Lecture #2: Models and Complexity of Robotic Networks

Francesco Bullo[1] **Jorge Cortés**[2] Sonia Martínez[2]

[1]Department of Mechanical Engineering
University of California, Santa Barbara
bullo@engineering.ucsb.edu

[2]Mechanical and Aerospace Engineering
University of California, San Diego
{cortes,soniamd}@ucsd.edu

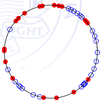
Workshop on "Distributed Control of Robotic Networks"
IEEE Conference on Decision and Control
Cancun, December 8, 2008

Acknowledgements: Emilio Frazzoli

Summary introduction

- Model for robotic networks that communicate and process information at discrete time instants, and move in continuous time
- Draw analogies with treatment on distributed algorithms for synchronous networks in previous lecture
- Special attention to spatial component – proximity graphs
- Illustrate complexity notions in simple agree-and-pursue example

Direction agreement and equidistance



Network size is unknown to agents

Problem (Direction agreement & equidistance)

Assume agents move in circle according to first-order integrator dynamics. Some move clockwise, others counterclockwise

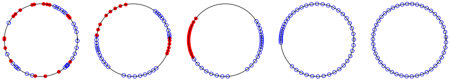
Agents talk to other agents within distance r

Objective: *agree on a common direction of motion and uniformly deploy over circle*

The agree-and-pursue algorithm

To solve the direction agreement and equidistance problem, each agent

- sets max UID received so far to its own UID
- initially transmits its direction of motion and UID to neighbors
- **at each communication round:** listens to messages from other agents and compares the received UIDs from agents moving toward its position with its own UID. If max UID is larger than own UID, resets UID and direction of motion
- **between communication rounds:** moves $k_{\text{prop}} \in (0, 1/2)$ times the distance to the immediately next neighbor in chosen direction, or, if no neighbors, k_{prop} times communication range r



The agree-and-pursue algorithm – cont

The agree-and-pursue algorithm **solves** the direction agreement and equidistance problem on a circle

- all agents agree on a common direction of motion – either clockwise or counterclockwise
- network asymptotically achieves uniform, equally-spaced rotating configuration

New issues arise when considering **robotic networks**

- As agents move, interconnection topology changes (e.g., network might be disconnected, and then leader election would not work)
- Tasks might not be achieved exactly, but asymptotically (e.g., equidistance)
- Need to **rethink model** and notions of **complexity** to account for **spatial component**

Proximity graphs model interconnection topology

Proximity graph
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

Appear in computational geometry and topology control of wireless networks

Definition (Proximity graph)

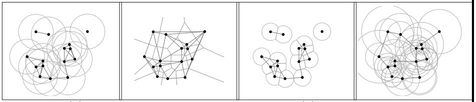
Let X be a d-dimensional space chosen among $\mathbb{R}^d$, $\mathbb{S}^d$, and $\mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$, with $d_1 + d_2 = d$. Let $\mathbb{G}(X)$ be the set of all undirected graphs whose vertex set is an element of $\mathbb{F}(X)$ (finite subsets of X)

A *proximity graph* $\mathcal{G} : \mathbb{F}(X) \to \mathbb{G}(X)$ associates to $\mathcal{P} = \{p_1, \dots, p_n\} \subset X$ an undirected graph with vertex set $\mathcal{P}$ and edge set $\mathcal{E}_{\mathcal{G}}(\mathcal{P}) \subseteq \{(p, q) \in \mathcal{P} \times \mathcal{P} \mid p \neq q\}$.

Examples of proximity graphs

On $(\mathbb{R}^d, \mathsf{dist}_2)$, $(\mathbb{S}^d, \mathsf{dist}_g)$, or $(\mathbb{R}^{d_1} \times \mathbb{S}^{d_2}, (\mathsf{dist}_2, \mathsf{dist}_g))$

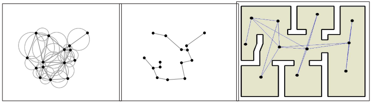
1. the **r-disk graph** $\mathcal{G}_{\text{disk}}(r)$, for $r \in \mathbb{R}_{>0}$, with $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{disk}}(r)}(\mathcal{P})$ if $\mathsf{dist}(p_i, p_j) \leq r$
2. the **Delaunay graph** $\mathcal{G}_{\text{D}}$, with $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{D}}}(\mathcal{P})$ if $V_i(\mathcal{P}) \cap V_j(\mathcal{P}) \neq \emptyset$ Definition
3. the **r-limited Delaunay graph** $\mathcal{G}_{\text{LD}}(r)$, for $r \in \mathbb{R}_{>0}$, with $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{LD}}(r)}(\mathcal{P})$ if $V_{i,\frac{r}{2}}(\mathcal{P}) \cap V_{j,\frac{r}{2}}(\mathcal{P}) \neq \emptyset$ Definition
4. the **relative neighborhood graph** $\mathcal{G}_{\text{RN}}$, with $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{RN}}}(\mathcal{P})$ if $p_k \notin B(p_i, \mathsf{dist}(p_i, p_j)) \cap B(p_j, \mathsf{dist}(p_i, p_j))$ for all $p_k \in \mathcal{P}$



$\mathcal{G}_{\text{disk}}(r)$ $\mathcal{G}_{\text{D}}$ $\mathcal{G}_{\text{LD}}(r)$ $\mathcal{G}_{\text{RN}}$

More examples of proximity graphs on Euclidean space

1. the *Gabriel graph* $\mathcal{G}_{\text{G}}$, with $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{G}}}(\mathcal{P})$ if $p_k \notin B\big(\frac{p_i+p_j}{2}, \frac{\mathsf{dist}(p_i,p_j)}{2}\big)$ for all $p_k \in \mathcal{P}$
2. the *Euclidean minimum spanning tree* $\mathcal{G}_{\text{EMST}}$, that assigns to each $\mathcal{P}$ a minimum-weight spanning tree of the complete weighted digraph $(\mathcal{P}, \{(p, q) \in \mathcal{P} \times \mathcal{P} \mid p \neq q\}, A)$, with weighted adjacency matrix $a_{ij} = \|p_i - p_j\|_2$, for $i, j \in \{1, \dots, n\}$
3. given a simple polygon Q in $\mathbb{R}^2$, the *visibility graph* $\mathcal{G}_{\text{vis},}$, with $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{vis},}}(\mathcal{P})$ if the closed segment $[p_i, p_j]$ from p_i to p_j is contained in Q



$\mathcal{G}_{\text{G}}$ $\mathcal{G}_{\text{EMST}}$ $\mathcal{G}_{\text{vis},}$

Set of neighbors map

For proximity graph $\mathcal{G}$, $p \in X$, and $\mathcal{P} = \{p_1, \dots, p_n\} \in \mathbb{F}(X)$

associate **set of neighbors** map $\mathcal{N}_{\mathcal{G},p} : \mathbb{F}(X) \to \mathbb{F}(X)$

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

Typically, p is a point in $\mathcal{P}$, but this works for any $p \in X$

When does a proximity graph provide sufficient information to compute another proximity graph?

Spatially distributed graphs

E.g., if a node knows position of its neighbors in the complete graph, then it can compute its neighbors with respect to any proximity graph

Formally, given $\mathcal{G}_1$ and $\mathcal{G}_2$,

1. $\mathcal{G}_1$ is a **subgraph** of $\mathcal{G}_2$, 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 $\mathcal{P} \in \mathbb{F}(X)$
2. $\mathcal{G}_1$ is **spatially distributed over** $\mathcal{G}_2$ if, for all $p \in \mathcal{P}$, ▸ Illustration

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

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

$\mathcal{G}_1$ spatially distributed over $\mathcal{G}_2 \implies \mathcal{G}_1 \subset \mathcal{G}_2$

Converse not true: $\mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r) \subset \mathcal{G}_{\mathrm{disk}}$, but $\mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r)$ not spatially distributed over $\mathcal{G}_{\mathrm{disk}}(r)$ ▸ Illustration

Inclusion relationships among proximity graphs

Theorem

For $r \in \mathbb{R}_{>0}$, the following statements hold:

1. $\mathcal{G}_{\mathrm{EMST}} \subset \mathcal{G}_{\mathrm{RN}} \subset \mathcal{G}_{\mathrm{G}} \subset \mathcal{G}_{\mathrm{D}}$;
2. $\mathcal{G}_{\mathrm{G}} \cap \mathcal{G}_{\mathrm{disk}}(r) \subset \mathcal{G}_{\mathrm{LD}}(r) \subset \mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r)$
3. $\mathcal{G}_{\mathrm{RN}} \cap \mathcal{G}_{\mathrm{disk}}(r)$, $\mathcal{G}_{\mathrm{G}} \cap \mathcal{G}_{\mathrm{disk}}(r)$, *and* $\mathcal{G}_{\mathrm{LD}}(r)$ *are spatially distributed over* $\mathcal{G}_{\mathrm{disk}}(r)$

The inclusion $\mathcal{G}_{\mathrm{LD}}(r) \subset \mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r)$ is in general strict

Since $\mathcal{G}_{\mathrm{EMST}}$ is by definition connected, (1) implies that $\mathcal{G}_{\mathrm{RN}}$, $\mathcal{G}_{\mathrm{G}}$ and $\mathcal{G}_{\mathrm{D}}$ are connected

Connectivity properties of $\mathcal{G}_{\mathrm{disk}}(r)$

Theorem

For $r \in \mathbb{R}_{>0}$, the following statements hold:

1. $\mathcal{G}_{\mathrm{EMST}} \subset \mathcal{G}_{\mathrm{disk}}(r)$ *if and only if* $\mathcal{G}_{\mathrm{disk}}(r)$ *is connected;*
2. $\mathcal{G}_{\mathrm{EMST}} \cap \mathcal{G}_{\mathrm{disk}}(r)$, $\mathcal{G}_{\mathrm{RN}} \cap \mathcal{G}_{\mathrm{disk}}(r)$, $\mathcal{G}_{\mathrm{G}} \cap \mathcal{G}_{\mathrm{disk}}(r)$ *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).*

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 $\exists$ a map $\tilde{T} : X \times \mathbb{F}(X) \to Y$ such that for all $(p_1, \dots, p_n) \in X^n$ and for all $j \in \{1, \dots, n\}$,

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

where T_j denotes the jth-component of T

Equivalently,

the jth component of a spatially distributed map at $(p_1, \dots, p_n)$ can be computed with the knowledge of the vertex p_j and the neighboring vertices in the undirected graph $\mathcal{G}(P)$

Physical components of a robotic network

Group of robots with the ability to exchange messages, perform local computations, and control motion

Mobile robot: continuous-time continuous-space dynamical system,

1. 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*;
2. U is a compact subset of $\mathbb{R}^m$ containing $\mathbf{0}$, called the *input space*;
3. X_0 is a subset of X, called the *set of allowable initial states*;
4. $f : X \times U \to \mathbb{R}^d$ is a smooth control vector field on X

Synchronous robotic network

Definition (Robotic network)

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

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

Map $x \mapsto (I, E_{\text{cmm}}(x))$ models topology of the communication service among robots – proximity graph induced by network capabilities

A couple of examples

Locally-connected first-order robots in $\mathbb{R}^d$: $\mathcal{S}_{\text{disk}}$
n points $x^{[1]}, \dots, x^{[n]}$ in $\mathbb{R}^d$, $d \geq 1$, obeying $\dot{x}^{[i]}(t) = u^{[i]}(t)$, with $u^{[i]} \in [-u_{\max}, u_{\max}]$. These are identical robots of the form

$$(\mathbb{R}^d, [-u_{\max}, u_{\max}]^d, \mathbb{R}^d, (\mathbf{0}, \boldsymbol{e}_1, \dots, \boldsymbol{e}_d))$$

Each robot can communicate to other robots within r, $\mathcal{G}_{\text{disk}}(r)$ on $\mathbb{R}^d$

Locally-connected first-order robots in $\mathbb{S}^1$: $\mathcal{S}_{\text{circle}}$
n robots $\theta^{[1]}, \dots, \theta^{[n]}$ in $\mathbb{S}^1$, moving along on the unit circle with angular velocity equal to the control input. Each robot is described by

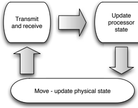
$$(\mathbb{S}^1, [-u_{\max}, u_{\max}], \mathbb{S}^1, (0, \boldsymbol{e}))$$

($\boldsymbol{e}$ describes unit-speed counterclockwise rotation). Each robot can communicate to other robots within r along the circle, $\mathcal{G}_{\text{disk}}(r)$ on $\mathbb{S}^1$

Uniform control and communication law

1. communication schedule — $\mathbb{Z}_{\geq 0} = \{t_\ell\}_{\ell \in \mathbb{Z}_{\geq 0}} \subset \mathbb{R}_{\geq 0}$
2. communication alphabet — $\mathbb{A}$ including the `null` message
3. processor state space — W, with initial allowable $W_0^{[i]}$
4. message-generation function — $\text{msg} : \mathbb{Z}_{\geq 0} \times X \times W \times I \to L$
5. state-transition functions — $\text{stf} : X \times W \times L^n \to W$
6. control function — $\text{ctl} : \mathbb{Z}_{\geq 0} \times X \times W \times L^n \to U$

Execution: discrete-time communication
discrete-time computation
continuous-time motion



The agree-and-pursue algorithm – formally

`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)
return (θ, w)

function stf(w, y)
for each non-`null` message $(\theta_{\text{rcvd}}, (\texttt{dir}_{\text{rcvd}}, \texttt{max-id}_{\text{rcvd}}))$ **do**
  if $(\texttt{max-id}_{\text{rcvd}} > \texttt{max-id})$ AND $(\mathsf{dist}_{\mathsf{cc}}(\theta, \theta_{\text{rcvd}}) \leq r$ AND $\texttt{dir}_{\text{rcvd}} = \texttt{c})$ OR $(\mathsf{dist}_{\mathsf{c}}(\theta, \theta_{\text{rcvd}}) \leq r$ AND $\texttt{dir}_{\text{rcvd}} = \texttt{cc})$ **then**
    $\texttt{new-dir} := \texttt{dir}_{\text{rcvd}}$
    $\texttt{new-id} := \texttt{max-id}_{\text{rcvd}}$
return $(\texttt{new-dir}, \texttt{new-id})$

function ctl$(\theta_{\text{smpld}}, w, y)$
$d_{\text{tmp}} := r$
for each non-`null` message $(\theta_{\text{rcvd}}, (\texttt{dir}_{\text{rcvd}}, \texttt{max-id}_{\text{rcvd}}))$ **do**
  if $(\texttt{dir} = \texttt{cc})$ AND $(\mathsf{dist}_{\mathsf{cc}}(\theta_{\text{smpld}}, \theta_{\text{rcvd}}) < d_{\text{tmp}})$ **then**
    $d_{\text{tmp}} := \mathsf{dist}_{\mathsf{cc}}(\theta_{\text{smpld}}, \theta_{\text{rcvd}})$ and $u_{\text{tmp}} := k_{\text{prop}} d_{\text{tmp}}$ $(k_{\text{prop}} \in (0, \frac{1}{2}))$
  if $(\texttt{dir} = \texttt{c})$ AND $(\mathsf{dist}_{\mathsf{c}}(\theta_{\text{smpld}}, \theta_{\text{rcvd}}) < d_{\text{tmp}})$ **then**
    $d_{\text{tmp}} := \mathsf{dist}_{\mathsf{c}}(\theta_{\text{smpld}}, \theta_{\text{rcvd}})$ and $u_{\text{tmp}} := -k_{\text{prop}} d_{\text{tmp}}$
return u_{tmp}

Evolution of a robotic network – formal definition

Evolution of $(\mathcal{S}, \mathcal{CC})$ from $x_0^{[i]} \in X_0{}^{[i]}$ 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 $w^{[i]} : \mathbb{Z}_{\geq 0} \to W^{[i]}$, $i \in I$

$$\dot{x}^{[i]}(t) = f\Big(x^{[i]}(t), \text{ctl}^{[i]}\big(t, x^{[i]}(\lfloor t \rfloor), w^{[i]}(\lfloor t \rfloor), y^{[i]}(\lfloor t \rfloor)\big)\Big),$$

where $\lfloor t \rfloor = \max\{\ell \in \mathbb{Z}_{\geq 0} \mid \ell < t\}$, and

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

with $x^{[i]}(0) = x_0^{[i]}$, and $w^{[i]}(-1) = w_0^{[i]}$, $i \in I$

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

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

Processor state set and alphabet quantization

We allow the processor state set and the communication alphabet to contain an infinite number of symbols – equivalently, we neglect inaccuracies due to quantization

Convenient to allow messages to contain real numbers because, in many control and communication laws, the robots exchange their states, including both their processor and their physical states

For such laws, communication alphabet $\mathbb{A} = (X \times W) \cup \{\texttt{null}\}$; and message generation function $\text{msg}_{\text{std}}(x, w, j) = (x, w)$ is **standard message-generation function**

Robotic networks with relative sensing

Model assumes ability of each robot to know its own absolute position

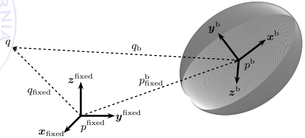
Alternative setting: robots do not communicate amongst themselves, but instead

- detect and measure each other's relative position through appropriate sensors
- perform measurements of the environment without having a priory knowledge

Robots do not have the ability to perform measurements expressed in a common reference frame

Kinematic motions

No common reference frame



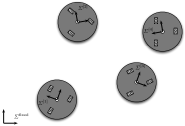
On Euclidean space $\mathbb{R}^3$, for a point q and a vector v,

$$q_{\text{fixed}} = R^{\text{b}}_{\text{fixed}} q_{\text{b}} + p^{\text{b}}_{\text{fixed}}$$
$$v_{\text{fixed}} = R^{\text{b}}_{\text{fixed}} v_{\text{b}}$$

Physical components

n robots moving in $Q \subset \mathbb{R}^d$, $d \in \{2, 3\}$

Reference frame $\Sigma^{[i]}$ attached to each robot, for $i \in \{1, \dots, n\}$



Motion: Constant $R^{[i]}_{\text{fixed}}$ and own control $u^{[i]}_i$ ($\in U$ compact)

$$\dot{p}^{[i]}_{\text{fixed}}(t) = R^{[i]}_{\text{fixed}} u^{[i]}_i$$

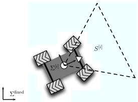
Sensing: relative position of any object inside "sensor footprint"

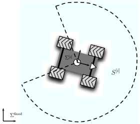
Relative sensing

Sensing other robots' positions: *sensing alphabet* $\mathbb{A}_{\text{rbt}}$ (containing `null`) and *sensing function* rbt-sns : $\mathbb{R}^d \to \mathbb{A}_{\text{rbt}}$. Robot i acquires

$$\text{rbt-sns}(p^{[j]}_i) \in \mathbb{A}_{\text{rbt}}, \quad j \in \{1, \dots, n\} \setminus \{i\}$$

Sensing the environment: *environment sensing alphabet* $\mathbb{A}_{\text{env}}$ (containing `null`) and *environment sensing function* env-sns : $\mathbb{P}(\mathbb{R}^d) \to \mathbb{A}_{\text{env}}$. Robot i acquires env-sns$(Q_i) \in \mathbb{A}_{\text{env}}$





No information about robots/boundaries outside *sensor footprint* $S^{[i]}_i$

A couple of examples

Disk sensor and corresponding relative-sensing network: $\mathcal{S}^{\text{rs}}_{\text{disk}}$

- *Disk sensor* has sensor footprint $\overline{B}(\mathbf{0}_d, r)$
- $\mathbb{A}_{\text{rbt}} = \mathbb{R}^d \cup \{\texttt{null}\}$ and

$$\text{rbt-sns}(p_i^{[j]}) = \begin{cases} p_i^{[j]} & \text{if robot } j \text{ in } \overline{B}(\mathbf{0}_d, r) \\ \texttt{null} & \text{otherwise} \end{cases}$$

- $\mathbb{A}_{\text{env}} = \mathbb{P}(\mathbb{R}^d)$ and $\text{env-sns}(Q_i) = Q_i \cap \overline{B}(\mathbf{0}_d, r)$

Range-limited visibility sensor and corresponding relative-sensing network: $\mathcal{S}^{\text{rs}}_{\text{vis-disk}}$

- *Range-limited visibility sensor* has sensor footprint $\overline{B}(\mathbf{0}_d, r)$, performs measurements of objects with unobstructed line of sight
- $\mathbb{A}_{\text{rbt}} = \mathbb{R}^d \cup \{\texttt{null}\}$ and

$$\text{rbt-sns}(p_i^{[j]}) = \begin{cases} p_i^{[j]} & \text{if robot } j \text{ in } \mathsf{Vi}_{\text{disk}}(\mathbf{0}_2; Q_i) \\ \texttt{null} & \text{otherwise} \end{cases}$$

- $\mathbb{A}_{\text{env}} = \mathbb{P}(\mathbb{R}^d)$ and $\text{env-sns}(Q_i) = \mathsf{Vi}_{\text{disk}}(\mathbf{0}_2; Q_i)$

Relative-sensing control law

Relative-sensing control law $\mathcal{RSC}$ for $\mathcal{S}^{\text{rs}}$ consists of

1. W, called the *processor state set*, with corresponding set of *allowable initial values* $W_0 \subseteq W$;
2. $\text{stf} : W \times \mathbb{A}^n_{\text{rbt}} \times \mathbb{A}_{\text{env}} \to W$, called the *(processor) state-transition function*; and
3. $\text{ctl} : W \times \mathbb{A}^n_{\text{rbt}} \times \mathbb{A}_{\text{env}} \to U$, called the *(motion) control function*.

Equivalence can be established between **invariant** control and communication laws and relative-sensing control laws – equivalent evolutions

Coordination tasks

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?

A **coordination task** for a robotic network $\mathcal{S}$ is a map $\mathcal{T} : X^n \times \mathcal{W}^n \to \{\texttt{true}, \texttt{false}\}$

Logic-based: agree, synchronize, form a team, elect a leader
Motion: deploy, gather, flock, reach pattern
Sensor-based: search, estimate, identify, track, map

A control and communication law $\mathcal{CC}$ **achieves** the task $\mathcal{T}$ if, for all initial conditions $x_0^{[i]} \in X_0^{[i]}$ and $w_0^{[i]} \in W_0^{[i]}$, $i \in I$, the corresponding network evolution $t \mapsto (x(t), w(t))$ has the property that there exists $T \in \mathbb{R}_{>0}$ such that $\mathcal{T}(x(t), w(t)) = \texttt{true}$ for all $t \geq T$

Task definitions via 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

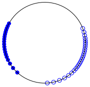
More general tasks based on more expressive predicates on trajectories can be defined through temporal and propositional logic, e.g.

periodically visiting a desired set of configurations

Direction agreement and equidistance tasks

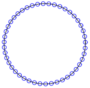
Direction agreement task $\mathcal{T}_{\mathtt{dir}} : (\mathbb{S}^1)^n \times W^n \to \{\mathtt{true}, \mathtt{false}\}$

$$\mathcal{T}_{\mathtt{dir}}(\theta, w) = \begin{cases} \mathtt{true}, & \text{if } \mathtt{dir}^{[1]} = \cdots = \mathtt{dir}^{[n]} \\ \mathtt{false}, & \text{otherwise} \end{cases}$$



For $\varepsilon > 0$, **equidistance** task $\mathcal{T}_{\varepsilon\text{-eqdstnc}} : (\mathbb{S}^1)^n \to \{\mathtt{true}, \mathtt{false}\}$ is $\mathtt{true}$ iff

$$\Big| \min_{j \neq i} \mathsf{dist}_{\mathsf{c}}(\theta^{[i]}, \theta^{[j]}) - \min_{j \neq i} \mathsf{dist}_{\mathsf{cc}}(\theta^{[i]}, \theta^{[j]}) \Big| < \varepsilon, \quad \text{for all } i \in I$$



Complexity notions for control and communication laws

For network $\mathcal{S}$, task $\mathcal{T}$, and algorithm $\mathcal{CC}$, define **costs/complexity**
control effort, communication packets, computational cost

Time complexity: maximum number of communication rounds required to achieve $\mathcal{T}$

Space complexity: maximum number of basic memory units required by a robot processor among all robots

Communication complexity: maximum number of basic messages transmitted over entire network

(among all allowable initial physical and processor states until termination)

basic memory unit/message contain $\log(n)$ bits

More formally: time complexity

The **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

$$\mathsf{TC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) = \inf \{\ell \mid \mathcal{T}(x(t_k), w(t_k)) = \mathtt{true}, \text{ for all } k \geq \ell\},$$

where $t \mapsto (x(t), w(t))$ is the evolution of $(\mathcal{S}, \mathcal{CC})$ from the initial condition (x_0, w_0)

The **time complexity to achieve $\mathcal{T}$ with $\mathcal{CC}$** is

$$\mathsf{TC}(\mathcal{T}, \mathcal{CC}) = \sup \Big\{ \mathsf{TC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) \mid (x_0, w_0) \in \prod_{i \in I} X_0^{[i]} \times \prod_{i \in I} W_0^{[i]} \Big\}.$$

The **time complexity of $\mathcal{T}$** is

$$\mathsf{TC}(\mathcal{T}) = \inf\{\mathsf{TC}(\mathcal{T}, \mathcal{CC}) \mid \mathcal{CC} \text{ compatible with } \mathcal{T}\}$$

More formally: communication complexity

The set of all non-null messages generated during one communication round from network state (x, w)

$$\mathcal{M}(x, w) = \{(i, j) \in E_{\mathrm{cmm}}(x) \mid \mathrm{msg}^{[i]}(x^{[i]}, w^{[i]}, j) \neq \mathtt{null}\}.$$

The **mean communication complexity** and the **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,

$$\mathsf{MCC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) = \frac{|\mathbb{A}|_{\mathrm{basic}}}{\lambda} \sum_{\ell=0}^{\lambda-1} |\mathcal{M}(x(\ell), w(\ell))|,$$

$$\mathsf{TCC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) = |\mathbb{A}|_{\mathrm{basic}} \sum_{\ell=0}^{\lambda-1} |\mathcal{M}(x(\ell), w(\ell))|,$$

where $|\mathbb{A}|_{\mathrm{basic}}$ is number of basic messages required to represent elements of $\mathbb{A}$ and $\lambda = \mathsf{TC}(\mathcal{CC}, \mathcal{T}, x_0, w_0)$

Variations and extensions

Asymptotic results

Complexities in $O(f(n))$, $\Omega(f(n))$, or $\Theta(f(n))$ as $n \to \infty$

1. **Infinite-horizon mean communication complexity:** mean communication complexity to maintain true the task for all times

$$\text{IH-MCC}(\mathcal{CC}, x_0, w_0) = \lim_{\lambda\to+\infty} \frac{|\mathbb{A}|_{\text{basic}}}{\lambda} \sum_{\ell=0}^{\lambda} |\mathcal{M}(x(\ell), w(\ell))|$$

2. **Communication complexity in omnidirectional networks:** All neighbors of a to receive the signal it transmits. Makes sense to count the number of transmissions, i.e., a unit cost per node, rather than a unit cost per edge of the network
3. **Energy complexity**
4. **Expected** notions, rather than **worst-case** notions

Time complexity of agree-and-pursue law

Let $r : \mathbb{N} \to]0, 2\pi[$ be a monotone non-increasing function of number of agents n – modeling wireless communication congestion

Theorem

In the limit as $n \to +\infty$ and $\varepsilon \to 0^+$, the network $\mathcal{S}_{\text{circle}}$, the law $\mathcal{CC}_{\text{AGREE \& PURSUE}}$, and the tasks $\mathcal{T}_{\texttt{dir}}$ and $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$ together satisfy:

1. $\text{TC}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Theta(r(n)^{-1})$;
2. *if $\delta(n) = nr(n) - 2\pi$ 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 lower bounded by a negative constant, then $\mathcal{CC}_{\text{AGREE \& PURSUE}}$ does not achieve $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$ in general.

Proof sketch - O bound for $\mathcal{T}_{\texttt{dir}}$

Claim: $\text{TC}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \le 2\pi/(k_{\text{prop}} r(n))$

By contradiction, assume there exists initial condition such that execution has time complexity $> 2\pi/(k_{\text{prop}} r(n))$
Without loss of generality, $\texttt{dir}^{[n]}(0) = \texttt{c}$. For $\ell \le 2\pi/(k_{\text{prop}} r(n))$, let

$$k(\ell) = \text{argmin}\{\text{dist}_{\text{cc}}(\theta^{[n]}(0), \theta^{[i]}(\ell)) \mid \texttt{dir}^{[i]}(\ell) = \texttt{cc},\ i \in I\}$$

Agent $k(\ell)$ is agent moving counterclockwise that has smallest counterclockwise distance from the initial position of agent n

Recall that according to $\mathcal{CC}_{\text{AGREE \& PURSUE}}$

- messages with $\texttt{dir} = \texttt{cc}$ can only travel counterclockwise
- messages with $\texttt{dir} = \texttt{c}$ can only travel clockwise

Therefore, 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

Proof sketch - O bound for $\mathcal{T}_{\texttt{dir}}$

How fast the message from agent n travels clockwise?

For $\ell \le 2\pi/(k_{\text{prop}} r(n))$, define

$$j(\ell) = \text{argmax}\{\text{dist}_{\text{c}}(\theta^{[n]}(0), \theta^{[i]}(\ell)) \mid \texttt{max-id}^{[i]}(\ell) = n,\ i \in I\}$$

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 $\ell \le 2\pi/(k_{\text{prop}} r(n))$, we claim

$$\text{dist}_{\text{c}}(\theta^{[j(\ell)]}(\ell), \theta^{[j(\ell+1)]}(\ell+1)) \ge k_{\text{prop}} r(n)$$

Proof sketch - O bound for $\mathcal{T}_{\mathtt{dir}}$

$\mathsf{TC}(\mathcal{T}_{\mathtt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \leq 2\pi/(k_{\text{prop}} r(n))$

This happens because either (1) there is no agent clockwise-ahead of $\theta^{[j(\ell)]}(\ell)$ within clockwise distance r 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 from $\theta^{[j(\ell)]}(\ell)$. Then,

$$\begin{aligned}
&\mathsf{dist_c}(\theta^{[j(\ell)]}(\ell), \theta^{[j(\ell+1)]}(\ell+1)) \\
&\quad = \mathsf{dist_c}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell+1)) \\
&\quad = \mathsf{dist_c}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell)) + \mathsf{dist_c}(\theta^{[m]}(\ell), \theta^{[m]}(\ell+1)) \\
&\quad \geq \mathsf{dist_c}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell)) + k_{\text{prop}}\big(r - \mathsf{dist_c}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell))\big) \\
&\quad = k_{\text{prop}} r + (1 - k_{\text{prop}})\, \mathsf{dist_c}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell)) \ \geq k_{\text{prop}} r
\end{aligned}$$

Therefore, after $2\pi/(k_{\text{prop}} r(n))$ 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 $k(\ell)$ **Contradiction**

Proof sketch - O bound for $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$

Assume $\mathcal{T}_{\mathtt{dir}}$ has been achieved and all agents are moving clockwise

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,

$$H(\ell) = \{x \in \mathbb{S}^1 \mid \min_{i \in I} \mathsf{dist_c}(x, \theta^{[i]}(\ell)) + \min_{j \in I} \mathsf{dist_{cc}}(x, \theta^{[j]}(\ell)) > r\}.$$

$H(\ell)$ does not contain any point between two agents separated by a distance less than r, and each connected component has length at least r

Let $n_H(\ell)$ be number of connected components of $H(\ell)$,

- if $H(\ell)$ is empty, then $n_H(\ell) = 0$
- $n_H(\ell) \leq n$
- if $n_H(\ell) > 0$, then $t \mapsto n_H(\ell + t)$ is non-increasing

Proof sketch- O bound for $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$

Number of connected components is strictly decreasing

Claim: if $n_H(\ell) > 0$, then $\exists t > \ell$ such that $n_H(t) < n_H(\ell)$

By contradiction, assume $n_H(\ell) = n_H(t)$ for all $t > \ell$. Without loss of generality, let $\{1, \dots, m\}$ be a set of agents with the properties

- $\mathsf{dist_{cc}}\left(\theta^{[i]}(\ell), \theta^{[i+1]}(\ell)\right) \leq r$, for $i \in \{1, \dots, m\}$
- $\theta^{[1]}(\ell)$ and $\theta^{[m]}(\ell)$ belong to the boundary of $H(\ell)$
- there is no other set with the same properties and more agents

One can show that, for $\tau \geq \ell$ and $i \in \{2, \dots, m\}$

$$\theta^{[1]}(\tau + 1) = \theta^{[1]}(\tau) - k_{\text{prop}} r$$
$$\theta^{[i]}(\tau + 1) = \theta^{[i]}(\tau) - k_{\text{prop}}\, \mathsf{dist_c}(\theta^{[i]}(\tau), \theta^{[i-1]}(\tau))$$

Tridiagonal and circulant linear dynamical systems

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

Linear dynamical systems

$$y(\ell + 1) = A y(\ell), \quad \ell \in \mathbb{Z}_{\geq 0}$$

Rates of convergence to set of equilibria can be characterized – carefully look at eigenvalues. Statements of the form

if $a \geq 0$, $c \geq 0$, $b > 0$, and $a + b + c = 1$, then $\lim_{\ell \to +\infty} y(\ell) = y_{\text{ave}} \mathbf{1}$, where $y_{\text{ave}} = \frac{1}{n} \mathbf{1}^T y_0$, and maximum time required (over all initial conditions $y_0 \in \mathbb{R}^n$) for $\|y(\ell) - y_{\text{ave}} \mathbf{1}\|_2 \leq \varepsilon \|y_0 - y_{\text{ave}} \mathbf{1}\|_2$ is $\Theta(n^2 \log \varepsilon^{-1})$

Proof sketch- O bound for $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$

Contradiction argument

For $d(\tau) = \big(\mathsf{dist}_{cc}(\theta^{[1]}(\tau), \theta^{[2]}(\tau)), \dots, \mathsf{dist}_{cc}(\theta^{[m-1]}(\tau), \theta^{[m]}(\tau))\big)$,

$$d(\tau+1) = \mathsf{Trid}_{m-1}(k_{\text{prop}}, 1 - k_{\text{prop}}, 0)\, d(\tau) + r[k_{\text{prop}},\ 0,\ \cdots,\ 0]^T$$

Unique equilibrium point is $r(1,\dots,1)$. For $\eta_1 \in]0,1[$, $\tau \mapsto d(\tau)$ reaches ball of radius η_1 centered at equilibrium in $O(m \log m + \log \eta_1^{-1})$

This implies that $\tau \mapsto \sum_{i=1}^m d_i(\tau)$ is larger than $(m-1)(r-\eta_1)$ in time $O(m \log m + \log \eta_1^{-1}) = O(n \log n + \log \eta_1^{-1})$. After this time,

$$\begin{aligned} 2\pi &\geq n_H(\ell) r + \sum_{j=1}^{n_H(\ell)} (r - \eta_1)(m_j - 1) \\ &= n_H(\ell) r + (n - n_H(\ell))(r - \eta_1) = n_H(\ell)\eta_1 + n(r - \eta_1) \end{aligned}$$

Proof sketch- O bound for $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$

Take $\eta_1 = (nr - 2\pi)n^{-1} = \delta(n)n^{-1}$, and the contradiction follows from

$$\begin{aligned} 2\pi &\geq n_H(\ell)\eta_1 + nr - n\eta_1 \\ &= n_H(\ell)\eta_1 + nr + 2\pi - nr = n_H(\ell)\eta_1 + 2\pi \end{aligned}$$

Therefore $n_H(\ell)$ decreases by one in time $O(n \log n)$

Iterating this argument n times, in time $O(n^2 \log n)$ the set H becomes empty. At that time, resulting network obeys

$$d(\tau+1) = \mathsf{Circ}_n(k_{\text{prop}}, 1 - k_{\text{prop}}, 0)\, d(\tau)$$

In time $O(n^2 \log \varepsilon^{-1})$, the error 2-norm satisfies the contraction inequality $\|d(\tau) - d_*\|_2 \leq \varepsilon \|d(0) - d_*\|_2$, for $d_* = \frac{2\pi}{n}\mathbf{1}$

The conversion of this inequality into an appropriate inequality on ∞-norms yields the result

Communication complexity of agree-and-pursue law

Theorem

Total communication complexity of agree-and-pursue law In the limit as $n \to +\infty$ and $\varepsilon \to 0^+$, the network $\mathcal{S}_{\text{circle}}$, the law $\mathcal{CC}_{\text{AGREE \& PURSUE}}$, and the tasks $\mathcal{T}_{\texttt{dir}}$ and $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$ together satisfy:

1. *if $\delta(n) \geq \pi(1/k_{\text{prop}} - 2)$ as $n \to +\infty$, then*

   $$\mathsf{TCC}_{\mathsf{unidir}}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Theta(n^2 r(n)^{-1}),$$

   otherwise if $\delta(n) \leq \pi(1/k_{\text{prop}} - 2)$ as $n \to +\infty$, then

   $$\begin{aligned} &\mathsf{TCC}_{\mathsf{unidir}}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Omega(n^3 + nr(n)^{-1}), \\ &\mathsf{TCC}_{\mathsf{unidir}}(\mathcal{T}_{\texttt{dir}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in O(n^2 r(n)^{-1}); \end{aligned}$$

2. *if $\delta(n)$ is lower bounded by a positive constant as $n \to +\infty$, then*

   $$\begin{aligned} &\mathsf{TCC}_{\mathsf{unidir}}(\mathcal{T}_{\varepsilon\text{-eqdstnc}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Omega(n^3 \delta(n) \log(n\varepsilon)^{-1}), \\ &\mathsf{TCC}_{\mathsf{unidir}}(\mathcal{T}_{\varepsilon\text{-eqdstnc}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in O(n^4 \log(n\varepsilon^{-1})). \end{aligned}$$

Comparison with leader election

- **Leader election task** is different from, but closely related to, $\mathcal{T}_{\texttt{dir}}$
- **LCR algorithm** operates on a static ring network, and achieves leader election with time and total communication complexity, respectively, $\Theta(n)$ and $\Theta(n^2)$
- **Agree-and-pursue** law operates on robotic network with $r(n)$-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 $1/n$, then identical time complexity and the LCR algorithm has better communication complexity

Computations on a possibly disconnected, dynamic network are more complex than on a static ring topology

Summary and conclusions

Cooperative robotic network model

- proximity graphs
- control and communication law, task, execution
- time, space, and communication complexity
- agree and pursue

Complexity analysis is **challenging** even in 1 dimension! Blend of math

Plenty of **open problems**

- Quantization, asynchronism, delays
- What is best algorithm to achieve a task?
- What tools are useful to characterize complexity?
- How does combination of algorithms affect individual complexities?

References

Proximity graphs:

- J. W. Jaromczyk and G. T. Toussaint. Relative neighborhood graphs and their relatives. *Proceedings of the IEEE*, 80(9):1502--1517, 1992
- J. Cortés, S. Martínez, and F. Bullo. Spatially-distributed coverage optimization and control with limited-range interactions. *ESAIM. Control, Optimisation & Calculus of Variations*, 11:691--719, 2005

Robotic network model:

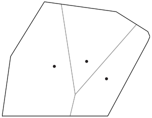
- S. Martínez, F. Bullo, J. Cortés, and E. Frazzoli. On synchronous robotic networks { Part I: Models, tasks and complexity. *IEEE Transactions on Automatic Control*, 52(12):2199--2213, 2007

Voronoi partitions

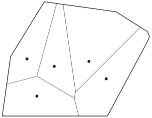
Let $(p_1, \dots, p_n) \in Q^n$ denote the positions of n points

The **Voronoi partition** $\mathcal{V}(P) = \{V_1, \dots, V_n\}$ generated by $(p_1, \dots, p_n)$

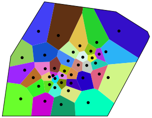
$$\begin{aligned} V_i &= \{q \in Q \mid \; \|q - p_i\| \leq \|q - p_j\|, \; \forall j \neq i\} \\ &= Q \cap_j \mathcal{HP}(p_i, p_j) \qquad \text{where } \mathcal{HP}(p_i, p_j) \text{ is half plane } (p_i, p_j) \end{aligned}$$



3 generators



5 generators



50 generators

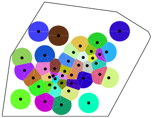
Return

r-limited Voronoi partition

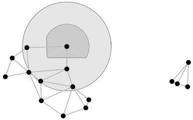
Let $(p_1, \dots, p_n) \in Q^n$ denote the positions of n points

The **r-limited Voronoi partition** $\mathcal{V}_r(P) = \{V_{1,r}, \dots, V_{n,r}\}$ generated by $(p_1, \dots, p_n)$

$$V_{i,r}(\mathcal{P}) = V_i(\mathcal{P}) \cap \overline{B}(p_i, r)$$



Return

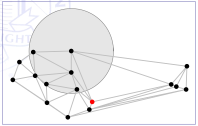


$\mathcal{G}_{\mathrm{LD}}(r)$ is **spatially distributed** over $\mathcal{G}_{\mathrm{disk}}(r)$

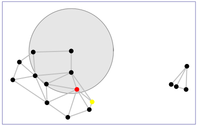
Return

$\mathcal{G}_{\mathrm{D}}$ and $\mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r)$ computation

$\mathcal{G}_{\mathbf{D}}$



$\mathcal{G}_{\mathbf{D}} \cap \mathcal{G}_{\mathbf{disk}}(r)$



$\mathcal{G}_{\mathrm{D}}$ and $\mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r)$ are **not** spatially distributed over $\mathcal{G}_{\mathrm{disk}}(r)$

Return