On Close Enough Orienteering Problem with Dubins Vehicle

Jan Faigl Robert Pěnička

***Abstract*— In this paper, we address a generalization of the Orienteering Problem (OP) for curvature-constrained vehicles and to problems where it is allowed to collect a reward associated to each target location within a specified distance from the target. The addressed problem combines challenges of the combinatorial optimization of the OP (to select the most rewarding targets and find the optimal sequence to visit them) with the continuous optimization related to the determination of the waypoint locations and suitable headings at the waypoints for the considered Dubins vehicle such that the curvature-constrained path does not exceed the given travel budget and the sum of the collected rewards is maximized. The proposed generalization is called the Close Enough Dubins Orienteering Problem (CEDOP) and novel unsupervised learning approach is proposed to address computational requirements of this challenging planning problem. Based on the presented results, the proposed approach is feasible and provides a bit worse solution of CEDOP than the existing combinatorial approach but with significantly lower computational requirements.**

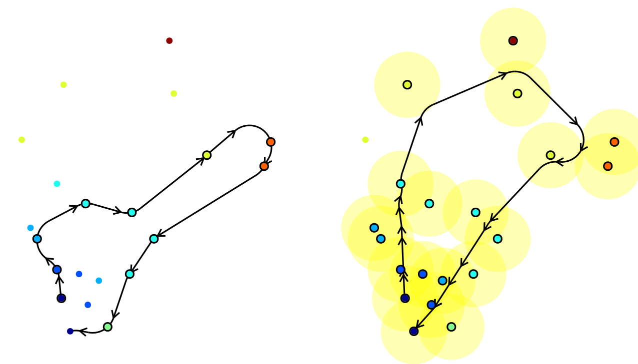
I. Introduction

The herein studied problem is motivated by surveillance [1] and data collection missions [2] in which a robotic vehicle is requested to take a snapshot of the given target locations or to remotely retrieve data from a sensor network using a specified communication range. The proposed approach is focused on Unmanned Aerial Vehicles (UAVs) with kinematic constraints that can be modeled by Dubins vehicle [3]. Besides, we aim to respect the limited operational time of UAVs, and thus we focus on the problem formulation that fits the requirements on the curvature-constrained data collection paths for UAVs and which allows to exploit a non-zero distance for collecting the rewards from the targets.

The addressed problem as a suitable problem formulation for data collection missions with curvature-constrained vehicles is called the Close Enough Dubins Orienteering Problem (CEDOP). It combines the Orienteering Problem (OP) and Traveling Salesman Problem (TSP) both considered with Dubins vehicle. Moreover, the problem is generalized to save unnecessary travels to the target locations by retrieving data within a specified communication radius, and thus the problem includes determination of the suitable waypoint locations in the neighborhood of each of the selected target location. An example of the expected benefit of the CEDOP formulation is demonstrated in Fig. 1, where the saved travel cost by collecting rewards at the non zero distance from the targets allows to collect more rewards.

Authors are with Faculty of Electrical Engineering, Czech Technical University in Prague, {faiglj|penicrob}@fel.cvut.cz

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



(a) Solution of Dubins OP, $R = 240$ (b) Solution of CEDOP, $R = 420$

Fig. 1. An example of solutions for the same configuration of the target locations with Dubins vehicle with the minimum turning radius $\rho = 1$ without (on left) and with the allowed remote data collection within the communication radius $\delta = 1.5$. The travel budget is limited to $T_{max} = 30$ and the sum of the collected rewards is denoted by R. The black curve is the found data collection Dubins tour and the target locations are shown as the small disks with the color denoting the associated reward (high rewards are in red and lower rewards are in blue). The yellow disks represent the communication radius δ at which data can be collected from the particular sensor. A solution of the Close Enough Dubins Orienteering Problem (CEDOP) with $\delta = 1.5$ increases the sum of the collected rewards for the same travel budget and data are retrieved from almost all sensors.

To the best of the authors knowledge, the only existing approach to solve CEDOP has been proposed recently in [4]. The solution is based on the combinatorial metaheuristic the Variable Neighborhood Search (VNS) [5] that has been previously applied to the Dubins Orienteering Problem (DOP) in [6]. The continuous part of the optimization is addressed by sampling possible heading values. In the case of CEDOP, the continuous optimization of finding the suitable waypoint locations around the targets is addressed by an additional sampling of possible waypoint locations. Then, the problem is solved as a pure combinatorial optimization problem [4]. Although the VNS-based approach provides feasible solution, the sampling of the possible waypoint locations and headings becomes quickly computationally demanding and the reported computational times are in tens of minutes.

In the present work, we aim to address the computational complexity of CEDOP by recent advancements on Self-Organizing Maps (SOMs) for the OP [7] and DTSP [8] and their generalization for neighborhoods proposed in [9] and [10], respectively. The proposed approach is directly leveraging on the existing SOM-based solution of the OP that is combined with the principles employed in solving the DTSP. Although an application of SOM to combinatorial optimization problems may be questionable, the main expected benefit of the proposed unsupervised learning is in solving the continuous optimization part of CEDOP, where it is needed to determine the most suitable headings of the vehicle at the waypoints and the waypoint locations themselves. The reported results indicate the proposed SOM approach is viable and has significantly lower computational requirements than the VNS, albeit the found solutions are bit worse than the computationally demanding VNS for CEDOP. Notice, the herein addressed problem is called CEDOP to emphasize the disk shaped neighborhood utilized in the proposed SOM approach, rather than the Dubins Orienteering Problem with Neighborhoods (DOPN) as in [4].

The rest of the paper is organized as follows. A brief overview of the related work is presented in the next section. The addressed problem is formally introduced in Section III. An overview of the existing VNS-based approach for DOP [6] and DOPN [4] is presented in Section IV. The proposed SOM-based approach for CEDOP is presented in Section V. The empirical evaluation of the existing CEDOP solvers is reported in Section VI. Concluding remarks and future work are in Section VII.

II. Related Work

The problem of finding a cost efficient curvature-constrained path to visit a set of target locations can be formulated as the Dubins Traveling Salesman Problem (DTSP) [11]. The DTSP stands to determine the optimal sequence of visits to the target locations as in a regular TSP, but in addition, it also contains continuous optimization to find the optimal heading of the vehicle at each location. Finding headings is a challenging problem because each heading value can be arbitrarily selected from the interval $[0, 2\pi)$ and every change of a single heading may greatly affect the length of Dubins tour connecting the waypoints.

If it is allowed to reach a close vicinity of each target location, the Dubins Traveling Salesman Problem with Neighborhoods (DTSPN) [12] can be used to find a cost efficient path. In the DTSPN, in addition to the DTSP, the solution includes a determination of suitable waypoint locations. Due to the computational complexity of these problems, approximation algorithms [13], heuristics [14], [15], [16], sampling-based [17] and evolutionary approaches [18], [19] have been proposed. However, the operational time of UAVs is often limited, and therefore, finding a cost efficient path to visit all locations may not been enough to respect limited travel budget of real UAVs. Therefore, we can explicitly restrict the travel budget to T_{max} and ask for finding a path not exceeding T_{max} such that the vehicle will visit the most important target locations while safely returns to the desired final location, e.g., for refuel or battery replacement.

To respect the limited operational time, the problem can be formulated as the Orienteering Problem (OP) [20]. For a set of target locations S in a plane $S \subset \mathbb{R}^2$, each location with the associated reward, and specified starting and ending locations of the vehicle, the problem is to determine a tour to visit a subset of the locations $S_k \subseteq S$ such that the sum of the collected rewards of S_k is maximized and the tour length is shorter than the specified travel budget T_{max} [21].

Although several approaches for the OP have been proposed [22], [23], two important extensions are considered in the presented work. The first is related to the allowed reward collection within specified distance from the target, which may greatly reduce the total tour length [24]. For the TSP, such a problem is known as the TSP with Neighborhoods (TSPN) or specifically with the disk-shaped neighborhood as the Close Enough TSP (CE-TSP) [25]. Even though many approaches and approximation algorithms for the TSPN (including prior work based on SOM [26], [27]) have been proposed, according to the literature about the OP, none of the approaches has been employed in solving the OP with Neighborhoods (OPN) and probably the only approach already deployed to the OPN is based on SOM.

In SOM based solutions of the OPN [28], [7], the target locations with high rewards are preferred by a duplication of such targets and repeated adaptation of the network to such high rewarding targets. Unfortunately it increases the computational burden, which has been addressed by a new adaptation procedure proposed in [9]. The reported results show that competitive solutions to the previous SOM-based approaches are found with significantly less computational requirements and solutions are provided in less than 100 ms.

The second extension is related to the requirements on the curvature-constrained path of Dubins vehicle. Even though several approaches have been proposed for the DTSP and DTSPN, the OP for Dubins vehicles as the Dubins Orienteering Problem (DOP) has not been addressed up to the recently proposed [6]. The motivation and importance of the direct solution of the OP is advocated in [6], and thus the reader is referred to [6] for comments and supportive results.

A generalization of the VNS-based approach for DOP [6] to the DOP with Neighborhoods (DOPN) has been proposed in [4]. The extension is based on sampling a discrete set of possible locations around each target and the problem is solved in very similar way as the original solution of DOP in [6]. However, it is necessary to consider possible heading values for each such a sampled possible waypoint location, and thus the approach is computationally demanding because the size of the problem grows with the number of samples that are the headings values times possible waypoint locations sampled in the neighborhood of each target location.

III. Problem Statement

The addressed Close Enough Dubins Orienteering Problem (CEDOP) is a combination of the combinatorial OP with the continuous optimization of headings at the waypoints together with determination of the waypoint locations themselves. The problem is to determine a cost efficient curvature-constrained path to retrieve the most valuable measurements from a set of locations $S \subset \mathbb{R}^2$ and the total tour length does not exceed the given travel budget T_{max}. Each location $s_i \in S$ has associated reward r_i, such that $r_i \in R$ and $r \geq 0$, that can be remotely collected by the vehicle within the distance δ from the individual target location. Regarding the existing formulations of the OP [29], the starting and final locations of the vehicle are prescribed as s_1 and s_n

with the zero rewards $r_1 = r_n = 0$, where n is the total number of locations S, $n = |S|$.

The requested maximal rewarding data collection path has to satisfy the kinematic constraints of Dubins vehicle which is moving with a constant forward velocity v and its minimal turning radius is ρ [3]. The state of the 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 shortest curvature-constrained path connecting two states $q_i, q_j \in SE(2)$ can be computed analytically [3] and it is one of six possible Dubins maneuvers consisting of straight line segment and arcs with the curvature ρ. However, in CEDOP, we first need to determine the respective locations of such waypoints $p_i, p_j \in \mathbb{R}^2$ and also the particular headings $\theta_i, \theta_j \in \mathbb{S}^1$ of the vehicle at the waypoints, where $q_i = (p_i, \theta_i)$. Moreover, it may not be possible to retrieve data from all the locations S within the travel budget T_{max}, and therefore, a subset S_k of k locations $S_k \subseteq S$ has to be selected to collect the associated rewards 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 $1 \le \sigma_i \le n$ and $q_{\sigma_1} = (s_1, \theta_1)$ and $q_{\sigma_k} = (s_n, \theta_n)$, such that each waypoint $q_{\sigma_i} = (p_{\sigma_i}, \theta_i)$ consists of the location $p_{\sigma_i} \in \mathbb{R}^2$ at which data from the corresponding location s_{σ_i} can be retrieved, i.e., $|(p_{\sigma_i}, s_{\sigma_i})| \le \delta$ and θ_{σ_i} is the suitable heading at p_{σ_i} with respect to Dubins maneuvers connecting the waypoints. Having the k locations S_k and the determined waypoint locations, 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 k locations S_k, the particular suitable waypoints locations $P_k = (p_{\sigma_1}, \ldots, p_{\sigma_k})$, the permutation of the visits $\Sigma = (\sigma_1, \ldots, \sigma_k)$ to the waypoints and the respecting headings $\Theta = (\theta_{\sigma_1}, \ldots, \theta_{\sigma_k})$. The Close Enough Dubins Orienteering Problem (CEDOP) can be formulated as the optimization problem:

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

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}.

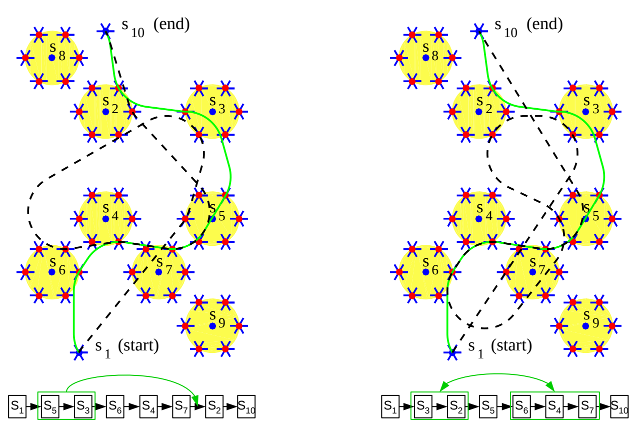
IV. Overview of the VNS-based Solver for DOPN

The solver for the DOP(N) proposed in [6], [4] builds on the VNS metaheuristic [5] in which the continuous optimization problem of the DOPN is addressed by sampling the neighborhood of each target location $s \in S$ by the o samples. Moreover, for each such a sample, h heading values are proportionally sampled in the interval $[0, 2\pi)$. Particular Dubins maneuvers connecting all such possible waypoints are precomputed and the problem is then solved by the VNS-based combinatorial optimization heuristic.

The VNS defines l predefined neighborhood structures $N_l, l = 1, \ldots, l_{max}$ and iteratively improves the actual best found solution inside *shake* and *local search* procedures. Let the sum of the collected rewards by the path P be $R(P)$ and the length of P be $\mathcal{L}(P)$. The overall structure of the algorithm is summarized in Fig. 2 and an example of shaking operators is visualized in Fig. 3. Due to the limited space, the reader is referred to [6] and [4] for further details.

function VNS METHOD FOR CEDOP(S, ρ, δ, o, h)
  $S_r \leftarrow$ getReachableLocations(S)
  $P \leftarrow$ createInitialPath(S_r,T_{max})
  while Stopping condition is not met **do**
    $l \leftarrow 1$
    while ($l \le l_{max}$) **do**
      $P' \leftarrow$ shake(P, l)
      $P'' \leftarrow$ localSearch(P', l)
      if $\mathcal{L}_d(P'') \le T_{max}$ **and** $R(P'') > R(P)$ **then**
        $P \leftarrow P''$; $l \leftarrow 1$
      else
        $l \leftarrow l + 1$
  return (P) // return the found path
end function

Fig. 2. A summary of the VNS-based algorithm for DOPN with o samples per each target location and further h heading values per each such a sample



(a) **Path Move** (b) **Path Exchange**

Fig. 3. Visualization of the shaking neighborhood operators with $o = 6$ samples of the neighborhood of each target with $h = 6$ heading values per each waypoint location are considered. The original paths (dashed and black) are changed within the neighborhood to new and shorter paths shown in green. Further details can be found in [4] and [6].

V. Proposed Solution of the Close Enough Dubins Orienteering Problem

The proposed unsupervised learning to solve CEDOP is directly based on the previous application of the SOM principles in the OPN proposed in [28], [7], [9] and SOM for the DTSP introduced in [8] and later generalized for the DTSPN in [10]. Even though the proposed method uses the same principles, and thus the description can be referred to the previous work, there are two important modifications needed for solving CEDOP to respect the limited budget of Dubins vehicle. Therefore an overview of the whole adaptation procedure is presented together with the highlighted modifications to make the paper more self-contained.

The SOM for CEDOP is a growing two-layered neural network with the same internal structure as the SOM for PC-TSPN [27]. The input layer serves for presenting the locations $s_i \in S$ and the output layer consists of neurons $\mathcal{N} = \{\nu_1, \ldots, \nu_m\}$ organized into an array of output units that forms a ring of nodes (neuron weights). The unsupervised learning of SOM is an iterative procedure in which locations are presented to the network and the most suitable neuron is determined and possibly selected as the winner neuron that is then adapted together with its neighboring neurons towards the presented input location. The important difference of solving OP in comparison to the TSP is that the adaptation is performed only if the ring of the winner neurons (after the adaptation) would represent a tour that does not exceed the travel budget T_{max} [7], [9]. Therefore, the network may or may not be adapted to the particular presented input.

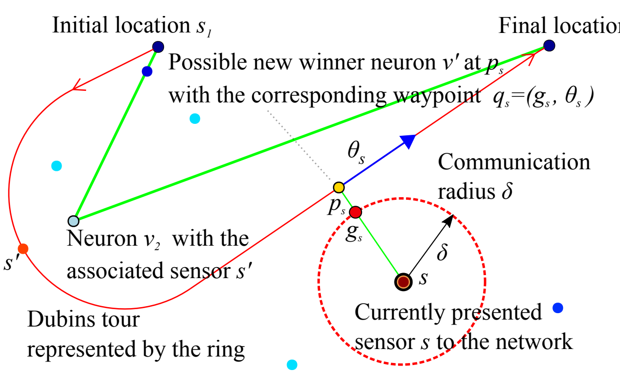


Fig. 4. Proposed selection of the possible winner neuron together with the expected heading at the particular waypoint. The green straight line segments denote connected ring of neurons. Each neuron ν has associated location $s \in S$ for which the neuron has been selected as the winner neuron, a particular heading θ, and a set of the supporting headings Θ_ν. The showed ring consists of the first neuron at the initial location s_1, the neuron ν_2 with the associated the left most location visualized as the red small disk, and the last neuron corresponding to the final location s_n. The location 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 (g_s, θ_s), where the heading θ_s is the heading on the red Dubins tour at the point p_s and g_s is the waypoint location at which data from s can be retrieved within the δ communication radius.

For solving CEDOP, each neuron $\nu_i \in \mathcal{N}$ has associated the neuron weights representing the location of the neuron in the input space $\mathbb{R}^2$ as for the TSP. In addition, each ν_i has also associated a particular target location $s \in S$ for which ν_i has been selected as the winner, an expected waypoint location g_s at which data from s can be retrieved within the δ communication radius, and an expected vehicle heading θ_s at the waypoint g_s that are all determined in the proposed winner selection depicted in Fig. 4. Besides, each ν_i has up to h additional heading values $\Theta_{\nu_i} = \{\theta_1, \ldots, \theta_h\}$ around θ_s to further support finding a short Dubins tour connecting the determined waypoints associated to the neurons.

The first important modifications of the existing SOM for the OP [9] is that the possible winner is determined not according to the connected neurons in the ring, but according to Dubins tour connecting the associated waypoints to the winner neurons, see Fig. 4. Such Dubins tour represented by the ring is considered as the current solution of CEDOP that has to satisfy the budget constraint T_{max}. Dubins tour is sensitive to the vehicle heading at the waypoints, and therefore, additional headings Θ_{ν_i} are considered in finding Dubins tour represented by the ring by the same forward search procedure as in solving the DTSP, see Fig. 3 in [10].

The second modification is that contrary to [9], where neurons weights directly represent the waypoints, for Dubins tour, we need to associate the waypoint locations with the particular headings. Therefore, to determine if the ring would represent a tour satisfying T_{max} after the adaptation, the adaptation has to be performed and in the case of the budget violation, the network has to be reverted to the state before the adaptation. The proposed unsupervised learning for CEDOP is summarized in Fig. 5.

A. *Computational Complexity*

Complexity of the proposed learning procedure depends on the number of locations n and the number of supporting headings h used in finding the shortest Dubins tour to visit the particular waypoints. The number of neurons m never exceeds the number of locations, i.e., $m \leq n$, because neurons are removed during the adaptation and after each learning epoch in Step 6(d)ii and Step 7 in Fig. 5, respectively. All locations are presented to the network in each learning epoch and for each $s \in S$ a possible winner neuron is determined using Dubins tour represented by the current ring of neurons that is found in $O(nh^2)$ for the worst case of n neurons in the ring. Due to the conditional adaptation to fit the budget constraint, Dubins tour is determined two times, Step 6c and Step 6(d)i in Fig. 5. 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 $O(3n^3h^2)$. Notice, SOM for the Euclidean OP [9] has complexity $O(n^2)$ because of the direct usage of Euclidean distance.

The learning does not depend on the radius δ because the expected waypoint g_s is determined during the winner selection. This is in a direct contrast to the VNS-based approach [4] in which the size of the problem grows with the number of sampled headings and possible waypoint locations. However, the proposed SOM-based approach repeatedly determines Dubins tour for each single winner selection, which is avoided in [4] by the precomputed Dubins tours. The real computational requirements and influence of the number of additional headings h to the required computational time and solution quality have been empirically

▷ **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 the learning.
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 locations $S = \{s_1, \ldots, s_n\}$ except s_1 and s_n to avoid local optima; $\Pi \leftarrow \mathrm{permute}(S \setminus \{s_1, s_n\})$.
6) For each $s \in \Pi$:

▷ **Conditional Adapt:**

   a) Save the current network $\mathcal{N}' \leftarrow \mathcal{N}$.
   b) $(\nu', p_s, \theta_s, g_s) \leftarrow$ *determine a possible winner neuron* ν' for s at the location p_s with the heading θ_s and the waypoint location at g_s regarding the current ring $\mathcal{N}$, the minimal turning radius ρ, and the communication radius δ, see Fig. 4.
   c) $\mathcal{N} \leftarrow \mathrm{adapt}(\mathcal{N}, \nu', \theta_s, g_s)$ – adapt the network $\mathcal{N}$ with the new winner ν' towards g_s [10] and determine the length of Dubins tour represented by the ring as $\mathcal{L}(\mathcal{N})$.
   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 location $g_{s_{\nu_f}}$ and the neuron with the associated location with the lowest reward, respectively [9].
      ii) $\mathcal{N} \leftarrow \mathrm{adapt}(\mathcal{N} \setminus \{\nu_f, \nu_l\}, \nu', \theta_s, g_s)$ – 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 a 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 at Step 5; `Otherwise`: `Stop` the learning.

Fig. 5. A summary of the proposed SOM-based adaptation procedure for the Close Enough Dubins Orienteering Problem (CEDOP)

evaluated in a set of selected instances of the DOP and CEDOP. The achieved results are reported in Section VI.

VI. RESULTS

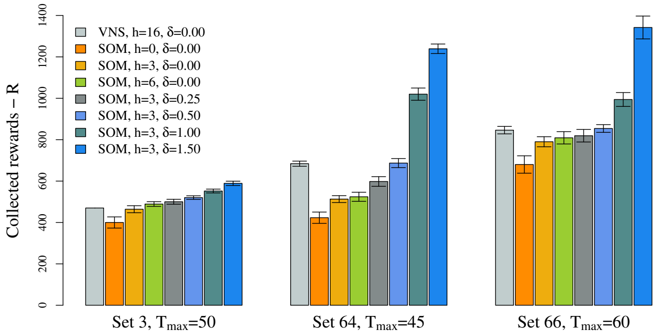
The proposed SOM-based solution of CEDOP has been evaluated using selected problem instances of the existing standard benchmarks for the OP [23], [30] already utilized in the first introduction of DOP in [6]. The only known existing DOP and CEDOP solvers have been proposed in [6] and [4], respectively, and therefore, the proposed SOM-based approach is compared with the VNS-based solver, which is further referenced as the VNS in the rest of the paper.

The standard benchmarks for the OP consist of a set of problems with the particular number of locations n, each with the specified rewards accompanied with the particular travel budget T_{max} [30]. The benchmarks evaluated in [6], [4] are Set 1 ($n = 32$), Set 2 ($n = 21$), and Set 3 ($n = 33$) [31] and diamond-shaped Set 64 ($n = 64$) and squared-shaped Set 66 ($n = 66$) [32] with the 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., $\rho \in \{0, 0.5, 1.0, 1.5\}$, which can give 356 instances of DOP. Moreover, for CEDOP with the communication radius $\delta > 0$, additional problem instances can be defined. Due to such an excessive number of instances, limited space, and novelty of the introduced CEDOP, it is rather preferred to focus the evaluation on the influence of ρ and δ to the performance of the existing solvers. Therefore, selected problem instances with specific budget in the middle of the range of budgets are used first.

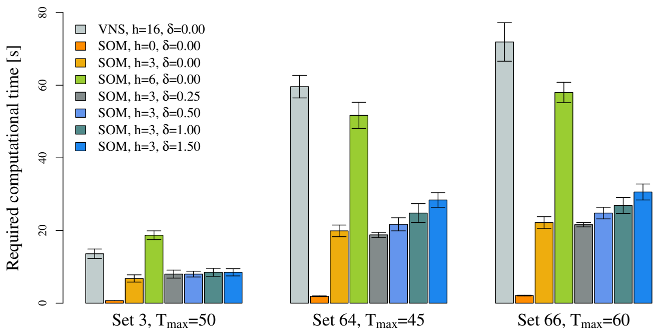
The proposed SOM algorithm is stochastic, therefore each problem instance is solved 20 times and the reported results are average values accompanied with the respective standard deviations, i.e., the average sum of the collected rewards R and the average required computational time T_{cpu}. Besides, the algorithm performance is evaluated using the relative percentage error (RPE) defined as the relative error between the reference value R_{ref} and the best found solution R_{best} over the performed trials per particular problem instance, where R_{ref} is the highest reward found by the evaluated algorithms. The RPE is computed as $RPE = (R_{ref} - R_{best})/R_{ref} \cdot 100$. In addition, the robustness of the algorithm over the performed trials is shown as the average relative percentage error (ARPE), i.e., $ARPE = (R_{ref} - R)/R_{ref} \cdot 100$.

The proposed SOM-based algorithm has only two parameters: the number of learning epochs i_{max} and the number of supporting headings h. i_{max} has been experimentally set to $i_{max} = 150$ and h from the set $h \in \{0, 3, 6\}$. The VNS-based approach is also stochastic because of underlying randomized variant of the VNS [6]; however, solutions of CEDOP with $\delta > 0$ are computationally very demanding [4], therefore the reported results are computed from 10 trials. The parameters of the VNS algorithm follows [6], [4] and for DOP instances, the number of heading values per each target is $h = 16$ and the number of neighborhood structures $l_{max} = 2$. The used stopping criterion is the maximal number of 10,000 iterations with the maximal 3,000 iterations without improvement. For CEDOP instances with $\delta > 0$, the number of samples per each neighborhood o is set to $o = 16$ and the stopping criterion is a combination of the maximal 10,000 iterations and 5,000 iterations without improvement.

Both evaluated algorithms are implemented in C++ and run within the same computational environment using a single core of the iCore7 CPU running at 4 GHz unless



(a) Average sum of the collected rewards



(b) Average computational time

Fig. 6. Performance of the VNS [6] and the proposed SOM according to the number of the headings h and the radius δ in problems with $\rho = 1$

stated otherwise. The VNS algorithm pre-computes all the Dubins maneuvers between all the targets and particular headings prior the combinatorial optimization with an optimized structure for inserting/removing waypoint into Dubins tour. The SOM-based approach searches for the suitable headings during the unsupervised learning and maneuvers are computed on demand and for the presented results no special optimizations in Dubins tour computations are utilized.

First, the proposed algorithm has been evaluated on the selected instances of the Set 3, Set 64, and Set 66 with $\rho = 1$, $h \in \{0, 3, 6\}$, and $\delta \in [0.0, 1.5]$. The average values with the standard deviations showed as error bars are visualized in Fig. 6 together with the respective required computational times. The results clearly show that increasing h improves the solution at the cost of the increased computational burden. A suitable choice seems to be relatively low $h = 3$ which does not increase the computational burden significantly and provides similar solutions as $h = 6$.

Regarding the influence of the communication radius δ, a longer radius saves the travel cost and more rewards can be collected for the same travel budgets. Notice that for a relatively negligible $\delta = 0.25$, SOM provides solutions competitive to the solution of DOP provided by the VNS. On the other hand, increasing ρ the computational burden is decreasing because of exactly opposite effect as higher minimal turning radius ρ limits the maneuverability of the vehicle, and therefore, fewer rewards are collected.

Both CEDOP solvers have been further evaluated on the benchmark instances of the Set 3, Set 64, and Set 66 for the whole range of the travel budget, $\rho = 1.0$, and $\delta \in \{0.0, 0.5, 1.0\}$. The results in Table I are reported

TABLE I

AGGREGATED RESULTS FOR SET 3, SET 64, AND SET 66 WITH $\rho = 1.0$

Problem set	VNS [4]		Proposed SOM ($h = 3$)		
	$\overline{\text{ARPE}}$	T_{cpu}* [s]	$\overline{\text{RPE}}$	$\overline{\text{ARPE}}$	T_{cpu} [s]
Set 3, $\delta = 0.0$	1.0	1,178.86	3.6	7.4	7.0
Set 3, $\delta = 0.5$	0.9	13,273.31	6.6	10.6	7.9
Set 3, $\delta = 1.0$	0.5	13,304.44	5.5	9.2	8.3
Set 64, $\delta = 0.0$	1.9	5,272.22	17.4	23.8	17.9
Set 64, $\delta = 0.5$	2.8	13,595.56	18.7	24.2	20.2
Set 64, $\delta = 1.0$	1.3	13,792.25	9.9	15.2	22.2
Set 66, $\delta = 0.0$	1.5	6,546.63	3.6	9.1	22.9
Set 66, $\delta = 0.5$	1.4	13,650.09	6.7	11.8	25.5
Set 66, $\delta = 1.0$	3.2	13,824.48	16.1	21.3	26.7

*Due to computational requirements of the VNS, the results have been obtained with a grid of Xeon CPUs running at 2.2 GHz to 3.4 GHz.

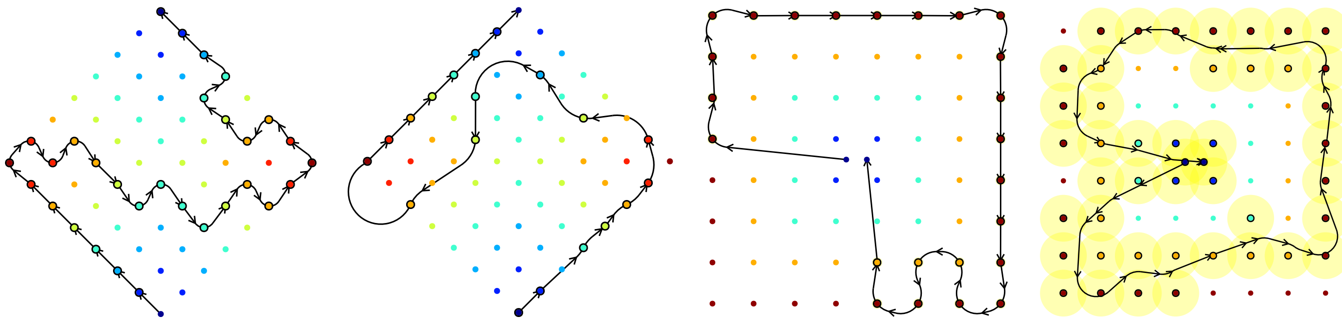
as aggregated values of the respective indicators RPE and ARPE among all instances in the particular benchmark sets that are denoted by $\overline{\text{RPE}}$ and $\overline{\text{ARPE}}$, respectively. Selected solutions found by the SOM algorithm are shown in Fig. 7.

A. Discussion

The presented results support feasibility of the proposed SOM-based approach to solve computationally challenging instances of CEDOP. The SOM approach samples the vehicle headings during the unsupervised learning, and thus it needs few samples of supporting headings h. 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), the SOM-based approach is less computationally demanding at the cost of worse solutions than the purely combinatorial optimization based on the VNS [4]. 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.

VII. CONCLUSION

In this paper, a novel SOM-based adaptation procedure is proposed to solve a variant of the Dubins Orienteering Problem (DOP) with disk shaped neighborhoods called the Close Enough Dubins Orienteering Problem (CEDOP). The proposed approach is based on the previous SOM-based adaptation procedures for the Euclidean OPN and DTSPN. The presented results support feasibility of the proposed generalization of the existing approaches. The SOM adaptation procedure provides competitive results to the VNS-based solution of DOP in problems with relatively sparse target locations such as Set 3, 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 mainly because of relatively simple local rules for removing neurons and the visited targets during the learning to meet the requirements on the travel budget. On the other hand, the SOM approach is less computationally demanding than the VNS albeit Dubins maneuvers are repeatedly computed during the learning.



(a) Set 64, T_{max}=45, δ=0.0, ρ=1.0, R=732

(b) Set 64, T_{max}=45, δ=0.0, ρ=1.5, R=426

(c) Set 66, T_{max}=66, δ=0.25, ρ=1.0, R=810

(d) Set 66, T_{max}=66, δ = 1.25, ρ=1.0, R=1185

Fig. 7. Selected solutions of the Dubins Orienteering Problem (DOP) and Close Enough Dubins Orienteering Problem (CEDOP) found by the proposed SOM-based approach. The colored disks represent the locations and the color denotes the value of the reward (high rewards are in red and low rewards are in blue). The yellow disks represent the communication radius within which the reward can be collected from the respective target location.

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 speedup computation of Dubins tour represented by the ring. Moreover, a combination of the local improvement strategies, tabu search techniques, and possibly the ideas of shaking procedures of the VNS can be considered to improve solutions provide by SOM, and thus combine benefits of both solvers for CEDOP. These ideas are subject for a future work.

REFERENCES

[1] K. Vicencio, B. Davis, and I. Gentilini, "Multi-goal path planning based on the generalized traveling salesman problem with neighborhoods," in *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] 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.

[4] R. Pěnička, J. Faigl, P. Váňa, and M. Saska, "Dubins orienteering problem with neighborhoods," in *ICUAS*, 2017, pp. 1555–1562.

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

[6] R. Pěnička, J. Faigl, P. Váňa, and M. Saska, "Dubins orienteering problem," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1210–1217, 2017.

[7] J. Faigl, R. Pěnička, and G. Best, "Self-organizing map-based solution for the orienteering problem with neighborhoods," in *IEEE SMC*, 2016, pp. 1315–1321.

[8] J. Faigl and P. Váňa, "Self-organizing map for the curvature-constrained traveling salesman problem," in *ICANN*, 2016, pp. 497–505.

[9] J. Faigl, "On self-organizing maps for orienteering problems," in *IJCNN*, 2017, pp. 2611–2620.

[10] J. Faigl and P. Váňa, "Unsupervised learning for surveillance planning with team of aerial vehicles," in *IJCNN*, 2017, pp. 4340–4347.

[11] 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.

[12] J. T. Isaacs and J. P. Hespanha, "Dubins traveling salesman problem with neighborhoods: a graph-based approach," *Algorithms*, vol. 6, no. 1, pp. 84–99, 2013.

[13] P. Oberlin, S. Rathinam, and S. Darbha, "Today's traveling salesman problem," *Robotics & Automation Magazine, IEEE*, vol. 17, no. 4, pp. 70–77, 2010.

[14] 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.

[15] P. Váňa and J. Faigl, "On the dubins traveling salesman problem with neighborhoods," in *IROS*, 2015, pp. 4029–4034.

[16] P. Isaiah and T. Shima, "Motion planning algorithms for the Dubins Travelling Salesperson Problem," *Automatica*, vol. 53, pp. 247–255, 2015.

[17] 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.

[18] 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.

[19] X. Zhang, J. Chen, B. Xin, and Z. Peng, "A memetic algorithm for path planning of curvature-constrained uavs performing surveillance of multiple ground targets," *Chinese Journal of Aeronautics*, vol. 27, no. 3, pp. 622–633, 2014.

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

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

[22] P. Vansteenwegen, W. Souffriau, and D. V. Oudheusden, "The orienteering problem: A survey," *European Journal of Operational Research*, vol. 209, no. 1, pp. 1–10, 2011.

[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] Bo Yuan, M. Orlowska, and S. Sadiq, "On the Optimal Robot Routing Problem in Wireless Sensor Networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 9, pp. 1252–1261, 2007.

[25] F. Carrabs, C. Cerrone, R. Cerulli, and M. Gaudioso, "A novel discretization scheme for the close enough traveling salesman problem," *Computers & Operations Research*, vol. 78, pp. 163–171, 2017.

[26] J. Faigl, V. Vonásek, and L. Přeučil, "Visiting Convex Regions in a Polygonal Map," *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1070–1083, 2013.

[27] J. Faigl and G. Hollinger, "Unifying multi-goal path planning for autonomous data collection," in *IROS*, 2014, pp. 2937–2942.

[28] G. Best, J. Faigl, and R. Fitch, "Multi-robot path planning for budgeted active perception with self-organising maps," in *IROS*, 2016, pp. 3164–3171.

[29] 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.

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

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

[32] 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.