Data Collection Planning – Multi-Goal Planning

Jan Faigl

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

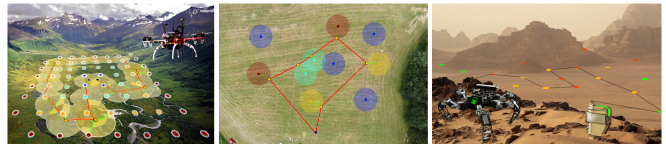
Lecture 06

B4M36UIR – Artificial Intelligence in Robotics

Overview of the Lecture

Data Collection Planning as a Solution of the Routing Problem

- Provide cost-efficient path to collect **all** or the **most valuable data** (measurements) with **shortest possible path/time** or under **limited travel budget**.



Visiting all locations

- The **Traveling Salesman Problem** (**TSP**).
- Well-studied combinatorial routing problem with many existing approaches.

Limited travel budget

- We need to prioritize some locations – routing problem with profits.
- The **Orienteering Problem** (**OP**).

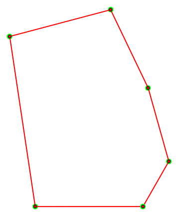
- In both problems, we can improve the solution by exploiting non-zero sensing range.

Data Collection Planning as the Traveling Salesman Problem

- Let S be a set of n sensor locations $S = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n\}$, $\boldsymbol{s}_i \in \mathbb{R}^2$ and $c(\boldsymbol{s}_i, \boldsymbol{s}_j)$ is a cost of travel from $\boldsymbol{s}_i$ to $\boldsymbol{s}_j$.
- The problem is to determine a closed tour visiting each $\boldsymbol{s} \in S$ such that the total tour length is minimal, i.e., **determine a sequence of visits** $\Sigma = (\sigma_1, \ldots, \sigma_n)$.

$$\text{minimize}_{\Sigma} \quad L = \left(\sum_{i=1}^{n-1} c(\boldsymbol{s}_{\sigma_i}, \boldsymbol{s}_{\sigma_{i+1}})\right) + c(\boldsymbol{s}_{\sigma_n}, \boldsymbol{s}_{\sigma_1})$$

$$\text{subject to} \quad \Sigma = (\sigma_1, \ldots, \sigma_n), 1 \le \sigma_i \le n, \sigma_i \ne \sigma_j \text{ for } i \ne j$$



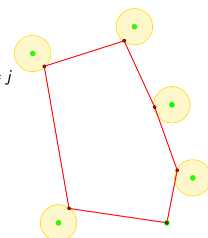
- The TSP is a **pure combinatorial optimization** problem to find the best sequence of visits Σ.

Data Collection Planning with Non-zero Sensing Range – the Traveling Salesman Problem with Neighborhood

- The travel cost can be saved by **remote data collection** using wireless communication or range measurements; instead visiting $\boldsymbol{s} \in S$, we can visit $\boldsymbol{p}$ within **δ distance** from $\boldsymbol{s}$.
- In addition to Σ, we need to determine n waypoint locations $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$.

$$\text{minimize}_{\Sigma,P} \quad L = \left(\sum_{i=1}^{n-1} c(\boldsymbol{p}_{\sigma_i}, \boldsymbol{p}_{\sigma_{i+1}})\right) + c(\boldsymbol{p}_{\sigma_n}, \boldsymbol{p}_{\sigma_1})$$

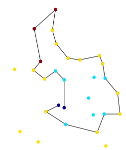
$$\text{subject to} \quad \Sigma = (\sigma_1, \ldots, \sigma_n), 1 \le \sigma_i \le n, \sigma_i \ne \sigma_j \text{ for } i \ne j$$
$$P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}, \|(p_i, s_i)\| \le \delta$$



- The problem becomes a combination of combinatorial and **continuous optimization** with at least n-variables.
- The problem is a variant of the **TSP with Neighborhoods** or **Close Enough TSP** for disk-shaped neighborhoods.

Orienteering Problem (OP) – Routing with Profits

- Let each of n sensors $S = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n\}$, $\boldsymbol{s}_i \in \mathbb{R}^2$ be associated with a score ζ_i characterizing the reward if data from s_i are collected.
- The vehicles start at $\boldsymbol{s}_1$, terminates at $\boldsymbol{s}_n$, its travel cost between $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ is the Euclidean distance $|(\boldsymbol{p}_i - \boldsymbol{p}_j)|$, and it has **limited travel budget T_{max}**.
- The OP stands to determine a subset of k locations $S_k \subseteq S$ **maximizing the collected rewards** while the tour cost visiting S_k does not exceed T_{max}.



- The OP combines the problem of determining the most valuable locations S_k with finding the shortest tour T visiting the locations S_k.

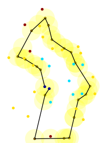
$$\text{maximize}_{k,S_k,\Sigma} \quad R = \sum_{i=1}^{k} \zeta_{\sigma_i}$$

$$\text{subject to} \quad \sum_{i=2}^{k} |(\boldsymbol{s}_{\sigma_{i-1}} - \boldsymbol{s}_{\sigma_i})| \le T_{max}$$
$$\text{and } \boldsymbol{s}_{\sigma_1} = \boldsymbol{s}_1, \boldsymbol{s}_{\sigma_k} = \boldsymbol{s}_n.$$

- Optimal solution (ILP-based) and heuristics have been proposed.
  - 4-phase heuristic algorithm (Ramesh & Brown, 1991);
  - CGW proposed Chao, et al. 1996;
  - Guided local search algorithm (Vansteenwegen et al., 2009).
- Standard benchmarks have been established by Tsiligirides and Chao.

Data Collection with Limited Travel Budget OP with Neighborhoods (OPN) and Close Enough OP (CEOP)

- Data collection using wireless data transfer or remote sensing allows to reliably retrieve data within some sensing range δ.
- The OP becomes the **Orienteering Problem with Neighborhoods** (**OPN**).
- For the disk-shaped δ-neighborhood, we call it the **Close Enough OP** (**CEOP**).
- In addition to S_k and Σ, we need to determine the most suitable waypoint locations P_k that maximize the collected rewards and the path connecting P_k does not exceed T_{max}.



$$\text{maximize}_{k,P_k,\Sigma} \quad R = \sum_{i=1}^{k} \zeta_{\sigma_i}$$

$$\text{subject to} \quad \sum_{i=2}^{k} |(\boldsymbol{p}_{\sigma_{i-1}}, \boldsymbol{p}_{\sigma_i})| \le T_{max},$$
$$|(\boldsymbol{p}_{\sigma_i}, s_{\sigma_i})| \le \delta, \quad \boldsymbol{p}_{\sigma_i} \in \mathbb{R}^2,$$
$$\boldsymbol{p}_{\sigma_1} = \boldsymbol{s}_1, \boldsymbol{p}_{\sigma_k} = \boldsymbol{s}_n.$$

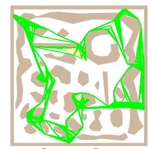
- OPN/CEOP has been firstly tackled by SOM-based approach. (Best, Faigl, Fitch, 2016)
- Later addressed by the GSOA and **Variable Neighborhoods Search** (**VNS**) (Pěnička, Faigl, Saska, 2016)
- and optimal solution of the discrete **Set OP**. (Pěnička, Faigl, Saska, 2019)
- The currently best performing method is based on the **Greedy Randomized Adaptive Search Procedure** (**GRASP**). (Štefaníková, Faigl 2020)

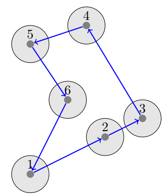
Approaches to the Close Enough TSP and TSP with Neighborhoods

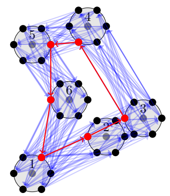
- **A direct solution** of the TSPN
  - Approximation algorithms for special cases with particular shapes of the neighborhoods.
    In general, the TSPN is APX-hard, and cannot be approximated to within a factor $2 - \epsilon$, $\epsilon > 0$, unless P=NP. (Safra, S., Schwartz, O. (2006))
  - Heuristic algorithms such as evolutionary techniques or unsupervised learning.
- **Decoupled approach**
  1. Determine sequence of visits Σ independently on the locations P.
     E.g., Solution of the TSP for the centroids of the (convex) neighborhoods.
  2. For the sequence Σ determine the locations P to minimize the total tour length, e.g.,
     - Solving the Touring polygon problem (TPP);
     - Sampling possible locations and use a forward search for finding the best locations;
     - Continuous optimization such as hill-climbing.
- **Sampling-based** approaches
  - Sample possible locations of visits within each neighborhood into a discrete set of locations.
  - Formulate the problem as the **Generalized Traveling Salesman Problem** (**GTSP**).

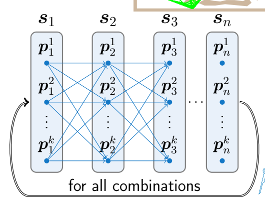
Decoupled Approach with Locations Sampling

- Solve the problem as a regular TSP using centroids of the regions (disks) to get the sequence of visits Σ.
- Sample each neighborhood with k samples (e.g., $k = 6$) and find the shortest tour by forward search in $\mathcal{O}(nk^2)$ for nk^2 edges in the sequence.
  - For k possible initial locations, the optimal solution can be found in $\mathcal{O}(nk^3)$.



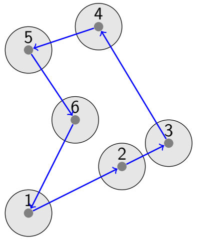


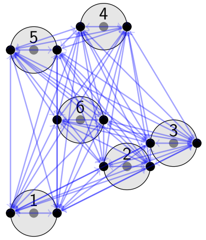




Sampling-based Solution of the TSPN

- For an unknown sequence of the visits to the regions, there are $\mathcal{O}(n^2k^2)$ possible edges.
- Finding the shortest path is NP-hard, we need to determine the sequence of visits, which is the solution of the TSP.

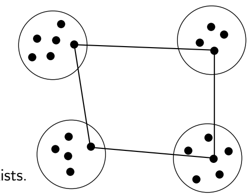




The descrite variant of the TSPN can be formulated as the GTSP

Generalized Traveling Salesman Problem (GTSP)

- For sampled neighborhoods into discrete sets of locations, we can formulate the problem as the **Generalized Traveling Salesman Problem** (GTSP). Also known as the Set TSP.
- For a set of n sets $S = \{S_1, \ldots, S_n\}$, each with particular set of locations (nodes) $S_i = \{s^i_1, \ldots, s^i_{n_i}\}$, determine the shortest tour visiting each set S_i.



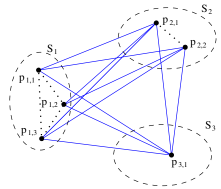
$$\text{minimize}_{\Sigma} \quad L = \left(\sum_{i=1}^{n-1} c(s^{\sigma_i}, s^{\sigma_{i+1}})\right) + c(s^{\sigma_n}, s^{\sigma_1})$$

$$\text{subject to} \quad \Sigma = (\sigma_1, \ldots, \sigma_n), 1 \le \sigma_i \le n, \sigma_i \neq \sigma_j \text{ for } i \neq j$$
$$s^{\sigma_i} \in S_{\sigma_i}, S_{\sigma_i} = \{s^{\sigma_i}_1, \ldots, s^{\sigma_i}_{n_{\sigma_i}}\}, S_{\sigma_i} \in S$$

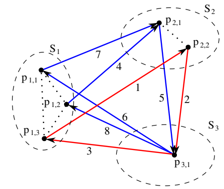
- Optimal ILP-based solution and heuristic algorithms exists.
  - **GLKH** – http://akira.ruc.dk/~keld/research/GLKH/
    Helsgaun, K (2015), *Solving the Equality Generalized Traveling Salesman Problem Using the Lin-Kernighan-Helsgaun Algorithm.*
  - **GLNS** – https://ece.uwaterloo.ca/~sl2smith/GLNS (in Julia)
    Smith, S. L., Imeson, F. (2017), *GLNS: An effective large neighborhood search heuristic for the Generalized Traveling Salesman Problem*, Computers and Operations Research.

Transformation of the GTSP to the Asymmetric TSP

- The Generalized TSP can be transformed into the Asymmetric TSP that can be then solved, e.g., by LKH or exactly using Concorde with further transformation of the problem to the TSP.



GTSP



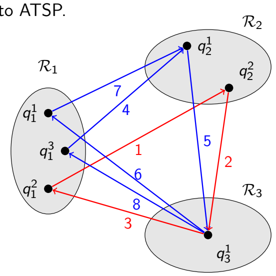
GATSP

- A transformation of the GTSP to the ATSP has been proposed by Noon and Bean in 1993, and it is called as the **Noon-Bean Transformation**.

Noon, C.E., Bean, J.C.: *An efficient transformation of the generalized traveling salesman problem*, INFOR: Information Systems and Operational Research, 31(1):39–44, 1993.
Ben-Arieg, D., Gutin, G., Penn, M., Yeo, A., Zverovitch, A.: *Transformations of generalized ATSP into ATSP*, Operations Research Letters, 31(5):357–365.

Noon-Bean Transformation

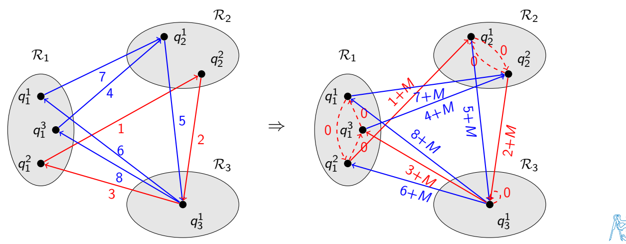
- Noon-Bean transformation to transfer GTSP to ATSP.
- Modify weight of the edges (arcs) such that the optimal ATSP tour visits all vertices of the same cluster before moving to the next cluster.
  - Adding a large a constant M to the weights of arcs connecting the clusters, e.g., a sum of the n heaviest edges.
  - Ensure visiting all vertices of the cluster in prescribed order, i.e., creating zero-length cycles within each cluster.
- The transformed ATSP can be further transformed to the TSP.
  - For each vertex of the ATSP created 3 vertices in the TSP, i.e., it increases the size of the problem three times.



Noon, C.E., Bean, J.C.: *An efficient transformation of the generalized traveling salesman problem*, INFOR: Information Systems and Operational Research, 31(1):39–44, 1993.

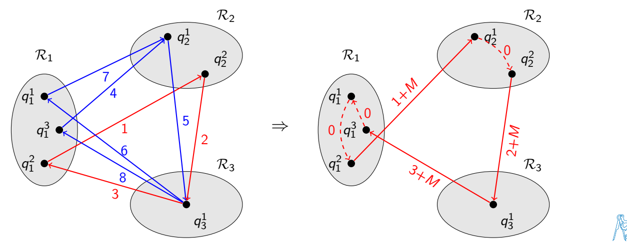
Example – Noon-Bean transformation (GATSP to ATSP)

1. Create a zero-length cycle in each set and set all other arcs to ∞ (or 2M).
   To ensure all vertices of the cluster are visited before leaving the cluster.
2. For each edge (q^m_i, q^n_j) create an edge (q^m_i, q^{n+1}_j) with a value increased by sufficiently large M.
   To ensure visit of all vertices in a cluster before the next cluster.



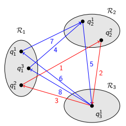
Example – Noon-Bean transformation (GATSP to ATSP)

1. Create a zero-length cycle in each set and set all other arcs to ∞ (or 2M).
   To ensure all vertices of the cluster are visited before leaving the cluster.
2. For each edge (q^m_i, q^n_j) create an edge (q^m_i, q^{n+1}_j) with a value increased by sufficiently large M.
   To ensure visit of all vertices in a cluster before the next cluster.



Noon-Bean transformation – Matrix Notation

- 1. Create a zero-length cycle in each set; and 2. for each edge (q^m_i, q^n_j) create an edge (q^m_i, q^{n+1}_j) with a value increased by sufficiently large M.



⇒

	q^1_1	q^2_1	q^3_1	q^1_2	q^2_2	q^1_3
q^1_1	∞	∞	∞	7	–	–
q^2_1	∞	∞	∞	–	1	–
q^3_1	∞	∞	∞	4	–	–
q^1_2	–	–	–	∞	∞	5
q^2_2	–	–	–	∞	∞	2
q^1_3	6	3	8	–	–	∞

∞ represents there are not edges inside the same set; and '–' denotes unused edge.

Original GATSP

	q^1_1	q^2_1	q^3_1	q^1_2	q^2_2	q^1_3
q^1_1	∞	∞	∞	7	–	–
q^2_1	∞	∞	∞	–	1	–
q^3_1	∞	∞	∞	4	–	–
q^1_2	–	–	–	∞	∞	5
q^2_2	–	–	–	∞	∞	2
q^1_3	6	3	8	–	–	∞

Transformed ATSP

	q^1_1	q^2_1	q^3_1	q^1_2	q^2_2	q^1_3
q^1_1	∞	0	∞	–	$7{+}M$	–
q^2_1	∞	∞	0	$1{+}M$	–	–
q^3_1	0	∞	∞	–	$4{+}M$	–
q^1_2	–	–	–	∞	0	$5{+}M$
q^2_2	–	–	–	0	∞	$2{+}M$
q^1_3	$8{+}M$	$6{+}M$	$3{+}M$	–	–	0

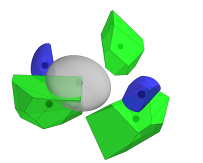
Noon-Bean Transformation – Summary

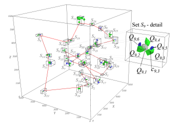
- It transforms the GATSP into the ATSP which can be further.
  - Solved by existing solvers, e.g., the Lin-Kernighan heuristic algorithm (LKH). http://www.akira.ruc.dk/~keld/research/LKH
  - The ATSP can be further transformed to the TSP and solve it optimaly, e.g., by the Concorde solver. http://www.tsp.gatech.edu/concorde.html
- It runs in $\mathcal{O}(k^2n^2)$ time and uses $\mathcal{O}(k^2n^2)$ memory, where n is the number of sets (regions) each with up to k samples.
- The transformed ATSP problem contains kn vertices.

Noon, C.E., Bean, J.C.: *An efficient transformation of the generalized traveling salesman problem*, INFOR: Information Systems and Operational Research, 31(1):39–44, 1993.

Generalized Traveling Salesman Problem with Neighborhoods (GTSPN)

- The **GTSPN** is a multi-goal path planning problem to determine a cost-efficient path to visit a set of 3D regions.
- A variant of the **TSPN**, where a particular neighborhood may consist of multiple (possibly disjoint) 3D regions.
- Redundant manipulators, inspection tasks with multiple views, multi-goal aircraft missions. Gentilini, I., et al. (2014)
- Regions are polyhedron, ellipsoid, and combination of both.
- We proposed decoupled approach Centroids-GTSP and GSOA-based methods with post-processing optimization.





Method	PDB [%]	PDM [%]	T_{CPU} [s]
HRGKA (Vicencio, et al, IROS 2014)	0.94	1.76	59.2
Centroids-GTSP	4.67	5.01	0.75
Centroids-GTSP$^+$	**0.06**	**0.47**	0.76
GSOA	0.74	3.43	**0.15**
GSOA-OPT	0.75	3.51	**0.31**

Faigl, J., Deckerová, J., and Váňa, P.: *Fast Heuristics for the 3D Multi-Goal Path Planning based on the Generalized Traveling Salesman Problem with Neighborhoods*, IEEE Robotics and Automation Letters, 4(3):2439-2446, 2019.

The Orienteering Problem (OP)

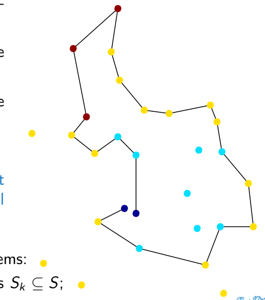
- The problem is to collect as many rewards as possible within the given **travel budget** (**T_{max}**), which is suitable for robotic vehicles with limited operational time.
- The starting and termination locations are prescribed and can be different.

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

Travel budget $T_{max} = 50$, Collected rewards $R = 190$ — Travel budget $T_{max} = 75$, Collected rewards $R = 270$

Orienteering Problem – Specification

- Let the given set of n sensors be located in $\mathbb{R}^2$ with the locations $S = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n\}$, $\boldsymbol{s}_i \in \mathbb{R}^2$.
- Each sensor $\boldsymbol{s}_i$ has an associated score ζ_i characterizing the reward if data from $\boldsymbol{s}_i$ are collected.
- The vehicle is operating in $\mathbb{R}^2$, and the travel cost is the Euclidean distance.
- Starting and final 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}.



The Orienteering Problem (**OP**) combines two NP-hard problems:

- Knapsack problem in determining the most valuable locations $S_k \subseteq S$;
- Travel Salesman Problem (**TSP**) in determining the shortest tour.

Orienteering Problem – Optimization Criterion

- Let $\Sigma = (\sigma_1, \ldots, \sigma_k)$ be a permutation of k sensor labels, $1 \le \sigma_i \le n$ and $\sigma_i \ne \sigma_j$ for $i \ne j$.
- Σ defines a **tour** $T = (\boldsymbol{s}_{\sigma_1}, \ldots, \boldsymbol{s}_{\sigma_k})$ 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

$$\begin{aligned} \text{maximize}_{k, S_k, \Sigma} \quad & R = \sum_{i=1}^{k} \zeta_{\sigma_i} \\ \text{subject to} \quad & \sum_{i=2}^{k} |(\boldsymbol{s}_{\sigma_{i-1}} - \boldsymbol{s}_{\sigma_i})| \le T_{max} \text{ and} \\ & \boldsymbol{s}_{\sigma_1} = \boldsymbol{s}_1, \boldsymbol{s}_{\sigma_k} = \boldsymbol{s}_n. \end{aligned} \quad (1)$$

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 $\boldsymbol{s}_1 = \boldsymbol{s}_n$ and particular S_k it becomes the TSP.

Existing Heuristic Approaches for the OP

- The Orienteering Problem has been addressed by several approaches, e.g.,

RB	4-phase heuristic algorithm proposed in [3];
PL	Results for the method proposed by Pillai in [2];
CGW	Heuristic algorithm proposed in [1];
GLS	Guided local search algorithm proposed in [4].

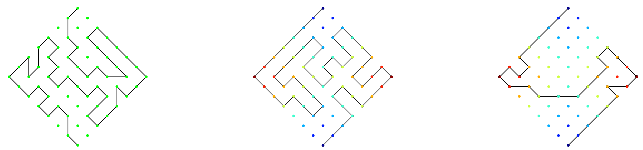
[1] I.-M. Chao, B. L. Golden, and E. A. Wasil.
A fast and effective heuristic for the orienteering problem.
European Journal of Operational Research, 88(3):475–489, 1996.

[2] R. S. Pillai.
The traveling salesman subset-tour problem with one additional constraint (TSSP+ 1).
Ph.D. thesis, The University of Tennessee, Knoxville, TN, 1992.

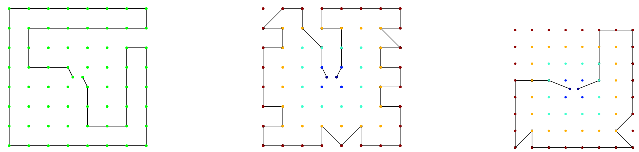
[3] R. Ramesh and K. M. Brown.
An efficient four-phase heuristic for the generalized orienteering problem.
Computers & Operations Research, 18(2):151–165, 1991.

[4] P. Vansteenwegen, W. Souffriau, G. V. Berghe, and D. V. Oudheusden.
A guided local search metaheuristic for the team orienteering problem.
European Journal of Operational Research, 196(1):118–127, 2009.

OP Benchmarks – Example of Solutions



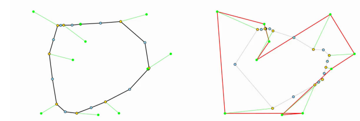
T_{max}=80, R=1248 — T_{max}=80, R =1278 — T_{max}=45, R=756

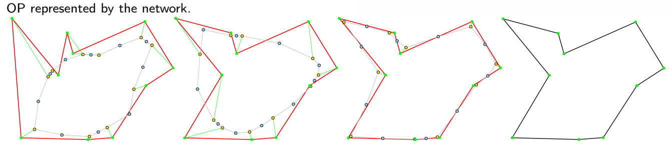


T_{max}=95, R=1395 — T_{max}=95, R=1335 — T_{max}=60, R=845

Unsupervised Learning for the OP 1/2

- A solution of the OP is similar to the solution of the PC-TSP and TSP.
- We need to satisfy the limited travel budget T_{max}, which needs the final tour over the sensing locations.
- During the unsupervised learning, the winners are associated with the particular sensing locations, which can be utilized to determine the tour as a solution of the OP represented by the network.





Learning epoch 7 — Learning epoch 55 — Learning epoch 87 — Final solution

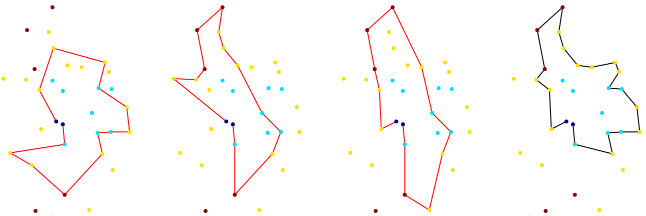
- This is utilized in the **conditional adaptation** of the network towards the sensing location and the adaptation is performed only if the tour represented by the network after the adaptation would satisfy T_{max}.

Unsupervised Learning for the OP 2/2

- The winner selection for $\boldsymbol{s}' \in S$ is conditioned according to T_{max}.
  - The network is adapted only if the tour T_{win} represented by the current winners would be shorter or equal than T_{max}.

$$\mathcal{L}(T_{win}) - |(\boldsymbol{s}_{\nu_p} - \boldsymbol{s}_{\nu_n})| + |(\boldsymbol{s}_{\nu_p} - \boldsymbol{s}')| + |(\boldsymbol{s}' - \boldsymbol{s}_{\nu_n})| \le T_{max}.$$

- The unsupervised learning performs a *stochastic search* steered by the rewards and the length of the tour to be below T_{max}.



Epoch 155, R=150 — Epoch 201, R=135 — Epoch 273, R=125 — Final solution, R=190

Comparison with Existing Algorithms for the OP

- Standard benchmark problems for the Orienteering Problem represent various scenarios with several values of T_{max}.
- The results (rewards) found by different OP approaches presented as the average ratios (and standard deviations) to the best-known solution.

Instances of the Tsiligirides problems

Problem Set	RB	PL	CGW	Unsupervised Learning
Set 1, $5 \le T_{max} \le 85$	0.99/0.01	1.00/0.01	1.00/0.01	1.00/0.01
Set 2, $15 \le T_{max} \le 45$	1.00/0.02	0.99/0.02	0.99/0.02	0.99/0.02
Set 3, $15 \le T_{max} \le 110$	1.00/0.00	1.00/0.00	1.00/0.00	1.00/0.00

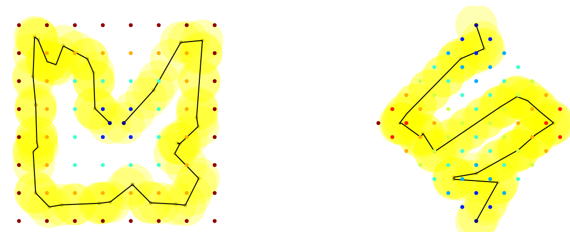
Diamond-shaped (Set 64) and Square-shaped (Set 66) test problems

Problem Set	RB†	PL	CGW	Unsupervised Learning
Set 64, $5 \le T_{max} \le 80$	0.97/0.02	1.00/0.01	0.99/0.01	0.97/0.03
Set 66, $15 \le T_{max} \le 130$	0.97/0.02	1.00/0.01	0.99/0.04	0.97/0.02

Required computational time is up to units of seconds, but for small problems tens or hundreds of milliseconds.

Orienteering Problem with Neighborhoods

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



T_{max}=60, δ=1.5, R=1600 — T_{max}=45, δ=1.5, R=1344

Orienteering Problem with Neighborhoods

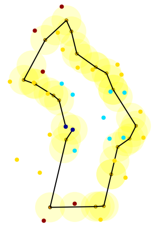
- Data collection using wireless data transfer allows to reliably retrieve data within some communication radius δ.
  - Disk-shaped δ-neighborhood – **Close Enough OP** (**CEOP**).
- We need to determine the most suitable locations P_k such that

$$\text{maximize}_{k,P_k,\Sigma} \quad R = \sum_{i=1}^{k} \zeta_{\sigma_i}$$

$$\text{subject to} \quad \sum_{i=2}^{k} |(\boldsymbol{p}_{\sigma_{i-1}} - \boldsymbol{p}_{\sigma_i})| \le \mathrm{T}_{\max},$$

$$|(\boldsymbol{p}_{\sigma_i}, \boldsymbol{s}_{\sigma_i})| \le \delta, \quad \boldsymbol{p}_{\sigma_i} \in \mathbb{R}^2,$$

$$\boldsymbol{p}_{\sigma_1} = \boldsymbol{s}_1, \boldsymbol{p}_{\sigma_k} = \boldsymbol{s}_n.$$



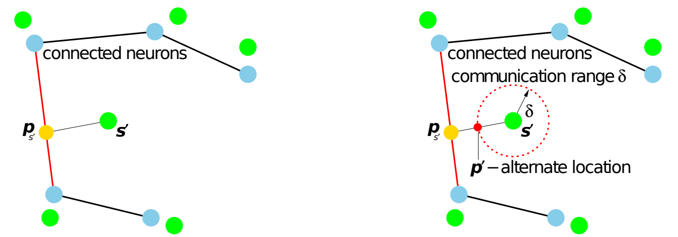
$\mathrm{T}_{\max} = 50$, $R = 270$

Introduced by Best, Faigl, Fitch (IROS 2016, SMC 2016, IJCNN 2017).

- More rewards can be collected than for the OP formulation with the same travel budget $\mathrm{T}_{\max}$.

Generalization of the Unsupervised Learning to the Orienteering Problem with Neighborhoods

- The same idea of the alternate location as in the TSPN.



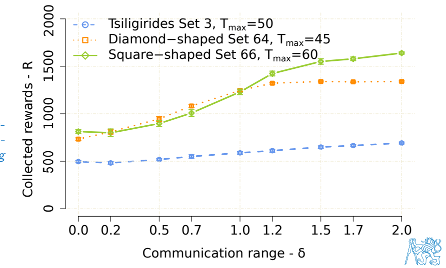
- The location $\boldsymbol{p}'$ for retrieving data from $\boldsymbol{s}'$ is determined as the alternate goal location during the conditioned winner selection.

Influence of the δ-Sensing Distance

- Influence of increasing communication range to the sum of the collected rewards.

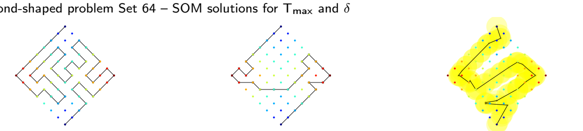
Problem	Solution of the OP R_{best}	R_{SOM}
Set 3, $\mathrm{T}_{\max}$=50	520	510
Set 64, $\mathrm{T}_{\max}$=45	860	750
Set 66, $\mathrm{T}_{\max}$=60	915	845

- *Allowing to data reading within the communication range δ may significantly increases the collected rewards, while keeping the budget under $T_{\max}$.*



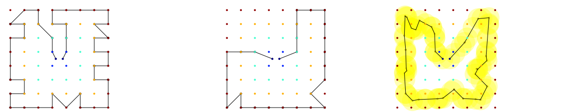
OP with Neighborhoods (OPN) – Example of Solutions

- Diamond-shaped problem Set 64 – SOM solutions for $\mathrm{T}_{\max}$ and δ



$\mathrm{T}_{\max}$=80, δ=0.0, R=1278 | $\mathrm{T}_{\max}$=45, δ=0.0, R=756 | $\mathrm{T}_{\max}$=45, δ=1.5, R=1344

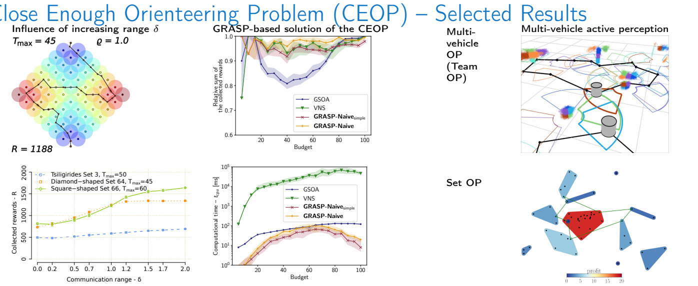
- Square-shaped problem Set 66 – SOM solutions for $\mathrm{T}_{\max}$ and δ



$\mathrm{T}_{\max}$=95, δ=0.0, R=1335 | $\mathrm{T}_{\max}$=60, δ=0.0, R=845 | $\mathrm{T}_{\max}$=60, δ=1.5, R=1600

In addition to unsupervised learning, **Variable Neighborhood Search** (**VNS**) for the OP has been generalized to the OPN.

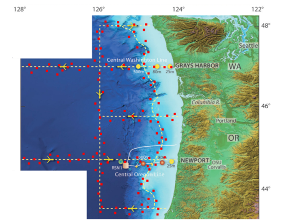
Close Enough Orienteering Problem (CEOP) – Selected Results



- Faigl, J.: *On self-organizing maps for orienteering problems*, International Joint Conference on Neural Networks (IJCNN), 2017, pp. 2611-2620.
- Štefaníková, P., Váňa, P., and Faigl, J.: *Greedy Randomized Adaptive Search Procedure for Close Enough Orienteering Problem*, 35th Annual ACM Symposium on Applied Computing, 2020, pp. 808-814.
- Best, G., Faigl, J., and Fitch, R.: *Online planning for multi-robot active perception with self-organising maps*, **Autonomous Robots**, 42(4):715-738, 2018.
- Pěnička, R., Faigl, J., and Saska, M.: *Variable Neighborhood Search for the Set Orienteering Problem and its application to other Orienteering Problem variants*, **European Journal of Operational Research**, 276(3):816-825, 2019.

Autonomous (Underwater) Data Collection

- Having a set of sensors (sampling stations), we aim to determine a cost-efficient path to retrieve data by autonomous underwater vehicles (AUVs) from the individual sensors. *E.g., Sampling stations on the ocean floor.*
- The planning problem is a variant of the **Traveling Salesman Problem.**



Two practical aspects of the data collection can be identified.

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

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

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

- Let n sensors be located in $\mathbb{R}^2$ at the locations $S = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n\}$.
- Each sensor has associated penalty $\xi(\boldsymbol{s}_i) \ge 0$ characterizing additional cost if the data are not retrieved from $\boldsymbol{s}_i$.
- Let the data collecting vehicle operates in $\mathbb{R}^2$ with the motion cost $c(\boldsymbol{p}_1, \boldsymbol{p}_2)$ for all pairs of points $\boldsymbol{p}_1, \boldsymbol{p}_2 \in \mathbb{R}^2$.
- The data from $\boldsymbol{s}_i$ can be retrieved within δ distance from $\boldsymbol{s}_i$.

PC-TSPN – Optimization Criterion

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

- **Determine a set of unique locations** $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k\}$, $k \le n$, $\boldsymbol{p}_i \in \mathbb{R}^2$, at which data readings are performed.
- **Find a cost efficient tour** T visiting P such that the total cost $C(T)$ of T is minimal

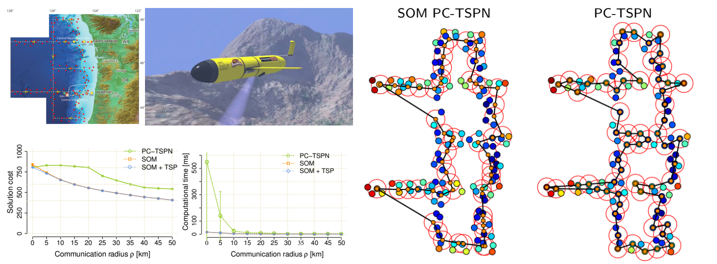
$$\mathcal{C}(T) = \sum_{(\boldsymbol{p}_{l_i}, \boldsymbol{p}_{l_{i+1}}) \in T} |(\boldsymbol{p}_{l_i} - \boldsymbol{p}_{l_{i+1}})| + \sum_{s \in S \setminus S_T} \xi(\boldsymbol{s}),$$

where $S_T \subseteq S$ are sensors such that for each $\boldsymbol{s}_i \in S_T$ there is $\boldsymbol{p}_{l_j}$ on $T = (\boldsymbol{p}_{l_1}, \ldots, \boldsymbol{p}_{l_{k-1}}, \boldsymbol{p}_{l_k})$ and $\boldsymbol{p}_{l_j} \in P$ for which $|(\boldsymbol{s}_i - \boldsymbol{p}_{l_j})| \le \delta$.

- PC-TSPN includes other variants of the TSP:
  - for $\delta = 0$ it is the PC-TSP;
  - for $\xi(\boldsymbol{s}_i) = 0$ and $\delta \ge 0$ it is the TSPN;
  - for $\xi(\boldsymbol{s}_i) = 0$ and $\delta = 0$ it is the ordinary TSP.

PC-TSPN – Example of Solution

Ocean Observatories Initiative (OOI) scenario



Faigl, J. and Hollinger, G.: *Autonomous Data Collection Using a Self-Organizing Map*, **IEEE Transactions on Neural Networks and Learning Systems**, 29(5):1703-1715, 2018.

Summary of the Lecture

Topics Discussed

- Data collection planning formulated as variants of
  - **Traveling Salesman Problem** (**TSP**)
  - **Orienteering Problem** (**OP**)
  - *Prize-Collecting Traveling Salesman Problem with Neighborhoods (PC-TSPN)*
- Exploiting the **non-zero sensing range** can be addressed as
  - **TSP with Neighborhoods** (**TSPN**) or specifically as the **Close Enough TSP** (**CETSP**) for disk-shaped neighborhoods.
  - **OP with Neighborhoods** (**OPN**) or the **Close Enough OP** (**CEOP**).
- Problems with continuous neighborhoods include continuous optimization that can be addressed by **sampling** the neighborhoods into discrete sets.
  - **Generalized TSP** and **Set OP**
- Existing solutions include
  - Approximation algorithms and heuristics (combinatorial, unsupervised learning, evolutionary methods)
  - Sampling-based and decoupled approaches
  - ILP formulations for discrete problem variants (sampling-based approaches)
  - Transformation based approaches (GTSP→ATSP) / Noon-Bean transformation
  - Combinatorial heuristics such as **VNS** and **GRASP**
  - TSP can be solved by efficient heuristics such as **LKH**
- Next: Curvature-constrained data collection planning