Motion Coordination for Multi-Agent Networks

Mechanical Engineering
University of California at Berkeley, April 15, 2005

Francesco Bullo
Mechanical and Environmental Engineering
Center for Control Engineering and Computation
University of California at Santa Barbara
Santa Barbara, CA 93106
http://motion.mee.ucsb.edu

Thanks to: Sonia Martínez, Anurag Ganguli
Jorge Cortés (UCSC), Emilio Frazzoli (UCLA)

Multi-agent networks

What kind of systems?
Groups of systems with control, sensing, communication and computing

Individual members in the group can

- sense its immediate environment
- communicate with others
- process the information gathered
- take a local action in response

Swarms
"A large number of insects or other small organisms, especially when in motion."

Dynamical and biological systems examples

- Biological populations


Wildebeest herd in the Serengeti


Geese flying in formation


Atlantis aquarium, CDC Conference 2004

- Distributed information systems, large-scale complex systems
  (intelligent buildings, stock market, self-managed air-traffic systems)

- Sensor networks
  (embedded sensor systems w/ communication networks)

Applications of robotic sensor networks

- distributed environmental monitoring systems — e.g., portable chemical and biological sensor arrays detecting toxic pollutants
- high-stress, rapid deployment systems — e.g., disaster recovery networks
- autonomous sampling networks in biology — e.g., monitoring of species in risk, validation of (micro-)climate and oceanographic models
- health monitoring of civil infrastructure — e.g., bridges, buildings, pipelines


UIUC/UCSB Multi Rover Lab


Sandia National Labs

Broad challenge

What useful engineering tasks can be performed with small, inexpensive, limited-communication individual vehicles/sensors?

Problem lack of understanding of how to assemble and coordinate individual devices into a coherent whole

Need integration of control, communication, sensing, computing resources

Development of systematic methodologies,
to analyze and design large-scale multi-agent networks

Broad challenge

What useful engineering tasks can be performed with small, inexpensive, limited-communication individual vehicles/sensors?

Problem lack of understanding of how to assemble and coordinate individual devices into a coherent whole

Need integration of control, communication, sensing, computing resources

Development of systematic methodologies,
to analyze and design large-scale multi-agent networks

Scientific interest
Military interest
NAE report "Embedded, Everywhere" '01
Commercial interest

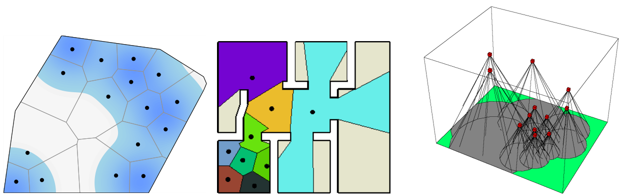
Acknowledgment

- NSF/ONR Sensors and Sensor Networks
- AFOSR MURI on Peer-to-Peer Networked Vehicles
- ARO MURI on Swarms

Sample problems for robotic sensor networks

(i) sensing tasks
detection, localization, visibility, exploration, search, plume tracing

(ii) motion tasks
deployment, rendezvous, flocking, self-assembly



Objective: Modeling, analysis and design of
motion coordination algorithms for multi-vehicle networks

Fundamental challenges in motion coordination

(i) multiple tasks are relevant
arbitrary static/dynamic pattern formation

(ii) adaptive and distributed

centralized computation for known and static environment
versus
anonymous, distributed, adaptive, asynchronous setting

(iii) rigorous understanding of meaningful subproblems

increasingly important with increasing complexity of network, task, environment, constraints, guarantees

convergence, adaptation, limited information

Outline: an emerging discipline

(i) models:
multi-agent network with (1) motion and (2) communication

(ii) algorithms:
deployment, rendezvous, vehicle routing

(iii) tools

Deployment: performance metrics and algorithms

DESIGN of performance metric

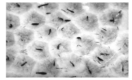
(i) how to cover a region with n minimum radius overlapping disks?

(ii) how to design a minimum-distorsion (fixed-rate) vector quantizer? (Lloyd '57)

(iii) where to place mailboxes in a city / cache servers on the internet?

ANALYSIS of cooperative distributed behaviors

(iv) how do animals share territory?
what if every fish in a swarm goes
toward center of own dominance region?



Barlow, Hexagonal territories. Anim. Behav. '74

(v) what if each vehicle moves toward center of mass of own Voronoi cell?

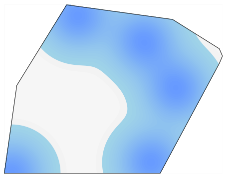
(vi) what if each vehicle moves away from closest vehicle?

Scenario 1: min-expected-distance deployment

Objective: Given sensors/nodes/robots/sites $(p_1, \dots, p_n)$ moving in environment Q
achieve optimal coverage defined according to

Scenario 1 —expected value performance measure
(unlimited-range sensor or communication radius)

given distribution density function ϕ



minimize $\mathcal{H}_{\mathsf{C}}(p_1, \dots, p_n) = E_\phi \left[\min_i \|q - p_i\|^2 \right]$

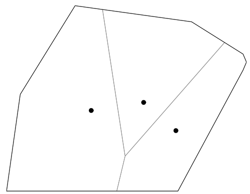
Scenario 1: Voronoi partition

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

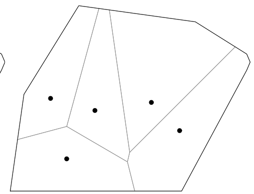
Voronoi partition $\mathcal{V}(P) = \{V_1, \dots, V_n\}$ generated by $(p_1, \dots, p_n)$

$$V_i = \{q \in Q \mid \; \|q - p_i\| \leq \|q - p_j\|, \; \forall j \neq i\}$$

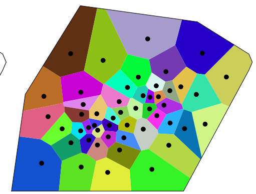
Voronoi neighbors are points with adjacent regions



3 generators



5 generators



50 generators

Scenario 1: coverage algorithm

Name: Coverage behavior
Goal: distributed optimal agent deployment
Requires: (i) own Voronoi cell computation
(ii) centroid computation

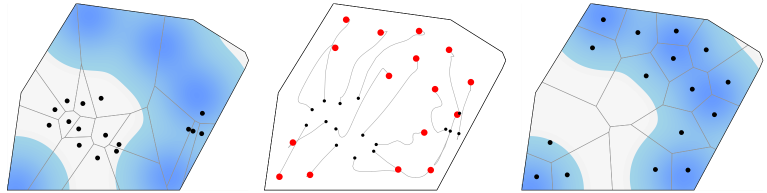
For all i, agent i synchronously performs:

1: determine own Voronoi cell V_i — distributed computation
2: determine centroid C_{V_i} of V_i
3: move towards centroid (e.g. $u_i = \mathrm{sat}(C_{V_i} - p_i)$)

$$\frac{\partial \mathcal{H}_{\mathsf{C}}}{\partial p_i} = M_i(p_i - C_{V_i})$$

Scenario 1: simulation

run: 16 agents, density ϕ is sum of 4 Gaussians, 1st order dynamics



initial configuration — gradient descent — final configuration

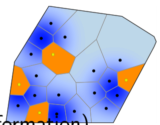
Scenario 1: analysis

Convergence: Gradient of $\mathcal{H}_{\mathsf{C}}$+ LaSalle Invariance Principle

Distributed: required information is
location of Voronoi neighbors

Adaptive: changing ϕ, changing number of agents

Asynchronous implementation:



wake up
(i) determine local Voronoi diagram (w/ outdated information)
(ii) determine centroid of own Voronoi region
(iii) take a step in that direction
go to sleep

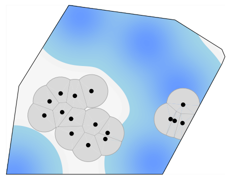
IEEE Tran. Rob. Automation '04

Scenario 2: max-area deployment

Objective: Given sensors/nodes/robots/sites $(p_1, \dots, p_n)$ moving in environment Q
achieve optimal coverage defined according to

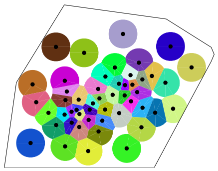
Scenario 2 —area (with communication radius r, and sensing radius $r/2$)

given distribution density function ϕ



maximize $\quad \mathrm{area}_\phi(\cup_{i=1}^n B_{\frac{r}{2}}(p_i)) = \int_Q \left(\max_i 1_{B_{\frac{r}{2}}(p_i)}(q) \right) \phi(q) dq$

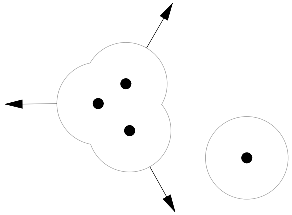
Scenario 2: partition of covered area



Partition of $\cup_i B_{r/2}(p_i)$:
$$\{V_1 \cap B_{r/2}(p_1), \dots, V_n \cap B_{r/2}(p_n)\}.$$

"Limited Voronoi" neighbors
those with adjacent cells

For constant density $\phi = 1$,



$$\int_{\mathrm{arc}(r)} n_{B_{\frac{r}{2}}(p)}\, \phi$$

Scenario 2: coverage algorithm

Name: Coverage behavior
Goal: distributed optimal agent deployment
Requires: (i) own cell computation
(ii) weighted normal computation

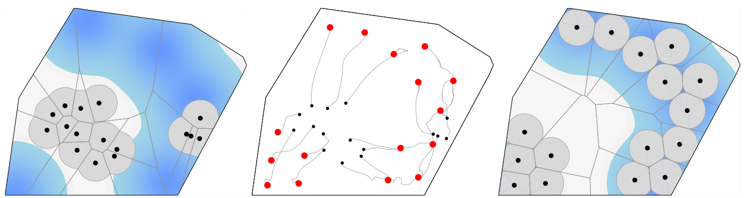
For all i, agent i synchronously performs:

1: determines own cell $V_i \cap B_{\frac{r}{2}}(p_i)$
2: determines weighted normal $\int_{\mathrm{arc}(r)} n_{B_{\frac{r}{2}}(p)}\, \phi$
3: moves in the direction of weighted normal

Caveat: convergence only to local maximum of $\mathrm{area}_\phi(\cup_{i=1}^n B_{\frac{r}{2}}(p_i))$

Scenario 2: simulation

run: 20 agents, density ϕ is sum of 4 Gaussians, 1st order dynamics



initial configuration | gradient descent | final configuration

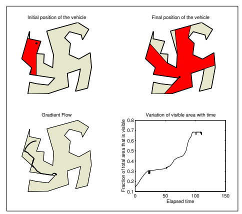
analysis results as for Scenario 1

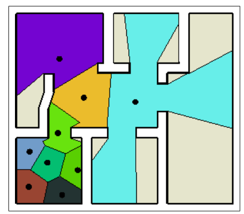
Scenario 3: max-visibility deployment

Objective: Given agents $(p_1, \dots, p_n)$ moving in **nonconvex** environment Q

Scenario 3 —maximize area within line-of-sight

maximize $\quad \mathcal{H}_{\text{visibility}}(p_1, \dots, p_n) = \mathrm{Area}(\cup_i V_i)$



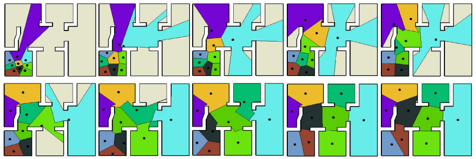


Scenario 3: algorithm and simulations

Name:	Move toward the furthest vertex
Goal:	Maximize area visible to network in nonconvex polygon
Requires:	(i) own visibility cell computation
	(ii) spatially distributed over $G_{\text{double vis}}$

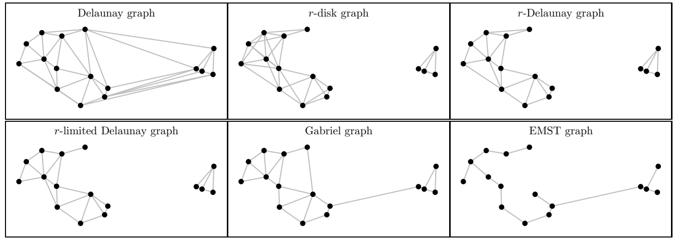
For all i, agent i synchronously performs:

1: determine own visibility polygon $V_i \subset Q$
2: compute $W_i \subset V_i$ as points for which p_i is the closest visible agent
3: compute $D_i \subset W_i$ as the connected component of W_i containing p_i
4: move towards the furthest vertex in D_i



Tools 1: proximity graphs

Proximity graphs: graphs $\mathcal{G} = (V, \mathcal{E})$ over $V = \{p_1, \dots, p_n\} \subset \mathbb{R}^d$ where edges depend on positions. Examples:



Some relationships:

(i) $\mathcal{G}_{\text{EMST}} \subset \mathcal{G}_{\text{G}} \subset \mathcal{G}_{\text{Voronoi}}$

(ii) $\mathcal{G}_{\text{disk}}(r)$ is connected if and only if $\mathcal{G}_{\text{EMST}} \subset \mathcal{G}_{\text{disk}}(r)$

Tools 1: making sense of "distributed"

"Rule is locally computable, i.e., control depends only on information about immediate neighbors"

$\mathcal{N}_{\mathcal{G}}(p_i)$ set of neighbors of p_i in proximity graph $\mathcal{G}$

An algorithm $T : (\mathbb{R}^d)^n \longrightarrow Y^n$ is **spatially distributed over** $\mathcal{G}$ if there exist $\tilde{T}_i : \mathbb{R}^d \times 2^{\mathbb{R}^d} \longrightarrow Y$, $i \in \{1, \dots, n\}$ such that

$$T_i(p_1, \dots, p_n) = \tilde{T}_i(p_i, \{p_j \in \mathbb{R}^d \mid p_j \in \mathcal{N}_{\mathcal{G}}(p_i)\})$$

Scenario 1 is spatially distributed over $\mathcal{G}_{\text{Voronoi}}$

Scenario 2 is spatially distributed over $\mathcal{G}_{\text{LVoronoi}}$ and $\mathcal{G}_{\text{disk}}(r)$
Not spatially-distributed over $\mathcal{G}_{\text{disk}}(r) \cap \mathcal{G}_{\text{Voronoi}}$

Scenario 3 is spatially distributed over $\mathcal{G}_{\text{disk}}(r)$

Outline

(i) models:
robotic networks with (1) motion control and (2) communication along $\mathcal{G}$

(ii) algorithms:
deployment

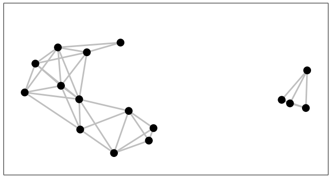
(iii) tools:
proximity graphs

(iv) rendezvous algorithms

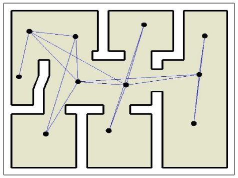
(v) vehicle routing algorithms

Scenario 4: rendezvous problems

Objective: Given agents $(p_1, \dots, p_n)$ moving in environment Q
achieve rendezvous at single point, while maintaining connectivity



- r-disk connectivity constraint

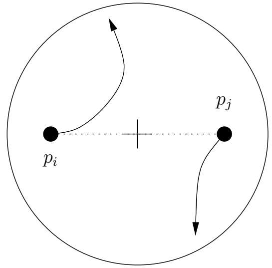


- visibility connectivity constraint

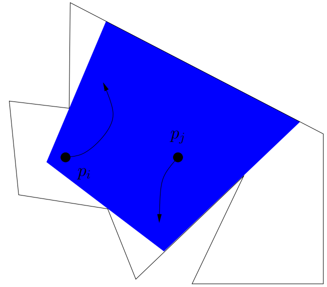
Scenario 4 —diameter minimization
(limited-range sensor or communication radius r)

minimize $\quad \mathcal{H}_{\text{diam}}(p_1, \dots, p_n) = \max\{\|p_i - p_j\| \mid i, j \in \{1, \dots, n\}\}$

Scenario 4: constraint sets for connectivity



r-disk pair-wise constraint set



visibility pair-wise constraint set

Scenario 4: circumcenter algorithms

- each agent minimizes "local version" of $\mathcal{H}_{\text{diam}}$:

$$\mathcal{H}_{\text{diam,local},i} = \max\{\|p_i - p_j\| \mid p_j \text{is neighbor of } p_i\}$$

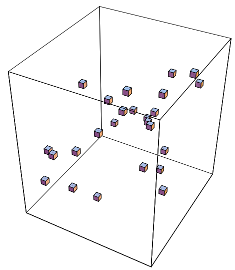
  i.e., each agent goes toward circumcenter of neighbors
- each agent maintains connectivity by moving inside constraint set,
  i.e., intersection of all neighbors' constraint sets

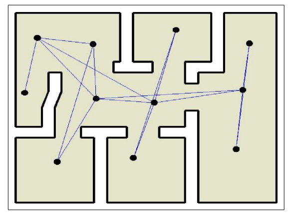
Name: Circumcenter Algorithm over $\mathcal{G}_{\text{disk}}(r)$

Agent i performs:

1: compute r-disk neighbors $\mathcal{M}_i$
2: compute connectivity constraint set $Q_i := \bigcap_{q \in \mathcal{M}_i} \overline{B}\left(\frac{p_i + q}{2}, \frac{r}{2}\right)$
3: move from p_i towards $\mathrm{CC}(\mathcal{M}_i)$ while remaining in Q_i

Scenario 4: simulations





Tools 2: network objective functions

design of aggregate network-wide cost/objective/utility functions

- objective functions to encode motion coordination objective
- objective functions as Lyapunov functions
- objective functions for gradient flows

	$\mathcal{H}_{\mathrm{C}}$	$\mathcal{H}_{\mathrm{area}}$	$\mathcal{H}_{\mathrm{diam}}$
DEFINITION	$E\left[\min d(q, p_i)\right]$	$\mathrm{area}_\phi(\cup_i B_{r/2}(p_i))$	$\max_{i,j} \lVert p_i - p_j \rVert$
SMOOTHNESS	C^1	globally Lipschitz	continuous, locally Lipschitz
CRITICAL POINTS MINIMA	Centroidal Voronoi configurations	r-limited Voronoi configurations*	common location for p_i
HEURISTIC DESCRIPTION	expected distortion	area covered	diameter connected component

Tools 3: set-valued dynamical systems

Properties of trajectories of cooperative systems:

- Non-smoothness due to switching of neighbors
- Non-determinism due to:
  - a design choice to allow different possibilities
  - failures in communication/sensing
  - different asynchronous behaviors

Solution: Analyze a set of trajectories or set-valued map

Tools 3: convergence theorems

Cooperative systems $\Longrightarrow$ Non-deterministic dynamical system

$$(p_1^+, \dots, p_n^+) \in T(p_1, \dots, p_n)$$

Arises from switching topology, asynchronism, link failures, design choices

LaSalle Invariance Theorem for set-valued maps:

(i) T is closed at $P = (p_1, \dots, p_n) \in \mathbb{R}^d$,

(ii) $V : \mathbb{R}^d \to \mathbb{R}$, continuous & decreasing along T on $\mathbb{R}^d$

(iii) Assume $\{P_m\}_{m \in \mathbb{N}}$ is bounded

Let M be the largest weakly positively invariant set contained in

$$\{P \in \mathbb{R}^d \mid \exists\, P' \in T(P) \text{ with } V(P') = V(P)\}$$

Then, $\exists\, c \in \mathbb{R}$ such that $P_m \to M \cap V^{-1}(c)$

Outline

(i) models:
robotic networks with (1) motion control and (2) communication along $\mathcal{G}$

(ii) algorithms:
deployment, rendezvous

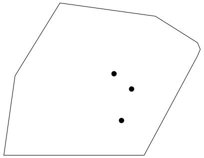
(iii) tools:
proximity graphs, objective functions, convergence & invariance

(iv) analysis of behaviors

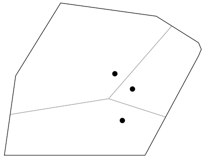
(v) vehicle routing algorithms

Tools 4: Analysis of behaviors

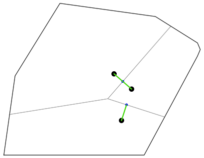
Consider n points in Q



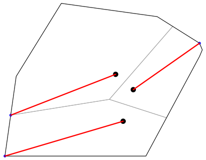
Draw Voronoi partition



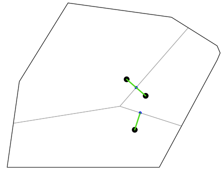
Identify closest point

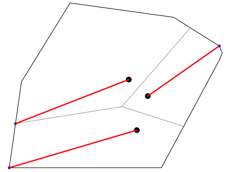


Identify furthest point

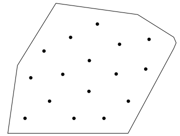


Tools 4: Analysis of behaviors, cont'd

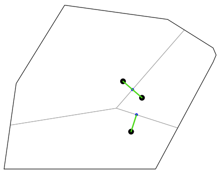


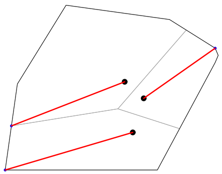


Basic behaviors
"move away from closest"
"move towards furthest"

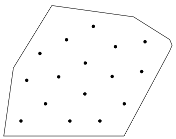


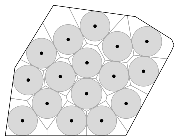
Tools 4: Analysis of behaviors, cont'd



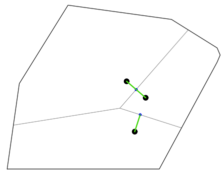


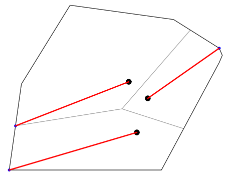
Basic behaviors
"move away from closest"
"move towards furthest"



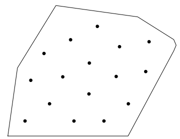


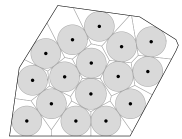
Tools 4: Analysis of behaviors, cont'd





Basic behaviors
"move away from closest"
"move towards furthest"





Questions: critical points or periodic trajectories? convergence? optimize? local minima? equidistant?

Outline

(i) models:
robotic networks with (1) motion control and (2) communication along $\mathcal{G}$

(ii) algorithms:
deployment, rendezvous, vehicle routing

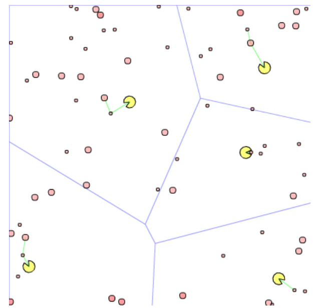
(iii) tools:
proximity graphs, objective functions, convergence & invariance

Scenario 5: Vehicle Routing

Objective: Given agents $(p_1, \dots, p_n)$ moving in environment Q
service targets in environment

Model:

- targets arise randomly in space/time
- vehicle know of targets arrivals
- low and high traffic scenarios
- low traffic scenario = deployment



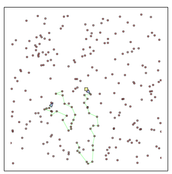
Scenario 5 —min expected waiting time

Scenario 5: receding-horizon TSP algorithm, I

Name: (Single Vehicle) Receding-horizon TSP

For $\eta \in (0, 1]$, single agent performs:

1: while no targets, move towards centroid
2: while targets waiting,
  (i) compute optimal TSP tour through all targets
  (ii) find a η-tour fragment with maximal number of targets
  (iii) service optimal fragment from the current location
3: Repeat



Asymptotically optimal in light and high traffic

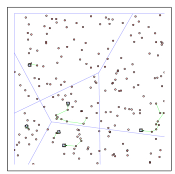
Scenario 5: receding-horizon TSP algorithm, II

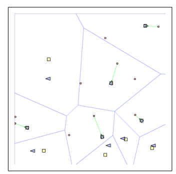
Name: Receding-horizon TSP

For $\eta \in (0, 1]$, agent i performs:

1: compute own Voronoi cell V_i
2: apply Single-Vehicle RH-TSP policy on V_i

Asymptotically optimal in light and high traffic (simulations only)





Emerging Motion Coordination Discipline

Emerging behaviors

optimal deployment, rendezvous, vehicle routing

Approach for design of algorithms:

- meaningful aggregate cost functions
- class of (gradient) algorithms local, distributed
- theoretical ingredients for analysis: proximity graphs, invariance principles

Features of algorithms:

scalable, adaptive, convergent, asynchronous

Ongoing Work

- **motion coordination**
  - dynamic 3D environments
  - vehicle dynamics
  - pattern formation

- **integrations, extensions and implications**
  - integrated motion coordination and sensor/estimation tasks
  - experimental implementation and verification
  - connections with biological networks
  - complexity analysis: communication/time/control cost