

Robot Motion Planning II / Multi-Goal Planning

Jan Faigl

Department of Computer Science
Faculty of Electrical Engineering
Czech Technical University in Prague

Lecture 10

A4M36PAH - Planning and Games



Multi-Goal Planning

Multi-Goal Path Planning

Dubins Planning

Data Collection Planning

Mobile Robotic Exploration (TSP-based)



Lecture Goals

- Provide an overview of the existing problem formulations in robotic planning
- Multi-Goal Path Planning a.k.a. robotic Traveling Salesman Problem (TSP)
- Inspection, exploration, and data collection missions
- Challenges in planning for non-holonomic vehicle (Dubins vehicle)
- Example of problem formulations suitable for **robotic data collection planning**

During the lecture, several problems formulation will be defined. Most of them are variants of the TSP. Each problem aims to address a specific issue related to a particular robotic application.

The main goal of the lecture is to make you familiar with the key challenges in the related problems and existing approaches.

*The goal is **not to memorize** all the details and definitions!*



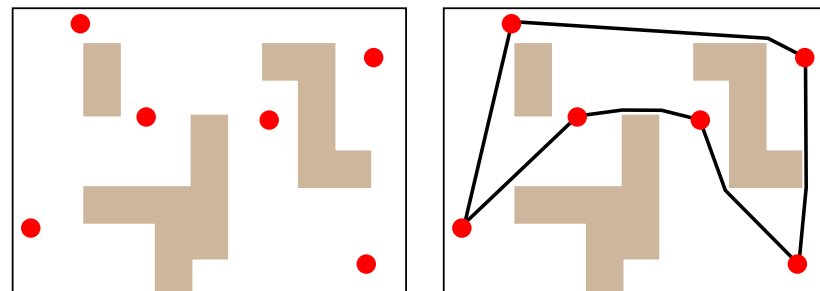
Multi-Goal Path Planning

Motivation

Having a set of locations (goals) to be visited, determine the cost efficient path to visit them and return to a starting location.

- Locations where a robotic arm performs some task
- Locations where a mobile robot has to be navigated

To perform measurements such as scan the environment or read data from sensors.



Alatartsev et al. (2015) – Robotic Task Sequencing Problem: A Survey



Traveling Salesman Problem (TSP)

Given a set of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city.

- The TSP can be formulated for a graph $G(V, E)$, where V denotes a set of locations (cities) and E represents edges connecting two cities with the associated travel cost c (distance), i.e., for each $v_i, v_j \in V$ there is an edge $e_{ij} \in E$, $e_{ij} = (v_i, v_j)$ with the cost c_{ij} .
- If the associated cost of the edge (v_i, v_j) is the Euclidean distance $c_{ij} = |(v_i, v_j)|$, the problem is called the **Euclidean TSP** (ETSP).
In our case, $v \in V$ represents a point in \mathbb{R}^2 and solution of the ETSP is a path in the plane.
- It is known, the TSP is NP-hard (its decision variant) and several algorithms can be found in literature.

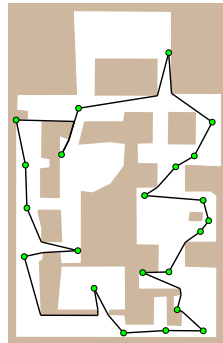
William J. Cook (2012) – In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation



Multi-Goal Path Planning (MTP) Problem

Given a map of the environment \mathcal{W} , mobile robot \mathcal{R} , and a set of locations, what is the shortest possible **collision free path** that visits each location exactly once and returns to the origin location.

- MTP problem is de facto the TSP with the cost associated to the edges as the length of the *shortest* path connecting the locations
- For n locations, we need to compute up to n^2 shortest paths (solve n^2 motion planning problems)
- The paths can be found as the shortest path in a graph (roadmap), from which the $G(V, E)$ for the TSP can be constructed

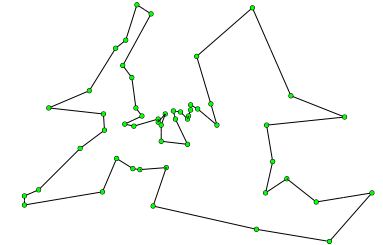


Visibility graph as the roadmap for a point robot provides a straight forward solution, but such a shortest path may not be necessarily feasible for more complex robots



Solutions of the TSP

- Efficient heuristics from the Operational Research have been proposed
- LKH – K. Helsgaun efficient implementation of the Lin-Kernighan heuristic (1998)
<http://www.akira.ruc.dk/~keld/research/LKH/>
- Concorde – Solver with several heuristic and also optimal solver
<http://www.math.uwaterloo.ca/tsp/concorde.html>



Problem Berlin52 from the TSPLIB

Beside the heuristic and approximations algorithms (such as Christofides 3/2-approximation algorithm), other („soft-computing”) approaches have been proposed, e.g., based on genetic algorithms, and memetic approaches, ant colony optimization (ACO), and **neural networks**.



Multi-Goal Path Planning in Robotic Missions

Multi-goal path planning

- It builds on a simple path and trajectory planning
- It is a **combinatorial optimization problem** to **determine the sequence** to visit the given locations
- It allows to solve (or improve performance of) more complex problems such as
 - **Inspection planning** - Find the shortest tour to see (inspect) the whole environment
 - **Data collection planning** – Determine a cost efficient path to collect data from the sensor stations (locations)
 - **Robotic exploration** - Create a map of unknown environment as quickly as possible



Inspection Planning

Motivations (examples)

- Periodically visit particular locations of the environment to check, e.g., for intruders, and return to the starting locations
- Based on available plans, provide a guideline how to search a building to find possible victims as quickly as possible (search and rescue scenario)



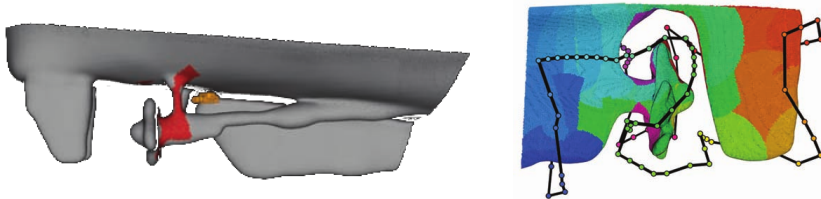
Jan Faigl, 2016

A4M36PAH – Lecture 10: Multi-Goal Planning

10 / 62

Example – Inspection Planning with AUV

- Determine shortest inspection path for Autonomous Underwater Vehicle (AUV) to inspect a propeller of the vessel



Three-dimensional coverage planning for an underwater inspection robot
 Brendan Englot and Franz S. Hover
 International Journal of Robotic Research, 32(9-10):1048–1073, 2013.



Jan Faigl, 2016

A4M36PAH – Lecture 10: Multi-Goal Planning

12 / 62

Inspection Planning – Decoupled Approach

1. Determine sensing locations such that the whole environment would be inspected (seen) by visiting them

A solution of the Art Gallery Problem

2. Create a roadmap connecting the sensing location

E.g., using visibility graph or randomized sampling based approaches

3. Find the inspection path visiting all the sensing locations as a solution of the multi-goal path planning

De facto solution of the TSP

Inspection planning is also called coverage path planning in literature.



Jan Faigl, 2016

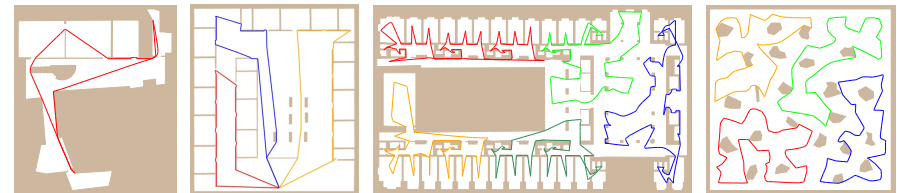
A4M36PAH – Lecture 10: Multi-Goal Planning

11 / 62

Inspection Planning – “Continuous Sensing”

- If we do not prescribe a discrete set of sensing locations, we can formulate the problem as the **Watchman route problem**

Given a map of the environment \mathcal{W} determine the shortest, closed, and collision free path, from which the whole environment is covered by an omnidirectional sensors with the radius ρ .



Approximate Solution of the Multiple Watchman Routes Problem with Restricted Visibility Range
 Jan Faigl
 IEEE Transactions on Neural Networks, 21(10):1668–1679, 2010.



Jan Faigl, 2016

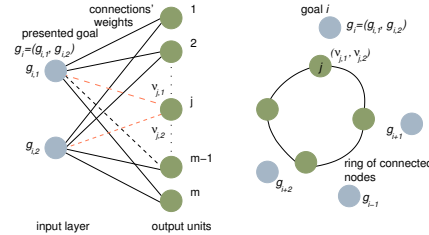
A4M36PAH – Lecture 10: Multi-Goal Planning

13 / 62

Self-Organizing Maps based Solution of the TSP

Kohonen's type of unsupervised two-layered neural network

- Neurons' **weights** represent **nodes** $\mathcal{N} = \{\nu_1, \dots, \nu_m\}$ in a **plane**.
- Nodes are organized into a **ring**.
- Sensing locations $\mathbf{S} = \{s_1, \dots, s_n\}$ are presented to the network in a **random** order.
- Nodes **compete** to be winner according to their distance to the presented goal s



$$\nu^* = \operatorname{argmin}_{\nu \in \mathcal{N}} |\mathcal{D}(\nu, s)|$$

- The **winner** and its **neighbouring** nodes are adapted (**moved**) towards the city according to the neighbouring function

$$f(\sigma, d) = \begin{cases} e^{-\frac{d^2}{\sigma^2}} & \text{for } d < m/n_f, \\ 0 & \text{otherwise,} \end{cases}$$

- Best matching unit ν to the presented prototype s is determined according to distance function $|\mathcal{D}(\nu, s)|$
- For the Euclidean TSP, \mathcal{D} is the Euclidean distance
- However, for problems with obstacles, the multi-goal path planning, \mathcal{D} should correspond to the length of the shortest, collision free path.

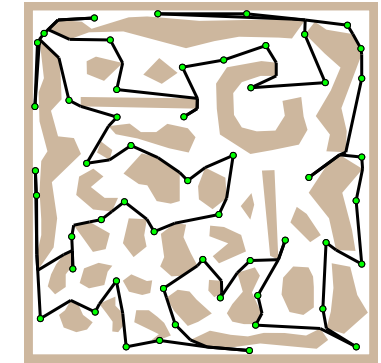


SOM for the Multi-Goal Path Planning

Unsupervised learning procedure

```

 $\mathcal{N} \leftarrow \text{initialization}(\nu_1, \dots, \nu_m)$ 
repeat
   $error \leftarrow 0$ 
  foreach  $g \in \Pi(\mathbf{S})$  do
     $\nu^* \leftarrow$ 
    selectWinner  $\operatorname{argmin}_{\nu \in \mathcal{N}} |S(g, \nu)|$ 
    adapt  $(S(g, \nu), \mu f(\sigma, l) |S(g, \nu)|)$ 
     $error \leftarrow \max\{error, |S(g, \nu^*)|\}$ 
   $\sigma \leftarrow (1 - \alpha) \cdot \sigma$ 
until  $error \leq \delta$ 
    
```



- For multi-goal path planning – the **selectWinner** and **adapt** procedures are based on the solution of the path planning problem

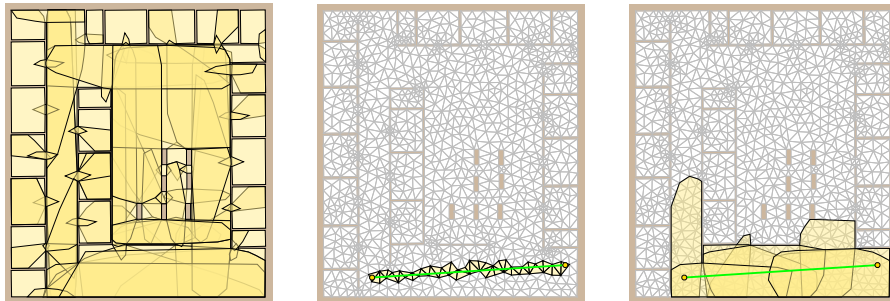
An Application of Self-Organizing Map in the non-Euclidean Traveling Salesman Problem
 Jan Faigl, Miroslav Kulich, Vojtěch Vonásek and Libor Přeučil
 Neurocomputing, 74(5):671–679, 2011.



SOM for the TSP in the Watchman Route Problem

During the unsupervised learning, we can compute **coverage** of \mathcal{W} from the current **ring** (solution represented by the neurons) and **adapt** the network **towards uncovered parts** of \mathcal{W}

- Convex cover set of \mathcal{W} created on top of a triangular mesh
- Incident convex polygons with a straight line segment are found by walking in a triangular mesh technique



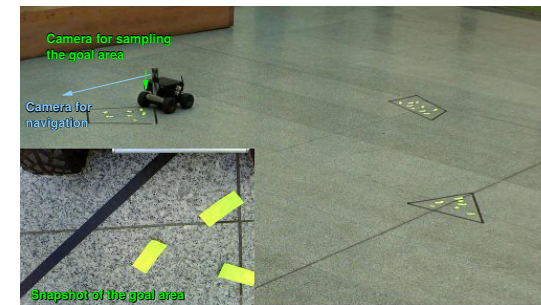
Jan Faigl (2010), TNN



Multi-Goal Path Planning with Goal Areas

- It may be sufficient to visit a goal region instead of the particular point location

E.g., to take a sample measurement at each goal



Not only a sequence of goals visit has to be determined, but also an appropriate sensing location for each goal need to be found.

The problem with goal regions can be considered as a variant of the **Traveling Salesman Problem with Neighborhoods (TSPN)**.



Traveling Salesman Problem with Neighborhoods

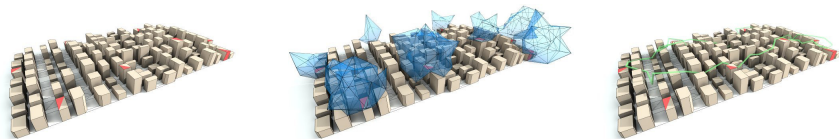
Given a set of n regions (neighbourhoods), what is the shortest closed path that visits each region.

- The problem is NP-hard and APX-hard, it cannot be approximated to within factor $2 - \epsilon$, where $\epsilon > 0$
Safra and Schwartz (2006) – Computational Complexity
- Approximate algorithms exists for particular problem variants
E.g., Disjoint unit disk neighbourhoods
- Flexibility of SOM for the TSP allows to generalize the unsupervised learning procedure to address the TSPN
- **TSPN provides a suitable problem formulation for planning various inspection and data collection missions**



Example – TSPN for Inspection Planning with UAV

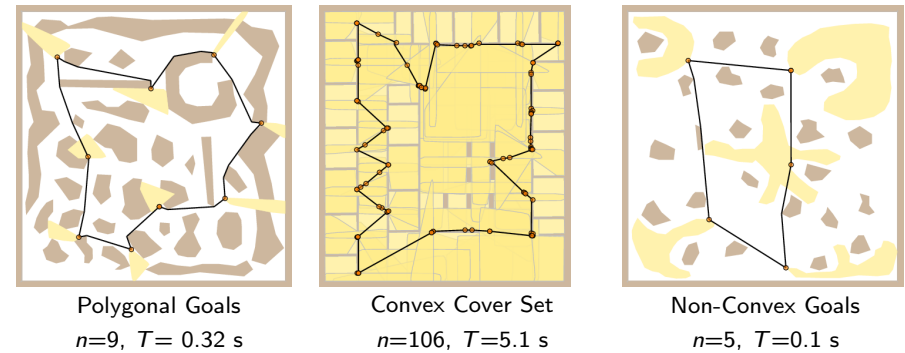
- Determine a cost efficient trajectory from which a given set of target regions is covered
- For each target region a subspace $S \subset \mathbb{R}^3$ from which the target can be covered is determined
S represents the neighbourhood
- The PRM motion planning algorithm is utilized to construct a motion planning roadmap (a graph)
- SOM based solution of the TSP with a graph input is generalized to the TSPN



Janoušek and Faigl, (2013) – ICRA



SOM-based Solution of the Traveling Salesman Problem with Neighborhoods (TSPN)



Visiting Convex Regions in a Polygonal Map, Jan Faigl, Vojtěch Vonásek and Libor Přeučil Robotics and Autonomous Systems, 61(10):1070–1083, 2013.



Example – TSPN for Planning with Localization Uncertainty

- Selection of waypoints from the neighbourhood of each location
- P3AT ground mobile robot in an outdoor environment



Real overall error at the goals decreased from 0.89 m → 0.58 m (about 35%)

- Decrease localization error at the target locations (indoor)



Error decreased from 16.6 cm → 12.8 cm



Improved success of the locations' visits 83% → 95%

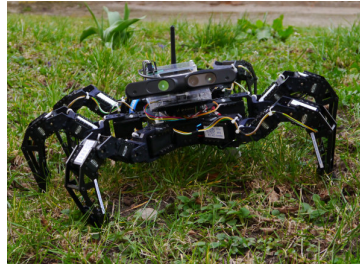
Faigl et al., (2012) – ICRA



Multi-Goal Motion Planning

- In the previous cases, we consider existing roadmap or relatively “simple” collision free (shortest) paths in the polygonal domain
- However, determination of the collision free path in a high dimensional configuration space (\mathcal{C} -space) can be a challenging problem itself
- Therefore, we can generalize the MTP to multi-goal **motion** planning (MGMP) considering motion (trajectory) planners in \mathcal{C} -space.
- An example of MGMP can be

Plan a cost efficient trajectory for hexapod walking robot to visit a set of target locations.

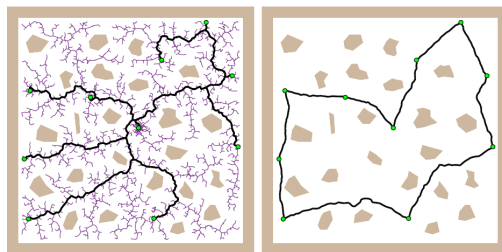
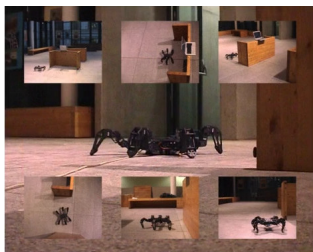


MGMP – Examples of Solutions

- We aim to avoid explicit determination of all paths connecting two locations $g_i, g_j \in \mathcal{G}$
- Various approaches can be found in literature, e.g.,
 - Considering Euclidean distance as approximation in solution of the TSP as the Minimum Spanning Tree (MST)
- Steering RRG roadmap expansion by unsupervised learning of SOM for the TSP

Saha et al. (2006), IJRR

Faigl (2016), WSOM



Problem Statement – MGMP Problem

- The working environment $\mathcal{W} \subset \mathbb{R}^3$ is represented as a set of obstacles $\mathcal{O} \subset \mathcal{W}$ and the robot configuration space \mathcal{C} describes all possible configurations of the robot in \mathcal{W}
- For $q \in \mathcal{C}$, the robot body $\mathcal{A}(q)$ at q is collision free if $\mathcal{A}(q) \cap \mathcal{O} = \emptyset$ and all collision free configurations are denoted as \mathcal{C}_{free}
- Set of n goal locations is $\mathcal{G} = (g_1, \dots, g_n)$, $g_i \in \mathcal{C}_{free}$
- Collision free path from q_{start} to q_{goal} is $\kappa : [0, 1] \rightarrow \mathcal{C}_{free}$ with $\kappa(0) = q_{start}$ and $d(\kappa(1), q_{end}) < \epsilon$, for an admissible distance ϵ
- Multi-goal path τ is **admissible** if $\tau : [0, 1] \rightarrow \mathcal{C}_{free}$, $\tau(0) = \tau(1)$ and there are n points such that $0 \leq t_1 \leq t_2 \leq \dots \leq t_n$, $d(\tau(t_i), v_i) < \epsilon$, and $\bigcup_{1 < i \leq n} v_i = \mathcal{G}$
- **The problem is to find path τ^* for a cost function c such that $c(\tau^*) = \min\{c(\tau) \mid \tau \text{ is admissible multi-goal path}\}$**



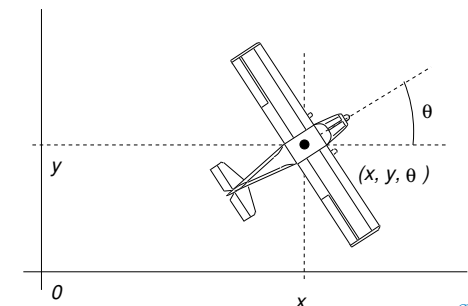
Dubins Vehicle

- Non-holonomic vehicle such as car-like or aircraft can be modeled as the Dubins vehicle
 - Constant forward velocity
 - Limited minimal turning radius ρ
 - Vehicle state is represented by a triplet $q = (x, y, \theta)$, where
 - $(x, y) \in \mathbb{R}^2$, $\theta \in \mathbb{S}^2$ and thus, $q \in SE(2)$

The vehicle motion can be described by the equation:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = v \begin{bmatrix} \cos \theta \\ \sin \theta \\ \frac{u}{\rho} \end{bmatrix}, \quad |u| \leq 1,$$

where u is the control input.



Optimal Maneuvers for Dubins Vehicle

- For two states $q_1 \in SE(2)$ and $q_2 \in SE(2)$ in the environment **without obstacles** $\mathcal{W} = \mathbb{R}^2$ the optimal paths can be characterized as one of two main types

- CCC type: LRL, RLR;
- CSC type: RLR, LSL, LSR, RSL, RSR;

where S – straight line arc, C – circular arc oriented to left (L) or right (R)

L. E. Dubins (1957) – American Journal of Mathematics

- The optimal paths are called **Dubins maneuvers**:
 - Constant velocity: $v(t) = v$ and turning radius ρ
 - 6** types of trajectories connecting any configuration in $\mathbb{R}^2 \times \mathbb{S}^1$ *without obstacles*
 - The control u is according to C and S type one of the three possible values $u \in \{-1, 0, 1\}$



Planning with Dubins vehicle

- The optimal path connecting two configurations can be found analytically

E.g., for UAVs that usually operates in environment without obstacles
- The Dubins maneuvers can be used in randomized-sampling based motion planners, such as RRT, in the control based sampling
- We can consider the model of Dubins vehicle in the multi-goal path planning
 - Surveillance, inspection or monitoring missions to periodically visits given target locations (areas)

Dubins Traveling Salesman Problem DTSP

Given a set of locations, what is the shortest Dubins path that visits each location exactly once and returns to the origin location.



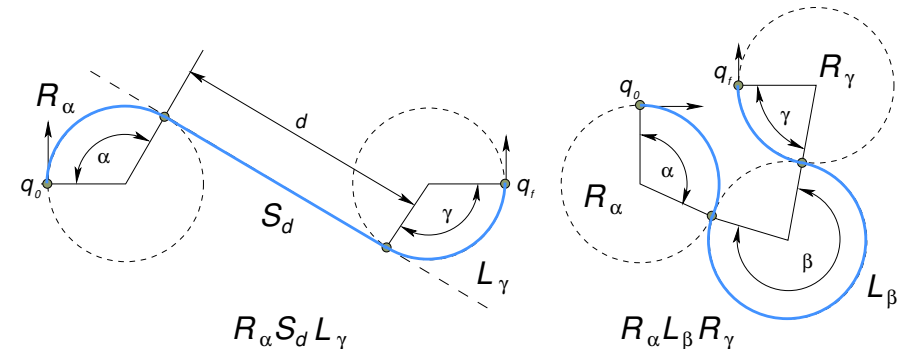
Parametrization of Dubins Maneuvers

- Parametrization of each trajectory phase:

$$\{L_\alpha R_\beta L_\gamma, R_\alpha L_\beta R_\gamma, L_\alpha S_d L_\gamma, L_\alpha S_d R_\gamma, R_\alpha S_d L_\gamma, R_\alpha S_d R_\gamma\}$$

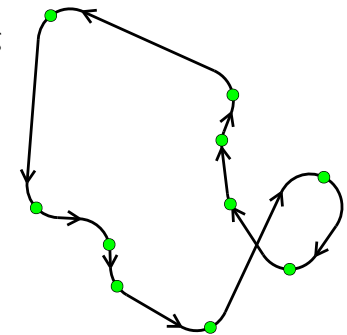
for $\alpha \in [0, 2\pi)$, $\beta \in (\pi, 2\pi)$, $d \geq 0$

Notice the prescribed orientation at q_0 and q_f .



Dubins Traveling Salesman Problem (DTSP)

- Let have Dubins vehicle with minimal turning radius ρ
- Let the given set of n target locations be $G = \{g_1, \dots, g_n\}$
- Let $\Sigma = (\sigma_1, \dots, \sigma_n)$ be a permutation of $\{1, \dots, n\}$
- Let \mathcal{P} be projection form $SE(2)$ to \mathbb{R}^2 such that $\mathcal{P}(q_i) = (x_i, y_i)$, $q_i \in SE(2)$ and $g_i = (x_i, y_i)$.



- DTSP is a problem to determine the minimum length tour that visits every location $g_i \in G$ while satisfying motion constraints of the Dubins vehicle



DTSP – Optimization Criterion

- DTSP is an optimization problem over all permutations Σ and headings $\Theta = \{\theta_{\sigma_1}, \dots, \theta_{\sigma_n}\}$ for the states $(q_{\sigma_1}, \dots, q_{\sigma_n})$ such that $q_{\sigma_i} = (g_{\sigma_i}, \theta_{\sigma_i})$

$$\text{minimize}_{\Sigma, \Theta} \sum_{i=1}^{n-1} \mathcal{L}(q_{\sigma_i}, q_{\sigma_{i+1}}) + \mathcal{L}(q_{\sigma_n}, q_{\sigma_1}) \quad (1)$$

$$\text{subject to } q_i = (g_i, \theta_i) \quad i = 1, \dots, n \quad (2)$$

- $\mathcal{L}(q_{\sigma_i}, q_{\sigma_j})$ is the length of the shortest possible Dubins maneuver connecting the states q_{σ_i} and q_{σ_j} .



Algorithms for the DTSP

Two fundamental approaches can be found in literature

- Considering a sequence of the visits is given
E.g., found by a solution of the Euclidean TSP
- Sampling the headings at the locations into discrete sets of values and considering the problem as the variant of the Generalized TSP
Sampling based approaches

Besides, further approaches are

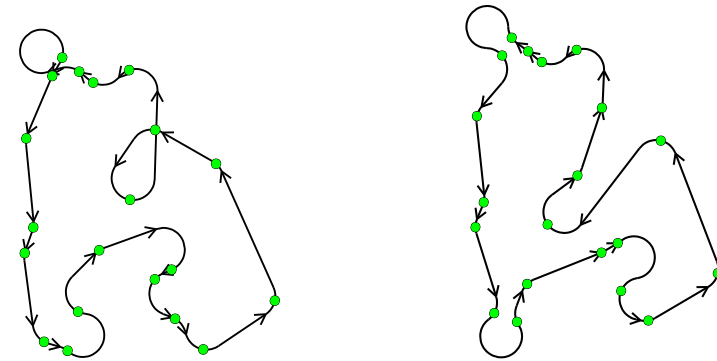
- Approximation algorithms; optimal solutions for restricted variants
- Soft-computing technique such as genetic and memetic technique or neural networks



Challenges of the Dubins Traveling Salesman Problem

- The key difficulty of the DTSP is that the path length mutually depends on
 - Order of the visits to the locations
 - Headings at the target locations

We need the sequence to determine headings, but headings may influence the sequence

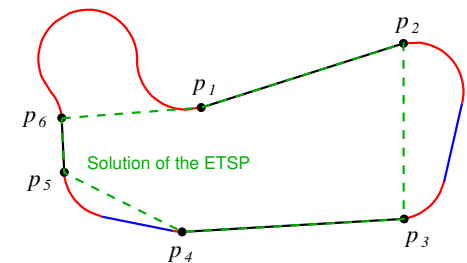


DTSP – Alternating Algorithm

Alternating Algorithm (AA) provides a solution of the DTSP for an **even** number of targets n

Savla et al. (2005)

- Solve the related Euclidean TSP
Relaxed motion constraints
- Establish headings for even edges using straight line segments
- Determine optimal maneuvers for odd edges



Courtesy of P. Váňa

AA is heuristic algorithm which solutions can be bounded by $L_{TSP} \kappa \lceil n/2 \rceil \pi \rho$, where L_{TSP} is the length of the optimal solution of the ETSP and $\kappa < 2.658$.



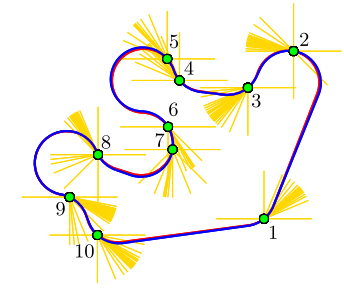
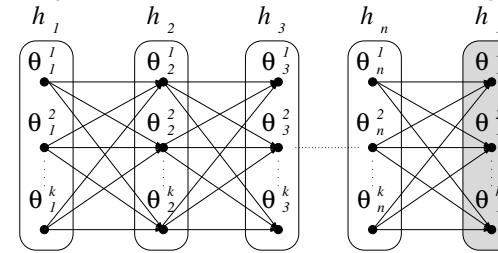
DTSP with the Given Sequence of the Visits to the Targets

- If the sequence of the visits Σ to the target locations is given
- the problem is to determine the optimal heading at each location
- We call the problem as the **Dubins Touring Problem (DTP)**
Váňa and Faigl (2016)
- Let for each location $g_i \in G$ sample possible heading to k values, i.e., for each g_i the set of headings be $h_i = \{\theta_1^1, \dots, \theta_1^k\}$.
- Since Σ is given, we can construct a graph connecting two consecutive locations in the sequence by all possible headings
- For such a graph and particular headings $\{h_1, \dots, h_n\}$, we can find an optimal headings and thus, **the optimal solution of the DTP.**



DTSP as a Solution of the DTP

The first layer is duplicated layer to support the forward search method



- Better solution of the DTP can be found for more samples
- Better solution of the DTP will also improve the DTSP, but only for the given sequence

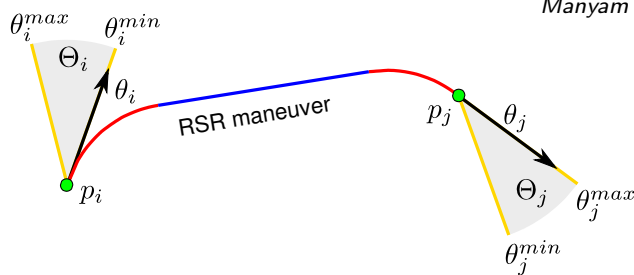
Two questions arise for a practical solution of the DTP

- **How to sample the headings? Since more samples makes finding solution more demanding**
- **We need it sample the headings in a "smart" way.**
- **What is the solution quality? Is there a tight lower bound?**



Dubins Interval Problem

- Dubins Interval Problem (DIP) is a generalization of Dubins maneuvers to the shortest path connecting two points p_i and p_j
- In the DIP, an leaving interval Θ_i at p_i and arrival interval Θ_j at p_j are allowed
- The optimal solution can be found analytically
Manyam et al. (2015)

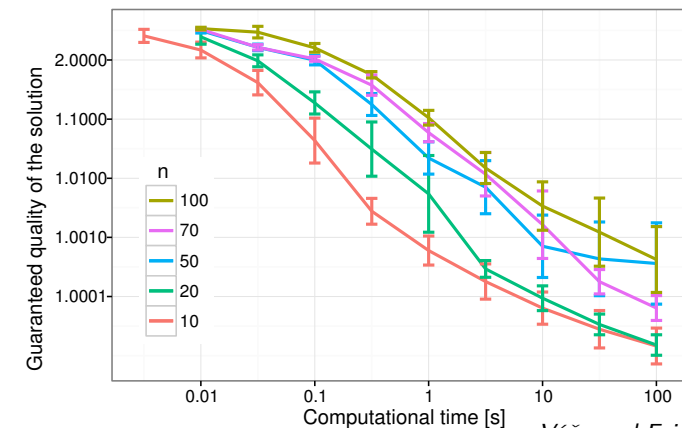


- Solution of the DIP is a tight lower bound for the DTP
Manyam and Rathinam (2015)
- Solution of the DIP is not a feasible solution of the DTP
Notice, for $\Theta_i = \Theta_j = \langle 0, 2\pi \rangle$ the optimal maneuver for DIP is straight line segment



The DIP-based Sampling of Headings in the DTP

- A similar graph as for DTP can be used for heading intervals
- The solution of the DIP is a lower bound of the DTP
- It can be used to inform how to splitting heading intervals
- The ratio between the lower bound and feasible solution of the DTP provides estimation of the solution quality



Váňa and Faigl (2016)



DTSP – Sampling-based Approach

- Sampled heading values can be directly utilized to find the sequence as a solution of the **Generalized Traveling Salesman Problem (GTSP)**

Also called Set TSP or Covering Salesman Problem

Notice For Dubins vehicle, it is the Generalized Asymmetric TSP (GATSP)

The problem is to determine a shortest tour in a graph that visits all specified subsets of the graph's vertices.

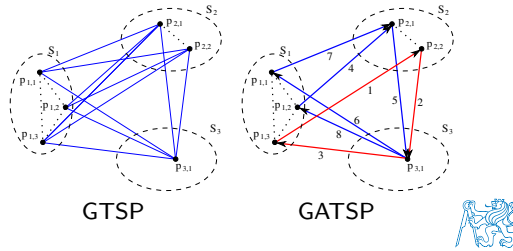
The TSP is a special case of the GTSP when each subset to be visited consists just a single vertex.

- GATSP → ATSP

Noon and Bean (1991)

- ATSP can be solved by LKH
- ATSP → TSP, which can be solved optimally

E.g., by Concorde



DTSPN – Optimization Criterion

- DTSPN is an optimization problem over all permutations Σ , headings $\Theta = \{\theta_{\sigma_1}, \dots, \theta_{\sigma_n}\}$ and points $P = (p_{\sigma_1}, \dots, p_{\sigma_n})$ for the states $(q_{\sigma_1}, \dots, q_{\sigma_n})$ such that $q_{\sigma_i} = (p_{\sigma_i}, \theta_{\sigma_i})$ and $p_{\sigma_i} \in R_{\sigma_i}$:

$$\text{minimize}_{\Sigma, \Theta, P} \sum_{i=1}^{n-1} \mathcal{L}(q_{\sigma_i}, q_{\sigma_{i+1}}) + \mathcal{L}(q_{\sigma_n}, q_{\sigma_1}) \quad (3)$$

$$\text{subject to } q_i = (p_i, \theta_i), p_i \in R_i \quad i = 1, \dots, n \quad (4)$$

- $\mathcal{L}(q_{\sigma_i}, q_{\sigma_j})$ is the length of the shortest possible Dubins maneuver connecting the states q_{σ_i} and q_{σ_j} .



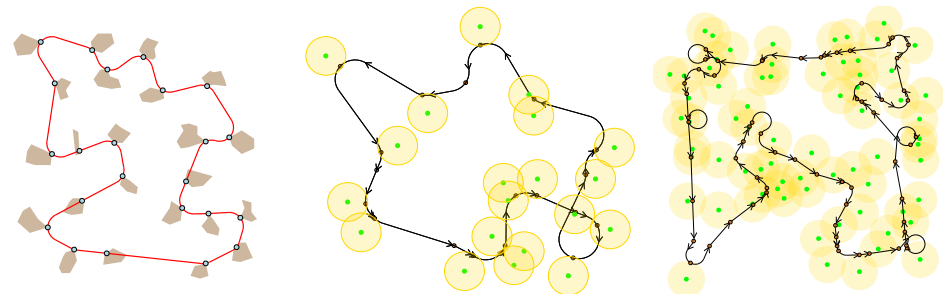
Dubins Traveling Salesman Problem with Neighborhoods

- In surveillance planning, it may be required to visit a set of target regions $\mathbf{G} = \{R_1, \dots, R_n\}$ by the Dubins vehicle
- Then, for each target region R_i , we have to determine a particular point of the visit $p_i \in R_i$ and DTSP becomes the **Dubins Traveling Salesman Problem with Neighborhoods (DTSPN)**

In addition to Σ and headings Θ , locations P have to be determined.



DTSPN – Examples of Solution



Váña and Faigl (2015), (IROS)

Faigl and Váña (2016)

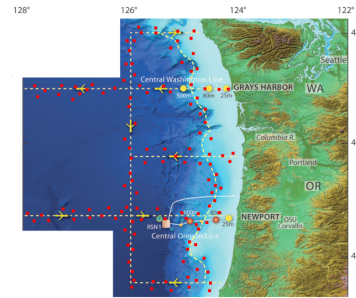


Autonomous Data Collection

- Having a set of sensors (sampling stations), we aim to determine a cost efficient path to retrieve data from the individual sensors

E.g., Sampling stations on the ocean floor

- The planning problem is a variant of the **Traveling Sampling Problem**



Two practical aspects of the data collection can be identified

- Data from particular sensors may be of different importance
- Data from the sensor can be retrieved using wireless communication

These two aspects can be considered in Prize-Collecting Traveling Salesman Problem (PC-TSP) and Orienteering Problem (OP) and their extensions with neighborhoods.



PC-TSPN – Optimization Criterion

The **PC-TSPN** is a problem to

- Determine a set of unique locations** $G = \{g_1, \dots, g_k\}$, $k \leq n$, $g_i \in \mathbb{R}^2$, at which data readings are performed
- Find a cost efficient tour** T visiting G such that the total cost $C(T)$ of T is minimal

$$C(T) = \sum_{(g_i, g_{i+1}) \in T} c(g_i, g_{i+1}) + \sum_{s \in S \setminus S_T} \zeta(s), \quad (5)$$

where $S_T \subseteq S$ are sensors such that for each $s_i \in S_T$ there is g_{l_j} on $T = (g_{l_1}, \dots, g_{l_{k-1}}, g_{l_k})$ and $g_{l_j} \in G$ for which $|(s_i, g_{l_j})| \leq \delta$.

- PC-TSPN includes other variants of the TSP
 - for $\delta = 0$ it is the PC-TSP
 - for $\zeta(s_i) = 0$ and $\delta \geq 0$ it is the TSPN
 - for $\zeta(s_i) = 0$ and $\delta = 0$ it is the ordinary TSP



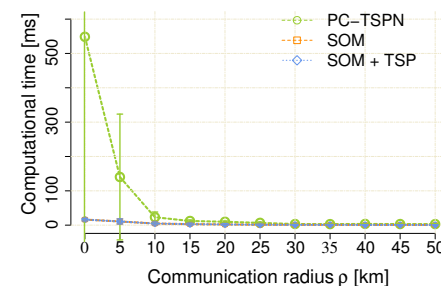
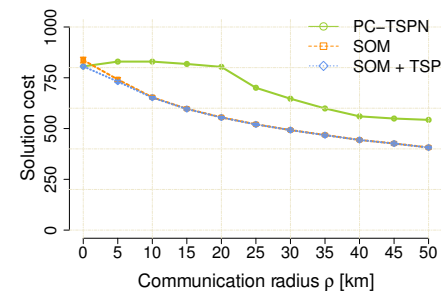
Prize-Collecting Traveling Salesman Problem with Neighborhoods (PC-TSPN)

- Let n sensors be located in \mathbb{R}^2 at the locations $S = \{s_1, \dots, s_n\}$
- Each sensor has associated penalty $\zeta(s_i) \geq 0$ characterizing additional cost if the data are not retrieved from s_i
- Let the data collecting vehicle operates in \mathbb{R}^2 with the motion cost $c(p_1, p_2)$ for all pairs of points $p_1, p_2 \in \mathbb{R}^2$
- The data from s_i can be retrieved within δ distance from s_i

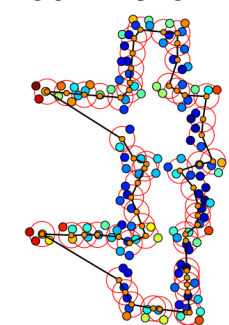


PC-TSPN – Example of Solution

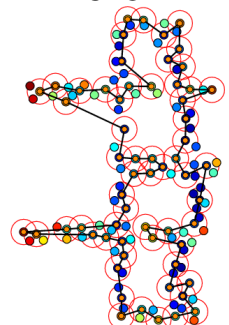
Ocean Observatories Initiative (OOI) scenario



SOM PCTSPN



PCTSPN



Orienteering Problem

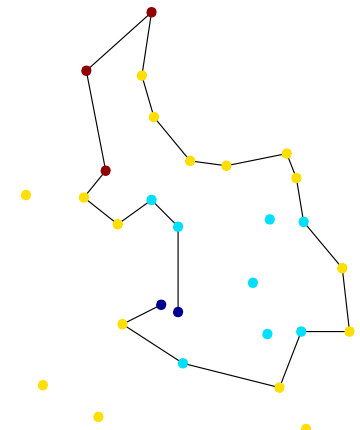
- The **Orienteering Problem (OP)** originates from the orienteering outdoor sport
- The problem is to collect as many rewards as possible within the given travel budget
- It is similar to the PC-TSP, but the tour length must not exceed the prescribed maximize tour length T_{max}
- In OP, the starting and termination locations are prescribed, and they can be different

The solution may not be a closed tour as in the TSP



Orienteering Problem – Specification

- Let the given set of n sensors be located in \mathbb{R}^2 with the locations $S = \{s_1, \dots, s_n\}$, $s_i \in \mathbb{R}^2$
- Each sensor s_i has an associated score s_i characterizing the reward if data from s_i are collected
- The vehicle is operating in \mathbb{R}^2 and the travel cost is the Euclidean distance
- The starting and termination locations are prescribed
- We aim to determine a subset of k locations $S_k \subseteq S$ that maximizes the sum of the collected rewards while the travel cost to visit them is below T_{max} .



Orienteering Problem – Optimization Criterion

- Let $\Sigma = (\sigma_1, \dots, \sigma_k)$ be a permutation of k sensor labels, $1 \leq \sigma_i \leq n$ and $\sigma_i \neq \sigma_j$ for $i \neq j$
- Σ defines a tour $T = (s_{\sigma_1}, \dots, s_{\sigma_k})$ visiting the selected sensors S_k
- Let the start and end points of the tour be $\sigma_1 = 1$ and $\sigma_k = n$
- The **Orienteering problem (OP)** is to determine the number of sensors k , the subset of sensors S_k , and their sequence Σ such that

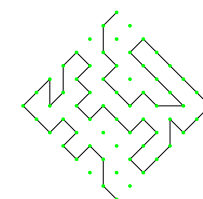
$$\begin{aligned}
 & \text{maximize}_{k, S_k, \Sigma} && R = \sum_{i=1}^k s_{\sigma_i} \\
 & \text{subject to} && \sum_{i=2}^k |(s_{\sigma_{i-1}}, s_{\sigma_i})| \leq T_{max} \text{ and} \\
 & && s_{\sigma_1} = s_1, s_{\sigma_k} = s_n.
 \end{aligned} \tag{6}$$

The OP combines the problem of determining the most valuable locations S_k with finding the shortest tour T visiting the locations S_k . It is NP-hard, since for $s_1 = s_n$ and particular S_k it becomes the TSP.

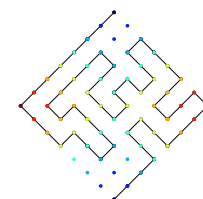


Orienteering Problem – Example of Solutions

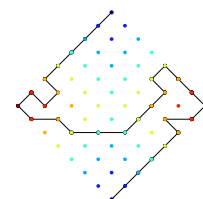
- Heuristic algorithms have been proposed
E.g., Ramesh et al. (1991), Chao et. al. (1996)



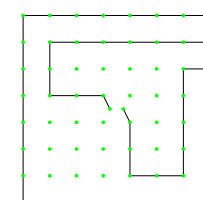
$T_{max}=80, R=1248$



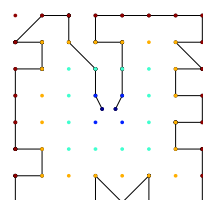
$T_{max}=80, R=1278$



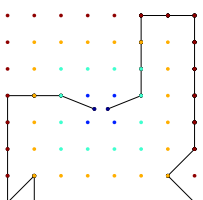
$T_{max}=45, R=756$



$T_{max}=95, R=1395$



$T_{max}=95, R=1335$

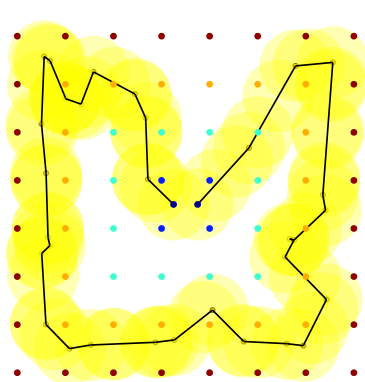


$T_{max}=60, R=845$

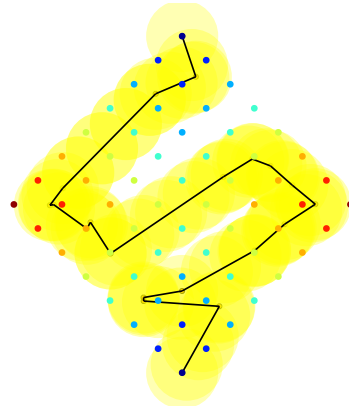


Orienteering Problem with Neighborhoods

- Similarly to the TSP with Neighborhoods and PC-TSPN we can formulate the **Orienteering Problem with Neighborhoods**.



$T_{max}=60, \delta=1.5, R=1600$



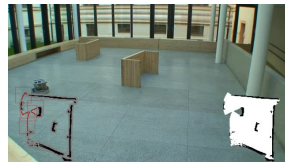
$T_{max}=45, \delta=1.5, R=1344$



Multi-Robot Exploration Strategy

- A set of m robots at positions $\mathbf{R} = \{r_1, r_2, \dots, r_m\}$
- At time t , let a set of n goal candidates be $\mathbf{G}(t) = \{g_1, \dots, g_n\}$

e.g., frontiers



- The exploration strategy (at the planning step t):

Select a goal $g \in \mathbf{G}(t)$ for each robot $r \in \mathbf{R}$ that will minimize the required time to explore the environment.

The problem is formulated as the **task-allocation problem**

$$(\langle r_1, g_{r_1} \rangle, \dots, \langle r_m, g_{r_m} \rangle) = \text{assign}(\mathbf{R}, \mathbf{G}(t), \mathcal{M}),$$

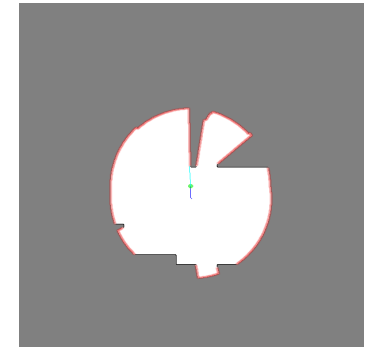
where \mathcal{M} is the current map

We consider only the distance cost for the assignment



Mobile Robot Exploration

- Create a map of the environment
- **Frontier**-based approach
Yamauchi (1997)
- Occupancy grid
Moravec and Elfes (1985)
- Laser scanner sensor
- Next-best-view approach
Select the next robot goal



Performance metric:

Time to create the map of the whole environment

search and rescue mission



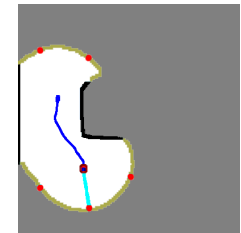
Distance Cost Variants

Simple robot-goal distance

- Evaluate all goals using the robot-goal distance
a length of the path from the robot position to the goal candidate

- Greedy goal selection

Select the closest goal candidate

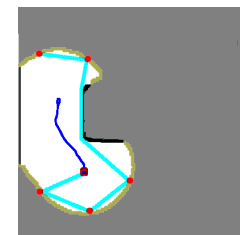


TSP distance cost

- **On Distance Utility in the Exploration Task**
Miroslav Kulich, Jan Faigl and Libor Přeucil ICRA, 2011, 4455-4460.

- Consider visitations of all goals
Solve the associated traveling salesman problem (TSP)
- A length of the tour visiting all goals
- Goal representatives

TSP distance cost improves performance about 10-30%



Multi-Robot Exploration – Problem Definition

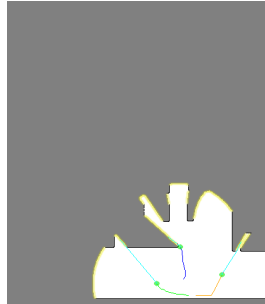
A problem of creating a grid map of the unknown environment by a set of m robots $\mathbf{R} = \{r_1, r_2, \dots, r_m\}$.

Exploration is an iterative procedure:

1. Collect new sensor measurements
2. Determine a set of goal candidates

$$\mathbf{G}(t) = \{g_1, g_2, \dots, g_n\}$$

e.g., frontiers



3. At time step t , select next goal for each robot as the task-allocation problem

$$\langle \langle r_1, g_{r_1} \rangle, \dots, \langle r_m, g_{r_m} \rangle \rangle = \text{assign}(\mathbf{R}, \mathbf{G}(t), \mathcal{M}(t))$$

using the distance cost function

4. Navigate robots towards goal
5. If $|\mathbf{G}(t)| > 0$ go to Step 1; otherwise terminate



Comparison – Goal Assignment Strategies

1. Greedy Assignment

Yamauchi B, Robotics and Autonomous Systems 29, 1999

- Randomized greedy selection of the closest goal candidate

2. Iterative Assignment

Werger B, Mataric M, Distributed Autonomous Robotic Systems 4, 2001

- Centralized variant of the broadcast of local eligibility algorithm (BLE)

3. Hungarian Assignment

- Optimal solution of the task-allocation problem for assignment of n goals and m robots in $O(n^3)$

Stachniss C, C implementation of the Hungarian method, 2004

4. MTSP Assignment

- $\langle \text{cluster-first, route-second} \rangle$, the TSP distance cost

In all strategies, we use the identical selection of the goal candidates from the frontiers.



Proposed Multiple Traveling Salesman Approach

- Consider the task-allocation problem as the **Multiple Traveling Salesman Problem (MTSP)**

- MTSP heuristic $\langle \text{cluster-first, route-second} \rangle$

1. Cluster the goal candidates \mathbf{G} to m clusters

$$\mathbf{C} = \{C_1, \dots, C_m\}, C_i \subseteq \mathbf{G}$$

using *K-means*

2. For each robot $r_i \in \mathbf{R}, i \in \{1, \dots, m\}$ select the next goal g_i from C_i using the TSP distance cost

Kulich et al., ICRA (2011)

- Solve the TSP on the set $C_i \cup \{r_i\}$

the tour starts at r_i

- The next robot goal g_i is the first goal of the found TSP tour



Goal Assignment using Distance Cost in Multi-Robot Exploration

*Jan Faigl, Miroslav Kulich and Libor Přeučil
IROS, 2012, 3741–3741.*



Statistical Evaluation of the Exploration Strategies

- Evaluation for the number of robots m and sensor range ρ

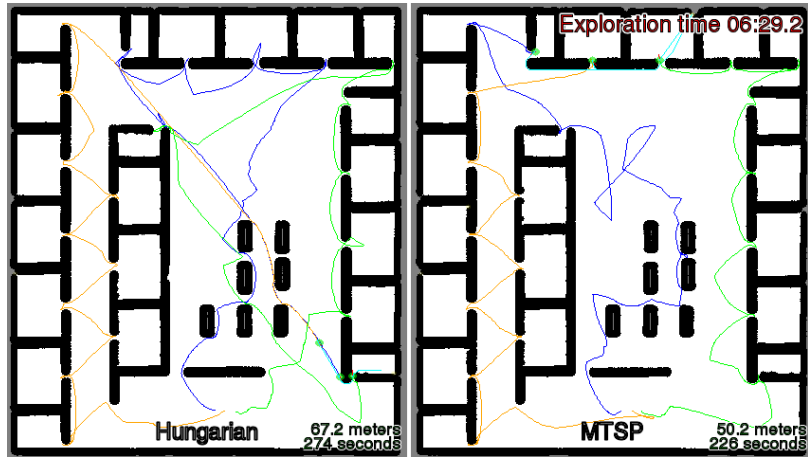
ρ	m	Iterative	Hungarian	MTSP
		vs Greedy	vs Iterative	vs Hungarian
3.0	3	+	=	+
3.0	5	+	=	+
3.0	7	+	=	+
3.0	10	+	+	-
4.0	3	+	=	+
4.0	5	+	=	=
4.0	7	+	=	+
4.0	10	+	+	-
5.0	3	+	=	+
5.0	5	+	=	+
5.0	7	+	=	+
5.0	10	+	+	-

Total number of trials 14 280



Performance of the MTSP vs Hungarian Algorithm

- Replanning as quickly as possible; $m = 3, \rho = 3 m$



The MTSP assignment provides better performance



Summary

- Introduction to multi-goal path planning
- Overview of Dubins planning and DTSP
- Data collection planning
- Overview of multi-robot exploration based on the TSP

Robotic TSP

