Dynamic Vehicle Routing for Robotic Networks
Lecture #7: Vehicle Models

Francesco Bullo[1] Emilio Frazzoli[2] Marco Pavone[2]
Ketan Savla[2] Stephen L. Smith[2]

[1]CCDC
University of California, Santa Barbara
bullo@engineering.ucsb.edu

[2]LIDS and CSAIL
Massachusetts Institute of Technology
{frazzoli,pavone,ksavla,slsmith}@mit.edu

Workshop at the 2010 American Control Conference
Baltimore, Maryland, USA, June 29, 2010, 8:30am to 5:00pm

Outline of the lecture

Vehicle routing with differential constraints

- What happens if the vehicles are subject to non-integrable differential constraints on their motion?
  - Minimum turn radius, constant speed (UAVs, Dubins cars)
  - Minimum turn radius, able to reverse (Reeds-Shepps cars)
  - Differential drive robots (e.g., tanks).
  - Bounded acceleration vehicles (e.g., helicopters, spacecraft).

- Fundamentally different problems, combining combinatorial task specifications with differential geometry and optimal control.

- Decompose the problem, study the asymptotic cases:
  - Heavy load: Traveling salesperson problems.
  - Light load: optimal loitering "stations".

Models of vehicles with differential constraints

Dubins vehicle

$$\dot{x} = \cos\theta,\quad \dot{y} = \sin\theta,\quad \dot{\theta} = \omega$$
$$|\omega| \leq 1/\rho$$

Reeds-Shepp car

$$\dot{x} = v\cos\theta,\quad \dot{y} = v\sin\theta,\quad \dot{\theta} = \omega$$
$$v \in \{-1, 1\}$$
$$|\omega| \leq 1/\rho$$

Differential drive

$$\dot{x} = \tfrac{1}{2}(\omega_l + \omega_r)\cos\theta,\quad \dot{y} = \tfrac{1}{2}(\omega_l + \omega_r)\sin\theta,\quad \dot{\theta} = \tfrac{1}{\rho}(\omega_r - \omega_l)$$
$$|\omega_l| \leq 1;\ |\omega_r| \leq 1$$

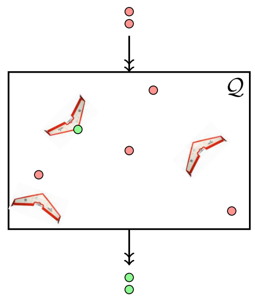
Double integrator

$$\ddot{x} = u$$
$$\|\dot{x}\| \leq 1$$
$$\|u\| \leq 1$$

DTRP formulation

Problem setup

- m identical vehicles in $\mathcal{Q}$
- Spatio-temporal Poisson process: rate λ and uniform spatial density
- On-site service time $s = 0$



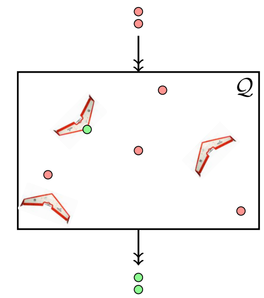
Objective

- Control policy $\pi = \{$task assignment, scheduling, loitering$\}$
- $T_\pi := \limsup_{i\to\infty} \mathbb{E}[\text{wait time of task } i]$; $\overline{T}^* = \inf_\pi T_\pi$
- Design π for which T_π is equal to or within a constant factor of $\overline{T}^*$

DTRP formulation

Problem setup

- m identical vehicles in $\mathcal{Q}$
- Spatio-temporal Poisson process: rate λ and uniform spatial density
- On-site service time $s = 0$



Objective

- Control policy $\pi = \{$task assignment, scheduling, loitering$\}$
- $T_\pi := \limsup_{i\to\infty} \mathbb{E}[\text{wait time of task } i]$; $\overline{T}^* = \inf_\pi T_\pi$
- Design π for which T_π is equal to or within a constant factor of $\overline{T}^*$

Stabilizability

- $$\underbrace{\lambda}_{\text{task generation rate}} - \underbrace{m \cdot \frac{n}{\text{TSPlength}(n)}}_{\text{task service rate}} = \text{task growth rate}$$

  n: # outstanding tasks
- TSPlength(n) strictly sub-linear $\implies$ stability $\forall \lambda, m$
- Euclidean TSPlength(n) $= \Theta(n^{1/2})$ (Beardwood et. al. '59)
- Euclidean TSP based path planning heuristic $\implies O(n)$
- Traveling salesperson problems for differential vehicles.

Stabilizability

- $$\underbrace{\lambda}_{\text{task generation rate}} - \underbrace{m \cdot \frac{n}{\text{TSPlength}(n)}}_{\text{task service rate}} = \text{task growth rate}$$

  n: # outstanding tasks
- TSPlength(n) strictly sub-linear $\implies$ stability $\forall \lambda, m$
- Euclidean TSPlength(n) $= \Theta(n^{1/2})$ (Beardwood et. al. '59)
- Euclidean TSP based path planning heuristic $\implies O(n)$
- Traveling salesperson problems for differential vehicles.

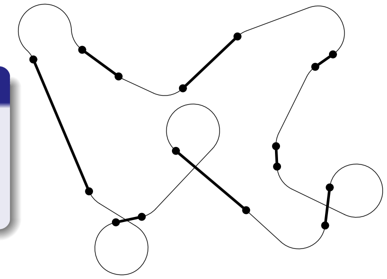
Outline of the lecture

1. Models of vehicles with differential constraints
2. Traveling salesperson problems
3. The heavy load case
4. The light load case
5. Phase transition in the light load

Traveling Salesperson Problem

Problem Statement

Find the shortest closed curve feasible for the vehicle through a given finite set of points in the plane

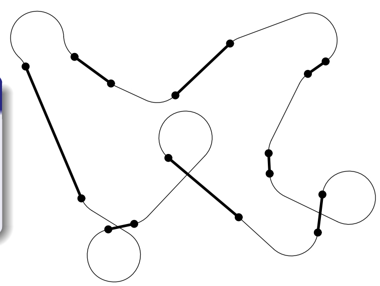


- NP-hardness a consequence of the NP-hardness of the Euclidean TSP.
- Does the cost of this TSP increase SUBLINEARLY with n?
- Is there a polynomial-time algorithm that returns a tour of length $o(n)$??
- What is the quality of the solution?

Traveling Salesperson Problem

Problem Statement

Find the shortest closed curve feasible for the vehicle through a given finite set of points in the plane



- NP-hardness a consequence of the NP-hardness of the Euclidean TSP.
- Does the cost of this TSP increase SUBLINEARLY with n?
- Is there a polynomial-time algorithm that returns a tour of length $o(n)$??
- What is the quality of the solution?

Literature review

1. K. Savla, E. Frazzoli, and F. Bullo. Traveling Salesperson Problems for the Dubins vehicle. *IEEE Transactions on Automatic Control*, 53(6):1378–1391, 2008
2. K. Savla, F. Bullo, and E. Frazzoli. Traveling Salesperson Problems for a double integrator. *IEEE Transactions on Automatic Control*, 54(4):788–793, 2009
3. J. J. Enright, K. Savla, E. Frazzoli, and F. Bullo. Stochastic and dynamic routing problems for multiple UAVs. *AIAA Journal of Guidance, Control, and Dynamics*, 34(4):1152–1166, 2009
4. J. J. Enright and E. Frazzoli. The stochastic Traveling Salesman Problem for the Reeds-Shepp car and the differential drive robot. In *IEEE Conf. on Decision and Control*, pages 3058–3064, San Diego, CA, December 2006
5. K. Savla and E. Frazzoli. On endogenous reconfiguration for mobile robotic networks. In *Workshop on Algorithmic Foundations of Robotics*, Guanajuato, Mexico, December 2008
6. M. Pavone, K. Savla, and E. Frazzoli. Sharing the load. *IEEE Robotics and Automation Magazine*, 16(2):52–61, 2009
7. F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. L. Smith. Dynamic vehicle routing for robotic systems. *Proceedings of the IEEE*, May 2010. Submitted
8. S. Rathinam, R. Sengupta, and S. Darbha. A resource allocation algorithm for multi-vehicle systems with non holonomic constraints. *IEEE Transactions on Automation Sciences and Engineering*, 4(1):98–104, 2007
9. J. Le Ny, E. Feron, and E. Frazzoli. On the curvature-constrained traveling salesman problem. *IEEE Transactions on Automatic Control*, 2009. to appear
10. S. Itani. *Dynamic Systems and Subadditive Functionals*. PhD thesis, Massachusetts Institute of Technology, 2009

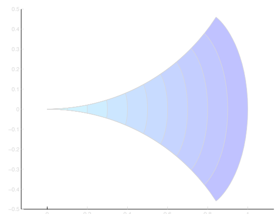
Stochastic TSP: A nearest-neighbor lower bound

Outline of the calculations

- Calculate (an upper bound on) expected distance from an arbitrary vehicle configuration to closest point, δ^*
  - Calculate (an upper bound on) the area of the set reachable with a path of length δ, $\mathcal{R}_\delta$.
  - $\Pr(\delta^* \geq \delta) \geq \max\{0, 1 - n|\mathcal{R}_\delta|/|\mathcal{Q}|\}$
- Expected length of the tour cannot be less than n times $\mathbb{E}[\delta^*]$

Example: Dubins vehicle

- $|\mathcal{R}_\delta| = \frac{\delta^3}{3\rho}$
- $\mathbb{E}[\delta^*] = \frac{3}{4}\left(\frac{3\rho|\mathcal{Q}|}{n}\right)^{1/3}$.
- $\lim_{n\to\infty} \frac{\mathbb{E}[\mathrm{TSP}(n)]}{n^{2/3}} \geq \frac{3}{4}(3\rho|\mathcal{Q}|)^{1/3}$.



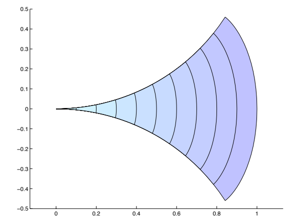
Stochastic TSP: A nearest-neighbor lower bound

Outline of the calculations

- Calculate (an upper bound on) expected distance from an arbitrary vehicle configuration to closest point, δ^*
  - Calculate (an upper bound on) the area of the set reachable with a path of length δ, $\mathcal{R}_\delta$.
  - $\Pr(\delta^* \geq \delta) \geq \max\{0, 1 - n|\mathcal{R}_\delta|/|\mathcal{Q}|\}$
- Expected length of the tour cannot be less than n times $\mathbb{E}[\delta^*]$

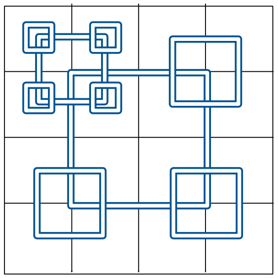
Example: Dubins vehicle

- $|\mathcal{R}_\delta| = \frac{\delta^3}{3\rho}$
- $\mathbb{E}[\delta^*] = \frac{3}{4}\left(\frac{3\rho|\mathcal{Q}|}{n}\right)^{1/3}$.
- $\lim_{n\to\infty} \frac{\mathbb{E}[\mathrm{TSP}(n)]}{n^{2/3}} \geq \frac{3}{4}(3\rho|\mathcal{Q}|)^{1/3}$.



Towards an upper bound: tiling based algorithms

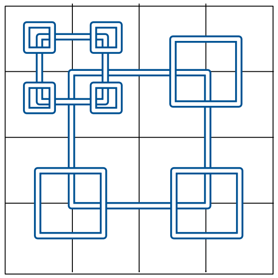
- The way the ETSP tours are constructed relies on the scaling properties of tours: the length of the tour scales as the coordinates of the points.



- No such scaling exists for the TSP for vehicles with differential constraints, e.g., the bound on the curvature for the Dubins vehicle does not scale with the coordinates of the points!
- Any tiling-based algorithm must account for a "preferential direction", e.g., by penalizing turning for Dubins vehicles

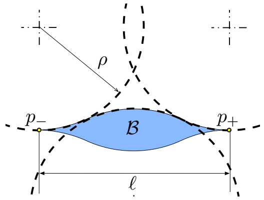
Towards an upper bound: tiling based algorithms

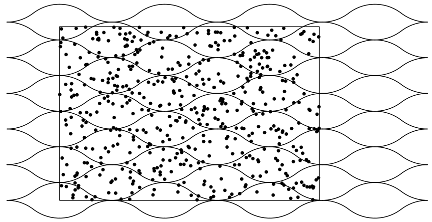
- The way the ETSP tours are constructed relies on the scaling properties of tours: the length of the tour scales as the coordinates of the points.



- No such scaling exists for the TSP for vehicles with differential constraints, e.g., the bound on the curvature for the Dubins vehicle does not scale with the coordinates of the points!
- Any tiling-based algorithm must account for a "preferential direction", e.g., by penalizing turning for Dubins vehicles

Bead construction

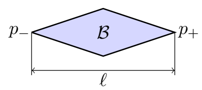




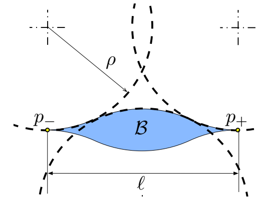
Bead properties

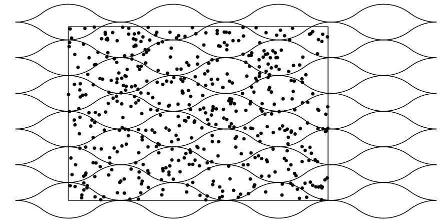
- Length$(p_-, q, p_+) \leq \ell + o(\ell^2)$ for all $q \in \mathcal{B}$
- Width: $w(\ell) = \frac{\ell^2}{8\rho} + o(\ell^3)$
- The beads tile the plane
- Useful for Dubins vehicle, Reeds-Shepp car and double integrator

- Diamond-like cell for differential drive



Bead construction

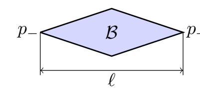




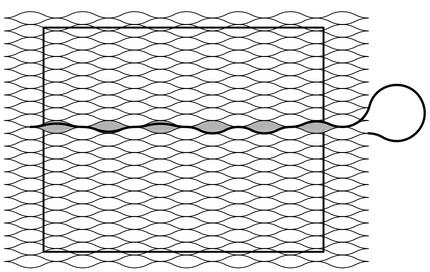
Bead properties

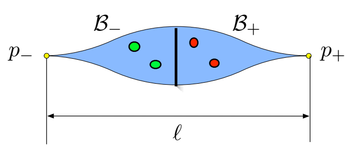
- Length$(p_-, q, p_+) \leq \ell + o(\ell^2)$ for all $q \in \mathcal{B}$
- Width: $w(\ell) = \frac{\ell^2}{8\rho} + o(\ell^3)$
- The beads tile the plane
- Useful for Dubins vehicle, Reeds-Shepp car and double integrator

- Diamond-like cell for differential drive



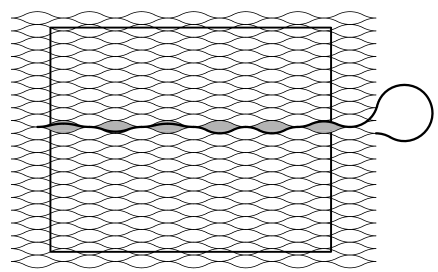
The single-sweep tiling algorithm

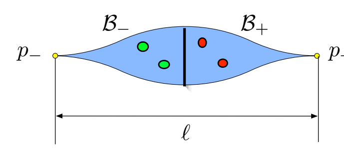




- Tile the region with beads
- Sweep the bead rows, while servicing all the targets in every bead as follows:
  - Service every task q in $\mathcal{B}_-$ using the "$p_- \to q \to p_-$" protocol
  - Move from p_- to p_+
  - Service every task q in $\mathcal{B}_+$ using the "$p_+ \to q \to p_+$" protocol

The single-sweep tiling algorithm



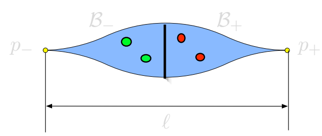


- Tile the region with beads
- Sweep the bead rows, while servicing all the targets in every bead as follows:
  - Service every task q in $\mathcal{B}_-$ using the "$p_- \to q \to p_-$" protocol
  - Move from p_- to p_+
  - Service every task q in $\mathcal{B}_+$ using the "$p_+ \to q \to p_+$" protocol

Analysis of the single-sweep tiling algorithm

Path length calculations

$TSP(n) =$ (bead row length + move to next bead row) $\times$ # bead rows + move to service each task $\times$ # tasks + tour closure length



- For a Reeds-Shepp car, as $\ell \to 0$:

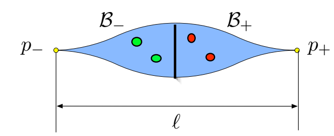
$$TSP(n) \leq \left(\sqrt{|\mathcal{Q}|} + \ell/2\right)\frac{\sqrt{|\mathcal{Q}|}}{w(\ell)/2} + \ell n + 2\left(\sqrt{|\mathcal{Q}|} + \rho\pi\right)$$
$$\leq 16\rho\frac{|\mathcal{Q}|}{\ell^2} + 8\rho\frac{\sqrt{|\mathcal{Q}|}}{\ell} + \ell n + 2\left(\sqrt{|\mathcal{Q}|} + \rho\pi\right) \quad \left(\because w(\ell) \approx \frac{\ell^2}{8\rho}\right)$$

- Pick $\ell = \left(\frac{32\rho|\mathcal{Q}|}{n}\right)^{1/3}$ $\left(\text{i.e., } \frac{|\mathcal{B}|}{|\mathcal{Q}|} = \frac{2}{n}\right) \implies TSP(n) = O(n^{2/3})$.

Analysis of the single-sweep tiling algorithm

Path length calculations

$TSP(n) =$ (bead row length + move to next bead row) $\times$ # bead rows + move to service each task $\times$ # tasks + tour closure length



- For a Reeds-Shepp car, as $\ell \to 0$:

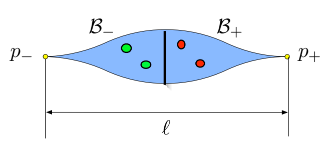
$$TSP(n) \leq \left(\sqrt{|\mathcal{Q}|} + \ell/2\right)\frac{\sqrt{|\mathcal{Q}|}}{w(\ell)/2} + \ell n + 2\left(\sqrt{|\mathcal{Q}|} + \rho\pi\right)$$
$$\leq 16\rho\frac{|\mathcal{Q}|}{\ell^2} + 8\rho\frac{\sqrt{|\mathcal{Q}|}}{\ell} + \ell n + 2\left(\sqrt{|\mathcal{Q}|} + \rho\pi\right) \quad \left(\because w(\ell) \approx \frac{\ell^2}{8\rho}\right)$$

- Pick $\ell = \left(\frac{32\rho|\mathcal{Q}|}{n}\right)^{1/3}$ $\left(\text{i.e., } \frac{|\mathcal{B}|}{|\mathcal{Q}|} = \frac{2}{n}\right) \implies TSP(n) = O(n^{2/3})$.

Analysis of the single-sweep tiling algorithm

Path length calculations

$TSP(n) =$ (bead row length + move to next bead row) $\times$ # bead rows + move to service each task $\times$ # tasks + tour closure length



- For a Reeds-Shepp car, as $\ell \to 0$:

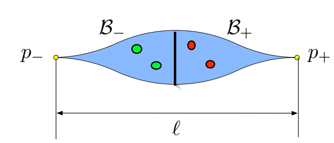
$$TSP(n) \leq \left(\sqrt{|\mathcal{Q}|} + \ell/2\right)\frac{\sqrt{|\mathcal{Q}|}}{w(\ell)/2} + \ell n + 2\left(\sqrt{|\mathcal{Q}|} + \rho\pi\right)$$
$$\leq 16\rho\frac{|\mathcal{Q}|}{\ell^2} + 8\rho\frac{\sqrt{|\mathcal{Q}|}}{\ell} + \ell n + 2\left(\sqrt{|\mathcal{Q}|} + \rho\pi\right) \quad \left(\because w(\ell) \approx \frac{\ell^2}{8\rho}\right)$$

- Pick $\ell = \left(\frac{32\rho|\mathcal{Q}|}{n}\right)^{1/3}$ $\left(\text{i.e., } \frac{|\mathcal{B}|}{|\mathcal{Q}|} = \frac{2}{n}\right) \implies TSP(n) = O(n^{2/3})$.

Analysis of the single-sweep tiling algorithm

Path length calculations

$TSP(n) =$ (bead row length + move to next bead row) $\times$ # bead rows + move to service each task $\times$ # tasks + tour closure length



- For a Dubins vehicle, as $\ell \to 0$:

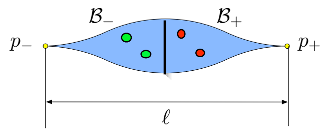
$$TSP(n) \leq \left(\sqrt{|\mathcal{Q}|} + \frac{w(\ell)}{2} + \kappa\right)\frac{\sqrt{|\mathcal{Q}|}}{w(\ell)/2} + (\ell + \kappa)n + 2\sqrt{|\mathcal{Q}|} + \kappa$$
$$\leq 16\rho\frac{|\mathcal{Q}|}{\ell^2} + \sqrt{|\mathcal{Q}|} + 16\kappa\frac{\sqrt{|\mathcal{Q}|}}{\ell^2} + \ell n + \underline{\kappa \mathbf{n}} + 2\sqrt{|\mathcal{Q}|} + \kappa$$

- The κn term grows linearly in n for all $\ell \implies TSP(n) = O(n)$

Analysis of the single-sweep tiling algorithm

Path length calculations

$TSP(n) =$ (bead row length + move to next bead row) $\times$ # bead rows + move to service each task $\times$ # tasks + tour closure length



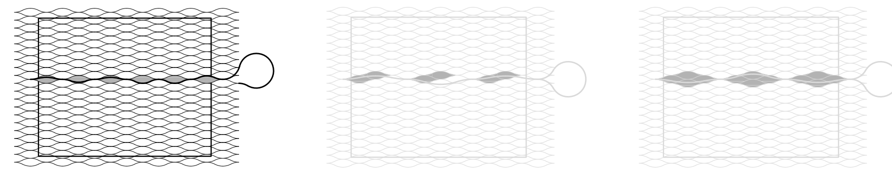
- For a Dubins vehicle, as $\ell \to 0$:

$$TSP(n) \leq \left(\sqrt{|\mathcal{Q}|} + \frac{w(\ell)}{2} + \kappa\right) \frac{\sqrt{|\mathcal{Q}|}}{w(\ell)/2} + (\ell + \kappa)n + 2\sqrt{|\mathcal{Q}|} + \kappa$$
$$\leq 16\rho\frac{|\mathcal{Q}|}{\ell^2} + \sqrt{|\mathcal{Q}|} + 16\kappa\frac{\sqrt{|\mathcal{Q}|}}{\ell^2} + \ell n + \underline{\kappa \mathbf{n}} + 2\sqrt{|\mathcal{Q}|} + \kappa$$

- The κn term grows linearly in n for all ℓ $\implies$ $\text{TSP}(n) = O(n)$

The recursive sweep tiling algorithm

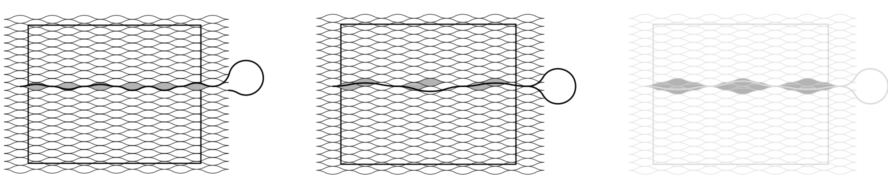
- Tile $\mathcal{Q}$ with beads such that: $\frac{|\mathcal{B}|}{|\mathcal{Q}|} = \frac{1}{2n}$ (i.e., $\ell \sim n^{-1/3}$)
- Sweep the bead rows, visiting one target per non-empty bead.
- Iterate, using at the i-th phase a "meta-bead" composed of 2^{i-1} beads.
- After $\log n$ phases, visit the outstanding targets in any arbitrary order, e.g., with a greedy strategy.



Phase 1 Phase 2 Phase 3

The recursive sweep tiling algorithm

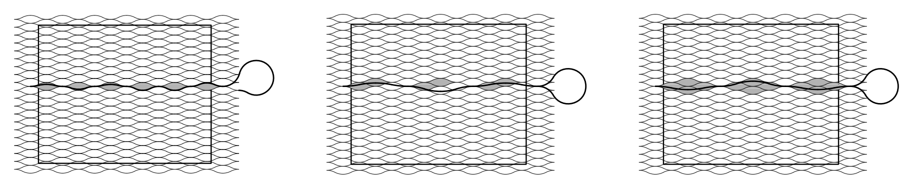
- Tile $\mathcal{Q}$ with beads such that: $\frac{|\mathcal{B}|}{|\mathcal{Q}|} = \frac{1}{2n}$ (i.e., $\ell \sim n^{-1/3}$)
- Sweep the bead rows, visiting one target per non-empty bead.
- Iterate, using at the i-th phase a "meta-bead" composed of 2^{i-1} beads.
- After $\log n$ phases, visit the outstanding targets in any arbitrary order, e.g., with a greedy strategy.



Phase 1 Phase 2 Phase 3

The recursive sweep tiling algorithm

- Tile $\mathcal{Q}$ with beads such that: $\frac{|\mathcal{B}|}{|\mathcal{Q}|} = \frac{1}{2n}$ (i.e., $\ell \sim n^{-1/3}$)
- 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} 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 recursive algorithm

- Theorem: For a Dubins vehicle, with probability one,

$$\limsup_{n\to\infty} \frac{\mathrm{TSP}(n)}{n^{2/3}} \leq 24\sqrt[3]{\rho|\mathcal{Q}|}\left(1+\frac{7}{3}\pi\frac{\rho}{\sqrt{|\mathcal{Q}|}}\right)$$

Outline of the proof

- $\Pr(\lim_{n\to\infty}$ # tasks remaining after phase $i^* > 24\log n) = 0$
- Path length calculations:
  - Phase 1 path length $O\left(\frac{1}{\ell^2}\right) = O\left(n^{2/3}\right)$ $\quad (\because \quad \ell \sim n^{-1/3})$
  - Subsequent phase path lengths are decreasing geometric series; path length for all i^* phases is $O\left(n^{2/3}\right)$
  - Path length by greedy heuristic is $O(\log n)$

Analysis of the recursive algorithm

- Theorem: For a Dubins vehicle, with probability one,

$$\limsup_{n\to\infty} \frac{\mathrm{TSP}(n)}{n^{2/3}} \leq 24\sqrt[3]{\rho|\mathcal{Q}|}\left(1+\frac{7}{3}\pi\frac{\rho}{\sqrt{|\mathcal{Q}|}}\right)$$

Outline of the proof

- $\Pr(\lim_{n\to\infty}$ # tasks remaining after phase $i^* > 24\log n) = 0$
- Path length calculations:
  - Phase 1 path length $O\left(\frac{1}{\ell^2}\right) = O\left(n^{2/3}\right)$ $\quad (\because \quad \ell \sim n^{-1/3})$
  - Subsequent phase path lengths are decreasing geometric series; path length for all i^* phases is $O\left(n^{2/3}\right)$
  - Path length by greedy heuristic is $O(\log n)$

FB, EF, MP, KS, SLS (UCSB, MIT) Dynamic Vehicle Routing (Lecture 7/8) 29jun10 @ Baltimore, ACC 17 / 36

Summary of TSPs

- Lower bound: $\mathbb{E}\left[\mathrm{TSP}(n)\right] \in \Omega(n^{2/3})$
- Upper bound: $\mathbb{E}\left[\mathrm{TSP}(n)\right] \in O(n^{2/3})$
- $TSP(n)$ is of order $n^{2/3}$; constant factor approximation algorithms
- Computational complexity of the algorithms is of order n

Stabilizability of the DTRP

- $$\underbrace{\lambda}_{\text{task generation rate}} - \underbrace{m\cdot\frac{n}{\mathrm{TSP}(n)}}_{\text{task service rate}} = \text{task growth rate}$$
  n: # outstanding tasks
- $\mathbb{E}[\mathrm{TSP}(n)] \in \Theta(n^{2/3}) \implies$ trivial receding horizon TSP-based policies are stable for the DTRP for all λ and m

Summary of TSPs

- Lower bound: $\mathbb{E}\left[\mathrm{TSP}(n)\right] \in \Omega(n^{2/3})$
- Upper bound: $\mathbb{E}\left[\mathrm{TSP}(n)\right] \in O(n^{2/3})$
- $TSP(n)$ is of order $n^{2/3}$; constant factor approximation algorithms
- Computational complexity of the algorithms is of order n

Stabilizability of the DTRP

- $$\underbrace{\lambda}_{\text{task generation rate}} - \underbrace{m\cdot\frac{n}{\mathrm{TSP}(n)}}_{\text{task service rate}} = \text{task growth rate}$$
  n: # outstanding tasks
- $\mathbb{E}[\mathrm{TSP}(n)] \in \Theta(n^{2/3}) \implies$ trivial receding horizon TSP-based policies are stable for the DTRP for all λ and m

FB, EF, MP, KS, SLS (UCSB, MIT) Dynamic Vehicle Routing (Lecture 7/8) 29jun10 @ Baltimore, ACC 18 / 36

Outline of the lecture

1. Models of vehicles with differential constraints
2. Traveling salesperson problems
3. The heavy load case
4. The light load case
5. Phase transition in the light load

The heavy load case: nearest neighbor lower bound

Outline of the calculations

- Let n_π be the number of outstanding tasks at steady-state under stable policy π
- Calculate (an upper bound on) expected distance from an arbitrary vehicle configuration to closest among n_π points, $\delta^*(n_\pi)$
- At steady-state: $\frac{\lambda}{m} = \frac{1}{\mathbb{E}[\delta^*(n_\pi)]}$
- Little's formula: $\lambda T_\pi = n_\pi$

Example: Dubins vehicle

- $\mathbb{E}[\delta^*(n_\pi)] = \frac{3}{4}\left(\frac{3\rho|\mathcal{Q}|}{n_\pi}\right)^{1/3}$
- Steady state+ Little's formula: $\frac{\lambda}{m} = \frac{4}{3}\left(\frac{\lambda T_\pi}{3\rho|\mathcal{Q}|}\right)^{1/3}$
- $\liminf_{\frac{\lambda}{m}\to+\infty} \overline{T}^* \frac{m^3}{\lambda^2} \geq \frac{81}{64}\rho|\mathcal{Q}|$

The heavy load case: nearest neighbor lower bound

Outline of the calculations

- Let n_π be the number of outstanding tasks at steady-state under stable policy π
- Calculate (an upper bound on) expected distance from an arbitrary vehicle configuration to closest among n_π points, $\delta^*(n_\pi)$
- At steady-state: $\frac{\lambda}{m} = \frac{1}{\mathbb{E}[\delta^*(n_\pi)]}$
- Little's formula: $\lambda T_\pi = n_\pi$

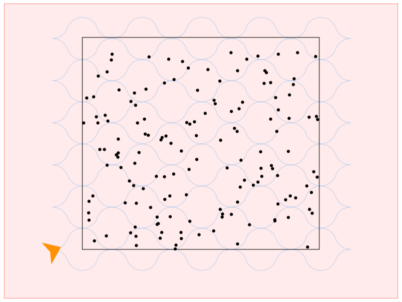
Example: Dubins vehicle

- $\mathbb{E}[\delta^*(n_\pi)] = \frac{3}{4}\left(\frac{3\rho|\mathcal{Q}|}{n_\pi}\right)^{1/3}$
- Steady state+ Little's formula: $\frac{\lambda}{m} = \frac{4}{3}\left(\frac{\lambda T_\pi}{3\rho|\mathcal{Q}|}\right)^{1/3}$
- $\liminf_{\frac{\lambda}{m}\to+\infty} \overline{T}^* \frac{m^3}{\lambda^2} \geq \frac{81}{64}\rho|\mathcal{Q}|$

The multiple sweep tiling algorithm

The single vehicle version

1. Tile $\mathcal{Q}$ with beads of length $\ell = c/\lambda$
2. Update outstanding task list
3. Execute single sweep tiling algorithm
4. Goto 2.



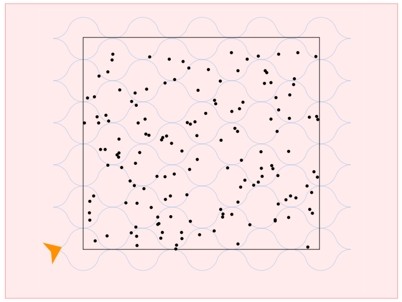
The multi-vehicle version

- Divide $\mathcal{Q}$ into m equal "strips"
- Assign one vehicle to every strip
- Each vehicle executes the multiple sweep algorithm in its own strip

The multiple sweep tiling algorithm

The single vehicle version

1. Tile $\mathcal{Q}$ with beads of length $\ell = c/\lambda$
2. Update outstanding task list
3. Execute single sweep tiling algorithm
4. Goto 2.



The multi-vehicle version

- Divide $\mathcal{Q}$ into m equal "strips"
- Assign one vehicle to every strip
- Each vehicle executes the multiple sweep algorithm in its own strip

Analysis of the multiple sweep algorithm

General protocol

- Each bead can be treated as a separate queue, with Poisson arrival process with intensity $\lambda_{\mathcal{B}} = \lambda \frac{|\mathcal{B}|}{|\mathcal{Q}|}$
- The vehicle visits each bead with at a rate no smaller than $\mu_{\mathcal{B}} \approx$ (single sweep path length)$^{-1}$
- The system time is no greater than the system time for the corresponding M/D/1 queue: $\overline{T}^* \leq \frac{1}{\mu_{\mathcal{B}}} \left(1 + \frac{1}{2} \frac{\lambda_{\mathcal{B}}}{\mu_{\mathcal{B}} - \lambda_{\mathcal{B}}}\right)$
- Optimize over ℓ

Example: Dubins vehicle

- $\lambda_{\mathcal{B}} = \frac{\ell^3 \lambda}{16 \rho |\mathcal{Q}|}$; $\mu_{\mathcal{B}} \geq \frac{\ell^2 m}{16 \rho |\mathcal{Q}|} \left(1 + \frac{7}{3}\pi \frac{\rho}{\sqrt{|\mathcal{Q}|}}\right)^{-1}$
- $\limsup_{\frac{\lambda}{m} \to +\infty} \overline{T}^* \frac{m^3}{\lambda^2} \leq 71 \rho |\mathcal{Q}| \left(1 + \frac{7}{3}\pi \frac{\rho}{\sqrt{|\mathcal{Q}|}}\right)^3$

Analysis of the multiple sweep algorithm

General protocol

- Each bead can be treated as a separate queue, with Poisson arrival process with intensity $\lambda_{\mathcal{B}} = \lambda \frac{|\mathcal{B}|}{|\mathcal{Q}|}$
- The vehicle visits each bead with at a rate no smaller than $\mu_{\mathcal{B}} \approx$ (single sweep path length)$^{-1}$
- The system time is no greater than the system time for the corresponding M/D/1 queue: $\overline{T}^* \leq \frac{1}{\mu_{\mathcal{B}}} \left(1 + \frac{1}{2} \frac{\lambda_{\mathcal{B}}}{\mu_{\mathcal{B}} - \lambda_{\mathcal{B}}}\right)$
- Optimize over ℓ

Example: Dubins vehicle

- $\lambda_{\mathcal{B}} = \frac{\ell^3 \lambda}{16 \rho |\mathcal{Q}|}$; $\mu_{\mathcal{B}} \geq \frac{\ell^2 m}{16 \rho |\mathcal{Q}|} \left(1 + \frac{7}{3}\pi \frac{\rho}{\sqrt{|\mathcal{Q}|}}\right)^{-1}$
- $\limsup_{\frac{\lambda}{m} \to +\infty} \overline{T}^* \frac{m^3}{\lambda^2} \leq 71 \rho |\mathcal{Q}| \left(1 + \frac{7}{3}\pi \frac{\rho}{\sqrt{|\mathcal{Q}|}}\right)^3$

Outline of the lecture

The light load case

- The target generation rate is very small: $\lambda/m \to 0^+$

In such case:

- Almost surely all vehicles will have enough time to return to some "loitering station" between task completion/generation times
- The problem is reduced to the choice of the loitering stations that minimizes the system time

Introducing differential constraints

- Novel challenges:
  - Vehicles possibly cannot stop (e.g., Dubins vehicle, Reeds-Shepp car)
  - Strategies are more complex than defining a loitering "point"
- How many of the results from the Euclidean case carry over to this case?

The light load case

- The target generation rate is very small: $\lambda/m \to 0^+$

In such case:

- Almost surely all vehicles will have enough time to return to some "loitering station" between task completion/generation times
- The problem is reduced to the choice of the loitering stations that minimizes the system time

Introducing differential constraints

- Novel challenges:
  - Vehicles possibly cannot stop (e.g., Dubins vehicle, Reeds-Shepp car)
  - Strategies are more complex than defining a loitering "point"
- How many of the results from the Euclidean case carry over to this case?

A simple lower bound

- The length of shortest feasible path from a vehicle positioned at $p \in \mathbb{R}^2$ to an arbitrary point $q \in \mathcal{Q}$ is lower bounded by $\|q - p\|$
- A simple lower bound on $\overline{T}^*$ is obtained by relaxing differential constraints
- $\overline{T}^* \geq \mathcal{H}_m^*(\mathcal{Q})$
- $\mathcal{H}_m^*(\mathcal{Q}) = \Theta\left(\frac{1}{\sqrt{m}}\right)$

The Median Circling (MC) Policy

Assign "virtual" generators to each agent. All agents do the following, in parallel (possibly asynchronously):

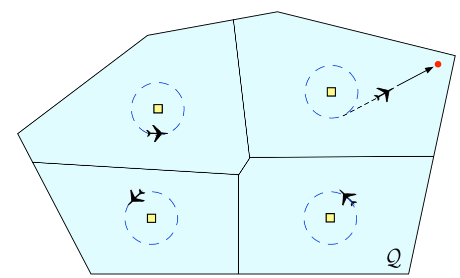
- Update the generator position according to a gradient descent law.
- Service targets in own region, returning to a "loitering circle" of radius 2.91ρ centered on their generator position when done

- We have

$$\lim_{\lambda/m \to 0^+} T_{\mathrm{MC}} \leq \mathcal{H}_m^*(\mathcal{Q}) + 3.76\rho$$

- Furthermore,

$$\lim_{\mathcal{H}_m^* \to +\infty, \lambda/m \to 0^+} \frac{T_{\mathrm{MC}}}{\overline{T}^*} = 1.$$



The Median Circling (MC) Policy

Assign "virtual" generators to each agent. All agents do the following, in parallel (possibly asynchronously):

- Update the generator position according to a gradient descent law.
- Service targets in own region, returning to a "loitering circle" of radius 2.91ρ centered on their generator position when done

- We have

$$\lim_{\lambda/m \to 0^+} T_{\mathrm{MC}} \leq \mathcal{H}_m^*(\mathcal{Q}) + 3.76\rho$$

- Furthermore,

$$\lim_{\mathcal{H}_m^* \to +\infty, \lambda/m \to 0^+} \frac{T_{\mathrm{MC}}}{\overline{T}^*} = 1.$$

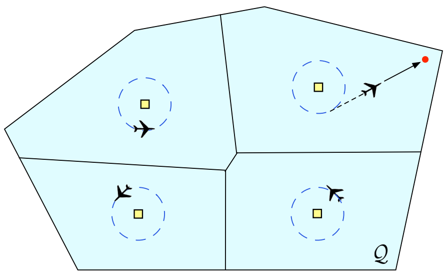
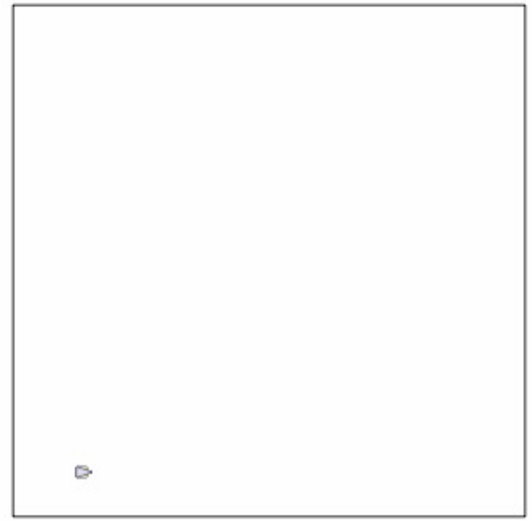


Illustration of the MC policy

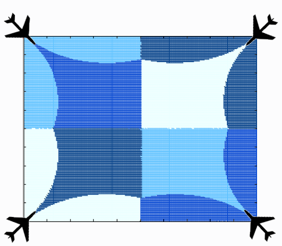


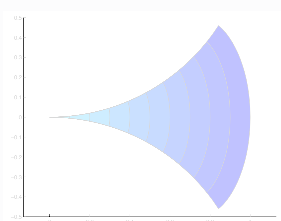
Tighter lower bound using differential constraints

General protocol

- Consider a "frozen moment in time"
- Consider the "modified Voronoi" diagram of the vehicles.
- Relaxation: approximate vehicle Voronoi region by their reachable sets
- Optimize over the vehicle configurations

Example: Dubins vehicle





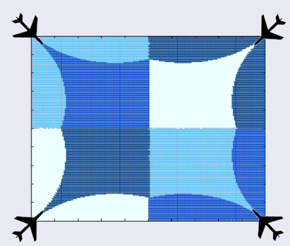
- For $m \geq m_{\mathrm{crit}}$, $\overline{T}^* \geq \frac{k_1(|\mathcal{Q}|, \rho)}{m^{1/3}}$

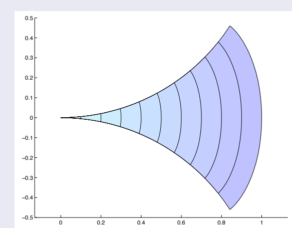
Tighter lower bound using differential constraints

General protocol

- Consider a "frozen moment in time"
- Consider the "modified Voronoi" diagram of the vehicles.
- Relaxation: approximate vehicle Voronoi region by their reachable sets
- Optimize over the vehicle configurations

Example: Dubins vehicle

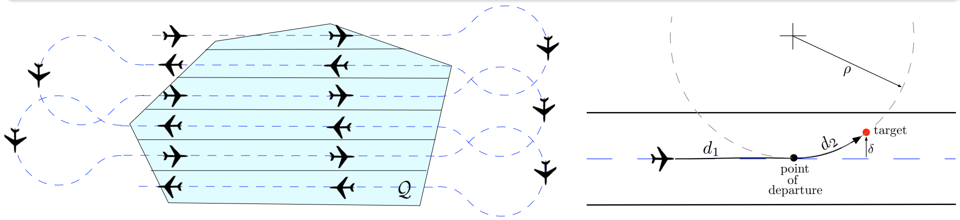




- For $m \geq m_{\mathrm{crit}}$, $\overline{T}^* \geq \frac{k_1(|\mathcal{Q}|, \rho)}{m^{1/3}}$

The Strip Loitering (SL) policy

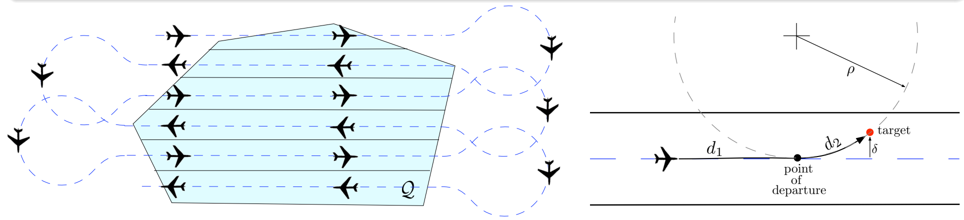
- Divide the environment $\mathcal{Q}$ into strips of width $\min\left\{\frac{k_2(\mathcal{Q},\rho)}{m^{2/3}}, 2\rho\right\}$
- Design a closed loitering path that bisects the strips. All vehicles move along this path, equally spaced, with dynamic regions of responsibility.
- Each vehicle services targets in own region, returning to the "nominal" position on the loitering path.



$$\lim_{m\to+\infty} T_{\mathrm{SL}}\, m^{1/3} \le k_3(\mathcal{Q},\rho), \quad \text{and} \quad \lim_{m\to+\infty} \frac{T_{\mathrm{SL}}}{T^*} \le k_4(\mathcal{Q},\rho).$$

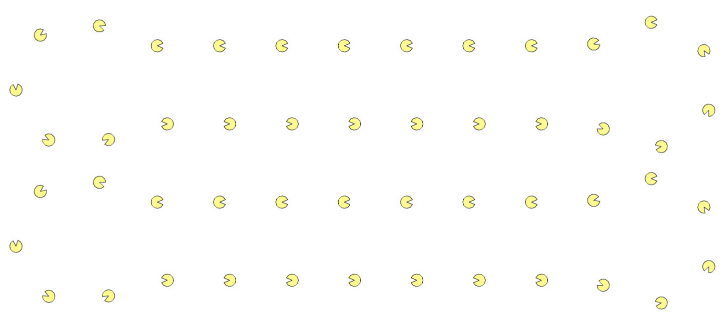
The Strip Loitering (SL) policy

- Divide the environment $\mathcal{Q}$ into strips of width $\min\left\{\frac{k_2(\mathcal{Q},\rho)}{m^{2/3}}, 2\rho\right\}$
- Design a closed loitering path that bisects the strips. All vehicles move along this path, equally spaced, with dynamic regions of responsibility.
- Each vehicle services targets in own region, returning to the "nominal" position on the loitering path.



$$\lim_{m\to+\infty} T_{\mathrm{SL}}\, m^{1/3} \le k_3(\mathcal{Q},\rho), \quad \text{and} \quad \lim_{m\to+\infty} \frac{T_{\mathrm{SL}}}{T^*} \le k_4(\mathcal{Q},\rho).$$

Illustration of the SL policy



Outline of the lecture

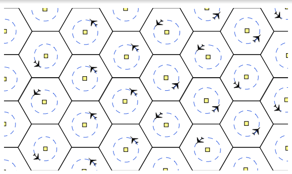
Phase transition in the light load

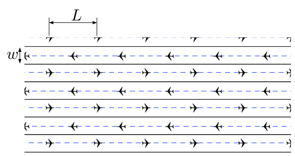
- We have two policies: Median Circling (MC), and Strip Loitering (SL). Which is better?
- Define the non-holonomic density $d_\rho = \frac{\rho^2 m}{|\mathcal{Q}|}$.
  - MC is optimal when $d_\rho \to 0$,
  - SL is within a constant factor of the optimal as $d_\rho \to +\infty$.
- **phase transition:** the optimal organization changes from territorial (MC) to gregarious (SL) depending on the non-holonomic density of the agents.

Phase transition in the light load

- We have two policies: Median Circling (MC), and Strip Loitering (SL). Which is better?
- Define the non-holonomic density $d_\rho = \frac{\rho^2 m}{|\mathcal{Q}|}$.
  - MC is optimal when $d_\rho \to 0$,
  - SL is within a constant factor of the optimal as $d_\rho \to +\infty$.
- **phase transition:** the optimal organization changes from territorial (MC) to gregarious (SL) depending on the non-holonomic density of the agents.

Estimate of the critical density



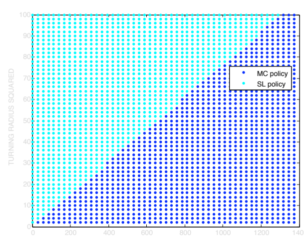


- Ignoring boundary conditions (e.g., consider the unbounded plane), we can compare the coverage cost for the two policies analytically:

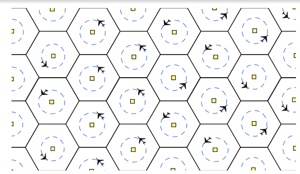
$$T_{\rm SL} < T_{\rm MC} \quad \Leftrightarrow \quad d_\rho > 0.0587$$

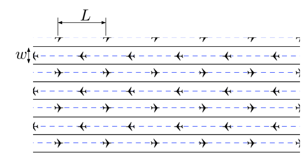
(i.e., transition occurs when the area of the dominance region is about 4-5 times the area of the minimum turning radius circle).

- Simulation results yield $d_\rho^{\rm crit} \approx 0.0759$ (within a factor 1.3 of the analytical result).



Estimate of the critical density



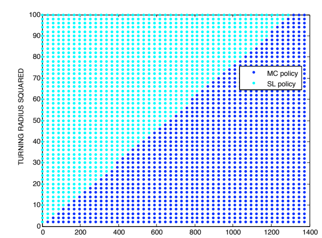


- Ignoring boundary conditions (e.g., consider the unbounded plane), we can compare the coverage cost for the two policies analytically:

$$T_{\rm SL} < T_{\rm MC} \quad \Leftrightarrow \quad d_\rho > 0.0587$$

(i.e., transition occurs when the area of the dominance region is about 4-5 times the area of the minimum turning radius circle).

- Simulation results yield $d_\rho^{\rm crit} \approx 0.0759$ (within a factor 1.3 of the analytical result).



Dynamic Vehicle Routing Summary

	Euclidean vehicle	Dubins vehicle, Reeds-Shepp car Double integrator, Differential drive
$\mathbb{E}$[TSP Length] $(n \to \infty)$	$\Theta(n^{\frac{1}{2}})$	$\Theta(n^{\frac{2}{3}})$
$\overline{T}^*$ $(\frac{\lambda}{m} \to \infty)$	$\Theta(\frac{\lambda}{m^2})$	$\Theta(\frac{\lambda^2}{m^3})$
$\overline{T}^*$ $(\frac{\lambda}{m} \to 0, \frac{m}{\lvert\mathcal{Q}\rvert} \to 0)$	$\Theta(m^{-\frac{1}{2}})$	$\Theta(m^{-\frac{1}{2}})$
$\overline{T}^*$ $(\frac{\lambda}{m} \to 0, \frac{m}{\lvert\mathcal{Q}\rvert} \to \infty)$	$\Theta(m^{-\frac{1}{2}})$	$\Theta(m^{-\frac{1}{3}})$

Lecture outline

1. Models of vehicles with differential constraints
2. Traveling salesperson problems
3. The heavy load case
4. The light load case
5. Phase transition in the light load

Workshop Structure and Schedule

8:00-8:30am	*Coffee Break*	
8:30-9:00am	Lecture #1:	Intro to dynamic vehicle routing
9:05-9:50am	Lecture #2:	Prelims: graphs, TSPs and queues
9:55-10:40am	Lecture #3:	The single-vehicle DVR problem
10:40-11:00am	*Break*	
11:00-11:45pm	Lecture #4:	The multi-vehicle DVR problem
11:45-1:10pm	*Lunch Break*	
1:10-2:10pm	Lecture #5:	Extensions to vehicle networks
2:15-3:00pm	Lecture #6:	Extensions to different demand models
3:00-3:20pm	*Coffee Break*	
3:20-4:20pm	Lecture #7:	Extensions to different vehicle models
4:25-4:40pm	Lecture #8:	Extensions to different task models
4:45-5:00pm		Final open-floor discussion