On Self-Organizing Maps for Orienteering Problems

Jan Faigl
Czech Technical University, Faculty of Electrical Engineering
Technicka 2, 166 27, Prague, Czech Republic
Email: faiglj@fel.cvut.cz

***Abstract*—This paper concerns principles of unsupervised learning of self-organizing maps (SOMs) to address optimization routing problems called the Orienteering Problem (OP) and its multi-vehicle variant called the Team Orienteering Problem (TOP). The problems are similar to the traveling salesman problem in finding an optimal tour to visit all the given locations, but here, each location has specified reward that can be collected by the tour and the problem is to select the most valuable subset of the locations that can be visited within the travel budget. In existing SOM for the OP, the locations to be visited are duplicated to adapt the network to locations with higher rewards more frequently. The proposed novel SOM-based solution overcomes this necessity and based on the presented results it significantly reduces the computational burden of the adaptation procedure. Besides, the proposed approach improves the quality of solutions and makes SOM competitive to existing heuristics for the OP, but still behind computationally expensive metaheuristics for the TOP. On the other hand, the main benefit of the SOM-based approaches over the existing heuristics is in solving the generalized variant of the OP and TOP with neighborhoods. These variants of the problem formulation allow to better utilize the travel budget for instances where the reward associated with the location can be collected by visiting a particular neighborhood of the location and not exactly the location itself. This generalized problem formulation better models situations of the robotic data collection, e.g., using wireless communication or range sensors.**

I. Introduction

The Orienteering Problem (OP) was introduced in [1] for an inventory routing problem and for orienteering in [2]. The OP can be considered as a variant of the popular combinatorial the Traveling Salesman Problem (TSP) in which we aim to determine a tour that maximizes the total collected profit [3]. Contrary to the variant of the TSP called the Prize-Collecting TSP [4], where the objective function is to determine the shortest tour maximizing the profit, the OP stands to determine a tour that maximizes the collected rewards associated with the given set of locations while the total tour length is shorter than the given travel budget T_{max}. Hence, the OP can be considered as a combination of the Knapsack problem in selecting the most valuable locations to be visited and the TSP in finding the shortest tour visiting such a subset. Therefore, the OP is at least NP-hard [1] as for visiting all the given locations the problem of determining the shortest tour is the TSP.

Motivations for solving the OP are related to routing and logistic tasks in which the travel budget is limited and where there is a demand to serve many potential locations (customers). Several practical applications can be found in [5], [6]. The main motivation of the approach studied in this paper is arising from the robotic information gathering, where a single or team of autonomous vehicles is requested to collect data from a pre-deployed sensor network or collect measurements at particular locations. Not all sensor locations provide data of the same importance, and thus a reward characterizing the importance of the data can be associated with each sensor location. The operational time of robotic vehicles is often limited, and thus the goal is to maximize the collected information from the sensors while the required travel cost is under the given limit.

Having a fleet of vehicles, the total collected rewards can be increased by simultaneous data collection by individual vehicles [7]. Then, the problem is to determine a set of data collection tours, one for each individual vehicle, such that the reward associated with a particular sensor location is counted only for a single vehicle, i.e., multiple visits to the same location do not increase the total reward. Such a formulation of the OP has been introduced as the Team Orienteering Problem (TOP) in [8] and it can also be used in situations where a tour for one vehicle has to be determined for a single day and other important customers are served on the next days [9].

Moreover, it is often not necessary to visit the particular sensor locations exactly in motivational scenarios of data collection for environment monitoring. It may be rather more suitable to save the travel cost by remotely read data from the sensors within a wireless communication range [10] or using range measurements such as a camera in surveillance and inspection missions [11], [12]. This can be addressed by the variant of the TSP called the Traveling Salesman Problem with Neighborhoods (TSPN) [13], [14] and a similar generalization of the OP has been introduced as the Orienteering Problem with Neighborhoods (OPN) in [15], [16]. The main difference between the ordinary OP and its generalization the OPN is that in addition to the determined subset of the sensors providing the most valuable measurements, it is also required to determine the most suitable waypoints from which the measurements (rewards) from the sensors can be collected.

The both formulations (the OP and OPN) share the main challenge of orienteering problems that is the determination of the subsets of the locations according to the tour visiting them with respect to the given travel budget. Regarding the existing approaches for the orienteering problems [5], [6], the

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

only approaches capable of a direct solution of the OPN are those based on self-organizing maps proposed in [16], [15].

In this paper, SOM for the OP is investigated and a novel adaptation method of the SOM-based unsupervised learning is proposed. The proposed growing self-organizing structure is independent on the rewards, and thus it scales better than the previous methods. Moreover, the learning procedure keeps the number of neurons lower or equal to the number of locations to be visited, which further decreases the required computational time. The proposed SOM method also provides a solution of the OPN. In addition, the proposed method is employed to solve the Team Orienteering Problem (TOP) and its variant with neighborhoods. The developed algorithm has been evaluated using standard benchmarks for the OP and TOP [6], [17] and found solutions are compared with existing results available in the literature [18], [9]. Regarding the results for the OP, the proposed SOM-based approach provides better solutions than the previous approach [16] and it is competitive to existing heuristics. Moreover, the proposed approach is about one order of magnitude faster than [16] and solutions are found in tens of milliseconds using a conventional desktop computer. The main contributions of the paper are:

- Novel growing self-organizing map for the OP, TOP, and their variants with neighborhoods.
- Improved performance of SOM for orienteering problems with computational complexity independent on the rewards, contrary to the previous work [15], [16].
- Evaluation of SOM solvers in standard benchmarks for the OP and TOP with comparison to existing heuristics.

The rest of the paper is organized as follows. An overview of existing approaches is presented in the next section. The OP and OPN are formally introduced in Section III together with the generalized the TOP. The proposed novel SOM-based unsupervised learning for orienteering problems is presented in Section V. Results and comparison of the proposed method with the previous SOM approach and existing heuristics are presented in Section VI. Concluding remarks and future research work are in Section VII.

II. Related Work

Several approaches to the OP and TOP and their further variants have been proposed in the literature [5], [6]. The existing methods include exact solutions based on branch-and-bound techniques [19], [20] but due to a high computational complexity of the exact solvers, heuristic methods have been developed, e.g., [21], [18], [22]. The solvers are further improved by metaheuristics [23], [9] based on a combination of several techniques such as evolutionary methods, tabu search [24], Variable Neighborhood Search (VNS), Particle Swarm Optimization [25], [26], and even improvement heuristics for the TSP, i.e., LKH [27].

Regarding approaches to the orienteering problems based on neural networks, to the best of our knowledge, there is only a single approach [28] based on continuous Hopfield neural network in the literature, except the recent works [16], [15]. The approach [28] relies on traditional heuristics for the TSP that are employed in an iterative minimization of the proposed relatively complex energy function. The authors report that the proposed combination of heuristics is crucial to find solutions that are competitive to existing combinatorial solution of the OP. On the other hand, the approaches [16], [15] are solely based on the unsupervised learning and they originate from the work [13] that introduces growing SOM to address the Prize-Collecting Traveling Salesman Problem with Neighborhoods (PC-TSPN) and which provides better results than existing combinatorial heuristics in data collection planning in terms of the solution quality and the required computational time.

Based on the findings reported in [13], the main source of the data collection plan improvement is in the determination of suitable waypoints at which the requested data from the sensors are collected, i.e., the neighborhoods part of the problem formulation. Therefore, it is desirable to consider the neighborhoods also in the formulation of the Orienteering Problem for the data collection planning that better fits the practical limitations of the operational time of robotic vehicles than the PC-TSPN, in which the tour length is minimized but not guaranteed to be within the budget limit.

The SOM for the TSP follows the main idea of the Kohonen's unsupervised learning, where the neuron weights share the space with the input space. The network forms a uni-dimensional structure (not a usual 2D grid of cells in an ordinary SOM based approaches) that forms a ring of neurons that represents a tour in the input space. The unsupervised learning for the TSP is an iterative procedure in which the given locations (cities) to be visited are presented to the network and the best matching neuron is adapted to the presented location together with its neighboring neurons. The power of the adaptation is according to the neighboring function that decreases the power of the adaptation with increasing cardinal distance (i.e., measured as the number of neurons) of the neighboring neuron to the winner neuron. During the learning, the ring of neurons evolves in the input space to fit the presented locations (cities) and the final tour to visit the locations is retrieved by traversing the ring as a sequence of the locations associated with the particular neurons.

The main difficulty to address the OP by the SOM for the TSP is related to the constrained travel budget and rewards to prefer locations with high rewards. In [15], the authors propose to duplicate each individual input location (which in fact is an observation region to collect data about a particular object of interest) by a factor determined from the greatest common divisor of the set of rewards. This is motivated to adapt the network towards the areas providing higher rewards more often in each learning epoch. The main drawback of the duplication is that the computational complexity is dependent on the size of the input, and thus it depends on the particular setup of the rewards and the duplication can result in a significantly larger input than the original problem. In addition, a feasible solution of the OP has to satisfy the budget limit. In the SOM learning, this is achieved by a conditional winner selection and its actual adaptation according to the expected length of the ring (route) after the adaptation. Therefore, if the adaptation results in budget constraint violation, the winner neuron is not selected, and the learning proceeds with the next input.

The first application of SOM to the OP has been introduced in [15] to solve active perception problem with a group of mobile robots, and the performance of the SOM-based solver is not compared with existing solutions and standard benchmarks. Besides, the solution for the multi-robot case has been found as individual patrolling routes that support spatial partitioning of the input space into independent regions that are determined by the SOM during the learning. The SOM learning has been further applied in the single-competitor OP and compared with existing heuristics for the OP on the available benchmarks in [16]. The SOM for the OP is competitive to the existing heuristics proposed in [21], [22], [18], [5] on the instances of the Tsiligirides problem sets [2], [17]. For problem instances from the Set 64 and Set 66 [18], [17] benchmarks, the solutions provided by SOM are slightly worse than solution found by the heuristics algorithms. On the other hand, the main benefit of the SOM solver is its ability to solve the OPN, in which even a short communication radius significantly improves the total collect rewards in the problems of the Set 64 and Set 66. However, the main drawback of the previous SOM approaches proposed in [15], [16] is that their computational complexity depends on particular values of the rewards that directly influence the number of duplicated locations used for learning. The herein proposed novel adaptation procedure addresses this drawback and significantly decreases the computational burden while it also improves the solution quality in the existing benchmarks for the OP.

III. Problem Statement

The studied Orienteering Problem (OP) and its variant with Neighborhoods (OPN) together with their extension for multi-vehicle missions is motivated by robotic data collection planning for autonomous vehicles to collect sensor measurements from a given set of sensor locations S. Each sensor location may provide measurements of various importance, and thus each sensor location $s_i \in S$ has associated reward ς_i that is collected by retrieving data from the sensor. It is assumed a robotic vehicle has limited travel budget, and therefore, the problem to determine the most valuable sensor measurements can be formulated as the OP or TOP. In the case, the requested measurements at the sensor locations can be collected remotely, e.g., using a wireless communication, the range for a reliable data transfer is denoted δ and the problem can be formulated as the OPN, where the neighborhood of $s_i \in S$ is defined as a disk centered at s_i with the radius δ. The problems are formally introduced in the following sections.

A. Orienteering Problem (OP)

Let $S = \{s_1, \ldots, s_n\}$ be a set of n sensor locations in $\mathbb{R}^2$ with a particular position denoted by $s_i \in \mathbb{R}^2$. Each $s_i \in S$ has its associated reward $\varsigma_i \geq 0$. The data collection vehicle operates in $\mathbb{R}^2$ and its travel cost between any two points $p_1, p_2 \in \mathbb{R}^2$ is the Euclidean distance $|(p_1, p_2)|$. The initial location of the vehicle is s_1 and the requested final location is s_n, the locations can be different $s_1 \neq s_n$, and their associated rewards are zero $\varsigma_1 = 0$ and $\varsigma_n = 0$ [21]. The problem is to determine at most T_{max} long path from s_1 to s_n that maximizes the sum of collected rewards by visiting some subset of locations $S_k \subseteq S$. Therefore we need to determine the most valuable subset S_k such that a path connecting $s_i \in S_k$ starts at s_1, terminates at s_n, and its length is not longer than T_{max}. The requested tour T can be represented as a permutation of the k sensor locations $T = (s_{\sigma_1}, \ldots, s_{\sigma_k})$ where σ_i is the sensor label from the permutation $\Sigma = (\sigma_1, \ldots, \sigma_k)$ where $1 \leq \sigma_i \leq n$, $\sigma_i \neq \sigma_j$ for $i \neq j$ and with the defined start and end locations of the tour $\sigma_1 = 1$ and $\sigma_k = n$. The OP can be then defined as the problem to determine the number of sensors k, the subset of sensors S_k, and their sequence Σ such that

$$\begin{aligned} &\textit{maximize}_{k,S_k,\Sigma} && R = \sum_{i=1}^{k} \varsigma_{\sigma_i} \\ &\textit{subject to} && \sum_{i=2}^{k} |(s_{\sigma_{i-1}}, s_{\sigma_i})| \leq T_{max}, \\ &&& s_{\sigma_1} = s_1,\ s_{\sigma_k} = s_n, \end{aligned} \tag{1}$$

where R is the sum of the collected rewards.

B. Orienteering Problem with Neighborhoods (OPN)

The Orienteering Problem with Neighborhoods (OPN) is an extension of the above introduced OP to address situations when rewards at the locations S can be collected within δ distance from the individual sensor locations. Here, the problem is not only to determine the number of locations k, the subset $S_k \subseteq S$, and the permutation Σ, but also particular waypoints $p_{\sigma_i} \in \mathbb{R}^2$ at which data can be reliably collected from $s_{\sigma_i} \in S_k$, i.e., $|(p_{\sigma_i}, s_{\sigma_i})| \leq \delta$. In this case, the solution is not a tour T visiting the locations of S, but a sequence of waypoints $P = (p_{\sigma_1}, \ldots, p_{\sigma_k})$ with the length $|P|$ shorter than or equal to the travel budget T_{max}. Similarly to the OP, the initial and end locations of the path are the locations s_1 and s_n, respectively. The problem can be defined as

$$\begin{aligned} &\textit{maximize}_{k,S_k,P_k,\Sigma} && R = \sum_{i=1}^{k} \varsigma_{\sigma_i} \\ &\textit{subject to} && \sum_{i=2}^{k} |(p_{\sigma_{i-1}}, p_{\sigma_i})| \leq T_{max}, \\ &&& |(p_{\sigma_i}, s_{\sigma_i})| \leq \delta, \quad p_{\sigma_i} \in \mathbb{R}^2, \\ &&& p_{\sigma_1} = s_1,\ p_{\sigma_k} = s_n. \end{aligned} \tag{2}$$

C. Team Orienteering Problem (with Neighborhoods)

In the case, a fleet of m vehicles is available, the problem is to determine m tours $\mathcal{T} = \{T^1, \ldots, T^m\}$, one for each vehicle, such that each individual tour $T^r \in \mathcal{T}$, $1 \leq r \leq m$ is not longer than the budget T_{max}. The vehicles can be requested to start and terminate at the same locations (as in the existing benchmarks [9], [6]), e.g., s_1 and s_n, but otherwise it is supposed the tours visit different locations, i.e., the collection of the reward from a single location is not cumulative. The formulation of the TOP is basically an extension of the previously introduced OP for an individual vehicle that is denoted by the superscript r. Thus, the problem is to determine for each vehicle r, the subset $S^r_{k^r} \subseteq S$ with k^r locations, and the permutation $\Sigma^r = (\sigma^r_1, \dots \sigma^r_{k^r})$ such that the locations (except s_1 and s_n) are not visited by more than a single tour and all tours satisfy the budget T_{max}. The problem can be formally defined as:

$$\textit{maximize}_{(k^r, S^r_{k^r}, \Sigma^r) \text{ for } r\in\{1,\dots,m\}} \quad R = \sum_{r=1}^{m}\sum_{i=1}^{k^r} \varsigma_{\sigma_i^r} \tag{3}$$

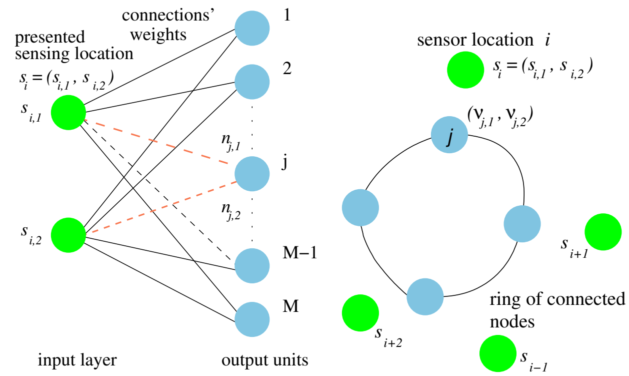
$$\begin{aligned} \textit{subject to} \quad & \sum_{i=2}^{k^r} |(s_{\sigma^r_{i-1}}, s_{\sigma^r_i})| \leq T_{max}, \\ & s_{\sigma^r_1} = s_1, s_{\sigma^r_{k^r}} = s_n, \text{ for } r \in \{1,\dots,m\}, \\ & S^i_{k^i} \bigcap S^j_{k^j} = \{s_1, s_n\} \text{ for } i \neq j, \\ & i,j \in \{1,\dots,m\}. \end{aligned} \tag{4}$$

The formulation of the Team Orienteering Problem with Neighborhoods (TOPN) is pretty much similar to the OPN and TOP. The problem is to determine individual paths $\mathcal{P} = \{P^1, \dots, P^m\}$, where each $P^r \in \mathcal{P}$ is not longer than T_{max}.

The formulations of the OP and TOP follow the existing problem definitions [6], albeit they are usually formulated as Integer Linear Programming (ILP) problem. However, the variants with the neighborhoods are not well suitable for the ILP, as the particular waypoints can be arbitrarily selected from the neighborhood represented by a disk. Moreover, it is possible that a single waypoint allows to collect rewards from several locations. The proposed SOM-based approach is capable of solving all the introduced problems: OP, OPN, TOP, and TOPN in a single unifying way. Its performance is compared with existing heuristic in standard benchmarks for the OP and TOP. Besides, selected instances of the benchmarks are considered with the non-zero communication radius δ to demonstrate a solution of the OPN and TOPN.

IV. Winner Selection in Self-Organizing Maps for Routing Problems with Neighborhoods

The key idea of the SOM for routing problems with neighborhoods is that during the selection of the winner node, it is also determined the particular waypoint from the neighborhood of the sensor location. The winner selection used for solving orienteering problems is based on the growing self-organizing map for the PC-TSPN introduced in [13]. Therefore, the main idea of the winner selection is briefly described in this section to provide an overview of the fundamental concepts to the readers, not familiar with [13]. The idea follows the standard SOM for the TSP which is a two-layered neural network. The input layer serves for presenting the sensor locations (cities) and the output layer is organized into an array of neurons. The neuron weights share the space with the input, and thus the connected neurons form a ring in the input space, see Fig. 1. Having M neurons $\mathcal{N} = \{\nu_1, \dots, \nu_M\}$, the winner neuron is determined as the best matching neuron ν^* for the currently presented location s_i to the input of the network. In a regular SOM for the TSP, e.g., [29], the winner neuron ν^* is determined as the closest neuron to s_i:



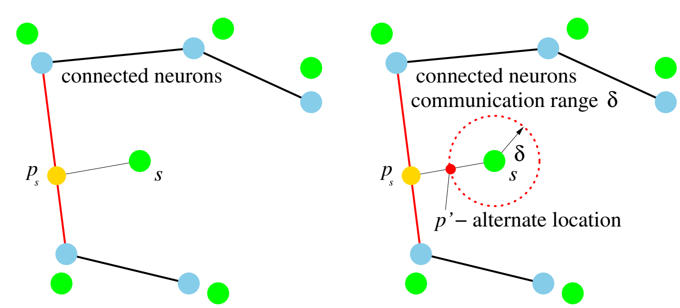
(a) Two-layered SOM (b) Ring of neurons

Fig. 1. Structure of the SOM for the TSP

$$\nu^* = \text{argmin}_{\nu \in \{\nu_1, \dots, \nu_M\}} |(\nu, s_i)|. \tag{5}$$

Even though this seems to be a suitable choice for the ordinary TSP with a fixed number of locations (cities) visited by the final route, it is not suitable for the routing variants in which a subset of the given locations are selected, like in the PC-TSPN or the addressed orienteering problems.

In [13], the authors consider the ring of the connected neurons as a sequence of straight line segments and expected location of the winner neuron is determined as the closest point p_s of the ring to the presented location s, see Fig. 2a. The point p_s is determined prior the selection of the winner neuron and the actual winner neuron is determined only if the adaptation of the network is performed, e.g., the conditional adaptation in the PC-TSPN [13] or considering the limited travel budget in the OP [16]. Once p_s is determined, the winner neuron for s is selected as an existing neuron with the weights identical to p_s or a new neuron is created between the particular neurons corresponding to the segment endpoints and the weights of the new neuron are set to be identical to p_s.



(a) Determined point p_s for the weights of the winner neuron (b) Alternate location p' to collect data from s

Fig. 2. The winner selection in the PC-TSPN and OP(N) and the alternate location p' to collect data from s within the communication radius δ

If data from s can be remotely collected using a wireless communication, it is not necessary to visit a particular location s, and therefore, the point p_s is used to determine an alternate location p' at which data can be reliably read from s. The point p' is an intersection point of the straight line segment (p_s, s) and the disk with the radius δ centered at s. Then, such a point p' is used as an alternate location towards which the network is adapted instead of s. If p_s is already inside the disk, the data from s can be retrieved from p_s, and thus only new neuron is added to the network and the network is not adapted towards s [16].

Notice the described winner selection may create a new neuron with the weights set to p_s. To avoid only increasing the number of neurons in the network, the neurons not selected as winners in the current learning epoch (defined as a presentation of all sensor locations S to the network) are deleted at the end of each learning epoch. Thus, the number of neurons is adapted to the currently selected subset of S during the learning.

The ideas of the conditional adaptation and determination of the weights of the expected winner neuron are used in the proposed novel SOM for the OP. The idea of the alternate location (shown in Fig. 2b) is used in the OPN. The both ideas are further developed in the proposed new adaptation procedure presented in the next section.

V. Proposed Self-Organizing Map for Orienteering Problems

The proposed unsupervised learning to address challenges arising in solving orienteering problems builds on the previous work on SOM for the PC-TSPN [13], OP [16], and its application in active perception [15]. The crucial property of the network for orienteering problems is to satisfy the travel budget constraint, and thus the network is not adapted if the tour represented by the network would violate the constraint. Besides, the learning should prefer adaptation to locations with higher rewards. In this paper, we propose to weight the power of the adaptation according to the reward associated with the location, which addresses the main drawback of the previous approaches [15], [16] that duplicate the sensor locations, and thus increase computational complexity of the learning.

The conditional adaptation is achieved by the determination of the neuron weights for the possible winner neuron as the point p_s (visualized in Fig. 2). The determination of the neuron weights is further developed to support an escape from local optima. The original idea of the SOM for the OP [16] uses only the conditional adaptation solely based on the length of the tour represented by the ring after such an adaptation that is supposed to be shorter than the travel budget T_{max}. Once the network represents a tour that is close to the travel budget, such a network is unlikely adapted to other locations and the learning is stuck in local optima.

The herein proposed procedure considers ideas of insertion heuristics in which a location with the lowest reward or a location that mainly contributes to the length of the route are deleted in benefit of newly added location to the route. In the network adaptation, the corresponding neuron to such a location is removed from the network instead of the sensor location itself as in insertion heuristics. However, such a modification of the network still may not be enough to allow the adaptation due to the travel budget constraint. Therefore the configuration of the network is saved prior the determination of neuron weights of the expected winner (i.e., the point p_s). If the network is not adapted because of T_{max}, all modifications of the network performed during the determination of p_s are reverted back to the state before p_s selection and the learning proceeds with the next sensor location. The proposed unsupervised learning based on these ideas is summarized in the rest of this section.

In the proposed SOM learning, all sensor locations $S \in \{s_1, \ldots, s_n\}$ are presented to the network in a single learning epoch. Then, the learning gain G is updated according to the gain decreasing rate α. The adaptation uses the determination of the point p_s as the neuron weights of the expected winner. The determination may modify the network, and thus the state of the network as a sequence of the neuron weights $\mathcal{N} = \{\nu_1, \ldots, \nu_M\}$ is saved and it is eventually reverted to that state if the adaptation is not performed because of the travel budget T_{max}. [1] The winner neuron ν^* is adapted towards the presented location s together with its neighboring neurons in the d neighborhood (counted as the number of neurons in the ring) according to the neighboring function $f(G, d)$:

$$f(G,d) = \begin{cases} e^{\frac{-d^2}{G^2}} & \text{for } d < 0.2M \\ 0 & \text{otherwise} \end{cases}, \qquad (6)$$

where G is the learning gain and M is the current number of the neurons in the ring. The learning procedure is as follows:

▷ **Initialization:**

1) Initialize the network as $\mathcal{N} = (\nu_1, \nu_{end})$ with two neurons ν_1 and ν_{end} that are located at s_1 and s_n. The neurons represent the requested start and end locations of the tour. These neurons are never removed nor adapted during learning. They define the ring end points.
2) The learning parameters are initialized to the following values: the learning gain $G = 10$, the default learning rate $\mu = 0.6$, and the gain decreasing rate $\alpha = 0.1$.
3) Determine the maximal reward $R_{max} = \text{argmax}_{s \in S}\, \varsigma(s)$.
4) Set the current best found solution $T = (s_1, s_n)$ and its sum of rewards $R = 0$ because $\varsigma_1 = \varsigma_n = 0$.
5) Set the learning epoch counter i to $i = 1$.

▷ **Learning epoch:**

6) Randomize the sensor locations $S = \{s_1, \ldots, s_n\}$ except s_1 and s_n; $\Pi \leftarrow \text{permute}(S \setminus \{s_1, s_n\})$.
7) For each $s \in \Pi$:
    a) Save the current network $\mathcal{N}' \leftarrow \mathcal{N}$.
    b) $(\mathcal{N}, p_s) \leftarrow \text{winner\_weights}(s, \mathcal{N}, T_{max}, i)$.

[1] In fact, it is implemented as saving changes to the network and reversion to the previous state by applying inverse changes, which is more computationally efficient than saving the whole network.

c) If p_s has identical coordinates with ν_1 or ν_{end} mark the respective neuron as the winner neuron in the current epoch i.

d) For p_s lying on the ring, determine the first previous winner ν_p (in the direction towards ν_1) and the first next winner ν_n (in the direction towards ν_{end}) from p_s.

e) Let a tour represented by the current winner neurons of $\mathcal{N}$ be T_{win} and its length be $L_{T_{win}}$. The expected length of the tour represented by the ring after a possible adaptation can be expressed as

$$L_{T_{win}} - |(s_{\nu_p}, s_{\nu_n})| + |(s_{\nu_p}, s)| + |(s, s_{\nu_n})| \leq T_{max}, \quad (7)$$

where s_{ν_p} and s_{ν_n} are the associated sensor locations to the winners ν_p and ν_n, respectively.

▷ **Conditional Adapt:**

f) `If` (7) holds `Then`
- Create a new neuron ν^* with the weights identical to p_s and insert it to $\mathcal{N}$ at the position corresponding to the particular edge connecting the neurons, see Fig. 2b.
- Adapt ν^* and its neighboring neurons to s using the neighboring function (6), i.e., for each neuron ν in the d neighborhood of ν^* adjust the weights of ν according to new weights ν'

$$\nu' = \nu + R_s \mu f(G, d)(s - \nu), \quad (8)$$

  where R_s is the ratio of the reward of s and R_{max}, i.e., $R_s = \varsigma(s)/R_{max}$. Avoid adaptation of ν_1 and ν_{end}.
- Associate s with ν^* and mark ν^* to be one of the winner neurons in the current epoch i.

g) `Else`
- Revert the network to the state before the determination of the point p_s, i.e., $\mathcal{N} \leftarrow \mathcal{N}'$.

h) `End` – conditional adaptation.

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

8) Remove all non-winner neurons from the ring $\mathcal{N}$.
9) Update learning parameters: $G \leftarrow G(1 - i\alpha)$, $i \leftarrow i + 1$.
10) `If` the tour T_{win} represented by the current winners in $\mathcal{N}$ provides higher sum of the rewards than the best solution found so far, update the solution $T \leftarrow T_{win}$ and its collected rewards $R \leftarrow \sum_{s_i \in T} \varsigma(s_i)$.
11) `If` the number of learning epochs i reaches the limit i_{max} `Stop` the learning; `Otherwise` go to Step 6.

The adaptation procedure iteratively determines the point p_s and creates a new neuron with the weights set to p_s that is then adapted towards the currently presented sensor location s. After each learning epoch, the non-winner neurons are removed and additional neurons can be removed from the network in the winner_weights() procedure to further support the adaptation of the network to s while still preserve the travel budget constraint. Thus, the number of neurons M in the ring never exceeds the number of sensor locations n.

Procedure winner_weights$(s, \mathcal{N}, T_{max}, i)$:
1) Get all neurons marked as winners in the epoch i $\mathcal{N}_{win} \leftarrow$ winners$(\mathcal{N} \setminus \{\nu_1, \nu_{end}\}, i)$.
2) Let each winner $\nu \in \mathcal{N}_{win}$ has associated sensor location $s_\nu = s(\nu)$ with the reward $\varsigma_\nu = \varsigma(s_\nu)$.
3) Determine the winner ν_f which has the longest distance to its associated location s_{ν_f}, i.e.,

$$\nu_f = \text{argmax}_{\nu \in \mathcal{N}_{win}} |(\nu, s_\nu)|. \quad (9)$$

4) Determine the winner ν_l which associated sensor location s_{ν_l} has the lowest reward, i.e.,

$$\nu_l = \text{argmin}_{\nu \in \mathcal{N}_{win}} \varsigma(s_\nu). \quad (10)$$

5) Determine the closest point p_s of the ring $\mathcal{N}$ to the location s as in Fig. 2a.
6) `If` the expected route length after adapting a winner node created at the location p_s would be longer than T_{max}
   - `If` $\varsigma(s_{\nu_f}) < \varsigma(s)$ AND $|(\nu_f, s(\nu_f)| > |(p_s, s)|$ `Then` remove ν_f from the ring $\mathcal{N} \leftarrow \mathcal{N} \setminus \{\nu_f\}$.
   - `If` $\varsigma(s_{\nu_l}) < \varsigma(s)$ AND $|(\nu_l, s(\nu_l)| > |(p_s, s)|$ `Then` remove ν_l from the ring $\mathcal{N} \leftarrow \mathcal{N} \setminus \{\nu_l\}$.
7) return$(\mathcal{N}, p_s)$.

Fig. 3. The procedure to determine the point p_s as the expected weights of the winner neuron for the sensor location s in the current ring $\mathcal{N}$. The procedure may change $\mathcal{N}$ to respect the travel budget T_{max}.

The procedure winner_weights() is depicted in Fig. 3. It searches for a neuron ν_f from the winner neurons of the current epoch $\mathcal{N}_{win}$ such that ν_f is the farthest neuron from its associated sensor location and a neuron $\nu_l \in \mathcal{N}_{win}$ that is associated with the sensor location with the lowest reward. Then, if the adaptation to s using a new winner at p_s would exceed the budget T_{max} the neurons ν_f and ν_l are removed from the network to shorten the tour represented by the ring, but only if the current location s would add a higher reward to the tour than the locations associated to ν_f or ν_l.

Even though the proposed adaptation originates from [16], its main advantages are that it does not need the duplication of the locations from S and it also keeps the number of neurons lower than the original procedure [16]. The performance of the proposed adaptation procedure is empirically evaluated in Section VI where it is compared with the previous SOM for the OP and also with other existing heuristic approaches.

A. SOM for the Orienteering Problem with Neighborhoods

In the case the data from a sensor location $s \in S$ can be retrieved within the communication radius δ, an alternate location p' at which data can be read from s is determined from the point p_s and the disk with radius δ centered at s, see Fig. 2b. Such a point p' is then used as the alternate location instead of the sensor location s in the adaptation procedure described in the previous section, i.e., s becomes p' ($s \rightsquigarrow p'$). The ring is adapted towards p' and the ring itself is the data collection path. Therefore, p' is used in the conditional deletion of ν_f and ν_l in the procedure winner_weights() in Fig. 3. The point p' is also used (instead of s) in the adaptation condition (7) and the adaptation itself (8). Because all non-winner neurons are removed from the network at the end of each learning epoch, the requested data collection path is retrieved from the sequence of the winners as a sequence of the associated alternate locations p'.

The main benefit of non-zero communication radius δ is in saving the travel cost, and thus a higher sum of the collected rewards can be expected for a solution of the OPN with $\delta > 0$ with the same T_{max} as for the ordinary OP.

B. SOM-based Solution of the Team Orienteering Problems

The TOP can be addressed by creating an individual ring for each vehicle. Then, the rings can compete to be selected for the adaptation towards the particular s according to the ratio of the current length of the tour represented by the ring and the travel budget as in [15]. However, for dense problems with a high number of locations, the individual rings become quickly saturated and the tour lengths would be quickly close to the travel budget T_{max}. Another alternative can be a greedy selection of the ring for the adaptation based on the highest ratio of the gained reward to the expected tour length.

In an early evaluation of the both aforementioned approaches, such a competitive SOM provides relatively poor performance and the total collected reward is only slightly increasing with increasing the number of vehicles. Therefore, an incremental approach is rather evaluated in this paper and the TOP with m vehicles is solved as a sequence of the m OP instances. For each individual vehicle $r \in \{1, \ldots, m\}$, sensor locations visited by the solution of the OP for one vehicle are not considered in the consecutive solution for the next vehicle.

The same approach is also used for the Orienteering Problem with Neighborhoods (TOPN), where individual solutions are based on the above proposed SOM-based solution to the OPN. Empirical results of this naive SOM-based approach to the TOP and TOPN are reported in Section VI.

C. Computational Complexity

The computational complexity of the proposed learning procedure depends on the number of locations n and the number of neurons in the ring. The number of neurons is proportional to the number of locations and it is $M \leq n$. In each learning epoch, one winner neuron is eventually determined for each sensing location from up to n existing neurons in the ring, and thus the time complexity of a single epoch can be bounded by $O(n^2)$. In the case of the TOP, we consider up to m individual rings, but the total number of neurons among the rings cannot exceed n (except the path endpoints), and therefore, the time complexity of a single learning epoch can be bounded by $O(n(n+m))$. The variants with neighborhoods only slightly increase the computational time as it contains only the additional determination of the point p'; hence, the time complexity of a single learning epoch can also be bounded by $O(n^2)$ and $O(n(n+m))$, respectively.

VI. Results

The proposed SOM adaptation for orienteering problems has been evaluated using the existing benchmarks for the OP and TOP [6]. The evaluation methodology follows the results in [16] where the first systematic evaluation of the SOM-based approach to the OP has been presented. Here, the novel SOM based approach denoted *Proposed SOM* is compared with the previous approach denoted SOM-OP [16].

Both SOM approaches are relatively computationally inexpensive, and therefore, the maximal number of learning epochs is set to $i_{max} = 500$. Besides, they are also stochastic search procedures, and thus each particular problem instance is solved 50 times. The results for the other algorithms are solutions reported in the literature, and thus they represent a single trial.

The algorithm performance is evaluated using the relative percentage error (RPE) defined as relative error between the reference value R_{ref} and the best found solution R over the performed trials [25], [26], where R_{ref} is the highest reward of the best known solution of the particular instance reported in the literature [26], [9], [6]. The RPE is computed as $RPE = (R_{ref} - R)/R_{ref} \cdot 100$. The robustness of the algorithm over the performed trials is shown as the average relative percentage error (ARPE) [26], i.e., $ARPE = (R_{ref} - R_{avg})/R_{ref} \cdot 100$.

Due to limited space, aggregated results are presented as the respective average values of the RPE and ARPE of the solved problem instances for a particular problem set. Aggregated indicators are denoted $\overline{\text{RPE}}$ and $\overline{\text{ARPE}}$, respectively. The required computational time is reported as the average time needed to solve a single trial. The standard deviations of the time are below 5%, and therefore, they are omitted from the presentation. The SOM algorithms have been implemented in C++ and executed within the computational environment using a single core of the iCore7 CPU running at 4 GHz.

A. Results for the Orienteering Problem

The evaluation of the Orienteering Problem (OP) has been performed using the problem instances proposed by Tsiligirides in [2] (the problems Set 1, Set 2, and Set 3) and problems proposed by Chao et al. in [18] that are available at [17] as the Set 64 and Set 66. In total, 89 different instances of the OP with various budgets have been solved. Regarding the previous evaluation presented in [16], we consider only the heuristic approach [18] denoted by the CGW, as the representative state-of-the-art heuristic for the OP. All aggregated results per particular problem set are depicted in Table VI-A, where the last column *Speedup* denotes how many times is the newly proposed SOM algorithm faster than the SOM-OP [16].

The results indicate that overall the proposed algorithm provides improved or similar results to the SOM-OP [16] but it is always significantly faster. More detailed results for problem instances of the Set 64 and Set 66 are presented in Table II and Table III, respectively. It can be observed that the proposed approach provides even better results than heuristics CGW in few cases but almost always better results than the previous SOM approach [16]. Solutions of the selected problems and their further improvement for $\delta > 0$ are visualized in Fig. 5.

TABLE I
AGGREGATED RESULTS FOR THE OP

Set	CGW [18]	SOM-OP [16]		Proposed SOM		
	$\overline{\text{RPE}}$	$\overline{\text{RPE}}$	$\overline{\text{ARPE}}$	$\overline{\text{RPE}}$	$\overline{\text{ARPE}}$	Speedup ×
Set 1	0.10	0.25	2.45	0.10	1.05	2.1
Set 2	0.92	0.92	1.94	0.92	1.10	7.4
Set 3	0.00	0.00	1.73	0.00	0.89	13.5
Set 64	0.33	2.76	6.25	1.55	4.29	11.4
Set 66	0.41	2.08	6.11	1.51	5.25	20.4

TABLE II
SOLUTIONS OF THE OP FOR INSTANCES FROM SET 64

T_{max}	CGW	SOM-OP			Proposed SOM		
	RPE	RPE	ARPE	CPU*	RPE	ARPE	CPU*
15	0.00	0.00	0.00	96.7	0.00	0.00	13.0
20	0.00	0.00	0.00	475.2	0.00	1.31	26.4
25	0.00	0.00	6.15	578.2	1.54	7.51	31.5
30	0.00	3.80	8.91	640.1	2.53	8.03	38.5
35	0.00	1.05	6.21	655.3	0.00	3.77	45.1
40	0.00	3.36	8.45	677.1	2.52	5.36	51.6
45	0.00	6.62	10.69	691.1	1.47	5.97	58.2
50	0.00	6.67	10.25	693.8	3.33	6.04	63.8
55	0.00	4.88	10.67	689.4	2.44	4.95	68.5
60	1.69	4.52	9.36	673.3	2.82	5.15	72.5
65	0.00	3.23	6.01	645.9	1.61	3.54	76.0
70	1.01	3.54	5.44	613.5	2.02	3.83	79.8
75	0.97	0.97	3.61	578.4	0.49	2.85	82.9
80	0.93	0.00	1.74	537.3	0.93	1.82	84.6

*The reported computational times are in milliseconds.

TABLE III
SOLUTIONS OF THE OP FOR INSTANCES FROM SET 66

T_{max}	CGW	SOM-OP			Proposed SOM		
	RPE	RPE	ARPE	CPU*	RPE	ARPE	CPU*
15	0.00	0.00	0.00	281.2	0.00	0.00	15.2
20	4.88	0.00	1.76	558.3	0.00	3.51	22.8
25	0.00	0.00	1.76	866.4	0.00	3.83	26.2
30	0.00	0.00	9.20	999.3	0.00	8.62	29.9
35	1.08	1.08	7.96	1128.3	1.08	7.38	32.1
40	0.00	5.22	11.70	1254.3	4.35	11.22	37.0
45	0.00	0.77	11.65	1326.4	0.77	11.22	39.9
50	0.00	4.11	10.71	1390.3	4.79	10.47	45.0
55	0.00	0.61	11.07	1396.8	1.21	10.61	49.2
60	0.00	6.56	11.65	1411.5	3.28	10.50	54.2
65	0.00	1.02	9.18	1423.3	4.59	8.08	59.0
70	0.00	2.80	9.23	1413.9	2.80	7.54	63.6
75	0.00	5.26	9.34	1405.6	2.63	6.91	67.3
80	0.00	4.94	9.09	1383.8	2.47	6.30	70.2
85	0.00	3.54	7.79	1346.9	1.97	4.64	73.1
90	0.00	5.22	8.00	1305.7	2.24	4.63	76.0
95	1.08	4.66	6.86	1252.1	1.79	4.11	79.5
100	2.05	3.41	6.66	1193.4	2.05	4.64	80.5
105	0.66	3.29	6.12	1126.6	2.30	4.41	82.6
110	0.00	1.29	3.74	1061.4	0.65	2.87	85.0
115	0.00	0.31	2.75	989.7	0.31	2.26	85.8
120	0.00	0.00	1.38	909.9	0.00	1.49	87.3
125	0.90	0.00	1.04	739.4	0.00	1.05	89.0
130	0.00	0.00	0.32	505.7	0.00	0.08	62.9

*The reported computational times are in milliseconds.

B. Results for the Orienteering Problem with Neighborhoods

The performance of the SOM-based solvers to the OPN is evaluated using standard benchmarks for the OP but with non-zero communication radius δ. The average collected rewards from 50 trials of the selected problems are depicted in Fig. 4. As expected, allowing to collect rewards within the δ distance saves the travel cost, and thus more rewards are collected. The proposed SOM provides outstanding results in the problem Set 66 with the travel budget $T_{max} = 60$, but it can be noticed that for Set 3 and $T_{max} = 50$, the new proposed approach provides a bit worse results for the communication radius $\delta = 1.0$, albeit it provides better results for zero communication radius. This is probably caused by a far location that may add

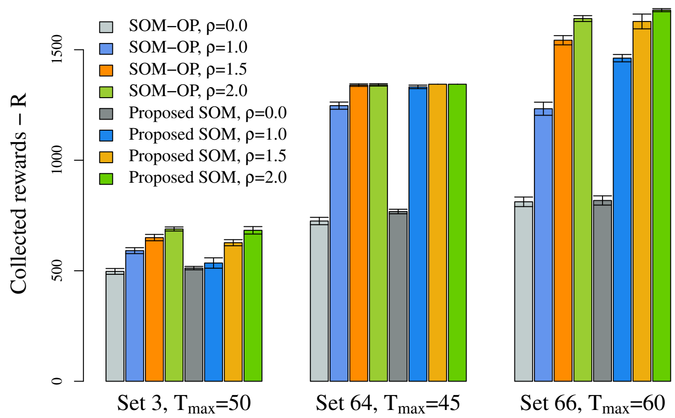
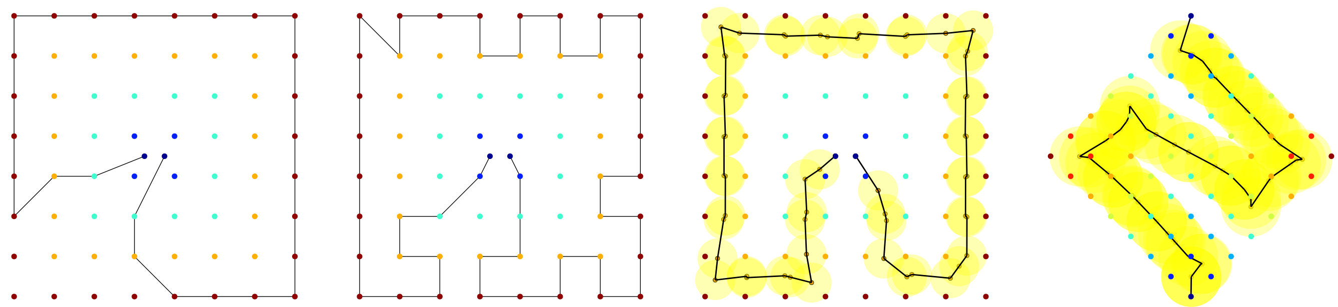


Fig. 4. Average sum of the collected rewards with increasing communication range δ. The standard deviations are showed as error bars and in some cases.

a significant reward to the solution. In the proposed approach, such a location can be preferred at the early epochs and then, it is never removed because of its high reward.

C. Results for the Team Orienteering Problem

The proposed SOM-based solver for the TOP has been evaluated in available datasets [17] and compared with the results published in [9] with various numbers of locations, budgets, and numbers of vehicles. The number of the evaluated problem instances is 148, and thus to report on evaluation within the limited space, the results are aggregated using the $\overline{\text{RPE}}$ and $\overline{\text{ARPE}}$ indicators for which the reference value is the best known solution reported in [9], [26]. The selected approaches for comparison is one of the first heuristic CGW [8], the tabu search and VNS-based approaches proposed in [9], and one of the latest approach PSO inspired algorithm [26] denoted the PSOiA. The results are reported in Table IV, where the first column denotes a set of problems with various budgets with the following notation. The first number denotes the problem set, i.e., Set 4, Set 5, etc., see [17], and the second number denotes the number of vehicles. The particular TOP instances for the individual budgets are selected according to the available results for CGW, TABU, VNS, and PSOiA approaches reported in [9], [26]. For example the problem set p4.4 represents problem instances of the Chao's Set 4 with 4 vehicles and with 16 different budget values from the range $20 \leq T_{max} \leq 60$, see [9], [26], [17] for further



(a) Set 66, T_{max}=60, δ=0.0, R=885 (b) Set 66, T_{max}=95, δ=0.0, R=1370 (c) Set 66, T_{max}=60, δ=1.0, R=1495 (d) Set 64, T_{max}=45, δ=1.5, R=1344

Fig. 5. Selected solutions of the Orienteering Problem (OP) and Orienteering Problem with Neighborhoods (OPN) found by the proposed SOM-based approach

TABLE IV
AGGREGATED RESULTS FOR THE TOP

Set	CGW	TABU	VNS	PSOiA	**Proposed SOM**		
	$\overline{\text{RPE}}$	$\overline{\text{RPE}}$	$\overline{\text{RPE}}$	$\overline{\text{RPE}}$	$\overline{\text{RPE}}$	$\overline{\text{ARPE}}$	CPU [ms]
p4.2	4.53	0.89	0.12	0.00	4.92	9.14	76.7
p4.3	4.45	0.36	0.06	0.00	5.03	8.90	52.8
p4.4	4.38	0.19	0.20	0.01	4.88	8.39	40.7
p5.2	0.92	0.04	0.09	0.00	3.63	7.96	41.0
p5.3	1.11	0.00	0.00	0.00	2.67	6.04	30.8
p5.4	2.32	0.00	0.00	0.00	1.22	4.18	25.8
p6.2	1.17	0.18	0.00	0.00	4.56	7.88	48.4
p6.3	0.80	0.00	0.00	0.00	1.90	5.80	29.5
p6.4	0.43	1.65	1.65	1.65	2.63	6.64	20.8
p7.2	1.38	0.19	0.03	0.00	4.21	9.32	60.3
p7.3	2.48	0.40	0.13	0.00	3.53	6.90	44.1
p7.4	4.42	0.37	0.03	0.00	3.23	5.89	31.8

details. The columns TABU and VNS denote results for the GEN_TABU_FEASIBLE and SLOW VNS_FEASIBLE approaches reported in [9], respectively.

The presented results indicate that the proposed SOM-based incremental solution of the TOP as a series of consecutive solutions of the OP does not compete with the computationally expensive metaheuristic such as the TABU and VNS based algorithms and currently the best performing PSOiA [26]. On the other hand, in few cases, the SOM-based approach provides competitive (or slightly better) results than the heuristic approach CGW proposed by Chao et al. in [8]. Regarding the required computational time, a solution of a single trial is found in tens of milliseconds, which is far faster than the required computational times reported in [9], which are hundreds of seconds. The reporting times for the PSOiA are in tens of seconds using dual-core processor with frequency around 2.5 GHz. Even though the reported times of the previous approaches [9], [26] are for slower computers, the SOM is about four orders of magnitude faster and it can be expected it would be still faster even on that computers.

Due to lack of the standard benchmarks for the TOP with Neighborhoods (TOPN), the ability of the proposed SOM-based approach to solve TOPN instances is demonstrated in Fig. 6, where increased rewards gained for δ>0 are reported.

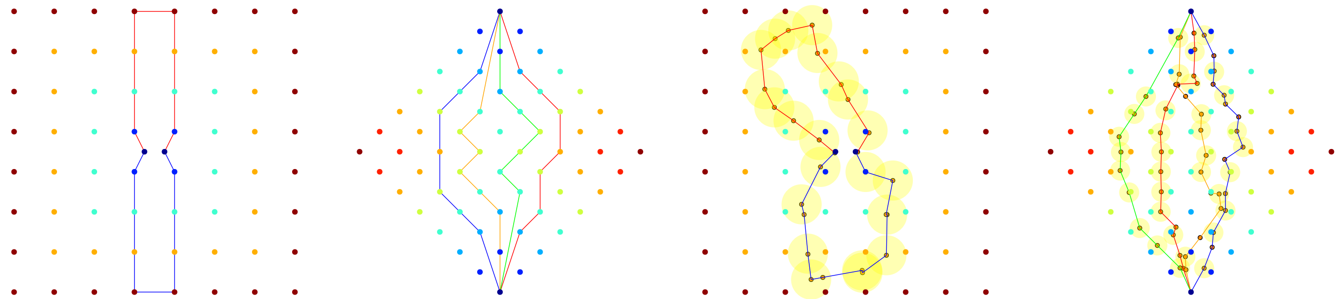
D. Discussion

The presented results indicate that the novel unsupervised learning procedure improves the performance of the SOM-based solution to the OP and OPN in comparison to the previous approach [16] while it is also significantly less computationally demanding. Regarding the motivational scenario of robotic data collection missions, the formulation of the problem as the OPN provides additional benefit in increased collected rewards for the same travel budget. However, the results indicate that in some cases, the previous SOM solver [16] provides better results, e.g., for Set 3 in Fig. 4. By a detailed analysis of the found paths, it is caused by the preference of the far locations with a relatively high reward in the new SOM procedure, while the previous SOM-based stochastic search [16] is able to escape from such a local optimum. This observation motivates us to further consider the SOM-based solution as a construction heuristic and improve solutions by evolutionary techniques, e.g., as in the memetic approach for the TSP presented in [30].

The proposed incremental approach to the TOP as a solution of a sequence of OP instances provides a feasible solution; however, the solution quality does not compete to the existing heuristic algorithms. On the other hand, the proposed SOM approach provides a solution of the TOPN, which is more suitable problem formulation for data collection using wireless communication than the ordinary TOP.

VII. CONCLUSION

In this paper, a novel unsupervised learning is proposed to solve orienteering problems by principles of self-organizing maps. The proposed approach builds on the previous work on SOM for the TSPN, OP, and OPN. Regarding the presented results based on standard benchmarks for the OP, the proposed approach provides competitive solutions to the previous SOM-based solutions while it is significantly faster and solutions are provided in tens of milliseconds using conventional computational resources. Moreover, the approach has been generalized to the Team Orienteering Problem and its variant with neighborhoods. Even though the proposed SOM-based approach



(a) p5.2, T_{max}=17.5, δ=0.0, R=320 (b) p6.4, T_{max}=16.2, δ=0.0, R=522 (c) p5.2, T_{max}=17.5, δ=1.0, R=440 (d) p6.4, T_{max}=16.2, δ=0.5, R=618

Fig. 6. Selected solutions found by the proposed SOM-based approach for the Team Orienteering Problem (TOP) (on the left) and the Team Orienteering Problem with Neighborhoods (TOPN) (on the right)

does not compete with the existing heuristics for the TOP, it provides solutions very quickly. In addition, the proposed approach is able to directly utilize a non-zero communication radius, and thus it may find solutions with higher rewards than demanding metaheuristics for the OP and TOP without considering the neighborhoods. Therefore the proposed SOM-based approach can be considered as a construction heuristic and combined with evolutionary techniques and existing meta-heuristics to improve solution not only in the OP and TOP but mainly in orienteering problems with neighborhoods which are suitable formulations for robotic information gathering scenarios with single or fleet of robotic vehicles.

References

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

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

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

[4] D. Bienstock, M. Goemans, D. Simchi-Levi, and D. Williamson, "A note on the prize collecting traveling salesman problem," *Mathematical Programming*, vol. 59, pp. 413–420, 1993.

[5] 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.

[6] 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.

[7] J. Yu, M. Schwager, and D. Rus, "Correlated orienteering problem and its application to persistent monitoring tasks," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1106–1118, 2016.

[8] I.-M. Chao, B. L. Golden, and E. A. Wasil, "The team orienteering problem," *European Journal of Operational Research*, vol. 88, no. 3, pp. 464–474, 1996.

[9] C. Archetti, A. Hertz, and M. G. Speranza, "Metaheuristics for the team orienteering problem," *Journal of Heuristics*, vol. 13, no. 1, pp. 49–76, 2007.

[10] J. Goerner, N. Chakraborty, and K. Sycara, "Energy efficient data collection with mobile robots in heterogeneous sensor networks," in *ICRA*, 2013, pp. 2527–2533.

[11] D. Thakur, M. Likhachev, J. Keller, V. Kumar, V. Dobrokhodov, K. Jones, J. Wurz, and I. Kaminer, "Planning for opportunistic surveillance with multiple robots," in *IROS*, 2013, pp. 5750–5757.

[12] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

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

[14] E. M. Arkin and R. Hassin, "Approximation algorithms for the geometric covering salesman problem," *Discrete Applied Mathematics*, vol. 55, no. 3, pp. 197–218, 1994.

[15] 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.

[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] "The Orienteering Problem: Test Instances – Department of Mechanical Engineering," citied on 2016-11-10. [Online]. Available: http://www.mech.kuleuven.be/en/cib/op#section-0

[18] 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.

[19] M. Fischetti, J. J. S. Gonzlez, and P. Toth, "Solving the orienteering problem through branch-and-cut," *INFORMS Journal on Computing*, vol. 10, no. 2, pp. 133–148, 1998.

[20] D.-C. Dang, R. El-Hajj, and A. Moukrim, "A branch-and-cut algorithm for solving the team orienteering problem," in *International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems*. Springer, 2013, pp. 332–339.

[21] 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.

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

[23] M. Schilde, K. F. Doerner, R. F. Hartl, and G. Kiechle, "Metaheuristics for the bi-objective orienteering problem," *Swarm Intelligence*, vol. 3, no. 3, pp. 179–201, 2009.

[24] H. Tang and E. Miller-Hooks, "A TABU search heuristic for the team orienteering problem," *Computers & Operations Research*, vol. 32, no. 6, pp. 1379–1407, 2005.

[25] S. Muthuswamy and S. S. Lam, "Discrete particle swarm optimization for the team orienteering problem," *Memetic Computing*, vol. 3, no. 4, pp. 287–303, 2011.

[26] D.-C. Dang, R. N. Guibadj, and A. Moukrim, "An effective PSO-inspired algorithm for the team orienteering problem," *European Journal of Operational Research*, vol. 229, no. 2, pp. 332–344, 2013.

[27] K. Helsgaun, "An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic," *European Journal of Operational Research*, vol. 126, no. 1, 2000.

[28] Q. Wang, X. Sun, B. L. Golden, and J. Jia, "Using artificial neural networks to solve the orienteering problem," *Annals of Operations Research*, vol. 61, no. 1, pp. 111–120, 1995.

[29] E. M. Cochrane and J. E. Beasley, "The co-adaptive neural network approach to the Euclidean travelling salesman problem," *Neural Networks*, vol. 16, no. 10, pp. 1499–1525, 2003.

[30] 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.