Dynamic Task-Based Coordination of Mobile Robotic Networks

Emilio Frazzoli
Mechanical and Aerospace Engineering
University of California, Los Angeles

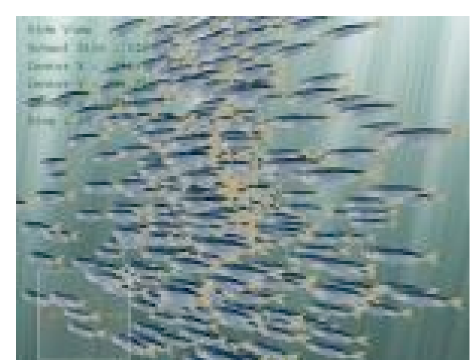
Thanks to:
A. Bertozzi, J. Enright, C. Hsieh, M. Pavone (UCLA)
F. Bullo, K. Savla (UCSB)
V. Sharma, P. Voulgaris (UIUC)

IPAM Workshop on
Swarming by Nature and by Design
March 3, 2006

ARES
Aerospace Robotics and Embedded Systems Laboratory

Large-scale Cooperative Robotics: Swarm engineering

- Large groups of animals often exhibit sophisticated collective behaviors, arising from the local interaction of "**agents**" with individual goals, and limited ability to exchange information.
  - The emergence of complex collective behaviors from local interactions is a fascinating research topic.
  - Today large-scale teams or **"swarms" of robots** are becoming increasingly feasible; growing interest in motion coordination algorithms, **scalable** to large groups.
- Substantial efforts devoted to understanding **HOW** to
  - Design local interaction rules to **recreate natural-looking large-scale swarm dynamics**, in simulation and, more recently, in experimental demonstrations.
  - Design decentralized algorithms to achieve certain **basic tasks** (formation flight, rendez-vous, deployment, etc.), in such a way that they are scalable to large-scale systems.

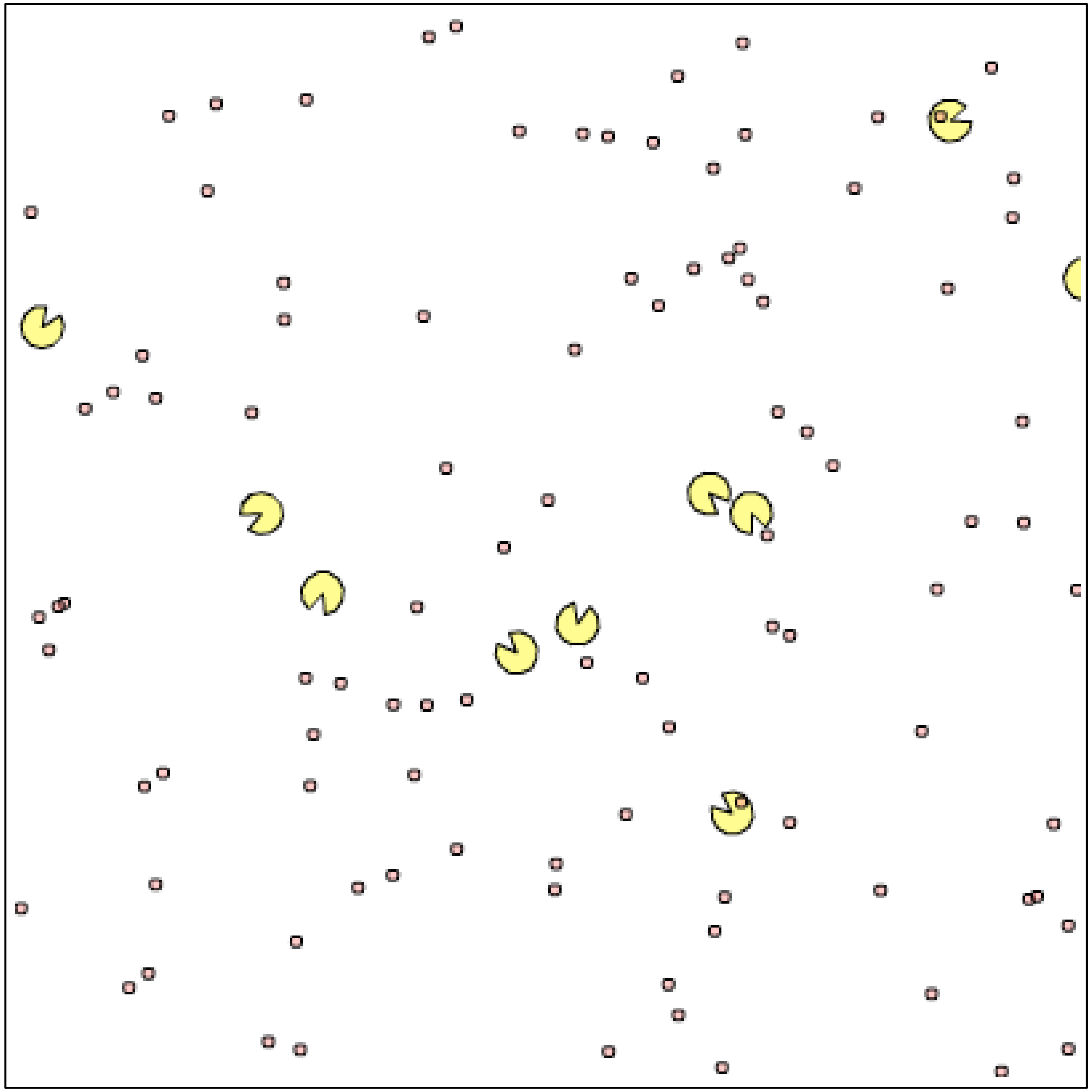


ARES
Aerospace Robotics and Embedded Systems Laboratory

Why swarms?

- Little effort has been devoted to understand **WHY** we should design large-scale systems, and develop "swarming" technology.
  - If it is done by biological systems, it must be good somehow (?)
  - Large-scale systems offer redundancy, graceful performance degradation, etc.
- Should we design/field "large" robotic teams, and **WHY**? Are there any fundamental benefits associated to large-scale networks?
  - Are there any tasks that are better suited to large teams?
  - What are **the advantages of numbers**? Are there any (fundamental) **disadvantages** to numbers? What are the tradeoffs?
  - Given a task, can we determine what is the **most appropriate size** of a robotic team for best efficiency?
- There is a need for the development of models and tools for the **Algorithmic Analysis of Multi-Agent Mobile Systems.**

The multi-PACMAN™ problem



Aerospace Robotics and Embedded Systems Laboratory

Problem Formulation 1/2

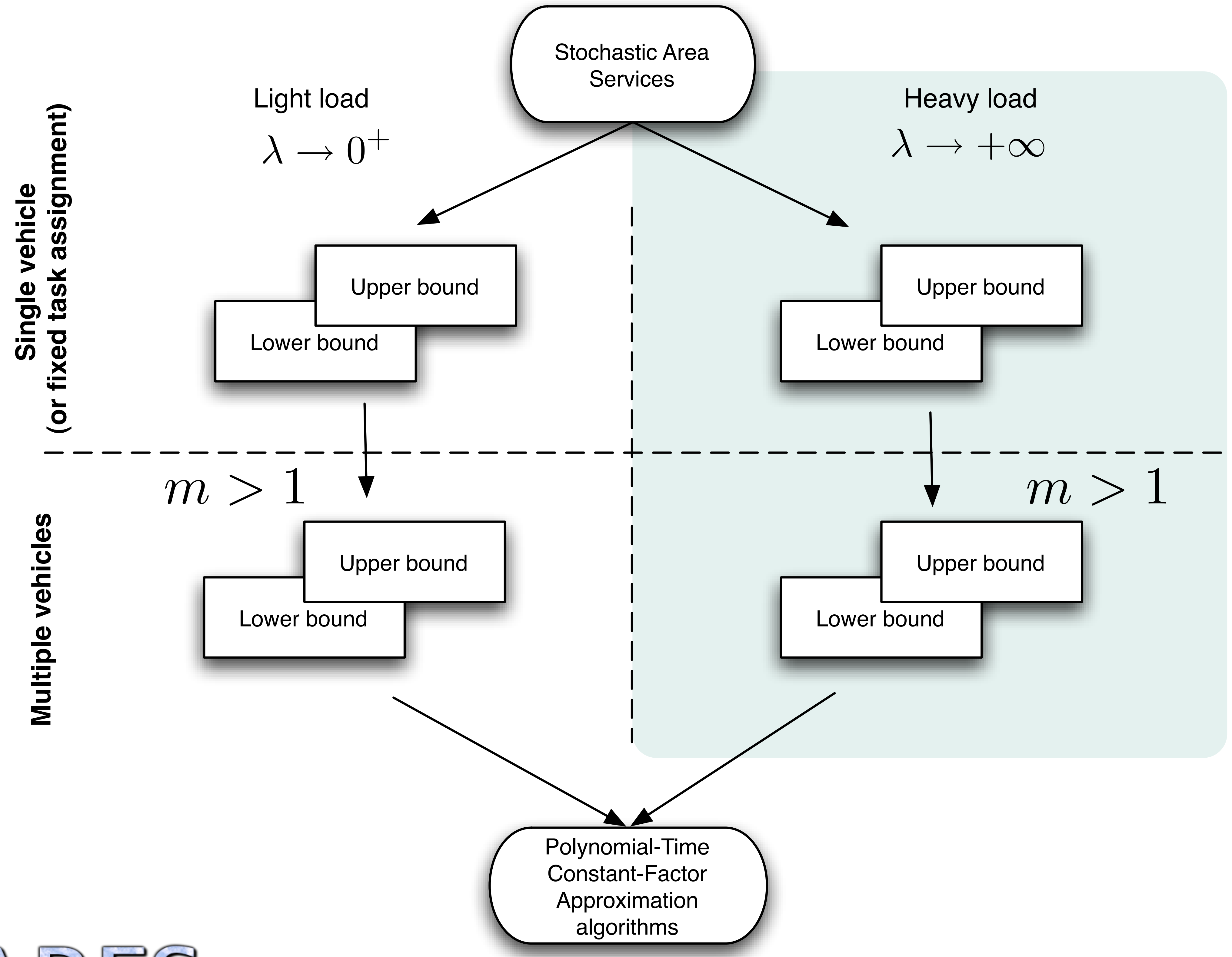
- **Targets** are generated within a convex set Q on the plane. Targets may represent locations of:
  - "Service requests" (**Dynamic Traveling Repairperson Problem**);
  - "Food items" (**Foraging**);
  - "Threats to investigate" (**UAV Routing**)
  - "Information packets" (**Data harvesting, communication relays**)
- Target generation is modeled as a stochastic (Poisson) point process $\Pi(\lambda, \varphi)$:
  - **Static/Dynamic** generation: n_0 targets present at the initial time; afterwards, λ targets generated, on average, per unit time.
  - **Uniform/Non-uniform** spatial p.d.f. φ_0 and φ (resp. for the static/dynamic distributions).
  - *In other words, the expected number N of events generated over time [0, t] in* $\mathcal{S} \subseteq Q$ *is*

$$\mathrm{E}[N(\mathcal{S}, t)] = \lambda t \cdot \varphi(\mathcal{S})$$

Problem Formulation 2/2

- **Mobile Agents:** a team of *m* non-holonomic vehicles,
  - Bounded speed, non-integrable constraints on the direction of the velocity. **Dubins**' car, **Reeds-Shepp**'s car, **Differential Drive** robots, etc. (good models for UAV's, wheeled/threaded vehicles).
  - All vehicles are identical and have unlimited target-servicing capacity.
- **Objective:** Devise a control policy μ to
  - make the target queue **stable**, and
  - minimize the exp. waiting time T_μ (**max. QoS**). Little's formula: $n_\mu = \lambda \, T_\mu$.
- The policy μ can be split into:
  - a **task assignment** policy.
  - a **path-optimization policy**, which determines the best sequence in which assigned targets must be serviced, and the best path to do so.
- The policy does not have access to future events, but may use a model of the stochastic process $\Pi(\lambda,\phi)$.

A roadmap to the analysis & design process



ARES

Aerospace Robotics and Embedded Systems Laboratory

Previous work: the Euclidean case

- The earliest study on a similar class of problems, considering **holonomic vehicles** with bounded speed, is due to **Psaraftis** (1988). A thorough discussion and important results are given in a sequence of papers by **Bertsimas** and coworkers (DTRP, 1990-'93).

- Bertsimas et al. established the following lower bounds on the waiting time (assuming a uniform density):
  - In the case of **light load** $(\lambda \to 0^+)$:

$$T^* \geq \min_{p \in Q^m} H_m(p) = \min_{p \in Q^m} \sum_{i=1}^{m} \int_{\mathcal{V}_i(p)} \|q - p_i\| \, dq = \Theta(1/\sqrt{m})$$

where H_m is the continuous Weber function, or multi-median function, and $\mathcal{V}(p) = \{\mathcal{V}_1(p), \mathcal{V}_2(p), \ldots, \mathcal{V}_m(p)\}$ is the Voronoi partition of the set Q generated by the points p.

  - In the case of **heavy load** $(\lambda \to +\infty)$:

$$T^* \geq \frac{\gamma^2 \lambda}{m^2} = \frac{2}{9\pi} \frac{\lambda}{m^2}$$

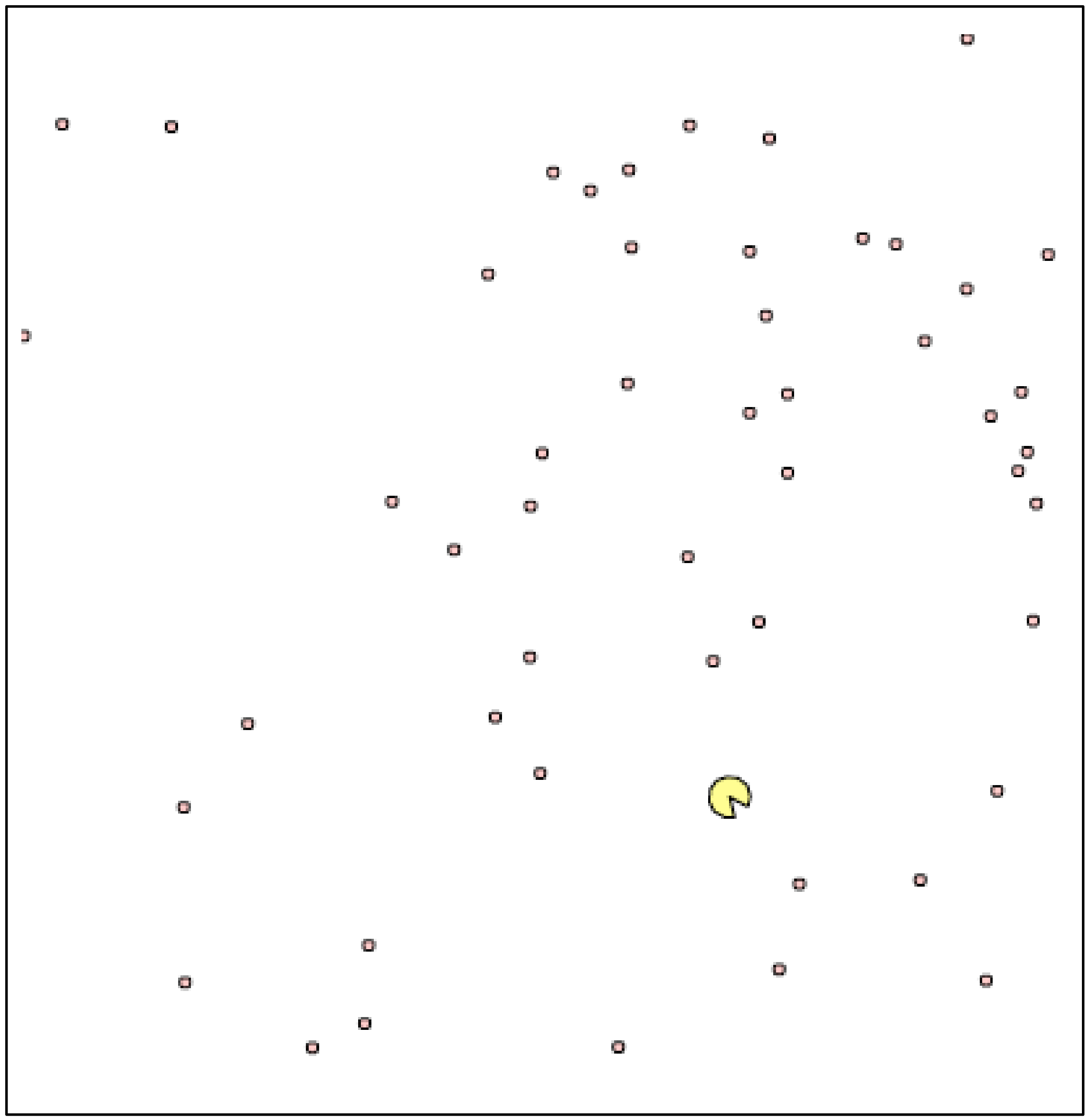
The single holonomic agent case

- Consider a single holonomic vehicle, moving with bounded velocity on the plane.
- **NN strategy**: greedily visit the **nearest target**
  - Static case: log *n* approximation to the TSP
  - Dynamic case: constant factor approximation to min. expected waiting time. (No formal proof available.)
- **sRH strategy** *[Frazzoli, Bullo CDC'04]*: pick the **densest cluster** including at least a fraction η of all outstanding targets, visit all targets in the cluster in min time (solve a "local TSP"). Repeat. If there are no targets, move to the **median**.
  - **Static case:** same as TSP (choose η = 1)
  - **Dynamic case:** Optimal in light load, in heavy load recovers (as η ↓ 0) the performance of the **best known policy**, conjectured to be in fact optimal. Min. expected waiting time approx. **37% lower than NN policy**.

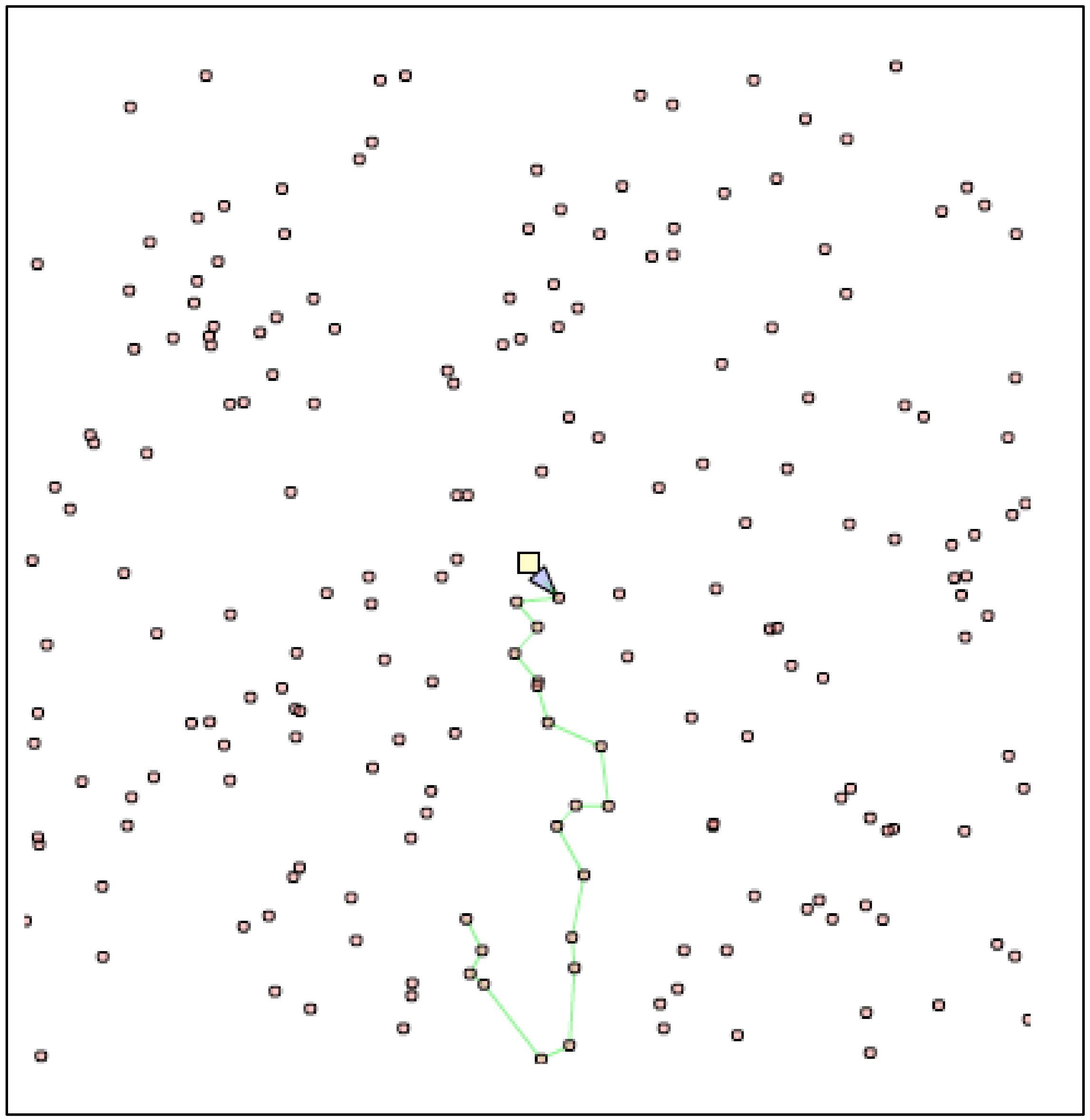
$$T_{\mathrm{sRH}} \leq \frac{\beta_{\mathrm{TSP}}^2}{2-\eta} \frac{A\lambda}{v_{\max}^2} < T_{\mathrm{NN}} = \gamma_{\mathrm{NN}}^2 \frac{A\lambda}{v_{\max}^2}, \qquad \text{for } \eta < 0.7$$

Aerospace Robotics and Embedded Systems Laboratory

Simulation Examples



Nearest neighbor



sRH (“densest cluster”)

ARES

Aerospace Robotics and Embedded Systems Laboratory

The multiple-vehicle case

- Associate to each agent a virtual generator g_i, $i=1, \ldots, m$.
- Let $\mathcal{V}_i(g, \mathcal{Q})$ be the Voronoi cell generated by the i-th generator.
- The mRH/VG policy is the composition of the following:

$$\dot{g}_i = \begin{cases} -k\dfrac{\partial}{\partial g_i} H_m(g, \mathcal{Q}) & \text{if } D \cap \mathcal{V}_i(g, \mathcal{Q}) = \emptyset \\ -k\dfrac{\partial}{\partial g_i} \displaystyle\sum_{i=1}^{m} \mathrm{Area}[\mathcal{V}_i(g, \mathcal{Q})]^3 & \text{otherwise} \end{cases}$$

$$\dot{p}_i = \mathrm{sRH}(p_i, D \cap \mathcal{V}_i(g, \mathcal{Q}))$$

- **Remark:** The gradients appearing in the mRH policy can be computed in a spatially decentralized way [Cortes et al., 02], and the sRH relies only on local information: the mRH/VG policy is **spatially decentralized**.

mRH/VG performance

- Theorem: The mRH/VG policy is (locally) optimal in the light load case
- Theorem: The mRH/VG policy (locally) recovers the performance of the best known algorithms, that is,

$$T_{\mathrm{mRH/VG}} \to \frac{\beta^2}{(2-\eta)\gamma^2} T^*, \qquad \text{as } \lambda \to +\infty$$

- Expected waiting time in heavy load for fixed generators:

$$T^*_{\mathrm{mRH/VG}} = \frac{\beta^2 \lambda}{(2-\eta)} \sum_{i=1}^{m} \frac{\mathrm{Area}[\mathcal{V}_i(g, \mathcal{Q})]^3}{A^2}$$

- Since $\sum_{i=1}^{m} \mathrm{Area}[\mathcal{V}_i(g, \mathcal{Q})] = A$, the expected waiting time is minimized when all regions have equal area, i.e.,

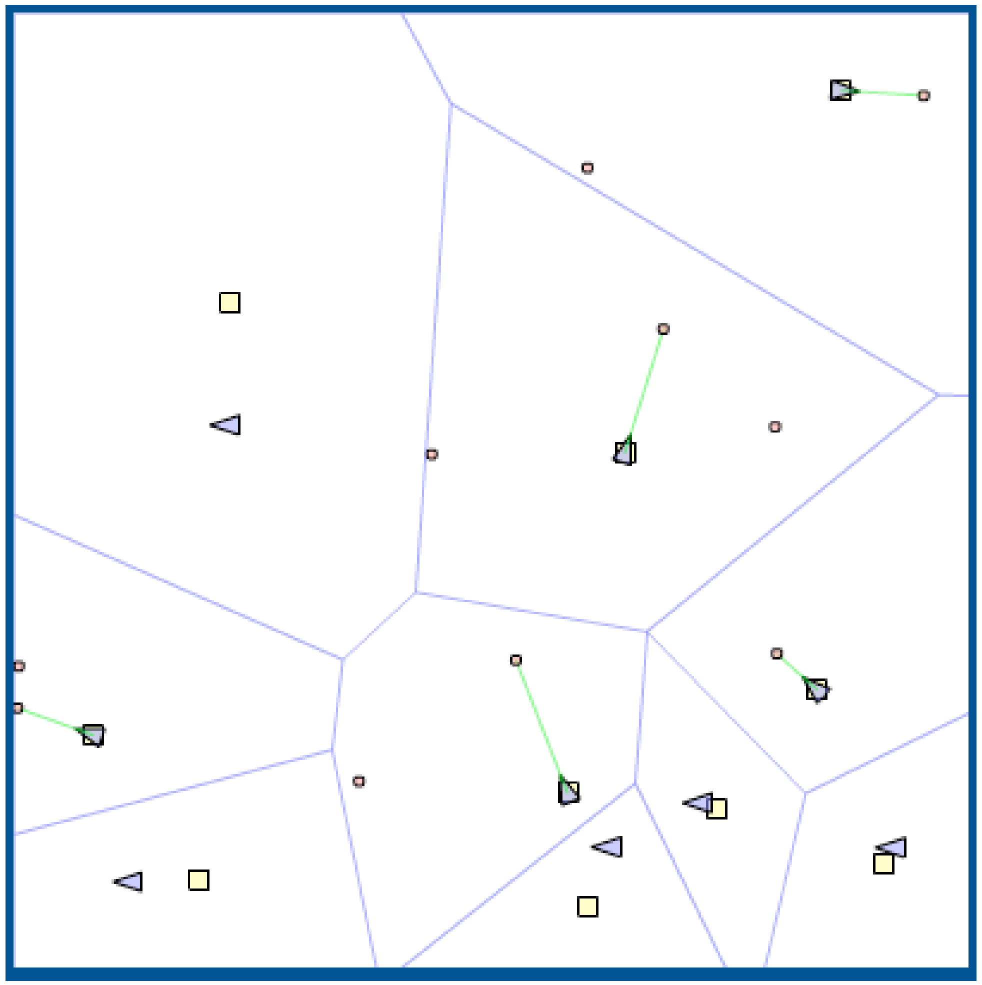
$$T^*_{\mathrm{mRH/VG}} = \frac{\beta^2 \lambda}{(2-\eta) m^2} A$$

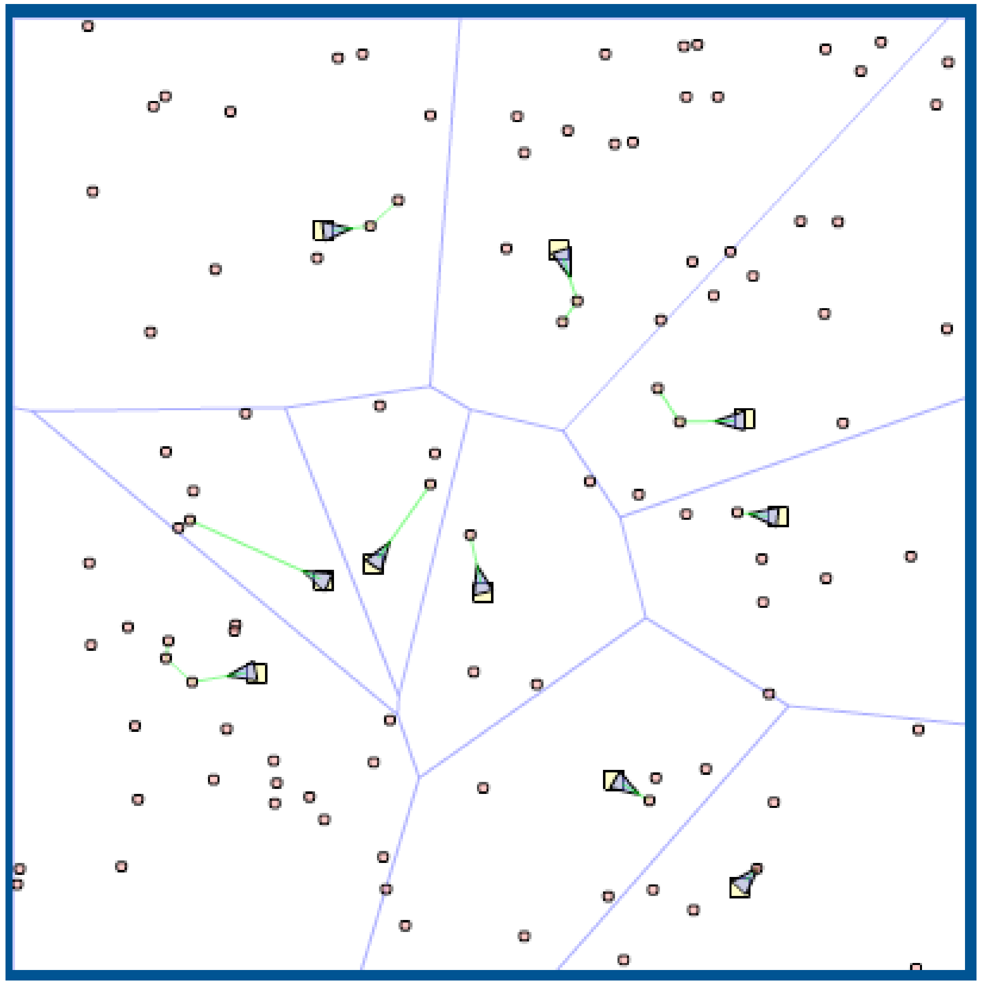
ARES

Aerospace Robotics and Embedded Systems Laboratory

Simulation Results

- The proposed **decentralized, scalable algorithms** achieve (locally) the same performance as the best known centralized algorithms



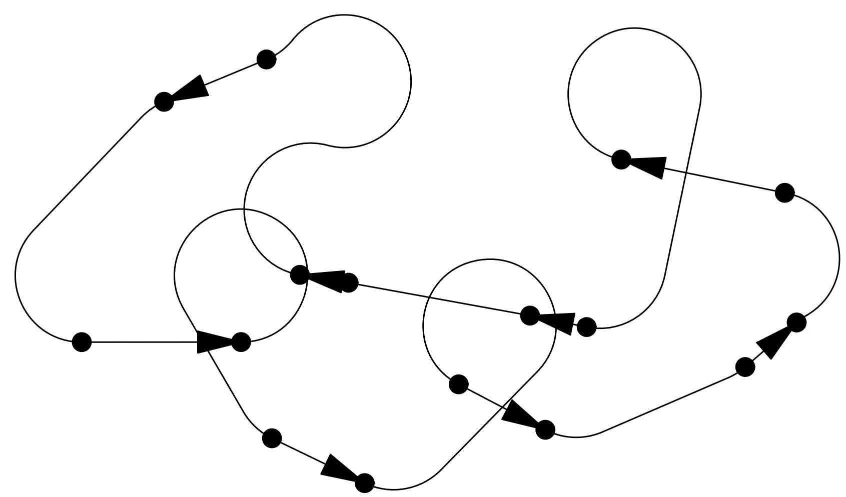


What happens if the vehicles' dynamics are subject to non-holonomic constraints?

ARES
Aerospace Robotics and Embedded Systems Laboratory

The single-vehicle problem in heavy load

- The solution of the heavy-load case requires an understanding of the asymptotic cost of **Traveling Salesperson Problems** with differential constraints on the path.
  - ***NP-hardness a consequence of the NP-hardness of the Euclidean TSP.***



- Any reasonable algorithm will yield feasible paths of length $O(n)$.
  - If the cost of the path increases linearly with n, the average cost per target is a constant. There is a threshold value for λ beyond which stability is no longer guaranteed.
- ***Does the cost of the TSP with diff. constraints increase SUBLINEARLY with n?***
- ***Is there a polynomial-time algorithm that returns a tour of length < O(n)?***

ETSP vs. DTSP

- The Euclidean TSP (ETSP) is one of the prototypical "hard" combinatorial optimization problems.
  - The exact solution is extremely hard to compute.
  - Good approximations are "easy" to obtain.
- **Stochastic ETSP** [Beardwood et al., '59]: let ETSP(n) be a random variable representing the minimum length of a tour through n points sampled from a uniform distribution in a *d*-dimensional set of measure 1.

$$\lim_{n \to \infty} \frac{\mathrm{ETSP}(n)}{n^{1-1/d}} = \beta_d, \qquad \text{a.s.}$$

- The **Dubins TSP (DTSP)** is fundamentally different:
  - Non-metric problem: might not be even approximable.
  - No known reduction to a problem on a finite graph.
  - **Open problem, present in all UAV routing applications.**

A nearest-neighbor lower bound

- The area of the set of points reachable with a path of length δ by a Dubins' car with turning radius >= ρ is

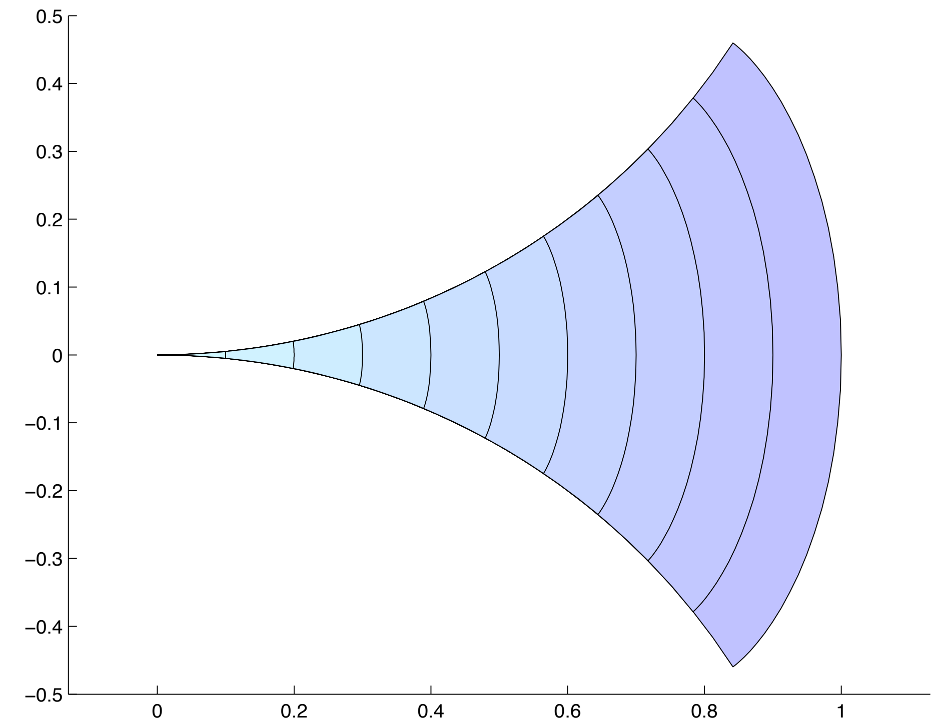
$$\mathrm{Area}[\mathcal{R}_\delta] = \frac{\delta^3}{3\rho}$$

- The expected distance to the nearest target, out of *n* uniformly-distributed targets is

$$\mathrm{E}[\delta^*] = \frac{3}{4}\left(\frac{3\rho}{n}\right)^{\frac{1}{3}}$$

- The length of the tour cannot be less than *n* times such a distance, hence:

$$\boxed{\mathrm{E}[\mathrm{DTSP}_\rho(n)] \geq \frac{3}{4}\left(3\rho n^2\right)^{\frac{1}{3}}}$$



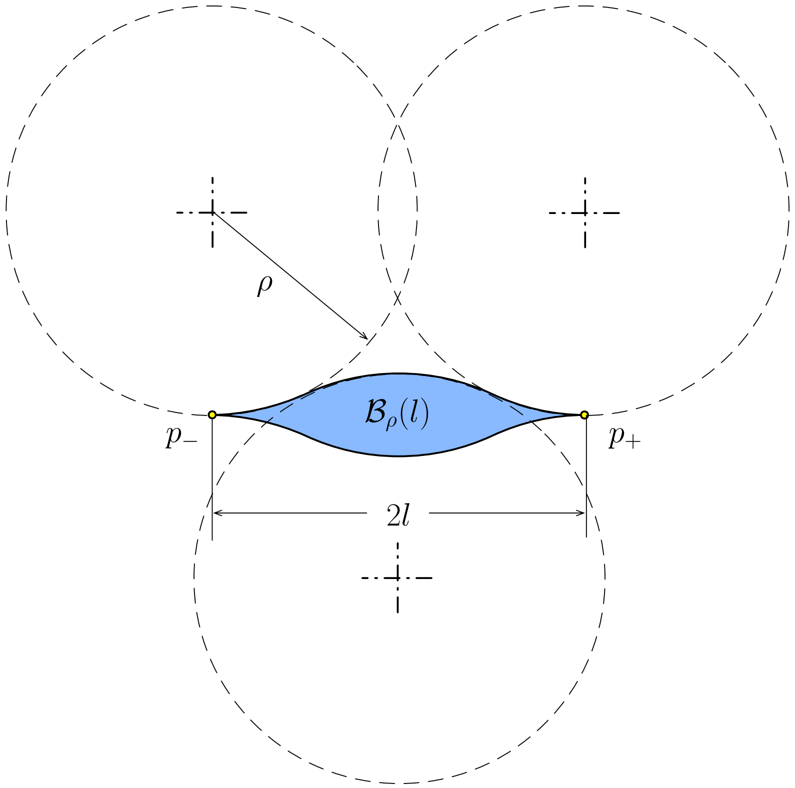
The small-time reachable set for Dubins' vehicle.

ARES

Aerospace Robotics and Embedded Systems Laboratory

The Bead Tiling Algorithm 1/2

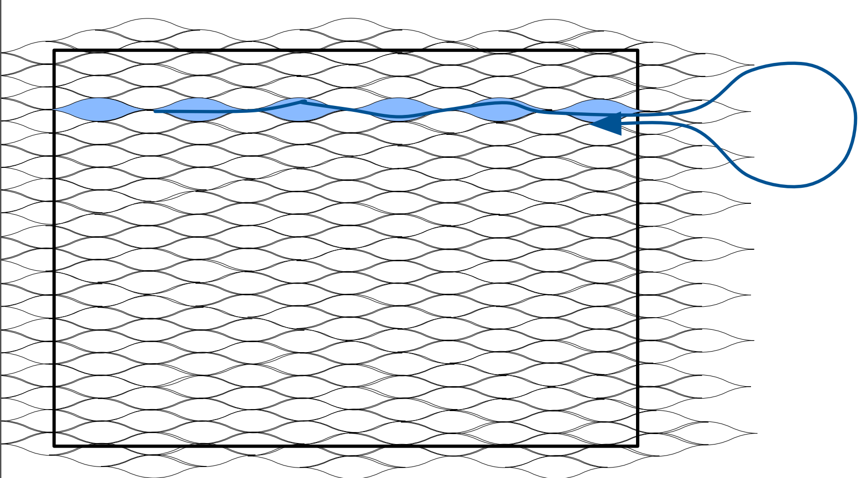
- Basic geometric construction: the **"Bead"**



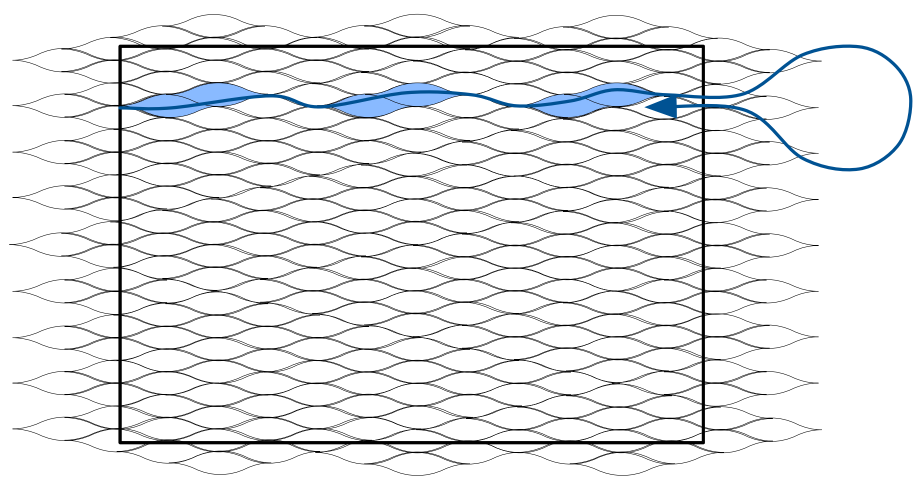
- Properties of a bead of length *l*:
  - A path of length $l+o(l^2)$ always exists between the end points and an arbitrary point in the bead.
  - The "width" of the bead is $l^2 + o(l^3)$.

The Bead Tiling Algorithm 2/2

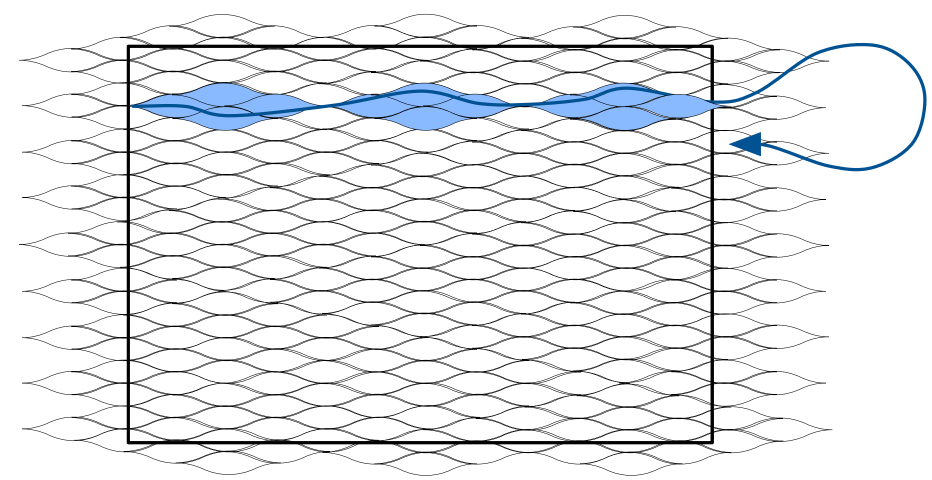
- Tile the region of interest with beads such that: $\mathrm{Area}[\mathcal{B}_\rho(l)] = \dfrac{\mathrm{Area}[\mathcal{Q}]}{2n}$
  Sweep the bead rows, visiting one target per non-empty bead.
- Iterate, using at the *i*-th phase a "meta-bead" composed of 2^{i-1} original beads.
- After log *n* phases, visit the outstanding targets in any arbitrary order, e.g., with a greedy strategy.



Phase 1



Phase 2



Phase 3

Analysis of the BTA

- **Theorem:** The cost of stochastic DTSP satisfies the following inequalities, *with high probability*

$$\beta^-(\rho) := \frac{3}{4}(3\rho)^{1/3} \leq$$

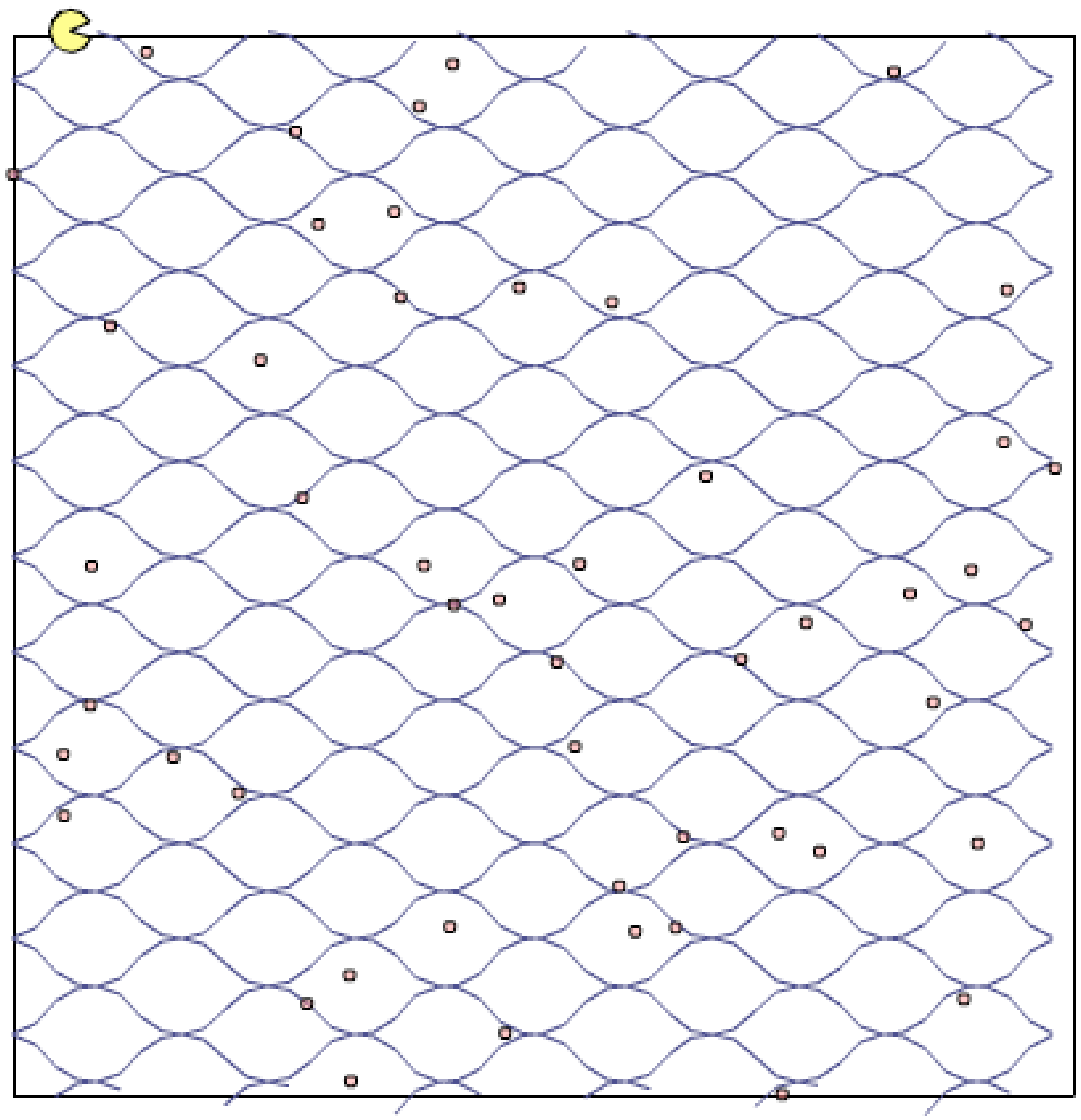
$$\leq \lim_{n\to\infty} \frac{\mathrm{DTSP}_\rho(n)}{n^{2/3}} \leq$$

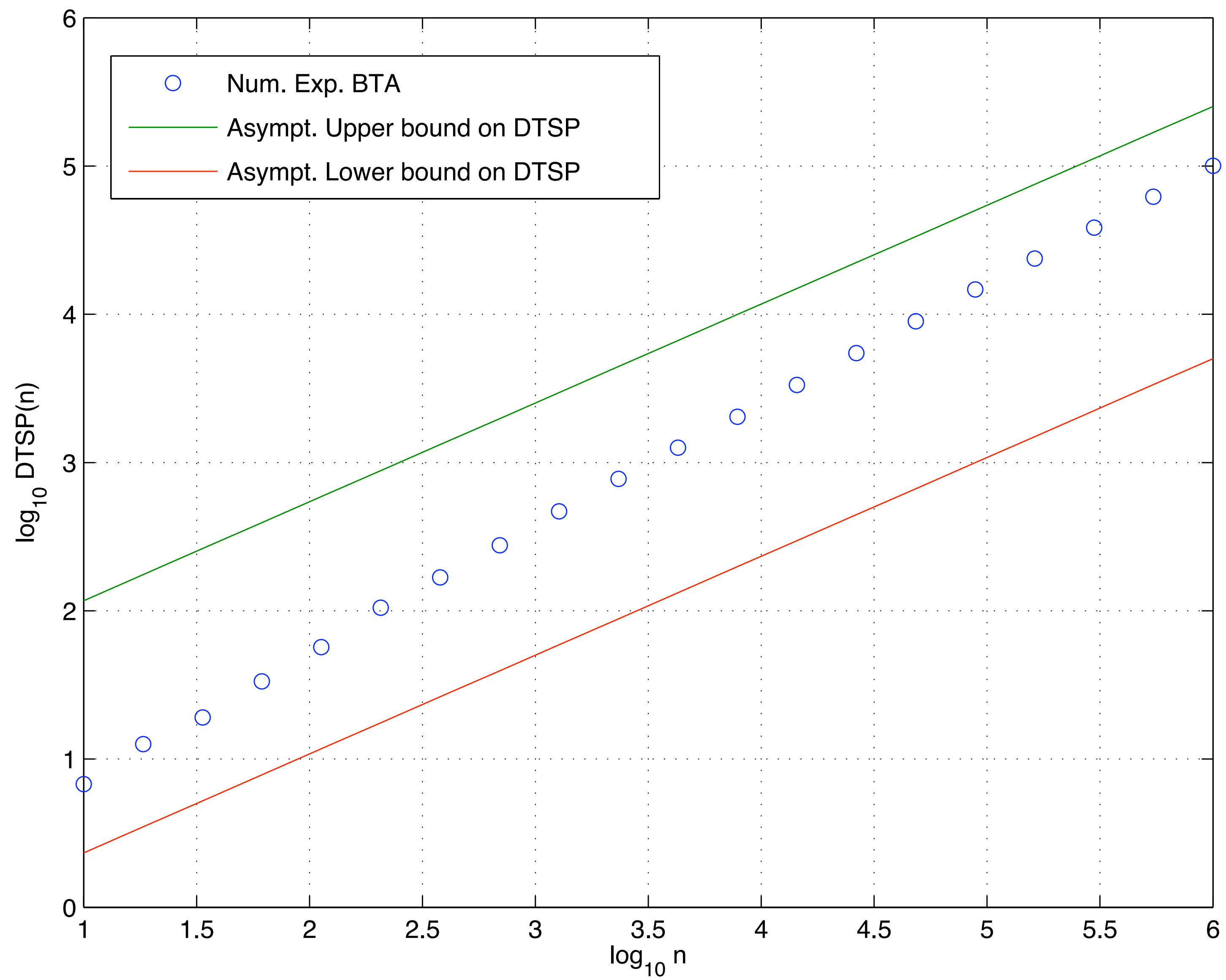
$$\leq 9.88\sqrt[3]{\rho W H}\left(1 + \frac{7}{3}\pi\frac{\rho}{W}\right) =: \beta^+(\rho)$$

- Outline of the proof:
  - Let v_i be the number of non-empty beads at the inception of the i-th phase.
  - Show (by induction) that $v_i \leq 2^{1-i}\, n$ w.h.p., for all $i \leq i^* \leq \log_2 n$.
  - Show that at the end of the i*-th phase, almost all (i.e., n-O(log n)) targets have been visited. The cost of visiting the O(log n) leftovers is negligible.
  - Show that the cost of the first i* phases is a constant time the cost of the first phase, which in turn is $O(n^{2/3})$

ARES
Aerospace Robotics and Embedded Systems Laboratory

Numerical Experiment Results





The dynamic case

- The static DTSP results can be directly applied to the dynamic case, using a **receding horizon strategy**
  - Take a snapshot of outstanding targets at time t_i, and visit them using the BTA.
  - Meanwhile, new targets accumulate, until $t_{i+1} \leq t_i + \mathrm{DTSP}(n(t_i))$.
  - Is there an "invariant set", in which $E[n(t_{i+1})] \leq n(t_i)$?
- **Theorem:**

$$\beta^-(\rho)^3 \leq \lim_{\lambda \to \infty} \frac{T^*}{\lambda^2} \leq \beta^+(\rho)^3$$

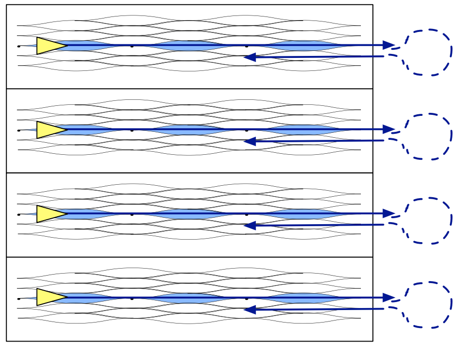
- **Notes**:
  - First result showing the stability of area services with Dubins vehicles.
  - Stronger dependency on λ (quadratic) than in the Euclidean case (linear).
  - Unlike the Euclidean case, such stability cannot be maintained for an adversarial target selection.

Cooperative policy design

- For an environment bounded by a W x H rectangle, the bead-tiling algorithm provides an upper bound of the form:

$$\lim_{\lambda\to\infty} \frac{T^*}{\lambda^2} \leq 9.88^3 \rho W H \left(1 + \frac{7}{3}\pi \frac{\rho}{\max\{W, H\}}\right)^3$$

  - The area of the region is not the only important factor: the shape plays a major role
  - The "non-STLC" penalty decreases as W/ρ increases (for constant WH).
- If *m* agents share the same region, the most efficient policy assigns distinct rows to each agent. The agent move roughly along parallel paths.
  - max{W,H} is unchanged, min{W,H} and λ scale down by *m:*



$$\lim_{\lambda\to\infty} \frac{T^*}{\lambda^2} \leq 9.88^3 \frac{\rho W H}{m^3} \left(1 + \frac{7}{3}\pi \frac{\rho}{\max\{W, H\}}\right)^3 = \left(\frac{\beta^+(\rho)}{m}\right)^3$$

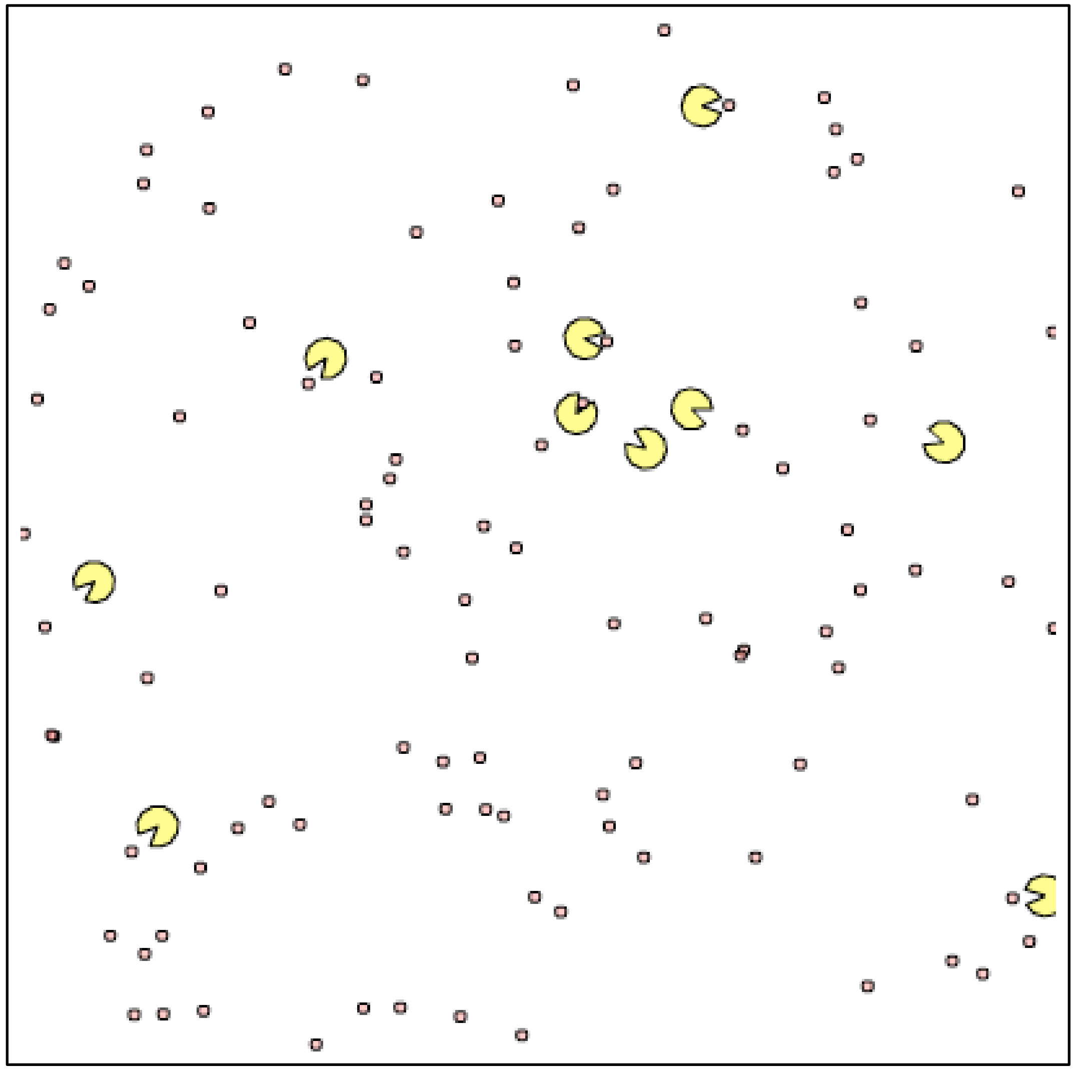
- **For non-holonomic vehicles, the benefits of multiple-vehicle cooperation are even stronger than in the Euclidean case.**
- **Implications for behavioral ecology?**

The multi- holonomic agent case

- Let m agents share the environment.
- Is a greedy policy any good?
  - Let $\Delta_{ij}(t)$ be the distance between agents i and j at time t. If all agents execute the NN policy, then we conjecture that

$$\lim_{t\to\infty} \mathrm{E}[\Delta_{ij}(t)] = 0, \forall i, j$$

  - Moreover, the rate of convergence increases as the density of targets decreases.



ARES
Aerospace Robotics and Embedded Systems Laboratory

The multi- holonomic agent case

- In other words, if all agents greedily pursue the nearest targets, the diameter of the team will collapse to zero.
  - A mechanism for **swarm cohesion** that is independent of inter-agent interactions.
  - Each agent will get its fair share of targets, i.e., each agent gets, on average, $E[U_i] = n/m$ targets.
  - However, it will take a long time to clear a static point distribution, and the exp. waiting time in the dynamic case is the same as the single-vehicle case!!!
- Introduce another policy:
  - **Voronoi-NN policy:** each agent pursues the nearest target within its own Voronoi region. (An agent moves to the centroid of its own Voronoi region while no targets are available.)
  - There will be no overlaps with others: once an agents decides to pursue a target, no other agent will be able to “steal” it.
  - Note: there is no need to compute the Voronoi region explicitly. A target at distance r is in an agent’s Voronoi region if there are no other agents within a circle of radius r from the target.

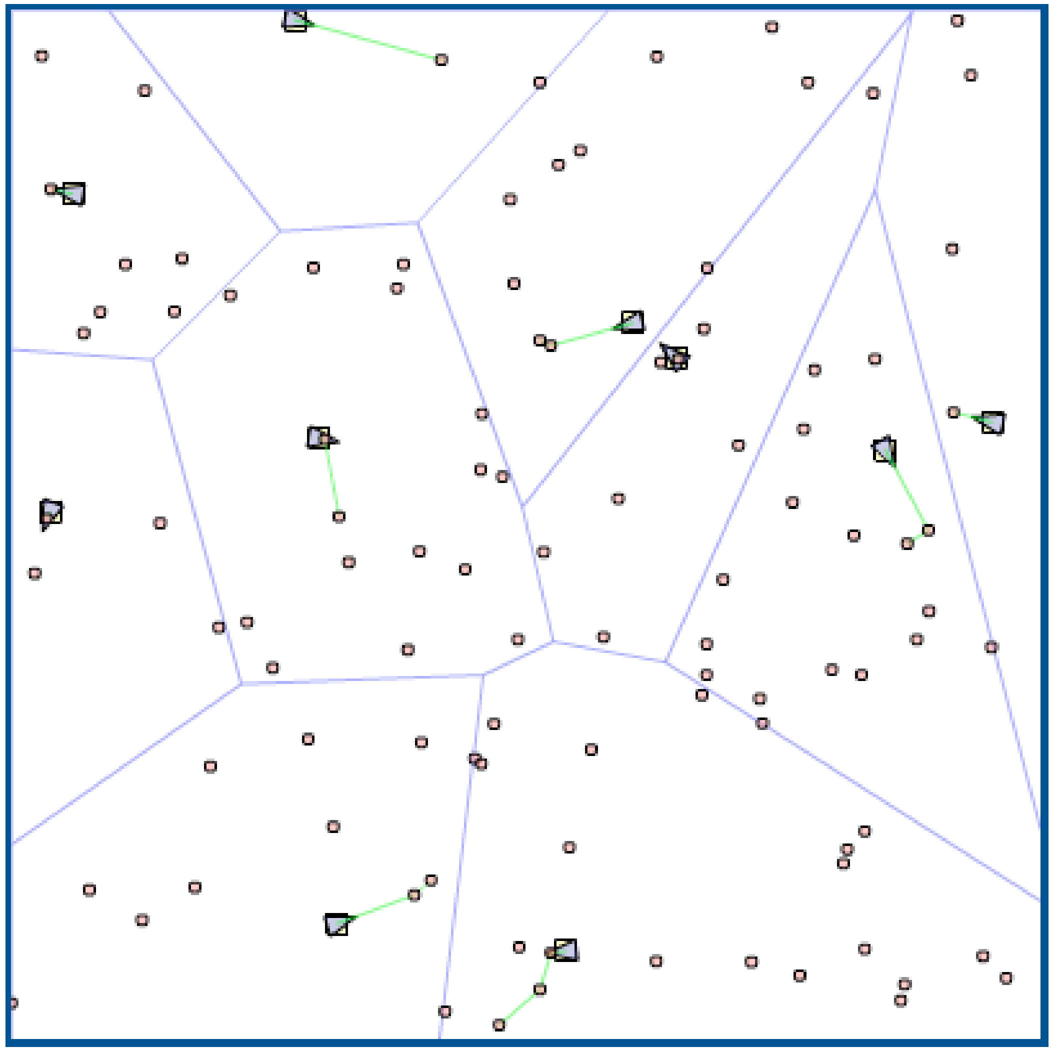
A static policy (mRH)

- A simpler version of the mRH/VG policy, with no virtual generators:

$$\dot{p}_i = \begin{cases} -k \dfrac{\partial}{\partial p_i} H_m(p, \mathcal{Q}) & \text{if } D \cap \mathcal{V}_i(p, \mathcal{Q}) = \emptyset \\ \mathrm{sRH}(p_i, D \cap V_i(p, \mathcal{Q})) & \text{otherwise} \end{cases}$$

- The mRH/VG policy is optimal in light load.

- Simulation experiments show that mRH performs at least as well as mRH/VG in heavy load.



A non-cooperative game view

- Assume that m_v agents execute the Voronoi-NN policy; then the expected payoff is *(assuming at least one agent is executing the NN policy)*
  - $E[U_i] \geq n/(m_v+1)$ for an agent executing the Voronoi-NN policy.
  - $E[U_i] \leq n/(m_v+1)/(m-m_v)$ for an agent executing the NN policy.

in the 2-agent case...	2: NN	2: Voronoi-NN
1: NN	n/2, n/2	$< n/2$, $> n/2$
1: Voronoi-NN	$> n/2$, $< n/2$	n/2, n/2

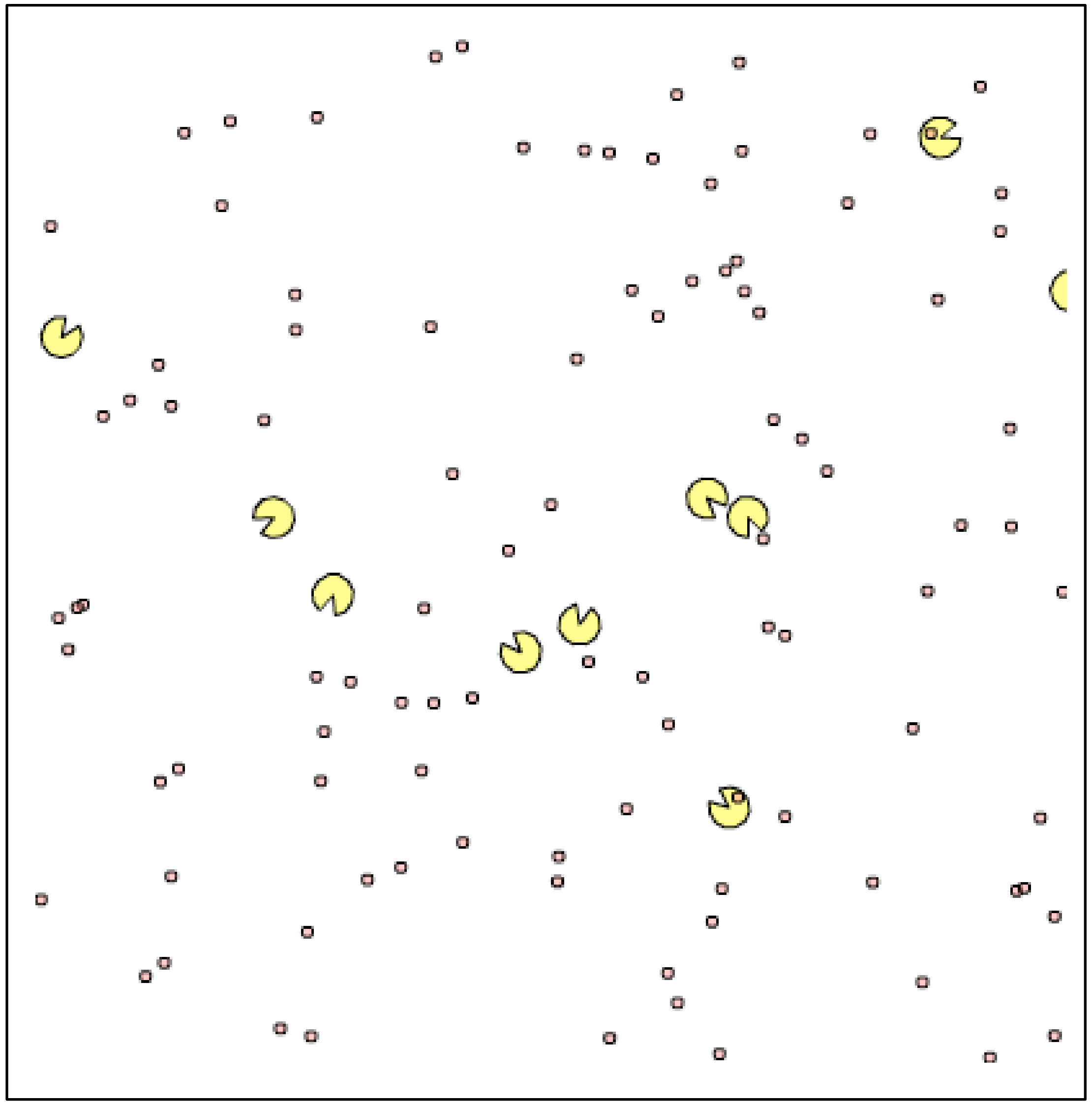
- In other words, all agents executing the Voronoi-NN policy is a (**pure strategy) Nash equilibrium**.
  - The individual payoff at equilibrium is no better than the "fair share"
  - However, the distance traveled/time needed to clear a static instance is decreased, as is the min. expected waiting time

$$T_{\mathrm{Voronoi-NN}} = \gamma_{NN}^2 \frac{A\lambda}{v_{\max}^2 m^2}$$

  - Note that the waiting time decreases with the square of the number of vehicles!

Greedy policy for non-holonomic agents

- For non-holonomic agents, if the average inter-target distance is small enough with respect to the the "turning radius" (i.e., if $\delta^*/\rho \ll 1$) the probability of two agents sharing the same nearest neighbor vanishes---for **any distance** between the agents.
  - **"Swarms" of non-holonomic agents executing a NN policy do not collapse, if the target density is high enough.**
  - **No inefficiencies occur with respect to the single-agent NN rule!**
- Numerical evidence suggests that a pure NN rule provides a constant-factor approximation to optimal performance.
- What if δ^*/ρ is not very small?



The light-load case: summary

- **Lower bound** for individual territory:

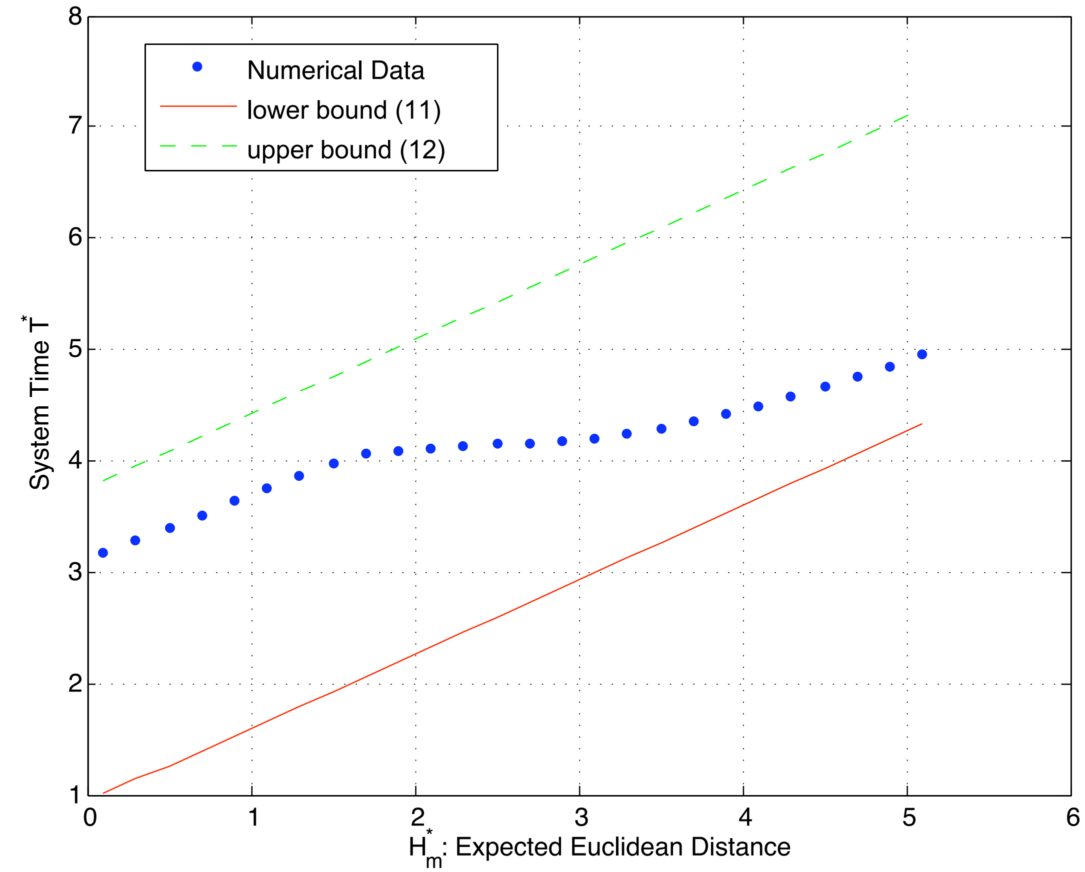
$$T^* \geq H_m^*(\mathcal{Q}) + \rho\left(\frac{\pi}{2} - \frac{2}{\pi}\right) \approx H_m^*(\mathcal{Q}) + 0.9342\rho$$

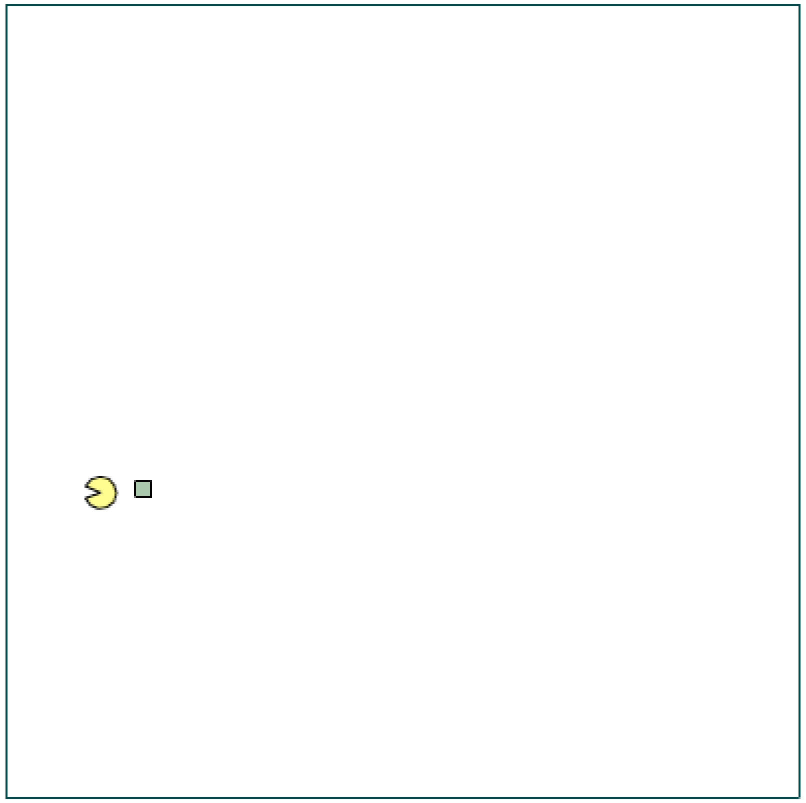
- **Upper bound**: The **Offset Median** policy
  - Loiter around the multi-median points with radius ~ 2.9 ρ

$$T^* \leq H_m^*(\mathcal{Q}) + 3.756\rho$$

- Lower bound with *l* teams of *m*/*l* agents each ("**wolf packs**"):

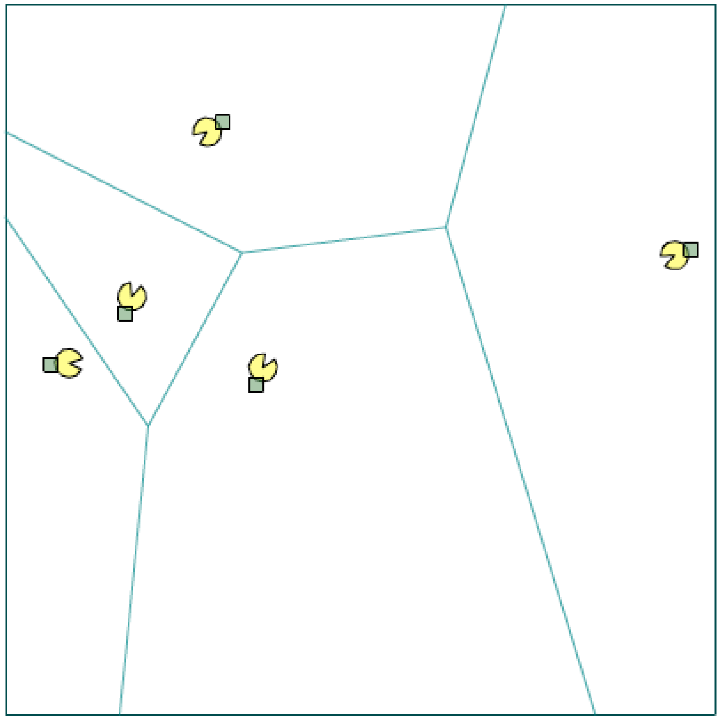
$$T^* \geq H_l^*(\mathcal{Q}) + \rho\left(\frac{\pi l}{2m} + \frac{m}{\pi l}\left(\cos\frac{\pi l}{m} - 1\right)\right)$$

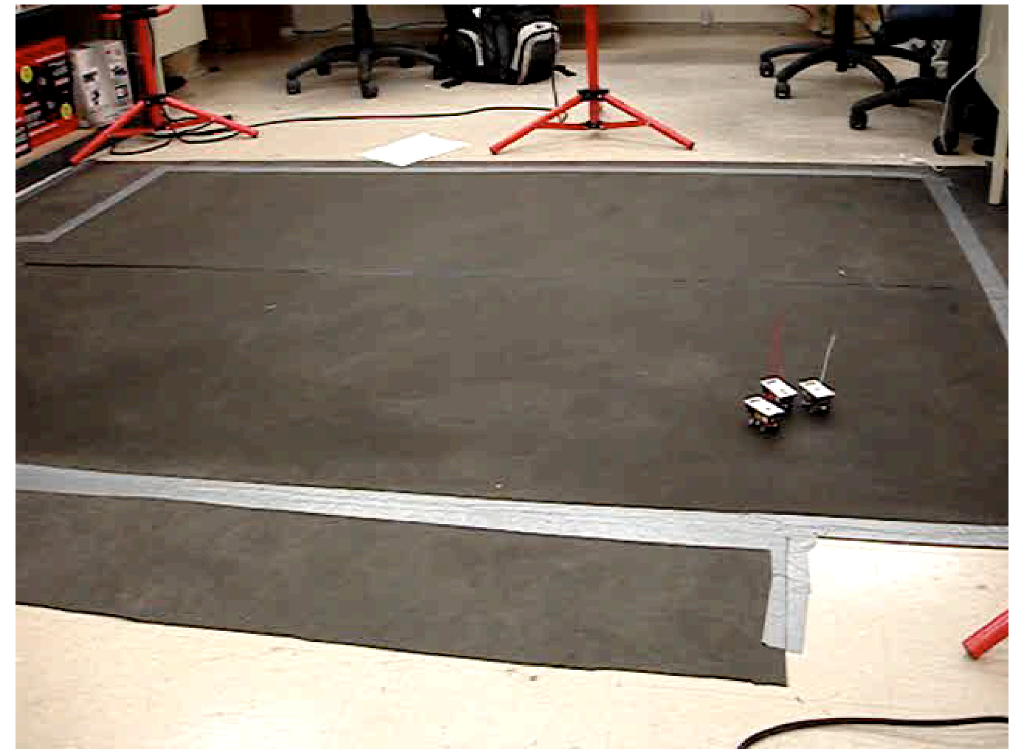


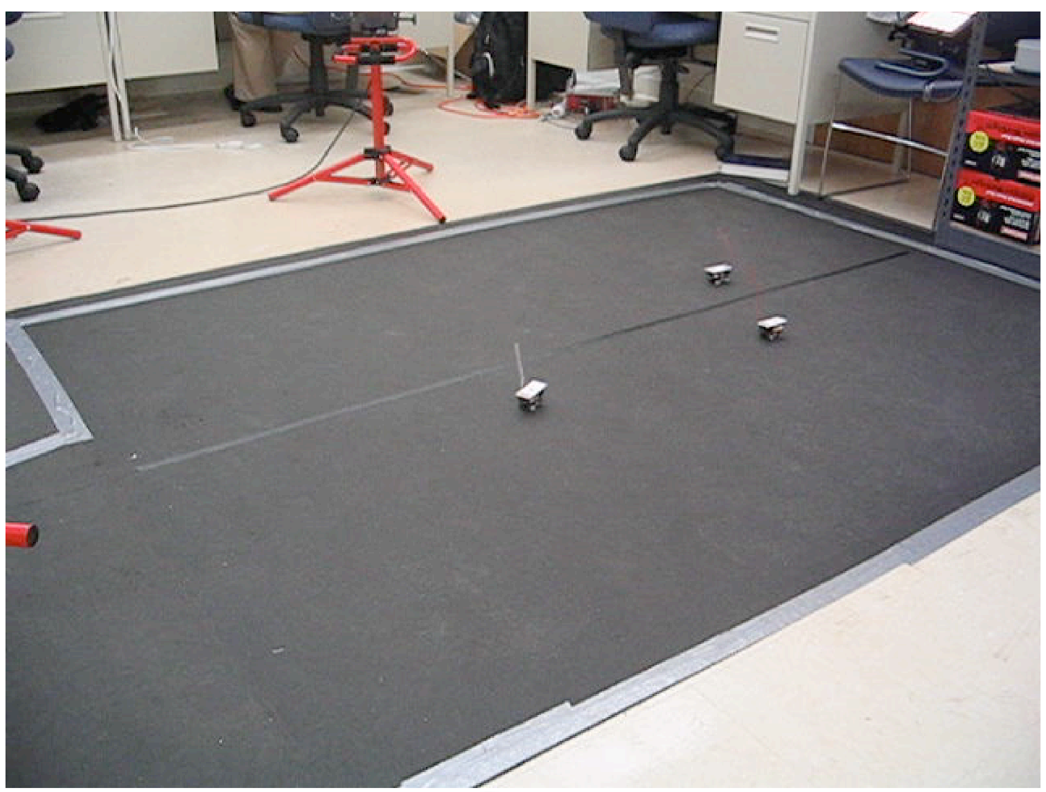


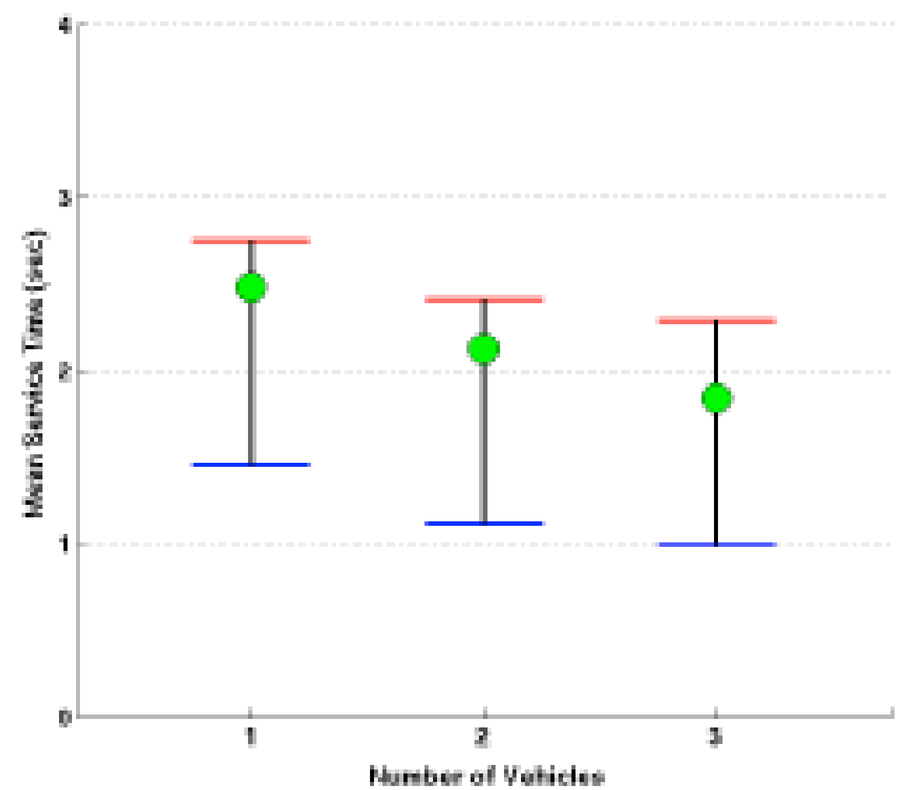
Aerospace Robotics and Embedded Systems Laboratory

Light load: results









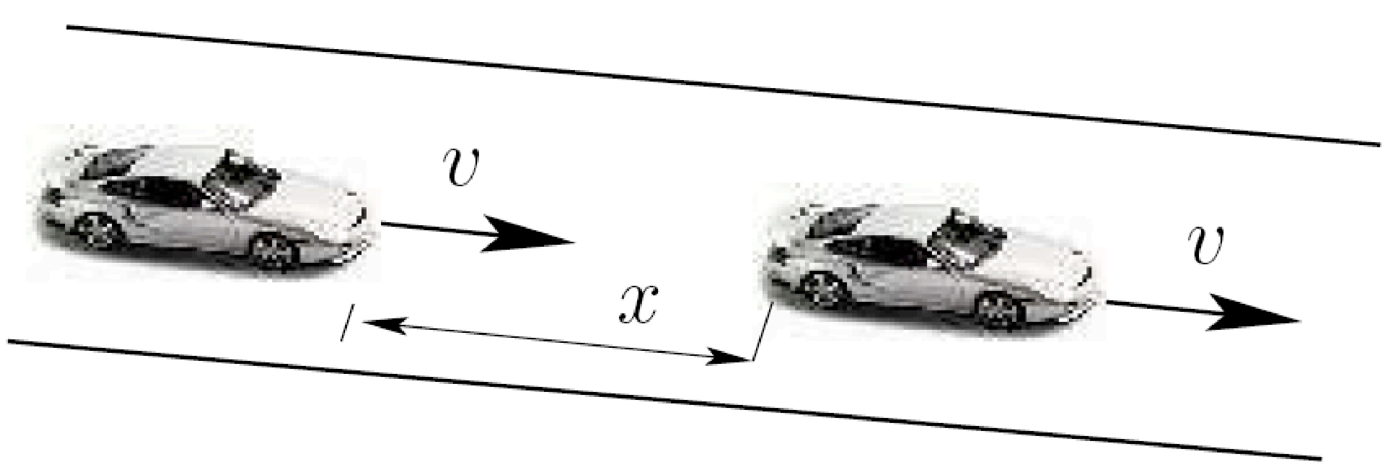
Traffic congestion in robotic networks

- So far we have seen that in area service applications, the system times always decreases with the number of agents:
  - As $1/\sqrt{m}$ in the light load case
  - As $1/m^2$ in the heavy load case
  - As $1/m^3$ (**!**) in the heavy load case, with non-holonomic constraints.
- We have not modeled conflicts (i.e., collisions/near misses) between agent: What happens if we do?
- Let us consider a simpler problem, in which origin and destination points are given a priori, for each agent.

Safe decentralized mobile systems

- Consider a system composed on *n* independent mobile agents
  - Automotive traffic
  - Air Traffic Control (Free Flight concept)
  - Uninhabited + human-piloted aircraft operations
  - Robotic swarms
  - Factory automation systems
- Safety concerns:
  - An agent does not know the intentions of nearby agents.
  - Delays and uncertainties in sensing/estimating own and others' motion
  - Need a **safety buffer**; The dimension of the safety buffer depend, among other things, on the vehicle's velocity

$$x > kv$$



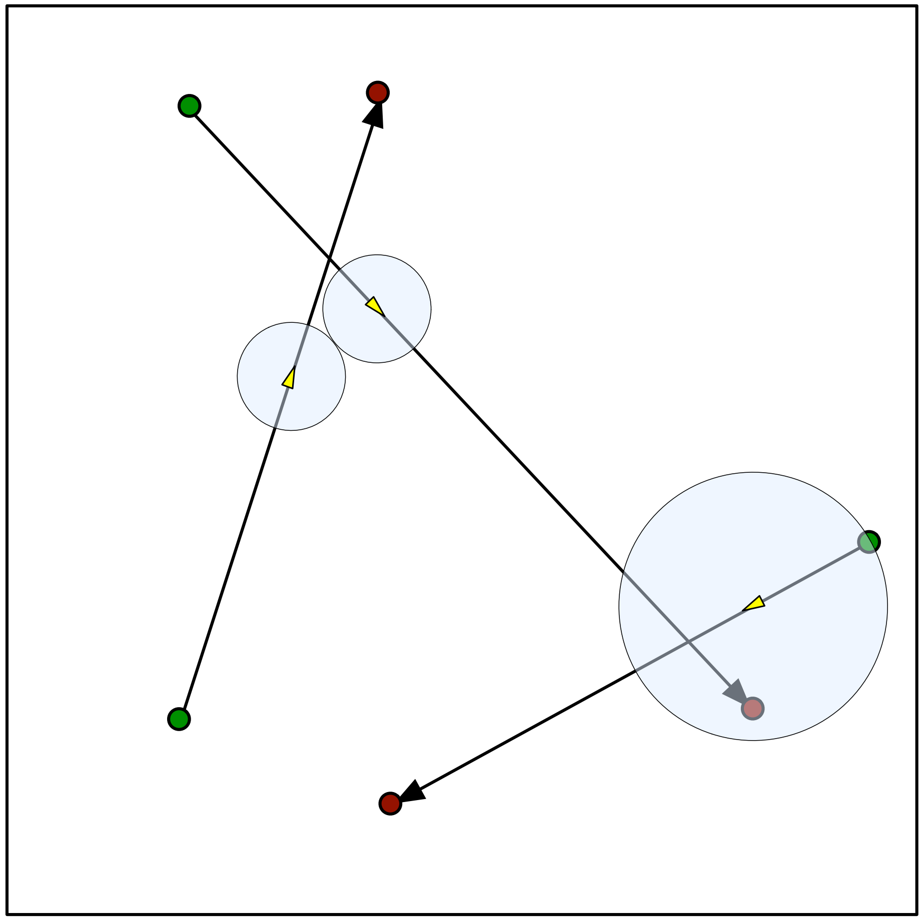
Traffic congestion analysis

- As *n* increases, how do safety (collision-avoidance) constraints affect the performance of the system?
  - Traffic throughput
  - Transportation efficiency
  - Quality of service in point-to-point tasks
- We need a **precise characterization** of how much the traffic volume affects the achievable performance:
  - Benchmark for traffic control infrastructure/protocol/algorithms.
  - Quantitative analysis and design of large-scale robotic systems.
- Formally determine **fundamental limitations** on performance of large-scale, decentralized, mobile systems.

Sensor-based Vehicle Routing

- Consider a compact, path-connected set $\mathcal{Q}$ in the plane.
- Let us consider n mobile agents, each assigned a pair of Source-Destination *(SD)* points in $\mathcal{Q}$.
- Each agent can move in any direction with bounded speed.
- Velocity-dependent **exclusion region**:

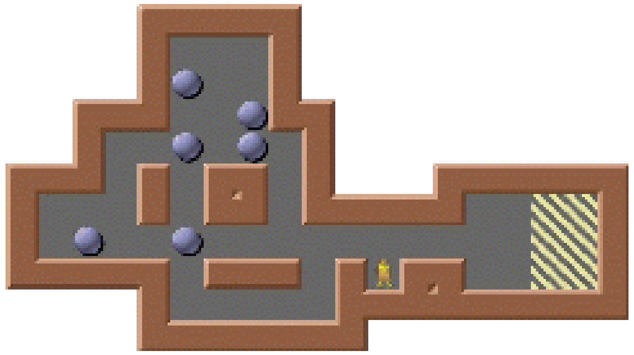
$$C_i(t) = \{z \in \mathbb{R}^2 : |z - x_i(t)| \leq r(n) + \kappa |v_i(t)|\}$$



What is the minimum time needed to transfer all agents from their sources to the respective destination, maintaining openly disjoint exclusion regions?

(Some) Related work

- Classic work in robotics:
  - "Piano movers' problem" [Reif '79, Schwarz & Sharir '83]
  - "Warehousemen's problem" [Schwarz & Sharir '83]
  - PSPACE-hard problems



- Solution techniques
  - Canny's algorithm
  - Path coordination (e.g., Laumond et al., 02, Akella et al '02)
  - Roadmaps (e.g., LaValle '98 -- present)
  - Pareto optima (LaValle, Ghrist et al., '04)
- Beyond computational complexity
  - Comm. and time complexity of robotic tasks (Klavins '02, Martinez et. al, '05)
- Physics/Operations Research/Air Traffic /Wireless networks
  - Analysis of transportation networks, air traffic control algorithms (Tomlin et al. '03, Feron et al. '98---), Capacity of wireless networks (Gupta & Kumar, 2000).
  - "Fundamental traffic law" (on a **ring**)

A simplifying "trick"

- Deciding the feasibility of the warehousemen's problem is extremely hard.
- We make the following assumption:
  - A vehicle enters the environment, at its source point, only upon **activation**
  - Vehicles are removed from the environment (**deactivated**) as soon as they reach their destination.
- Justification:
  - Airports, hangars, parking facilities, etc. are "**safe havens**" for mobile agents.
- Benefit:
  - All vehicle routing problems are feasible.

Preliminary results

- **Proposition:** The minimum transfer time T^* is $O(n)$
  - That is, we can always activate and deliver agents in turn
- **Proposition:** If $r(n) \geq r_0 > 0$ the minimum transfer time T^* is $\Omega(n)$
  - If the size of the excl. region is finite, there is a limit to the constant number of agents active at the same time.
- An intuitive constraint:
  - Assume $r(n) = O(1/\sqrt{n})$
  - Analyze the **interplay between traffic volume and achievable average velocity.**
- Will show that as long as the *m* vehicles physically fit in the environment, their **physical dimensions are not consequential** in determining congestion!
  - Air traffic control: exclusion region (5 nm) >> aircraft dimensions
  - Automotive traffic: “comfort” buffer from a lead vehicle approx. 2 s, approx. 30 meters

ARES
Aerospace Robotics and Embedded Systems Laboratory

Lower bound - Arbitrary case

- **Lemma:** The minimum time to transfer m agents is $\Omega(L\sqrt{n})$, where L is the average distance between origin and destination points.
  - Break down the path followed by the agents in a sequence of straight-line segments traversed at constant speed on a time schedule of length h. Each time interval has duration τ. Clearly,

$$\sum_{i=1}^{n}\sum_{j=1}^{h} r_i^j \geq nL$$

  - The velocity of the agents must be such that the sum of the areas of all exclusion regions is no more than:

$$\sum_{i=1}^{n} A_i^j = \frac{\kappa(\pi\kappa + 2\tau)}{\tau^2}\sum_{i=1}^{n}(r_i^j)^2 \leq 4$$

  - Applying Jensen's inequality, and with a little more algebra, we get

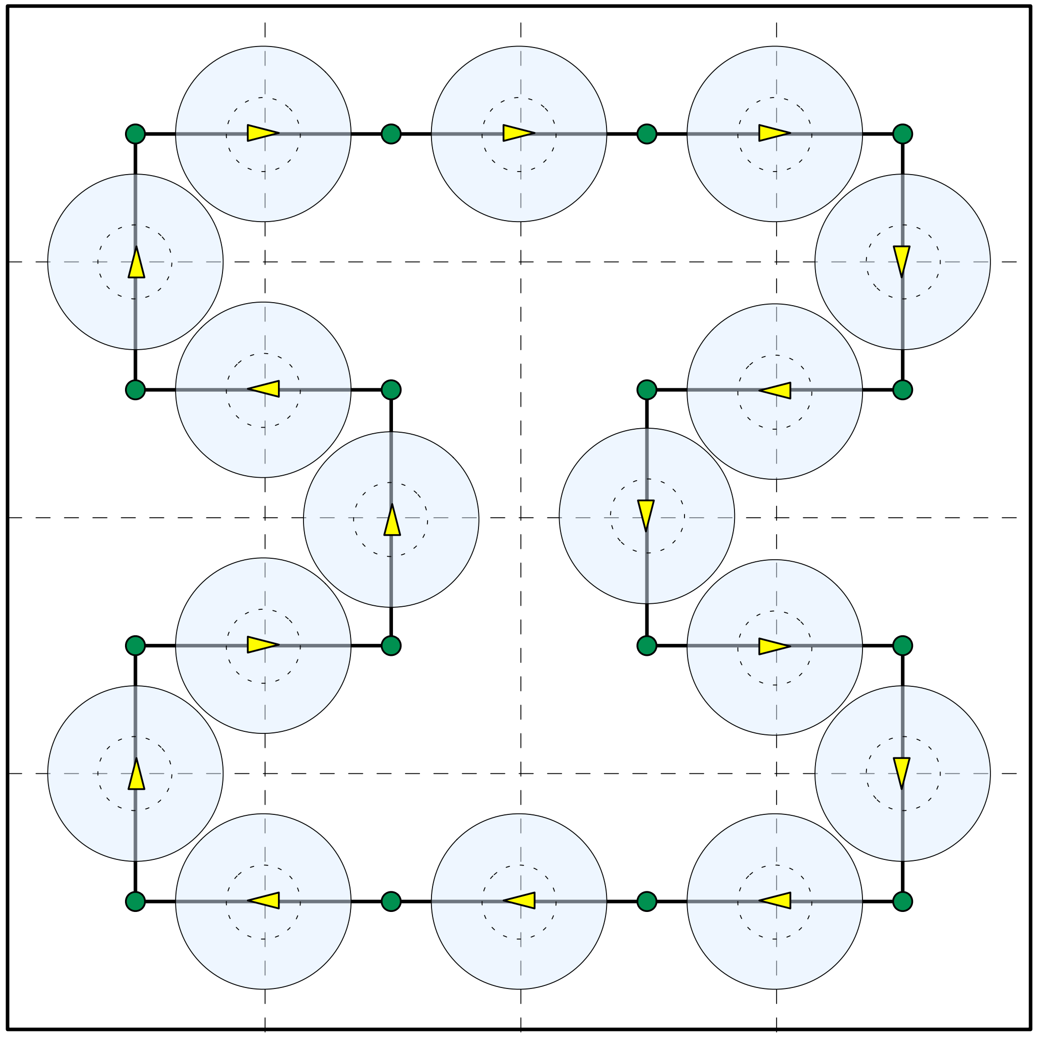
$$T^* = h\tau \geq \frac{L}{2}\sqrt{k(\pi k + 2\tau)n}$$

Upper bound ("best case")

- **Lemma:** There exists a selection of source-destination pairs such that

$$T^* = cL\sqrt{n}$$

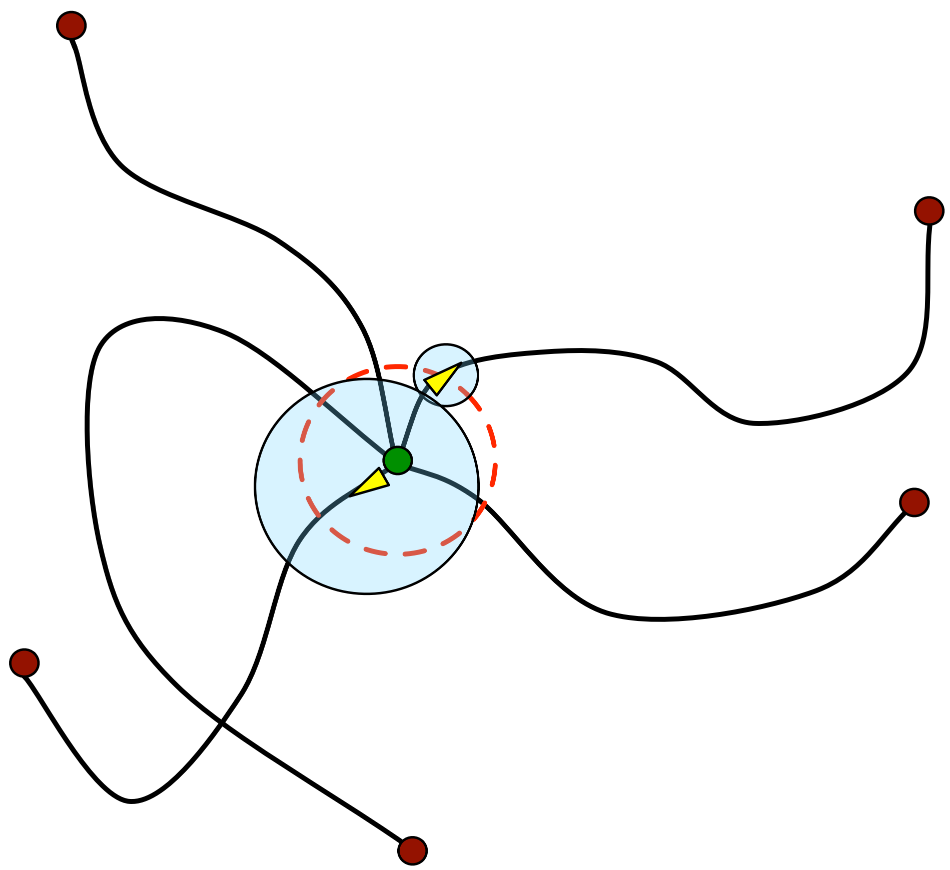
Theorem: The minimum time to transfer n agents between the respective source and destination points is $\Theta(L\sqrt{n})$



In other words, the average velocity decreases at least as $1/\sqrt{n}$

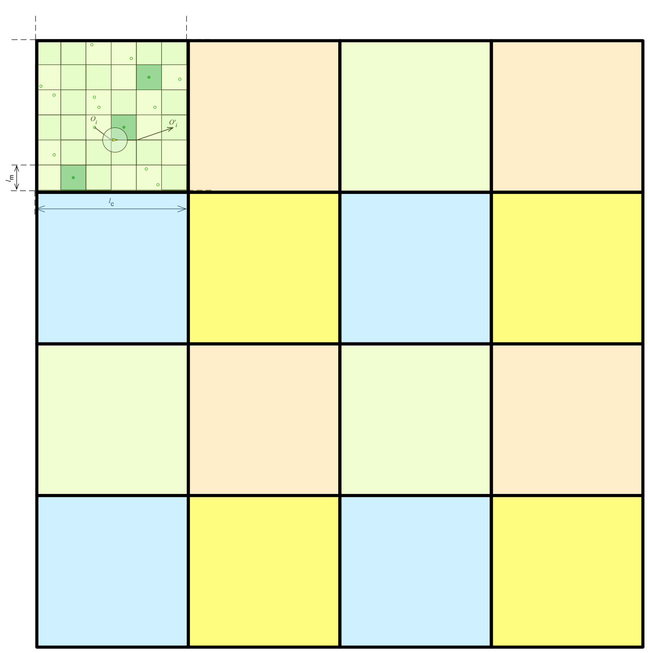
Worst case

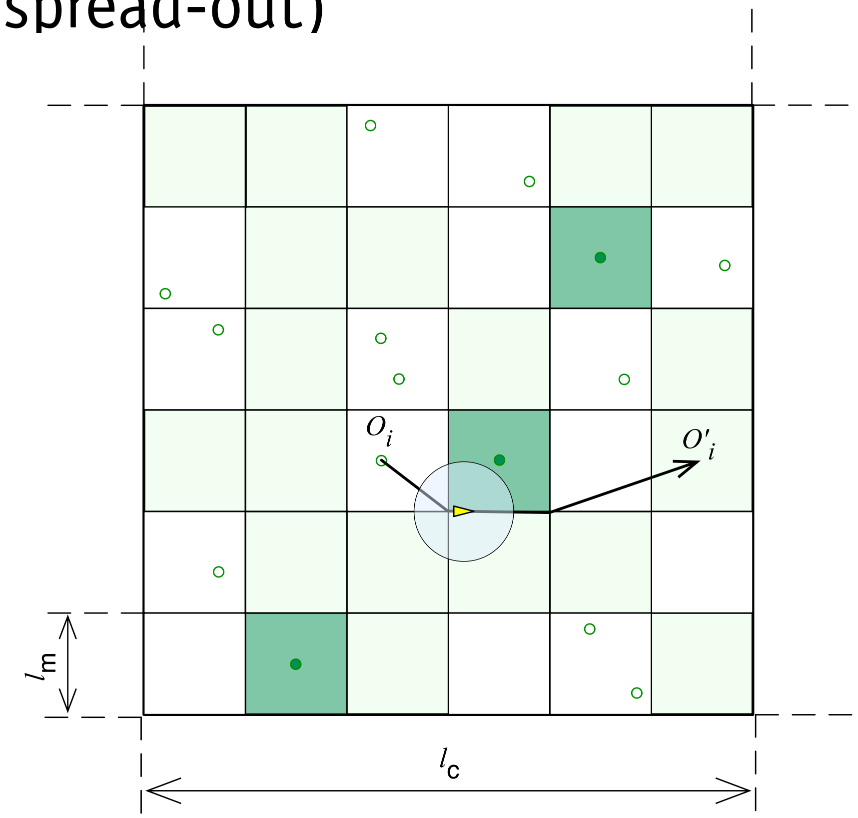
- In addition to the "best" case, there are "**worst**"-case choices:
  - Choose source-destination pairs in such a way that all sources are colocated at the same point S.
  - Activation times are in a strictly increasing sequence.
  - The time needed for the last agent to exit a disk of radius d centered at S is Ω(d n); the time complexity of the problem is then Ω(n).

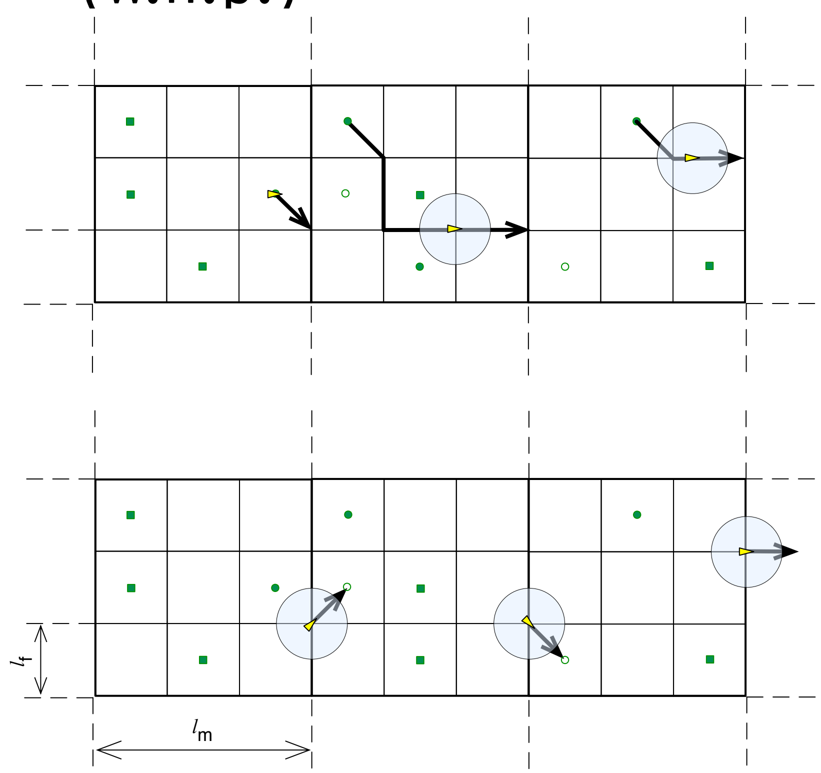


Random (S,D) pairs

- Three phases:
  - Initialization (spread-out): $O((\log n)^{2/3})$ (w.h.p.)
    - Uniform convergence in the weak law of large numbers.
  - Main phase (mesh routing) $O(\sqrt{n})$
    - Permutation routing with small queues.
  - Termination (reverse spread-out) $O((\log n)^{2/3})$ (w.h.p.)

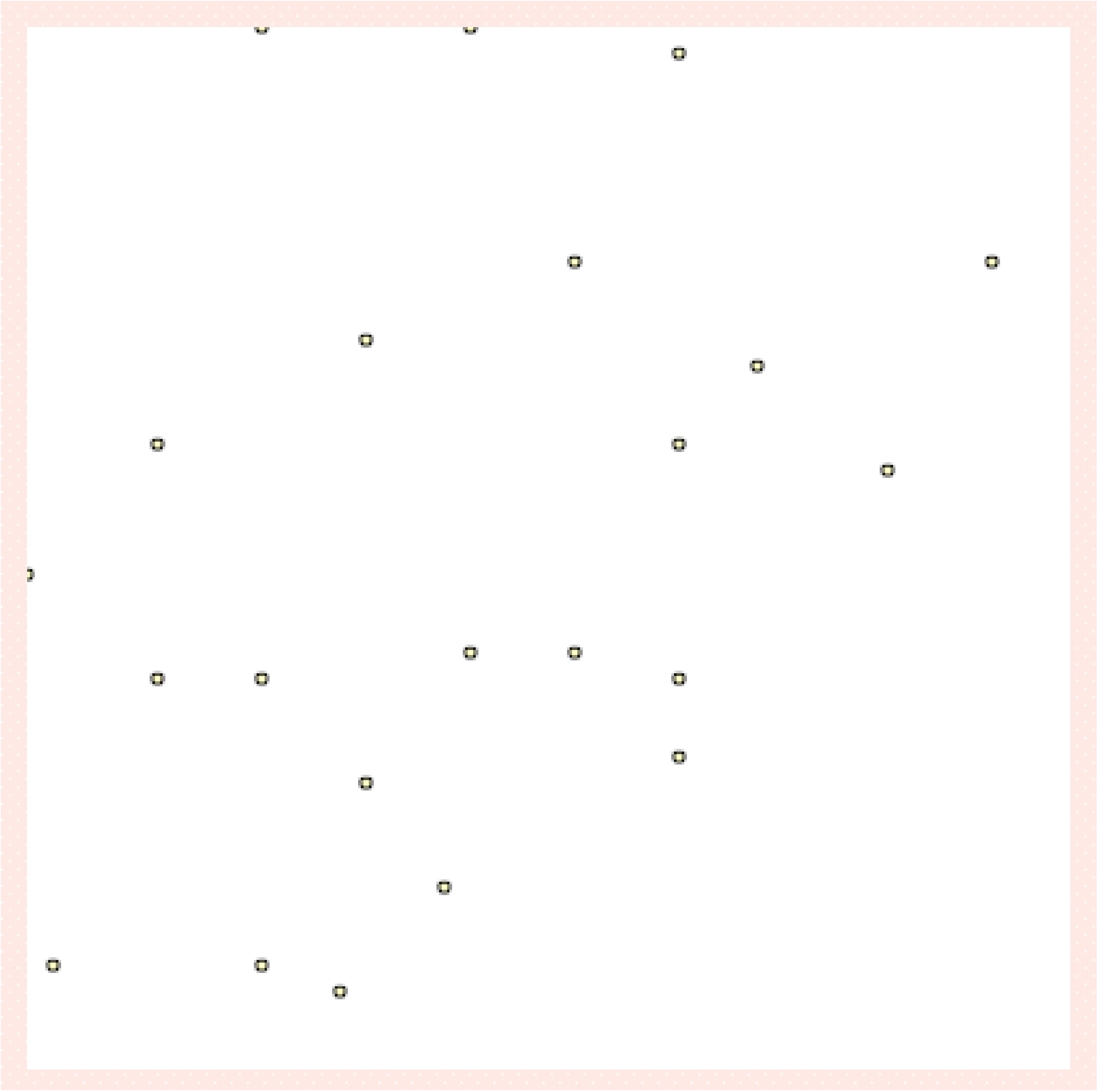






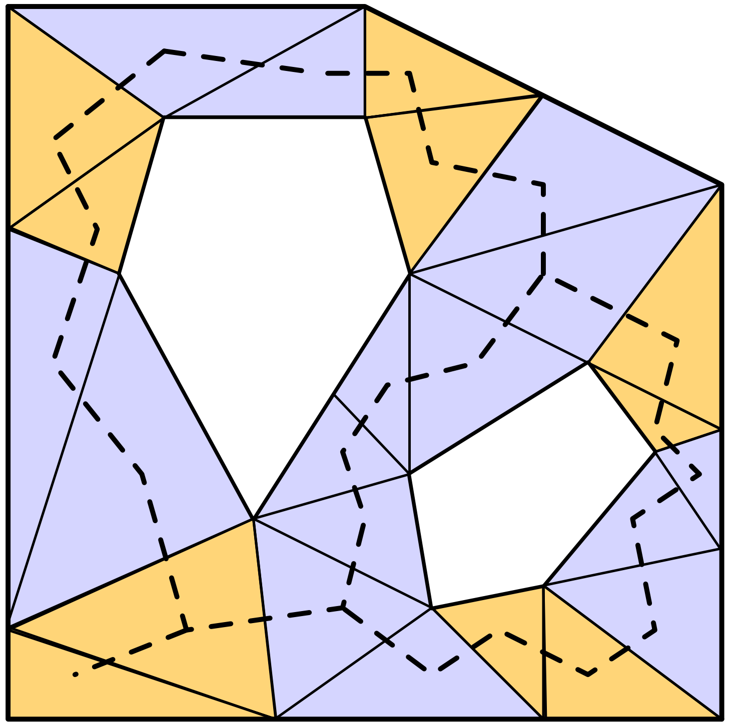
Aerospace Robotics and Embedded Systems Laboratory

Simulation Example - 100 agents



Generalizations

- Similar results hold in the following cases:
  - Convex environments.
  - Absolutely continuous prob. distributions (wrt area)
    - Singular probability distributions require sequential activation.
  - Non-convex, path-connected environments
    - Simple polygons.
    - Simple polygons with holes.



- The time complexity of sensor-based vehicle routing scales as the

The average velocity of individual agents, in any sensor-based vehicle routing problem, decreases as the inverse square root of the number of agents.

Decentralized path coverage

- Consider *m* agents, each carrying a sensor with a circular footprint of radius δ
- It is desired that the agents follow paths in such a way that the union of their footprints over time covers a region of interest.
  - Minefield clearing
  - Search and Rescue
  - Autonomous vacuuming/painting
- Desired features:
  - Simple robots: minimal sensing and comm. abilities, no GPS, no “pheromones.”
  - Efficiency and robustness to navigation errors and individual failures

A variation on cyclic pursuit

- In cyclic pursuit, agents are arranged on a ring; each agent moves towards the agent immediately preceding it [Marshall, Broucke, Francis 2004-'05; Paley, Leonard, Sepulchre '05].
- Cyclic pursuit with an offset angle α, i.e.,

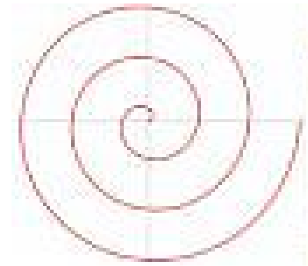
$$\dot{x}_i = R(\alpha_i)(x_{i+1} - x_i)$$

- If all α_i are equal, then
  - if $\alpha = \pi/n$: Robots converge to a regular configuration on a circle, i.e., a simple regular polygon (the radius depends on initial conditions)
  - if $\alpha < \pi/n$: Robots converge to a point (the center of mass)
  - if $\alpha > \pi/n$: Robots converge to a regular polygonal configuration, but they describe diverging logarithmic spirals.
- Proofs based on spectral analysis of certain circulant matrices.

Path coverage by Archimede's spirals

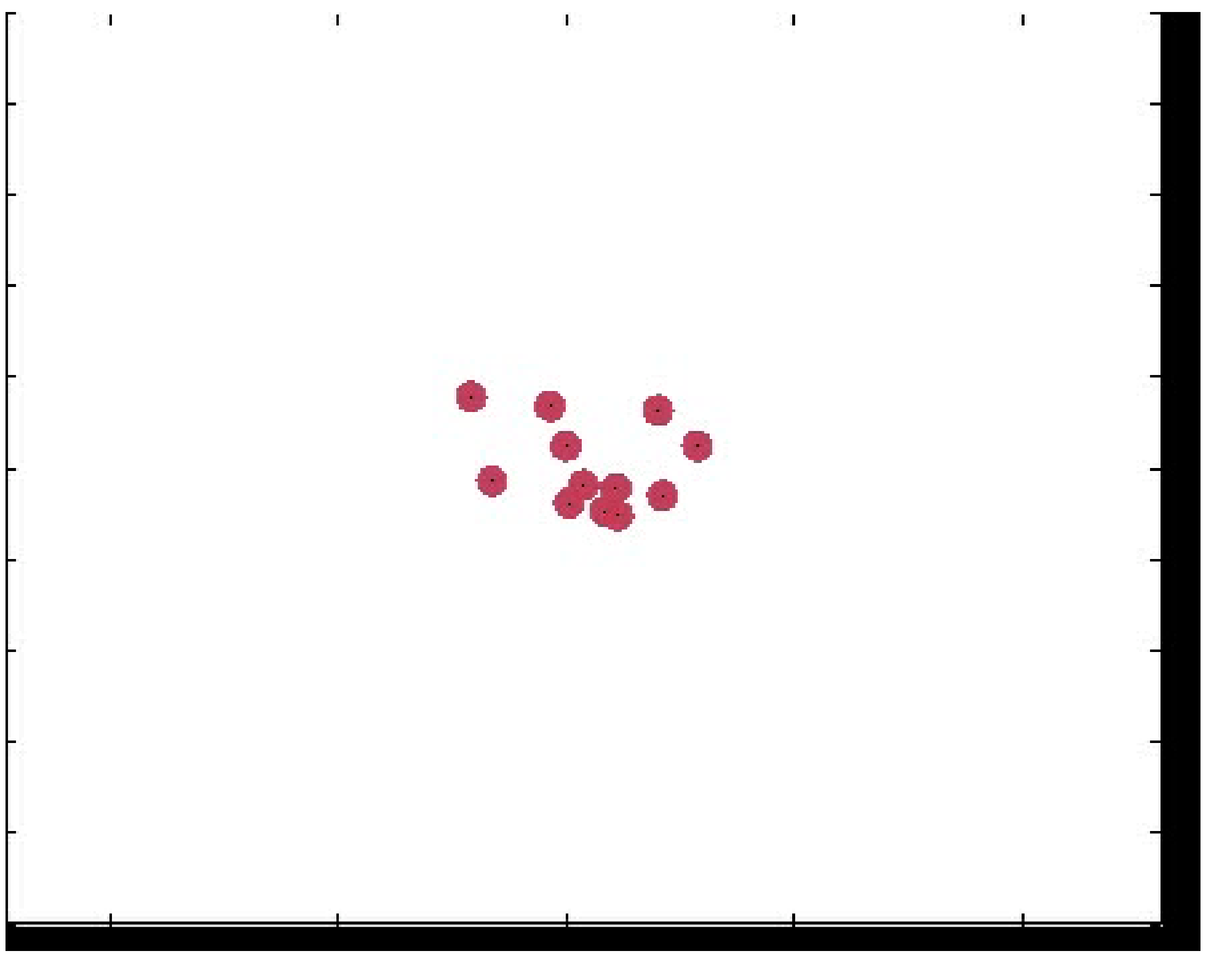
- A curve that is not found in nature, but was known by the ancient Greeks.

$$\rho(\theta) = \frac{n\delta}{2\pi}\,\theta$$



- The following selection of the offset angle provides locally ``stable'' tracking of an Archimede's spiral providing path coverage:

$$\alpha_i = \frac{\pi}{n} + \arctan\left(\frac{\delta}{\|x_{i+1} - x_i\|}\,\frac{\sin(\pi/n)}{\pi/n}\right)$$



- Local estimate of π/n computed via consensus
- Time needed for coverage is inversely proportional to the number of agents.

Conclusions

- The **algorithmic analysis of mobile robotic networks** requires new tools combining **combinatorial optimization**, **optimal control**, **differential geometry**, **systems theory**, **probability** and **stochastic systems**.
- Designed algorithms for the solution of foraging/area services with **provable performance**, for important classes of vehicles with non-holonomic constraints.
  - Multiple-vehicle cooperation can **greatly enhance performance**, esp. in the heavy-load, non-holonomic case.
  - (Individual/team) There are natural incentives on the formation of teams (or “swarms”) sharing the same region of interest, depending on the **scale of the network** and and **on the agents’ dynamics**.
- Derived sharp bounds on the average velocity of independent agents under collision-avoidance constraints:
  - Traffic congestion can severely decrease the efficiency of large-scale systems, by reducing the average motion speed and traffic throughput.
  - Look for **trade-offs between teamwork and congestion** in order to figure out the