Data Collection Planning with Curvature-Constrained Vehicles
–
Dubins Traveling Salesman Problem with Neighborhoods (DTSPN)
and
Dubins Orienteering Problem with Neighborhoods (DOPN)

Jan Faigl

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

Lecture 09

B4M36UIR – Artificial Intelligence in Robotics

Overview of the Lecture

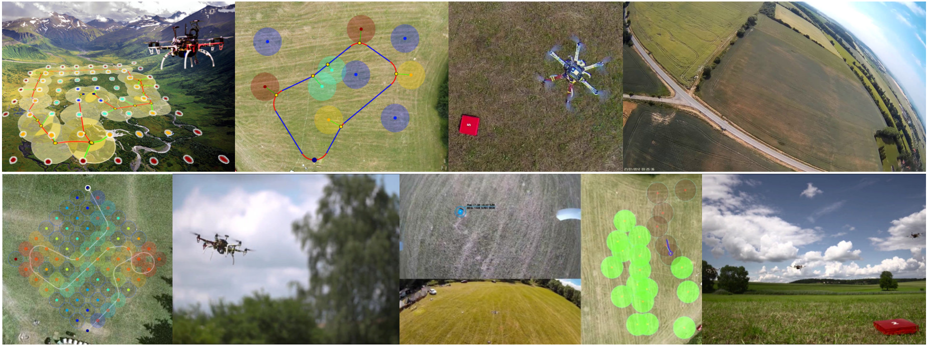
- Part 1 – Data Collection Planning – Aerial Surveillance Missions
  - Dubins Vehicle and Dubins Planning
  - Dubins Touring Problem (DTP)
  - Dubins Traveling Salesman Problem
  - Dubins Traveling Salesman Problem with Neighborhoods
  - Dubins Orienteering Problem
  - Dubins Orienteering Problem with Neighborhoods
  - Planning in 3D – Examples and Motivations
- Part 2 – Bonus HW03b – Data Collection Planning for Surveillance Missions
  - HW03b – Motivation and Assignment

Part I

Part 1 – Data Collection Planning – Aerial Surveillance Missions

Motivation – Surveillance Missions with Aerial Vehicles

- Provide **curvature-constrained** path to collect the most valuable measurements with shortest possible path/time or under limited travel budget



- Formulated as routing problems with Dubins vehicle
  - **Dubins Traveling Salesman Problem with Neighborhoods**
  - **Dubins Orienteering Problem with Neighborhoods**

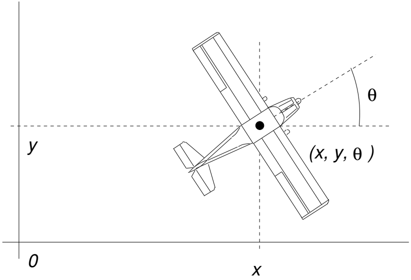
Dubins Vehicle

- 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
  - Position is $(x, y) \in \mathbb{R}^2$, vehicle heading is $\theta \in \mathbb{S}^2$, 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 path connecting q_1 with q_2 can be characterized as one of two main types
  - **CCC** type: LRL, RLR;
  - **CSC** type: 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
  - **Six** types of trajectories connecting any configuration in $SE(2)$

    without obstacles
  - The control u is according to C and S type one of three possible values $u \in \{-1, 0, 1\}$

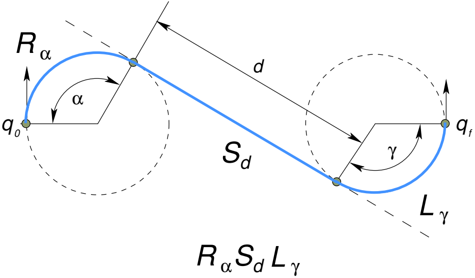
Parametrization of Dubins Maneuvers

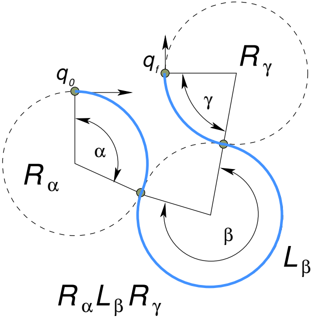
- Parametrization of each trajectory phase:

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

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

Notice the prescribed orientation at q_0 and q_f.





Planning with Dubins Vehicle – Summary

- 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 also be used in randomized-sampling based motion planners, such as RRT, in the control based sampling
- Dubins vehicle model can be considered in the multi-goal path planning
  - Surveillance, inspection or monitoring missions to periodically visits given target locations (areas)
- **Dubins Touring Problem**(**DTP**)

  Given a sequence of locations, what is the shortest path visting the locations, i.e., what are the headings of the vehicle at the locations
- **Dubins Traveling Salesman Problem** (**DTSP**)

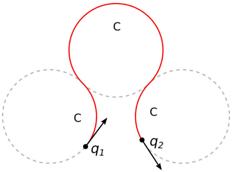
  Given a set of locations, what is the shortest Dubins path that visits each location exactly once and returns to the origin location
- **Dubins Orienteering Problem** (**DOP**)

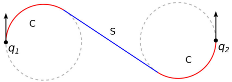
  Given a set of locations, each with associated reward, what is the Dubins path visiting the most rewarding locations and not exceeding the given travel budget

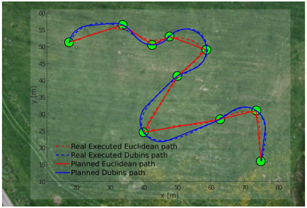
Dubins (Multi-Goal) Path

- Minimal turning radius ρ
- Constant forward velocity v
- State of the Dubins vehicle is $q = (x, y, \theta)$, $q \in SE(2)$, $(x, y) \in \mathbb{R}^2$ and $\theta \in \mathbb{S}^1$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = v \begin{bmatrix} \cos\theta \\ \sin\theta \\ \frac{u}{\rho} \end{bmatrix}$$







Smooth Dubins path connecting a sequence of locations is also suitable for multi-rotor aerial vehicle

- Optimal path connecting $q_1 \in SE(2)$ and $q_2 \in SE(2)$ consists only of straight line arcs and arcs with the maximal curvature, i.e., two types of maneuvers CCC and CSC and the solution can be found analytically (Dubins, 1957)

The main difficulty is to determine the vehicle headings for a given sequence of waypoints

Dubins Touring Problem – DTP

- For a sequence of the n waypoint locations $P = (p_1, \dots p_n)$, $p_i \in \mathbb{R}^2$, the **Dubins Touring Problem** (**DTP**) stands to determine the **optimal headings** $T = \{\theta_1, \dots, \theta_n\}$ at the waypoints q_i such that

$$\begin{aligned} \textit{minimize}_T \quad & \mathcal{L}(T, P) = \sum_{i=1}^{n-1} \mathcal{L}(q_i, q_{i+1}) + \mathcal{L}(q_n, q_1) \\ \textit{subject to} \quad & q_i = (p_i, \theta_i), \;\; \theta_i \in [0, 2\pi), \;\; p_i \in P, \end{aligned}$$

  where $\mathcal{L}(q_i, q_j)$ is the length of the Dubins maneuver connecting q_i with q_j
- The DTP is a continuous optimization problem
- The term $\mathcal{L}(q_n, q_1)$ is for possibly closed tour that can be for example requested in the TSP with Dubins vehicle, a.k.a. DTSP

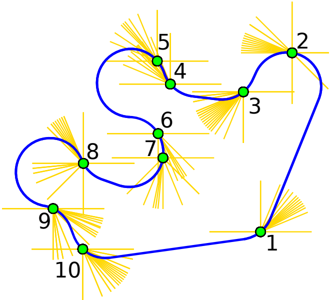
  On the other, the DTP can also be utilized for open paths such as solutions of the OP with Dubins vehicle
- In some cases, it may be suitable to relax the heading at the first/last locations in finding closed tours (i.e., solving DTSP)

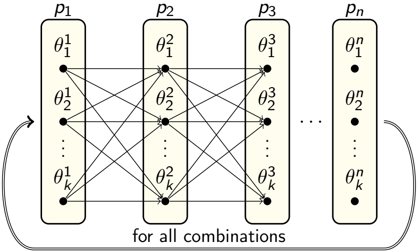
Sampling-based Solution of the DTP

- For a closed sequence of the waypoint locations

$$P = (p_1, \ldots, p_n)$$

- We can sample possible heading values at each location i into a discrete set of k headings, i.e., $\Theta^i = \{\theta^i_1, \ldots, \theta^i_k\}$ and create a graph of all possible Dubins maneuvers



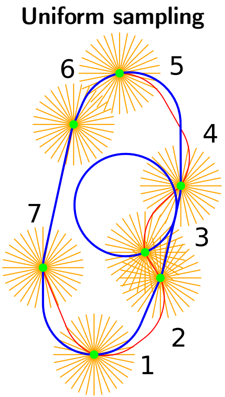


- For a set of heading samples, the optimal solution can be found by a forward search of the graph in $O(nk^3)$

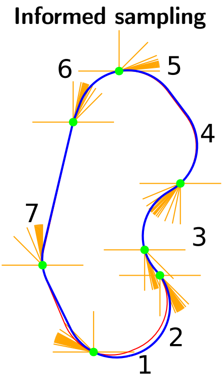
  For open sequence we do not need to evalute all possible initial headings, and the complexity is $O(nk^2)$

- **The key is to determined the most suitable heading samples per each waypoint**

Example of Heading Sampling – Uniform vs. Informed



$N = 224,\ T_{cpu} = 128$ ms
$\mathcal{L} = 19.8,\ \mathcal{L}_U = 13.8,$



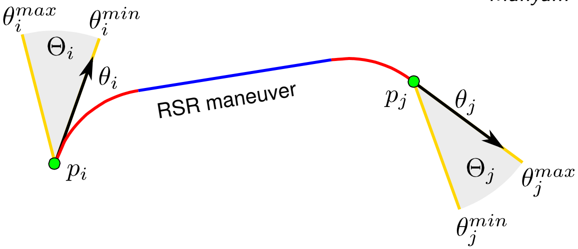
$N = 128,\ T_{cpu} = 76$ ms
$\mathcal{L} = 14.4,\ \mathcal{L}_U = 14.2,$

- N is the total number of samples, i.e., 32 samples per waypoint for uniform sampling
- $\mathcal{L}$ is the length of the tour (blue) and $\mathcal{L}_U$ is the lower bound (red) determined as a solution of the **Dubins Interval Problem** (**DIP**)

Dubins Interval Problem (DIP)

- **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, the leaving interval Θ_i at p_i and the arrival interval Θ_j at p_j are consider (not a single heading value)
- The optimal solution can be found analytically

Manyam et al. (2015)

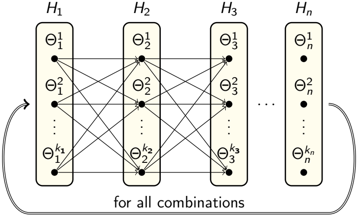


- Solution of the DIP is a tight lower bound for the DTP
- Solution of the DIP is not a feasible solution of the DTP

Notice, for $\Theta_i = \Theta_j = \langle 0, 2\pi)$ the optimal maneuver for DIP is a straight line segment

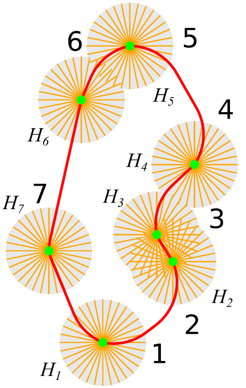
Lower Bound of the DTP

- For a discrete set of heading intervals $\mathcal{H} = \{H_1, \ldots, H_n\}$, where $H_i = \{\Theta_i^1, \Theta_i^2, \ldots, \Theta_i^{k_i}\}$, a similar graph as for the DTP can be constructed with the edge cost determined by the solution of the associated DIP



- The forward search of the graph with dense samples provides a tight lower bound of the DTP

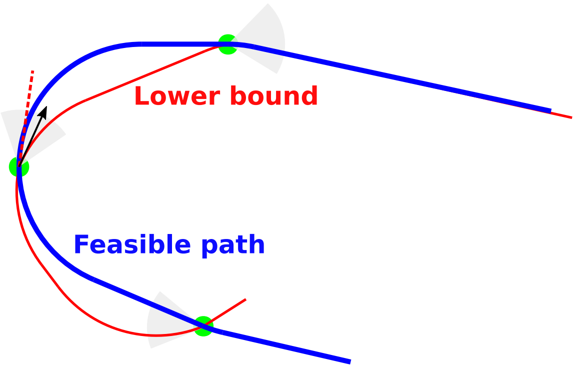
Manyam and Rathinam, 2015



Lower Bound and Feasible Solution of the DTP

- The arrival and departure angles may not be the same

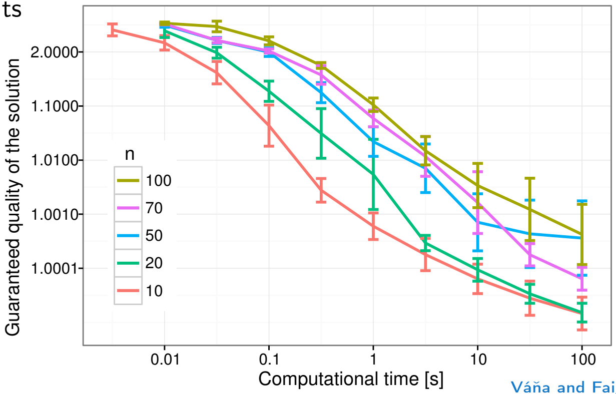
The lower bound solution is not a feasible solution of the DTP



- DTP solution – use any particular heading of each interval in the lower bound solution

The DIP-based Sampling of Headings in the DTP

- A similar forward search graph as for the DTP can be used for heading intervals instead of particular headings
- Using DIP for a sequence of waypoints 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 an estimation of the solution quality, e.g., for problems with n waypoints



Váňa and Faigl (2016)

Informed Sampling of Headings in Solution of the DTP

- Iterative refinement of the heading intervals $\mathcal{H}$ up to the angular resolution ϵ_{req}
- The angular resolution is gradually decreased for the most promising intervals
- `refineDTP` – divide the intervals of the lower bound solution
- `solveDTP` – solve DTP using the heading from the refined intervals

Algorithm 1: Iterative Informed Sampling-based DTP Algorithm

Vstup: P – Target locations to be visited
Vstup: ϵ_{req} – Requested angular resolution
Vstup: α_{req} – Requested quality of the solution
Výstup: T – A tour visiting the targets

```
ε ← 2π                          // initial angular resolution;
H ← createIntervals(P, ε)       // initial intervals;
L_L ← 0                         // init lower bound;
L_U ← ∞                         // init upper bound;
while ε > ε_req and L_U/L_L > α_req do
    ε ← ε/2;
    (H, L_L) ← refineDTP(P, ε, H);
    (T, L_U) ← solveDTP(P, H);
end
return T;
```

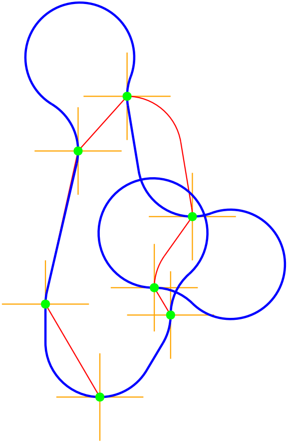
Faigl, Váňa et al. (2017)

- It simultaneously provides **feasible** and **lower bound** solutions of the DTP

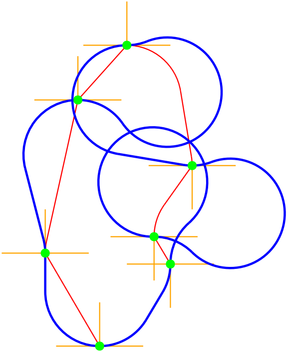
The lower bound provides a tight estimation of the solution quality

- The first solution is provided very quickly – **any-time algorithm**

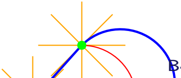
Uniform vs Informed Sampling

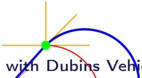


$\epsilon = 2\pi/4$, $N = 28$, $T_{CPU} = 8$ ms
$\mathcal{L} = 27.9$, $\mathcal{L}_U = 13.2$



$\epsilon = 2\pi/4$, $N = 21$, $T_{CPU} = 8$ ms
$\mathcal{L} = 29.9$, $\mathcal{L}_U = 13.2$



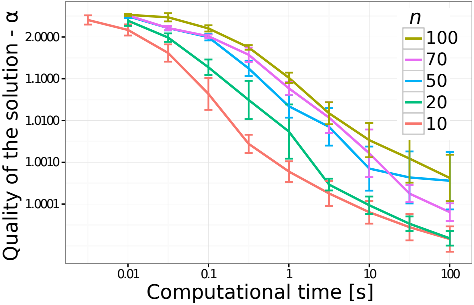


Results and Comparison with Uniform Sampling

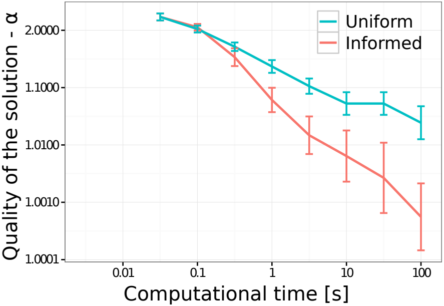
- Random instances of the DTSP with a sequence of visits to the targets determined as a solution of the Euclidean TSP
- The waypoints placed in a squared bounding box with the side $s = (\rho\sqrt{n})/d$ for the $\rho = 1$ and density $d = 0.5$

It matters on the Density of targets!

Quality of solution for increasing n

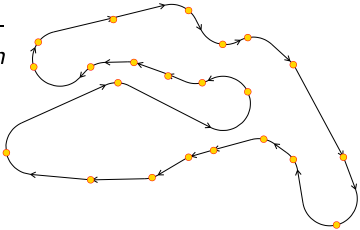


Comparision with the uniform sampling



- The Informed sampling-based approach provides solutions up to 0.01% from the optima
- A solution of the DTP is a fundamental bulding block for **routing problems with Dubins vehicle**

Dubins Traveling Salesman Problem (DTSP)



1. Determine a closed shortest Dubins path visiting each location $p_i \in P$ of the given set of n locations $P = \{p_1, \ldots, p_n\}$, $p_i \in \mathbb{R}^2$
2. Permutation $\Sigma = (\sigma_1, \ldots, \sigma_n)$ of visits

   Sequencing part of the problem
3. Headings $\Theta = \{\theta_{\sigma_1}, \theta_{\sigma_2}, \ldots, \theta_{\sigma_n}\}$ for $p_{\sigma_i} \in P$

   Continuous optimization

- **DTSP** is an optimization problem over all possible **permutations Σ** and **headings Θ** in the states $(q_{\sigma_1}, q_{\sigma_2}, \ldots, q_{\sigma_n})$ such that $q_{\sigma_i} = (p_{\sigma_i}, \theta_{\sigma_i})$

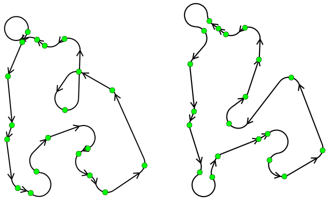
$$\begin{aligned} \textit{minimize}_{\Sigma,\Theta} \quad & \sum_{i=1}^{n-1} \mathcal{L}(q_{\sigma_i}, q_{\sigma_{i+1}}) + \mathcal{L}(q_{\sigma_n}, q_{\sigma_1}) && (1) \\ \textit{subject to} \quad & q_i = (p_i, \theta_i) \; i = 1, \ldots, n, && (2) \end{aligned}$$

where $\mathcal{L}(q_{\sigma_i}, q_{\sigma_j})$ is the length of Dubins path between q_{σ_i} and q_{σ_j}.

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



Two fundamental approaches can be found in literature

- **Decoupled** approach based on a given sequence of the locations

  E.g., found by a solution of the Euclidean TSP

- **Sampling-based** approach with sampling of the headings at the locations into discrete sets of values and considering the problem as the variant of the **Generalized TSP**

Besides, further approaches are

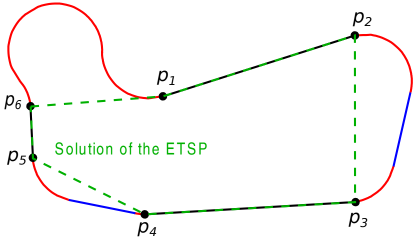
- Genetic and memetic techniques (evolutionary algorithms)
- Unsupervised learning based approaches

Decoupled Solution of the DTSP – Alternating Algorithm

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

Savla et al. (2005)

1. Solve the related Euclidean TSP
   Relaxed motion constraints
2. Establish headings for even edges using straight line segments
3. Determine optimal maneuvers for odd edges using the analytical form for Dubins maneuvers
   Headings are known



Courtesy of P. Váňa

DTSP with the Given Sequence of the Visits to the Targets

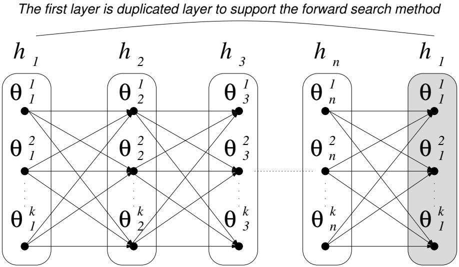
- If the sequence of the visits Σ to the target locations is given
- the problem is to determine the optimal heading at each location
- and the problem becomes the **Dubins Touring Problem** (**DTP**)

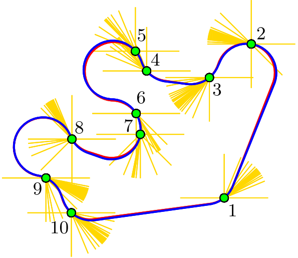
Váňa and Faigl (2016)

- 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 $h_i = \{\theta_1^1, \ldots, \theta_1^k\}$.

- 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 $\{h_1, \ldots, h_n\}$, we can find an optimal headings and thus, the optimal solution of the DTP.

DTSP as a Solution of the DTP





- The edge cost corresponds to the length of Dubins maneuver
- Better solution of the DTP can be found for more samples, which 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 to sample the headings in a "smart" way, i.e., guided sampling using lower bound of the DTP

- What is the solution quality? Is there a tight lower bound?

  Yes, the lower bound can be computed as a solution of Dubins Interval Problem (DIP)

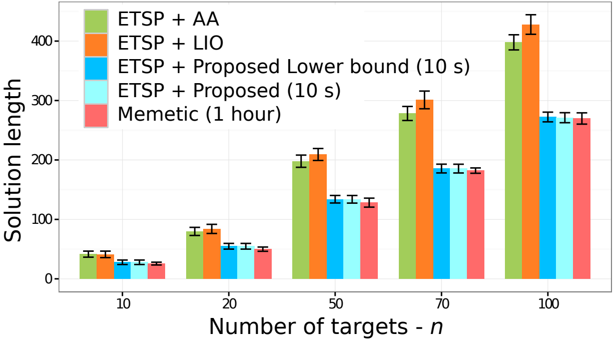
DTP Solver in Solution of the DTSP

- The solution of the DTP can be used to solve DTSP for the given sequence of the waypoints

E.g., determined as a solution of the Euclidean TSP as in the Alternating Algorithm

- Comparision with the Alternating Algorithm (AA), Local Iterative Optimization (LIO), and Memetic algorithm

AA – Savla et al., 2005, LIO – Váňa & Faigl, 2015, Memetic – Zhang et al. 2014



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**)

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

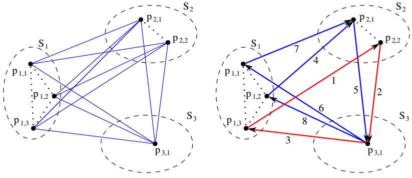
> The problem is to determine a shortest tour in a graph that visits all specified subsets of the graph's vertices

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



GTSP GATSP

Dubins Traveling Salesman Problem with Neighborhoods

- In surveillance planning, it may be required to visit a set of target regions $\boldsymbol{G} = \{R_1, \ldots, R_n\}$ 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, waypoint locations P have to be determined

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

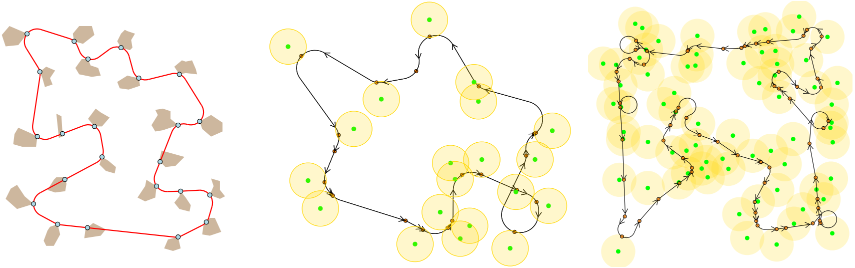
$$\text{minimize}_{\Sigma,\Theta,P} \quad \sum_{i=1}^{n-1} \mathcal{L}(q_{\sigma_i}, q_{\sigma_{i+1}}) + \mathcal{L}(q_{\sigma_n}, q_{\sigma_1}) \tag{3}$$

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

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

DTSPN – Approches and Examples of Solution

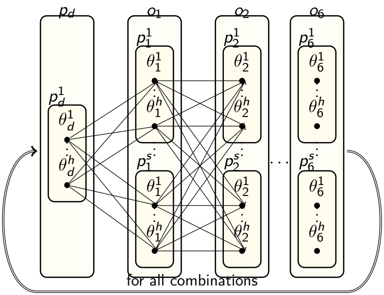
- Similarly to the DTSP, also DTSPN can be addressed by
  - **Decoupled approaches** for which a sequence of visits to the regions can be found as a solution of the ETSP(N)
  - **Sampling-based approaches** and transformation to the GTSP
    - Clusters of sampled waypoint locations each with sampled possible heading values
  - **Soft-computing** techniques such as memetic algorithms
  - **Unsupervised learning** techniques

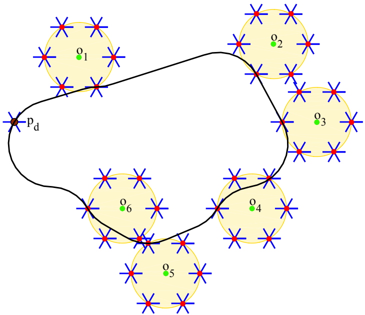


Váňa and Faigl (IROS 2015), Faigl and Váňa (ICANN 2016, IJCNN 2017)

DTSPN – Decoupled Approach

1. Determine a sequence of visits to the n target regions as the solution of the ETSP
2. Sample possible waypoint locations and for each such a location sample possible heading values, e.g., s locations per each region and h heading per each location
3. Construct a search graph and determine a solution in $O(n(sh)^3)$
4. An example of the search graph for $n = 6$, $s = 6$, and $h = 6$





Dubins Touring Region Problem (DTRP)

DTSPN – Local Iterative Optimization (LIO)

- Instead of sampling into a discrete set of waypoint locations ech with sampled possible headings, we can perform local optimization, e.g., hill-climbing technique
- At each waypoint location p_i, the heading can be $\theta_i \in [0, 2\pi)$
- A waypoint location p_i can be parametrized as a point on the bounday of the respective region R_i, i.e., as a parameter $\alpha \in [0, 1)$ measuring a normalized distance on the boundary of R_i
- The multi-variable optimization is treated independently for each particular variable θ_i and α_i iteratively

Algorithm 2: Local Iterative Optimization (LIO) for the DTSPN

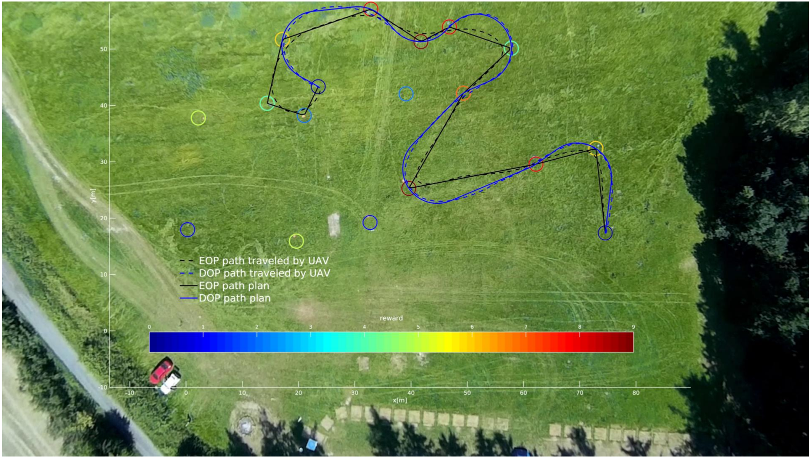
Data: Input sequence of the goal regions $\boldsymbol{G} = (R_{\sigma_1}, \ldots, R_{\sigma_n})$, for the permutation Σ
Result: Waypoints $(q_{\sigma_1}, \ldots, q_n)$, $q_i = (p_i, \theta_i)$, $p_i \in \delta R_i$
initialization() // random assignment of $q_i \in \delta R_i$;
while *global solution is improving* **do**
  for *every* $R_i \in \boldsymbol{G}$ **do**
    θ_i := optimizeHeadingLocally(θ_i);
    α_i := optimizePositionLocally(α_i);
    q_i := checkLocalMinima(α_i, θ_i);
  end
end

Váňa and Faigl (IROS 2015)

Data Collection / Surveillance Planning with Travel Budget

- Visit the most important targets because of limited travel budget
- The problem can be formulated as the **Orienteering Problem** with Dubins vehicle, a.k.a. **Dubins Orienteering Problem** (**DOP**)

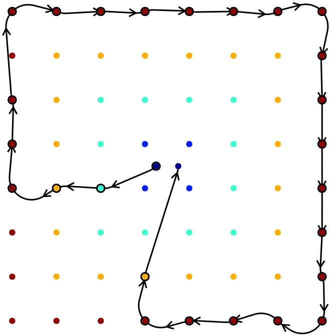
Robert Pěnička, Jan Faigl, Petr Váňa and Martin Saska, RA-L 2017



http://mrs.felk.cvut.cz/icra17dop

Dubins Orienteering Problem

- Curvature-constrained data collection path respecting Dubins vehicle model with the minimal turning radius ρ and constant forward velocity v
- The path is a sequence of waypoints $q_i \in SE(2)$, $q = (s, \theta)$, $\theta \in \mathbb{S}^1$.
- In addition to S_k, k, Σ (OP) determine headings $\Theta = (\theta_{\sigma_1}, \ldots, \theta_{\sigma_k})$ such that



$$\underset{k, S_k, \Sigma}{\text{maximize}} \quad R = \sum_{i=1}^{k} r_{\sigma_i}$$

$$\text{subject to} \quad \sum_{i=2}^{k} \mathcal{L}(q_{\sigma_{i-1}}, q_{\sigma_i}) \leq T_{max},$$

$$q_{\sigma_i} = (s_{\sigma_i}, \theta_{\sigma_i}), s_{\sigma_i} \in S, \theta_{\sigma_i} \in \mathbb{S}$$

$$s_{\sigma_1} = s_1, s_{\sigma_k} = s_n$$

The problem combines discrete combinatorial optimization (OP) with the continuous optimization for determining the vehicle headings

Variable Neighborhood Search (VNS)

- **Variable Neighborhood Search** (**VNS**) is a general metaheuristic for combinatorial optimization (routing problems)

  Hansen, P. and Mladenović, N. (2001): Variable neighborhood search: Principles and applications. European Journal of Operational Research.
- The VNS is based on **shake** and **local search** procedures
  - Shake procedure aims to escape from local optima by changing the solution within the neighborhoods $N_{1,\ldots,k_{max}}$

    The neighborhoods are particular operators
  - Local search procedure searches fully specific neighborhoods of the solution using l_{max} predefined operators

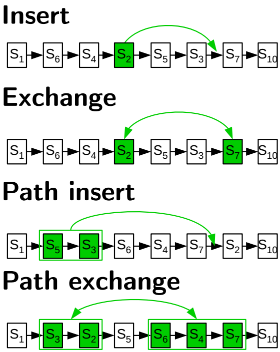
Variable Neighborhood Search (VNS) for the DOP

- The solution is the first k locations of the sequence of all target locations satisfying T_{max}

VNS for the OP – Sevkli, Z. et al. (2006)

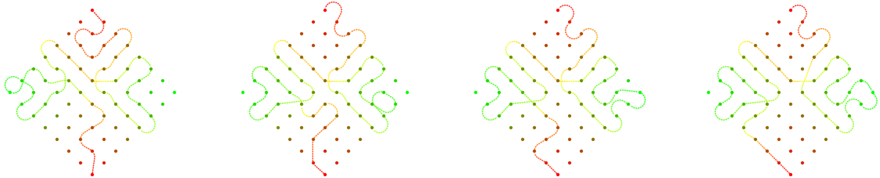
- It is an improving heuristics, i.e., an initial solution has to be provided
- A set of predefined neighborhoods are explored to find a better solution

- **Shake** – explores the configuration space and escape from a local minima using
  - **Insert** – moves one random element
  - **Exchange** – exchanges two random elements
- **Local Search** – optimizes the solution
  - **Path insert** – moves a random sub-sequence
  - **Path exchange** – exchanges two random sub-sequences
- **Randomized VNS** – examines only n^2 changes in the *Local Search* procedure in each iteration



Evolution of the VNS Solution to the DOP

Initial solution	4710*th* iteration (4*th* improvement)	4790*th* iteration (12*th* improvement)	5560*th* iteration (16*th* improvement)



$T_{CPU} = 10.9$ s, $\mathcal{L} = 79.6,\ R = 960$	$T_{CPU} = 144.8$ s, $\mathcal{L} = 79.7,\ R = 990$	$T_{CPU} = 147.3$ s, $\mathcal{L} = 79.3,\ R = 1008$	$T_{CPU} = 170.0$ s, $\mathcal{L} = 79.1,\ R = 1050$

Possible Solutions of the Dubins Orienteering Problem

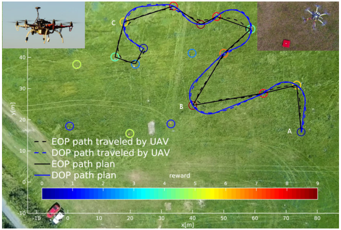
1. Solve the Euclidean OP (EOP) and then determine Dubins path

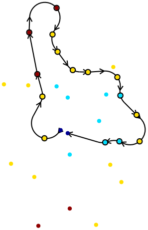
   The final path may exceed the budget and the vehicle can miss the locations because of motion control

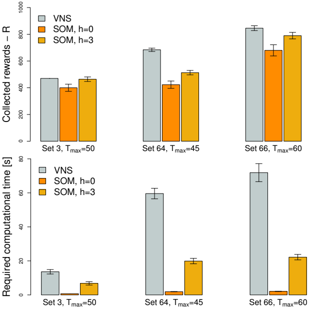
2. Directly solve the **Dubins Orienteering Problem (DOP)**, e.g.,
   - Sample possible heading values and use Variable Neighborhood Search (VNS)

     Pěnička, Faigl, Váňa, Saska (RA-L 2017)
   - Unsupervised learning based on Self-Organizing Maps (SOM)

     Faigl, (WSOM+ 2017)



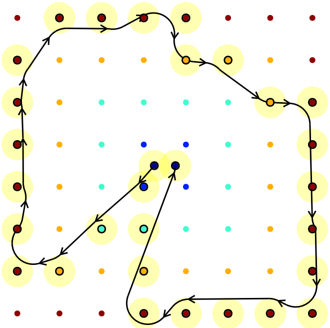




VNS-based approach provides better solutions than SOM, but it tends to be more computationally demanding

Dubins Orienteering Problem with Neighborhoods

- Curvature-constrained path respecting Dubins vehicle model
- Each waypoint consists of location $p \in \mathbb{R}^2$ and the heading $\theta \in \mathbb{S}^1$
- In addition to S_k, k, Σ determine locations $P_k = (p_{\sigma_1}, \ldots, p_{\sigma_k})$ and headings $\Theta = (\theta_{\sigma_1}, \ldots, \theta_{\sigma_k})$ such that



$$\underset{k,S_k,\Sigma}{\text{maximize}} \quad R = \sum_{i=1}^{k} r_{\sigma_i}$$

$$\text{subject to} \quad \sum_{i=2}^{k} \mathcal{L}(q_{\sigma_{i-1}}, q_{\sigma_i}) \leq T_{max},$$

$$q_{\sigma_i} = (p_{\sigma_i}, \theta_{\sigma_i}), p_{\sigma_i} \in \mathbb{R}^2, \theta_{\sigma_i} \in \mathcal{S}^1$$

$$||p_{\sigma_i}, s_{\sigma_i}|| \leq \delta, s_{\sigma_i} \in S_k$$

$$p_{\sigma_1} = s_1, p_{\sigma_k} = s_n$$

We need to solve the continuous optimization for determining the vehicle heading at each waypoint and the waypoint locations $P_k = \{p_{\sigma_1}, \ldots, p_{\sigma_k}\}$, $p_{\sigma_i} \in \mathbb{R}^2$

Variable Neighborhoods Search (VNS) for the DOPN

Algorithm 3: VNS based method for the DOPN

```
Input  : S – Set of the target locations
Input  : T_max – Maximal allowed budget
Input  : o – Initial number of position waypoints for each
         target
Input  : m – Initial number of heading values for each
         waypoints
Input  : r_i – Local waypoint improvement ratio
Input  : l_max – Maximal neighborhood number
Output: P – Found data collecting path
S_r ← getReachableLocations(S, T_max)
P ← createInitialPath(S_r, T_max)                    // greedy
while Stopping condition is not met do
    l ← 1
    while l ≤ l_max do
        P' ← shake(P, l)
        P'' ← localSearch(P', l, r_i)
        if L_d(P'') ≤ T_max and
        [[R(P'') > R(P)] or [R(P'') == (P) and
        L_d(P'') < L_d(P)L_d(P'')]] then
            P ← P''
            l ← 1
        else
            l ← l + 1
        end
    end
end
```

The particular l for the individual operators of the **shake** procedure are:

- **Waypoint Shake** ($l = 1$)
- **Path Move** ($l = 2$)
- **Path Exchange** ($l = 3$)

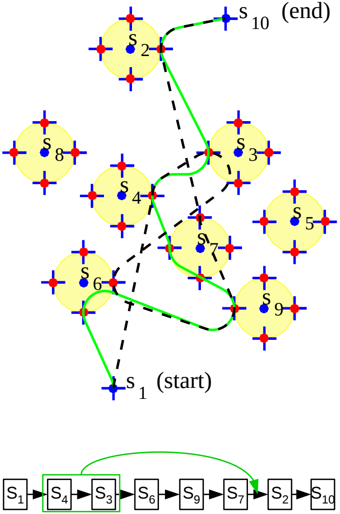
The **local search** procedure consists of three operators and the particular l for the individual operators of the **local search** procedure are:

- **Waypoint Improvement** ($l = 1$)
- **One Point Move** ($l = 2$)
- **One Point Exchange** ($l = 3$)

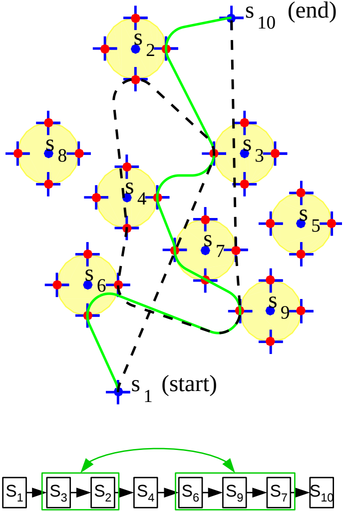
Pěnička, R., Faigl, J., Saska, M., Váňa, P. (2017)

VNS for DOPN – Example of the Shake Operators

Path Move



Path Exchange



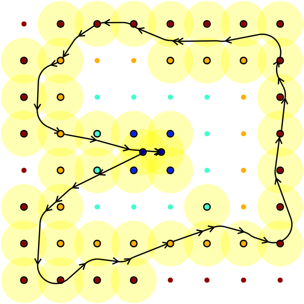
Comparision of the DOPN Solvers

- VNS-based DOPN solver with $s = 16$ sampled waypoint locations per sensor and $h = 16$ heading samples per waypoint location

  Pěnička, Faigl, et al. (ICUAS 2017)
- SOM-based DOPN solver with $h = 3$ Faigl, Pěnička (IROS 2017)
- Aggregate results using average relative percentage error (ARPE) and relative percentage error (RPE) to the reference (best found) solution

Problem set	VNS-based		SOM-based ($h = 3$)		
	$\overline{\text{ARPE}}$	T_{cpu}* [s]	$\overline{\text{RPE}}$	$\overline{\text{ARPE}}$	T_{cpu} [s]
Set 3, $\delta = 0.0$	1.0	1,178.9	3.6	7.4	7.0
Set 3, $\delta = 0.5$	0.9	13,273.3	6.6	10.6	7.9
Set 3, $\delta = 1.0$	0.5	13,304.4	5.5	9.2	8.3
Set 64, $\delta = 0.0$	1.9	5,272.2	17.4	23.8	17.9
Set 64, $\delta = 0.5$	2.8	13,595.6	18.7	24.2	20.2
Set 64, $\delta = 1.0$	1.3	13,792.3	9.9	15.2	22.2
Set 66, $\delta = 0.0$	1.5	6,546.6	3.6	9.1	22.9
Set 66, $\delta = 0.5$	1.4	13,650.1	6.7	11.8	25.5
Set 66, $\delta = 1.0$	3.2	13,824.5	16.1	21.3	26.7

*The results have been obtained with a grid Xeon CPUs running at 2.2 GHz to 3.4 GHz due to computational requirements.

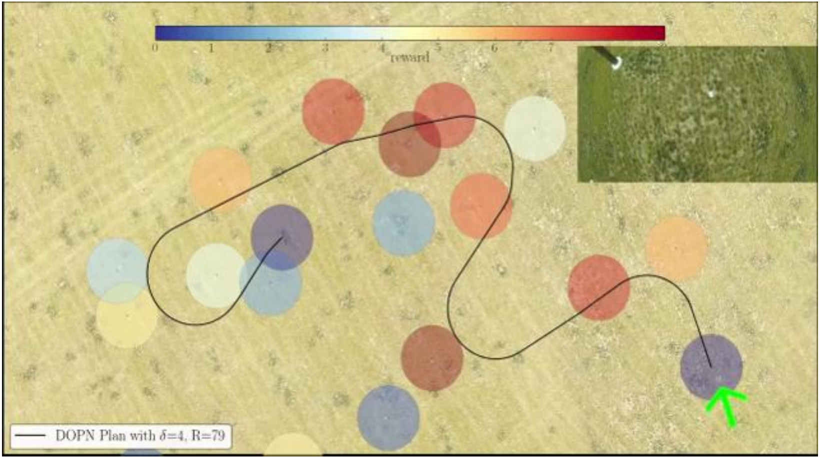


$\rho = 1.0$, $\delta = 1.25$, $R = 1185$

DOPN – Example of Solution and Practical Deployment

- VNS-based solution of the DOPN

Robert Pěnička, Jan Faigl, Martin Saska and Petr Váňa, ICUAS 2017



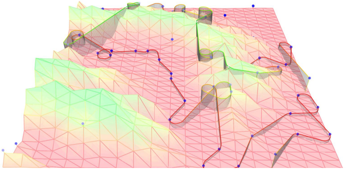
http://mrs.felk.cvut.cz/jint17dopn

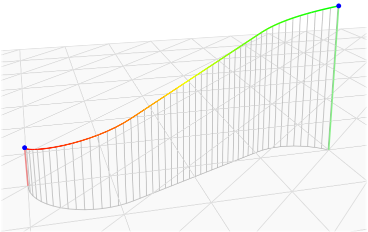
3D Data Collection Planning with Dubins Airplane Model

- Dubins Airplane model describes the vehicle state $q = (p, \theta, \psi)$, $p \in \mathbb{R}^3$ and $\theta, \psi \in \mathbb{S}^1$ as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta} \end{bmatrix} = v \begin{bmatrix} \cos\theta \cdot \cos\psi \\ \sin\theta \cdot \cos\psi \\ \sin\psi \\ u_\theta \cdot \rho^{-1} \end{bmatrix} \qquad (5)$$

Chitsaz, H., LaValle, S.M. (2017)

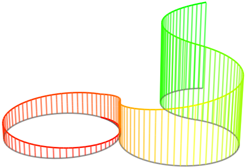




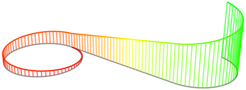
- Constant forward velocity v, the minimal turning radius ρ, and limited pitch angle, i.e., $\psi \in [\psi_{min}, \psi_{max}]$
  - u_θ controls the vehicle heading, $|u_\theta| \leq 1$, and v is the forward velocity
  - Generation of the 3D trajectory is based on the 2D Dubins maneuver
  - If altitude changes are too high, additional helix segments are inserted

DTSPN in 3D

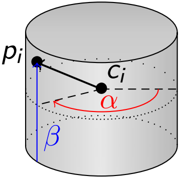
- Using the same principles as for the DTSPN in 2D, we can generalize the approaches for 3D planning using the Dubins Airplane model instead of simple Dubins vehicle
- The regions can be generalized to 3D and the problem can be addressed by decoupled or sampling-based approaches, i.e., using GATSP formulation
- In the case of LIO, we need a parametrization of the possible waypoint location, e.g.,

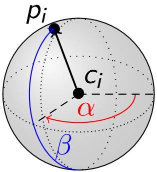


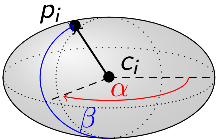
CCC maneuver



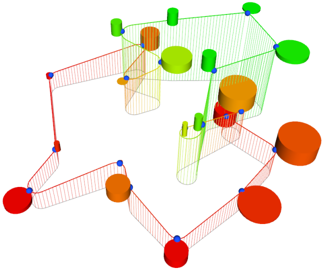
CSC maneuver







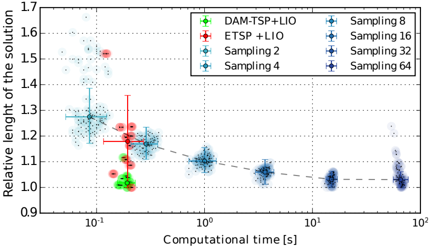
Solutions of the 3D-DTSPN

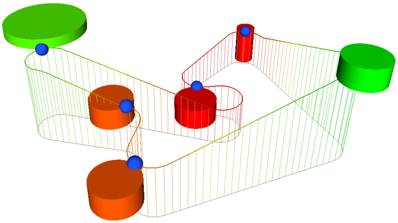


Algorithm 4: LIO-based Solver for 3D-DTSPN

Data: Regions $\mathcal{R}$
Result: Solution represented by $\mathcal{Q}$ and Σ
$\Sigma \leftarrow$ getInitialSequence($\mathcal{R}$);
$\mathcal{Q} \leftarrow$ getInitialSolution($\mathcal{R}, \Sigma$);
while *terminal condition* **do**
  $\mathcal{Q} \leftarrow$ optimizeHeadings($\mathcal{Q}, \mathcal{R}, \Sigma$);
  $\mathcal{Q} \leftarrow$ optimizeAlpha($\mathcal{Q}, \mathcal{R}, \Sigma$);
  $\mathcal{Q} \leftarrow$ optimizeBeta($\mathcal{Q}, \mathcal{R}, \Sigma$);
end
return $\mathcal{Q}, \Sigma$;

- **Solutions based on LIO (ETSP+LIO), TSP with the travel cost according to Dubins Airplane Model (DAM-TSP+LIO), and sampling-based approach with transformation of the GTSP to the ATSP solved by LKH**



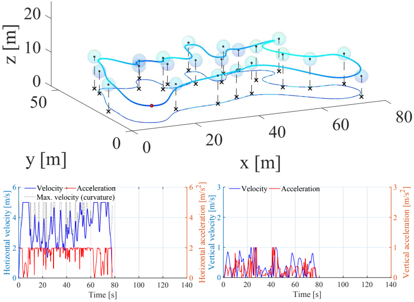


Váňa and Faigl (2017)

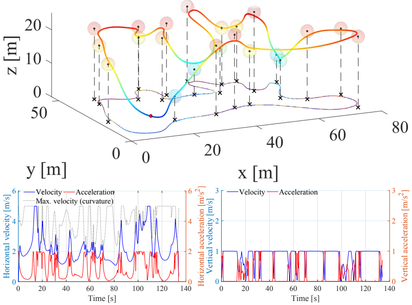
3D Surveillance Planning

- Parametrization of smooth 3D multi-goal trajectory as a sequence of Bézier curves
- Unsupervised learning for the TSPN can be generalized for such trajectories
- During the solution of the sequencing part of the problem, we can determine a velocity profile along the curve and compute the so-called *Travel Time Estimation* (TTE)
- Bézier curves better fit the limits of the multi-rotor UAVs that are limited by the maximal accelerations and velocities rather than minimal turning radius as for Dubins vehicle

Low altitude differences



High altitude differences



Faigl and Váňa (2017)

- **Low altitude differences saturate horizontal velocity while high altitudes changes saturate vertical velocity**

Part II

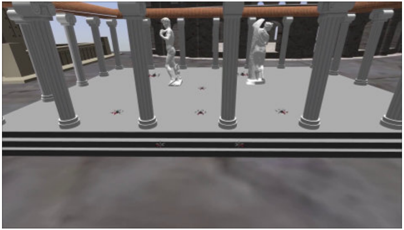
Part 2 – Data Collection Planning for Surveillance Missions

Motivation

- There is a framework for testing and evaluation of UAVs control strategies developed and maintained by the winners of the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) 2017

  Multi-robot Systems (MRS) group

  http://mrs.felk.cvut.cz



https://www.youtube.com/watch?v=ju3YbCtXpEw

- The framework allows a direct evaluation of the planned trajectories, i.e., Dubins trajectories, in the same way for the simulator and also for real vehicles
- It provides an unique opportunity to become more familiar with multi-rotor unmanned aerial vehicles and gain experience with practical deployment of the planned trajectories to UAVs
- Full support of the evaluation environment is provided together with setuped computers at the dedicated computer lab of the MRS (KN:E-118)
- A practical deployment on real UAVs would be possible during the first campaigns in spring 2018

Assignment – HW03b

Topic: Data Collection Planning for Surveillance Missions

Goal: Solve data collection planning problem formulated as the DTSP (DTSPN) and deploy the planned path to the model of UAVs and eventually experimentally verify the paths using real UAV

Assignment: https://cw.fel.cvut.cz/wiki/courses/b4m36uir/hw/hw03b

Up to additional **15 points** can be gained for the implementation of the DTS and/or DTSPN, and execution of the trajectories in the MRS simulation framework

- Implement a solution of the DTSP, e.g., one of the following methods
  - (2 points) for simple ETSP and Alternating Algorithm (AA), a.k.a ETSP+AA;
  - (6 points) become familiar with the MRS simulation framework and deploy the planned trajectories within the simulator
- Voluntary implementation of the DTSP and DTSPN sampling-based solvers
  - (4 points) ETSP+DTP (forward search graph) or GATSP→ATSP and solution using LKH
  - (3 points) Extension of the DTSP to DTSPN, e.g., forward search graph for DTP generalized for the DTRP or GATSP based approach

Summary of the Lecture

Topics Discussed

- Dubins vehicles and planning – Dubins maneuvers
- Dubins Interval Problem (DIP)
- Dubins Touring Problem (DTP)
- Dubins Traveling Salesman Problem (DTSP) and Dubins Traveling Salesman with Neighborhoods (DTSPN)
  - Decoupled approaches – Alternating Algorithm
  - Sampling-based approaches – GATSP
- Dubins Orienteering Problem (OP) and Dubins Orienteering Problem with Neighborhoods (DOPN)
- Data collection and surveillance planning in 3D

- Next: Multirobot Path Planning (MPP)