Self-Organizing Map for Orienteering Problem with Dubins Vehicle

Jan Faigl
Czech Technical University in Prague
Department of Computer Science
Technická 2, 166 27 Prague, Czech Republic
Email: faiglj@fel.cvut.cz

***Abstract*—This paper reports on the application of the self-organizing map (SOM) to solve a novel generalization of the Orienteering Problem (OP) for curvature-constrained vehicles that is called the Dubins Orienteering Problem (DOP). Having a set of target locations, each with associated reward, and a given travel budget, the problem is to find the most valuable curvature-constrained path connecting the target locations such that the path does not exceed the travel budget. The proposed approach is based on two existing SOM-based approaches to solving the OP and Dubins Traveling Salesman Problem (Dubins TSP) that are further generalized to provide a solution of the more computational challenging DOP. DOP combines challenges of the combinatorial optimization of the OP and TSP to determine a subset of the most valuable targets and the optimal sequence of the waypoints to collect rewards of the targets together with the continuous optimization of determining headings of Dubins vehicle at the waypoints such that the total length of the curvature-constrained path is shorter than the given travel budget and the total sum of the collected rewards is maximized.**

I. Introduction

The problem addressed in this paper is motivated by data collection missions in which an Unmanned Aerial Vehicle (UAV) is requested to collect the most valuable measurements from a set of target locations while respecting the limited operational time of the UAV. Each target location has associated reward that is collected by the vehicle whenever the vehicle passes the location and takes a snapshot of the target surroundings by a downward-looking camera in surveillance missions [1] or using remote communication to read data from a sensor [2], [3]. The problem is formulated as a variant of the Orienteering Problem (OP) [4] that fits the requirements on the curvature-constrained data collection paths for UAVs. The considered problem formulation as a suitable form to address data collection missions with curvature-constrained vehicles is called the Dubins Orienteering Problem (DOP).

Having a set of target locations each with the associated reward and specified the initial and final locations of the vehicle, the regular OP stands to maximize the sum of the collected rewards by a tour that does not exceed the given travel budget [6]. The OP is similar to the combinatorial optimization the Traveling Salesman Problem (TSP) in finding the shortest tour connecting the given locations. However, contrary to connecting all the locations by the shortest tour in the TSP, a subset of the most valuable locations are connected by the shortest tour in the OP to respect the given travel budget. Therefore, the OP combines the Knapsack problem in choosing the most rewarding locations and the TSP in finding the shortest tour connecting the waypoints. In a case the travel budget allows to visit all the target locations, the OP becomes the TSP; hence the OP is at least NP-hard [4].

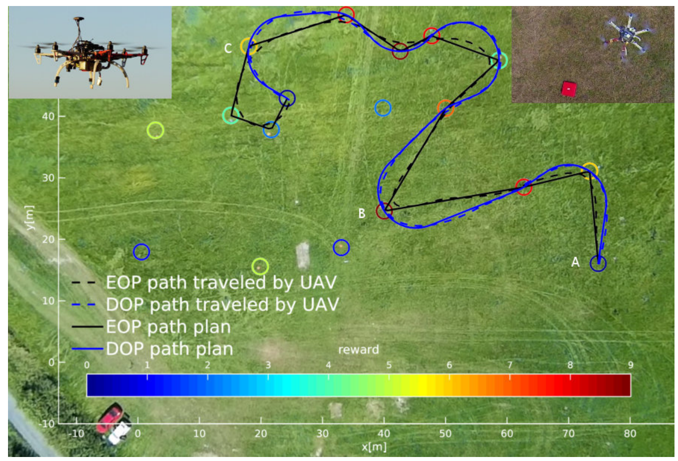


Fig. 1. An example of real deployment of the UAV in data collection missions and necessity to directly solve Dubins Orienteering Problem (DOP) [5]. A solution of the Euclidean Orienteering Problem (EOP) provides a path with sharp turns which require to slow down the vehicle (which may not be possible for fixed-wing UAV), or the vehicle misses the target locations because of steering the vehicle motion by the path following controller.

An important generalization of the OP to address the motivational data collection is related to the kinematics constraints of UAVs, especially fixed-wing aircraft, which needs a curvature-constrained path. A widely used vehicle model is Dubins vehicle [7] and the generalized TSP to determine the shortest curvature-constrained path visiting a set of target locations is called the Dubins Traveling Salesman Problem (DTSP) [8]. The main difficulty of solving the routing problems with Dubins vehicle is that the length of the path depends not only on the distance between the waypoints but also on the particular headings of the vehicle at each waypoint. Hence, the DTSP combines challenges of the combinatorial optimization of the TSP with the determination of the most suitable headings which is a continuous optimization problem as the vehicle heading at each target location can be arbitrarily selected from the interval $[0, 2\pi)$. Due to the computational complexity of the DTSP, heuristics [9], [10], sampling-based algorithms [11], evolutionary approaches [12], and recently also SOM-based approaches [13], [14] have been developed.

Despite many approaches for the Dubins TSP have been proposed, there is an only little effort to the generalization of the Orienteering Problem for planning with Dubins vehicle, i.e., the Dubins Orienteering Problem (DOP). The motivation and importance of solving DOP is demonstrated in Fig. 1, where a solution of the regular Euclidean OP does not conform motion limits of the real utilized UAV. A possible way to solve DOP can be based on the decoupled approach in which a subset of the target locations to be visited by the vehicle is determined as the regular Euclidean OP. Then, the final data collection path can be found as a solution of the DTSP for such a subset. However, such a tour may exceed the given travel budget, and therefore, a direct solution of DOP is preferred. To the best of our knowledge, the first such a direct approach for solving DOP has been proposed in [5], where the randomized variant of the Variable Neighborhood Search (VNS) technique [15] has been augmented to deal with the constraints of Dubins vehicle. The algorithm follows sampling-based approach for the DTSP and possible headings at each target locations are sampled into a finite set of headings, and the problem is solved as a combinatorial optimization. In [5], 16 heading values per each sensor location have been used as a trade-off between the solution quality and required computational time.

The recent advancements on SOM for the OP and DTSP are the main sources of motivation to address the DOP by unsupervised learning and eventually avoid dense sampling of the heading values necessary in the VNS-based approach [5]. The proposed approach is directly leveraging on the existing SOM-based solution of the OP [16], [17] that is combined with the principles employed in solving the DTSP proposed in [13] and [14]. Although an application of SOM to solving a combinatorial problem may be questionable, the main expected benefit of the unsupervised learning is in solving the continuous optimization part of the problem, where it is needed to determine the most suitable headings of the vehicle at the waypoints. The presented results indicate that the proposed SOM-based approach may provide Dubins tours of similar length with a lower number of samples of the vehicle headings, which significantly speedups finding a solution of DOP.

The rest of the paper is organized as follows. The addressed problem is formally introduced in the next section. The proposed SOM-based approach is presented in Section III. The empirical evaluation of the proposed solution and comparison with the VNS-based approach is presented in Section IV. Concluding remarks and future work is in Section V.

II. Problem Statement

The addressed Dubins Orienteering Problem (DOP) is a combination of the combinatorial Orienteering Problem (OP) with the continuous optimization of headings at the waypoints. The problem is to determine a cost efficient curvature-constrained path to retrieve the most valuable measurements from a set of locations S placed in a plane $S \subset \mathbb{R}^2$ such that, the total tour length does not exceed the given travel budget T_{max}. Each sensor $s_i \in S$ has associated reward r_i that can be collected by the vehicle that visits the location s during its travel along the data collection path. Regarding the existing formulations of the OP [18], the starting and final locations of the vehicle are prescribed as s_1 and s_n and their associated rewards are zero, $r_1 = r_n = 0$, where n is the total number of locations S, $n = |S|$. We aim to determine the maximal rewarding data collection path that satisfies the kinematic constraint of Dubins vehicle.

The state of Dubins vehicle can be described as $q = (x, y, \theta)$ where $p = (x, y)$ is the vehicle position in the plane $p \in \mathbb{R}^2$ and θ is the vehicle heading $\theta \in [0, 2\pi)$, i.e., $\theta \in \mathbb{S}^1$, and thus $q \in SE(2)$. The model of Dubins vehicle [7] assumes the vehicle is moving with a constant forward velocity v and its minimal turning radius is ρ. Having the control input u, the vehicle motion can be described as

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

The shortest curvature-constrained path respecting (1) that connects two states $q_i, q_j \in SE(2)$ is a straight line segment (S) or consists of S and arcs with the curvature ρ that can be one of two types: left (L) and right (R). The optimal path connecting q_i and q_j can be computed analytically and the path is one of six possible Dubins maneuvers: LSL, LSR, RSL, RSR, LRL, and RLR [7]. However, to determine the optimal path connecting two states in DOP, we first need to determine the respective sensor locations $s_i, s_j \in \mathbb{R}^2$ and also the particular headings $\theta_i, \theta_j \in \mathbb{S}^1$ at these locations. Moreover, it may not be possible to retrieve data from all the locations S within the travel budget T_{max}, and therefore, it is necessary to select a subset S_k of k locations $S_k \subseteq S$ from which data can be retrieved by Dubins vehicle (1) traveling along Dubins path with the length that does not exceed T_{max}.

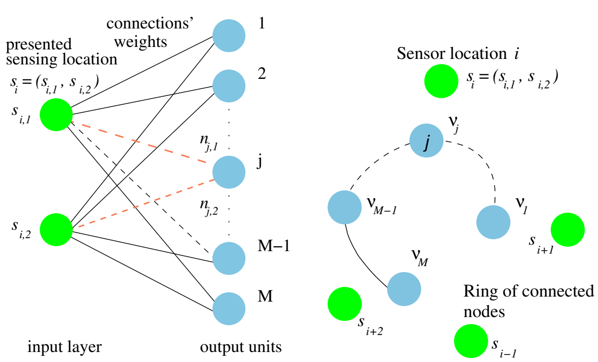
The initial and final locations of the vehicle are prescribed, $s_1 \in S_k$ and $s_n \in S_k$, and thus we need to determine a sequence of waypoints $(q_{\sigma_1}, \ldots, q_{\sigma_k})$, where $0 \leq \sigma_i \leq n$ and $q_{\sigma_1} = (s_1, \theta_1)$ and $q_k = (s_n, \theta_n)$, such that each waypoint $q_{\sigma_i} = (s_{\sigma_i}, \theta_i)$ consists of the sensor location $s_{\sigma_i} \in \mathbb{R}^2$ and the suitable heading at s_{σ_i} with respect to Dubins maneuvers connecting the waypoints. Having the k sensor locations S_k, the problem to find the shortest Dubins path connecting the waypoints, i.e., determining the sequence to their visits and the respecting headings, can be considered as the Dubins Traveling salesman problem (DTSP); however, the required data collection path has to fulfill the travel budget T_{max}. Therefore, during the optimization, we need to search for the k sensor locations $S_k = (s_{\sigma_1}, \ldots, s_{\sigma_k})$, the permutation of their visits $\Sigma = (\sigma_1, \ldots, \sigma_k)$, and the respecting headings $\Theta = (\theta_{\sigma_1}, \ldots, \theta_{\sigma_k})$. Having these preliminaries, the Dubins Orienteering Problem (DOP) can be formulated as the optimization problem:

$$\begin{aligned} \underset{k, S_k, \Sigma, \Theta}{\textit{maximize}} \quad & R = \sum_{i=1}^{k} r_{\sigma_i} \\ \textit{subject to} \quad & \sum_{i=2}^{k} \mathcal{L}\left(q_{\sigma_{i-1}}, q_{\sigma_i}\right) \leq T_{max}, \\ & q_{\sigma_i} = (s_{\sigma_i}, \theta_{\sigma_i}), s_{\sigma_i} \in S_k, s_{\sigma_i} \in \mathbb{R}^2, \theta_{\sigma_i} \in \mathbb{S}^1, \\ & s_{\sigma_1} = s_1, s_{\sigma_k} = s_n, \end{aligned} \tag{2}$$

where R is the sum of the collected rewards and $\mathcal{L}(q_{\sigma_{i-1}}, q_{\sigma_i})$ is the length of Dubins maneuver from $q_{\sigma_{i-1}}$ to q_{σ_i} [7].

III. Self-Organizing Map for the Dubins Orienteering Problem

The proposed approach to solving the Dubins Orienteering Problem (DOP) leverages on the previous application of the Self-Organizing Maps (SOMs) to the Orienteering Problem (OP) proposed in [19], [16], [17] and SOM for the Dubins Traveling Salesman Problem (DTSP) introduced in [13] and later generalized for multi-vehicle missions in [14]. The proposed solution can be considered as a direct combination of the adaptation procedures for the OP and DTSP. However, the solution has to respect the limited travel budget T_{max} and due to the strong dependence of the length of Dubins path on the particular waypoints the conditional adaptation (introduced in [20] and used for the Euclidean OP in [16], [14]) cannot be directly utilized. The representation of the solution in the SOM structure has to allow an evaluation of the budget constraint during the unsupervised learning, and therefore, the winner neuron is selected not as the closest neuron to the presented location, but as the closest point of Dubins tour represented by the current ring. This is the main source of the proposed modifications of the existing SOM approaches for the OP and DTSP that are described in the following parts of this section. Here, it is worth noting that even though the overall idea of the adaptation follows SOM for the OP and DTSP, and thus the description of the proposed SOM for DOP can be referenced to the previous work, a full description of the main ideas behind the proposed learning method are presented to make the paper self-contained.



(a) Two-layered SOM (b) Ring of neurons

Fig. 2. Structure of SOM for solving orienteering problems

The basic structure of the employed SOM neural network follows the structure of SOM for solving routing problems, see Fig. 2. The input layer serves for presenting the particular sensor locations towards which the prototypes are adapted. The output layer consists of connecting neurons that are organized into an array of output units that form a ring of nodes in the input space $\mathbb{R}^2$. The employed unsupervised learning is an iterative procedure in which the network grows by adding new neurons that are adapted towards the presented input, e.g., as in the early work on SOM for the TSP [21]. After each learning epoch (i.e., presentation of all input locations to the network), all nonwinning neurons (for the current epoch) are removed to reduce the computational burden and improve the network convergence [22]. The representation of the solution by the ring is crucial to evaluate the solution cost and respect the limited travel budget considered in orienteering problems [17].

The connected ring of nodes itself represents a path in $\mathbb{R}^2$ which evolves during the learning by adaptation of the winner neurons and the respective neighboring neurons towards the presented locations. After the network stabilization, the winners fit the locations, and the ring becomes the requested tour. Besides, the associated sensor locations to the particular winners represent the final tour over the requested locations, and thus a solution of the TSP can be retrieved after each learning epoch by traversing the ring. In the herein addressed DOP, this idea is further generalized and the solution represented by the network is utilized during the winner selection and adaptation to satisfy the budget constraint.

The important difference of the Orienteering problem (OP) and regular TSP is that it may not be necessary (or possible) to visit all locations in the OP and we are searching for a subset of the locations such that it maximizes the sum of the collected rewards R and the final tour satisfies the travel budget T_{max}. Therefore, the solution is represented as the sequence of the associated locations to the winners obtained by traversing the output layer. Such a tour is utilized to respect the maximal allowed travel budget T_{max} and the network is not adapted towards the locations if such an adaptation would result in a tour that would violate the budget T_{max} [16], i.e., the adaptation of the network is conditioned to the expected tour length after the adaptation. The length of the tour can be directly computed for the Euclidean OP [16], which is not the case of the OP with Dubins vehicle, where the Dubins tour has to satisfy the kinematic model (1) and the tour strongly depends on the waypoints and the headings. In DOP, we need to determine the most suitable heading values at each waypoint to have a short Dubins tour connecting the selected sensors, and thus the expected headings are determined during the winner selection similarly as in the Dubins Traveling Salesman Problem (DTSP) [14].

The minimal turning radius ρ of Dubins vehicle is addressed by determination of the optimal Dubins maneuvers connecting the expected waypoints to retrieve data from the subset S_k. Therefore each neuron ν_i has associated one main heading θ^i and up to h additional headings $\{\theta^i_1, \ldots, \theta^i_h\}$ that are utilized to determine the shortest Dubins path connecting the waypoints represented by the winner neurons. Contrary to the previous SOM-based approaches for the DTSP [13], [14] the herein proposed SOM-based approach for DOP does not utilize the Dubins path connecting the neurons but directly the waypoints associated with the winners are used. It is because we need to evaluate the length of Dubins tour over the waypoints to meet the requirements on the travel budget T_{max}.

The number of neurons in the used growing self-organized neural network [17] is changing during the learning, and it corresponds to the number of the selected locations $S_k \subseteq S$, but the network always has at least two neurons: the first one for the initial location s_1 and the last neuron for the final location s_n. Let the current number of neurons be m and the ring of neurons be $\mathcal{N} = \{\nu_1, \ldots, \nu_m\}$, then ν_1 corresponds to s_1 and ν_m to s_n, see visualization of Dubins tour represented by the ring as the red curve in Fig. 3. Each neuron ν_i has associated waypoint q_i consisting of the vehicle heading θ^i and the sensor location s_i. For ν_1 and ν_m the locations are s_1 and s_n, respectively, because of the prescribed initial and final

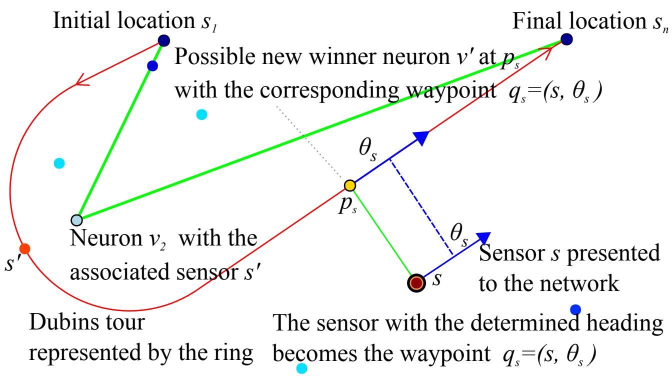


Fig. 3. Proposed selection of the possible winner neuron together with the expected heading at the particular waypoint. The green straight line segments denote connected ring of neurons. Each neuron has associated sensor location $s \in S$ for which the neuron has been selected as the winner neuron, a particular heading θ, and a set of the supporting headings Θ. The showed ring consists of the first neuron at the initial location s_1, the neuron ν_2 with the associated the left most sensor visualized as the small red disk, and the last neuron corresponding to the final location s_n. The current sensor presented to the network is s. Dubins tour represented by the current ring is shown as the red curve, and the closest point on the curve to s is denoted by the point p_s. The point p_s corresponds to the weights of possible winner neuron ν' which has the associated state (s, θ_s), where the heading θ_s is the heading on the red Dubins tour at the point p_s.

locations of the data collection path.

The learning of the network is performed for the fixed number of learning epochs and for each learning epoch, all sensor locations S are presented to the network in a random order to avoid local optima. The important part of the proposed adaptation is that the network is adapted to the particular $s \in S$ only if Dubins tour represented by the current ring would be shorter than T_{max} after the adaptation. Therefore, a possible winner neuron ν' for the presented s is determined (see Fig. 3) and it is tried to adapt the network towards s. If the length of Dubins tour represented by the ring after the adaptation would be longer than T_{max} there are probably too much-associated sensors to the ring and the network is reverted to the state prior such an adaptation and up to two neurons are removed. After that, the adaptation of the possible winner neuron ν' is performed one more time and if Dubins tour represented by the ring after the adaptation is shorter than T_{max} the network is adapted; otherwise the network is reverted to the state prior the selection of the possible winner neuron ν' for s. Then, the learning continues with the next sensor.

Notice, the utilized SOM-based neural network is a bit more complex than a traditional SOM for the TSP. It is because each neuron ν_i (in addition to the neuron weights corresponding to a location in $\mathbb{R}^2$) is also associated to the particular sensor s_{ν_i} with the heading θ^{ν_i}, and finally it is also associated with a set of h supporting headings $\Theta^{\nu_i} = \{\theta_1^{\nu_i}, \ldots, \theta_h^{\nu_i}\}$. The headings θ^{ν_i} and Θ^{ν_i} are utilized in the determination of Dubins tour represented by the ring during which the most suitable heading for each particular waypoint is selected according to the other waypoints in the sequence. Thus, such a heading is then set as the main heading of each particular neuron. Moreover, the additional difference of the proposed SOM for DOP and existing SOM for the Euclidean OP [16], [17] is in the winner selection and the adaptation because of Dubins vehicle. For the Euclidean problem, the connected ring of the neurons directly represents the requested data collection path. In the proposed DOP procedure, the winner is determined according to the Dubins tour (see Fig. 3) but the neuron weights are adapted using the regular adaptation. The new neuron weights ν' are set according to the previous weights ν, the sensor location s, and the neighboring function $f(G, d)$ as

$$\nu' = \nu + R_s \mu f(G, d)(s - \nu), \tag{3}$$

where μ is the learning rate ($\mu = 0.6$), R_s is the ratio of the reward of the sensor s and the maximal reward of the sensors in the set S, G is the learning rate and d is the distance of the neuron ν from the winner neuron ν^* in the ring (i.e., in the number of neurons in the output layer). The neighboring function has the standard form

$$f(G, d) = \begin{cases} e^{\frac{-d^2}{G^2}} & \text{for } d \text{ in the activation bubble of } \nu^* \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

which decreases the power of adaptation of the neighbouring nodes to the winner neuron ν^* with increasing distance d. The activation bubble is determined with respect to the Dubins tour [14] and the active neighborhood of ν^* is defined by the neurons ν_{prev} and ν_{next} that are determined from the neurons within the distance $d \leq 0.2m$ from ν^*, where m is the current number of the neurons in the ring. The neurons ν_{prev} and ν_{next} are selected such that the length of Dubins path to visit the waypoint at s is minimized:

$$L_g = \mathcal{L}(s_{\nu_{prev}}, (s, \theta)) + \mathcal{L}((s, \theta), s_{\nu_{next}}), \tag{5}$$

where θ is one of the heading values $\theta \in \{\theta^{\nu^*}\} \bigcup \Theta^{\nu^*}$ of the winner neuron ν^* and $s_{\nu_{prev}}$ and $s_{\nu_{next}}$ are the corresponding waypoints (sensor locations with the determined main heading) of ν_{prev} and ν_{next}, respectively.

Finally, the adaptation procedure not only adds new neurons to the network but also removes them during the evaluation of the possible adaptation. Up to two neurons (ν_f and ν_l) can be removed if Dubins tour represented by the ring exceeds the travel budget T_{max}. ν_f is the neuron with the longest distance from its associated sensor s_{ν_f}, i.e., one of the neurons that do not fully adapt to the respective waypoint yet. ν_l is the neuron which is associated with the sensor location with the lowest reward. Besides, all non-winner neurons are removed from the ring after each learning epoch. The proposed SOM-based unsupervised learning procedure for solving the Dubins Orienteering Problem (DOP) is summarized in Fig. 4.

A. Computational Complexity

Due to Dubins vehicle with the kinematic model (1), the real computational requirements of the proposed SOM-based solution of DOP are higher than in the case of computing just the Euclidean distance between the locations. Regarding the computational complexity, the learning procedure depends on the number of locations n and also on the number of additional headings h that support finding a shorter Dubins tour to visit the particular waypoints. The number of neurons m never exceeds the number of sensor locations, i.e., $m \leq n$, because of removing neurons during the adaptation (Step 6(d)ii in Fig. 4) and after each learning epoch (Step 7 in Fig. 4), and thus the number of neurons can be bounded by n. For

▷ **Initialization:**

1) Initialize the ring $\mathcal{N} = (\nu_1, \nu_{end})$ with the neurons ν_1 and ν_{end} corresponding to the sensors s_1 and s_n. These two neurons are never removed nor adapted during learning. The number of supporting heading values per each neuron is set to h, e.g., $h = 3$.
2) Set the learning parameters: the learning gain $G = 10$, the learning rate $\mu = 0.6$, and the gain decreasing rate $\alpha = 0.1$.
3) Set the current best found solution $T = (s_1, s_n)$ and its sum of rewards $R = 0$ because $r_1 = r_n = 0$.
4) Set the learning epoch counter i to $i = 1$.

▷ **Learning epoch:**

5) Randomize the sensor locations $S = \{s_1, \ldots, s_n\}$ except s_1 and s_n; $\Pi \leftarrow \mathrm{permute}(S \setminus \{s_1, s_n\})$.
6) For each $s \in \Pi$:

▷ **Conditional Adapt:**

a) Save the current network $\mathcal{N}' \leftarrow \mathcal{N}$
b) $(\nu', p_s, \theta_s, s) \leftarrow$ *determine a possible winner neuron* ν' for s at the location p_s with the heading θ_s and the waypoint location at s regarding the current ring $\mathcal{N}$, and the minimal turning radius ρ, see Fig. 3.
c) $\mathcal{N} \leftarrow \mathrm{adapt}(\mathcal{N}, \nu', \theta_s, s)$ – adapt the network $\mathcal{N}$ with the new winner ν' towards s according to (5) and determine the length of Dubins tour represented by the ring as $\mathcal{L}(\mathcal{N})$. The new neuron ν' is added to the network, and thus the network grows during the adaptation.
d) `If` $\mathcal{L}(\mathcal{N}) > T_{max}$ `Then`
   i) $\mathcal{N} \leftarrow \mathcal{N}'$ – Revert the changes of the adaptation and determine neurons ν_f and ν_l as the neuron with the farthest associated s_{ν_f} and the neuron with the associated sensor location with the lowest reward, respectively.
   ii) $\mathcal{N} \leftarrow \mathrm{adapt}(\mathcal{N} \setminus \{\nu_f, \nu_l\}, \nu', \theta_s, s)$ – adapt the ring without ν_f and ν_l and determine $\mathcal{L}(\mathcal{N})$.
   iii) `If` $\mathcal{L}(\mathcal{N}) > T_{max}$ `Then`
      $\mathcal{N} \leftarrow \mathcal{N}'$ – Revert the changes to the ring and the network is not adapted towards s.

▷ **Update** *(at the end of each learning epoch)*:

7) Remove all non-winner neurons from the ring $\mathcal{N}$.
8) Update learning parameters: $G \leftarrow G(1 - i\alpha)$, $i \leftarrow i + 1$.
9) `If` the Dubins tour T_{win} represented by the current ring has higher sum of the collected rewards than T update the best solution found so far $T \leftarrow T_{win}$ and its sum of the collected rewards $R \leftarrow \sum_{s_i \in T} r_{s_i}$.
10) `If` $i < i_{max}$ repeat the learning for the next learning epoch (Step 5); `Otherwise: Stop` the learning.

Fig. 4. A summary of the proposed SOM-based adaptation procedure of the self-organizing map for the Dubins Orienteering Problem (DOP)

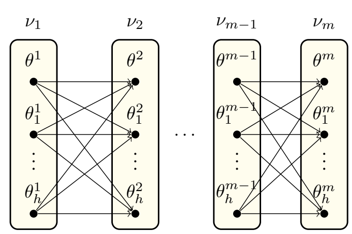


Fig. 5. Each neuron ν_i consists of its position in $\mathbb{R}^2$ as the neuron weights, but it also has associated the main heading θ^i and to support finding Dubins tour, each neuron may have up to h additional headings. The sequence of the neurons (winners for the current epoch) provides a sequence of the waypoints, and the neurons' headings are utilized to construct a search graph in which each layer corresponds to one neuron ν_i with the heading values $\Theta^i = \{\theta^i, \theta^i_1, \ldots, \theta^i_h\}$. Two neighboring layers are fully connected by the oriented edges representing the direction of the vehicle.

each learning epoch, all sensor locations S are presented to the network, and for each location $s \in S$, a possible winner neuron is selected. The selection of the winner is based on the determination of Dubins tour represented by the ring which depends on the number of heading values h associated to each neuron and the forward search on the graph, which is shown in Fig. 5. The complexity of the search procedure can be bounded by $O(nh^2)$ for the worst case of n neurons in the ring. Dubins tour represented by the ring is used for evaluation of the budget constraint after the adaptation, which, in the worst case, is performed two times for each presented sensor location s to the network (Step 6c and Step 6(d)i in Fig. 4). Thus, the computational complexity of a single winner selection and adaptation can be bounded by $O(3n^2h^2)$, and the computational complexity of a single learning epoch can be bounded by $O(3n^3h^2)$.

IV. Results

The proposed SOM-based solution of the Dubins Orienteering Problem (DOP) has been evaluated using selected problem instances of the existing standard benchmarks for the OP [23], already utilized in the first introduction of DOP in [5]. The only known existing solver to the DOP has been proposed in [5], and therefore, the proposed SOM-based approach is compared with the VNS-based combinatorial optimization approach [5], and thus this reference solution is denoted the VNS in the rest of this paper.

The benchmarks for the OP consists of a set of problems with particular locations and rewards accompanied with a specific value of the travel budget T_{max} [24]. The considered benchmark problems of the Set 1, Set 2, and Set 3 proposed by Tsiligirides [25] and diamond-shaped Set 64 and squared-shaped Set 66 [26] are accompanied with budget values from the range 5 to 130 which gives 89 instances of the OP. For DOP, each instance can be further specified by the minimal turning radius ρ, e.g., from the set $\rho \in \{0, 0.5, 1.0, 1.5\}$, which can give 356 instances of DOP. Due to such an excessive number of instances and the novelty of the proposed SOM-based solution to DOP, it is rather preferred to focus on the evaluation of the influence of ρ and h to the problem and its solution. Therefore, the problem instances listed in Table I have been selected for the presented performance evaluation. The selected instances are for the budgets in the middle of the

TABLE I. SELECTED INSTANCES OF THE OP

Set 1	Instance with 32 locations and $T_{max} = 46$
Set 2	Instance with 21 locations and $T_{max} = 30$
Set 3	Instance with 33 locations and $T_{max} = 50$
Set 64	Instance with 64 locations and $T_{max} = 45$
Set 66	Instance with 66 locations and $T_{max} = 60$

range of budgets for the particular problem set, which represent the most challenging instances as for high budgets most of the target locations can be visited, and thus the problem becomes a variant of the DTSP. On the other hand, for small budgets, the problem is more about picking the highest rewarding target locations reachable within the budget.

Both evaluated algorithms, the VNS and SOM are stochastic, therefore each problem instance is solved 20 times and the reported results are average values accompanied with the respective standard deviations, i.e., the sum of the collected reward R and the required computational time T_{cpu}. Also, the best-found solution from 20 trials is reported as R_{max} for the individual problem instances and algorithms. The parameters of the VNS algorithm follow the results presented in [5] where the number of heading values per each target is 16, and the stopping criterion is the maximal number of 10 000 iterations with the maximal 3 000 iterations without improvement. The proposed SOM-based algorithm has only two parameters: the number of learning epochs i_{max} and the number of additional supporting headings h. The number of learning epochs i_{max} has been experimentally set to $i_{max} = 150$ and h is considered from the set $h \in \{0, 3, 6, 9\}$.

The VNS and SOM based algorithms have been implemented in C++ and run within the same computational environment using a single core of the iCore7 CPU running at 4 GHz. The VNS algorithm pre-computes all the Dubins maneuvers between all the target locations and all heading values prior the combinatorial optimization with an optimized structure for inserting/removing waypoints into Dubins tour. On the other hand, SOM-based approach searches for the suitable headings during the unsupervised learning. Dubins maneuvers are computed on demand, and for the presented early results no special optimizations in Dubins tour computation are utilized.

The proposed SOM-based approach has been evaluated for the selected DOP instances together with the study of the influence of the number of supporting headings h. The average values with the standard deviations showed as error bars are visualized for the selected instances in Fig. 6 and the overview of the respective required computational times in Fig. 7. The bar plots clearly show that increasing h improves the solution at the cost of the increased computational burden, but the improvement is noticeable only for $h = 3$ and more headings only increase the computational burden. Thus, it seems that a suitable choice is $h = 3$ for which detailed results are presented in Table II. Selected solutions found by the proposed SOM are shown in Fig. 10.

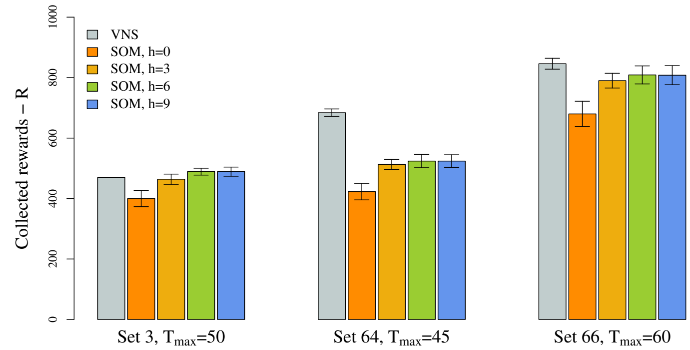


Fig. 6. Solution quality as an average sum of the collected rewards for VNS [5] and the proposed SOM according to the number of the headings h in problems with $\rho = 1$

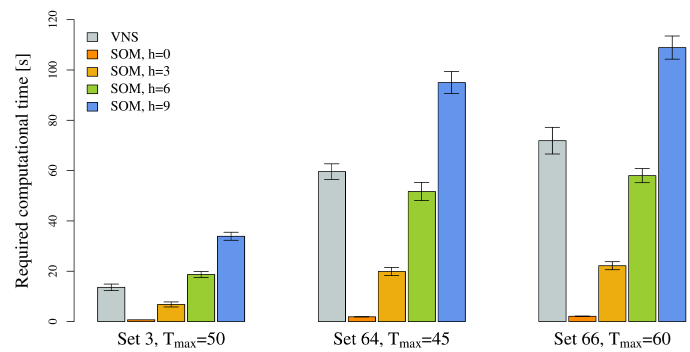


Fig. 7. Average required computational time for VNS [5] and the proposed SOM according to the number of the headings h in problems with $\rho = 1$

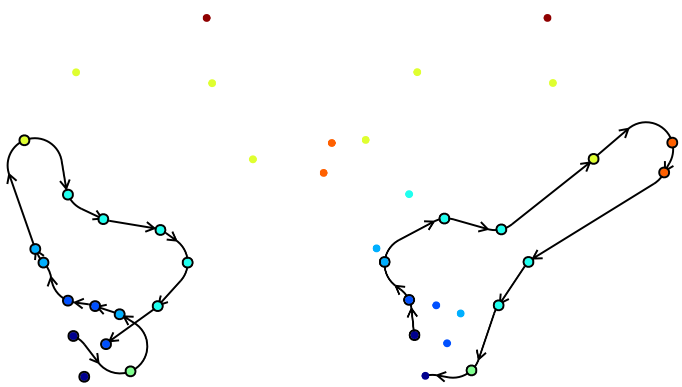
TABLE II. RESULTS FOR THE DUBINS ORIENTEERING PROBLEM

Problem instance	VNS (16 headings) [5]			SOM ($h = 3$)		
(T_{max}, ρ)	R_{max}	R	T_{cpu} [s]	R_{max}	R	T_{cpu} [s]
Set 1 (46, 0.5)	175	168	11.0	170	168	5.1
Set 1 (46, 1.0)	165	158	11.3	160	153	6.3
Set 1 (46, 1.5)	140	135	10.2	140	126	5.0
Set 2 (30, 0.5)	255	252	4.4	255	249	2.3
Set 2 (30, 1.0)	230	230	4.8	240	221	2.9
Set 2 (30, 1.5)	210	210	3.6	195	181	2.2
Set 3 (50, 0.5)	510	508	14.1	510	492	6.1
Set 3 (50, 1.0)	470	470	13.6	480	464	6.8
Set 3 (50, 1.5)	440	430	12.4	440	403	6.1
Set 64 (45, 0.5)	792	778	61.2	744	697	24.5
Set 64 (45, 1.0)	702	684	59.6	540	513	19.9
Set 64 (45, 1.5)	636	603	55.3	456	413	14.5
Set 66 (60, 0.5)	895	850	63.2	860	814	22.9
Set 66 (60, 1.0)	890	846	71.9	835	790	22.2
Set 66 (60, 1.5)	795	722	61.5	765	698	18.0

The results in Table II indicates that for the instances of the Set 1, Set 2, and Set 3 the proposed SOM provides competitive results regarding the best-found solution among the performed trials. In the case of the Set 64 and Set 66, with the locations positioned in a grid, the VNS-based approach provides better results with about 2–3 times higher computational requirements.

A. Discussion

The presented results support the feasibility of the proposed SOM-based approach to solve computationally challenging instances of DOP. Although the combinatorial optimization VNS-based approach provides overall better results than the proposed SOM approach, the main advantage of the proposed unsupervised learning is the ability to solve the problem with very few supporting headings h, which results in lower computational requirements. Even though the early results have been achieved by non-optimized implementation of the Dubins tour computation during the learning (contrary to the precomputed structure in the case of VNS [5]), the SOM-based approach is less computationally demanding for $h = 3$.

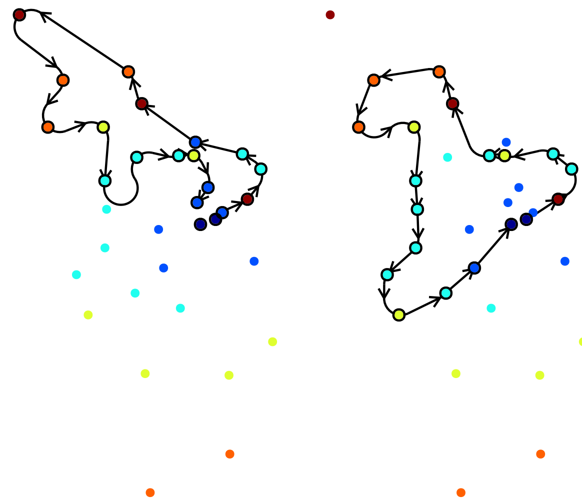


(a) VNS, R=230, $\mathcal{L} = 28.43$ (b) SOM, R=240, $\mathcal{L} = 29.65$

Fig. 8. Best solutions for the Set 2 problem instance with $T_{max} = 30$ and $\rho = 1.0$ found by the VNS [5] and the proposed SOM-based approach. The $\mathcal{L}$ denotes the length of the found Dubins tour. The VNS approach found the solution in $\mathrm{T_{cpu}} = 4.8$ sec while the proposed SOM-based approach found the solution in $\mathrm{T_{cpu}} = 2.9$ sec using the same computational environment.

On the other hand, the simple local improvements utilized in the proposed conditional adaptation (i.e., the deletion of the ν_f and ν_l neurons) provide only limited capability of improving the solution and a more systematic search in the VNS provides better results at the cost of higher computational requirements. Therefore, it is expected that additional local improvements combined with the memetic algorithm would improve the solution similarly as in [27] for the Euclidean TSP, while preserving the benefit of the SOM approach with only a few supporting heading values. Note that for Set 2 and Set 3 the best-found solutions for $\rho = 1$ have been found by the SOM approach with $h = 3$ while the VNS with 16 headings per waypoint provides a bit worse solution, e.g., see Fig. 8 and Fig. 9. In these cases, the VNS stucks in local optima and all performed trials the solutions are identical. The proposed SOM-based approach provides better results in few trials for the instance of Set 2, and for Set 3 the solution with the highest reward $R = 480$ is found in 8 of 20 trials. Individual solutions are slightly different, but the sum of the collected rewards is the same $R = 480$. Even though these examples are the only case when SOM provides better results than the VNS approach, this behavior indicates the SOM-based approach is capable of finding good solutions with lower computational requirements.

V. Conclusion



(a) VNS, R=470, $\mathcal{L} = 49.74$ (b) SOM, R=480, $\mathcal{L} = 48.53$

Fig. 9. Best solutions for the Set 3 problem instance with $T_{max} = 50$ and $\rho = 1.0$ found by the VNS [5] and the proposed SOM-based approach. The $\mathcal{L}$ denotes the length of the found Dubins tour. The VNS approach found the solution in $\mathrm{T_{cpu}} = 12.1$ sec while the proposed SOM-based approach found the solution in $\mathrm{T_{cpu}} = 6.7$ sec using the same computational environment.

In this paper, a novel SOM-based adaptation procedure is proposed to solve the Dubins orienteering problem. The proposed approach leverages on the SOM adaptation procedures for the Euclidean OP and DTSP, and the presented results support the feasibility of the proposed idea. The developed learning procedure provides competitive results with the existing VNS solution of DOP in problems with relatively sparse target locations, but it provides worse results in instances of the Set 64 and Set 66 problem sets where the target locations are placed in a grid. It is because of relatively simple local rules for removing neurons and the visited locations during the learning to meet the requirements on the travel budget. On the other hand, the SOM-based approach is less computationally demanding than the VNS albeit Dubins maneuvers are repeatedly computed during the learning. The presented results support the proposed approach is viable, and the further research aims to address the identified drawbacks of the SOM-based approach by a more sophisticated structure to speed up the computation of Dubins tour represented by the ring. Besides, a combination of several local improvement strategies and combination with the memetic algorithm are subjects of the future work.

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

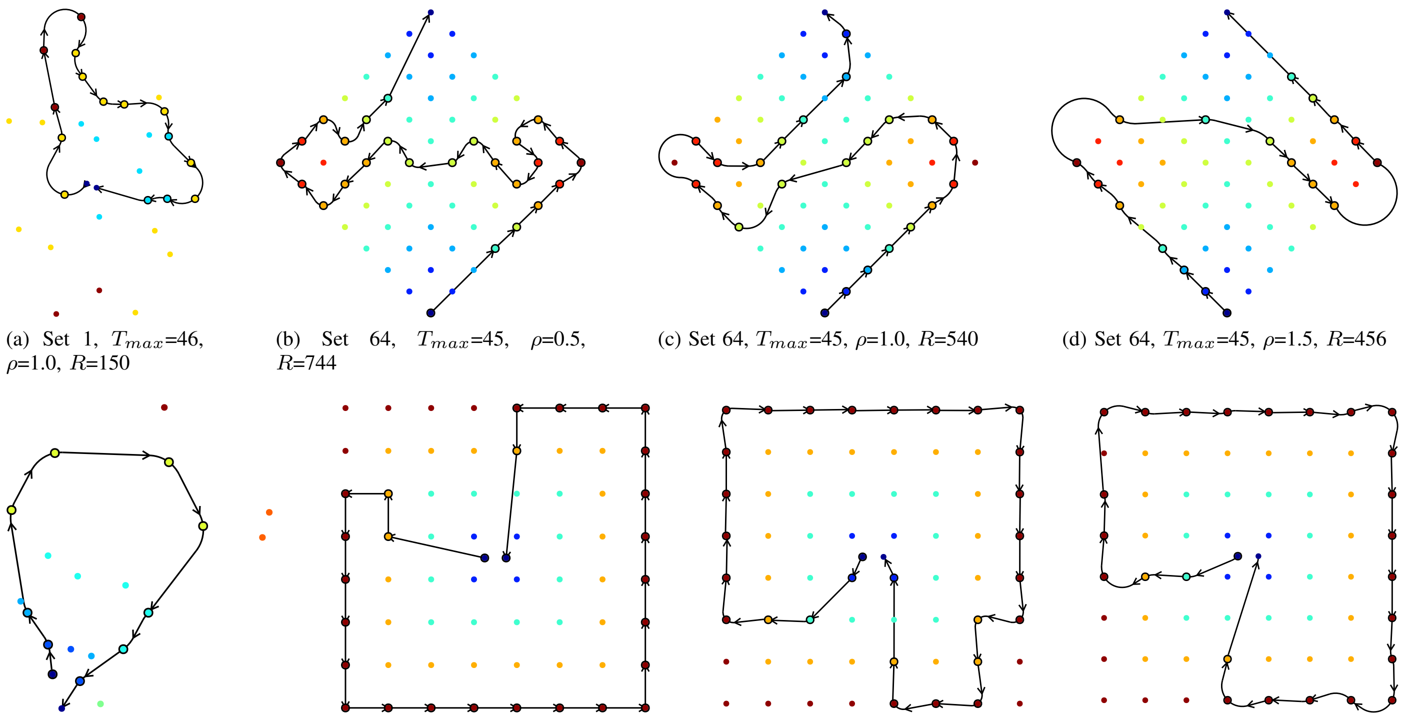
References

[1] K. Vicencio, B. Davis, and I. Gentilini, "Multi-goal path planning based on the generalized traveling salesman problem with neighborhoods," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 2985–2990.

[2] D. Bhadauria, O. Tekdas, and V. Isler, "Robotic data mules for collecting data over sparse sensor fields," *Journal of Field Robotics*, vol. 28, no. 3, pp. 388–404, 2011.

[3] J. Faigl and G. A. Hollinger, "Autonomous data collection using a self-organizing map," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–13, 2017.

[4] B. L. Golden, L. Levy, and R. Vohra, "The orienteering problem," *Naval Research Logistics (NRL)*, vol. 34, no. 3, pp. 307–318, 1987.



(a) Set 1, T_{max}=46, ρ=1.0, R=150
(b) Set 64, T_{max}=45, ρ=0.5, R=744
(c) Set 64, T_{max}=45, ρ=1.0, R=540
(d) Set 64, T_{max}=45, ρ=1.5, R=456
(e) Set 2, T_{max}=30, ρ=1.5, R=195
(f) Set 66, T_{max}=66, ρ=0.0, R=880
(g) Set 66, T_{max}=60, ρ=0.5, R=860
(h) Set 66, T_{max}=60, ρ=1.0, R=835

Fig. 10. Select solutions of the Dubins Orienteering Problem (DOP) found by the proposed SOM-based approach. The colored disks represent the sensor locations, and the color denotes the value of the reward (high rewards are in the red, and low rewards are in blue).

[5] R. Pěnička, J. Faigl, P. Váňa, and M. Saska, "Dubins orienteering problem," *Robotics and Automation Letters*, 2017, (accepted, to appear).

[6] D. Feillet, P. Dejax, and M. Gendreau, "Traveling salesman problems with profits," *Transportation Science*, vol. 39, no. 2, pp. 188–205, 2005.

[7] L. E. Dubins, "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.

[8] J. Le Ny, E. Feron, and E. Frazzoli, "On the Dubins Traveling Salesman Problem," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 265–270, 2012.

[9] K. Savla, E. Frazzoli, and F. Bullo, "On the point-to-point and traveling salesperson problems for Dubins' vehicle," in *Proceedings of the American Control Conference*. IEEE, 2005, pp. 786–791.

[10] P. Váňa and J. Faigl, "On the dubins traveling salesman problem with neighborhoods," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 4029–4034.

[11] K. J. Obermeyer, P. Oberlin, and S. Darbha, "Sampling-based path planning for a visual reconnaissance unmanned air vehicle," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 2, pp. 619–631, 2012.

[12] D. G. Macharet, J. W. Monteiro, G. R. Mateus, and M. F. Campos, "Bi-objective data gathering path planning for vehicles with bounded curvature," *Computers & Operations Research*, 2016.

[13] J. Faigl and P. Váňa, "Self-organizing map for the curvature-constrained traveling salesman problem," in *International Conference on Artificial Neural Networks (ICANN)*, 2016, pp. 497–505.

[14] ——, "Unsupervised learning for surveillance planning with team of aerial vehicles," in *International Joint Conference on Neural Networks (IJCNN)*, 2017, (accepted, to appear).

[15] P. Hansen and N. Mladenović, "Variable neighborhood search: Principles and applications," *European Journal of Operational Research*, vol. 130, no. 3, pp. 449–467, 2001.

[16] J. Faigl, R. Pěnička, and G. Best, "Self-organizing map-based solution for the orienteering problem with neighborhoods," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2016, pp. 1315–1321.

[17] J. Faigl, "On self-organizing maps for orienteering problems," in *International Joint Conference on Neural Networks (IJCNN)*, 2017, (accepted, to appear).

[18] R. Ramesh and K. M. Brown, "An efficient four-phase heuristic for the generalized orienteering problem," *Computers & Operations Research*, vol. 18, no. 2, pp. 151–165, 1991.

[19] G. Best, J. Faigl, and R. Fitch, "Multi-robot path planning for budgeted active perception with self-organising maps," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 3164–3171.

[20] J. Faigl and G. Hollinger, "Self-organizing map for the prize-collecting traveling salesman problem," *Advances in Intelligent Systems and Computing*, vol. 295, pp. 281–291, 2014.

[21] B. Angéniol, G. de la C. Vaubois, and J.-Y. L. Texier., "Self-organizing feature maps and the travelling salesman problem," *Neural Networks*, vol. 1, pp. 289–293, 1988.

[22] J. Faigl and G. Hollinger, "Unifying multi-goal path planning for autonomous data collection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 2937–2942.

[23] A. Gunawan, H. C. Lau, and P. Vansteenwegen, "Orienteering Problem: A survey of recent variants, solution approaches and applications," *European Journal of Operational Research*, vol. 255, no. 2, pp. 315–332, 2016.

[24] "The Orienteering Problem: Test Instances – Department of Mechanical Engineering," citied on 2017-02-26. [Online]. Available: http://www.mech.kuleuven.be/en/cib/op#section-0

[25] T. Tsiligirides, "Heuristic methods applied to orienteering," *The Journal of the Operational Research Society*, vol. 35, no. 9, pp. 797–809, 1984.

[26] I.-M. Chao, B. L. Golden, and E. A. Wasil, "A fast and effective heuristic for the orienteering problem," *European Journal of Operational Research*, vol. 88, no. 3, pp. 475–489, 1996.

[27] J.-C. Créput and A. Koukam, "A memetic neural network for the Euclidean traveling salesman problem," *Neurocomputing*, vol. 72, no. 4-6, pp. 1250–1264, 2009.