On Unsupervised Learning based Multi-Goal Path Planning for Visiting 3D Regions

Jan Faigl
Computational Robotics Laboratory
Faculty of Electrical Engineering (FEE)
Czech Technical University in Prague (CTU)
Technicka 2 Prague Czechia 16627
faiglj@fel.cvut.cz

Jindřiška Deckerová
Computational Robotics Laboratory
Faculty of Electrical Engineering (FEE)
Czech Technical University in Prague (CTU)
Technicka 2 Prague Czechia 16627
deckejin@fel.cvut.cz

ABSTRACT

In this paper, we report on our early results on deploying unsupervised learning technique for solving a multi-goal path planning problem to determine a shortest path to visit a given set of 3D regions. The addressed problem is motivated by data collection missions in which a robotic vehicle is requested to visit a set of locations to perform particular measurements. Instead of precise visitation of the specified locations, it is allowed to take the measurements at the respective distance from the locations, and thus save the travel cost by exploiting non-zero sensing radius of the vehicle. In particular, the problem is formulated as a 3D variant of the Close-Enough Traveling Salesman Problem (CETSP), and the proposed approach is based on the recently introduced technique called the Growing Self-Organizing Array (GSOA). The GSOA is a neural network for routing problems that is accompanied with unsupervised learning procedure to determine a solution of the TSP-like problems in a finite number of learning epochs. Based on the reported results, the proposed GSOA-based approach provides competitive or better results than existing combinatorial heuristics based on the so-called Steiner zones, while the computational requirements are significantly lower.

CCS Concepts

• Computing methodologies → Artificial intelligence → Planning and scheduling → Robotic planning • Computing methodologies → Machine learning → Learning paradigms → Unsupervised learning • Computing methodologies → Machine learning → Machine learning approaches → Neural networks

Keywords

CETSP, GSOA, Unsupervised Learning, Robotics, Multi-Goal Path Planning, Surveillance Planning, Inspection Planning

ICRAI 2018, November 17–19, 2018, Guangzhou, China

ACM ISBN 978-1-4503-6584-0/18/11…$15.00
https://doi.org/10.1145/3297097.3297099

1. INTRODUCTION

Having a set of locations to be visited by a robot, the problem to find a shortest path to visit all the locations is called multi-goal path planning problem (MTP) [1] and it can be formulated as one of the most popular combinatorial routing problems, the Traveling Salesman Problem (TSP). The TSP is a purely combinatorial problem to determine the optimal sequence of visits to the given locations, and it is known to be NP-hard, unless P=NP. The TSP is an important problem, and it can be considered as a well-studied problem of operational research with many existing approaches [16] including exact, approximate, and heuristic algorithms [2]. On the other hand, in complex situations [19], such as surveillance and data collection missions [10], it is desirable to optimize not only the sequence of visits but also the particular configurations (locations) of the visits. Hence, the problem becomes more challenging because it contains not only the NP-hard combinatorial optimization part but also continuous optimization in determining the most suitable points of visits to minimize the requested multi-goal path.

A particular motivation of the herein studied problem is a surveillance mission where a robotic vehicle is requested to take a snapshot of the object of interest using its downward looking camera, and it is sufficient to take a snapshot of the object within the field of view of the utilized camera, i.e., considering non-zero sensing radius. Then, it is sufficient to visit the location of the object of interest within the sensing distance, and thus eventually save the travel cost by avoiding precise visitation of the prescribed locations. Such a variant of the TSP is called the TSP with Neighborhoods (TSPN) in the literature [9] and this problem formulation has also been utilized for data collection planning [11], robotic environmental monitoring [5, 8], but also sequencing problems of robotic manipulators [1, 19]. In general, the addressed problem of multi-goal path planning for visiting given set of regions can be considered as the TSPN; however, following the notation introduced in [15], the problem is rather considered as the Close-Enough TSP (CETSP) to emphasize a continuous disk-shaped neighborhood with the radius δ.

The CETSP has been explicitly introduced by the authors of [15] to solve data collection planning problem to retrieve monthly data about customer utility measures using wireless communication. The authors of [15] proposed six heuristics based on determining a subset of possible locations of visits (called supernode set) and a solution of the regular TSP using the supernode set. Further, the CETSP has been extensively studied by Mennell in [17] where he proposed series of benchmarks for 2D and 3D problem instances. In [17], he reports on the evaluation of 15 heuristics based on the previous work on supernode set [7, 5, 15] including a sampling-based approach based on an explicit sampling of the neighborhoods and transformation of the CETSP to the Generalized TSP (GTSP). Although Mennell reports that the best solutions are found by the GTSP-based approach, he notes it is too computationally demanding, and the best trade-off between the solution quality and computational requirements are reported for his heuristic solution based on the Steiner zones (SZ), i.e., intersections of the respective neighborhoods, further denoted as SZ approaches.

In this paper, we report on early results on deploying recently proposed the Growing Self-Organizing Array (GSOA) [13] in a solution of the 3D variant of the CETSP. The GSOA can be considered as a variant of the unsupervised learning based neural network for a solution of various routing problems, and it originates in the self-organizing map for the TSP [12] generalized for polygonal regions in [14]. Beside the solution of the TSP, the same principles as for the solution of the TSP have also been deployed in a solution of the orienteering problem in [3] that has been further employed in active perception scenarios [4].

In the GSOA, a solution of the CETSP is represented as an array of nodes that evolves in the input space according to the unsupervised learning based adaptation procedure. The centers of the regions to be visited are iteratively considered for the adaptation of the GSOA, and a new node is created for each such a region as the closest point of the path formed by the connected nodes in the array. Then, the new node is adapted (moved) towards the determined location that is inside the respective δ neighborhood of the particular center of the region. The learning is performed in learning epochs, where a single learning epoch is a presentation of all regions to the GSOA. After each learning epoch, only the newly determined nodes are preserved, and all other nodes are discarded to balance the number of nodes with the number of the regions to be visited. Hence, the preserved nodes with their respective associated waypoint locations in the particular neighborhood (regions) represent a feasible solution of the CETSP because the order of visits is defined by order of the nodes in the array.

Although the GSOA introduced in [13] allows a straightforward extension for solving 3D variants of the CETSP directly, only results for the planar instances have been reported so far. Therefore, in this paper, we report on the early deployment of the GSOA for the CETSP [13] in a solution of the selected benchmarks of the 3D CETSP proposed in [17]. The achieved results are compared with the best performing heuristics based on the Steiner zones proposed in [17]. Based on the reported results, the proposed GSOA-based solution of the 3D CETSP provides solutions of the competitive quality, and in several cases, with better results than the best performing heuristics proposed in [17]. Moreover, the proposed GSOA approach is significantly less computationally demanding. It better scales with the problem size and for the largest and complex problems with many overlapping regions (neighborhoods) it is up to three orders of magnitude faster than the SZ-based approaches.

The rest of the paper is organized as follows. The formal introduction of the addressed problem is presented in the following section. A brief overview of the GSOA is presented in Section 3 because the same algorithm as for the 2D instances of the CETSP proposed in [13] is employed for the solution of the 3D CETSP, with only minor adjustments for 3D distances. The early achieved results in the selected benchmarks for the 3D CETSP with the comparison to the SZ based approaches proposed in [17] are reported in Section 4. Finally, concluding remarks are in Section 5.

2. PROBLEM FORMULATION

The addressed problem is multi-goal path planning, where the goal is to find a shortest path to visit a given set of 3D regions. The problem is formulated as a variant of the Close-Enough Traveling Salesman Problem (CETSP) where each region is defined by its center and radius δ, and thus the regions are spherical. The motivation is arising from data collection planning and following the notation utilized in the groundwork [13], we consider a given set of n locations $S = \{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n\}$, $\boldsymbol{s}_i \in \mathbb{R}^3$, each location possibly with an individual radius $\delta_i \in \mathbb{R}_0^+$. Notice if all $\delta_i = 0$, the problem becomes the regular TSP. The travel cost between any two points $\boldsymbol{p}_1, \boldsymbol{p}_2 \in \mathbb{R}^3$ is the Euclidean distance$\|(\boldsymbol{p}_1, \boldsymbol{p}_2)\|$.

In the CETSP, a particular region (location) $\boldsymbol{s}_i$ is considered visited if a point $\boldsymbol{p}_i$ of the final path is within δ_i distance from $\boldsymbol{s}_i$, i.e., $\|(\boldsymbol{s}_i, \boldsymbol{p}_i)\| \leq \delta_i$ .Hence, the problem is to determine a sequence of visits to the locations that can be expressed as a permutation $\Sigma = (\sigma_1, \cdots, \sigma_n)$, where $1 \leq \sigma_i \leq n$ and $\sigma_i \neq \sigma_j$ for $i \neq j$, and the respective waypoints $P = \{\boldsymbol{p}_1, \cdots, \boldsymbol{p}_n\}, \boldsymbol{p}_i \in \mathbb{R}^3$, in the δ-neighborhood of the particular location, and thus $\|(\boldsymbol{s}_i, \boldsymbol{p}_i)\| \leq \delta_i$. Having these preliminaries, the CETSP can be formulated as a discrete combinatorial optimization in finding the sequence Σ and continuous optimization in finding the waypoint locations P as follows

Problem 1 (Close Enough Traveling Salesman Problem)

$$minimize_{\Sigma,P}\ L(\Sigma, P, S)$$

$$L(\Sigma, P, S) = \sum_{i=1}^{n-1} \|(\boldsymbol{p}_{\sigma_i}, \boldsymbol{p}_{\sigma_{i+1}})\| + \|(\boldsymbol{p}_{\sigma_n}, \boldsymbol{p}_{\sigma_1})\|$$

subject to

$$S = \{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n\};\ \boldsymbol{s}_i \in \mathbb{R}^3$$

$$\Sigma = (\sigma_1, \cdots, \sigma_n);\ 1 \leq \sigma_i \leq n$$

$$P = \{\boldsymbol{p}_1, \cdots, \boldsymbol{p}_n\}; \boldsymbol{p}_i \in \mathbb{R}^3$$

$$\boldsymbol{s}_{\sigma_1} = \boldsymbol{s}_1 \text{and} \|(\boldsymbol{s}_i, \boldsymbol{p}_i)\| \leq \delta_i, \text{for } 1 \leq i \leq n$$

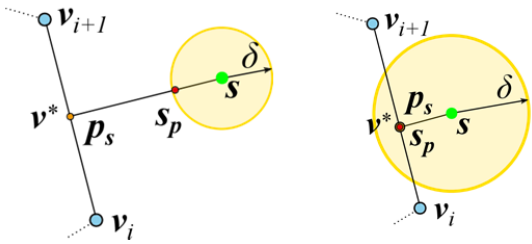
For simplicity and without loss of generality, we assume the first location $\boldsymbol{s}_1$ is the initial location of the vehicle, and therefore, we consider $\boldsymbol{s}_{\sigma_1} = \boldsymbol{s}_1$. Besides, we can further assume $\delta_1 = 0$if the initial (and final) location of the vehicle is strictly requested to be at the particular location.

3. GSOA FOR 3D CETSP

The Growing Self-Organizing Array (GSOA) has been introduced in [13] together with its evaluation in 2D benchmarks of the CETSP. The same algorithm has been directly employed in the solution of the 3D instances of the CETSP with the only extension of the dimension of the data representation for 3D and the related computation of the distances and determination of the locations within a spherical neighborhood around each location, which is analogical for the disk-shaped neighborhood in 2D. Therefore, only a brief overview of the GSOA is presented here to make the paper self-contained.

The GSOA is an array of nodes $\mathcal{N} = \{\nu_1, \cdots, \nu_M\}$, which represents a solution of the TSP that is accompanied by an unsupervised learning procedure. Each node corresponds to a location in the problem space $\boldsymbol{v}_i \in \mathbb{R}^3$, i.e., $\boldsymbol{v}_i = (x_{v_i}, y_{v_i}, z_{v_i})$, and it is further associated with the particular location to be visited $\boldsymbol{s} \in S$ and also waypoint location $\boldsymbol{p} \in \mathbb{R}^3$ inside the δ-neighborhood of $\boldsymbol{s}$. Thus, a solution of the CETSP as a sequence of waypoints can be constructed by traversing the array.

The learning procedure starts with a single node $\mathcal{N} = \{v_1\}$ that is initialized, e.g., as the first location $\boldsymbol{s}_1$. Then, new nodes are iteratively determined for all $\boldsymbol{s} \in S$, and each new node is added to the array and adapted towards the respective waypoint location.



$\boldsymbol{p}_s$ outside the δ-neighborhood of $\boldsymbol{s}$ $\quad$ $\boldsymbol{p}_s$ inside the δ-neighborhood of $\boldsymbol{s}$

Figure 1. Principle of determining new node $\boldsymbol{v}^*$for $\boldsymbol{s}$ with the corresponding waypoint location $\boldsymbol{s}_p$. Adapted from [13].

During the learning, the GSOA is adapted towards every sensor location $\boldsymbol{s} \in S$, and adaptation to all locations is called learning epoch. The locations are considered in a random order to avoid local optima and for each $\boldsymbol{s}$ a new node $\boldsymbol{v}^*$is determined as the closest point $\boldsymbol{p}_s$ of the array to $\boldsymbol{s}$. In particular, for an array of M nodes, two nodes $\boldsymbol{v}_i$ and $\boldsymbol{v}_{i+1}$ form a straight line segment and we can iterate through the array and determine the point $\boldsymbol{p}_s$ as the closest point of the i-th segment $(\boldsymbol{v}_i,\ \boldsymbol{v}_{i+1})$ to $\boldsymbol{s}$ that has the minimal distance to $\boldsymbol{s}$, e.g., as in (1).

$$\boldsymbol{p}_s = \operatorname{argmin}_{1 \le i \le M} \|(\boldsymbol{p}_i', \boldsymbol{s})\|$$
$$\text{subject to} \qquad (1)$$
$$\boldsymbol{p}_i' \in (\boldsymbol{v}_i, \boldsymbol{v}_{i+1}) \text{and} \boldsymbol{v}_{M+1} = \boldsymbol{v}_1$$

Then, the corresponding waypoint is determined as a point on the straight line segment connecting the location and the closest point, $(\boldsymbol{p}_s, \boldsymbol{s})$. We can distinguish two cases, see Fig. 1. For $\boldsymbol{p}_s$ outside the δ-neighborhood of$\boldsymbol{s}$, the waypoint is determined as a point on the straight line segment $(\boldsymbol{p}_s, \boldsymbol{s})$ with the distance from $\boldsymbol{s}$ equal to the respective radius δ (the real distance is $\delta - \epsilon$, where ϵ is a small constant), i.e.,

$$\boldsymbol{s}_p = \boldsymbol{s} + (\boldsymbol{p}_s - \boldsymbol{s}) \frac{\delta - \epsilon}{\|(\boldsymbol{p}_s, \boldsymbol{s})\|} \qquad (2)$$

For $\boldsymbol{p}_s$ inside the δ-neighborhood, the point $\boldsymbol{p}_s$ is directly the waypoint location itself.

After determination of$\boldsymbol{p}_s$, a new node $\boldsymbol{v}^* = \boldsymbol{p}_s$is added to the array and it is together with its neighboring nodes adapted towards the waypoint location $\boldsymbol{s}_p$ with decreasing power of adaptation for farther neighbors using the neighboring function

$$f(\sigma, d) = \begin{cases} e^{\frac{-d^2}{\sigma^2}} & \text{for } d < 0.2M, \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

where M is the number of nodes in the array, σ is the learning gain, d is the distance of the adapted node $\boldsymbol{v}$ to $\boldsymbol{v}^*$ counted in the number of nodes in the array. The adaptation is adjustment of the node locations according to

$$\boldsymbol{v}' = \boldsymbol{v} + \mu f(\sigma, d)(v^*.\boldsymbol{s}_p - \boldsymbol{v}^*), \qquad (4)$$

where μ is the learning rate, $f(\sigma, d)$ is the neighboring function (3) and $v^*.\boldsymbol{s}_p$ is the determined waypoint of the new node $\boldsymbol{v}^*$. At the end of the learning epoch, each location $\boldsymbol{s} \in S$ is assigned to newly determined node added to the array in the current epoch, and therefore, all other nodes are removed from the array. Besides, the learning parameters are updated to ensure convergence of the learning, and the solution is obtained by traversing the array and connecting the corresponding waypoints associated with the nodes. Then, the best solution found so far is maintained and the learning is repeated for the next learning epoch until the GSOA converges to a stable solution, which is usually in less than one hundred epochs. Finally, the solution is improved using Two-opt heuristic [6].

The complexity of the single learning epoch can be bounded by $O(n^2)$for the n number of locations S. The initial values of the learning parameters are $\alpha = 0.0005$, $\mu = 0.6$, and σ =10. The learning gain σ is updated after each learning epoch as $\sigma \leftarrow (1 - i\alpha)\sigma$, where i is the epoch counter. More details together with the computational complexity and convergence analyses can be found in [13].

4. RESULTS

The performance of the GSOA in solving 3D instances of the CETSP has been evaluated in a series of instances proposed by Mennell in [17]. In particular, we consider benchmarks with arbitrary δ-neighborhood radius per each location. The benchmarks include instances with hundreds of locations up to one thousand locations, and thus the reported results provide an overview how the algorithms scale with the size of the problem and its complexity (regarding overlapping neighborhoods) by means of the solution quality and the required computational time. The deployed GSOA [13] is compared with the best performing heuristics based on the computation of the Steiner zones proposed in [17]. The selected heuristics are the SZ2 and SZ3, where the SZ3 is more demanding than the SZ2, but it is reported to provide high-quality solutions [17].

The performance indicators are the solution quality and required computational time. The quality is defined by the length of the found multi-goal path that is measured as the percentage deviation from the reference solution of the best solution value among the found solutions and it is denoted %PDB

$$\%\text{PDB} = \frac{L - L_{ref}}{L_{ref}} \cdot 100\%, \qquad (5)$$

where L_{ref} is the best known solution of the particular problem instance and L is the length of the best solution found among the performed trials. The initial value of the reference solutions are the best solutions reported in [17]; however, the deployed GSOA provides better results in several cases, and therefore, we denote the previous reference solutions of [17] as L_{ref}' and L_{ref} denotes the new best solutions among all solutions found by the SZ2, SZ3, and GSOA-based solvers in the rest of this paper.

In addition to %PDB, the percentage deviation from the reference of the mean solution value over the performed trials (denoted %PDM) is computed to measure the robustness of the solver similarly as in [13] because the GSOA is a randomized algorithm. The %PDM value is computed as

$$\%\mathrm{PDM} = \frac{L_{avg} - L_{ref}}{L_{ref}} \cdot 100\%, \qquad (6)$$

where L_{avg} is the average solution length among the performed trials.

Table 1. Computational Results for 3D instances of the CETSP with arbitrary radius per each location

Problem Instance	n	L_{ref}	L'_{ref}	SZ2 %P	SZ2 T_{CPU}	SZ3 %P	SZ3 T_{CPU}	**GSOA** %P	**GSOA** %P	**GSOA** T_{CPU}
team1_100*rdmRad*	101	907.59	907.59	4.61	8.457	**0.15**	96.641	1.67	6.54	0.024
team2_200*rdmRad*	201	**1051.26**	1055.95	3.31	52.461	0.45	789.356	**0.00**	1.41	0.091
team3_300*rdmRad*	301	**1028.29**	1053.38	6.16	28.056	2.44	9471.162	**0.00**	3.00	0.166
team4_400*rdmRad*	401	**1275.69**	1276.90	2.19	484.922	0.37	2855.303	**0.00**	2.50	0.364
team5_499*rdmRad*	500	**819.27**	840.48	9.86	50.723	2.59	66682.734	**0.00**	3.61	0.419
team6_500*rdmRad*	501	**1016.52**	1076.35	11.03	78.096	5.89	29265.781	**0.00**	2.00	0.469
kroD100*rdmRad*	1001	**2185.44**	2689.41	23.06	84.043	23.06	56.396	**0.00**	1.75	2.117
rat195*rdmRad*	100	171.57	171.57	1.82	5.723	**0.00**	75.410	0.96	2.00	0.023
lin318*rdmRad*	195	**83.51**	84.47	3.72	8.936	1.14	1215.791	**0.00**	2.92	0.042
rd400*rdmRad*	318	2189.43	2189.43	7.18	52.451	**0.00**	21649.931	0.68	5.29	0.176
pcb442*rdmRad*	400	3595.97	3595.97	0.08	130.625	**0.00**	1040.186	5.40	6.52	0.377
d493*rdmRad*	442	**251.66**	258.40	7.28	36.846	2.68	4628.740	**0.00**	1.29	0.346
dsj1000*rdmRad*	493	**736.26**	761.07	9.44	218.096	3.37	15632.871	**0.00**	2.65	0.452
bonus1000*rdmRad*	1000	**1652.30**	2074.84	25.57	39.864	25.57	29.287	**0.00**	2.26	1.813

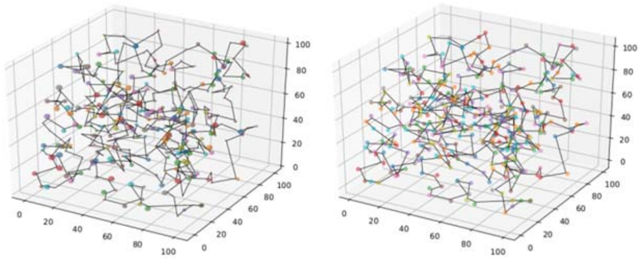
Only a single solution is reported for the SZ2 and SZ3 in [17], and therefore, the only %PDB is reported for these heuristic methods. On the other hand, the GSOA-based solution is found 20 times for each particular problem instance, and thus the reported results include also the overview of the algorithm robustness measured as %PDM.

The computational requirements are measured as the real required computational time denoted as T_{CPU}report in seconds. The GSOA is implemented in C++, and all the reported results have been found using a single core of the Intel i5-5200U processor running at 2.2 GHz that is about 1.57 times faster than the reported results for the SZ2 and SZ3 [17] obtained using the Intel Pentium E2220 processor according to single thread rating [18]. Therefore, the reported required computational times in [17] are divided by 1.6 to make the herein presented computational times comparable with the computational times of the GSOA achieved by the different computational environment.

It is worth mentioning that computation of the Steiner zones in the 3D is much more computationally demanding than for the 2D. It is not the case of the unsupervised learning in the GSOA, where only 3D Euclidean distances are computed together with intersections of the straight line segments with a sphere (instead of circle); hence, it can be expected the computational requirements are only slightly changed in comparison to the solution of the 2D instances of the CETSP reported in [13].

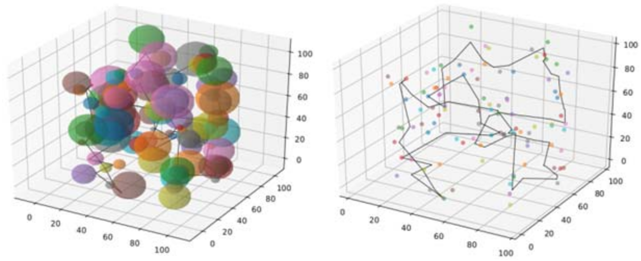
The results on 3D instances of the CETSP with arbitrary radii per each location [17] are listed in Table 1. Selected best solutions found by the GSOA are visualized in Fig. 2–Fig. 4 with and without visualization of the respective spherical neighborhoods to improve clarity of the complex solutions in 3D scenarios.

The employed GSOA provides new best solutions for most of the evaluated instances of the 3D CETSP, which is indicated by the highlighted values of L_{ref} and also by the respective zero values of %PDB in Table 1. Moreover, computational times are below one second except for two cases with one thousand locations. Therefore the GSOA is significantly less computationally demanding than the heuristics based on the Steiner zones [17]. Regarding the results for 2D instances of the CETSP reported in [13], the extension of the GSOA from 2D to 3D instances seems to be computationally negligible which is not the case of the SZ approaches. The online sampling of the possible waypoint locations during the unsupervised learning is the main benefit of the GSOA-based approach in comparison to the explicit computation of the Steiner zones, which is computationally very demanding for the 3D instances.



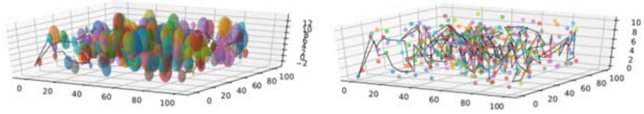
With neighborhoods Without neighborhoods

Figure 2. Best found solution of the *rd400rdmRad* problem with $L = 3759.76$.



With neighborhoods Without neighborhoods

Figure 3. Best found solution of the *team1_100rdmRad* problem with $L = 922.71$.



With neighborhoods Without neighborhoods

Figure 4. Best found solution of the *team4_400rdmRad* problem with $L = 1275.69$.

5. CONCLUSION

In this paper, we report on the deployment of the unsupervised learning based approach called the Growing Self-Organizing Array (GSOA) for routing problems in a solution of the 3D instances of the CETSP. The GSOA for 2D instances is directly deployed with a straightforward extension to 3D that consists only of computing distances in R3 and intersections of straight line segments with a sphere. Based on the reported results, the GSOA seems to be vital also for a solution of the 3D scenarios, where it provides competitive solutions with significantly lower computational requirements. In our future work, we plan to thoroughly evaluate the performance of the GSOA in all available 3D benchmarks proposed in [17] but also in other scenarios motivated by multi-goal path planning for visiting 3D regions, e.g., [19]. Besides, we plan to investigate the cases where the SZ-based approaches provide better results than the GSOA with the aim to eventually combine the benefits of both the unsupervised learning based and combinatorial heuristic approaches in a single solver.

6. ACKNOWLEDGMENTS

This work has been supported by the Czech Science Foundation (GAČR) under research project No. 16-24206S. The authors acknowledge the support of the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics".

7. REFERENCES

[1] Alatartsev, S., Stellmacher, S., and Ortmeier, F. 2015. Robotic Task Sequencing Problem: A Survey. *Journal of Intelligent & Robotic Systems*, 80, (Nov. 2015), 279–298. DOI=https://doi.org/10.1007/s10846-015-0190.

[2] Applegate, D., Bixby, R., Chvátal, V., and Cook, W. 2007. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, Princeton, NJ, USA.

[3] Best, G., Faigl, J., and Fitch, R.2016. Multi-robot path planning for budgeted active perception with self-organising maps*. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. DOI= https://doi.org/10.1109/IROS.2016.7759489.

[4] Best, G., Faigl, J., and Fitch, R. 2018. Online planning for multi-robot active perception with self-organising maps. *Autonomous Robot.* 42, 4 (Apr. 2018), :715–736. DOI= https://doi.org/10.1007/s10514-017-9691-4.

[5] Yuan, B., Orlowska, W., and Sadiq, S. On the Optimal Robot Routing Problem in Wireless Sensor Networks. *IEEE Transactions on Knowledge and Data Engineering*, 19, 9 (2007), 1252–1261. DOI= https://doi.org/10.1109/TKDE.2007.1062.

[6] Croes, G. A. 1958. A method for solving traveling-salesman problems. *Operations Research*, 6, 6 (1958),791–812.

[7] Dong, J., Yang, N., and Chen, M.. 2007. *Heuristic Approaches for a TSP Variant: The Automatic Meter Reading Shortest Tour Problem*, Springer US, Boston, MA. DOI= https://doi.org/10.1007/978-0-387-48793-9_10.

[8] Dunbabin, M. and Marques, L. 2012. Robots for Environmental Monitoring: Significant Advancements and Applications. *IEEE Robotics & Automation Magazine*, 19, 1 (Mar. 2012), 24–39, DOI= https://doi.org/10.1109/MRA.2011.2181683.

[9] Elbassioni, K., Fishkin, A. V., and Sitters, R. 2009. Approximation algorithms for the Euclidean traveling salesman problem with discrete and continuous neighborhoods. *International Journal of Computational Geometry & Applications*, 19,02 (2009), 173–193. DOI= https://doi.org/10.1142/S0218195909002897.

[10] Faigl, J. and Hollinger, G. 2014. Unifying multi-goal path planning for autonomous data collection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. DOI= https://doi.org/10.1109/IROS.2014.6942967.

[11] Faigl, J. and Hollinger , G. 2018. Autonomous data collection using a self-organizing map. *IEEE Transactions on Neural Networks and Learning Systems*, 29,5 (May 2018),1703–1715,. DOI= https://doi.org/10.1109/TNNLS.2017.2678482.

[12] Faigl, J. 2011. On the performance of self-organizing maps for the non-euclidean traveling salesman problem in the polygonal domain. *Information Sciences*, 181(October 2011), 4214–4229, DOI= https://doi.org/10.1016/j.ins.2011.05.019.

[13] Faigl, J. 2018. GSOA: Growing Self-Organizing Array – unsupervised learning for the close-enough traveling salesman problem and other routing problems.

Neurocomputing, 312, 27 (Oct. 2018), 120–134. DOI= https://doi.org/10.1016/j.neucom.2018.05.079.

[14] Faigl, J. and Přeučil, L. 2011. Self-Organizing Map for the Multi-Goal Path Planning with Polygonal Goals. In *International Conference on Artificial Neural Networks (ICANN)*Heidelberg,Springer. DOI= https://doi.org/10.1007/978-3-642-21735-7_11.

[15] Gulczynski D. J., Heath, J. W., and Price, C. C. 2006. *The Close Enough Traveling Salesman Problem: A Discussion of Several Heuristics*, Springer US, Boston, MA. DOI= https://doi.org/10.1007/978-0-387-39934-8_16.

[16] Gutin, G. and Punnen, A. P. editors. 2007.*The Traveling Salesman Problem and Its Variations*. Springer US, DOI= https://doi.org/10.1007/b101971.

[17] Mennell, W. K. 2009. *Heuristics for Solving Three Routing Problems: Close-Enough Traveling Salesman Problem, Close-Enough Vehicle Routing Problem, Sequence-Dependent Team Orienteering Problem*., Doctoral Thesis. University of Maryland.

[18] PassMark® Software Pty Ltd. Cpu performance comparison, 2018. [cited 10 Jul 2018].https://www.cpubenchmark.net/compare.php?cmp[]=1137&cmp[]=2440.

[19] Vicencio, K. Davis, B., and Gentilini, I., 2014.Multi-goal path planning based on the generalized traveling salesman problem with neighborhoods. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.DOI= https://doi.org/10.1109/IROS.2014.6942974.