# Self-Organizing Map for the Curvature-Constrained Traveling Salesman Problem

Jan Faigl and Petr Váňa

Czech Technical University in Prague - Department of Computer Science,
Technická 2, 166 27 Prague 6, Czech Republic
{faiglj,vanapet1}@fel.cvut.cz

**Abstract.** In this paper, we consider a challenging variant of the traveling salesman problem (TSP) where it is requested to determine the shortest closed curvature-constrained path to visit a set of given locations. The problem is called the Dubins traveling salesman problem in literature and its main difficulty arises from the fact that it is necessary to determine the sequence of visits to the locations together with particular headings of the vehicle at the locations. We propose to apply principles of unsupervised learning of the self-organizing map to simultaneously determine the sequence of the visits together with the headings. A feasibility of the proposed approach is supported by an extensive evaluation and comparison to existing solutions. The presented results indicate that the proposed approach provides competitive solutions to existing heuristics, especially in dense problems, where the optimal sequence of the visits cannot be determined as a solution of the Euclidean TSP.

## 1 Introduction

A problem of finding a shortest closed path to visit a given set of locations can be formalized as the traveling salesman problem (TSP) for which several approaches have been proposed [4]. The basic variant of the TSP is the Euclidean TSP where locations are placed in a plane and each pair of the locations can be connected by a straight line segment with the length computed as the Euclidean distance between the locations. Although this problem formulation addresses many practical problems [4], it does not fit surveillance missions with curvature-constrained vehicles such as aircraft, for which the shortest path connecting two locations depends on the particular headings of the vehicle at the locations.

Optimal path planning for a vehicle with a constant forward velocity and limited turning radius $\rho$ has been studied by Dubins who showed that the optimal path connecting two locations with prescribed headings is one of the six possible maneuvers [5]. The optimal maneuver can be determined analytically and it is called Dubins maneuver where the motion model is called the Dubins vehicle. However, the analytic solution does not allow to directly solve the so-called Dubins traveling salesman problem (DTSP), which stands to find a shortest closed path for the Dubins vehicle to visit a given set of locations [11]. It is because each heading can be selected from the interval $\langle 0, 2\pi)$ and the total length of the shortest path visiting the locations depends on the headings and also on the order of their visits. Therefore, it is necessary to determine both the headings and sequence of visits to the locations in the DTSP.

Three fundamental approaches for the DTSP can be found in literature. The first are methods based on a solution of the Euclidean TSP (ETSP) with the relaxed curvature constraint that include approximate algorithms with a relatively high approximation ratios [9] and heuristic algorithms such as the Alternating algorithm (AA) [12] or Local iterative optimization (LIO) [14]. Heuristics provide relatively good results in instances with locations far from each other, for which the solution of the ETSP provides optimal or close to optimal sequence in the DTSP. Moreover, for locations with mutual distance longer than $4\rho$, the optimal headings for a given sequence can be found by convex optimization [7]. Therefore, it seems that instances with dense locations are more challenging, since it is necessary to simultaneously determine the optimal sequence and headings.

Two additional types of approaches are sampling-based methods [10] and evolutionary techniques such as genetic [15] and memetic [16] algorithms that consider particular values of possible headings at each location and solve the sequencing part of the problem. Sampling-based methods need a prescribed discretization of the headings and address the DTSP as the Generalized Asymmetric TSP which is transformed into the Asymmetric TSP [10] that can be solved optimally by the Concorde solver [2]. Although sampling-based approaches are able to provide high quality solutions, they become quickly computationally intractable for increasing number of locations and samples. On the other hand, evolutionary methods provide the first feasible solutions relatively quickly, which is then further improved if more computational time is available.

In this paper, we consider principles of existing self-organizing map (SOM) approaches for the TSP [1, 13, 6] to address challenges of the DTSP. The main difficulty of applying SOM to the DTSP is in computation of the best matching unit, which needs to respect the locations and headings regarding the previous and next waypoints in the tour. The proposed SOM for the DTSP encodes expected headings at the locations into the network structure and heading values are refined during the unsupervised learning. Although the proposed approach does not provide optimal solution of the DTSP, which has been also observed in SOM for the ETSP [3], it provides better results than simple existing heuristics [12, 14] in problems where the optimal sequence of the visits is not the same as the optimal solution of the underlying ETSP. Moreover, the proposed SOM provides competitive results to the existing Memetic algorithm [16] with the computational time limited to 1 hour while SOM is significantly faster.

## 2 Problem Statement

The motivation of the addressed curvature-constrained traveling salesman problem is a solution of the surveillance missions with a fixed-wing aerial vehicle that is modeled as the Dubins vehicle with the minimum turning radius $\rho$ and constant forward velocity $v$. The state of the vehicle $q$ is a triplet $q = (x, y, \theta)$ from the special Euclidean group $q \in SE(2)$, where $(x, y)$ is the vehicle position in a plane and $\theta \in \mathbb{S}^1$ is the vehicle heading at $(x, y)$. The model can be formally described as:

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

where $u$ is the control input. For simplicity and without loss of generality, we consider $v$=1 and $\rho$=1 in the rest of the paper.

In surveillance missions, the Dubins vehicle is requested to visit a set of $n$ locations $P = \{p_1, \ldots, p_n\}$, $p_i \in \mathbb{R}^2$ by a closed path. Therefore, the problem stands to determine a sequence of visits to the locations together with the vehicle's heading at each location $p_i \in P$ [8]. The problem can be formally described as follows. Let $\Sigma = (\sigma_1, \ldots, \sigma_n)$ be an ordered permutation of $\{1, \ldots, n\}$ and $\mathcal{P}$ be a projection from $SE(2)$ to $\mathbb{R}^2$ such that $\mathcal{P}(q_i) = (x_i, y_i)$, where $q_i$ is an element of $SE(2)$ whose projection is the location $p_i = (x_i, y_i)$. The problem is to determine the minimum length tour that visits every location $p_i \in P$ while satisfying the constraints of the Dubins vehicle (1). This is an optimization problem over all possible permutations $\Sigma$ and headings $\Theta = \{\theta_{\sigma_1}, \theta_{\sigma_2}, \ldots, \theta_{\sigma_n}\}$ in the states $(q_{\sigma_1}, q_{\sigma_2}, \ldots, q_{\sigma_n})$ such that $q_{\sigma_i} = (p_{\sigma_i}, \theta_{\sigma_i})$:

$$\mathit{minimize}_{\,\Sigma,\Theta} \sum_{i=1}^{n-1} \mathcal{L}(q_{\sigma_i}, q_{\sigma_{i+1}}) + \mathcal{L}(q_{\sigma_n}, q_{\sigma_1}) \tag{2}$$

$$\mathit{subject\ to}\ \ q_i = (p_i, \theta_i)\ i = 1, \ldots, n, \tag{3}$$

where $\mathcal{L}(q_{\sigma_i}, q_{\sigma_j})$ is the length of the shortest possible path (Dubins maneuver) for the Dubins vehicle (1) between the states $q_{\sigma_i}$ and $q_{\sigma_j}$.

## 3 Proposed Self-Organizing Map for the DTSP

The proposed unsupervised learning procedure builds on existing self-organizing maps for the Euclidean TSP [13, 6]. SOM for the TSP is two-layer neural network which maps the input space $\mathbb{R}^2$ into an array of output units. The input of the network are the locations to be visited $P = \{p_1, \ldots, p_n\}$, $p_i \in \mathbb{R}^2$, while neurons $\mathcal{N}$ represent particular states of the Dubins vehicle in $SE(2)$, $\mathcal{N} = \{\nu_1, \ldots, \nu_m\}$, where $\nu_i \in SE(2)$ and we use $m$=$2n$ according to [13].

Similarly to SOM for the ETSP, connected neurons form a ring representing a closed path in the input space. Since the sequence is prescribed by the output layer and each neuron has associated heading, it is straightforward to determine the optimal curvature-constrained path for the Dubins vehicle (1) using analytic solution of the optimal Dubins maneuvers [5].

In contrast to the solution of the ETSP, we need to adapt not only neuron weights to the locations $P$ but we also need an adaptation rule to adjust the headings at the locations. It is known that the distance function $\mathcal{L}$ of the Dubins maneuvers is sensitive to headings, especially for two close locations. Therefore, in addition to the main heading $\theta_i$ associated to each neuron $\nu_i$, we consider up to $2k$ headings around $\theta_i$ according to the neighbouring function $f(\sigma, d)$, where $d$ is the distance in the number of nodes and $\sigma$ is the learning gain. These headings may be considered as additional neurons; however, they are utilized only in the evaluation of the winner and in the local improvement of the solution of the DTSP represented by the ring. Based on the empirical evaluation, $k$=12 provides a suitable tradeoff between the solution quality and computational requirements.

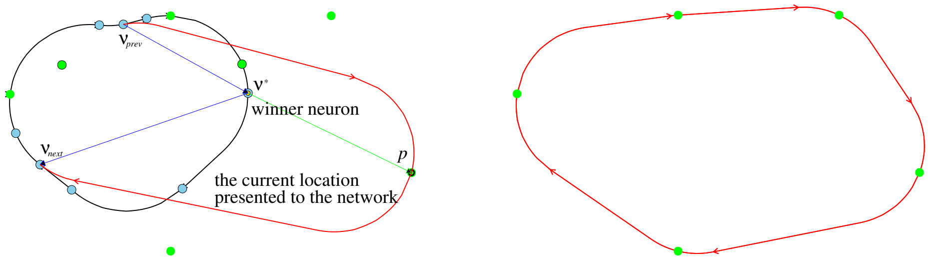



**Fig. 1.** Example of winner selection (left) and the final found solution (right). The locations to be visited are represented by green disks, the neurons are in blue and they are connected into a ring by Dubins maneuvers (black curve). The green straight line segment connects the current winner with its location $p \in P$ while the selected previous $\nu_{prev}$ and next $\nu_{next}$ neurons of the winner are highlighted by the blue segments. The red curve (left) is the Dubins path used for the selection of the winner neuron.

The key idea of the proposed SOM for the DTSP is the winner selection that considers headings and also the length of the Dubins path. The winner $\nu_i^*$ for $p \in P$ is selected as the best matching unit according to the distance computed as the length of the two Dubins maneuvers connecting $\nu_{prev}$ with the state $(p, \theta_i)$ and $(p, \theta_i)$ with $\nu_{next}$, where $\theta_i$ is the heading of $\nu_i^*$. The neurons $\nu_{prev}$ and $\nu_{next}$ represent the previous and next neighbouring neurons of $\nu_i$, i.e., $prev < i$ and $next > i$, and they are determined according to the neighbouring function $f(\sigma, d)$ as the farthest neighbors for which $f(\sigma, d) \geq 10^{-4}$. It has been empirically observed that such a selection of $\nu_{prev}$ and $\nu_{next}$ provides better results than the immediate neighbouring neurons. Besides, $\sigma$ is decreasing after each learning epoch and, therefore, in later epochs, the immediate neurons are utilized which further support stabilization of the network. An example of the relation between the winner, neighbouring neurons, and the presented location to the network is visualized in Fig. 1.

Let headings associated to $\nu_i \in \mathcal{N}$ be $\Theta_i = \{\theta_i^{-k}, \theta_i^{-k+1} \ldots, \theta_i, \theta_i^k, \ldots, \theta_i^k\}$ then, the winner neuron $\nu^*$ is selected with the heading $\theta$ according to:

$$(\nu^*, \theta) = \operatorname{argmin}_{\nu_i \in \mathcal{N}, \nu_i \notin \mathcal{I}, \theta \in \Theta_i} \mathcal{L}(\nu_{prev}, (p, \theta)) + \mathcal{L}((p, \theta), \nu_{next}), \qquad (4)$$

where $\mathcal{I}$ denotes all neurons selected as winners in the current epoch. After that, the winner $\nu^*$ is adapted towards $p$ and its main heading is set to $\theta$. The neighbouring neurons are also adapted towards $p$, but only using the position as in the standard SOM for the ETSP. The neighbouring function $f(\sigma, d) = exp(-d^2/\sigma^2)$ for $d < 0.2m$, and $f(\sigma, d) = 0$ otherwise, is used for the adaptation.

Finally, to further improve convergence of the network and selection of the most suitable headings at the locations $P$, we update the main headings of the current winners after each learning epoch, i.e., after complete presentation of all locations $P$ to the network. Since each location $p$ has a unique winner, the order of winners in the output layer prescribes the sequence of visits to the locations. We consider the associated headings to the winners and construct all possible feasible Dubins paths connecting the locations $P$ in the sequence defined by the winners. The best heading for each winner is determined by a forward search, which time complexity can be bounded by $O(nk^3)$. In comparison to the winner selection with the time complexity $O(n^2k)$, this is negligible since $k \ll n$. Beside improving the headings at the winners, this also provides a feasible solution of the DTSP at the end of each learning epoch. The selection of the winners, their adaptation and ring regeneration is repeated until the solution is not improving or after reaching the maximal number of learning epochs. The overall adaptation procedure is summarized as follows.

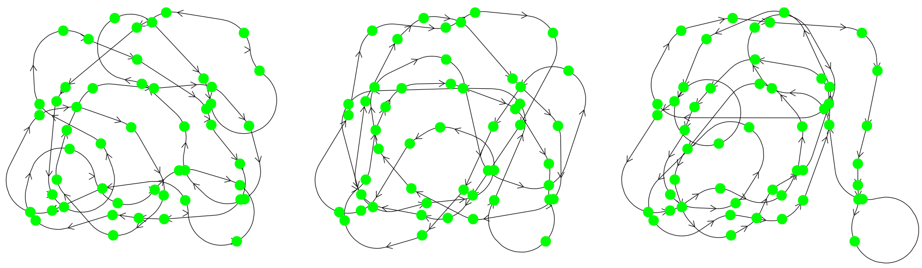
1. *Initialization:* For $n$ locations $P$ and the Dubins vehicle with the minimal turning radius $\rho$, create $2n$ nodes around the centroid of $P$ equidistantly placed on a circle with the radius $\rho$. The learning gain $\sigma$ is set to $\sigma = 12.41n + 0.06$, the learning rate $\mu$=0.6, and the gain decreasing rate $\alpha$=0.1 according to [13]. The epoch counter $i$ is set to 1, $i$=1.
2. *Randomizing:* Create a random permutation of the locations $\Pi(P)$.
3. *Learning epoch*: Clear inhibited neurons $\mathcal{I} = \emptyset$ and for each $p \in \Pi(P)$
   (a) *Select winner* $\nu^*$ and its heading $\theta$ for $p \in \Pi(P)$ using (4).
   (b) *Adapt* the winner and its neighbouring nodes to $p$ using $f(\sigma, d)$ and update headings of the winner according to the selected value $\theta$.
   (c) *Update* the inhibited neurons $\mathcal{I} = \mathcal{I} \cup \{\nu_i^*\}$.
4. *Ring regeneration:* Update headings of the current winners from the shortest Dubins path for the sequence of the locations defined by the winners in the ring and their associated headings.
5. *Update the learning gain and epoch counter:* $\sigma = \sigma(1 - \alpha)$, $i = i + 1$.
6. *Termination condition:* If solution is not improving or $i > i_{max}$ `Stop` the adaptation. Otherwise go to Step 2.
7. *Construct the final Dubins path* from the last winners and their headings.

An example of the final found solution is depicted in Fig. 1. Evaluation results and comparison with existing approaches are reported in the next section.

## 4 Experimental Results

The proposed SOM for the DTSP has been evaluated in several randomly generated problems with different numbers of locations $n$ and mutual distances between the locations. We consider a relative density $d$ of the $n$ locations and the minimal turning radius $\rho$ and generate the locations inside squared area with the side $s = (\rho\sqrt{n})/d$. In particular, we consider 20 instances for each $n \in \{10, 20, 50, 70, 100\}$ and $d \in \{0.3, 0.5, 1.0, 1.3, 1.7, 2.0\}$, which gives 600 different problem instances in total.

The performance of the proposed SOM algorithm has been compared with the AA [12] and LIO [14] heuristics and Memetic algorithm [16]. To evaluate the performance of the algorithms in so many instances, we consider the solution quality as the average ratio $R_L$ of the particular path length to the reference path length $L_{ref}$ and its standard deviation $\sigma_R$. Because optimal solution of the DTSP is not available, we consider the best found solution from all the solutions as $L_{ref}$. For providing high quality reference solutions, we consider the Memetic algorithm [16] with the computational time limited to one hour. On the other hand, for comparison with SOM and heuristics, we limit the computational time to 100 seconds to make the computational requirements of the Memetic algorithm competitive to SOM. Notice, AA and LIO are deterministic algorithms, while SOM is stochastic. Therefore, we performed 20 trials for SOM and each problem, which gives 13 800 trials in total. Only a single trial is performed by the Memetic algorithm to provide an overview of its convergence speed.



(a) Memetic–1 h, $L$=95 (b) Memetic–100 s, $L$=104 (c) SOM T=5.8 s, $L$= 102

**Fig. 2.** Selected found solutions for the same problem with $n$=50 locations and $d$=1.0

All the algorithms have been implemented in C++ and executed on a single core of the iCore7 CPU running at 3.4 GHz with 16 GB RAM and thus, the presented required computational times can directly compared.[1] The results are listed in Table 1, where $\mathrm{R'_L}$ is the average ratio of the best found solution for each problem from 20 trials. The standard deviation for $\mathrm{R'_L}$ is always less than 0.1 and typically around 0.05. Selected found solutions are shown in Fig. 2.

The fastest algorithms are the heuristics that provide a solution in less than one second, which includes optimal solution of the underlying ETSP by the Concorde [2]. Although the AA and LIO algorithms provides relatively good results for sparse problems, i.e., $d$=0.3, with increasing density, the solution quality is quickly decreased. The proposed SOM does not provide competitive results to the Memetic algorithm for sparse problems. However, with increasing density of the locations, SOM solutions are competitive with the Memetic algorithm with the running time limited to 100 seconds, while SOM provides solutions in less than 30 seconds.

[1] Reference solutions provided by the Memetic algorithm with 1 hour computational time has been found using a computational grid to decrease real time requirements.

**Table 1.** Average ratio of the solution length in the DTSP instances

| $d$ | $n$ | **ETSP-AA** [12] | | | **ETSP-LIO** [14] | | | **Memetic*** | | **Proposed** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $R_L$ | $\sigma_R$ | T [ms] | $R_L$ | $\sigma_R$ | T [ms] | $R_L$ | $\sigma_R$ | $R_L$ | $\sigma_R$ | T [s] | $R'_L$ |
| 0.3 | 10 | 1.35 | 0.14 | 3.7 | 1.32 | 0.15 | 9.4 | **1.04** | 0.06 | 1.11 | 0.10 | 0.5 | 1.00 |
| | 20 | 1.38 | 0.11 | 9.0 | 1.26 | 0.08 | 21.9 | **1.07** | 0.04 | 1.12 | 0.07 | 1.6 | 1.01 |
| | 50 | 1.35 | 0.11 | 52.7 | 1.27 | 0.08 | 148.0 | **1.10** | 0.06 | **1.10** | 0.06 | 7.2 | 1.01 |
| | 70 | 1.47 | 0.60 | 202.0 | 1.27 | 0.06 | 237.3 | 1.14 | 0.05 | **1.08** | 0.05 | 13.5 | 1.01 |
| | 100 | 1.31 | 0.06 | 346.5 | 1.24 | 0.05 | 641.8 | 1.23 | 0.04 | **1.07** | 0.05 | 26.2 | 1.00 |
| 0.5 | 10 | 1.65 | 0.20 | 4.7 | 1.73 | 0.31 | 11.5 | **1.09** | 0.09 | 1.23 | 0.16 | 0.5 | 1.02 |
| | 20 | 1.61 | 0.13 | 10.5 | 1.73 | 0.23 | 31.2 | **1.10** | 0.08 | 1.20 | 0.11 | 1.5 | 1.03 |
| | 50 | 1.71 | 0.54 | 60.9 | 1.68 | 0.08 | 145.7 | **1.10** | 0.05 | 1.14 | 0.07 | 7.4 | 1.03 |
| | 70 | 1.68 | 0.54 | 258.2 | 1.65 | 0.10 | 500.0 | 1.14 | 0.04 | **1.11** | 0.06 | 13.5 | 1.01 |
| | 100 | 1.50 | 0.05 | 766.8 | 1.63 | 0.10 | 587.9 | 1.23 | 0.05 | **1.09** | 0.05 | 25.8 | 1.01 |
| 1.0 | 10 | 1.72 | 0.21 | 6.6 | 2.33 | 0.27 | 14.8 | **1.12** | 0.14 | 1.29 | 0.18 | 0.4 | 1.06 |
| | 20 | 1.97 | 0.15 | 18.4 | 2.57 | 0.14 | 34.0 | **1.11** | 0.10 | 1.25 | 0.13 | 1.5 | 1.05 |
| | 50 | 1.94 | 0.11 | 93.4 | 2.63 | 0.16 | 174.2 | **1.12** | 0.06 | 1.17 | 0.08 | 7.3 | 1.03 |
| | 70 | 1.94 | 0.06 | 440.2 | 2.70 | 0.15 | 585.5 | 1.21 | 0.07 | **1.14** | 0.06 | 13.1 | 1.03 |
| | 100 | 1.93 | 0.07 | 332.8 | 2.63 | 0.12 | 537.9 | 1.31 | 0.06 | **1.11** | 0.06 | 25.1 | 1.01 |
| 1.3 | 10 | 1.64 | 0.17 | 7.0 | 2.35 | 0.22 | 14.8 | **1.12** | 0.10 | 1.29 | 0.15 | 0.4 | 1.04 |
| | 20 | 1.97 | 0.14 | 19.7 | 2.73 | 0.29 | 42.8 | **1.12** | 0.08 | 1.28 | 0.13 | 1.3 | 1.08 |
| | 50 | 2.09 | 0.10 | 113.7 | 3.04 | 0.11 | 183.8 | **1.13** | 0.07 | 1.18 | 0.08 | 7.0 | 1.04 |
| | 70 | 2.12 | 0.09 | 194.1 | 3.13 | 0.16 | 332.0 | 1.24 | 0.06 | **1.14** | 0.07 | 12.8 | 1.01 |
| | 100 | 2.05 | 0.07 | 443.0 | 2.97 | 0.11 | 540.6 | 1.32 | 0.06 | **1.10** | 0.06 | 24.4 | 1.00 |
| 1.7 | 10 | 1.56 | 0.15 | 6.6 | 2.31 | 0.28 | 11.7 | **1.12** | 0.11 | 1.30 | 0.17 | 0.4 | 1.07 |
| | 20 | 1.80 | 0.17 | 19.5 | 2.79 | 0.26 | 36.7 | **1.10** | 0.10 | 1.31 | 0.14 | 1.2 | 1.12 |
| | 50 | 2.16 | 0.12 | 90.6 | 3.34 | 0.21 | 174.2 | **1.17** | 0.07 | 1.20 | 0.10 | 6.7 | 1.05 |
| | 70 | 2.16 | 0.11 | 275.4 | 3.36 | 0.19 | 359.8 | 1.21 | 0.05 | **1.15** | 0.07 | 12.4 | 1.02 |
| | 100 | 2.24 | 0.11 | 354.3 | 3.46 | 0.19 | 554.3 | 1.33 | 0.06 | **1.11** | 0.07 | 23.7 | 1.00 |
| 2.0 | 10 | 1.40 | 0.08 | 7.8 | 2.17 | 0.18 | 13.7 | **1.11** | 0.11 | 1.26 | 0.13 | 0.4 | 1.06 |
| | 20 | 1.69 | 0.13 | 19.5 | 2.64 | 0.23 | 38.7 | **1.09** | 0.07 | 1.32 | 0.13 | 1.2 | 1.12 |
| | 50 | 2.11 | 0.15 | 120.7 | 3.36 | 0.23 | 268.4 | **1.16** | 0.10 | 1.25 | 0.11 | 6.5 | 1.10 |
| | 70 | 2.21 | 0.10 | 222.5 | 3.46 | 0.15 | 345.7 | 1.22 | 0.08 | **1.16** | 0.08 | 12.2 | 1.03 |
| | 100 | 2.25 | 0.10 | 339.5 | 3.59 | 0.19 | 541.0 | 1.33 | 0.07 | **1.11** | 0.06 | 23.6 | 1.00 |

*Computational time of the Memetic algorithm [16] has been limited to 100 seconds

## 5 Conclusion

We proposed probably the first SOM-based solution of the Dubins traveling salesman problem which includes challenges of the underlying combinatorial TSP with the continuous optimization of the headings at the locations. Although the results do not show significantly better solutions of SOM than a more computationally demanding Memetic algorithm, the results support feasibility of the proposed idea and better scalability for larger and denser problems.

The distance of the farthest neurons utilized in the winner selection influences how close the solution is to the underlying ETSP, which provides better results for sparse problems, or the adaptation is more focused on optimization of the headings. In this paper, we consider dense problems regarding the motivation of surveillance planning, because we aim to further deploy the proposed solver in more general problems with continuous sensing, i.e., sensing along the path and not only in a finite set of locations. This problem can be considered as the TSP with Neighborhoods, where SOM already exhibits its flexibility [6] for problems without curvature-constrained paths. The proposed SOM for the DTSP is an initial building block for solving this more general problem.

**Acknowledgments** – The presented work has been supported by the Czech Science Foundation (GAČR) under research project No. 16-24206S.

Computational resources were provided by the MetaCentrum under the program LM2010005 and the CERIT-SC under the program Centre CERIT Scientific Cloud, part of the Operational Program Research and Development for Innovations, Reg. No. CZ.1.05/3.2.00/08.0144.

## References

1. Angéniol, B., de la C. Vaubois, G., Texier., J.Y.L.: Self-organizing feature maps and the travelling salesman problem. Neural Networks 1, 289–293 (1988)
2. Applegate, D., Bixby, R., Chvátal, V., Cook, W.: Concorde tsp solver (2003), [cited 22 Jan 2016]
3. Cochrane, E.M., Beasley, J.E.: The co-adaptive neural network approach to the Euclidean travelling salesman problem. Neural Networks 16(10), 1499–1525 (2003)
4. Cook, W.: In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation. Princeton University Press (2012)
5. Dubins, L.E.: On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. American Journal of Mathematics pp. 497–516 (1957)
6. Faigl, J., Přeučil, L.: Self-Organizing Map for the Multi-Goal Path Planning with Polygonal Goals. In: International Conference on Artificial Neural Networks (ICANN). pp. 85–92. Springer, Heidelberg (2011)
7. Goaoc, X., Kim, H.S., Lazard, S.: Bounded-curvature shortest paths through a sequence of points using convex optimization. SIAM Journal on Computing 42(2), 662–684 (2013)
8. Le Ny, J., Feron, E., Frazzoli, E.: On the Dubins Traveling Salesman Problem. IEEE Transactions on Automatic Control 57(1), 265–270 (2012)
9. Oberlin, P., Rathinam, S., Darbha, S.: Today's Traveling Salesman Problem. IEEE Robotics & Automation Magazine 17(4), 70–77 (2010)
10. Obermeyer, K.J., Oberlin, P., Darbha, S.: Sampling-based path planning for a visual reconnaissance unmanned air vehicle. Journal of Guidance, Control, and Dynamics 35(2), 619–631 (2012)
11. Savla, K., Frazzoli, E., Bullo, F.: Traveling salesperson problems for the dubins vehicle. IEEE Transactions on Automatic Control 53(6), 1378–1391 (2008)
12. Savla, K., Frazzoli, E., Bullo, F.: On the point-to-point and traveling salesperson problems for Dubins' vehicle. In: Proceedings of the American Control Conference. pp. 786–791. IEEE (2005)
13. Somhom, S., Modares, A., Enkawa, T.: A self-organising model for the travelling salesman problem. Journal of the Operational Research Society pp. 919–928 (1997)
14. Váňa, P., Faigl, J.: On the dubins traveling salesman problem with neighborhoods. International Conference on Intelligent Robots and Systems pp. 4029–4034 (2015)
15. Yu, X., Hung, J.: A genetic algorithm for the dubins traveling salesman problem. In: IEEE International Symposium on Industrial Electronics. pp. 1256–1261 (2012)
16. Zhang, X., Chen, J., Xin, B., Peng, Z.: A memetic algorithm for path planning of curvature-constrained uavs performing surveillance of multiple ground targets. Chinese Journal of Aeronautics 27(3), 622–633 (2014)