}), \frac{r}{2}\big)$
2: **while** $\partial\mathcal{X}_{\text{temp}}$ contains a concavity **do**
3: $\quad v :=$ a strictly concave point of $\partial\mathcal{X}_{\text{temp}}$ closest to $[p^{[i]}, p^{[j]}]$
4: $\quad \mathcal{X}_{\text{temp}} := \mathcal{X}_{\text{temp}} \cap H_{Q_\delta}(v)$
5: **return** $\mathcal{X}_{\text{temp}}$

Note: in step `3:` multiple points belonging to distinct concavities may satisfy the required property. If so, v may be chosen as any of them.

Figure 4.3(b) illustrates an example convex constraint set computed by the ITERATED TRUNCATION algorithm. Figure 4.4 illustrates the step-by-step execution required to generate Figure 4.3(b).

Next, we characterize the main properties of the ITERATED TRUNCATION algorithm. It is convenient to define the following set:

$$J = \{(p, q) \in Q_\delta \times Q_\delta \mid [p, q] \in Q_\delta \text{ and } \|p - q\|_2 \leq r\}.$$

Proposition 4.9 (Properties of the iterated truncation algorithm). *Consider the δ-contraction of a compact allowable environment Q_δ with κ strict concavities, and let $(p^{[i]}, p^{[j]}) \in J$. The following statements hold:*

(i) The ITERATED TRUNCATION *algorithm, invoked with arguments $(p^{[i]}, p^{[j]}; Q_\delta)$, terminates in at most κ steps; denote its output by $\mathcal{X}_{\text{vis-disk}}(p^{[i]}, p^{[j]}; Q_\delta)$;*
(ii) $\mathcal{X}_{\text{vis-disk}}(p^{[i]}, p^{[j]}; Q_\delta)$ is nonempty, compact and convex;
(iii) $\mathcal{X}_{\text{vis-disk}}(p^{[i]}, p^{[j]}; Q_\delta) = \mathcal{X}_{\text{vis-disk}}(p^{[j]}, p^{[i]}; Q_\delta)$; and

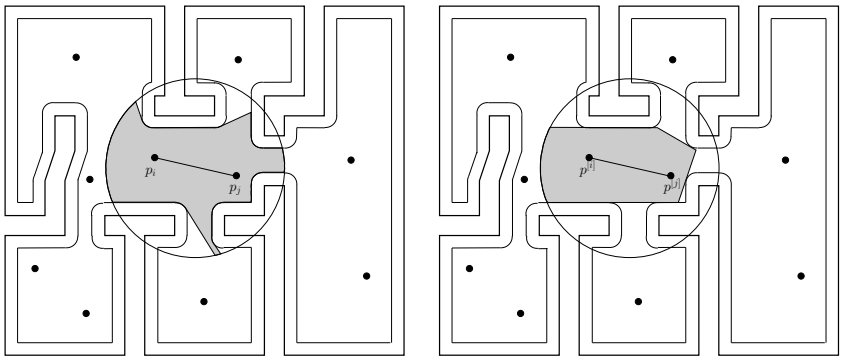


Fig. 4.3. The plot in (a) shows the set $\mathrm{Vi}_{\mathrm{disk}}(p^{[i]}; Q_\delta) \cap \overline{B}(\frac{1}{2}(p^{[i]} + p^{[j]}), \frac{r}{2})$. The plot in (b) shows the outcome of the execution of the ITERATED TRUNCATION algorithm. Robots i and j are constrained to remain inside the shaded region in (b), which is a convex subset of Q_δ and of the closed ball with center $\frac{1}{2}(p^{[i]} + p^{[j]})$ and radius $\frac{r}{2}$.

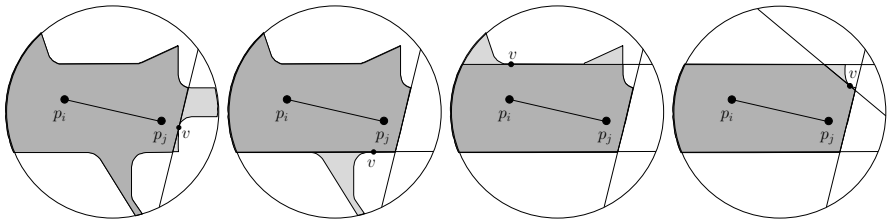


Fig. 4.4. From left to right, a sample run of the ITERATED TRUNCATION algorithm. The initial set $\mathcal{X}_{\mathrm{temp}} := \mathrm{Vi}_{\mathrm{disk}}(p^{[i]}; Q_\delta) \cap \overline{B}(\frac{1}{2}(p^{[i]} + p^{[j]}), \frac{r}{2})$ is shown in Figure 4.3(a). The lightly and darkly shaded regions together represent $\mathcal{X}_{\mathrm{temp}}$ at the current iteration. The darkly shaded region represents $\mathcal{X}_{\mathrm{temp}} \cap H_{Q_\delta}(v)$, where v is as described in step `3:`. The outcome of the execution is shown in Figure 4.3(b).

(iv) the set-valued map $(p, q) \mapsto \mathcal{X}_{\text{vis-disk}}(p, q; Q_\delta)$ *is closed at all* $(p, q) \in J$.

In the interest of brevity we do not include the proof here and instead refer the reader to [Ganguli et al., 2007b]. We just mention that fact (iii) is a consequence of the fact that all relevant concavities in the computation of $\mathcal{X}_{\text{vis-disk}}(p^{[i]}, p^{[j]}; Q_\delta)$ are visible from both agents $p^{[i]}$ and $p^{[j]}$. We are finally ready to provide analogs of Definition 4.4 and Lemma 4.5.

Definition 4.10 (Line-of-sight connectivity constraint set). Consider a nonconvex allowable environment Q_δ and two agents i and j within range-limited line of sight. We call $\mathcal{X}_{\text{vis-disk}}(p^{[i]}, p^{[j]}; Q_\delta)$ the *pairwise line-of-sight connectivity constraint set* of agent i with respect to agent j. Furthermore, given agents at positions $\mathcal{P} = \{p^{[1]}, \ldots, p^{[n]}\} \subset Q_\delta$ that are all within range-

limited line of sight of agent i, the *line-of-sight connectivity constraint sets* of agent i with respect to $\mathcal{P}$ is

$$\mathcal{X}_{\text{vis-disk}}(p^{[i]}, \mathcal{P}; Q_\delta) = \bigcap \{\mathcal{X}_{\text{vis-disk}}(p^{[i]}, q; Q_\delta) \mid q \in \mathcal{P} \setminus \{p^{[i]}\}\}. \qquad \bullet$$

The following result is a consequence of Proposition 4.9.

Lemma 4.11 (Maintaining network connectivity). *Consider a group of agents at positions $\mathcal{P}(\ell) = \{p^{[1]}(\ell), \dots, p^{[n]}(\ell)\} \subset Q_\delta$ at time ℓ. If each agent's control $u^{[i]}(\ell)$ takes value in*

$$u^{[i]}(\ell) \in \mathcal{X}_{\text{vis-disk}}\big(p^{[i]}(\ell), \mathcal{P}(\ell); Q_\delta\big) - p^{[i]}(\ell), \quad i \in \{1, \dots, n\},$$

then, according to the discrete-time motion model (4.1),

(i) each agent remains in its constraint set, that is,

$$p^{[i]}(\ell + 1) \in \mathcal{X}_{\text{vis-disk}}(p^{[i]}(\ell), \mathcal{P}(\ell); Q_\delta),$$

(ii) each edge of $\mathcal{G}_{\text{vis-disk},Q_\delta}$ at $\mathcal{P}(\ell)$ is maintained after the motion step, i.e., if $p^{[i]}$ and $p^{[j]}$ are within range-limited line of sight at time ℓ, then they are so also at time $\ell + 1$,

(iii) if $\mathcal{G}_{\text{vis-disk},Q_\delta}$ at $\mathcal{P}(\ell)$ is connected, then $\mathcal{G}_{\text{vis-disk},Q_\delta}$ at $\mathcal{P}(\ell+1)$ is connected, and

(iv) the number of connected components of the graph $\mathcal{G}_{\text{vis-disk},Q_\delta}$ at $\mathcal{P}(\ell+1)$ is equal to or smaller than the number of connected components of the graph $\mathcal{G}_{\text{vis-disk},Q_\delta}$ at $\mathcal{P}(\ell)$.

Remark 4.12 (Constraints for relative-sensing networks: cont'd). Following up on Remarks 4.3 and 4.6, we consider a relative-sensing network with range-limited visibility sensors, see Example 3.16. To compute the connectivity constraint set for this network, it suffices to provide a relative sensing version of the ITERATED TRUNCATION algorithm:

`function` RELATIVE-SENSING ITERATED TRUNCATION$(y; y_{\text{env}})$
% Executed by robot i with range-limited visibility sensor:
% robot measurement is $y = p_i^{[j]} \in \text{Vi}_{\text{disk}}(\mathbf{0}_2; (Q_\delta)_i)$ for some $j \neq i$
% environment measurement is $y_{\text{env}} = \text{Vi}_{\text{disk}}(\mathbf{0}_2; (Q_\delta)_i)$

1: $\mathcal{X}_{\text{temp}} := y_{\text{env}} \cap \overline{B}\big(\frac{p_i^{[j]}}{2}, \frac{r}{2}\big)$
2: **while** $\partial\mathcal{X}_{\text{temp}}$ contains a concavity **do**
3: $\quad v :=$ a strictly concave point of $\partial\mathcal{X}_{\text{temp}}$ closest to $[\mathbf{0}_2, y]$
4: $\quad \mathcal{X}_{\text{temp}} := \mathcal{X}_{\text{temp}} \cap H_{y_{\text{env}}}(v)$
5: **return** $\mathcal{X}_{\text{temp}}$

The algorithm output is $\mathcal{X}_{\text{vis-disk}}(\mathbf{0}_d, y)$, for $y = p_i^{[j]} \in \text{Vi}_{\text{disk}}(\mathbf{0}_2; (Q_\delta)_i)$. $\bullet$

Next, we relax the constraints in Definition 4.10 to provide the network nodes with larger, and therefore less conservative, motion constraint sets. Similarly to Section 4.2.2, we seek to enforce the preservation of a fewer number of range-limited line-of-sight links while still making sure that the overall network connectivity is preserved. To do this, we recall from Section 2.2 the notion of locally-cliqueless graph $\mathcal{G}_{\text{lc},\mathcal{G}}$ of a proximity graph $\mathcal{G}$. This proximity graph is illustrated in Figure 2.11. Let us use the shorthand notation $\mathcal{G}_{\text{lc-vis-disk},Q_\delta} \equiv \mathcal{G}_{\text{lc},\mathcal{G}_{\text{vis-disk},Q_\delta}}$. From Theorem 2.12(ii) and (iii), respectively, recall that $\mathcal{G}_{\text{lc-vis-disk},Q_\delta}$ has the following properties:

(i) it has the same connected components as $\mathcal{G}_{\text{vis-disk},Q_\delta}$, that is, for all point sets $\mathcal{P} \subset \mathbb{R}^d$, the graph has the same number of connected components consisting of the same vertices, and
(ii) it is spatially distributed over $\mathcal{G}_{\text{vis-disk},Q_\delta}$.

Because of (i), to maintain or decrease the number of connected components of a range-limited visibility graph, it is sufficient to maintain or decrease the number of connected components of the sparser graph $\mathcal{G}_{\text{lc-vis-disk},Q_\delta}$. Because of (ii), it is possible for each agent to determine which of its neighbors in $\mathcal{G}_{\text{vis-disk},Q_\delta}$ are its neighbors also in $\mathcal{G}_{\text{lc-vis-disk},Q_\delta}$. We formalize this discussion as follows.

Definition 4.13 (Locally-cliqueless line-of-sight connectivity constraint set). Consider a nonconvex allowable environment $Q_\delta \subset \mathbb{R}^2$ and a group of agents at positions $\mathcal{P} = \{p^{[1]}, \dots, p^{[n]}\} \subset Q$. The *locally-cliqueless line-of-sight connectivity constraint set* of agent i with respect to $\mathcal{P}$ is

$$\mathcal{X}_{\text{lc-vis-disk}}(p^{[i]}, \mathcal{P}; Q_\delta) = \bigcap \big\{ \mathcal{X}_{\text{vis-disk}}(p^{[i]}, q; Q_\delta) \mid q \in \mathcal{P} \text{ s. t. } (q, p^{[i]}) \text{ is an edge of } \mathcal{G}_{\text{lc-vis-disk},Q_\delta}(\mathcal{P}) \big\}.$$

•

Lemma 4.14 (Maintaining connectivity of sparser networks). *Consider a group of agents at positions $\mathcal{P}(\ell) = \{p^{[1]}(\ell), \dots, p^{[n]}(\ell)\} \subset Q_\delta$ at time ℓ. If each agent's control $u^{[i]}(\ell)$ takes value in*

$$u^{[i]}(\ell) \in \mathcal{X}_{\text{lc-vis-disk}}\big(p^{[i]}(\ell), \mathcal{P}(\ell); Q_\delta\big) - p^{[i]}(\ell), \quad i \in \{1, \dots, n\},$$

then, according to the discrete-time motion model (4.1),

(i) each agent remains in its locally-cliqueless line-of-sight connectivity constraint set,
(ii) two agents that are in the same connected component of $\mathcal{G}_{\text{lc-vis-disk},Q_\delta}$ remain at the same connected component after the motion step,
(iii) the number of connected components of the graph $\mathcal{G}_{\text{lc-vis-disk},Q_\delta}$ at $\mathcal{P}(\ell+1)$ is equal to or smaller than the number of connected components of the graph $\mathcal{G}_{\text{lc-vis-disk},Q_\delta}$ at $\mathcal{P}(\ell)$.

4.3 Rendezvous algorithms

In this section we present some algorithms that might be used by a robotic network to achieve rendezvous. Throughout the section we mainly focus on the uniform network $\mathcal{S}_{\text{disk}}$ of locally-connected first-order agents in $\mathbb{R}^d$; this robotic network was introduced in Example 3.4.

4.3.1 Averaging control and communication law

We first study a behavior in which agents move towards a position computed as the average of the received messages. This law is related to the distributed linear algorithms discussed in Section 1.5 and, in particular, to adjacency-based agreement algorithms and the Vicsek's model. This algorithm has also been studied in the context of "opinion dynamics under bounded confidence" and is known in the literature as the Krause model.

We loosely describe the AVERAGING law, that we denote by $\mathcal{CC}_{\text{AVERAGING}}$, as follows:

> *[Informal description]* At each communication round each agent performs the following tasks: (i) it transmits its position and receives its neighbors' positions; (ii) it computes the average of the point set comprised of its neighbors and of itself. Between communication rounds, each robot moves toward the average point it computed.

We next formulate the algorithm using the description model of Chapter 3. The law is uniform, static, and data-sampled, with standard message-generation function.

`Robotic Network:` $\mathcal{S}_{\text{disk}}$ with motion model (4.1) in $\mathbb{R}^d$,
with absolute sensing of own position, and
with communication range r

`Distributed Algorithm:` AVERAGING
`Alphabet:` $\mathbb{A} = \mathbb{R}^d \cup \{\texttt{null}\}$

`function` $\text{msg}(p, i)$
1: **return** p

`function` $\text{ctl}(p, y)$
1: **return** $\text{avrg}(\{p\} \cup \{p_{\text{rcvd}} \mid p_{\text{rcvd}} \text{ is a non-null message in } y\}) - p$

An implementation of this control and communication law is shown in Figure 4.5 for $d = 1$. Note that, along the evolution, (1) several robots *rendezvous*, i.e., agree upon a common location, and (2) some robots are connected at the simulation's beginning and not connected at the simulation's end. Our analysis of the performance of this law is contained in the following theorem, whose proof is postponed to Section 4.5.1.

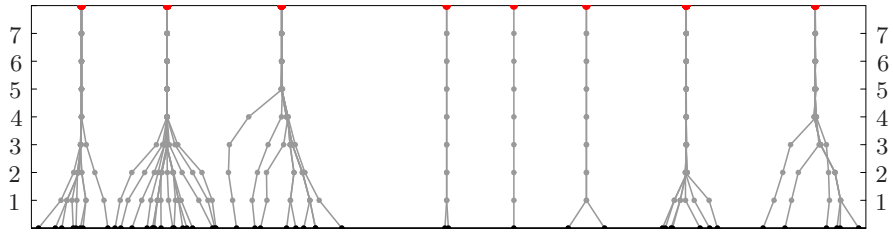


Fig. 4.5. Evolution of a robotic network under the AVERAGING control and communication law in Section 4.3.1 implemented over the network $\mathcal{S}_{\text{disk}}$, with $r = 1.5$. The vertical axis corresponds to the elapsed time, and the horizontal axis to the positions of the agents in the real line. The 51 agents are initially randomly deployed over the interval $[-15, 15]$.

Theorem 4.15 (Correctness and time complexity of averaging law). *For $d = 1$, the network $\mathcal{S}_{\text{disk}}$, the law $\mathcal{CC}_{\text{AVERAGING}}$ achieves the task $\mathcal{T}_{\text{rndzvs}}$ with time complexity*

$$\text{TC}(\mathcal{T}_{\text{rndzvs}}, \mathcal{CC}_{\text{AVERAGING}}) \in O(n^5),$$
$$\text{TC}(\mathcal{T}_{\text{rndzvs}}, \mathcal{CC}_{\text{AVERAGING}}) \in \Omega(n).$$

4.3.2 Circumcenter control and communication laws

Here we define the CIRCUMCENTER control and communication law for the network $\mathcal{S}_{\text{disk}}$. The law solves the rendezvous problem while maintaining the network connected. This law was introduced by Ando et al. [1999] and later studied in [Lin et al., 2007a, Cortés et al., 2006].

It is convenient to recall two useful geometric concepts: (i) given a bounded set S, its circumcenter $\text{CC}(S)$ is the center of the closed ball of minimum radius containing S (see Section 2.1.3); (ii) given a point p in a convex set Q and a second point q, the from-to-inside map $\text{fti}(p, q, S)$ is the point in the closed segment $[p, q]$ which is at the same time closest to q and inside S (see Section 2.1.1). Finally, recall also the connectivity constraint set introduced in Definition 4.4.

We loosely describe the CIRCUMCENTER law, that we denote by $\mathcal{CC}_{\text{CIRCUMCENTER}}$, as follows:

> *[Informal description]* At each communication round each agent performs the following tasks: (i) it transmits its position and receives its neighbors' positions; (ii) it computes the circumcenter of the point set comprised of its neighbors and of itself. Between communication rounds, each robot moves toward this circumcenter point while maintaining connectivity with its neighbors using appropriate connectivity constraint sets.

We next formulate the algorithm using the description model of Chapter 3. The law is uniform, static, and data-sampled, with standard message-generation function.

`Robotic Network:` $\mathcal{S}_{\text{disk}}$ with discrete-time motion model (4.1) in $\mathbb{R}^d$, with absolute sensing of own position, and with communication range r

`Distributed Algorithm:` CIRCUMCENTER
`Alphabet:` $\mathbb{A} = \mathbb{R}^d \cup \{\texttt{null}\}$

`function` $\text{msg}(p, i)$

1: **return** p

`function` $\text{ctl}(p, y)$

1: $p_{\text{goal}} := \text{CC}(\{p\} \cup \{p_{\text{rcvd}} \mid \text{for all non-null } p_{\text{rcvd}} \in y\})$
2: $\mathcal{X} := \mathcal{X}_{\text{disk}}(p, \{p_{\text{rcvd}} \mid \text{for all non-null } p_{\text{rcvd}} \in y\})$
3: **return** $\text{fti}(p, p_{\text{goal}}, \mathcal{X}) - p$

This algorithm is illustrated in Figure 4.6.

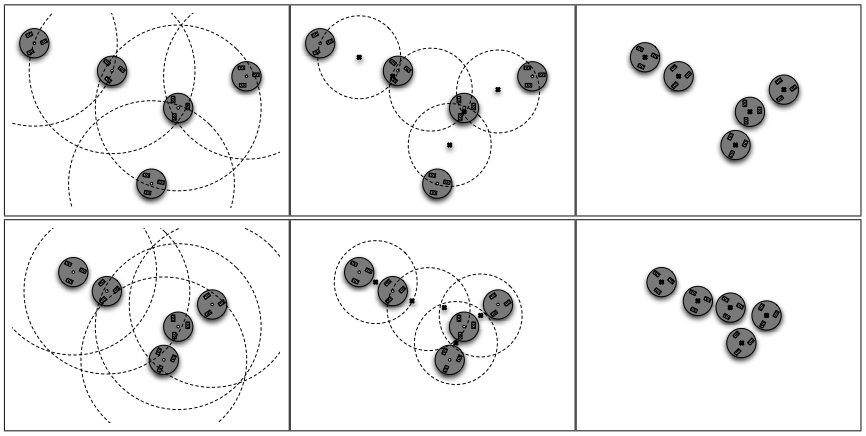


Fig. 4.6. Illustration of the execution of CIRCUMCENTER. Each row of plots represents an iteration of the law. At each round, each agent computes its goal point and its constraint set, and then moves towards the goal while remaining in the constraint set.

Remark 4.16 (Relative sensing version). It is possible and straightforward to implement the circumcenter law as a static relative-sensing control law on the relative-sensing network with disk sensors $\mathcal{S}^{\text{rs}}_{\text{disk}}$ introduced in Example 3.15

`Relative Sensing Network:` $\mathcal{S}^{\text{rs}}_{\text{disk}}$ with motion model (4.2), no communication, relative sensing for robot i given by: robot measurements y contains $p_i^{[j]} \in \overline{B}(\mathbf{0}_2, r)$ for all $j \neq i$

`Distributed Algorithm:` RELATIVE-SENSING CIRCUMCENTER

`function` ctl(y)

1: $p_{\text{goal}} := \text{CC}(\{\mathbf{0}_d\} \cup \{p_{\text{snsd}} \mid \text{for all non-null } p_{\text{snsd}} \in y\})$
2: $\mathcal{X} := \mathcal{X}_{\text{disk}}(\mathbf{0}_d, \{p_{\text{snsd}} \mid \text{for all non-null } p_{\text{snsd}} \in y\})$
3: **return** $\text{fti}(\mathbf{0}_d, p_{\text{goal}}, \mathcal{X})$

•

In the remainder of this section, we generalize the circumcenter law in a number of ways: (i) we modify the constraint set by imposing bounds on the control inputs and by relaxing the connectivity constraint as much as possible, while maintaining connectivity guarantees; and (ii) we implement the circumcenter law on two distinct communication graphs. Let us note that many of these generalized circumcenter laws can also be implemented as relative-sensing control laws; we do not present the details in the interest of brevity.

Circumcenter law with control bounds and relaxed connectivity constraints

First, we assume that the agents have a compact input space $U = \overline{B}(\mathbf{0}_d, u_{\max})$, with $u_{\max} \in \mathbb{R}_{>0}$. Additionally, we adopt the relaxed $\mathcal{G}$-connectivity constraint sets as follows. Let $\mathcal{G}$ be a proximity graph that is spatially distributed over $\mathcal{G}_{\text{disk}}(r)$ and that has the same connected components as $\mathcal{G}_{\text{disk}}(r)$; examples include $\mathcal{G}_{\text{RN}} \cap \mathcal{G}_{\text{disk}}(r)$, $\mathcal{G}_{\text{G}} \cap \mathcal{G}_{\text{disk}}(r)$ and $\mathcal{G}_{\text{LD}}(r)$. Recall the $\mathcal{G}$-connectivity constraint set from Definition 4.7. Combining the relaxed connectivity constraint and the control magnitude bound, we redefine the control function in the CIRCUMCENTER law to be:

`function` ctl(p, y)
% Includes control bound and relaxed $\mathcal{G}$-connectivity constraint

1: $p_{\text{goal}} := \text{CC}(\{p\} \cup \{p_{\text{rcvd}} \mid \text{for all non-null } p_{\text{rcvd}} \in y\})$
2: $\mathcal{X} := \mathcal{X}_{\text{disk},\mathcal{G}}(p, \{p_{\text{rcvd}} \mid \text{for all non-null } p_{\text{rcvd}} \in y\}) \cap \overline{B}(p, u_{\max})$
3: **return** $\text{fti}(p, p_{\text{goal}}, \mathcal{X}) - p$

Second, the circumcenter law can be implemented also on robotic networks with different proximity graphs. For example, we can implement the circumcenter algorithm without any change on the following network.

Circumcenter law on the limited Delaunay graph

We consider the same set of physical agents as in the $\mathcal{S}_{\text{disk}}$. For $r \in \mathbb{R}_{>0}$, we adopt as communication graph the *r-limited Delaunay graph* $\mathcal{G}_{\text{LD}}(r)$, described in Section 2.2. These data define the uniform robotic network

$\mathcal{S}_{\text{LD}} = (\{1, \ldots, n\}, \mathcal{R}, E_{\text{LD}})$, as described in Example 3.4. On this network, we implement the CIRCUMCENTER law without any change, that is, with the same message-generation and control function as we did for the implementation on the network $\mathcal{S}_{\text{disk}}$.

Parallel circumcenter law on the ∞-disk graph

We consider the network $\mathcal{S}_{\infty\text{-disk}}$ of first-order robots in $\mathbb{R}^d$ connected according to the $\mathcal{G}_{\infty\text{-disk}}(r)$ graph, see Example 3.4. For this network we define the PLL-CRCMCNTR law, that we denoted by $\mathcal{CC}_{\text{PLL-CRCMCNTR}}$, by designing d decoupled circumcenter laws running in parallel on each coordinate axis of $\mathbb{R}^d$. As before, this law is uniform and static. What is remarkable, however, is that no constraint is required to maintain connectivity, see Exercise E4.4.

The parallel circumcenter of the set S, denoted by $\text{PCC}(S)$, is the center of the smallest axis-aligned rectangle containing S. In other words, $\text{PCC}(S)$ is the component-wise circumcenter of S, see Figure 4.7. We state the parallel

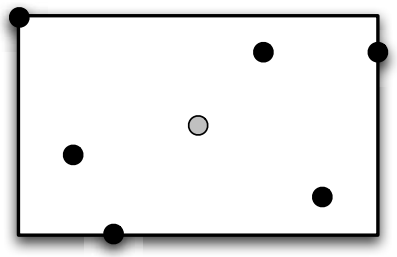


Fig. 4.7. The grey-colored point is the parallel circumcenter of the collection of black-colored points.

circumcenter law as follows.

`Robotic Network:` $\mathcal{S}_{\infty\text{-disk}}$ with discrete-time motion model (4.1) in $\mathbb{R}^d$, with absolute sensing of own position, and with communication range r in L^∞-metric

`Distributed Algorithm:` PLL-CRCMCNTR

`Alphabet:` $\mathbb{A} = \mathbb{R}^d \cup \{\texttt{null}\}$

`function` $\text{msg}(p, i)$

1: **return** p

`function` $\text{ctl}(p, y)$

1: $p_{\text{goal}} := \text{PCC}(\{p\} \cup \{p_{\text{rcvd}} \mid \text{for all non-null } p_{\text{rcvd}} \in y\})$
2: **return** $p_{\text{goal}} - p$

4.3.3 Correctness and complexity of circumcenter laws

In this section we characterize the convergence and complexity properties of the circumcenter law and of its variations. The robustness of the circumcenter control and communication laws can be characterized with respect to link failures; see Cortés et al. [2006].

The following theorem summarizes the results known in the literature about the asymptotic properties of the circumcenter law.

Theorem 4.17 (Correctness of the circumcenter laws). *For $d \in \mathbb{N}$, $r \in \mathbb{R}_{>0}$ and $\varepsilon \in \mathbb{R}_{>0}$, the following statements hold:*

(i) on the network $\mathcal{S}_{\text{disk}}$, the law $\mathcal{CC}_{\text{CIRCUMCENTER}}$ (with control magnitude bounds and relaxed $\mathcal{G}$-connectivity constraints) achieves the exact rendezvous task $\mathcal{T}_{\text{rndzvs}}$;

(ii) on the network $\mathcal{S}_{\text{LD}}$, the law $\mathcal{CC}_{\text{CIRCUMCENTER}}$ achieves the ε-rendezvous task $\mathcal{T}_{\varepsilon\text{-rndzvs}}$; and

(iii) on the network $\mathcal{S}_{\infty\text{-disk}}$, the law $\mathcal{CC}_{\text{PLL-CRCMCNTR}}$ achieves the exact rendezvous task $\mathcal{T}_{\text{rndzvs}}$.

Furthermore, the evolutions of $(\mathcal{S}_{\text{disk}}, \mathcal{CC}_{\text{CIRCUMCENTER}})$, of $(\mathcal{S}_{\text{LD}}, \mathcal{CC}_{\text{CIRCUMCENTER}})$, and of $(\mathcal{S}_{\infty\text{-disk}}, \mathcal{CC}_{\text{PLL-CRCMCNTR}})$ have the following properties:

(iv) if any two agents belong to the same connected component of the respective communication graph at $\ell \in \mathbb{Z}_{\geq 0}$, then they continue to belong to the same connected component for all subsequent times $k \geq \ell$; and

(v) for each evolution, there exists $P^* = (p_1^*, \ldots, p_n^*) \in (\mathbb{R}^d)^n$ such that:

a) the evolution asymptotically approaches P^*, and

b) for each $i, j \in \{1, \ldots, n\}$, either $p_i^* = p_j^*$, or $\|p_i^* - p_j^*\|_2 > r$ (for the networks $\mathcal{S}_{\text{disk}}$ and $\mathcal{S}_{\text{LD}}$) or $\|p_i^* - p_j^*\|_\infty > r$ (for the network $\mathcal{S}_{\infty\text{-disk}}$).

The results on $\mathcal{S}_{\text{disk}}$ appeared originally in [Ando et al., 1999]. The proof for the results on $\mathcal{S}_{\text{LD}}$ is provided by Cortés et al. [2006] and on $\mathcal{S}_{\infty\text{-disk}}$ are contained in [Martínez et al., 2007b].

Next, we analyze the time complexity of $\mathcal{CC}_{\text{CIRCUMCENTER}}$. As we see next, the complexity of $\mathcal{CC}_{\text{CIRCUMCENTER}}$ differs dramatically when applied to robotic networks with different communication graphs. We provide complete results for the networks $\mathcal{S}_{\text{disk}}$ and $\mathcal{S}_{\text{LD}}$ only for the case $d = 1$.

Theorem 4.18 (Time complexity of circumcenter laws). *For $r \in \mathbb{R}_{>0}$ and $\varepsilon \in]0, 1[$, the following statements hold:*

(i) on the network $\mathcal{S}_{\text{disk}}$, evolving on the real line $\mathbb{R}$ (i.e., with $d = 1$), $\text{TC}(\mathcal{T}_{\text{rndzvs}}, \mathcal{CC}_{\text{CIRCUMCENTER}}) \in \Theta(n)$;

(ii) on the network $\mathcal{S}_{\text{LD}}$, evolving on the real line $\mathbb{R}$ (i.e., with $d = 1$), $\text{TC}(\mathcal{T}_{(r\varepsilon)\text{-rndzvs}}, \mathcal{CC}_{\text{CIRCUMCENTER}}) \in \Theta(n^2 \log(n\varepsilon^{-1}))$; and

(iii) on the network $\mathcal{S}_{\infty\text{-disk}}$, evolving on Euclidean space (i.e., with $d \in \mathbb{N}$), $\text{TC}(\mathcal{T}_{\text{rndzvs}}, \mathcal{CC}_{\text{PLL-CRCMCNTR}}) \in \Theta(n)$.

The proof of this result is contained in [Martínez et al., 2007b].

Remark 4.19 (Analysis in higher dimensions). The results in Theorem 4.18(i) and (ii) induce lower bounds on the time complexity of the circumcenter law in higher dimensions. Indeed, for arbitrary $d \geq 1$, we have

(i) on the network $\mathcal{S}_{\text{disk}}$, $\text{TC}(\mathcal{T}_{\text{rndzvs}}, \mathcal{CC}_{\text{CIRCUMCENTER}}) \in \Omega(n)$;
(ii) on the network $\mathcal{S}_{\text{LD}}$, $\text{TC}(\mathcal{T}_{(r\varepsilon)\text{-rndzvs}}, \mathcal{CC}_{\text{CIRCUMCENTER}}) \in \Omega(n^2 \log(n\varepsilon^{-1}))$.

We have performed extensive numerical simulations for the case $d = 2$ and the network $\mathcal{S}_{\text{disk}}$. We run the algorithm starting from generic initial configurations (where, in particular, robots' positions are not aligned) contained in a bounded region of $\mathbb{R}^2$. We have consistently obtained that the time complexity to achieve $\mathcal{T}_{\text{rndzvs}}$ with $\mathcal{CC}_{\text{CIRCUMCENTER}}$ starting from these initial configurations is independent of the number of robots. This leads us to conjecture that initial configurations where all robots are aligned (equivalently, the 1-dimensional case) give rise to the worst possible performance of the algorithm. In other words, we conjecture that, for $d \geq 2$, $\text{TC}(\mathcal{T}_{\text{rndzvs}}, \mathcal{CC}_{\text{CIRCUMCENTER}}) = \Theta(n)$. •

Remark 4.20 (Congestion effects). As discussed in Remark 3.8, one way of incorporating congestion effects into the network operation is to assume that the parameters of the physical components of the network depend upon the number of robots. For instance, by assuming that the communication range decreases with the number of robots. Theorem 4.18 presents an alternative, equivalent way of looking at congestion: the results hold under the assumption that the communication range is constant, but allow for the diameter of the initial network configuration (the maximum inter-agent distance) to grow unbounded with the number of robots. •

4.3.4 Circumcenter law in nonconvex environments

In this section we adapt the circumcenter algorithm to work on networks in planar nonconvex allowable environments. Throughout the section, we only consider the case of a compact allowable nonconvex environment Q contracted into Q_δ for a small positive δ. We present the algorithm in two formats: for the communication-based network $\mathcal{S}_{\text{vis-disk}}$ described in Example 3.6 and for the relative-sensing network $\mathcal{S}^{\text{rs}}_{\text{vis-disk}}$ described in Example 3.16.

We modify the circumcenter algorithm in three ways: first, we adopt the connectivity constraints described in the previous section for range-limited line-of-sight links; second, we further restrict the robot motion to remain inside the relative convex hull of the sensed robot positions; and third, we move towards the circumcenter of the constraint set, instead of the circumcenter of the neighbors positions. The algorithm details are as follows.

`Robotic Network:` $\mathcal{S}_{\text{vis-disk}}$ with discrete-time motion model (4.1), absolute sensing of own position and of Q_δ, and communication range r within line of sight ($\mathcal{G}_{\text{vis-disk},Q_\delta}$)

`Distributed Algorithm:` CIRCUMCENTER
`Alphabet:` $\mathbb{A} = \mathbb{R}^2 \cup \{\texttt{null}\}$

`function` $\mathrm{msg}(p, i)$

1: **return** p

`function` $\mathrm{ctl}(p, y)$

1: $\mathcal{X}_1 := \mathcal{X}_{\text{vis-disk}}(p, \{p_{\text{rcvd}} \mid \text{for all non-null } p_{\text{rcvd}} \in y\}; Q_\delta)$
2: $\mathcal{X}_2 := \mathrm{rco}(\{p\} \cup \{p_{\text{rcvd}} \mid \text{for all non-null } p_{\text{rcvd}} \in y\}; \mathrm{Vi}(p; Q_\delta))$
3: $p_{\text{goal}} := \mathrm{CC}(\mathcal{X}_1 \cap \mathcal{X}_2)$
4: **return** $\mathrm{fti}(p, p_{\text{goal}}, \overline{B}(p, u_{\max})) - p$

Next, we present the relative sensing version; recall that $p_i^{[i]} = \mathbf{0}_2$ and that, as discussed in Section 3.2.3 about the evolution of a relative sensing network with environment sensors, y_{env} denotes the environment measurement provided by the range-limited visibility sensor.

`Relative Sensing Network:` $\mathcal{S}^{\text{rs}}_{\text{vis-disk}}$ with motion model (4.2) in Q_δ,
no communication, relative sensing for robot i given by:
robot measurements y contains $p_i^{[j]} \in \mathrm{Vi}_{\text{disk}}(\mathbf{0}_2; (Q_\delta)_i)$ for $j \neq i$
environment sensing is $y_{\text{env}} = \mathrm{Vi}_{\text{disk}}(\mathbf{0}_2; (Q_\delta)_i)$

`Distributed Algorithm:` RELATIVE-SENSING CIRCUMCENTER
`function` $\mathrm{ctl}(y, y_{\text{env}})$

1: $\mathcal{X}_1 := \mathcal{X}_{\text{vis-disk}}(\mathbf{0}_2, \{p_{\text{snsd}} \mid \text{for all non-null } p_{\text{snsd}} \in y\}; y_{\text{env}})$
2: $\mathcal{X}_2 := \mathrm{rco}(\{\mathbf{0}_2\} \cup \{p_{\text{snsd}} \mid \text{for all non-null } p_{\text{snsd}} \in y\}; y_{\text{env}})$
3: $p_{\text{goal}} := \mathrm{CC}(\mathcal{X}_1 \cap \mathcal{X}_2)$
4: **return** $\mathrm{fti}(\mathbf{0}_2, p_{\text{goal}}, \overline{B}(\mathbf{0}_2, u_{\max}))$

Theorem 4.21 (Correctness of the circumcenter law in nonconvex environments). *For $\delta > 0$, let Q_δ be a contraction of a compact allowable nonconvex environment Q. For $r \in \mathbb{R}_{>0}$ and $\varepsilon \in \mathbb{R}_{>0}$, on the network $\mathcal{S}_{\text{vis-disk}}$, the law $\mathcal{CC}_{\text{CIRCUMCENTER}}$ (with control magnitude bounds) achieves the ε-rendezvous task $\mathcal{T}_{\varepsilon\text{-rndzvs}}$. Furthermore, the evolution has the following properties:*

(i) if any two agents belong to the same connected component of the graph $\mathcal{G}_{\text{vis-disk},Q_\delta}$ at $\ell \in \mathbb{Z}_{\geq 0}$, then they continue to belong to the same connected component for all subsequent times $k \geq \ell$; and
(ii) there exists $P^* = (p_1^*, \ldots, p_n^*) \in Q_\delta^n$ such that:
a) the evolution asymptotically approaches P^*, and
b) for each $i, j \in \{1, \ldots, n\}$, either $p_i^* = p_j^*$, or p_i^* and p_j^* are not within range-limited line of sight

The proof of this result can be found in [Ganguli et al., 2007b]. The complexity of CIRCUMCENTER law in nonconvex environments has not been characterized. However, note that the evolution from any initial configuration

such that $\mathcal{G}_{\text{vis},Q_\delta}$ is complete is also an evolution of the CIRCUMCENTER law discussed in Section 4.3.2, and hence Theorem 4.18(i) induces a lower bound on the time complexity.

4.4 Notes

The rendezvous problem and the circumcenter algorithm were originally introduced by Ando et al. [1999], where a convergence analysis is also provided. The circumcenter algorithm has been extended to other control policies, including asynchronous implementations, in [Lin et al., 2007a,b]. The circumcenter algorithm has been extended beyond planar problems to arbitrary dimensions in [Cortés et al., 2006], where its robustness properties are also characterized. Variations of the circumcenter law in the presence of noise and sensor errors are studied in [Martínez, 2008]. The continuous-time version of the circumcenter law, with no connectivity constraints, is analyzed in [Lin et al., 2007c]. Continuous-time control laws for groups of robots with simple first-order dynamics and unicycle dynamics are proposed in [Lin et al., 2004, 2005, Dimarogonas and Kyriakopoulos, 2007]. In these works, the inter-robot topology is time-dependent and assumed a priori to be connected at all times. Rendezvous under communication quantization is studied in [Fagnani et al., 2004, Carli and Bullo, 2007]. Rendezvous for unicycle robots with minimal sensing capabilities is studied by Yu et al. [2008b]. Relationships with classic curve-shortening flows are studied by Smith et al. [2007].

Rendezvous has also been studied within the computer science literature, where the problem is referred to as the "gathering" or point formation problem. Flocchini et al. [1999], Suzuki and Yamashita [1999] study the point formation problem under the assumption that each robot is capable of sensing all other robots. Flocchini et al. [2005] propose asynchronous algorithms to solve the gathering problem, and Agmon and Peleg [2006] study the solvability of the problem in the presence of faulty robots.

Multirobot rendezvous with line-of-sight sensors is considered in [Roy and Dudek, 2001], where solutions are proposed based on the exploration of the unknown environment and the selection of appropriate rendezvous points at pre-specified times. The paper [Hayes et al., 2003] also consider rendezvous at a specified location for visually-guided agents, but the proposed solution requires each agent to have knowledge of the location of all other agents. The problem of computing a multirobot rendezvous point in polyhedral surfaces made of triangular faces is considered in [Lanthier et al., 2005]. The perimeter minimizing algorithm presented by Ganguli et al. [2007b] solves the rendezvous problem for sensor-based networks with line-of-sight limited-range sensors in nonconvex environments.

Regarding the connectivity maintenance problem, a number of works have addressed the problem of designing a coordination algorithm that achieves a general, non-specified task while preserving connectivity. The centralized

solution proposed in [Zavlanos and Pappas, 2005] allows for a general range of agent motions. The distributed solution presented by [Savla et al., 2007b] gives connectivity maintaining constraints for second-order control systems with input magnitude bounds. A distributed algorithm to perform graph rearrangements that preserve the connectivity is presented in [Schuresko and Cortés, 2007]. Connectivity problems have been studied also in other contexts. Langbort and Gupta [2007] study the impact of the connectivity of the interconnection topology in a class of network optimization problems. Spanos and Murray [2005] generate connectivity-preserving motions between pairs of formations. Ji and Egerstedt [2007] design Laplacian-based control laws to solve formation control problems while preserving connectivity. Various works have focused on designing the network motion so that some desired measure of connectivity (e.g., algebraic connectivity) is maximized under position constraints. Boyd [2006], de Gennaro and Jadbabaie [2006] consider convex constraints, while Kim and Mesbahi [2006] deal with a class of nonconvex constraints. Zavlanos and Pappas [2007b] use potential fields to maximize algebraic connectivity.

A continuous-time version of the averaging control and communication law is also known as the Hegselmann-Krause model for "opinion dynamics under bounded confidence"; see [Hegselmann and Krause, 2002, Lorenz, 2007]. In this model, each agent may change its opinion by averaging it with that of neighbors' who are in an ε-confidence area. In other words, the difference between the agent opinion and its neighbors' should be bounded by ε. A similar model where the communication between agents is random is the Deffuant-Weisbuch model, inspired by a model of dissemination of culture; see Deffuant et al. [2000], Axelrod [1997].

4.5 Proofs

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

4.5.1 Proof of Theorem 4.15

Proof. One can easily prove that, along the evolution of the network, the ordering of the agents is preserved, i.e., the inequality $p^{[i]} \leq p^{[j]}$ is preserved at the next time step. However, links between agents are not necessarily preserved (see, e.g., Figure 4.5). Indeed, connected components may split along the evolution. However, merging events do not occur. Consider two contiguous connected components C_1 and C_2 of $\mathcal{G}_{\text{disk}}(r)$, with C_1 to the left of C_2. By definition, the rightmost agent in the component C_1 and the leftmost agent in the component C_2 are at a distance strictly larger than r. Now, by executing the algorithm, they can only but increase that distance, since the rightmost agent in C_1 will move to the left, and the leftmost agent in C_2 will move to the right. Therefore, connected components do not merge.

Consider first the case of an initial configuration of the network for which the communication graph remains connected throughout the evolution. Without loss of generality, assume that the agents are ordered from left to right according to their identifier, that is, $p^{[1]}(0) \leq \cdots \leq p^{[n]}(0)$. Let $\alpha \in \{3, \ldots, n\}$ have the property that agents $\{2, \ldots, \alpha - 1\}$ are neighbors of agent 1, and agent α is not. (If instead all agents are within an interval of length r, then rendezvous is achieved in 1 time instant, and the statement in theorem is easily seen to be true.) Note that we can assume that agents $\{2, \ldots, \alpha - 1\}$ are also neighbors of agent α. If this is not the case, then those agents that are neighbors of agent 1 and not of agent α, rendezvous with agent 1 at the next time instant. At the time instant $\ell = 1$, the new updated positions satisfy

$$p^{[1]}(1) = \frac{1}{\alpha - 1} \sum_{k=1}^{\alpha-1} p^{[k]}(0),$$
$$p^{[\gamma]}(1) \in \Big[\frac{1}{\alpha} \sum_{k=1}^{\alpha} p^{[k]}(0), *\Big], \ \gamma \in \{2, \ldots, \alpha - 1\},$$

where $*$ denotes a certain unimportant point.

Now, we show that

$$p^{[1]}(\alpha - 1) - p^{[1]}(0) \geq \frac{r}{\alpha(\alpha - 1)}. \tag{4.3}$$

Let us first show the inequality for $\alpha = 3$. Because of the assumption that the communication graph remains connected, agent 2 is still a neighbor of agent 1 at the time instant $\ell = 1$. Therefore $p^{[1]}(2) \geq \frac{1}{2}(p^{[1]}(1) + p^{[2]}(1))$, and from here we deduce

$$\begin{aligned} p^{[1]}(2) - p^{[1]}(0) &\geq \frac{1}{2}\big(p^{[2]}(1) - p^{[1]}(0)\big) \\ &\geq \frac{1}{2}\big(\frac{1}{3}\big(p^{[1]}(0) + p^{[2]}(0) + p^{[3]}(0)\big) - p^{[1]}(0)\big) \geq \frac{1}{6}\big(p^{[3]}(0) - p^{[1]}(0)\big) \geq \frac{r}{6}. \end{aligned}$$

Let us now proceed by induction. Assume that inequality (4.3) is valid for $\alpha - 1$, and let us prove it for α. Consider first the possibility when at the time instant $\ell = 1$, the agent $\alpha - 1$ is still a neighbor of agent 1. In this case, $p^{[1]}(2) \geq \frac{1}{\alpha-1} \sum_{k=1}^{\alpha-1} p^{[k]}(1)$, and from here we deduce

$$\begin{aligned} p^{[1]}(2) - p^{[1]}(0) &\geq \frac{1}{\alpha - 1}\Big(p^{[\alpha-1]}(1) - p^{[1]}(0)\Big) \geq \frac{1}{\alpha - 1}\Big(\frac{1}{\alpha}\sum_{k=1}^{\alpha} p^{[k]}(0) - p^{[1]}(0)\Big) \\ &\geq \frac{1}{\alpha(\alpha - 1)}\Big(p^{[\alpha]}(0) - p^{[1]}(0)\Big) \geq \frac{r}{\alpha(\alpha - 1)}, \end{aligned}$$

which in particular implies (4.3). Consider then the case when agent $\alpha - 1$ is not a neighbor of agent 1 at the time instant $\ell = 1$. Let $\beta < \alpha$ such that

agent $\beta-1$ is a neighbor of agent 1 at $\ell=1$, but agent β is not. Since $\beta<\alpha$, we have by induction $p^{[1]}(\beta)-p^{[1]}(1)\geq\frac{r}{\beta(\beta-1)}$. From here, we deduce that $p^{[1]}(\alpha-1)-p^{[1]}(0)\geq\frac{r}{\alpha(\alpha-1)}$.

It is clear that after $\ell_1=\alpha-1$, we could again consider two complementary cases (either agent 1 has all others as neighbors or not) and repeat the same argument once again. In that way, we would find ℓ_2 such that the distance traveled by agent 1 after ℓ_2 rounds would be lower bounded by $\frac{2r}{n(n-1)}$. Repeating this argument iteratively, the worst possible case is one in which agent 1 keeps moving to the right and, at each time step, there is always another agent which is not a neighbor. Since the diameter of the initial condition P_0 is upper bounded by $(n-1)r$, in the worst possible situation, there exists some time ℓ_k such that $\frac{kr}{(n-1)n}=O(r(n-1))$. This implies that $k=O((n-1)^2n)$. Now we can upper bound the total convergence time ℓ_k by $\ell_k=\sum_{i=1}^{k}\alpha_i-k\leq k(n-1)$, where we have used that $\alpha_i\leq n$ for all $i\in\{1,\dots,n\}$. From here we see that $\ell_k=O((n-1)^3n)$ and hence, we deduce that in $O(n(n-1)^3)$ time instants there cannot be any agent which is not a neighbor of the agent 1. Hence, all agents rendezvous at the next time instant. Consequently,

$$\mathrm{TC}(\mathcal{T}_{\mathrm{rndzvs}},\mathcal{CC}_{\textsc{averaging}},P_0)=O(n(n-1)^3).$$

Finally, for a general initial configuration P_0, because there are a finite number of agents, only a finite number of splittings (at most $n-1$) of the connected components of the communication graph can take place along the evolution. Therefore, we conclude $\mathrm{TC}(\mathcal{T}_{\mathrm{rndzvs}},\mathcal{CC}_{\textsc{averaging}})=O(n^5)$.

Let us now prove the lower bound. Consider an initial configuration $P_0\in\mathbb{R}^n$ where all agents are positioned in increasing order according to their identity, and exactly at a distance r apart, say $p^{[i+1]}(0)-p^{[i]}(0)=r$, $i\in\{1,\dots,n-1\}$. Assume for simplicity that n is odd - when n is even, one can reason in an analogous way. Because of the symmetry of the initial condition, in the first time step, only agents 1 and n move. All the remaining agents remain in their position because it coincides with the average of its neighbors' position and its own. At the second time step, only agents 1, 2, $n-1$ and n move, and the others remain static because of the symmetry. Applying this idea iteratively, one deduces that the time step when agents $\frac{n-1}{2}$ and $\frac{n+3}{2}$ move for the first time is lower bounded by $\frac{n-1}{2}$. Since both agents have still at least a neighbor (agent $\frac{n+1}{2}$), the task $\mathcal{T}_{\mathrm{rndzvs}}$ has not been achieved yet at this time step. Therefore, $\mathrm{TC}(\mathcal{T}_{\mathrm{rndzvs}},\mathcal{CC}_{\textsc{averaging}},P_0)\geq\frac{n-1}{2}$, and the result follows. ■

4.5.2 Proof of Theorem 4.17

Proof. We divide the proof of the theorem into three groups, one per network.

STEP 1: Facts on $(\mathcal{S}_{\mathrm{disk}},\mathcal{CC}_{\textsc{circumcenter}})$. Fact (iv) for $(\mathcal{S}_{\mathrm{disk}},\mathcal{CC}_{\textsc{circumcenter}})$ is a direct consequence of the control function definition of the CIRCUMCENTER law and Lemma 4.8.

Let us show fact (i). Because $\mathcal{G}$ has the same connected components as $\mathcal{G}_{\text{disk}}(r)$, fact (iv) implies that the number of connected components of $\mathcal{G}_{\text{disk}}(r)$ can only but decrease. In other words, the number of agents in each of the connected components of $\mathcal{G}_{\text{disk}}(r)$ is non-decreasing. Since there is a finite number of agents, there must exist ℓ_0 such that the identity of the agents in each connected component of $\mathcal{G}_{\text{disk}}(r)$ is fixed for all $\ell \geq \ell_0$ (i.e., no more agents are added to the connected component afterwards). In what follows, without loss of generality, we assume that there is only one connected component after ℓ_0, i.e., the graph is connected (if this is not the case, then the same argument follows through for each connected component).

Our strategy to prove that the law $\mathcal{CC}_{\text{CIRCUMCENTER}}$ (with control magnitude bounds and relaxed $\mathcal{G}$-connectivity constraints) achieves the exact rendezvous task $\mathcal{T}_{\text{rndzvs}}$ consists of two steps:

(a) we first define a set-valued dynamical system $((\mathbb{R}^d)^n, (\mathbb{R}^d)^n, T)$ such that the evolutions of $(\mathcal{S}_{\text{disk}}, \mathcal{CC}_{\text{CIRCUMCENTER}})$ starting from an initial configuration where $\mathcal{G}_{\text{disk}}(r)$ is connected are contained in the set of evolutions of the set-valued dynamical system;
(b) we then establish that any evolution of $((\mathbb{R}^d)^n, (\mathbb{R}^d)^n, T)$ converges to a point in $\text{diag}((\mathbb{R}^d)^n)$ (the point might be different for different evolutions).

This strategy is analogous to the discussion regarding the Overapproximation Lemma for time-dependent systems in Section 1.2.4.

Let as perform (a). Given a connected graph G with vertices $\{1, \dots, n\}$, let us consider the constraint sets and goal points defined with respect to G. In other words, given $P = (p_1, \dots, p_n) \in (\mathbb{R}^d)^n$, define for each $i \in \{1, \dots, n\}$,

$$(p_{\text{goal}})_i := \text{CC}(\{p_i\} \cup \{p_j \mid j \in \mathcal{N}_G(i)\}),$$
$$\mathcal{X}_i := \bigcap \left\{ \overline{B}\left(\tfrac{p_i + p_j}{2}, \tfrac{r_i(P)}{2}\right) \mid j \in \mathcal{N}_G(i) \right\} \cap \overline{B}(p_i, u_{\max}),$$

where $r_i(P) = \max\{r, \max\{\|p_i - p_j\|_2 \mid j \in \mathcal{N}_G(i)\}\}$. Since two neighbors according to G can be arbitrarily far from each other in $\mathbb{R}^d$, we need to modify the definition of the constraint set with the radius $r_i(P)$ to prevent $\mathcal{X}_i$ from becoming empty. Note that if $\|p_i - p_j\|_2 \leq r$ for all $j \in \mathcal{N}_G(i)$, then $r_i(P) = r$, and therefore $\mathcal{X}_i = \mathcal{X}_{\text{disk},G}(p_i, P) \cap \overline{B}(p_i, u_{\max})$. It is also worth observing that both $(p_{\text{goal}})_i$ and $\mathcal{X}_i$ change continuously with $(p_1, \dots, p_n)$.

Define the map $\text{fti}_G : (\mathbb{R}^d)^n \to (\mathbb{R}^d)^n$ by

$$\text{fti}_G(p_1, \dots, p_n) = (\text{fti}(p_1, (p_{\text{goal}})_1, \mathcal{X}_1), \dots, \text{fti}(p_n, (p_{\text{goal}})_n, \mathcal{X}_n)).$$

One can think of fti_G as a circumcenter law where the neighboring relationships among the agents never change. Because fti is continuous, and $(p_{\text{goal}})_i$ and $\mathcal{X}_i$, $i \in \{1, \dots, n\}$, change continuously with $(p_1, \dots, p_n)$, we deduce that fti_G is continuous.

We are now ready to define a set-valued dynamical system $((\mathbb{R}^d)^n, (\mathbb{R}^d)^n, T)$ through the set-valued map $T : (\mathbb{R}^d)^n \rightrightarrows (\mathbb{R}^d)^n$ given by

$$T(p_1, \dots, p_n) = \{\text{fti}_G(p_1, \dots, p_n) \mid G \text{ is a strongly connected digraph}\}.$$

Note that the evolution of the CIRCUMCENTER law using a proximity graph such as $\mathcal{G}_{\text{disk}}(r)$ is just one of the multiple evolutions described by this set-valued map. This concludes (a).

Let us now perform (b). To characterize the convergence properties of the set-valued dynamical system, we use the LaSalle Invariance Principle in Theorem 1.18. With the notation of this result, we select $W = (\mathbb{R}^d)^n$. This set is clearly strongly positively invariant for $((\mathbb{R}^d)^n, (\mathbb{R}^d)^n, T)$.

Closedness of the set-valued map. Since fti_G is continuous for each digraph G and there is a finite number of strongly connected digraphs on the vertices $\{1, \dots, n\}$, Exercise E1.4 implies that T is closed.

Common Lyapunov function. Define the function $V_{\text{diam}} : (\mathbb{R}^d)^n \to \mathbb{R}_{\geq 0}$ by

$$V_{\text{diam}}(P) = \max\{\|p_i - p_j\| \mid i, j \in \{1, \dots, n\}\}.$$

With a slight abuse of notation, we denote by $\text{co}(P)$ the convex hull of $\{p_1, \dots, p_n\} \subset \mathbb{R}^d$. Note that $V_{\text{diam}}(P) = \text{diam}(\text{co}(P))$. The function V_{diam} has the following properties.

(i) V_{diam} is continuous and invariant under permutations of its arguments;

(ii) $V_{\text{diam}}(P) = 0$ if and only if $P \in \text{diag}((\mathbb{R}^d)^n)$, where recall that $\text{diag}((\mathbb{R}^d)^n) = \{(p_1, \dots, p_n) \in (\mathbb{R}^d)^n \mid p^{[i]} = \cdots = p^{[n]} \in \mathbb{R}^d\}$ denotes the diagonal set of $(\mathbb{R}^d)^n$. This fact is an immediate consequence of the fact that, given a set $S \subset (\mathbb{R}^d)^n$, $\text{diam}(\text{co}(S)) = 0$ if and only if S is a singleton;

(iii) V_{diam} is non-increasing along T on $(\mathbb{R}^d)^n$. Consider a finite set of points $S \in \mathbb{F}((\mathbb{R}^d)^n)$ and let $\text{CC}(S)$ be its circumcenter. From Lemma 2.2(i), we have $\text{CC}(S) \in \text{co}(S)$. Therefore, for any strongly connected digraph G, we have that $\text{co}(\text{fti}_G(P)) \subset \text{co}(P)$ for any $P \in (\mathbb{R}^d)^n$. Since for any two sets S_1, $S_2 \subset (\mathbb{R}^d)^n$ such that $\text{co}(S_2) \subset \text{co}(S_2)$ it holds that $V_{\text{diam}}(S_2) \leq V_{\text{diam}}(S_1)$, then $V_{\text{diam}}(\text{fti}_G(P)) \leq V_{\text{diam}}(P)$ for any strongly connected digraph G, which implies that V_{diam} is non-increasing along T on $(\mathbb{R}^d)^n$.

Bounded evolutions. Consider any initial condition $(p_1(0), \dots, p_n(0)) \in (\mathbb{R}^d)^n$. For any strongly connected digraph, G, we have

$$\text{fti}_G(p_1(\ell), \dots, p_n(\ell)) \in \text{co}(p_1(0), \dots, p_n(0)),$$

for all $\ell \in \mathbb{Z}_{\geq 0}$. Therefore, any evolution of the set-valued dynamical system $((\mathbb{R}^d)^n, (\mathbb{R}^d)^n, T)$ is bounded.

Characterization of the invariant set. By the LaSalle Invariance for set-valued dynamical systems in Theorem 1.18, any evolution with initial condition in $W = (\mathbb{R}^d)^n$ approaches the largest weakly positively invariant set M contained in

$$\{P \in (\mathbb{R}^d)^n \mid \exists\, P' \in T(P) \text{ such that } V_{\text{diam}}(P') = V_{\text{diam}}(P)\}.$$

We show that $M = \text{diag}((\mathbb{R}^d)^n)$. Clearly, $\text{diag}((\mathbb{R}^d)^n) \subset M$. To prove the other inclusion, we reason by contradiction. Assume $P \in M \setminus \text{diag}((\mathbb{R}^d)^n)$,

and therefore $V_{\mathrm{diam}}(P) > 0$. Let G be a strongly connected digraph and consider $\mathrm{fti}_G(P)$. For each $i \in \{1, \dots, n\}$, we distinguish two cases depending on whether p_i is or is not a vertex of $\mathrm{co}(P)$. If $p_i \notin \mathrm{Ve}(\mathrm{co}(P))$, then Lemma 2.2(i) implies that $\mathrm{fti}(p_i, (p_{\mathrm{goal}})_i, \mathcal{X}_i) \in \mathrm{co}(P) \setminus \mathrm{Ve}(\mathrm{co}(P))$.

If $p_i \in \mathrm{Ve}(\mathrm{co}(P))$, then we must take into consideration the possibility of having more than one agent located at the same point. If the location of all the neighbors of i in the digraph G coincides with p_i, then agent i will not move, and hence $\mathrm{fti}(p_i, (p_{\mathrm{goal}})_i, \mathcal{X}_i) \in \mathrm{Ve}(\mathrm{co}(P))$. However, we can show that the application of fti_G strictly decreases the number of agents located at p_i. Let us denote this number by N_i, i.e.,

$$N_i = |\{j \in \{1, \dots, n\} \mid p_j = p_i \text{ and } p_j \in \{p_1, \dots, p_n\}\}|.$$

Since the digraph G is strongly connected, there must exist at least an agent located at p_i with a neighbor which is not located at p_i (otherwise, all agents would be at p_i, which is a contradiction). In other words, there exist $i_*, j \in \{1, \dots, n\}$ such that $p_{i_*} = p_i$, $p_j \neq p_i$, and $j \in \mathcal{N}_G(i_*)$. By Lemma 2.2(i), we have that $(p_{\mathrm{goal}})_{i_*} \in \mathrm{co}(P) \setminus \mathrm{Ve}(\mathrm{co}(P))$, and therefore $(p_{\mathrm{goal}})_{i_*} \neq p_{i_*}$. Combining this with the fact that

$$\{p_i\} \cup \{p_j \mid j \in \mathcal{N}_G(i)\} \subset \overline{B}(p_{i_*}, r_{i_*}(P)),$$

we can apply Lemma 2.2(ii) to ensure that $]p_{i_*}, (p_{\mathrm{goal}})_{i_*}[$ has nonempty intersection with $\mathcal{X}_{i_*}$. Therefore, $\mathrm{fti}(p_{i_*}, (p_{\mathrm{goal}})_{i_*}, \mathcal{X}_{i_*}) \in \mathrm{co}(P) \setminus \mathrm{Ve}(\mathrm{co}(P))$, and the number N_i of agents located at p_i decreases at least by one with the application of fti_G.

Next, we show that, after a finite number of steps, no agents remains at the location p_i. Define $N = \max\{N_i \mid p_i \in \mathrm{Ve}(\mathrm{co}(P))\} < n - 1$. Then all agents in the configuration $\mathrm{fti}_{G_1}(\mathrm{fti}_{G_2}(\dots \mathrm{fti}_{G_N}(P)))$ are contained in $\mathrm{co}(P) \setminus \mathrm{Ve}(\mathrm{co}(P))$, for any collection of strongly connected directed graphs $G_1, \dots, G_N$. Therefore, $\mathrm{diam}(\mathrm{co}(\mathrm{fti}_{G_1}(\mathrm{fti}_{G_2}(\dots \mathrm{fti}_{G_N}(P))))) < \mathrm{diam}(\mathrm{co}(P))$, which contradicts the fact that M is weakly invariant.

Point convergence. We have proved that any evolution of $((\mathbb{R}^d)^n, (\mathbb{R}^d)^n, T)$ approaches the set $\mathrm{diag}((\mathbb{R}^d)^n)$. To conclude the proof, let us show that the convergence of each trajectory is to a point, rather that to the diagonal set. Let $\{P(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\}$ be an evolution of the set-valued dynamical system. Since the sequence is contained in the compact set $\mathrm{co}(P(0))$, there exists a convergent subsequence $\{P(\ell_k) \mid k \in \mathbb{Z}_{\geq 0}\}$, i.e., there exists $p \in \mathbb{R}^d$ such that

$$\lim_{k \to +\infty} P(\ell_k) = (p, \dots, p). \tag{4.4}$$

Let us show that the whole sequence $\{P(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\}$ converges to $(p, \dots, p)$. Because of (4.4), for any $\varepsilon > 0$, there exists k_0 such that for $k \geq k_0$ one has $\mathrm{co}(P(\ell_k)) \subset \overline{B}(p, \varepsilon/\sqrt{n})$. From this, we deduce that $\mathrm{co}(P(\ell)) \subset \overline{B}(p, \varepsilon/\sqrt{n})$ for all $\ell \geq \ell_{k_0}$, which in turn implies that $\|P(\ell) - (p, \dots, p)\|_2 \leq \varepsilon$ for all $\ell \geq \ell_{k_0}$, as claimed. This concludes (b).

(a) and (b) imply that any evolution of $(\mathcal{S}_{\text{disk}}, \mathcal{CC}_{\text{CIRCUMCENTER}})$ starting from an initial configuration where $\mathcal{G}_{\text{disk}}(r)$ is connected converges to a point in $\text{diag}((\mathbb{R}^d)^n)$. To conclude the proof of fact (i), we only need to establish that this convergence is in finite time. This last fact is a consequence of Exercise E4.5.

Fact (v) for $(\mathcal{S}_{\text{disk}}, \mathcal{CC}_{\text{CIRCUMCENTER}})$ is a consequence of facts (i) and (iv).

STEP 2: Facts on $(\mathcal{S}_{\text{LD}}, \mathcal{CC}_{\text{CIRCUMCENTER}})$. The proof of facts (i), (iv), and (v) for $(\mathcal{S}_{\text{LD}}, \mathcal{CC}_{\text{CIRCUMCENTER}})$ is analogous to the proof of these facts for $(\mathcal{S}_{\text{disk}}, \mathcal{CC}_{\text{CIRCUMCENTER}})$, and we leave it to the reader.

STEP 3: Facts on $(\mathcal{S}_{\infty\text{-disk}}, \mathcal{CC}_{\text{PLL-CRCMCNTR}})$. From the expression for the control function of $\mathcal{CC}_{\text{PLL-CRCMCNTR}}$, we deduce that the evolution under $\mathcal{CC}_{\text{PLL-CRCMCNTR}}$ of the robotic network $\mathcal{S}_{\infty\text{-disk}}$ (in d dimensions) can be alternatively described as the evolution under $\mathcal{CC}_{\text{CIRCUMCENTER}}$ of d robotic networks $\mathcal{S}_{\text{disk}}$ in $\mathbb{R}$, see Exercise E4.4. Therefore, facts (i), (iv), and (v) for $(\mathcal{S}_{\infty\text{-disk}}, \mathcal{CC}_{\text{PLL-CRCMCNTR}})$ follow from facts (i), (iv), and (v) for $(\mathcal{S}_{\text{disk}}, \mathcal{CC}_{\text{CIRCUMCENTER}})$. ■

4.5.3 Proof of Theorem 4.18

Proof. Let $P_0 = (p^{[1]}(0), \dots, p^{[n]}(0)) \in \mathbb{R}^n$ denote the initial condition.
Fact (i). For $d = 1$, the connectivity constraints on each agent $i \in \{1, \dots, n\}$ imposed by the constraint set

$$\mathcal{X}_{\text{disk}}(p^{[i]}, \{p_{\text{rcvd}} \mid \text{for all non-null } p_{\text{rcvd}} \in y^{[i]}\}) \tag{4.5}$$

are superfluous. In other words, the goal configuration resulting from the evaluation by agent i of the control function of the CIRCUMCENTER law belongs to the constraint set in (4.5). Moreover, the order of the robots on the real line is preserved from one time step to the next one. Both observations are a consequence of Exercise E4.3.

Let us first establish the upper bound in fact (i). Consider the case when $\mathcal{G}_{\text{disk}}(r)$ is connected at P_0. Without loss of generality, assume that the agents are ordered from left to right according to their identifier, that is, $p^{[1]}(0) \leq \dots \leq p^{[n]}(0)$. Let $\alpha \in \{3, \dots, n\}$ have the property that agents $\{2, \dots, \alpha - 1\}$ are neighbors of agent 1, and agent α is not. (If instead all agents are within an interval of length r, then rendezvous is achieved after one time step, and the upper bound in fact (i) is easily seen to be true.) Figure 4.8 presents an illustration of the definition of α. Note that we can assume that agents $\{2, \dots, \alpha - 1\}$ are also neighbors of agent α. If this is not the case, then those agents that are neighbors of agent 1 and not of agent α, rendezvous with agent 1 after one time step. At the time instant $\ell = 1$, the new updated positions satisfy

$$p^{[1]}(1) = \frac{p^{[1]}(0) + p^{[\alpha-1]}(0)}{2},$$
$$p^{[\gamma]}(1) \in \left[\frac{p^{[1]}(0) + p^{[\alpha]}(0)}{2}, \frac{p^{[1]}(0) + p^{[\gamma]}(0) + r}{2}\right],$$

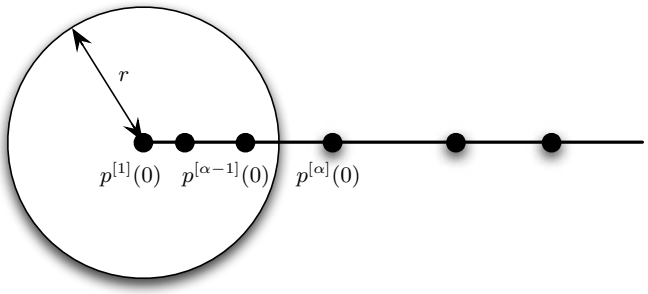


Fig. 4.8. Definition of $\alpha \in \{3, \dots, n\}$ for an initial network configuration.

for $\gamma \in \{2, \dots, \alpha - 1\}$. These equalities imply that $p^{[1]}(1) - p^{[1]}(0) = \frac{1}{2}\big(p^{[\alpha-1]}(0) - p^{[1]}(0)\big) \leq \frac{1}{2}r$. Analogously, we deduce $p^{[1]}(2) - p^{[1]}(1) \leq \frac{1}{2}r$, and therefore

$$p^{[1]}(2) - p^{[1]}(0) \leq r. \tag{4.6}$$

On the other hand, from $p^{[1]}(2) \in \big[\frac{1}{2}\big(p^{[1]}(1) + p^{[\alpha-1]}(1)\big), *\big]$ (where the symbol $*$ represents a certain unimportant point in $\mathbb{R}$), we deduce

$$\begin{aligned} p^{[1]}(2) - p^{[1]}(0) &\geq \frac{1}{2}\big(p^{[1]}(1) + p^{[\alpha-1]}(1)\big) - p^{[1]}(0) \\ &\geq \frac{1}{2}\big(p^{[\alpha-1]}(1) - p^{[1]}(0)\big) \geq \frac{1}{2}\Big(\frac{p^{[1]}(0) + p^{[\alpha]}(0)}{2} - p^{[1]}(0)\Big) \\ &= \frac{1}{4}\big(p^{[\alpha]}(0) - p^{[1]}(0)\big) \geq \frac{1}{4}r\,. \end{aligned} \tag{4.7}$$

Inequalities (4.6) and (4.7) mean that, after at most two time steps, agent 1 has traveled an amount larger than $r/4$. In turn this implies that

$$\frac{1}{r}\operatorname{diam}(\operatorname{co}(P_0)) \leq \operatorname{TC}(\mathcal{T}_{\text{rndzvs}}, \mathcal{CC}_{\text{CIRCUMCENTER}}, P_0) \leq \frac{4}{r}\operatorname{diam}(\operatorname{co}(P_0)).$$

If $\mathcal{G}_{\text{disk}}(r)$ is not connected at P_0, note that along the network evolution, the connected components of the r-disk graph do not change. Using the previous characterization on the amount traveled by the leftmost agent of each connected component in at most two time steps, we deduce

$$\operatorname{TC}(\mathcal{T}_{\text{rndzvs}}, \mathcal{CC}_{\text{CIRCUMCENTER}}, P_0) \leq \frac{4}{r}\max_{C \in \mathcal{C}(P_0)} \operatorname{diam}(\operatorname{co}(C)),$$

where $\mathcal{C}(P_0)$ denotes the collection of connected components of $\mathcal{G}_{\text{disk}}(r)$ at P_0. The connectedness of each $C \in \mathcal{C}(P_0)$ implies that $\operatorname{diam}(\operatorname{co}(C)) \leq (n-1)r$, and therefore $\operatorname{TC}(\mathcal{T}_{\text{rndzvs}}, \mathcal{CC}_{\text{CIRCUMCENTER}}) \in O(n)$.

The lower bound in fact (i) is established by considering $P_0 \in \mathbb{R}^n$ such that $p^{[i+1]}(0) - p^{[i]}(0) = r$, $i \in \{1, \dots, n-1\}$. For this configuration, we have $\operatorname{diam}(\operatorname{co}(P_0)) = (n-1)r$, and therefore $\operatorname{TC}(\mathcal{T}_{\text{rndzvs}}, \mathcal{CC}_{\text{CIRCUMCENTER}}, P_0) \geq n-1$. *Fact (ii).* In the r-limited Delaunay graph, two agents on the line that are at most at a distance r from each other are neighbors if and only if there are no other agents between them. Also, note that the r-limited Delaunay graph and the r-disk graph have the same connected components, cf. Theorem 2.9. A similar argument to the one used in the proof of fact (i) above guarantees that the connectivity constraints imposed by the constraint sets $\mathcal{X}_{\text{disk}}(p^{[i]}, \{p_{\text{rcvd}} \mid \text{for all non-null } p_{\text{rcvd}} \in y^{[i]}\})$ are again superfluous.

Consider first the case when $\mathcal{G}_{\text{LD}}(r)$ is connected at P_0. Note that this is equivalent to $\mathcal{G}_{\text{disk}}(r)$ being connected at P_0. Without loss of generality, assume that the agents are ordered from left to right according to their identifier, that is, $p^{[1]}(0) \leq \cdots \leq p^{[n]}(0)$. The evolution of the network under $\mathcal{CC}_{\text{CIRCUMCENTER}}$ can then be described as the discrete-time dynamical system

$$\begin{aligned}
p^{[1]}(\ell+1) &= \frac{1}{2}(p^{[1]}(\ell) + p^{[2]}(\ell)),\\
p^{[2]}(\ell+1) &= \frac{1}{2}(p^{[1]}(\ell) + p^{[3]}(\ell)),\\
&\vdots\\
p^{[n-1]}(\ell+1) &= \frac{1}{2}(p^{[n-2]}(\ell) + p^{[n]}(\ell)),\\
p^{[n]}(\ell+1) &= \frac{1}{2}(p^{[n-1]}(\ell) + p^{[n]}(\ell)).
\end{aligned}$$

Note that this evolution respects the ordering of the agents. Equivalently, we can write $P(\ell+1) = A\,P(\ell)$, where $A \in \mathbb{R}^{n\times n}$ is the matrix given by

$$A = \begin{bmatrix}
\frac{1}{2} & \frac{1}{2} & 0 & \dots & \dots & 0\\
\frac{1}{2} & 0 & \frac{1}{2} & \dots & \dots & 0\\
0 & \frac{1}{2} & 0 & \frac{1}{2} & \dots & 0\\
\vdots & & \ddots & \ddots & \ddots & \vdots\\
0 & \dots & \dots & \frac{1}{2} & 0 & \frac{1}{2}\\
0 & \dots & \dots & 0 & \frac{1}{2} & \frac{1}{2}
\end{bmatrix}.$$

Note that $A = \operatorname{ATrid}_n^+\left(\frac{1}{2}, 0\right)$, as defined in Section 1.5.4. Theorem 1.75(i) implies that, for $P_{\text{ave}} = \frac{1}{n}\mathbf{1}_n^T P_0$, we have that $\lim_{\ell\to+\infty} P(\ell) = P_{\text{ave}}\mathbf{1}_n$, and that the maximum time required for $\|P(\ell) - P_{\text{ave}}\mathbf{1}_n\|_2 \leq \eta\|P_0 - P_{\text{ave}}\mathbf{1}_n\|_2$ (over all initial conditions in $\mathbb{R}^n$) is $\Theta\big(n^2 \log \eta^{-1}\big)$. (Note that this also implies that agents rendezvous at the location given by the average of their initial positions. In other words, the asymptotic rendezvous position for this case can be expressed in closed form, as opposed to the case with the r-disk graph.)

Next, let us convert the contraction inequality on 2-norms into an appropriate inequality on ∞-norms. Note that $\operatorname{diam}(\operatorname{co}(P_0)) \leq (n-1)r$ because

$\mathcal{G}_{\mathrm{LD}}(r)$ is connected at P_0. Therefore

$$\|P_0 - P_{\mathrm{ave}}\mathbf{1}\|_\infty = \max_{i\in\{1,\dots,n\}} |p^{[i]}(0) - P_{\mathrm{ave}}| \leq |p^{[1]}(0) - p^{[n]}(0)| \leq (n-1)r.$$

For ℓ of order $n^2 \log \eta^{-1}$, we use this bound on $\|P_0 - P_{\mathrm{ave}}\mathbf{1}\|_\infty$ and the basic inequalities $\|v\|_\infty \leq \|v\|_2 \leq \sqrt{n}\|v\|_\infty$ for all $v \in \mathbb{R}^n$, to obtain

$$\begin{aligned}\|P(\ell) - P_{\mathrm{ave}}\mathbf{1}\|_\infty &\leq \|P(\ell) - P_{\mathrm{ave}}\mathbf{1}\|_2 \leq \eta\|P_0 - P_{\mathrm{ave}}\mathbf{1}\|_2 \\ &\leq \eta\sqrt{n}\|P_0 - P_{\mathrm{ave}}\mathbf{1}\|_\infty \leq \eta\sqrt{n}(n-1)r.\end{aligned}$$

This means that $(r\varepsilon)$-rendezvous is achieved for $\eta\sqrt{n}(n-1)r = r\varepsilon$, that is, in time $O(n^2 \log \eta^{-1}) = O(n^2 \log(n\varepsilon^{-1}))$.

Next, we show the lower bound. Consider the unit-length eigenvector $\mathbf{v}_n = \sqrt{\frac{2}{n+1}}(\sin\frac{\pi}{n+1}, \dots, \sin\frac{n\pi}{n+1})^T \in \mathbb{R}^n$ of $\mathrm{Trid}_{n-1}(\frac{1}{2}, 0, \frac{1}{2})$ corresponding to the largest singular value $\cos(\frac{\pi}{n})$. For $\mu = \frac{-1}{10\sqrt{2}} r n^{5/2}$, we then define the initial condition

$$P_0 = \mu P_+ \begin{bmatrix} 0 \\ \mathbf{v}_{n-1} \end{bmatrix} \in \mathbb{R}^n.$$

One can show that $p^{[i]}(0) < p^{[i+1]}(0)$ for $i \in \{1, \dots, n-1\}$, that $P_{\mathrm{ave}} = 0$, and that $\max\{p^{[i+1]}(0) - p^{[i]}(0) \mid i \in \{1, \dots, n-1\}\} \leq r$. Using Lemma 1.77 and because $\|w\|_\infty \leq \|w\|_2 \leq \sqrt{n}\|w\|_\infty$ for all $w \in \mathbb{R}^n$, we compute

$$\begin{aligned}\|P_0\|_\infty &= \frac{rn^{5/2}}{10\sqrt{2}} \left\| P_+ \begin{bmatrix} 0 \\ \mathbf{v}_{n-1} \end{bmatrix} \right\|_\infty \geq \frac{rn^2}{10\sqrt{2}} \left\| P_+ \begin{bmatrix} 0 \\ \mathbf{v}_{n-1} \end{bmatrix} \right\|_2 \\ &\geq \frac{rn}{10\sqrt{2}} \|\mathbf{v}_{n-1}\|_2 = \frac{rn}{10\sqrt{2}}.\end{aligned}$$

The trajectory $P(\ell) = (\cos(\frac{\pi}{n}))^\ell P_0$ therefore satisfies

$$\|P(\ell)\|_\infty = \left(\cos\left(\frac{\pi}{n}\right)\right)^\ell \|P_0\|_\infty \geq \frac{rn}{10\sqrt{2}} \left(\cos\left(\frac{\pi}{n}\right)\right)^\ell.$$

Therefore, $\|P(\ell)\|_\infty$ is larger than $\frac{1}{2}r\varepsilon$ so long as $\frac{1}{10\sqrt{2}} n (\cos(\frac{\pi}{n}))^\ell > \frac{1}{2}\varepsilon$, that is, so long as

$$\ell < \frac{\log(\varepsilon^{-1}n) - \log(5\sqrt{2})}{-\log\left(\cos(\frac{\pi}{n})\right)}.$$

Exercise E4.7 asks the reader to show that the asymptotics of this bound corresponds to the lower bound in fact (i).

Now consider the case when $\mathcal{G}_{\mathrm{LD}}(r)$ is not connected at P_0. Note that the connected components do not change along the network evolution. Therefore, the previous reasoning can be applied to each connected component. Since

the number of agents in each connected component is strictly less that n, the time complexity can only but improve. Therefore, we conclude that

$$\mathrm{TC}(\mathcal{T}_{\text{rndzvs}}, \mathcal{CC}_{\text{CIRCUMCENTER}}) \in \Theta(n^2 \log(n\varepsilon^{-1})).$$

Fact (iii). Recall from the proof of Theorem 4.5.2 that the evolution under $\mathcal{CC}_{\text{PLL-CRCMCNTR}}$ of the robotic network $\mathcal{S}_{\infty\text{-disk}}$ (in d dimensions) can be alternatively described as the evolution under $\mathcal{CC}_{\text{CIRCUMCENTER}}$ of d robotic networks $\mathcal{S}_{\text{disk}}$ in $\mathbb{R}$, see Exercise E4.4. Fact (iii) now follows from fact (i). ■

4.6 Exercises

E4.1 Prove Lemma 4.8.
Hint: *Use Lemma 4.2 and the fact that $\mathcal{G}$ and $\mathcal{G}_{\text{disk}}(r)$ have the same connected components.*

E4.2 Prove Lemma 4.11.
Hint: *Use Proposition 4.9.*

E4.3 **(Enforcing range-limited links is unnecessary for the circumcenter law on $\mathbb{R}$).** Let $\mathcal{P} = \{p_1, \dots, p_n\} \in \mathbb{F}(\mathbb{R})$. For $r \in \mathbb{R}_{>0}$, we work with the r-disk proximity graph $\mathcal{G}_{\text{disk}}(r)$ evaluated at $\mathcal{P}$. Let $i \in \{1, \dots, n\}$ and consider the circumcenter of the set comprised of p_i and of its neighbors,

$$(p_{\text{goal}})_i = \mathrm{CC}(\{p_i\} \cup \mathcal{N}_{\mathcal{G}_{\text{disk}}(r), p_i}(\mathcal{P})).$$

Show that:
(i) If p_i and p_j are neighbors in $\mathcal{G}_{\text{disk}}(r)$, then $(p_{\text{goal}})_i$ belongs to $\overline{B}(\frac{p_i+p_j}{2}, \frac{r}{2})$;
(ii) If p_i and p_j are neighbors in $\mathcal{G}_{\text{disk}}(r)$ and $p_i \leq p_j$, then $(p_{\text{goal}})_i \leq (p_{\text{goal}})_j$;
(iii) Discuss the implication of (i) and (ii) in the execution of the CIRCUMCENTER law on the 1-dimensional space $\mathbb{R}$.
Hint: *Express $(p_{\text{goal}})_i$ as a function of the position of the leftmost and rightmost points among the neighbors of p_i.*

E4.4 **(Enforcing range-limited links is unnecessary for the pll-crcmcntr law).** Let $\mathcal{P} = \{p_1, \dots, p_n\} \in \mathbb{F}(\mathbb{R}^d)$ and $r \in \mathbb{R}_{>0}$. For $k \in \{1, \dots, d\}$, denote by $\pi_k : \mathbb{R}^d \to \mathbb{R}$ the projection onto the kth component. Do the following:
(i) Show that p_i and p_j are neighbors in $\mathcal{G}_{\infty\text{-disk}}(r)$ if and only if, for all $k \in \{1, \dots, d\}$, $\pi_k(p_i)$ and $\pi_k(p_j)$ are neighbors in $\mathcal{G}_{\text{disk}}(r)$;
(ii) For $S \subset \mathbb{R}^d$, justify that the parallel circumcenter $\mathrm{PCC}(S) \in \mathbb{R}^d$ of S can be described as

$$\pi_k(\mathrm{PCC}(S)) = \mathrm{CC}(\pi_k(S)), \quad \text{for } k \in \{1, \dots, d\};$$

(iii) Use (i), (ii), and Exercise E4.3(i) to justify that no constraint is required to maintain connectivity of the ∞-disk graph in the PLL-CRCMCNTR law. In other words, show that if p_i and p_j are neighbors in $\mathcal{G}_{\infty\text{-disk}}(r)$, then $\mathrm{PCC}(\{p_i\} \cup \mathcal{N}_{\mathcal{G}_{\infty\text{-disk}}(r), p_i}(\mathcal{P}))$ and $\mathrm{PCC}(\{p_j\} \cup \mathcal{N}_{\mathcal{G}_{\infty\text{-disk}}(r), p_j}(\mathcal{P}))$ are also neighbors in $\mathcal{G}_{\infty\text{-disk}}(r)$.

E4.5 **(Finite-time convergence of the circumcenter law on $\mathcal{S}_{\text{disk}}$).** For $u_{\max}, r \in \mathbb{R}_{>0}$, let $a = \min\{u_{\max}, \frac{r}{2}\}$. Let $\mathcal{P} = \{p_1, \ldots, p_n\} \in \mathbb{F}(\mathbb{R}^d)$, and assume there exists $p \in \mathbb{R}^d$ such that

$$\{p_1, \ldots, p_n\} \subset \overline{B}(p, a).$$

Do the following:

(i) Show that $\mathcal{G}_{\text{disk}}(r)$ evaluated at $\{p_1, \ldots, p_n\}$ is the complete graph;
(ii) Justify why $\|p_i - \text{CC}(\{p_1, \ldots, p_n\})\|_2 \leq a$, for all $i \in \{1, \ldots, n\}$;
(iii) Show that $\text{CC}(\{p_1, \ldots, p_n\}) \in \mathcal{X}_{\text{disk}}(p_i, \mathcal{P}) \cap \overline{B}(p_i, u_{\max})$;
(iv) What is the evolution of $(\mathcal{S}_{\text{disk}}, \mathcal{CC}_{\text{CIRCUMCENTER}})$ (with control magnitude bounds) starting from $(p_1, \ldots, p_n)$?

E4.6 **(Variation of the circumcenter law).** Let $\mathcal{P} = \{p_1, \ldots, p_n\} \in \mathbb{F}(\mathbb{R}^d)$. For $r \in \mathbb{R}_{>0}$, we work with the r-disk proximity graph $\mathcal{G}_{\text{disk}}(r)$ evaluated at $\mathcal{P}$. For each $i \in \{1, \ldots, n\}$, consider the circumcenter of the set comprised of p_i and of the mid-points with its neighbors,

$$(p_{\text{goal}})_i = \text{CC}(\{p_i\} \cup \big\{ \frac{p_i + p_j}{2} \mid p_j \in \mathcal{N}_{\mathcal{G}_{\text{disk}}(r), p_i}(\mathcal{P}) \big\}).$$

Show that:

(i) If p_i and p_j are neighbors in $\mathcal{G}_{\text{disk}}(r)$, then $(p_{\text{goal}})_i$ and $(p_{\text{goal}})_j$ are neighbors in $\mathcal{G}_{\text{disk}}(r)$;
(ii) Use (i) to design a control and communication law on the network $\mathcal{S}_{\text{disk}}$ in $\mathbb{R}^d$ that, while not enforcing any connectivity constraints, preserves all neighboring relationships in $\mathcal{G}_{\text{disk}}(r)$ and achieves the ε-rendezvous task $\mathcal{T}_{\varepsilon\text{-rndzvs}}$;
(iii) Justify why the law designed in (ii) does not achieve the exact rendezvous task $\mathcal{T}_{\text{rndzvs}}$.

E4.7 **(Asymptotics of the lower bound in Theorem 4.18(ii)).** Show that, as $n \to +\infty$,

$$\frac{\log(\varepsilon^{-1} n) - \log(5\sqrt{2})}{-\log\big(\cos(\frac{\pi}{n})\big)} = \frac{n^2}{\pi^2}\big(\log(\varepsilon^{-1} n) - \log(5\sqrt{2})\big) + O(1).$$

Use this fact to complete the proof of the lower bound in the proof of Theorem 4.18(ii).
Hint: *Use the Taylor series expansion of* $\log(\cos(x))$ *at* $x = 0$.

5

Deployment

The aim of this chapter is to present various solutions to the deployment problem. The *deployment objective* is to optimally place a group of robots in an environment of interest. The approach taken here consists of identifying aggregate functions that measure the quality-of-deployment of a given network configuration and designing control and communication laws that optimize these measures.

The variety of algorithms presented in the chapter stems from two reasons. First, different solutions arise from the interplay between the spatially distributed character of the coordination algorithms and the limited sensing and communication capabilities of the robotic network. As an example, different solutions are feasible when agents have limited-range communication capabilities or when agents have omnidirectional line-of-sight visibility sensors. Second, there is no universal notion of deployment. Different scenarios give rise to different ways of measuring what constitutes a good deployment. As an example, a robotic network might follow a different strategy depending on whether it has information about areas of importance in the environment or not. In the first case, by incorporating the knowledge on the environment. In the second, by assuming a worst-case scenario, where important things can be happening precisely at the furthest away location from the network configuration.

Our exposition here follows [Cortés et al., 2004, 2005, Cortés and Bullo, 2005]. Our approach makes extensive use of the multicenter functions from geometric optimization introduced in Chapter 2. It is not difficult to synthesize continuous-time gradient ascent algorithms using the smoothness results presented in Section 2.3, and characterize its asymptotic convergence properties (as we ask the reader to do in Exercises E2.11 and E2.12). However, following the robotic network model of Chapter 3, we are interested in discrete-time algorithms. In general, gradient ascent algorithms implemented in discrete time require the selection of appropriate step sizes that guarantee the monotonic evolution of the objective function. This is usually accomplished via line search procedures, e.g., see Bertsekas and Tsitsiklis [1997]. In this chapter we

show that the special geometric properties of the multicenter functions and their gradients allow us to identify natural target locations for the robotic agents without the need to perform any line search.

The chapter is organized as follows. In Section 5.1 we formally define the notions of deployment via task maps and multicenter functions. In Section 5.2 we present motion coordination algorithms to achieve each deployment task. Specifically, we introduce control and communication laws based on various notions of geometric centers. In Section 5.2.3 we present convergence and complexity results on the proposed algorithms, along with simulations illustrating our analysis. Finally, in Section 5.5 we gather the proofs of the main results of the paper. Throughout the exposition, we make extensive use of proximity graphs, multicenter functions, and geometric optimization. The convergence and complexity analysis are based on the LaSalle Invariance Principle and on linear dynamical systems defined by Toeplitz matrices.

5.1 Problem statement

Here, we introduce various notions of deployment. Let $\mathcal{S} = (\{1, \dots, n\}, \mathcal{R}, E_{\text{cmm}})$ be a uniform robotic network, where the robots' physical state space is a (simple convex) polytope $Q \subset \mathbb{R}^d$ that describes an environment of interest. We define our notions of deployment relying upon the geometric optimization problems discussed in Section 2.3. Loosely speaking, we aim to deploy the robots in such a way as to optimize one of the multicenter functions, such as the the expected-value multicenter function $\mathcal{H}_{\text{exp}}$, the disk-covering multicenter function $\mathcal{H}_{\text{dc}}$, or the sphere-packing multicenter function $\mathcal{H}_{\text{sp}}$. Indeed these functions can be interpreted as quality-of-service measures for different scenarios. In order to formally define the task maps encoding the deployment objective, we take the following approach: since the optimizers of these measures are critical points, and these critical points are network configurations that make the gradients vanish, we define the task map to take the `true` value at these configurations.

The distortion, area, and mixed distortion-area deployment tasks

Here we define various notions of deployment originating from the expected-value multicenter function $\mathcal{H}_{\text{exp}}$. Recall the concepts of density and performance introduced in Section 2.3. Let $\phi : \mathbb{R}^d \to \mathbb{R}_{>0}$ be a density function on $\mathbb{R}^d$ with support Q. One can interpret ϕ as a function measuring the probability that some event takes place over the environment. Let $f : \mathbb{R}_{\geq 0} \to \mathbb{R}$ be a performance, i.e., a non-increasing and piecewise differentiable function possibly with finite jump discontinuities. Performance functions describe the utility of placing a robot at a certain distance from a location in the environment. Here, we restrict our attention to the cases $f(x) = -x^2$ (distortion problem), $f(x) = 1_{[0,a]}(x)$, $a \in \mathbb{R}_{>0}$ (area problem),

and $f(x) = -x^2\ 1_{[0,a]}(x) - a^2 \cdot 1_{]a,+\infty[}(x)$, with $a \in \mathbb{R}_{>0}$ (mixed distortion-area problem).

For $\varepsilon \in \mathbb{R}_{>0}$, the *$\varepsilon$-distortion deployment task* $\mathcal{T}_{\varepsilon\text{-distor-dply}} : Q^n \to \{\texttt{true}, \texttt{false}\}$ is defined as follows:

$$\mathcal{T}_{\varepsilon\text{-distor-dply}}(P) = \begin{cases} \texttt{true}, & \text{if } \left\|p^{[i]} - \mathrm{CM}_\phi(V^{[i]}(P))\right\|_2 \leq \varepsilon,\ i \in \{1, \dots, n\}, \\ \texttt{false}, & \text{otherwise,} \end{cases}$$

where $V^{[i]}(P)$ denotes the Voronoi cell of robot i, and $\mathrm{CM}_\phi(V^{[i]}(P))$ denotes its centroid computed according to ϕ, see Section 2.1. In other words, $\mathcal{T}_{\varepsilon\text{-distor-dply}}$ is `true` for those network configurations where each robot is sufficiently close to the centroid of its Voronoi cell. According to Theorem 2.17, centroidal Voronoi configurations are the critical points of the multicenter function $\mathcal{H}_{\text{distor}}$.

For $r, \varepsilon \in \mathbb{R}_{>0}$, the *$\varepsilon$-$r$-area deployment task* $\mathcal{T}_{\varepsilon\text{-}r\text{-area-dply}} : Q^n \to \{\texttt{true}, \texttt{false}\}$ is defined as follows:

$$\mathcal{T}_{\varepsilon\text{-}r\text{-area-dply}}(P) = \begin{cases} \texttt{true}, & \text{if } \left\| \int_{V^{[i]}(P) \cap \partial\overline{B}(p^{[i]}, \frac{r}{2})} \mathrm{n}_{\mathrm{out}, \overline{B}(p^{[i]}, \frac{r}{2})}(q)\phi(q)dq \right\|_2 \leq \varepsilon,\ i \in \{1, \dots, n\}, \\ \texttt{false}, & \text{otherwise.} \end{cases}$$

In other words, $\mathcal{T}_{\varepsilon\text{-}r\text{-area-dply}}$ is `true` for those network configurations where each agent is sufficiently close to a local maximum for the area of its $\frac{r}{2}$-limited Voronoi cell $V^{[i]}_{\frac{r}{2}}(P) = V^{[i]}(P) \cap \overline{B}(p^{[i]}, \frac{r}{2})$ at fixed $V^{[i]}(P)$. According to Theorem 2.17, $\frac{r}{2}$-limited area-centered Voronoi configurations are the critical points of the multicenter function $\mathcal{H}_{\text{area}, \frac{r}{2}}$.

For $r, \varepsilon \in \mathbb{R}_{>0}$, the *$\varepsilon$-$r$-distortion-area deployment task* $\mathcal{T}_{\varepsilon\text{-}r\text{-distor-area-dply}} : Q^n \to \{\texttt{true}, \texttt{false}\}$ is defined as follows:

$$\mathcal{T}_{\varepsilon\text{-}r\text{-distor-area-dply}}(P) = \begin{cases} \texttt{true}, & \text{if } \left\|p^{[i]} - \mathrm{CM}_\phi(V^{[i]}_{\frac{r}{2}}(P))\right\|_2 \leq \varepsilon,\ i \in \{1, \dots, n\}, \\ \texttt{false}, & \text{otherwise.} \end{cases}$$

In other words, $\mathcal{T}_{\varepsilon\text{-}r\text{-distor-area-dply}}$ is `true` for those network configurations where each robot is sufficiently close to the centroid of its $\frac{r}{2}$-limited Voronoi cell. According to Theorem 2.17, $\frac{r}{2}$-limited centroidal Voronoi configurations are the critical points of the multicenter function $\mathcal{H}_{\text{distor-area}, \frac{r}{2}}$.

The disk-covering and sphere-packing deployment tasks

Here we provide two additional notions of deployment based on the multicenter functions $\mathcal{H}_{\text{dc}}$ and $\mathcal{H}_{\text{sp}}$, respectively.

For $\varepsilon \in \mathbb{R}_{>0}$, the *$\varepsilon$-disk-covering deployment task* $\mathcal{T}_{\varepsilon\text{-dc-dply}} : Q^n \to \{\texttt{true}, \texttt{false}\}$ is defined as follows:

$$\mathcal{T}_{\varepsilon\text{-dc-dply}}(P) = \begin{cases} \texttt{true}, & \text{if } \|p^{[i]} - \mathrm{CC}(V^{[i]}(P))\|_2 \leq \varepsilon,\ i \in \{1,\dots,n\}, \\ \texttt{false}, & \text{otherwise}, \end{cases}$$

where $\mathrm{CC}(V^{[i]}(P))$ denotes the circumcenter of the Voronoi cell of robot i. In other words, $\mathcal{T}_{\varepsilon\text{-dc-dply}}$ is `true` for those network configurations where each robot is sufficiently close to the circumcenter of its Voronoi cell. According to Section 2.3.2, circumcenter Voronoi configurations are, under certain technical conditions, critical points of the multicenter function $\mathcal{H}_{\mathrm{dc}}$.

For $\varepsilon \in \mathbb{R}_{>0}$, the *$\varepsilon$-sphere-packing deployment task* $\mathcal{T}_{\varepsilon\text{-sp-dply}} : Q^n \to \{\texttt{true}, \texttt{false}\}$ is defined as follows:

$$\mathcal{T}_{\varepsilon\text{-sp-dply}}(P) = \begin{cases} \texttt{true}, & \text{if } \mathrm{dist}_2(p^{[i]}, \mathrm{IC}(V^{[i]}(P))) \leq \varepsilon,\ i \in \{1,\dots,n\}, \\ \texttt{false}, & \text{otherwise}, \end{cases}$$

where $\mathrm{IC}(V^{[i]}(P))$ denotes the incenter set of the Voronoi cell of robot i. In other words, $\mathcal{T}_{\varepsilon\text{-sp-dply}}$ is `true` for those network configurations where each robot is sufficiently close to the incenter set of its Voronoi cell. According to Section 2.3.3, incenter Voronoi configurations are, under certain technical conditions, critical points of the multicenter function $\mathcal{H}_{\mathrm{sp}}$.

5.2 Deployment algorithms

In this section we present algorithms that can be used by a robotic network to achieve the various notions of deployment introduced in the previous section. Throughout the discussion we mainly focus on the uniform networks $\mathcal{S}_{\mathrm{D}}$ and $\mathcal{S}_{\mathrm{LD}}$ of locally-connected first-order agents in a polytope $Q \subset \mathbb{R}^d$ with the Delaunay and r-limited Delaunay graphs as communication graphs, respectively; these robotic networks were introduced in Example 3.4.

All the laws presented in this chapter share a similar structure, that we loosely describe as follows.

> *[Informal description]* At each communication round each agent performs the following tasks: (i) it transmits its position and receives its neighbors' positions; (ii) it computes a notion of geometric center of its own cell determined according to some notion of partition of the environment. Between communication rounds, each robot moves toward this center.

The notions of geometric center and of partition of the environment are different for each algorithm, and specifically tailored to the deployment task at hand. Let us examine them for each case.

5.2.1 Geometric-center laws

We present control and communication laws defined on the network $\mathcal{S}_{\mathrm{D}}$. All the laws share in common the use of the notion of Voronoi partition of the

environment Q. We first introduce the VRN-CNTRD law, which makes use of the notion of centroid of a Voronoi cell. We then propose two sets of variations to this law. On the one hand, we present the VRN-CNTRD-DYNMCS law, which implements the same centroid strategy on a network of planar vehicles. On the other hand, we introduce the VRN-CRCMCNTR and VRN-NCNTR laws, which instead make use of the notions of circumcenter and incenter of a Voronoi cell, respectively.

Voronoi-centroid control and communication law

Here we define the VRN-CNTRD control and communication law for the network $\mathcal{S}_{\mathrm{D}}$, that we denote by $\mathcal{CC}_{\text{VRN-CNTRD}}$. This law was introduced by Cortés et al. [2004]. We formulate the algorithm using the description model of Chapter 3. The law is uniform, static, and data-sampled, with standard message-generation function.

`Robotic Network:` $\mathcal{S}_{\mathrm{D}}$ with discrete-time motion model (4.1) in Q, with absolute sensing of own position

`Distributed Algorithm:` VRN-CNTRD

`Alphabet:` $\mathbb{A} = \mathbb{R}^d \cup \{\texttt{null}\}$

`function` $\mathrm{msg}(p, i)$

1: **return** p

`function` $\mathrm{ctl}(p, y)$

1: $V := Q \cap \big(\bigcap\{H_{p,p_{\mathrm{rcvd}}} \mid \text{for all non-null } p_{\mathrm{rcvd}} \in y\}\big)$

2: **return** $\mathrm{CM}_\phi(V) - p$

Recall that $H_{p,x}$ is the half-space of points q in $\mathbb{R}^d$ with the property that $\|q-p\|_2 \leq \|q-x\|_2$. Since the centroid of a Voronoi cell belongs to the cell itself, the robots never leave the set Q or, in other words, the set Q^n is positively invariant with respect to the control and communication law $\mathcal{CC}_{\text{VRN-CNTRD}}$. Moreover, note that the direction of motion specified by the control function ctl coincides with the gradient of the distortion multicenter function $\mathcal{H}_{\mathrm{distor}}$. Hence, this law prescribes a gradient ascent strategy for each robot that, as we will show later, monotonically optimizes $\mathcal{H}_{\mathrm{distor}}$.

Voronoi-centroid law on planar vehicles

Next, we provide an interesting variation of the VRN-CNTRD law defined on the network $\mathcal{S}_{\mathrm{vehicles}}$ introduced in Example 3.5. Accordingly, we adopt the continuous-time motion model for the unicycle vehicle

$$\begin{aligned} \dot{p}^{[i]}(t) &= v^{[i]}(t)\,(\cos(\theta^{[i]}(t)),\, \sin(\theta^{[i]}(t))), \\ \dot{\theta}^{[i]}(t) &= \omega^{[i]}(t), \quad i \in \{1, \ldots, n\}, \end{aligned} \tag{5.1}$$

where we assume that forward and angular velocities are upper bounded. We refer to this law as VRN-CNTRD-DYNMCS, and we denote it by $\mathcal{CC}_{\text{VRN-CNTRD-DYNMCS}}$. The law was introduced by Cortés et al. [2004], is uniform and static, but not data-sampled.

`Robotic Network:` $\mathcal{S}_{\text{vehicles}}$ with motion model (5.1) in Q, with absolute sensing of own position

`Distributed Algorithm:` VRN-CNTRD-DYNMCS

`Alphabet:` $\mathbb{A} = \mathbb{R}^2 \cup \{\texttt{null}\}$

`function` $\text{msg}((p, \theta), i)$

1: **return** p

`function` $\text{ctl}((p, \theta), (p_{\text{smpld}}, \theta_{\text{smpld}}), y)$

1: $V := Q \cap \left(\bigcap \{ H_{p_{\text{smpld}}, p_{\text{rcvd}}} \mid \text{for all non-null } p_{\text{rcvd}} \in y \} \right)$
2: $v := -k_{\text{prop}}(\cos\theta, \sin\theta) \cdot (p - \text{CM}_\phi(V))$
3: $\omega := 2k_{\text{prop}} \arctan \frac{(-\sin\theta, \cos\theta) \cdot (p - \text{CM}_\phi(V))}{(\cos\theta, \sin\theta) \cdot (p - \text{CM}_\phi(V))}$
4: **return** (v, ω)

This algorithm is illustrated in Figure 5.1.

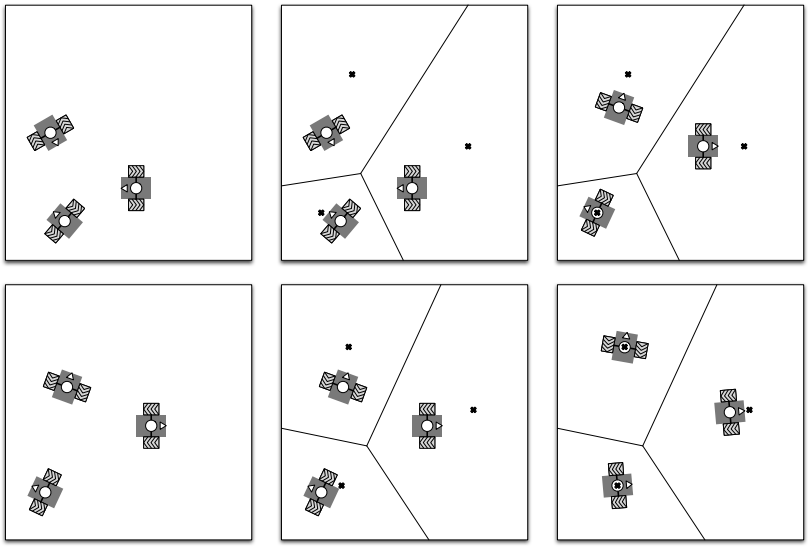


Fig. 5.1. Illustration of the execution of VRN-CNTRD-DYNMCS. Each row of plots represents an iteration of the law. At each round, each agent first computes its Voronoi cell, then determines the centroid, and then moves towards it.

In the above description, we require the feedback gain k_{prop} to belong to the interval $]0, \frac{1}{\max\{\pi, \text{diam}(Q)\}}]$. This guarantees that the controls v, ω in

the definition of ctl belong to the closed interval $[-1, 1]$, and are therefore implementable in the unicycle and the differential drive robot models.

Remark 5.1 (Vehicles with general dynamics). The definition of the control function ctl is based on the stabilizing feedback law by Astolfi [1999]. The robot position p is guaranteed to monotonically approach the target position $\mathrm{CM}_\phi(V)$. In general, the VRN-CNTRD-DYNMCS law can be implemented over a network of vehicles with arbitrary dynamics, as long as these vehicles are capable of strictly decreasing the distance to any specified position in Q in the time intervals between communication rounds. Under this assumption, the VRN-CNTRD-DYNMCS law enjoys the same convergence guarantees as the VRN-CNTRD law, see Theorem 5.5 below. •

Voronoi-circumcenter control and communication law

Here we define the VRN-CRCMCNTR control and communication law for the network $\mathcal{S}_\mathrm{D}$, that we denote by $\mathcal{CC}_{\text{VRN-CRCMCNTR}}$. This law was introduced by Cortés and Bullo [2005]. The law is uniform, static, and data-sampled, with standard message-generation function.

`Robotic Network:` $\mathcal{S}_\mathrm{D}$ with discrete-time motion model (4.1) in Q, with absolute sensing of own position

`Distributed Algorithm:` VRN-CRCMCNTR

`Alphabet:` $\mathbb{A} = \mathbb{R}^d \cup \{\texttt{null}\}$

`function` $\mathrm{msg}(p, i)$

1: **return** p

`function` $\mathrm{ctl}(p, y)$

1: $V := Q \cap \big(\bigcap\{H_{p,p_\mathrm{rcvd}} \mid \text{for all non-null } p_\mathrm{rcvd} \in y\}\big)$

2: **return** $\mathrm{CC}(V) - p$

Since the circumcenter of a Voronoi cell belongs to the cell itself, the set Q^n is positively invariant with respect to the control and communication law $\mathcal{CC}_{\text{VRN-CRCMCNTR}}$. From a geometric perspective, this law makes sense as a strategy to optimize the disk-covering multicenter function $\mathcal{H}_\mathrm{dc}$. From Section 2.1.3, for fixed V, the circumcenter location minimizes the cost given by the maximum distance to all points in V. From Section 2.3.2, $\mathcal{H}_\mathrm{dc}$ can be expressed (2.14) as the maximum over the network of each robot's individual cost.

Voronoi-incenter control and communication law

Here we define the VRN-NCNTR control and communication law for the network $\mathcal{S}_\mathrm{D}$, that we denote by $\mathcal{CC}_{\text{VRN-NCNTR}}$. This law was introduced by Cortés and Bullo [2005]. The law is uniform, static, and data-sampled, with standard message-generation function.

`Robotic Network:` $\mathcal{S}_{\mathrm{D}}$ with discrete-time motion model (4.1) in Q, with absolute sensing of own position

`Distributed Algorithm:` VRN-CRCMCNTR

`Alphabet:` $\mathbb{A} = \mathbb{R}^d \cup \{\texttt{null}\}$

`function` $\mathrm{msg}(p, i)$

1: **return** p

`function` $\mathrm{ctl}(p, y)$

1: $V := Q \cap \left(\bigcap\{H_{p,p_{\mathrm{rcvd}}} \mid \text{for all non-null } p_{\mathrm{rcvd}} \in y\}\right)$
2: **return** $x \in \mathrm{IC}(V) - p$

Since the incenter set of a Voronoi cell belongs to the cell itself, the set Q^n is positively invariant with respect to the control and communication law $\mathcal{CC}_{\text{VRN-NCNTR}}$. From a geometric perspective, this law makes sense as a strategy to optimize the sphere-packing multicenter function $\mathcal{H}_{\mathrm{sp}}$. From Section 2.1.3, for fixed V, the incenter locations maximize the cost given by the minimum distance to the boundary of V. From Section 2.3.3, $\mathcal{H}_{\mathrm{sp}}$ can be expressed (2.17) as the minimum over the network of each robot's individual cost.

Remark 5.2 ("Move-toward-furthest-vertex" and "away-from-closest-neighbor" coordination algorithms). Consider the coordination algorithm where, at each time step, each robot moves towards the furthest-away vertex of its own Voronoi cell. Alternatively, consider the coordination algorithm where, at each time step, each robot moves away from its closest neighbor. Both coordination algorithms define maps which depend discontinuously on the robots' positions. Cortés and Bullo [2005] study the asymptotic behavior of these laws, and show that the "move-toward-furthest-vertex" algorithm monotonically optimizes the multicenter function $\mathcal{H}_{\mathrm{dc}}$, while the "away-from-closest-neighbor" algorithm monotonically optimizes the multicenter function $\mathcal{H}_{\mathrm{sp}}$. •

5.2.2 Geometric-center laws with range-limited interactions

Next, we present two control and communication laws on the network $\mathcal{S}_{\mathrm{LD}}$. Both laws prescribe a geometric centering strategy for each robot and accomplish specific forms of expected-value optimization. The LMTD-VRN-NRML law optimizes the area multicenter function $\mathcal{H}_{\mathrm{area},\frac{r}{2}}$, while the LMTD-VRN-CNTRD law optimizes the mixed distortion-area multicenter function $\mathcal{H}_{\text{distor-area},\frac{r}{2}}$.

Limited-Voronoi-normal control and communication law

Here we define the LMTD-VRN-NRML control and communication law for the network $\mathcal{S}_{\mathrm{LD}}$. This law was introduced by Cortés et al. [2005]. The LMTD-VRN-NRML law, that we denote by $\mathcal{CC}_{\text{LMTD-VRN-NRML}}$, uses the notion of $\frac{r}{2}$-limited Voronoi partition inside Q. The law is uniform, static, and data-sampled, with standard message-generation function.

`Robotic Network:` $\mathcal{S}_{\text{LD}}$ with discrete-time motion model (4.1) in Q, with absolute sensing of own position, and with communication range r

`Distributed Algorithm:` LMTD-VRN-NRML
`Alphabet:` $\mathbb{A} = \mathbb{R}^d \cup \{\texttt{null}\}$

`function` $\text{msg}(p, i)$
1: **return** p

`function` $\text{ctl}(p, y)$
1: $V := Q \cap \left(\bigcap\{H_{p,p_{\text{rcvd}}} \mid \text{for all non-null } p_{\text{rcvd}} \in y\}\right)$
2: $v := \int_{V \cap \partial \overline{B}(p,\frac{r}{2})} \text{n}_{\text{out},\overline{B}(p,\frac{r}{2})}(q)\phi(q)dq$
3: $\lambda_* := \max\{\lambda \mid \delta \mapsto \int_{V \cap \overline{B}(p+\delta v,\frac{r}{2})} \phi(q)dq \text{ is strictly increasing on } [0, \lambda]\}$
4: **return** $\lambda_* v$

Note that the direction of motion v specified by the control function ctl coincides with the gradient of the multicenter function $\mathcal{H}_{\text{area},\frac{r}{2}}$. The parameter λ_* corresponds to performing a line search procedure along the direction v.

The control function has the property that the point $p+\text{ctl}(p, y)$ is guaranteed to be in the interior of V. This can be justified by noting that for fixed V, the gradient of the function $p \to \int_{V \cap \overline{B}(p,\frac{r}{2})} \phi(q)dq$ at points in the boundary of V is non-vanishing and points toward the interior of V (cf. Exercise E2.4). As a consequence, the line search procedure terminates before reaching the boundary of V. This discussion guarantees that the set Q^n is positively invariant with respect to the control and communication law $\mathcal{CC}_{\text{LMTD-VRN-NRML}}$.

Limited-Voronoi-centroid control and communication law

Here we define the LMTD-VRN-CNTRD control and communication law for the network $\mathcal{S}_{\text{LD}}$. This law was introduced by Cortés et al. [2005]. The LMTD-VRN-CNTRD law, that we denote by $\mathcal{CC}_{\text{LMTD-VRN-CNTRD}}$, uses the notion of $\frac{r}{2}$-limited Voronoi partition inside Q and of centroid of the individual $\frac{r}{2}$-limited Voronoi cells. The law is uniform, static, and data-sampled, with standard message-generation function.

`Robotic Network:` $\mathcal{S}_{\text{LD}}$ with discrete-time motion model (4.1) in Q, with absolute sensing of own position, and with communication range r

`Distributed Algorithm:` LMTD-VRN-CNTRD
`Alphabet:` $\mathbb{A} = \mathbb{R}^d \cup \{\texttt{null}\}$

`function` $\text{msg}(p, i)$
1: **return** p

`function` $\text{ctl}(p, y)$
1: $V := Q \cap \overline{B}(p, \frac{r}{2}) \cap \left(\bigcap\{H_{p,p_{\text{rcvd}}} \mid \text{for all non-null } p_{\text{rcvd}} \in y\}\right)$

2: **return** $\mathrm{CM}_\phi(V) - p$

The centroid of a $\frac{r}{2}$-limited Voronoi cell belongs to the cell itself, and this fact guarantees that the set Q^n is positively invariant with respect to the control and communication law $\mathcal{CC}_{\text{LMTD-VRN-CNTRD}}$. Moreover, note that the direction of motion specified by the control function ctl coincides with the gradient of the multicenter function $\mathcal{H}_{\text{distor-area},\frac{r}{2}}$.

Remark 5.3 (Relative sensing version). It is possible to implement the limited-Voronoi-normal and limited-Voronoi-centroid laws as static relative-sensing control laws on the relative-sensing network $\mathcal{S}^{\text{rs}}_{\text{disk}}$. This is a consequence of the fact that the r-limited Delaunay graph is spatially distributed over the r-disk graph (cf. Theorem 2.7(iii)). Let us present one of these examples for completeness.

`Relative Sensing Network:` $\mathcal{S}^{\text{rs}}_{\text{disk}}$ with motion model (4.2) in Q,
no communication, relative sensing for robot i given by:
robot measurements y contains $p_i^{[j]} \in \overline{B}(\mathbf{0}_2, r)$ for all $j \neq i$
environment measurement is $y_{\text{env}} = (Q_\varepsilon)_i \cap \overline{B}(\mathbf{0}_d, r)$

`Distributed Algorithm:` RELATIVE-SENSING LMTD-VRN-CNTRD

`function` $\mathrm{ctl}(y, y_{\text{env}})$
1: $V := y_{\text{env}} \cap \overline{B}(\mathbf{0}_d, \frac{r}{2}) \cap \big(\bigcap\{H_{\mathbf{0}_d, p_{\text{snsd}}} \mid \text{for all non-null } p_{\text{snsd}} \in y\}\big)$
2: **return** $\mathrm{CM}_\phi(V)$

Note that only the positions of neighboring robots in the r-limited Delaunay graph have an effect in the computation of the set V. •

Remark 5.4 (Range-limited version of Vrn-cntrd). The LMTD-VRN-NRML and LMTD-VRN-CNTRD laws can be combined into a single control and communication law to synthesize an algorithm that monotonically optimizes the function $\mathcal{H}_{\text{distor-area},\frac{r}{2},b}$, with $b = -\operatorname{diam}(Q)^2$. The law is uniform, static, and data-sampled, with standard message-generation function. This law, that we term RNG-VRN-CNTRD, is uniform, static, and data-sampled, with standard message-generation function.

`Robotic Network:` $\mathcal{S}_{\text{LD}}$ with discrete-time motion model (4.1) in Q,
with absolute sensing of own position, and
with communication range r

`Distributed Algorithm:` RNG-VRN-CNTRD
`Alphabet:` $\mathbb{A} = \mathbb{R}^d \cup \{\texttt{null}\}$

`function` $\mathrm{msg}(p, i)$
1: **return** p

`function` $\mathrm{ctl}(p, y)$
1: $V := Q \cap \big(\bigcap\{H_{p, p_{\text{rcvd}}} \mid \text{for all non-null } p_{\text{rcvd}} \in y\}\big)$
2: $v_1 := 2\,\mathrm{A}_\phi(V \cap \overline{B}(p, \frac{r}{2}))(\mathrm{CM}_\phi(V \cap \overline{B}(p, \frac{r}{2})) - p)$

3: $v_2 := (\mathrm{diam}(Q)^2 - \frac{r^2}{4}) \int_{V \cap \partial \overline{B}(p,\frac{r}{2})} \mathrm{n}_{\mathrm{out},\overline{B}(p,\frac{r}{2})}(q)\phi(q)dq$
4: $\lambda_* := \mathrm{argmax}\{\delta \mapsto \mathcal{H}_V(p + \delta(v_1 + v_2), \overline{B}(p + \delta(v_1 + v_2), \frac{r}{2}))$ is strictly increasing on $(0, \varepsilon)\}$
5: **return** $\lambda_*(v_1 + v_2)$

In the above algorithm, $\mathcal{H}_V$ is defined as in Exercise E2.7. For a point $p \in V$ and a closed ball $\overline{B}$ centered at a point in V with radius $\frac{r}{2}$, we define

$$\mathcal{H}_V(p, \overline{B}) = -\int_{V \cap \overline{B}} \|q - p\|_2^2 \phi(q)dq - \int_{V \cap (Q \setminus \overline{B})} a^2 \phi(q)dq.$$

The RNG-VRN-CNTRD law is relevant because of the following discussion. Recall from Proposition 2.18 that the general mixed distortion-area multicenter function can be used to provide constant-factor approximations of the distortion function $\mathcal{H}_{\mathrm{distor}}$. As we discussed in Section 2.3.1, robots with range-limited interactions cannot implement VRN-CNTRD because, for a given $r \in \mathbb{R}_{>0}$, $\mathcal{G}_{\mathrm{D}}$ is not in general spatially distributed over $\mathcal{G}_{\mathrm{disk}}(r)$ (cf. Remark 2.11). However, robotic agents with range-limited interactions can implement the computations involved in LMTD-VRN-NRML and LMTD-VRN-CNTRD, and hence can optimize $\mathcal{H}_{\mathrm{distor\text{-}area},\frac{r}{2},b}$. Assuming $r \leq 2\,\mathrm{diam}(Q)$, it is fair to say that the above algorithm can be understood as a range-limited version of the VRN-CNTRD law. •

5.2.3 Correctness and complexity of geometric-center laws

In this section we characterize convergence and complexity properties of the geometric-center laws. The asynchronous execution of the Voronoi-centroid control and communication law can be studied as an asynchronous gradient dynamical system; see Cortés et al. [2004].

The following theorem summarizes the results known in the literature about the asymptotic properties of these laws.

Theorem 5.5 (Correctness of the geometric-center algorithms). *For $d \in \mathbb{N}$, $r \in \mathbb{R}_{>0}$ and $\varepsilon \in \mathbb{R}_{>0}$, the following statements hold:*

(i) on the network $\mathcal{S}_{\mathrm{D}}$, the law $\mathcal{CC}_{\text{VRN-CNTRD}}$ and on the network $\mathcal{S}_{\mathrm{vehicles}}$, the law $\mathcal{CC}_{\text{VRN-CNTRD-DYNMCS}}$ both achieve the ε-distortion deployment task $\mathcal{T}_{\varepsilon\text{-distor-dply}}$. Moreover, any execution of $\mathcal{CC}_{\text{VRN-CNTRD}}$ and $\mathcal{CC}_{\text{VRN-CNTRD-DYNMCS}}$ monotonically optimizes the multicenter function $\mathcal{H}_{\mathrm{distor}}$;

(ii) on the network $\mathcal{S}_{\mathrm{D}}$, any execution of the law $\mathcal{CC}_{\text{VRN-CRCMCNTR}}$ monotonically optimizes the multicenter function $\mathcal{H}_{\mathrm{dc}}$;

(iii) on the network $\mathcal{S}_{\mathrm{D}}$, any execution of the law $\mathcal{CC}_{\text{VRN-NCNTR}}$ monotonically optimizes the multicenter function $\mathcal{H}_{\mathrm{sp}}$;

(iv) on the network $\mathcal{S}_{\mathrm{LD}}$, the law $\mathcal{CC}_{\text{LMTD-VRN-NRML}}$ achieves the ε-r-area deployment task $\mathcal{T}_{\varepsilon\text{-}r\text{-area-dply}}$. Moreover, any execution of $\mathcal{CC}_{\text{LMTD-VRN-NRML}}$ monotonically optimizes the multicenter function $\mathcal{H}_{\mathrm{area},\frac{r}{2}}$;

(v) on the network $\mathcal{S}_{\text{LD}}$*, the law* $\mathcal{CC}_{\text{Lmtd-Vrn-cntrd}}$ *achieves the* ε*-*r*-distortion-area deployment task* $\mathcal{T}_{\varepsilon\text{-}r\text{-distor-area-dply}}$*. Moreover, any execution of* $\mathcal{CC}_{\text{Lmtd-Vrn-cntrd}}$ *monotonically optimizes the multicenter function* $\mathcal{H}_{\text{distor-area},\frac{r}{2}}$*.*

The results on $\mathcal{CC}_{\text{Vrn-cntrd}}$ and $\mathcal{CC}_{\text{Vrn-cntrd-dynmcs}}$ appeared originally in [Cortés et al., 2004]. Note that an execution of $\mathcal{CC}_{\text{Vrn-cntrd}}$ can be viewed as an alternating sequence of configuration of points and partitions of the space with the properties that (i) each configuration of points corresponds to the set of centroid locations of the immediately preceding partition in the sequence, and (ii) each partition corresponds to the Voronoi partition determined by the immediately preceding configuration of points in the sequence. The monotonic behavior of $\mathcal{H}_{\text{distor}}$ now follows from Propositions 2.14 and 2.15. Similar interpretations can be given to all other laws. In particular, the monotonic behavior of $\mathcal{H}_{\text{dc}}$ along executions of $\mathcal{CC}_{\text{Vrn-crcmcntr}}$ can be established via Proposition 2.20, and the monotonic behavior of $\mathcal{H}_{\text{sp}}$ along executions of $\mathcal{CC}_{\text{Vrn-ncntr}}$ can be established via Proposition 2.22. Versions of these laws that run in continuous time, together with their asymptotic convergence properties, are studied by Cortés and Bullo [2005] via nonsmooth stability analysis. It is an open research question to show that the laws $\mathcal{CC}_{\text{Vrn-crcmcntr}}$ and $\mathcal{CC}_{\text{Vrn-ncntr}}$ achieve the ε-disk-covering deployment task $\mathcal{T}_{\varepsilon\text{-dc-dply}}$ and the ε-sphere-packing deployment task $\mathcal{T}_{\varepsilon\text{-sp-dply}}$, respectively. Finally, the results on $\mathcal{CC}_{\text{Lmtd-Vrn-nrml}}$ and $\mathcal{CC}_{\text{Lmtd-Vrn-cntrd}}$ appeared in [Cortés et al., 2005].

Next, we analyze the time complexity of $\mathcal{CC}_{\text{Lmtd-Vrn-cntrd}}$. We provide complete results only for the case $d = 1$ and uniform density. We assume that $\text{diam}(Q)$ is independent of n, r and ε.

Theorem 5.6 (Time complexity of Lmtd-Vrn-cntrd law). *Assume the robots evolve in a closed interval* $Q \subset \mathbb{R}$*, that is,* $d = 1$*, and assume that the density is uniform, that is,* $\phi \equiv 1$*. For* $r \in \mathbb{R}_{>0}$ *and* $\varepsilon \in \mathbb{R}_{>0}$*, on the network* $\mathcal{S}_{\text{LD}}$ $\text{TC}(\mathcal{T}_{\varepsilon\text{-}r\text{-distor-area-dply}}, \mathcal{CC}_{\text{Lmtd-Vrn-cntrd}}) \in O(n^3 \log(n\varepsilon^{-1}))$.

The proof of this result is contained in Martínez et al. [2007b].

Remark 5.7 (Congestion effects). Interestingly, Theorem 5.6 also holds if, motivated by wireless congestion considerations, we take the communication range r to be a monotone non-increasing function $r : \mathbb{N} \to]0, 2\pi[$ of the number of robotic agents n. •

5.3 Simulation results

In this section, we illustrate the execution of the various control and communication laws introduced in this chapter.

Geometric-center algorithms for expected-value optimization

The VRN-CNTRD, LMTD-VRN-NRML, and LMTD-VRN-CNTRD control and communication laws are implemented in Mathematica® as a library of routines and a main program running the simulation.

The objective of a first routine is to compute the $\frac{r}{2}$-limited Voronoi partition and parameterize each cell $V_{i,\frac{r}{2}}$, $i \in \{1, \dots, n\}$ in polar coordinates. The objective of a second routine is to compute the surface integrals on these sets and the line integrals on their boundaries via the numerical integration routine `NIntegrate`. We pay careful attention to numerical accuracy issues in the computation of the Voronoi diagram and in the integration.

Measuring displacements in meters, we consider the polygon Q determined by the vertices

$$\{(0,0), (2.125,0), (2.9325,1.5), (2.975,1.6), \\ (2.9325,1.7), (2.295,2.1), (0.85,2.3), (0.17,1.2)\}.$$

The diameter of Q is $\mathrm{diam}(Q) = 3.37796$. In all figures, the density function ϕ is the sum of four Gaussian functions of the form $11\exp(6(-(x - x_{\mathrm{center}})^2 - (y - y_{\mathrm{center}})^2))$ and is represented by means of its contour plot. Darker blue areas correspond to higher values of the density function. The centers $(x_{\mathrm{center}}, y_{\mathrm{center}})$ of the Gaussians are given by $(2.15, .75)$, $(1., .25)$, $(.725, 1.75)$ and $(.25, .7)$, respectively. The area of the polygon is $\mathrm{A}_\phi(Q) = 17.6352$.

We show evolutions of $(\mathcal{S}_{\mathrm{D}}, \text{VRN-CNTRD})$ and $(\mathcal{S}_{\mathrm{D}}, \text{VRN-CNTRD-DYNMCS})$ in Figures 5.2 and 5.3, respectively. One can verify that the final network configurations are centroidal Voronoi configuration. In other words, the task $\mathcal{T}_{\varepsilon\text{-distor-dply}}$ is achieved, as guaranteed by Theorem 5.5(i).

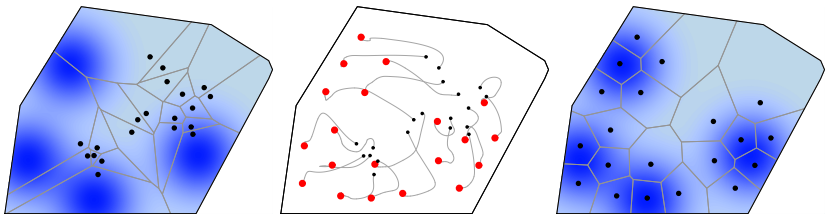


Fig. 5.2. Evolution of $(\mathcal{S}_{\mathrm{D}}, \text{VRN-CNTRD})$ with $n = 20$ robots. The left (respectively, right) figure illustrates the initial (respectively, final) locations and Voronoi partition. The central figure illustrates the evolution of the robots. After 13 seconds, the value of the multicenter function $\mathcal{H}_{\mathrm{distor}}$ has monotonically increased to approximately $-.515$.

We show an evolution of $(\mathcal{S}_{\mathrm{LD}}, \text{LMTD-VRN-NRML})$ in Figure 5.4. One can verify that the final network configuration is a $\frac{r}{2}$-limited area-centered Voronoi configuration. In other words, the task $\mathcal{T}_{\varepsilon\text{-}r\text{-area-dply}}$ is achieved, as guaranteed by Theorem 5.5(ii).

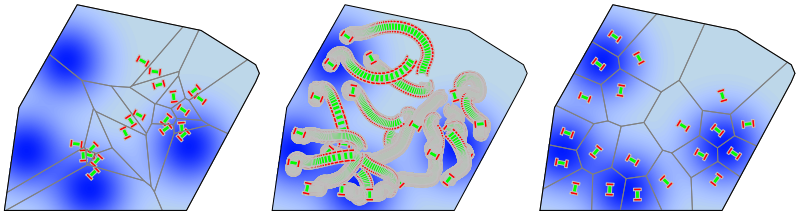


Fig. 5.3. Evolution of $(\mathcal{S}_{\mathrm{D}}, \text{VRN-CNTRD-DYNMCS})$ with $n = 20$ robots. The feedback gain is $k_{\mathrm{prop}} = 3.5$. The left (respectively, right) figure illustrates the initial (respectively, final) locations and Voronoi partition. The central figure illustrates the evolution of the robots. After 20 seconds, the value of the multicenter function $\mathcal{H}_{\mathrm{distor}}$ has monotonically increased to approximately $-.555$.

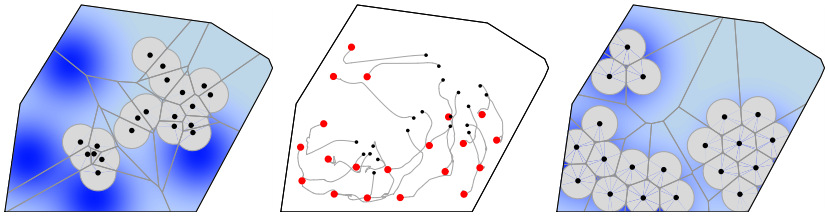


Fig. 5.4. Evolution of $(\mathcal{S}_{\mathrm{LD}}, \text{LMTD-VRN-NRML})$ with $n = 20$ robots and $r = 0.4$. The left (respectively, right) figure illustrates the initial (respectively, final) locations and Voronoi partition. The central figure illustrates the evolution of the robots. The $\frac{r}{2}$-limited Voronoi cell of each robot is plotted in light gray. After 36 seconds, the value of the multicenter function $\mathcal{H}_{\mathrm{area},\frac{r}{2}}$ is approximately 14.141.

We show an evolution of $(\mathcal{S}_{\mathrm{LD}}, \text{LMTD-VRN-CNTRD})$ in Figure 5.5. One can verify that the final network configuration is a $\frac{r}{2}$-limited centroidal Voronoi configuration. In other words, the task $\mathcal{T}_{\varepsilon\text{-}r\text{-distor-area-dply}}$ is achieved, as guaranteed by Theorem 5.5(iii).

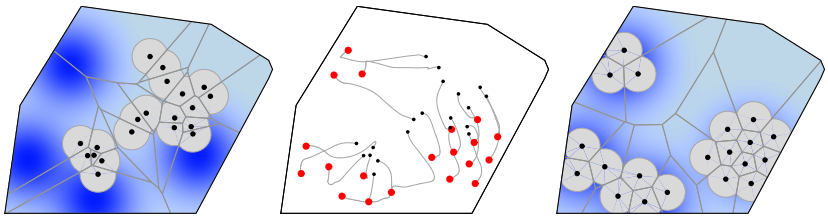


Fig. 5.5. Evolution of $(\mathcal{S}_{\mathrm{LD}}, \text{LMTD-VRN-CNTRD})$ with $n = 20$ robots and $r = 0.4$. The left (respectively, right) figure illustrates the initial (respectively, final) locations and Voronoi partition. The central figure illustrates the evolution of the robots. The $\frac{r}{2}$-limited Voronoi cell of each robot is plotted in light gray. After 90 seconds, the value of the multicenter function $\mathcal{H}_{\mathrm{distor\text{-}area},\frac{r}{2}}$ is approximately $-.386$.

We show an evolution of $(\mathcal{S}_{\mathrm{LD}}, \text{RNG-VRN-CNTRD})$ in Figure 5.6. One can verify that the final network configuration corresponds to a critical point of the mixed distortion-area multicenter function $\mathcal{H}_{\text{distor-area},\frac{r}{2},b}$, with $b = -\operatorname{diam}(Q)^2$, see Exercise E5.3.

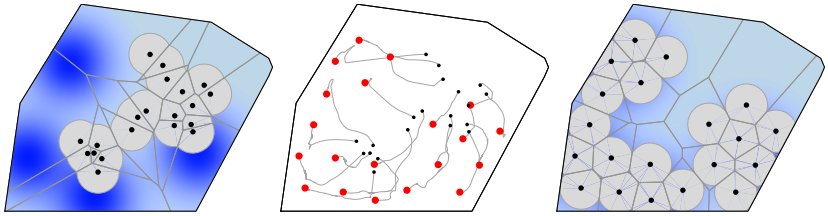


Fig. 5.6. Evolution of $(\mathcal{S}_{\mathrm{LD}}, \text{RNG-VRN-CNTRD})$ with $n = 20$ robots and $r = 0.47$. The left (respectively, right) figure illustrates the initial (respectively, final) locations and Voronoi partition. The central figure illustrates the evolution of the robots. The $\frac{r}{2}$-limited Voronoi cell of each robot is plotted in light gray. After 13 seconds, the value of the multicenter function $\mathcal{H}_{\text{distor-area},\frac{r}{2},b}$, with $b = -\operatorname{diam}(Q)^2$, is approximately -4.794.

As discussed in Remark 5.4, the RNG-VRN-CNTRD can be understood as a limited-range implementation of VRN-CNTRD in a network of robots with limited-range interactions. Let us briefly compare the evolutions depicted in Figures 5.2 and 5.6. According to Proposition 2.18, we compute

$$\beta = \frac{\frac{r}{2}}{\operatorname{diam} Q} \approx 0.06957.$$

From the constant-factor approximation (2.9), the absolute error is guaranteed to be less than or equal to $(\beta^2 - 1)\mathcal{H}_{\text{distor-area},\frac{r}{2},b}(P_{\text{final}}) \approx 4.77$, where P_{final} denotes the final configuration in Figure 5.6. The percentage error in the value of the multicenter function $\mathcal{H}_{\text{distor}}$ between the final configuration of the evolution in Figure 5.2 and the final configuration of the evolution in Figure 5.6 is approximately equal to 3.277%. As expected, one can verify in simulations that the percentage error of the performance of the limited-range implementation improves with higher values of the ratio $\frac{r}{\operatorname{diam} Q}$.

Geometric-center algorithms for disk-covering and sphere-packing

The VRN-CRCMCNTR and VRN-NCNTR control and communication laws are implemented in Mathematica® as a single centralized program running the simulation. We compute the bounded Voronoi diagram of a collection of points using the package `ComputationalGeometry`. We compute the circumcenter of a polygon via the algorithm in [Skyum, 1991] and the incenter set via the `LinearProgramming` solver in `Mathematica`.

Measuring displacements in meters, we consider the polygon determined by the vertices

$$\{(0,0),(2.5,0),(3.45,1.5),(3.5,1.6),(3.45,1.7),(2.7,2.1),(1.,2.4),(.2,1.2)\}.$$

We show an evolution of $(\mathcal{S}_{\mathrm{D}}, \textsc{Vrn-crcmcntr})$ in Figure 5.7. One can verify that at the final configuration all robots are at the circumcenter of their own Voronoi cell. In other words, the task $\mathcal{T}_{\varepsilon\text{-dc-dply}}$ is achieved by this evolution. As stated in Section 5.2.3, it is an open research question to show that this fact holds in general for $\mathcal{CC}_{\textsc{Vrn-crcmcntr}}$. Cortés and Bullo [2005] prove a similar result for a continuous-time implementation of this law.

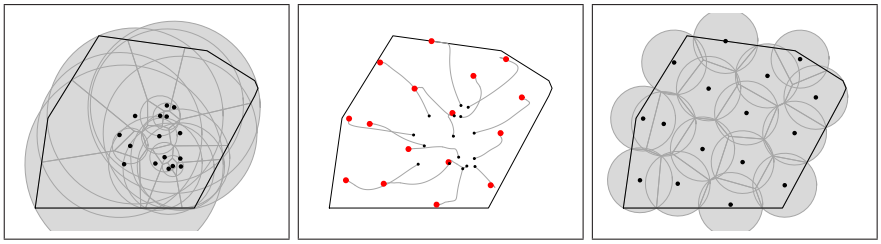


Fig. 5.7. Evolution of $(\mathcal{S}_{\mathrm{D}}, \textsc{Vrn-crcmcntr})$ with $n = 16$ robots. The left (respectively, right) figure illustrates the initial (respectively, final) locations and Voronoi partition. The central figure illustrates the evolution of the robots. After 20 seconds, the value of the multicenter function $\mathcal{H}_{\mathrm{dc}}$ has monotonically decreased to approximately 0.43273 meters.

We show an evolution of $(\mathcal{S}_{\mathrm{D}}, \textsc{Vrn-ncntr})$ in Figure 5.8. One can verify that at the final configuration all robots are at the incenter of their own Voronoi cell. In other words, the task $\mathcal{T}_{\varepsilon\text{-sp-dply}}$ is achieved by this evolution. As stated in Section 5.2.3, it is an open research question to show that this fact holds in general for $\mathcal{CC}_{\textsc{Vrn-ncntr}}$. Cortés and Bullo [2005] prove a similar result for a continuous-time implementation of this law.

5.4 Notes

The deployment problem studied in this chapter is related to the literature on facility location [Drezner, 1995, Okabe et al., 2000, Du et al., 1999] and geometric optimization [Agarwal and Sharir, 1998, Boltyanski et al., 1999], see also Section 2.4. These disciplines study spatial resource allocation problems and play an important role in quantization theory, mesh and grid optimization methods, clustering analysis, data compression, and statistical pattern recognition.

Dispersion laws have been traditionally studied in behavior control, see e.g., [Arkin, 1998, Schultz and Parker, 2002, Balch and Parker, 2002]. Deployment algorithms that make use of potential field methods are proposed

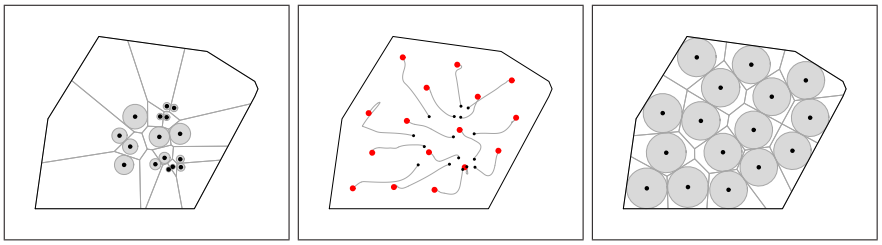


Fig. 5.8. Evolution of $(\mathcal{S}_{\mathrm{D}}, \textsc{Vrn-ncntr})$ with $n = 16$ robots. The left (respectively, right) figure illustrates the initial (respectively, final) locations and Voronoi partition. The central figure illustrates the evolution of the robots. After 20 seconds, the value of the multicenter function $\mathcal{H}_{\mathrm{sp}}$ has monotonically increased to approximately 0.2498 meters.

by Payton et al. [2001], Howard et al. [2002]. Other works include [Winfield, 2000] on distributed sensing in ad-hoc wireless networks, [Bulusu et al., 2001] on adaptive beacon placement for localization, [Poduri and Sukhatme, 2004] on network deployments that satisfy a pre-specified constraint in the number of neighbors of each robot, [Arsie and Frazzoli, 2007] on deployment strategies that minimize the expected time needed for any robot to service a newly-appeared target point in the environment, and [Hussein and Stipanovìc, 2007] on dynamically surveying a known environment.

Deployment algorithms for coverage control are a subject of active research. Among the most recent works, Martínez [2007], Schwager et al. [2008b] consider coverage problems where the density function is unknown, Lekien and Leonard [2007] propose centralized laws for non-uniform coverage using cartograms, de Silva and Ghrist [2007] study static coverage problems with minimal assumptions on the capabilities of individual sensors using algebraic topology, Kwok and Martínez [2008] propose distributed deployment strategies for energy-constrained networks, Laventall and Cortés [2008] design distributed algorithms for networks of robots whose sensors have range-limited wedge-shaped footprints, Gao et al. [2008] consider discrete coverage problems, Schwager et al. [2008a] consider combined exploration and deployment problems, and Cassandras and Zhong [2008], Pimenta et al. [2008], Caicedo-Nùñez and Żefran [2008] deal with centroidal Voronoi tessellations in nonconvex environments. Graham and Cortés [2007] study the optimality of circumcenter and incenter Voronoi configurations for the estimation of stochastic spatial fields. Susca et al. [2007] consider some planar interpolation problems.

Deployment problems play a relevant role in other coordination tasks, such as surveillance, search and rescue, and exploration and map building of unknown environments. Choset [2001] considers sweep coverage problems, where one or more robots equipped with limited footprint sensors have to visit all points in the environment. In [Simmons et al., 2000], deployment locations for a network of heterogeneous robots are user-specified after an initial map

of the unknown environment has been built. Gupta et al. [2006a] consider a combined sensor coverage and selection problem.

Deployment of robotic agents with visibility sensors has been studied under a variety of assumptions. When the environment is known a priori, the problem can be casted as the classical Art Gallery Problem [Chvátal, 1975] from computational geometry, where one is interested in achieving complete visibility with the minimum number of agents possible. The Art Gallery Problem is computationally hard [Lee and Lin, 1986, Eidenbenz et al., 2001] and the best known approximation algorithms yield solutions within a logarithmic factor of the optimum number of agents [Ghosh, 1987, Efrat and Har-Peled, 2006]. Hernández-Peñalver [1994], Pinciu [2003] study the problem of achieving full visibility while guaranteeing that the final network configuration will have a connected visibility graph. Recent works on multi-robot exploration of unknown environments include [Batalin and Sukhatme, 2004, Burgard et al., 2005, Howard et al., 2006]. Topological exploration of graph-like environments by single and multiple robots is studied in [Rekleitis et al., 2001, Dynia et al., 2006, Fraigniaud et al., 2004]. A simple one-step strategy for visibility deployment, without the need for synchronization, achieving the worst-case optimal bounds in terms of number of robots required, and under limited communication is presented in [Ganguli et al., 2007a].

5.5 Proofs

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

5.5.1 Proof of Theorem 5.5

Proof. Let $P_0 = (p^{[1]}(0), \dots, p^{[n]}(0)) \in Q^n$ denote the initial condition. The proof strategy for all five facts is similar. We first establish the monotonic behavior of the corresponding multicenter function along the executions of the control and communication law. Then, we use the LaSalle Invariance Principle to determine the sets of asymptotic convergence of the executions. Finally, we characterize these sets using geometric properties of the multicenter functions.

Fact (i). Let us start by showing that executions of $\mathcal{CC}_{\text{VRN-CNTRD}}$ monotonically optimize the function $\mathcal{H}_{\text{distor}}$. For convenience, we denote by $f_{\text{VRN-CNTRD}} : Q^n \to Q^n$ the map induced by the execution of one step of the law $\mathcal{CC}_{\text{VRN-CNTRD}}$. Using the extension of the multicenter function defined over set of points and partitions of Q, we deduce from Proposition 2.14 that, for $P \in Q^n$,

$$\begin{aligned}\mathcal{H}_{\text{distor}}(f_{\text{VRN-CNTRD}}(P)) &= \mathcal{H}_{\text{distor}}(f_{\text{VRN-CNTRD}}(P), \mathcal{V}(f_{\text{VRN-CNTRD}}(P))) \\ &\geq \mathcal{H}_{\text{distor}}(f_{\text{VRN-CNTRD}}(P), \mathcal{V}(P)).\end{aligned}$$

The application of Proposition 2.15 yields

$$\mathcal{H}_{\text{distor}}(f_{\text{VRN-CNTRD}}(P), \mathcal{V}(P)) \geq \mathcal{H}_{\text{distor}}(P, \mathcal{V}(P)),$$

and hence $\mathcal{H}_{\text{distor}}(f_{\text{VRN-CNTRD}}(P)) \geq \mathcal{H}_{\text{distor}}(P)$. Since all Voronoi cells have non-vanishing areas, the inequality is strict unless $f_{\text{VRN-CNTRD}}(P) = P$.

The result on the monotonic evolution of $\mathcal{H}_{\text{distor}}$ along the executions of $\mathcal{CC}_{\text{VRN-CNTRD-DYNMCS}}$ can be established in a similar way making use of Proposition 2.14 and Exercise E2.5. The key property of VRN-CNTRD-DYNMCS (see Remark 5.1) is that, when run over a finite time interval $[t, t+1]$, the control function ctl guarantees that, for each robot $i \in \{1, \dots, n\}$,

$$\|p^{[i]}(t+1) - \text{CM}_\phi(V)\|_2 \leq \|p^{[i]}(t) - \text{CM}_\phi(V)\|_2,$$

with strict inequality unless $p^{[i]}(t) = \text{CM}_\phi(V)$.

The convergence result for both laws is established in the same way using the LaSalle Invariance Principle stated in Theorem 1.16. Here, we state the proof for VRN-CNTRD. Since the set Q^n is compact and $-\mathcal{H}_{\text{distor}}$ is non-increasing along $f_{\text{VRN-CNTRD}}$, we deduce that the execution of $\mathcal{CC}_{\text{VRN-CNTRD}}$ starting from P_0 tends to the largest positively invariant set M contained in

$$\{P \in Q^n \mid \mathcal{H}_{\text{distor}}(f_{\text{VRN-CNTRD}}(P)) = \mathcal{H}_{\text{distor}}(P)\}.$$

The set M is precisely the set of centroidal Voronoi configurations. This is a consequence of the fact that $\mathcal{H}_{\text{distor}}(f_{\text{VRN-CNTRD}}(P)) = \mathcal{H}_{\text{distor}}(P)$ implies $f_{\text{VRN-CNTRD}}(P) = P$, i.e., P is a centroidal Voronoi configuration.

Facts (ii) and (iii). The proof of these facts runs parallely to the proof of fact (i). Propositions 2.20 and 2.22 are key in establishing the monotonic evolution of $\mathcal{H}_{\text{dc}}$ and $\mathcal{H}_{\text{sp}}$, respectively.

Fact (iv). For convenience, we denote by $f_{\text{LMTD-VRN-NRML}} : Q^n \to Q^n$ the map induced by the execution of one step of the law $\mathcal{CC}_{\text{LMTD-VRN-NRML}}$. Let us show that executions of $\mathcal{CC}_{\text{LMTD-VRN-NRML}}$ monotonically optimize the function $\mathcal{H}_{\text{area},\frac{r}{2}}$. Using the extension of the multicenter function defined over set of points and partitions of Q, we deduce from Proposition 2.14 that, for $P \in Q^n$,

$$\begin{aligned}\mathcal{H}_{\text{area},\frac{r}{2}}(f_{\text{LMTD-VRN-NRML}}(P)) &= \mathcal{H}_{\text{area},\frac{r}{2}}(f_{\text{LMTD-VRN-NRML}}(P), \mathcal{V}(f_{\text{LMTD-VRN-NRML}}(P))) \\ &\geq \mathcal{H}_{\text{area},\frac{r}{2}}(f_{\text{LMTD-VRN-NRML}}(P), \mathcal{V}(P)).\end{aligned}$$

The line search procedure for each robot embedded in the definition of the control function of $\mathcal{CC}_{\text{LMTD-VRN-NRML}}$ ensures that

$$\mathcal{H}_{\text{area},\frac{r}{2}}(f_{\text{LMTD-VRN-NRML}}(P), \mathcal{V}(P)) \geq \mathcal{H}_{\text{area},\frac{r}{2}}(P, \mathcal{V}(P)),$$

and hence $\mathcal{H}_{\text{area},\frac{r}{2}}(f_{\text{LMTD-VRN-NRML}}(P)) \geq \mathcal{H}_{\text{area},\frac{r}{2}}(P)$. Note that the inequality is strict unless $f_{\text{LMTD-VRN-NRML}}(P) = P$. The application of the LaSalle Invariance Principle as in the proof of fact (i) leads us now to the result.

Fact (v). The proof of this facts runs parallely to the proof of facts (i) and (iv). Propositions 2.14 and 2.16 are key in establishing the monotonic evolution of $\mathcal{H}_{\text{distor-area},\frac{r}{2}}$. The convergence result is established using the LaSalle Invariance Principle. ■

5.5.2 Proof of Theorem 5.6

Proof. For $d = 1$, Q is a compact interval on $\mathbb{R}$, say $Q = [q_-, q_+]$. We start with a brief discussion about connectivity. In the r-limited Delaunay graph, two agents that are at most at a distance r from each other are neighbors if and only if there are no other agents between them. Additionally, we claim that, if agents i and j are neighbors, then $|\operatorname{CM}_\phi(V^{[i]}) - \operatorname{CM}_\phi(V^{[j]})| \leq r$, where $V^{[i]}$ denotes the set defined by the control function ctl when evaluated by agent i. To show this fact, assume without loss of generality that $p^{[i]} \leq p^{[j]}$. Let us consider the case where the agents have neighbors on both sides (the other cases can be treated analogously). Let $p_-^{[i]}$ (respectively, $p_+^{[j]}$) denote the position of the neighbor of agent i to the left (respectively, of agent j to the right). Now,

$$\operatorname{CM}_\phi(V^{[i]}) = \frac{1}{4}(p_-^{[i]} + 2p^{[i]} + p^{[j]}),$$
$$\operatorname{CM}_\phi(V^{[j]}) = \frac{1}{4}(p^{[i]} + 2p^{[j]} + p_+^{[j]}),$$

where we have used the fact that $\phi \equiv 1$. Therefore,

$$|\operatorname{CM}_\phi(V^{[i]}) - \operatorname{CM}_\phi(V^{[j]})| \leq \frac{1}{4}\big(|p_-^{[i]} - p^{[i]}| + 2|p^{[i]} - p^{[j]}| + |p^{[j]} - p_+^{[j]}|\big) \leq r.$$

This implies that agents i and j belong to the same connected component of the r-limited Delaunay graph at the next time step.

Next, let us consider the case when $\mathcal{G}_{\mathrm{LD}}(r)$ is connected at the initial network configuration $P_0 = (p^{[1]}(0), \ldots, p^{[n]}(0))$. Without loss of generality, assume that the agents are ordered from left to right according to their unique identifier, that is, $p^{[1]}(0) \leq \cdots \leq p^{[n]}(0)$. We distinguish three cases depending on the proximity of the leftmost and rightmost agents 1 and n, respectively, to the boundary of the environment: case (**a**) both agents are within a distance $\frac{r}{2}$ of ∂Q; case (**b**) none of the two is within a distance $\frac{r}{2}$ of ∂Q; and case (**c**) only one of the agents is within a distance $\frac{r}{2}$ of ∂Q. Here is an important observation: from one time instant to the next one, the network configuration can fall into any of the cases described above. However, because of the discussion on connectivity, transitions can only occur from case (**b**) to either case (**a**) or (**c**); and from case (**c**) to case (**a**). As we show below, for each of these cases, the network evolution under $\mathcal{CC}_{\text{VRN-CNTRD}}$ can be described as a discrete-time linear dynamical system which respects agents' ordering.

Let us consider case (**a**). In this case, we have

$$
\begin{aligned}
p^{[1]}(\ell+1) &= \frac{1}{4}(p^{[1]}(\ell) + p^{[2]}(\ell)) + \frac{1}{2}q_-,\\
p^{[2]}(\ell+1) &= \frac{1}{4}(p^{[1]}(\ell) + 2p^{[2]}(\ell) + p^{[3]}(\ell)),\\
&\vdots\\
p^{[n-1]}(\ell+1) &= \frac{1}{4}(p^{[n-2]}(\ell) + 2p^{[n-1]}(\ell) + p^{[n]}(\ell)),\\
p^{[n]}(\ell+1) &= \frac{1}{4}(p^{[n-1]}(\ell) + p^{[n]}(\ell)) + \frac{1}{2}q_+\,.
\end{aligned}
$$

Equivalently, we can write $P(\ell+1) = A_{(\mathbf{a})} \cdot P(\ell) + b_{(\mathbf{a})}$, where the matrix $A_{(\mathbf{a})} \in \mathbb{R}^{n\times n}$ and the vector $b_{(\mathbf{a})} \in \mathbb{R}^n$ are given by

$$
A_{(\mathbf{a})} = \begin{bmatrix}
\frac{1}{4} & \frac{1}{4} & 0 & \dots & \dots & 0\\
\frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \dots & \dots & 0\\
0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \dots & 0\\
\vdots & & \ddots & \ddots & \ddots & \vdots\\
0 & \dots & \dots & \frac{1}{4} & \frac{1}{2} & \frac{1}{4}\\
0 & \dots & \dots & 0 & \frac{1}{4} & \frac{1}{4}
\end{bmatrix}, \quad
b_{(\mathbf{a})} = \begin{bmatrix}
\frac{1}{2}q_-\\ 0\\ \vdots\\ 0\\ \frac{1}{2}q_+
\end{bmatrix}.
$$

Note that the only equilibrium network configuration P_* respecting the ordering of the agents is given by

$$
p_*^{[i]} = q_- + \frac{1}{2n}(1+2(i-1))(q_+ - q_-)\,, \quad i \in \{1,\dots,n\}\,,
$$

and note that this is a $\frac{r}{2}$-centroidal Voronoi configuration (under the assumption of case (**a**)). We can therefore write $(P(\ell+1) - P_*) = A_{(\mathbf{a})}(P(\ell) - P_*)$. Now, note that $A_{(\mathbf{a})} = \mathrm{ATrid}_n^-\left(\frac{1}{4}, \frac{1}{2}\right)$. Theorem 1.75(ii) implies that $\lim_{\ell\to+\infty}\big(P(\ell) - P_*\big) = \mathbf{0}_n$, and that the maximum time required for $\|P(\ell) - P_*\|_2 \leq \varepsilon\|P_0 - P_*\|_2$ (over all initial conditions in $\mathbb{R}^n$) is $\Theta\big(n^2\log\varepsilon^{-1}\big)$. It is not obvious, but it can be verified, that the initial condition providing the lower bound in the time complexity estimate does indeed have the property of respecting the agents' ordering; this fact holds for all three cases (**a**), (**b**) and (**c**).

The case (**b**) can be treated in the same way. The network evolution takes now the form $P(\ell+1) = A_{(\mathbf{b})} \cdot P(\ell) + b_{(\mathbf{b})}$, where the matrix $A_{(\mathbf{b})} \in \mathbb{R}^{n\times n}$ and the vector $b_{(\mathbf{b})} \in \mathbb{R}^n$ are given by

$$
A_{(\mathbf{b})} = \begin{bmatrix}
\frac{3}{4} & \frac{1}{4} & 0 & \dots & \dots & 0\\
\frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \dots & \dots & 0\\
0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \dots & 0\\
\vdots & & \ddots & \ddots & \ddots & \vdots\\
0 & \dots & \dots & \frac{1}{4} & \frac{1}{2} & \frac{1}{4}\\
0 & \dots & \dots & 0 & \frac{1}{4} & \frac{3}{4}
\end{bmatrix}, \quad
b_{(\mathbf{b})} = \begin{bmatrix}
-\frac{1}{4}r\\ 0\\ \vdots\\ 0\\ \frac{1}{4}r
\end{bmatrix}.
$$

In this case, a (non-unique) equilibrium network configuration respecting the ordering of the agents is of the form

$$p_*^{[i]} = ir - \frac{1+n}{2}r\,, \quad i \in \{1,\dots,n\}\,.$$

Note that this is a $\frac{r}{2}$-centroidal Voronoi configuration (under the assumption of case (**b**)). We can therefore write $(P(\ell+1)-P_*) = A_{(\mathbf{b})}(P(\ell)-P_*)$. Now, note that $A_{(\mathbf{b})} = \mathrm{ATrid}_n^+\left(\frac{1}{4},\frac{1}{2}\right)$. We compute $P_{\mathrm{ave}} = \frac{1}{n}\mathbf{1}_n^T(P_0-P_*) = \frac{1}{n}\mathbf{1}_n^T P_0$. With this calculation, Theorem 1.75(i) implies that $\lim_{\ell\to+\infty}\left(P(\ell)-P_*-P_{\mathrm{ave}}\mathbf{1}_n\right) = \mathbf{0}_n$, and that the maximum time required for $\|P(\ell)-P_*-P_{\mathrm{ave}}\mathbf{1}_n\|_2 \leq \varepsilon\|P_0-P_*-P_{\mathrm{ave}}\mathbf{1}_n\|_2$ (over all initial conditions in $\mathbb{R}^n$) is $\Theta\left(n^2\log\varepsilon^{-1}\right)$.

Case (**c**) needs to be handled differently. Without loss of generality, assume that agent 1 is within distance $\frac{r}{2}$ of ∂Q and agent n is not (the other case is treated analogously). Then, the network evolution takes now the form $P(\ell+1) = A_{(\mathbf{c})}\cdot P(\ell) + b_{(\mathbf{c})}$, where the matrix $A_{(\mathbf{c})} \in \mathbb{R}^{n\times n}$ and the vector $b_{(\mathbf{c})} \in \mathbb{R}^n$ are given by

$$A_{(\mathbf{c})} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & 0 & \dots & \dots & 0 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \dots & \dots & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & \dots & \dots & 0 & \frac{1}{4} & \frac{3}{4} \end{bmatrix}, \quad b_{(\mathbf{c})} = \begin{bmatrix} \frac{1}{2}q_- \\ 0 \\ \vdots \\ 0 \\ \frac{1}{4}r \end{bmatrix}.$$

Note that the only equilibrium network configuration P_* respecting the ordering of the agents is given by

$$p_*^{[i]} = q_- + \frac{1}{2}(2i-1)r\,, \quad i \in \{1,\dots,n\}\,,$$

and note that this is a $\frac{r}{2}$-centroidal Voronoi configuration (under the assumption of case (**c**)). In order to analyze $A_{(\mathbf{c})}$, we recast the n-dimensional discrete-time dynamical system as a $2n$-dimensional one. To do this, we define a $2n$-dimensional vector y by

$$y^{[i]} = p^{[i]},\ i \in \{1,\dots,n\}, \quad \text{and} \quad y^{[n+i]} = p^{[n-i+1]},\ i \in \{1,\dots,n\}. \tag{5.2}$$

Now, one can see that the network evolution can be alternatively described in the variables $(y^{[1]},\dots,y^{[2n]})$ as a linear dynamical system determined by the $2n\times 2n$ matrix $\mathrm{ATrid}_{2n}^-(\frac{1}{4},\frac{1}{2})$. Using Theorem 1.75(ii), and exploiting the chain of equalities (5.2), we can infer that, in case (**c**), the maximum time required for $\|P(\ell)-P_*\|_2 \leq \varepsilon\|P_0-P_*\|_2$ (over all initial conditions in $\mathbb{R}^n$) is $\Theta\left(n^2\log\varepsilon^{-1}\right)$.

In summary, for all three cases (**a**), (**b**) and (**c**), our calculations show that, in time $O\left(n^2\log\varepsilon^{-1}\right)$, the error 2-norm satisfies the contraction inequality

$\|P(\ell) - P_*\|_2 \leq \varepsilon \|P_0 - P_*\|_2$. We convert this inequality on 2-norms into an appropriate inequality on ∞-norms as follows. Note that $\|P_0 - P_*\|_\infty = \max_{i\in\{1,\dots,n\}} |p^{[i]}(0) - p_*^{[i]}| \leq (q_+ - q_-)$. For ℓ of order $n^2 \log \eta^{-1}$, we have

$$\begin{aligned}\|P(\ell) - P_*\|_\infty &\leq \|P(\ell) - P_*\|_2 \leq \eta \|P_0 - P_*\|_2 \\ &\leq \eta\sqrt{n}\|P_0 - P_*\|_\infty \leq \eta\sqrt{n}(q_+ - q_-).\end{aligned}$$

This means that ε-r-deployment is achieved for $\eta\sqrt{n}(q_+ - q_-) = \varepsilon$, that is, in time $O(n^2 \log \eta^{-1}) = O(n^2 \log(n\varepsilon^{-1}))$.

Up to here we have proved that, if the graph $\mathcal{G}_{\mathrm{LD}}(r)$ is connected at P_0, then $\mathrm{TC}(\mathcal{T}_{\varepsilon\text{-}r\text{-dply}}, \mathcal{CC}_{\text{VRN-CNTRD}}, P_0) \in O(n^2 \log(n\varepsilon^{-1}))$. If $\mathcal{G}_{\mathrm{LD}}(r)$ is not connected at P_0, note that along the network evolution there can only be a finite number of time instants, at most $n-1$ where a merging of two connected components occurs. Therefore, the time complexity is at most $O(n^3 \log(n\varepsilon^{-1}))$, as claimed. ■

5.6 Exercises

E5.1 **(Monotonic evolution of $\mathcal{H}_{\mathbf{dc}}$ and $\mathcal{H}_{\mathbf{sp}}$).** Prove statements (ii) and (iii) in Theorem 5.5.
Hint: *Make use of the optimality of the Voronoi partition and of center locations stated in Propositions 2.20 and 2.22.*

E5.2 Prove statement (v) in Theorem 5.5.
Hint: *To establish the monotonic evolution of the multicenter function, make use of the optimality of the Voronoi partition stated in Proposition 2.14 and of centroid locations stated in Proposition 2.16. To establish the convergence result, make use of the LaSalle Invariance Principle stated in Theorem 1.16.*

E5.3 **(Correctness of Rng-Vrn-cntrd).** Mimic the proof of Theorem 5.5(iv) to show that the evolutions of RNG-VRN-CNTRD monotonically optimize the mixed distortion-area multicenter function

$$\mathcal{H}_{\text{distor-area},\frac{r}{2},b}, \quad \text{with } b = -\operatorname{diam}(Q)^2,$$

and asymptotically approach its set of critical points.

References

Agarwal, P. K. and Sharir, M. [1998] *Efficient algorithms for geometric optimization*, ACM Computing Surveys, **30**(4), 412–458.

Agmon, N. and Peleg, D. [2006] *Fault-tolerant gathering algorithms for autonomous mobile robots*, SIAM Journal on Computing, **36**(1), 56–82.

Alighanbari, M. and How, J. P. [2006] *Robust decentralized task assignment for cooperative UAVs*, in *AIAA Conf. on Guidance, Navigation and Control*, Keystone, CO.

Ando, H., Oasa, Y., Suzuki, I., and Yamashita, M. [1999] *Distributed memoryless point convergence algorithm for mobile robots with limited visibility*, IEEE Transactions on Robotics and Automation, **15**(5), 818–828.

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.

Arai, T., Pagello, E., and Parker, L. E. [2002] *Guest editorial: Advances in multirobot systems*, IEEE Transactions on Robotics and Automation, **18**(5), 655–661.

Arkin, R. C. [1998] *Behavior-Based Robotics*, MIT Press, Cambridge, MA, ISBN 0262011654.

Arsie, A. and Frazzoli, E. [2007] *Efficient routing of multiple vehicles with no communications*, International Journal on Robust and Nonlinear Control, **18**(2), 154–164.

Arslan, G., Marden, J. R., and Shamma, J. S. [2007] *Autonomous vehicle-target assignment: A game theoretic formulation*, ASME Journal on Dynamic Systems, Measurement, and Control, **129**(5), 584–596.

Aspnes, J., Eren, T., Goldenberg, D. K., Morse, A. S., Whiteley, W., Yang, Y. R., Anderson, B. D. O., and Belhumeur, P. [2006] *A theory of network localization*, IEEE Transactions on Mobile Computing, **5**(12), 1663–1678.

Astolfi, A. [1999] *Exponential stabilization of a wheeled mobile robot via discontinuous control*, ASME Journal on Dynamic Systems, Measurement, and Control, **121**(1), 121–127.

Aurenhammer, F. [1991] *Voronoi diagrams: A survey of a fundamental geometric data structure*, ACM Computing Surveys, **23**(3), 345–405.

Axelrod, R. [1997] *The dissemination of culture: A model with local convergence and global polarization*, Journal of Conflict Resolution, **41**(2), 203–226.

Baillieul, J. and Suri, A. [2003] *Information patterns and hedging Brockett's theorem in controlling vehicle formations*, in *IEEE Conf. on Decision and Control*, pages 556–563, Maui, HI.

Balch, T. and Parker, L. E., (editors) [2002] *Robot Teams: From Diversity to Polymorphism*, A K Peters Ltd., Natick, MA, ISBN 1568811551.

Barbašin, E. A. and Krasovskiĭ, N. N. [1952] *On stability of motion in the large*, Doklady Akad. Nauk SSSR, **86**, 453–456, (In Russian).

Barlow, G. W. [1974] *Hexagonal territories*, Animal Behavior, **22**, 876–878.

Barooah, P. and Hespanha, J. P. [2007] *Estimation from relative measurements: Algorithms and scaling laws*, IEEE Control Systems Magazine, **27**(4), 57–74.

Bartle, R. G. [1995] *The Elements of Integration and Lebesgue Measure*, Wiley-Interscience, ISBN 0471042226.

Batalin, M. A. and Sukhatme, G. S. [2004] *Coverage, exploration and deployment by a mobile robot and communication network*, Telecommunication Systems Journal, **26**(2), 181–196, Special Issue on Wireless Sensor Networks.

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.

Beard, R. W., McLain, T. W., Goodrich, M. A., and Anderson, E. P. [2002] *Coordinated target assignment and intercept for unmanned air vehicles*, IEEE Transactions on Robotics and Automation, **18**(6), 911–922.

Belta, C. and Kumar, V. [2004] *Abstraction and control for groups of robots*, IEEE Transactions on Robotics, **20**(5), 865–875.

Bertozzi, A. L., Kemp, M., and Marthaler, D. [2004] *Determining environmental boundaries: Asynchronous communication and physical scales*, in *Cooperative Control*, V. Kumar, N. E. Leonard, and A. S. Morse, editors, volume 309 of *Lecture Notes in Control and Information Sciences*, pages 25–42, Springer Verlag, ISBN 3540228616.

Bertsekas, D. P. and Castañón, D. A. [1991] *Parallel synchronous and asynchronous implementations of the auction algorithm*, Parallel Computing, **17**, 707–732.

— [1993] *Parallel primal-dual methods for the minimum cost flow problem*, Computational Optimization and Applications, **2**(4), 317–336.

Bertsekas, D. P. and Tsitsiklis, J. N. [1997] *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, Belmont, MA, 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.

Boinski, S. and Campbell, A. F. [1995] *Use of trill vocalizations to coordinate troop movement among whitefaced capuchins - a 2nd field-test*, Behaviour, **132**, 875–901.

Boissonnat, J.-D. and Cazals, F. [2002] *Smooth surface reconstruction via natural neighbour interpolation of distance functions*, Computational Geometry: Theory and Applications, **22**, 185–203.

Bollobás, B. [2001] *Random Graphs*, second edition, Cambridge University Press, Cambridge, UK, ISBN 0521809207.

Bollobás, B. and Riordan, O. [2006] *Percolation*, Cambridge University Press, Cambridge, UK.

Boltyanski, V., Martini, H., and Soltan, V. [1999] *Geometric methods and optimization problems*, volume 4 of *Combinatorial optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands, ISBN 0792354540.

Boyd, S. [2006] *Convex optimization of graph Laplacian eigenvalues*, in *Proceedings of the International Congress of Mathematicians*, volume 3.

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 electronically at `http://www.cs.technion.ac.il/tech-reports`.

Bullo, F. and Lewis, A. D. [2004] *Geometric Control of Mechanical Systems*, volume 49 of *Texts in Applied Mathematics*, Springer Verlag, New York, ISBN 0387221956.

Bulusu, N., Heidemann, J., and Estrin, D. [2001] *Adaptive beacon placement*, in *International Conference on Distributed Computing Systems*, pages 489–498, Mesa, AZ.

Burgard, W., Moors, M., Stachniss, C., and Schneider, F. E. [2005] *Coordinated multi-robot exploration*, IEEE Transactions on Robotics, **21**(3), 376–386.

Caicedo-Nùñez, C. H. and Žefran, M. [2008] *Performing coverage on nonconvex domains*, in *IEEE Multi-Conf. on Systems and Control*, San Antonio, TX, to appear.

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.

Cao, Y. U., Fukunaga, A. S., and Kahng, A. [1997] *Cooperative mobile robotics: Antecedents and directions*, Autonomous Robots, **4**(1), 7–27.

Carli, R. [2008] *Topics in the average consensus problems*, Ph.D. thesis, Universitá di Padova, Italy.

Carli, R. and Bullo, F. [2007] *Quantized coordination algorithms for rendezvous and deployment*, SIAM Journal on Control and Optimization, submitted.

Carli, R., Bullo, F., and Zampieri, S. [2008a] *Quantized average consensus via dynamic coding/decoding schemes*, International Journal on Robust and Nonlinear Control, submitted. Available electronically at `http://motion.mee.ucsb.edu/papers`.

Carli, R., Fagnani, F., Speranzon, A., and Zampieri, S. [2008b] *Communication constraints in the average consensus problem*, Automatica, **44**(3), 671–684.

Casbeer, D. W., Kingston, D. B., Beard, R. W., Mclain, T. W., Li, S.-M., and Mehra, R. [2006] *Cooperative forest fire surveillance using a team of small unmanned air vehicles*, International Journal of Systems Sciences, **37**(6), 351–360.

Cassandras, C. G. and Lafortune, S. [2007] *Introduction to Discrete-Event Systems*, second edition, Springer Verlag, ISBN 0387333320.

Cassandras, C. G. and Zhong, M. [2008] *Coverage control of sensor networks in environments with polygonal obstacles*, in *IFAC World Congress*, Seoul, Korea, to appear.

Castañón, D. A. and Wu, C. [2003] *Distributed algorithms for dynamic reassignment*, in *IEEE Conf. on Decision and Control*, pages 13–18, Maui, HI.

Chatterjee, S. and Seneta, E. [1977] *Towards consensus: Some convergence theorems on repeated averaging*, Journal of Applied Probability, **14**(1), 89–97.

Chavel, I. [1984] *Eigenvalues in Riemannian Geometry*, Academic Press, New York, ISBN 0121706400.

Chen, C.-T. [1984] *Linear System Theory and Design*, Holt, Rinehart, and Winston, New York, NY, ISBN 0030602890.

Chopra, N. and Spong, M. W. [2008] *On synchronization of Kuramoto oscillators*, IEEE Transactions on Automatic Control, to appear.

Chorin, A. J. and Marsden, J. E. [1994] *A Mathematical Introduction to Fluid Mechanics*, volume 4 of *Texts in Applied Mathematics*, third edition, Springer Verlag, New York, ISBN 0387979182.

Choset, H. [2001] *Coverage for robotics - a survey of recent results*, Annals of Mathematics and Artificial Intelligence, **31**, 113–126.

Chvátal, V. [1975] *A combinatorial theorem in plane geometry*, Journal of Combinatorial Theory. Series B, **18**, 39–41.

Clark, J. and Fierro, R. [2005] *Cooperative hybrid control of robotic sensors for perimeter detection and tracking*, in *American Control Conference*, pages 3500–3505, Portland, OR.

Clarke, F. H. [1983] *Optimization and Nonsmooth Analysis*, Canadian Mathematical Society Series of Monographs and Advanced Texts, John Wiley, ISBN 047187504X.

Cogburn, R. [1984] *The ergodic theory of Markov chains in random environments*, Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete, **66**(1).

Conradt, L. and Roper, T. J. [2003] *Group decision-making in animals*, Nature, **421**(6919), 155–158.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. [2001] *Introduction to Algorithms*, second edition, MIT Press, Cambridge, MA, ISBN 0262032937.

Cortés, J. [2006] *Finite-time convergent gradient flows with applications to network consensus*, Automatica, **42**(11), 1993–2000.

— [2007] *Distributed Kriged Kalman filter for spatial estimation*, IEEE Transactions on Automatic Control, submitted.

— [2008] *Distributed algorithms for reaching consensus on general functions*, Automatica, **44**(3), 726–737.

Cortés, J. and Bullo, F. [2005] *Coordination and geometric optimization via distributed dynamical systems*, SIAM Journal on Control and Optimization, **44**(5), 1543–1574.

Cortés, J., Martínez, S., and Bullo, F. [2005] *Spatially-distributed coverage optimization and control with limited-range interactions*, ESAIM. Control, Optimisation & Calculus of Variations, **11**, 691–719.

— [2006] *Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions*, IEEE Transactions on Automatic Control, **51**(8), 1289–1298.

Cortés, J., Martínez, S., Karatas, T., and Bullo, F. [2004] *Coverage control for mobile sensing networks*, IEEE Transactions on Robotics and Automation, **20**(2), 243–255.

Couzin, I. D., Krause, J., Franks, N. R., and Levin, S. A. [2005] *Effective leadership and decision-making in animal groups on the move*, Nature, **433**, 513–516.

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**, 279–301.

de Berg, M., van Kreveld, M., Overmars, M., and Schwarzkopf, O. [2000] *Computational Geometry: Algorithms and Applications*, second edition, Springer Verlag, New York, ISBN 3540656200.

de Gennaro, M. C. and Jadbabaie, A. [2006] *Decentralized control of connectivity for multi-agent systems*, in *IEEE Conf. on Decision and Control*, pages 3628–3633, San Diego, CA.

de Silva, V. and Ghrist, R. [2007] *Coverage in sensor networks via persistent homology*, Algebraic & Geometric Topology, **7**, 339–358.

Deffuant, G., Neau, D., Amblard, F., and Weisbuch, G. [2000] *Mixing beliefs among interacting agents*, Advances in Complex Systems, **3**, 87–98.

DeGroot, M. H. [1974] *Reaching a consensus*, Journal of the American Statistical Association, **69**(345), 118–121.

Dimarogonas, D. V. and Kyriakopoulos, K. J. [2007] *On the rendezvous problem for multiple nonholonomic agents*, IEEE Transactions on Automatic Control, **52**(5), 916–922.

Drezner, Z., (editor) [1995] *Facility Location: A Survey of Applications and Methods*, Springer Series in Operations Research, Springer Verlag, New York, ISBN 0-387-94545-8.

Drezner, Z. and Hamacher, H. W., (editors) [2001] *Facility Location: Applications and Theory*, Springer Verlag, New York, ISBN 3540421726.

Du, Q., Faber, V., and Gunzburger, M. [1999] *Centroidal Voronoi tessellations: Applications and algorithms*, SIAM Review, **41**(4), 637–676.

Dubins, L. E. [1957] *On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents*, American Journal of Mathematics, **79**, 497–516.

Dullerud, G. E. and Paganini, F. [2000] *A Course in Robust Control Theory*, number 36 in Texts in Applied Mathematics, Springer Verlag, New York, ISBN 978-0-387-98945-7.

Dynia, M., Kutylowski, J., Meyer auf der Heide, F., and Schindelhauer, C. [2006] *Smart robot teams exploring sparse trees*, in *International Symposium of Mathematical Foundations of Computer Science*, Stará Lesná, Slovakia.

Efrat, A. and Har-Peled, S. [2006] *Guarding galleries and terrains*, Information Processing Letters, **100**(6), 238–245.

Eidenbenz, S., Stamm, C., and Widmayer, P. [2001] *Inapproximability results for guarding polygons and terrains*, Algorithmica, **31**(1), 79–113.

Emerson, A. E. [1994] *Temporal and modal logic*, in *Handbook of Theoretical Computer Science, Vol. B: Formal Models and Semantics*, J. van Leeuwen, editor, pages 997–1072, MIT Press, ISBN 0262720159.

Fagnani, F., Johansson, K. H., Speranzon, A., and Zampieri, S. [2004] *On multi-vehicle rendezvous under quantized communication*, in *Mathematical Theory of Networks and Systems*, Leuven, Belgium, Electronic Proceedings.

Fagnani, F. and Zampieri, S. [2008] *Average consensus with packet drop communication*, SIAM Journal on Control and Optimization, special issue on "Control and Optimization in Cooperative Networks." Submitted.

Fainekos, G. E., Kress-Gazit, H., and Pappas, G. J. [2005] *Temporal logic motion planning for mobile robots*, in *IEEE Int. Conf. on Robotics and Automation*, pages 2032–2037, Barcelona, Spain.

Fang, L. and Antsaklis, P. J. [2008] *Asynchronous consensus protocols using nonlinear paracontractions theory*, IEEE Transactions on Automatic Control, to appear.

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.

Flocchini, P., Prencipe, G., Santoro, N., and Widmayer, P. [1999] *Hard tasks for weak robots: The role of common knowledge in pattern formation by autonomous mobile robots*, in *ISAAC 1999, 10th International Symposium on Algorithm and Computation (Chennai, India)*, A. Aggarwal and C. P.

Rangan, editors, volume 1741 of *Lecture Notes in Computer Science*, pages 93–102, Springer Verlag, New York, ISBN 3540669167.

— [2005] *Gathering of asynchronous oblivious robots with limited visibility*, Theoretical Computer Science, **337**(1-3), 147–168.

Fraigniaud, P., Gąsieniec, L., Kowalski, D. R., and Pelc, A. [2004] *Collective tree exploration*, in *LATIN 2004*, M. Farach-Colton, editor, volume 2976 of *Lecture Notes in Computer Science*, pages 141–151, Springer Verlag.

Francis, B. A. [2006] *Distributed control of autonomous mobile robots*, ECE1635 Course Notes, Version 1.01, University of Toronto, Canada.

Frazzoli, E. and Bullo, F. [2004] *Decentralized algorithms for vehicle routing in a stochastic time-varying environment*, in *IEEE Conf. on Decision and Control*, pages 3357–3363, Paradise Island, Bahamas.

Gallager, R. G. [1968] *Information Theory and Reliable Communication*, John Wiley, New York, 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.

Ganguli, A., Cortés, J., and Bullo, F. [2007a] *Distributed coverage of nonconvex environments*, in *Networked Sensing Information and Control (Proceedings of the NSF Workshop on Future Directions in Systems Research for Networked Sensing, May 2006, Boston, MA)*, V. Saligrama, editor, pages 289–305, Lecture Notes in Control and Information Sciences, Springer Verlag, ISBN 0387688439.

— [2007b] *Multirobot rendezvous with visibility sensors in nonconvex environments*, IEEE Transactions on Robotics, (Submitted Nov 2006) Conditionally Accepted.

Gao, C., Cortés, J., and Bullo, F. [2008] *Notes on averaging over acyclic digraphs and discrete coverage control*, Automatica, **44**(8), to appear.

Gazi, V. and Passino, K. M. [2003] *Stability analysis of swarms*, IEEE Transactions on Automatic Control, **48**(4), 692–697.

Gerkey, B. P. and Mataric, M. J. [2004] *A formal analysis and taxonomy of task allocation in multi-robot systems*, International Journal of Robotics Research, **23**(9), 939–954.

Ghosh, S. K. [1987] *Approximation algorithms for Art Gallery Problems*, in *Proceedings of the Canadian Information Processing Society*, pages 429–434.

Godsil, C. D. and Royle, G. F. [2001] *Algebraic Graph Theory*, volume 207 of *Graduate Texts in Mathematics*, Springer Verlag, New York, ISBN 0387952411.

Godwin, M. F., Spry, S., and Hedrick, J. K. [2006] *Distributed collaboration with limited communication using mission state estimates*, in *American Control Conference*, pages 2040–2046, Minneapolis, MN.

Goldsmith, A. [2005] *Wireless Communications*, Cambridge University Press, Cambridge, UK.

Goodman, J. E. and O'Rourke, J., (editors) [2004] *Handbook of Discrete and Computational Geometry*, second edition, CRC Press, Boca Raton, FL, ISBN 1584883014.

Graham, R. and Cortés, J. [2007] *Asymptotic optimality of multicenter Voronoi configurations for random field estimation*, IEEE Transactions on Automatic Control, submitted.

Gray, R. M. and Neuhoff, D. L. [1998] *Quantization*, IEEE Transactions on Information Theory, **44**(6), 2325–2383, Commemorative Issue 1948-1998.

Guckenheimer, J. and Holmes, P. [1990] *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, volume 42 of *Applied Mathematical Sciences*, Springer Verlag, ISBN 0387908196.

Gueron, S. and Levin, S. A. [1993] *Self-organization of front patterns in large wildebeest herds*, Journal of Theoretical Biology, **165**, 541–552.

Gupta, P. and Kumar, P. R. [2000] *The capacity of wireless networks*, IEEE Transactions on Information Theory, **46**(2), 388–404.

Gupta, V., Chung, T. H., Hassibi, B., and Murray, R. M. [2006a] *On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage*, Automatica, **42**(2), 251–260.

Gupta, V., Langbort, C., and Murray, R. M. [2006b] *On the robustness of distributed algorithms*, in *IEEE Conf. on Decision and Control*, pages 3473–3478, San Diego, CA.

Hatano, Y. and Mesbahi, M. [2005] *Agreement over random networks*, IEEE Transactions on Automatic Control, **50**, 1867–1872.

Hayes, J., McJunkin, M., and Košecká, J. [2003] *Communication enhanced navigation strategies for teams of mobile agents*, in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 2285–2290, Las Vegas, NV.

Hegselmann, R. and Krause, U. [2002] *Opinion dynamics and bounded confidence models, analysis, and simulations*, Journal of Artificial Societies and Social Simulation, **5**(3).

Hendrickx, J. M. [2008] *Graphs and Networks for the Analysis of Autonomous Agent Systems*, Ph.D. thesis, Université Catholique de Louvain, Belgium.

Hendrickx, J. M., Anderson, B. D., Delvenne, J.-C., and Blondel, V. D. [2007] *Directed graphs for the analysis of rigidity and persistence in autonomous agents systems*, International Journal on Robust and Nonlinear Control, **17**(10), 960–981.

Hernández-Peñalver, G. [1994] *Controlling guards*, in *Canadian Conference on Computational Geometry*, pages 387–392, Saskatoon, Canada.

Hibbeler, R. C. [2003] *Engineering Mechanics: Statics & Dynamics*, 10th edition, Prentice Hall, Englewood Cliffs, NJ.

Horn, R. A. and Johnson, C. R. [1985] *Matrix Analysis*, Cambridge University Press, Cambridge, UK.

Howard, A., Matarić, M. J., and Sukhatme, G. S. [2002] *Mobile sensor network deployment using potential fields: A distributed scalable solution to the area coverage problem*, in *International Conference on Distributed Autonomous Robotic Systems (DARS02)*, pages 299–308, Fukuoka, Japan.

Howard, A., Parker, L. E., and Sukhatme, G. S. [2006] *Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment, and detection*, International Journal of Robotics Research, **25**(5-6), 431–447.

Hu, J., Prandini, M., and Tomlin, C. [2007] *Conjugate points in formation constrained optimal multi-agent coordination: A case study*, SIAM Journal on Control and Optimization, **45**(6), 2119–2137.

Hussein, I. I. and Stipanovic̀, D. M. [2007] *Effective coverage control for mobile sensor networks with guaranteed collision avoidance*, IEEE Transactions on Control Systems Technology, **15**(4), 642–657.

Huygens, C. [1673] *Horologium oscilatorium*, 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.

Jaromczyk, J. W. and Toussaint, G. T. [1992] *Relative neighborhood graphs and their relatives*, Proceedings of the IEEE, **80**(9), 1502–1517.

Ji, M. and Egerstedt, M. [2007] *Distributed control of multiagent systems while preserving connectedness*, IEEE Transactions on Robotics, **23**(4), 693–703.

Justh, E. W. and Krishnaprasad, P. S. [2004] *Equilibria and steering laws for planar formations*, Systems & Control Letters, **52**(1), 25–38.

— [2006] *Steering laws for motion camouflage*, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, **462**(2076), 3629–3643.

Kang, K., Yan, J., and Bitmead, R. R. [2006] *Communication resources for disturbance rejection in coordinated vehicle control*, in *IEEE Conf. on Decision and Control and European Control Conference*, pages 5730–5735, Seville, Spain.

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, Englewood Cliffs, NJ, ISBN 0130673897.

Kim, Y. and Mesbahi, M. [2006] *On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian*, IEEE Transactions on Automatic Control, **51**(1), 116–120.

Klavins, E. [2003] *Communication complexity of multi-robot systems*, in *Algorithmic Foundations of Robotics V*, J.-D. Boissonnat, J. W. Burdick, K. Goldberg, and S. Hutchinson, editors, volume 7 of *Springer Tracts in Advanced Robotics*, Springer Verlag, Berlin Heidelberg, ISBN 3540404767.

Klavins, E., Ghrist, R., and Lipsky, D. [2006] *A grammatical approach to self-organizing robotic systems*, IEEE Transactions on Automatic Control, **51**(6), 949–962.

Klavins, E. and Murray, R. M. [2004] *Distributed algorithms for cooperative control*, IEEE Pervasive Computing, **3**(1), 56–65.

Korte, B. and Vygen, J. [2005] *Combinatorial Optimization: Theory and Algorithms*, third edition, number 21 in Algorithmics and Combinatorics, Springer Verlag, New York, 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.

Krick, L. [2007] *Application of Graph Rigidity in Formation Control of Multi-Robot Networks*, Master's thesis, University of Toronto, Canada.

Kuramoto, Y. [1975] *Self-entrainment of a population of coupled non-linear oscillators*, in *International Symposium on Mathematical Problems in Theoretical Physics*, volume 39 of *Lecture Notes in Physics*, pages 420–422, Springer Verlag, New York.

— [1984] *Chemical oscillations, waves, and turbulence*, Springer Verlag, ISBN 0387133224.

Kwok, A. and Martínez, S. [2008] *Deployment algorithms for a power-constrained mobile sensor network*, International Journal on Robust and Nonlinear Control, submitted.

Landau, H. J. and Odlyzko, A. M. [1981] *Bounds for eigenvalues of certain stochastic matrices*, Linear Algebra and its Applications, **38**, 5–15.

Langbort, C. and Gupta, V. [2007] *Minimal interconnection topology in distributed control*, SIAM Journal on Control and Optimization, special issue on "Control and Optimization in Cooperative Networks." Submitted.

Langetepe, E. and Zachmann, G. [2006] *Geometric Data Structures for Computer Graphics*, A. K. Peters, Ltd., Wellesley, MA.

Lanthier, M., Nussbaum, D., and Wang, T.-J. [2005] *Calculating the meeting point of scattered robots on weighted terrain surfaces*, in *Computing: The Australasian Theory Symposium (CATS)*, volume 27, pages 107–118, Newcastle, Australia.

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 Verlag, New York.

Laventall, K. and Cortés, J. [2008] *Coverage control by robotic networks with limited-range anisotropic sensory*, International Journal of Control, submitted.

Lee, D. and Spong, M. W. [2007] *Stable flocking of multiple inertial agents on balanced graphs*, IEEE Transactions on Automatic Control, **52**(8), 1469–1475.

Lee, D. T. and Lin, A. K. [1986] *Computational complexity of art gallery problems*, IEEE Transactions on Information Theory, **32**(2), 276–282.

Lekien, F. and Leonard, N. E. [2007] *Non-uniform coverage and cartograms*, SIAM Journal on Control and Optimization, special issue on "Control and Optimization in Cooperative Networks." Submitted.

Li, X.-Y. [2003] *Algorithmic, geometric and graphs issues in wireless networks*, Wireless Communications and Mobile Computing, **3**(2), 119–140.

Lin, J., Morse, A. S., and Anderson, B. D. O. [2007a] *The multi-agent rendezvous problem. part 1: The synchronous case*, SIAM Journal on Control and Optimization, **46**(6), 2096–2119.

— [2007b] *The multi-agent rendezvous problem. part 2: The asynchronous case*, SIAM Journal on Control and Optimization, **46**(6), 2120–2147.

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.

— [2007c] *State agreement for continuous-time coupled nonlinear systems*, SIAM Journal on Control and Optimization, **46**(1), 288–307.

Lloyd, E. L., Liu, R., Marathe, M. V., Ramanathan, R., and Ravi, S. S. [2005] *Algorithmic aspects of topology control problems for ad hoc networks*, Mobile Networks and Applications, **10**(1-2), 19–34.

Lorenz, J. [2007] *Repeated Averaging and Bounded Confidence - Modeling, Analysis and Simulation of Continuous Opinion Dynamics*, Ph.D. thesis, University of Bremen, Germany, available electronically 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 electronically 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, Budapest, Hungary.

Lumelsky, V. J. and Harinarayan, K. R. [1997] *Decentralized motion planning for multiple mobile robots: the cocktail party model*, Autonomous Robots, **4**(1), 121–135.

Lygeros, J., Johansson, K. H., Simić, S. N., Zhang, J., and Sastry, S. S. [2003] *Dynamical properties of hybrid automata*, IEEE Transactions on Automatic Control, **48**(1), 2–17.

Lynch, N. A. [1997] *Distributed Algorithms*, Morgan Kaufmann Publishers, San Mateo, CA, ISBN 1558603484.

Lynch, N. A., Segala, R., and Vaandrager, F. [2003] *Hybrid I/O automata*, Information and Computation, **185**(1), 105–157.

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. [2007] *Distributed interpolation schemes for field estimation by mobile sensor networks*, IEEE Transactions on Control Systems Technology, submitted.

— [2008] *Practical multiagent rendezvous through modified circumcenter algorithms*, Automatica, submitted.

Martínez, S. and Bullo, F. [2006] *Optimal sensor placement and motion coordination for target tracking*, Automatica, **42**(4), 661–668.

Martínez, S., Bullo, F., Cortés, J., and Frazzoli, E. [2007a] *On synchronous robotic networks – Part I: Models, tasks and complexity*, IEEE Transactions on Automatic Control, **52**(12), 2199–2213.

— [2007b] *On synchronous robotic networks – Part II: Time complexity of rendezvous and deployment algorithms*, IEEE Transactions on Automatic Control, **52**(12), 2214–2226.

Martínez, S., Cortés, J., and Bullo, F. [2007c] *Motion coordination with distributed information*, IEEE Control Systems Magazine, **27**(4), 75–88.

Meester, R. and Roy, R. [2008] *Continuum Percolation*, Cambridge University Press, Cambridge, UK.

Merris, R. [1994] *Laplacian matrices of a graph: A survey*, Linear Algebra its Applications, **197**, 143–176.

Mesbahi, M. [2005] *On state-dependent dynamic graphs and their controllability properties*, IEEE Transactions on Automatic Control, **50**(3), 387–392.

Meyer, C. D. [2001] *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, PA, ISBN 0898714540.

Miller, M. B. and Bassler, B. L. [2001] *Quorum sensing in bacteria*, Annual Review of Microbiology, **55**, 165–199.

Mirollo, R. E. and Strogatz, S. H. [1990] *Synchronization of pulse-coupled biological oscillators*, SIAM Journal on Applied Mathematics, **50**(6), 1645–1662.

Mitchell, J. S. B. [1997] *Shortest paths and networks*, in *Handbook of Discrete and Computational Geometry*, J. E. Goodman and J. O'Rourke, editors, chapter 24, pages 445–466, CRC Press, Boca Raton, FL, ISBN 0849385245.

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.

Moore, B. J. and Passino, K. M. [2007] *Distributed task assignment for mobile agents*, IEEE Transactions on Automatic Control, **52**(4), 749–753.

Moreau, L. [2003] *Time-dependent unidirectional communication in multi-agent systems*, Available electronically at `http://arxiv.org/abs/math/0306426`.

— [2004] *Stability of continuous-time distributed consensus algorithms*, Available electronically 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.

Moses, Y. and Tennenholtz, M. [1995] *Artificial social systems*, Computers and AI, **14**(6), 533–562.

Moshtagh, N. and Jadbabaie, A. [2007] *Distributed geodesic control laws for flocking of nonholonomic agents*, IEEE Transactions on Automatic Control, **52**(4), 681–686.

Nijmeijer, H. [2001] *A dynamical control view on synchronization*, Physica D, **154**(3-4), 219–228.

Ögren, P., Fiorelli, E., and Leonard, N. E. [2004] *Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment*, IEEE Transactions on Automatic Control, **49**(8), 1292–1302.

Oh, S., Schenato, L., Chen, P., and Sastry, S. S. [2007] *Tracking and coordination of multiple agents using sensor networks: system design, algorithms and experiments*, Proceedings of the IEEE, **95**(1), 163–187.

Okabe, A., Boots, B., Sugihara, K., and Chiu, S. N. [2000] *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, second edition, Wiley Series in Probability and Statistics, John Wiley, New York, ISBN 0471986356.

Okubo, A. [1986] *Dynamical aspects of animal grouping: swarms, schools, flocks and herds*, Advances in Biophysics, **22**, 1–94.

Olfati-Saber, R. [2006] *Flocking for multi-agent dynamic systems: Algorithms and theory*, IEEE Transactions on Automatic Control, **51**(3), 401–420.

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 Verlag, New York.

Olfati-Saber, R. and Murray, R. M. [2002] *Graph rigidity and distributed formation stabilization of multi-vehicle systems*, in *IEEE Conf. on Decision and Control*, pages 2965–2971, Las Vegas, NV.

— [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. [2007] *Convergence speed in distributed consensus and averaging*, SIAM Journal on Control and Optimization, special issue on "Control and Optimization in Cooperative Networks." Submitted.

O'Rourke, J. [2000] *Computational Geometry in C*, Cambridge University Press, ISBN 0521649765.

Pallottino, L., Scordio, V. G., Frazzoli, E., and Bicchi, A. [2007] *Decentralized cooperative policy for conflict resolution in multi-vehicle systems*, IEEE Transactions on Robotics, **23**(6), 1170–1183.

Papachristodoulou, A. and Jadbabaie, A. [2006] *Synchonization in oscillator networks with heterogeneous delays, switching topologies and nonlinear dynamics*, in *IEEE Conf. on Decision and Control*, pages 4307–4312, San Diego, CA.

Parhami, B. [1999] *Introduction to Parallel Processing: Algorithms and Architectures*, Plenum Series in Computer Science, Springer Verlag, New York.

Parrish, J. K., Viscido, S. V., and Grunbaum, D. [2002] *Self-organized fish schools: an examination of emergent properties*, Biological Bulletin, **202**, 296–305.

Passino, K. M. [2004] *Biomimicry for Optimization, Control, and Automation*, Springer Verlag, New York, ISBN 1852338040.

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.

Pavone, M. and Frazzoli, E. [2007] *Decentralized policies for geometric pattern formation and path coverage*, ASME Journal on Dynamic Systems, Measurement, and Control, to appear.

Pavone, M., Frazzoli, E., and Bullo, F. [2007] *Decentralized algorithms for stochastic and dynamic vehicle routing with general target distribution*, in *IEEE Conf. on Decision and Control*, pages 4869–4874, New Orleans, LA.

Payton, D., Daily, M., Estowski, R., Howard, M., and Lee, C. [2001] *Pheromone robotics*, Autonomous Robots, **11**, 319–324.

Pearl, J. [1988] *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Francisco, CA, ISBN 1558604790.

Peleg, D. [2000] *Distributed Computing. A Locality-Sensitive Approach*, Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, PA, ISBN 0898714648.

Penrose, M. [2003] *Random Geometric Graphs*, Oxford Studies in Probability, Oxford University Press, Oxford, UK, ISBN 0198506260.

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.

Pimenta, L. C. A., Kumar, V., Mesquita, R. C., and Pereira, G. A. S. [2008] *Sensing and coverage for a network of heterogeneous robots*, in *IEEE Conf. on Decision and Control*, Cancun, Mexico, submitted.

Pinciu, V. [2003] *A coloring algorithm for finding connected guards in art galleries*, in *Discrete Mathematical and Theoretical Computer Science*, volume 2731/2003 of *Lecture Notes in Computer Science*, pages 257–264, Springer Verlag.

Poduri, S. and Sukhatme, G. S. [2004] *Constrained coverage for mobile sensor networks*, in *IEEE Int. Conf. on Robotics and Automation*, pages 165–172, 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.

Preparata, F. P. and Shamos, M. I. [1993] *Computational Geometry: An Introduction*, Springer Verlag, New York, ISBN 0387961313.

Radke, J. D. [1988] *On the shape of a set of points*, in *Computational Morphology*, G. T. Toussaint, editor, pages 105–136, Elsevier, Oxford, UK.

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.

Rathinam, S., Sengupta, R., and Darbha, S. [2007] *A resource allocation algorithm for multi-vehicle systems with non holonomic constraints*, IEEE Transactions on Automation Sciences and Engineering, **4**(1), 98–104.

Reeds, J. A. and Shepp, L. A. [1990] *Optimal paths for a car that goes both forwards and backwards*, Pacific Journal of Mathematics, **145**(2), 367–393.

Rekleitis, I. M., Dudek, G., and Milios, E. [2001] *Multi-robot collaboration for robust exploration*, Annals of Mathematics and Artificial Intelligence, **31**(1-4), 7–40.

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 Verlag, New York, 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.

Robert, J.-M. and Toussaint, G. T. [1990] *Computational geometry and facility location*, in *International Conf. on Operations Research and Management Science*, volume B, pages 1–19, Manila, The Philippines.

Roy, N. and Dudek, G. [2001] *Collaborative exploration and rendezvous: Algorithms, performance bounds, and observations*, Autonomous Robots, **11**(2), 117–136.

Sack, J. R. and Urrutia, J., (editors) [2000] *Handbook of Computational Geometry*, North-Holland, Amsterdam, the Netherlands, ISBN 0444825371.

Sanfelice, R. G., Goebel, R., and Teel, A. R. [2007] *Invariance principles for hybrid systems with connections to detectability and asymptotic stability*, IEEE Transactions on Automatic Control, **52**(12), 2282–2297.

Santi, P. [2005] *Topology Control in Wireless Ad Hoc and Sensor Networks*, John Wiley, New York, ISBN 0470094532.

Santoro, N. [2001] *Distributed computations by autonomous mobile robots*, in *SOFSEM 2001: Conference on Current Trends in Theory and Practice of Informatics (Piestany, Slovak Republic)*, L. Pacholski and P. Ruzicka, editors, volume 2234 of *Lecture Notes in Computer Science*, pages 110–115, Springer Verlag, New York, ISBN 3-540-42912-3.

Sarlette, A. and Sepulchre, R. [2007] *Consensus optimization on manifolds*, SIAM Journal on Control and Optimization, special issue on "Control and Optimization in Cooperative Networks." Submitted.

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.

Savla, K., Bullo, F., and Frazzoli, E. [2007a] *Traveling Salesperson Problems for a double integrator*, IEEE Transactions on Automatic Control, (Submitted Nov 2006) To appear.

Savla, K., Frazzoli, E., and Bullo, F. [2008] *Traveling Salesperson Problems for the Dubins vehicle*, IEEE Transactions on Automatic Control, **53**(9), (Submitted Jun 2006) To appear.

Savla, K., Notarstefano, G., and Bullo, F. [2007b] *Maintaining limited-range connectivity among second-order agents*, SIAM Journal on Control and Optimization, special issue on "Control and Optimization in Cooperative Networks." Submitted.

Scardovi, L., Sarlette, A., and Sepulchre, R. [2007] *Synchronization and balancing on the N-torus*, Systems & Control Letters, **56**(5), 335–341.

Schultz, A. C. and Parker, L. E., (editors) [2002] *Multi-Robot Systems: From Swarms to Intelligent Automata*, Kluwer Academic Publishers, Dordrecht, The Netherlands, ISBN 1402006799, proceedings from the 2002 NRL Workshop on Multi-Robot Systems.

Schumacher, C., Chandler, P. R., Rasmussen, S. J., and Walker, D. [2003] *Task allocation for wide area search munitions with variable path length*, in *American Control Conference*, pages 3472–3477, Denver, CO.

Schuresko, M. and Cortés, J. [2007] *Safe graph rearrangements for distributed connectivity of robotic networks*, in *IEEE Conf. on Decision and Control*, pages 4602–4607, New Orleans, LA.

Schwager, M., Bullo, F., Skelly, D., and Rus, D. [2008a] *A ladybug exploration strategy for distributed adaptive coverage control*, in *IEEE Int. Conf. on Robotics and Automation*, Pasadena, CA, to appear.

Schwager, M., Rus, D., and Slotine, J. J. [2008b] *Decentralized, adaptive control for coverage with networked robots*, International Journal of Robotics Research, submitted.

Seeley, T. D. and Buhrman, S. C. [1999] *Group decision-making in swarms of honey bees*, Behavioral Ecology and Sociobiology, **45**, 19–31.

Seneta, E., (editor) [1981] *Non-negative Matrices and Markov Chains*, second edition, Springer Verlag, New York, ISBN 0387297650.

Sepulchre, R., Paley, D., and Leonard, N. E. [2007] *Stabilization of planar collective motion: All-to-all communication*, IEEE Transactions on Automatic Control, **52**, 811–824.

Sharma, V., Savchenko, M., Frazzoli, E., and Voulgaris, P. [2007] *Transfer time complexity of conflict-free vehicle routing with no communications*, International Journal of Robotics Research, **26**(3), 255–272.

Sibson, R. [1981] *A brief description of natural neighbour interpolation*, in *Interpreting Multivariate Data*, V. Barnett, editor, pages 21–36, John Wiley, ISBN 0471280399.

Simmons, R., Apfelbaum, D., Fox, D., Goldman, R., Haigh, K., Musliner, D., Pelican, M., and Thrun, S. [2000] *Coordinated deployment of multiple heterogenous robots*, in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 2254–2260, Takamatsu, Japan.

Sinclair, A. R. [1977] *The African Buffalo, A Study of Resource Limitation of Population*, The University of Chicago Press, Chicago, IL.

Sipser, M. [2005] *Introduction to the Theory of Computation*, second edition, Course Technology, ISBN 0534950973.

Skyum, S. [1991] *A simple algorithm for computing the smallest enclosing circle*, Information Processing Letters, **37**(3), 121–125.

Smith, R. S. and Hadaegh, F. Y. [2007] *Closed-loop dynamics of cooperative vehicle formations with parallel estimators and communication*, IEEE Transactions on Automatic Control, **52**(8), 1404–1414.

Smith, S. L., Broucke, M. E., and Francis, B. A. [2005] *A hierarchical cyclic pursuit scheme for vehicle networks*, Automatica, **41**(6), 1045–1053.

— [2007] *Curve shortening and the rendezvous problem for mobile autonomous robots*, IEEE Transactions on Automatic Control, **52**(6), 1154–1159.

Smith, S. L. and Bullo, F. [2007] *Monotonic target assignment for robotic networks*, IEEE Transactions on Automatic Control, submitted.

Spanos, D. P. and Murray, R. M. [2005] *Motion planning with wireless network constraints*, in *American Control Conference*, pages 87–92, Portland, OR.

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 (IPSN)*, pages 133–139, Los Angeles, CA.

Spong, M. W., Hutchinson, S., and Vidyasagar, M. [2006] *Robot Modeling and Control*, third edition, John Wiley, New York, ISBN 0-471-64990-2.

Stewart, K. J. and Harcourt, A. H. [1994] *Gorillas vocalizations during rest periods - signals of impending departure*, Behaviour, **130**, 29–40.

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, New York.

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.

Susca, S., Martínez, S., and Bullo, F. [2007] *Gradient algorithms for polygonal approximation of convex contours*, Automatica, to appear.

— [2008] *Monitoring environmental boundaries with a robotic sensor network*, IEEE Transactions on Control Systems Technology, **16**(2), 288–296.

Suzuki, I. and Yamashita, M. [1999] *Distributed anonymous mobile robots: Formation of geometric patterns*, SIAM Journal on Computing, **28**(4), 1347–1363.

Tabuada, P., Pappas, G. J., and Lima, P. [2005] *Motion feasibility of multi-agent formations*, IEEE Transactions on Robotics, **21**(3), 387–392.

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.

— [2008] *Consensus over random networks*, IEEE Transactions on Automatic Control, **53**, 791 – 795.

Tang, Z. and Özgüner, Ü. [2005] *Motion planning for multi-target surveillance with mobile sensor agents*, IEEE Transactions on Robotics, **21**(5), 898–908.

Tanner, H. G., Jadbabaie, A., and Pappas, G. J. [2007] *Flocking in fixed and switching networks*, IEEE Transactions on Automatic Control, **52**(5), 863–868.

Tanner, H. G., Pappas, G. J., and Kumar, V. [2004] *Leader-to-formation stability*, IEEE Transactions on Robotics and Automation, **20**(3), 443–455.

Tel, G. [2001] *Introduction to Distributed Algorithms*, second edition, Cambridge University Press, Cambridge, UK, ISBN 0521794838.

Toh, C.-K. [2001] *Ad Hoc Mobile Wireless Networks: Protocols and Systems*, Prentice Hall, Englewood Cliffs, NJ.

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 Verlag, New York.

Tsitsiklis, J. N. [1984] *Problems in Decentralized Decision Making and Computation*, Ph.D. thesis, MIT, Technical Report LIDS-TH-1424, Laboratory for Information and Decision Systems.

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.

Tutuncu, R. H., Toh, K. C., and Todd, M. J. [2003] *Solving semidefinite-quadratic-linear programs using SDPT3*, Mathematical Programming Ser. B, **95**, 189–217.

Vazirani, V. V. [2001] *Approximation Algorithms*, Springer Verlag, 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**, 220–221.

Whiteley, W. [1997] *Rigidity and scene analysis*, in *Handbook of Discrete and Computational Geometry*, J. E. Goodman and J. O'Rourke, editors, chapter 49, pages 893–916, CRC Press, Boca Raton, FL, ISBN 0849385245.

Wiener, N. [1958] *Nonlinear Problems in Random Theory*, MIT Press, Cambridge, MA.

Winfield, A. F. [2000] *Distributed sensing and data collection via broken ad hoc wireless connected networks of mobile robots*, in *Distributed Autonomous Robotics Systems*, L. E. Parker, G. W. Bekey, and J. Barhen, editors, volume 4, pages 273–282, Springer Verlag, New York.

Winfree, A. T. [1980] *The Geometry of Biological Time*, Springer Verlag, New York, 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 (IPSN)*, pages 63–70, Los Angeles, CA.

Xue, F. and Kumar, P. R. [2004] *The number of neighbors needed for connectivity of wireless networks*, Wireless Networks, **10**(2), 169–181.

Yang, P., Freeman, R. A., and Lynch, K. M. [2008] *Multi-agent coordination by decentralized estimation and control*, IEEE Transactions on Automatic Control, to appear.

Yu, C., Anderson, B. D., Dasgupta, S., and Fidan, B. [2008a] *Control of minimally persistent formations in the plane*, SIAM Journal on Control and Optimization, special issue on "Control and Optimization in Cooperative Networks." Submitted.

Yu, J., Liberzon, D., and LaValle, S. M. [2008b] *Rendezvous without coordinates*, in *IEEE Conf. on Decision and Control*, Cancun, Mexico, submitted.

Zavlanos, M. M. and Pappas, G. J. [2005] *Controlling connectivity of dynamic graphs*, in *IEEE Conf. on Decision and Control and European Control Conference*, pages 6388–6393, Seville, Spain.

— [2007a] *Dynamic assignment in distributed motion planning with local information*, in *American Control Conference*, pages 1173–1178, New York.

— [2007b] *Potential fields for maintaining connectivity of mobile networks*, IEEE Transactions on Robotics, **23**(4), 812–816.

Zhang, F. and Leonard, N. E. [2005] *Generating contour plots using multiple sensor platforms*, in *IEEE Swarm Intelligence Symposium*, pages 309–316, Pasadena, CA.

Zheng, Z., Spry, S. C., and Girard, A. R. [2008] *Leaderless formation control using dynamic extension and sliding control*, in *IFAC World Congress*, Seoul, Korea, to appear.

Zhu, M. and Martínez, S. [2008a] *Dynamic average consensus on synchronous communication networks*, in *American Control Conference*, Seattle, WA, to appear.

— [2008b] *On the convergence time of distributed quantized averaging algorithms*, in *IEEE Conf. on Decision and Control*, Cancun, Mexico, submitted.