Self-Organizing Map-based Solution for the Orienteering Problem with Neighborhoods

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

Graeme Best
Australian Centre for Field Robotics (ACFR)
The University of Sydney
New South Wales, Australia
Email: g.best@acfr.usyd.edu.au

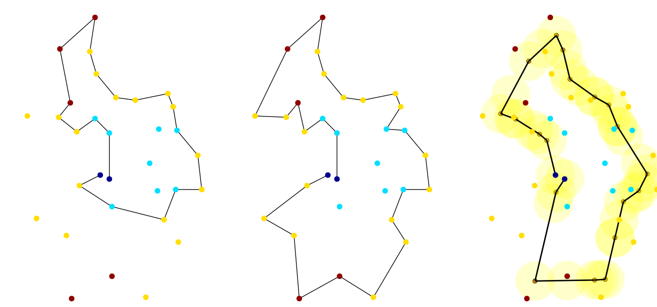
***Abstract*—In this paper, we address the Orienteering problem (OP) by the unsupervised learning of the self-organizing map (SOM). We propose to solve the OP with a new algorithm based on SOM for the Traveling salesman problem (TSP). Both problems are similar in finding a tour visiting the given locations; however, the OP stands to determine the most valuable tour that maximizes the rewards collected by visiting a subset of the locations while keeping the tour length under the specified travel budget. The proposed stochastic search algorithm is based on unsupervised learning of SOM and it constructs a feasible solution during each learning epoch. The reported results support feasibility of the proposed idea and show the performance is competitive with existing heuristics. Moreover, the key advantage of the proposed SOM-based approach is the ability to address the generalized OP with Neighborhoods, where rewards can be collected by traveling anywhere within the neighborhood of the locations. This problem generalization better fits data collection missions with wireless data transmission and it allows to save unnecessary travel costs to visit the given locations.**

I. Introduction

The orienteering problem (OP) originates from the orienteering outdoor sport in which individual players aim to visit as many control locations as possible within the given time budget [1]. In this paper, we consider the OP as a problem formulation for robotic information gathering and data collection tasks. The motivation of the addressed problem is to collect data from sensor stations by a mobile robot that is requested to visit sensor locations and retrieve data from them by a cost efficient path [2], [3].

The data collection planning is highly related to the Traveling Salesman Problem (TSP), which stands to determine the closed shortest tour visiting all the given locations. However, a more suitable extension of the TSP is called the Prize-Collecting Traveling Salesman Problem (PC-TSP) in which each location has an associated prize (reward) that might be collected and each prize has an associated penalty if it is not collected [4]. Therefore, the problem is to find a tour with the minimal total cost that is computed as the sum of the tour length (cost) in addition to the sum of penalties of all locations that are not visited by the tour. Moreover, considering a wireless data transmission to retrieve data from the sensors, an additional travel cost can be saved by data collection within the communication range and thus, it is not necessary the robot

978-1-5090-1897-0/16/$31.00 ©2016 IEEE



(a) T_{max}=50, R=190 (b) T_{max}=75, R=270 (c) T_{max}=50, R=270

Fig. 1. Examples of the found solutions of the Orienteering problem called Tsiligirides Set 1 and its generalization to the OP with Neighborhoods for a different limited budget T_{max}. The small disks represent particular locations to be visited and its color denotes the associated reward value (high values are in red, while low values are in blue). If it is allowed to retrieve data from the sensor with communication range δ=1.5 (**c**), the same reward R=270 as in (**b**) can be achieved with a lower travel budget, i.e., T_{max}=50. The yellow disks represent a communication range with the δ radius within which data from the particular sensors can be read along the determined tour.

travels exactly to the sensor location. This can be formulated as the PC-TSP with Neighborhoods (PC-TSPN) [5]. Although the PC-TSPN has been considered as a suitable problem formulation for periodic data collection missions [6], [7], its main disadvantage is that the two mutually conflicting and possibly incomparable goals: (1) to minimize the tour length; (2) and to minimize the sum of penalties (maximize the rewards); are combined in a single cost function.

Since the operational time of a mobile robot is usually limited, e.g., due to fuel constraints, the maximal length of the data collection tour can be constrained by a travel budget that will allow the mobile robot to safely return to some specified home location. Then, the problem is to maximize the amount of the collected information measured by a utility reward function associated to each particular location. This can be captured by the aforementioned Orienteering Problem (OP). On the other hand, the standard OP does not consider a non-zero communication range to retrieve data from the sensors. Even though several algorithms for variants of the OP have been proposed [8], to the best of our knowledge, there is not an algorithm for a variant of the Orienteering Problem with Neighborhoods (OPN). Therefore, the unifying approach for the PC-TSPN [7] based on self-organizing map for the TSP motivates us to study principles of SOM in the context of the OP with the aim to develop an algorithm for the OPN.

In this paper, we consider the unsupervised learning of SOM for the PC-TSPN [7] as a stochastic search method to determine a solution of the OP. Based on this idea, we propose a novel SOM-based algorithm to solve the OP and our initial results indicate the proposed algorithm is competitive with existing heuristics [1], [9]. Moreover, we propose an extension of the proposed algorithm to address the generalized Orienteering Problem with Neighborhoods (OPN) in which it is allowed to retrieve data within the given non-zero communication range and thus utilizes the remaining travel budget to collect more rewards, as illustrated in Fig. 1.

The paper is organized as follows. An overview of existing approaches is in the next section. The addressed problems are formally introduced in Section III. The proposed SOM-based algorithm is described in Section IV and evaluation results are presented in Section V. Concluding remarks and comments on future work are dedicated to the conclusion in Section VI.

II. Related Work

The Orienteering problem (OP) is being studied for over three decades since the first approaches were proposed by Tsiligirides in 1984. In [10], he proposes one stochastic and one deterministic heuristic to solve the OP. The stochastic algorithm uses the Monte Carlo method for generating many feasible solutions. The locations for insertion into a currently generated path are selected randomly with probability based on their reward and additional distance necessary to visit them. The deterministic algorithm uses the vehicle routing algorithm by Wren and Holliday (1972) [11]. In both algorithms, Tsiligirides utilizes a post-processing using 2-Opt heuristic [12] and possible deletion/insertion of additional locations. Beside the two proposed heuristics, Tsiligirides also created three benchmark problems with 42 test instances.

Ramesh et al. (1991) [1] suggested their four-phase heuristic for the OP. In the first insertion phase with a relaxed budget constraint, the locations are added to the tour while maximizing the prize per additional time required for reaching the location. The second phase consists of tour optimization using 2-Opt and 3-Opt mechanisms. In the third deletion phase, the vertex with maximal cost per reward is removed from the path. These three phases are iteratively repeated while gradually decreasing the relaxed budget constraint. Finally, the maximal insertion phase tries to insert as many locations as possible without violating the budget constraint using the locations with the highest reward first.

Pillai (1992) [13] proposed an exact procedure for solving the traveling salesman subset-tour problem with one additional constraint (TSSP + 1) based on branching and cutting plane method. In fact, the OP is a special case of the TSSP + 1 where the additional constraint is the travel budget T_{max}.

A heuristic by Chao et al. (1996) [9] considers locations within an ellipse with a major axis length equal to the travel budget T_{max} and with foci placed in the start and end locations. The five steps algorithm consists of *initialization*, *two-point exchange*, *one point movement*, *clean up*, and *reinitialization*. In the *initialization* phase, several paths are created using the cheapest insertion method with one mandatory location far from foci for each path. The path with the highest reward is denoted as $path_{op}$ and other locations are assigned to a set of feasible paths $path_{nop}$. In the *two-point exchange* step, the locations are exchanged between paths in $path_{nop}$ and $path_{op}$ in the cheapest way. During the *one point movement*, the algorithm moves one location between paths in a greedy way preserving all feasible paths. The *clean up* step uses the 2-Opt heuristic for shortening the $path_{op}$. The last step restarts the algorithm with several locations with the minimal reward per insertion cost removed from the $path_{op}$. Chao et al. also generated two benchmark problems, the first with 26 test instances with square-shaped locations and the second with 16 instances with diamond-shaped locations.

The aforementioned approaches for the OP represent an overview of the existing methods, where most of them rely on heuristic approaches [8]. Based on the literature review, it seems that none of the existing approaches address the considered generalization of the OP with Neighborhoods. Since considering neighborhoods has direct practical motivation in robotic information gathering and data collection missions [14], we propose to develop a new SOM-based approach that is inspired by the successful deployment of SOM in the related Prize-Collecting Traveling Salesman Problems with Neighborhoods (PC-TSPN) [7]. However, SOM has not been utilized for the OP and its first application to the routing problems with rewards (penalties) have been proposed in [15] and with limited travel budget for a multi-robot team in [14]. Regarding this and available heuristic solvers for the OP, we do not directly aim to develop a new SOM-based algorithm that will provide better results in OP instances than the existing solvers. The primary goal of this paper is to show feasibility of the SOM-based solution to the OP and its extension for the OPN. In this paper, we report on the results of the proposed approach and its comparison with existing algorithms [1], [13], [9] and [16] in the benchmark problems created by Tsiligirides (Table II, III, and IV) and by Chao et al. (Table V and VI).

III. Problem Statement

The addressed problem is motivated by data collection missions where a robotic vehicle is requested to maximize the reward collected by retrieving data from a collection of sensors while the total travel cost is below the given travel budget. This problem can be formulated as the Orienteering problem (OP), which is formally introduced as follows.

A. Orienteering Problem

We assume that n sensors are located in $\mathbb{R}^2$ and their locations are $S = \{s_1, \ldots, s_n\}$, $s_i \in \mathbb{R}^2$. Each sensor s_i has an associated score ς_i characterizing the reward if data from the sensor are collected. The vehicle is operating in $\mathbb{R}^2$ and the travel cost between any two points $p_1, p_2 \in \mathbb{R}^2$ is the Euclidean distance $|(p_1, p_2)|$. Similarly to [1], we assume the initial and final locations of the vehicle's tour are prescribed, i.e., they are represented as locations s_1 and s_n with the zero reward values $\varsigma_1 = \varsigma_n = 0$. [1] The given travel budget is T_{max} and the problem is to select a subset of k locations $S_k \subseteq S$ that maximizes the sum of rewards while the travel cost to visit them is below T_{max}. Here, it is worth mentioning that the number of k is not known a priori and it is a part of the solution. The problem can be formally defined as follows.

Let $\Sigma = (\sigma_1, \ldots, \sigma_k)$ be a permutation of k sensor labels, $1 \le \sigma_i \le n$ and $\sigma_i \ne \sigma_j$ for $i \ne j$. Σ represents a tour $T = (s_{\sigma_1}, \ldots, s_{\sigma_k})$ visiting the selected sensors and for the prescribed start and end points of the tour, $\sigma_1 = 1$ and $\sigma_k = n$. The Orienteering problem is to determine the number of sensors k, the subset of sensors S_k, and their sequence Σ such that:

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

Notice, the OP combines the problem of determining the most valuable locations S_k, which is similar to the Knapsack problem, with finding the shortest tour T visiting the locations S_k, which is similar to the TSP. It is known the OP is NP-hard [8], since for $s_1 = s_n$ finding the shortest tour connecting S_k is the Traveling salesman problem (TSP).

B. Orienteering Problem with Neighborhoods

The standard OP is a combinatorial problem to determine a subset of k locations S_k that maximizes the sum of rewards and the vehicle travels to the particular locations $s_i \in S_k$. If data from a sensor can be read within a communication range δ, it is not necessary to travel to the particular sensor location. The range δ can be used to form a disk shaped neighborhood around each sensor. Then, the problem becomes not only to select the subset S_k, but also to determine a particular point $p_{\sigma_i} \in \mathbb{R}^2$ for each selected s_{σ_i} such that the data from s_{σ_i} can be read at p_{σ_i} and the length of the path $P_k = (p_{\sigma_1}, \ldots, p_{\sigma_k})$ is equal or shorter than T_{max}. This problem is called the Orienteering Problem with Neighborhoods (OPN) and it can be defined as follows:

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

[1]The locations s_1 and s_n may be different ($s_1 \ne s_n$) and thus, the data collection path may not be a closed tour as in the TSP.

In the introduced OPN, we need to determine a particular point from which data from each sensor can be read. Such a location can be arbitrarily located in the neighborhood of each sensor. In some problem instances, the communication disks for multiple sensors may overlap, and therefore, the data from these sensors can be retrieved from a single point. The proposed SOM-based algorithm is able to provide a solution of the OP and also the OPN. It is empirical evaluation in selected instances of the OPN is reported in Section V.

IV. Proposed Self-Organizing Map for the OP

The proposed Self-Organizing Map (SOM)-based approach follows the principles of SOM for the TSP [17], [18], [19] utilized in the SOM approach for the PC-TSPN [7]. SOM for the TSP is a two layered neural network that maps the input 2D space of the sensor locations into a discrete number of output units. The first layer represents the input for presenting the sensor locations S. The second layer is organized into an array of m output units (neurons) $(\nu_1, \ldots, \nu_m)$ and the neuron weights represent a discrete point in $\mathbb{R}^2$ and thus, the connected neurons represent a path (ring of neurons) in $\mathbb{R}^2$.

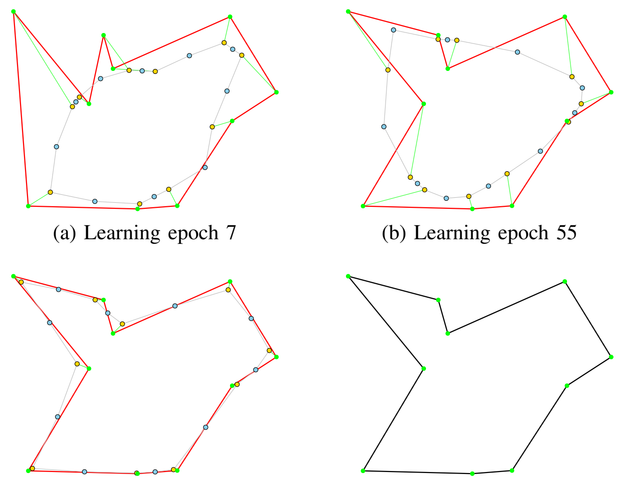
In SOM for the TSP, the sensor locations are iteratively presented to the network in a random order [18] and the neurons compete to be a winner for the currently presented sensor $s \in S$. Then, the winner neuron together with its neighboring nodes are adapted towards the presented sensor according to the neighboring function $f(G, d)$ [18]:

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

where G is the learning gain. During a single learning epoch, all sensor locations are presented to the network and a particular neuron can be a winner only once per each learning epoch. Therefore, each sensor has associated a unique winner and the ring of winners provides a sequence of visits to the sensors. Thus, a solution of the TSP is available after each learning epoch, see Fig. 2.

For the PC-TSP, the SOM adaptation procedure for the TSP has been modified to avoid adaptation to sensor locations for which the cost of the travel to the location is higher than the penalty for not visiting the location [15]. Besides, there is not an explicit constraint on the total travel cost in variants of the TSP, since the problem is to find the shortest tour possible. On the other hand, in the OP, we need to satisfy the travel budget T_{max} while optimizing the selection of the sensors that provide the highest reward. Therefore, we consider ideas of SOM for the TSP and propose to construct a solution of the OP during a single learning epoch. The unsupervised learning of SOM is used as a stochastic search algorithm to determine a subset $S_k \subseteq S$ of sensors and an order to visit them.

Contrary to SOM for the PC-TSP [15] we propose to adapt the network more often to sensors with high rewards during a single epoch. For simplicity and without loss of generality, we assume the rewards are integer values for which the greatest common divisor g can be determined. Then, each sensor s_i is replicated ς_i / g times and such sensors are used for adaptation.



(a) Learning epoch 7 (b) Learning epoch 55

(c) Learning epoch 87 (d) Final solution

Fig. 2. Evolution of the ring of neurons in SOM for the TSP and the associated solution of the TSP after a particular learning epoch. The locations to be visited are shown as the green disks. The ring of neurons is connected by a gray tiny straight line segments, the neurons are blue and yellow disks. The winner neurons are small yellow disks that are connected to the associated sensors by a green straight line segment. The tour visiting the sensor locations represented by the current ring is shown in red and it is a valid TSP solution after each learning epoch. If the network converges to a stable solution, the winners match the sensors and the ring itself becomes a solution of the TSP.

A selection of unique winners for the particular sensors during the learning epoch is utilized to construct a solution of the OP. The order of the winners in the output layer prescribes the tour T_{win} over the selected sensors, and therefore, the tour T_{win} is incrementally constructed during the learning epoch and the winner for the particular sensor is selected only if the length of T_{win} does not exceed the travel budget T_{max}. Considering the length of the tour during the learning allows to satisfy the travel budget T_{max} and the procedure provides a feasible solution of the OP after each learning epoch. However, since the tour is incrementally constructed in each learning epoch, the network does not converge to a stable solution with increasing learning epoch as in the TSP. The procedure is therefore more like a stochastic search to select the most suitable sensors and learning is terminated after a finite number of learning epochs i_{max}. The proposed SOM-based algorithm for the OP is summarized as follows:

1) *Initialization*: For n locations $s_i \in S$, replicate each location ς_i/g times, where g is the greatest common divisor of all non-zero rewards ς_i. The set of such locations is denoted as S'. Create two neurons located at s_1 and s_n. Set the learning gain G to $G = 10$, the learning rate $\mu = 0.6$, and the gain decreasing rate $\alpha = 0.1$. Initialize the best found solution as a sequence of sensor locations $T = \emptyset$ with the sum of the rewards $R = 0$. Set the current epoch counter i to $i = 1$.
2) *Randomizing*: Create a random permutation of the locations $\Pi(S')$ to avoid local minima [18].

▷ **Learning epoch:**

3) Let the current number of neurons be m. Adapt the first neuron ν_1 to s_1 and the last neuron ν_m to s_n to respect the initial and final locations of the tour. The same procedure based on (3) as in Step 4c is used. $\Pi(S') \leftarrow \Pi(S') \setminus \{s_1, s_n\}$ and mark ν_1 and ν_m as the first winners with the associated sensor locations s_1 and s_n, respectively.
4) For each $s' \in \Pi(S')$:
   a) *Budget constraint*: Consider neurons of the output layer as a path $\mathcal{P}$ in $\mathbb{R}^2$ connecting the neurons and determine the closets point $p_{s'}$ of $\mathcal{P}$ to the presented s'. Determine the first previous ν_p and the first next ν_n winner neurons from p in the output layer of the network for the current epoch i. The current sensor s' is removed from $\Pi(S')$, i.e., $\Pi(S') \leftarrow \Pi(S') \setminus \{s'\}$. Let the length of the tour represented by the current winners T_{win} be $\mathcal{L}(T_{win})$. The adaptation continues `If`
$$\mathcal{L}(T_{win}) - |(s_{\nu_p}, s_{\nu_n})| + |(s_{\nu_p}, s')| + |(s', s_{\nu_n})| \le T_{max},$$
   where s_{ν_p} and s_{ν_n} are the associated sensors to the winners ν_p and ν_n, respectively. `Otherwise` the procedure continues with the next sensor (Step 4).
   b) *Determine the winner neuron* ν^* for the point $p_{s'}$, i.e., the neuron with the identical weights, or create a new one if such does not exist or it has been already marked as a winner during the current epoch i. Associate ν^* with s'.
   c) *Adapt* the winner ν^* and its neighbors to s' using $f(G, d)$ according to $\nu' = \nu + \mu f(G, d)(s' - \nu)$.
5) *Preserve winners*: Remove all non-winner neurons.
6) *Update learning parameters*: $G \leftarrow G(1-i\alpha)$, $i \leftarrow i+1$.
7) *Update the best found solution*: `If` the current winners represent a solution with higher rewards, update R and T according to the sensors associated to the winners.
8) *Termination condition:* `If` $i \ge i_{max}$ `Stop` the learning; `Otherwise` go to Step 2.
9) *Return* the best found tour T with the total rewards R.

Notice, in the proposed SOM for the OP, we do not consider a closed ring of neurons, and therefore, the first neuron ν_1 is always adapted towards s_1 and the last neuron in the output array of neurons is always adapted towards s_n without competition with other neurons.

The complexity of a single learning epoch depends on the number of neurons, which is proportional to the number of sensors n. For each presented sensor to the network, up to m neurons is evaluated to find the winner neuron. Since $m \le 2n$ and only up to $2 \cdot 0.2m$ neurons can be adapted, the complexity of a single learning epoch can be bounded by $O(n'n)$ where n' is the number of sensors in S' which depends on the maximal reward ς_{max} and the greatest common divisor g. The number n' is always less than $\varsigma_{max}n$, and therefore, the time complexity of the learning epoch can be bounded by $O(\varsigma_{max}n^2)$. The considered problems have less than one hundred sensor locations and thus, real-time requirements for the adaptation are a fraction of a millisecond using a standard single core of a 3 GHz CPU. Therefore, $i_{max} = 500$ has been considered to obtain the results presented in this paper.

A. SOM for the Orienteering Problem with Neighborhoods

A flexibility of the SOM adaptation allows to extend the proposed algorithm for the OP to consider a communication range δ and thus, solve the OPN. The modification is relatively straightforward by considering an alternate location towards which the network is adapted, similarly as it has been utilized in [20]. The location is determined as follows.

During the determination of the point $p_{s'}$ in Step 4a, an intersection point p' of the straight line segment $(p_{s'}, s')$ with the δ-radius circle centered at s' is determined. If p' exists, the procedure continues as for the OP, but p' is utilized for the adaptation instead of s'. If $(p_{s'}, s')$ does not intersect the circle, p' is within the communication range from s', and therefore, p' is used to determine the winner for s', but the network is not adapted to s' since data can be collected from such a winner. Finally, the solution is constructed from the alternate locations p' and not from the sensor locations S' as in the OP.

V. Results

The proposed SOM-based algorithm has been firstly compared with existing heuristics on the available instances of the OP [21] for which solutions of other algorithms can be found in literature. In particular, we consider the problems proposed by Tsiligirides [10] and diamond-shaped and square-shaped problems proposed by Chao et al. [9] with the particular travel budget T_{max}. The quality of solutions is measured according to the sum of the collected rewards that is denoted by R. We consider available results for heuristic algorithms and one of the algorithms has been implemented to obtain additional results for Chao's problems [9]. The used abbreviations of the problems and algorithms are listed in Table I.

TABLE I
Abbreviations for the OP instances and algorithms

Set 1, Set 2, Set 3	Instances of the Tsiligirides problem sets [10], [21]
Set 64	Diamond-shaped test problems [9], [21]
Set 66	Square-shaped test problems [9], [21]
RB	4-phase heuristic algorithm proposed in [1]
RB†	Our implementation of the RB [1]
PL	Results for the method proposed by Pillai in [13]
CGW	Heuristic algorithm proposed in [9]
GLS	Guided local search algorithm proposed in [22]

The proposed SOM-based algorithm is a stochastic procedure, and therefore, each problem is solved 20 times and the best found solution is considered as the final solution of the problem. We also consider the average value of the rewards R_{avg} and its standard deviation R_{std} to study how much the solution might vary.

Comparison results for the Tsiligirides problems are listed in Table II, III, and IV. The best found solutions are highlighted in bold and in almost all cases all algorithms, including the proposed SOM-based algorithm, achieve the same performance. For the diamond and square shaped problems results are shown in Table V and Table VI. In these problems, the proposed algorithm achieves slightly poorer performance, except for Set 64 with T_{max}=80. Although the proposed SOM algorithm does not provide better results than the existing

TABLE II
Comparision of results on problem Set 1

T_{max}	**RB**	**RB†**	**PL**	**CGW**	**SOM-based**		
	R	R	R	R	R	R_{avg}	R_{std}
5	**10**	**10**	**10**	**10**	**10**	10	0.0
10	**15**	**15**	**15**	**15**	**15**	15	0.0
15	**45**	**45**	**45**	**45**	**45**	45	0.0
20	**65**	**65**	**65**	**65**	**65**	65	0.0
25	**90**	**90**	**90**	**90**	**90**	89	1.8
30	**110**	**110**	**110**	**110**	**110**	109	2.0
35	**135**	130	**135**	**135**	**135**	130	4.4
40	150	**155**	**155**	**155**	150	146	4.1
46	**175**	170	**175**	**175**	170	166	3.3
50	180	**190**	**190**	**190**	**190**	180	4.0
55	**205**	200	**205**	**205**	**205**	198	3.3
60	**225**	220	**225**	**225**	**225**	220	3.5
65	**240**	**240**	**240**	**240**	**240**	236	7.8
70	**260**	**260**	**260**	**260**	**260**	252	10.2
73	265	265	265	265	**270**	256	8.4
75	**275**	270	270	270	270	262	6.1
80	**280**	**280**	**280**	**280**	**280**	270	3.5
85	**285**	**285**	280	**285**	**285**	282	3.3

TABLE III
Comparision of results on problem Set 2

T_{max}	**RB**	**RB†**	**PL**	**CGW**	**SOM-based**		
	R	R	R	R	R	R_{avg}	R_{std}
15	**120**	**120**	**120**	**120**	**120**	120	0.0
20	**200**	**200**	**200**	**200**	**200**	200	0.0
23	**210**	200	**210**	**210**	**210**	210	0.0
25	**230**	**230**	**230**	**230**	**230**	230	0.0
27	**230**	215	**230**	**230**	**230**	230	0.0
30	260	265	**275**	265	265	264	1.5
32	**300**	**300**	**300**	**300**	**300**	293	7.7
35	**320**	305	**320**	**320**	**320**	319	5.4
38	**385**	350	360	360	360	352	4.3
40	**395**	**395**	**395**	**395**	**395**	379	8.0
45	**450**	**450**	**450**	**450**	**450**	441	3.0

heuristics, its main advantage is the ability to solve the generalized Orienteering problem with neighborhoods.

A. Results for the Orienteering Problem with Neighborhoods

The feasibility of the proposed SOM-based algorithm to solve the OPN has been studied in three OP instances where the SOM algorithm provides worse solutions than the existing heuristics. The communication range δ has been selected from the set $\delta \in \{0, 0.2, 0.5, 0.7, 1.0, 1.2, 1.5, 1.7, 2.0\}$ and the problems have been solved by the proposed SOM-based algorithm for the OPN. Average values of the total collected rewards from 20 trials are depicted in Fig. 3.

Based on mutual distances between the locations, a non-zero communication range allows to collect more rewards and thus, utilize the given travel budget more efficiently. Selected best found solutions are depicted in Fig. 4.

Even in the considered OP instances, the results support the idea of the improved performance in collecting rewards with non-zero communication radius and lower travel budget than for a higher budget and δ=0. Notice, multiple sensors are covered from a single path vertex and also sensors within

TABLE IV
COMPARISION OF RESULTS ON PROBLEM SET 3

T_{max}	RB R	RB† R	PL R	CGW R	SOM-based R	R_{avg}	R_{std}
15	**170**	**170**	**170**	**170**	**170**	170	0.0
20	**200**	190	**200**	**200**	**200**	192	3.6
25	**260**	**260**	**260**	**260**	**260**	256	5.0
30	**320**	**320**	**320**	**320**	**320**	314	5.8
35	**390**	**390**	**390**	**390**	**390**	368	11.2
40	**430**	**430**	**430**	**430**	**430**	418	8.9
45	**470**	**470**	**470**	**470**	**470**	451	10.4
50	**520**	**520**	**520**	**520**	510	493	11.4
55	**550**	**550**	**550**	**550**	**550**	536	9.7
60	**580**	**580**	**580**	**580**	**580**	571	10.4
65	**610**	**610**	**610**	**610**	**610**	600	8.4
70	**640**	**640**	**640**	**640**	**640**	630	4.5
75	**670**	**670**	**670**	**670**	**670**	666	4.8
80	**710**	700	**710**	**710**	**710**	703	5.6
85	**740**	730	**740**	**740**	**740**	734	4.9
90	**770**	750	**770**	**770**	**770**	770	0.0
95	**790**	**790**	**790**	**790**	**790**	789	3.0
100	**800**	**800**	**800**	**800**	**800**	800	2.2
105	**800**	**800**	**800**	**800**	**800**	800	0.0
110	**800**	**800**	**800**	**800**	**800**	800	0.0

TABLE V
COMPARISION OF RESULTS ON SET 64 (DIAMOND-SHAPED PROBLEMS)

T_{max}	RB† R	CGW R	GLS R	SOM-based R	R_{avg}	R_{std}
15	**96**	**96**	**96**	**96**	96	0.0
20	**294**	**294**	**294**	**294**	294	0.0
25	384	**390**	**390**	384	363	8.8
30	456	**474**	**474**	444	428	9.2
35	546	**570**	552	552	538	11.6
40	672	**714**	702	690	654	18.8
45	762	**816**	780	750	727	14.9
50	840	**900**	888	840	802	14.6
55	942	**984**	972	942	878	25.4
60	1050	1044	**1062**	1026	960	21.3
65	1110	**1116**	1110	1080	1050	16.0
70	1182	1176	**1188**	1158	1127	17.1
75	1200	1224	**1236**	1230	1197	17.5
80	1248	1272	1260	**1278**	1261	11.8

the δ distances from the found path are covered. Hence, we can expect better performance in data collection missions formulated as the OPN rather than OP.

VI. CONCLUSION

In this paper, we propose a novel SOM-based algorithm for the Orienteering problem. The presented early results support feasibility of the proposed approach, where it provides competitive solutions to the existing 4-phase and the other comparison algorithms. Although the proposed algorithm does not provide better results for the all evaluated problems, its main benefit is the ability to solve the generalized OP with non-zero communication range, denoted as the Orienteering problem with neighborhoods (OPN).

The proposed algorithm is based on the unsupervised learning of SOM for the TSP in which SOM converges to a stable solution representing the final tour. However, in the proposed algorithm for the OP, the neural network does not converge to a stable solution and the SOM principles are employed in a random search fashion. For the more general OPN, the solution tends to be more stable, but stability is not guaranteed, albeit a feasible solution is available after each learning epoch. The presented approach is a groundwork for further investigation of the convergence conditions of SOM for the OP and OPN. Besides, we also aim to address variants of the OP where it is necessary to address time needed to retrieve data from the particular sensor and the OP with time-windows.

TABLE VI
COMPARISION OF RESULTS ON SET 66 (SQUARE-SHAPED PROBLEMS)

T_{max}	RB† R	CGW R	GLS R	SOM-based R	R_{avg}	R_{std}
15	90	**120**	100	**120**	120	0.0
20	**205**	195	190	**205**	204	3.6
25	270	**290**	**290**	**290**	288	4.0
30	370	**400**	**400**	**400**	370	19.9
35	**465**	460	460	445	426	14.7
40	535	**575**	**575**	545	509	18.6
45	615	**650**	645	615	571	24.1
50	**730**	**730**	**730**	**730**	660	29.7
55	815	**825**	820	770	733	20.0
60	895	**915**	**915**	845	813	21.0
65	**980**	**980**	**980**	960	901	34.9
70	1060	**1070**	**1070**	1030	965	32.1
75	1110	**1140**	**1140**	1105	1037	31.6
80	1175	**1215**	**1215**	1145	1109	19.7
85	1230	**1270**	1265	1210	1172	19.6
90	1330	**1340**	**1340**	1270	1234	20.4
95	**1395**	1380	1390	1335	1293	19.5
100	**1465**	1435	1455	1395	1369	15.4
105	**1520**	1510	1515	1475	1430	28.9
110	**1550**	**1550**	**1550**	1510	1489	14.0
115	**1595**	**1595**	1590	1580	1551	18.6
120	**1635**	**1635**	**1635**	**1635**	1614	20.7
125	**1670**	1655	1655	**1670**	1654	8.5
130	**1680**	**1680**	1670	**1680**	1676	3.8

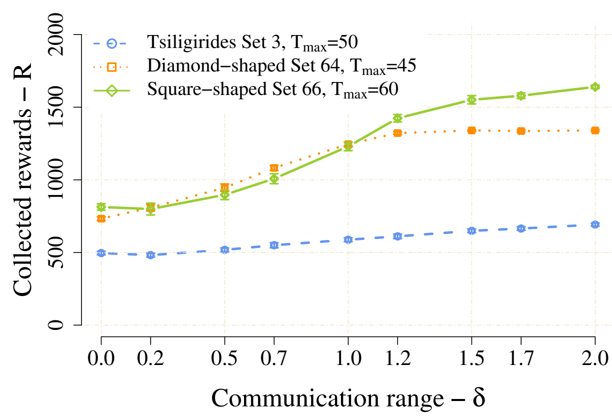
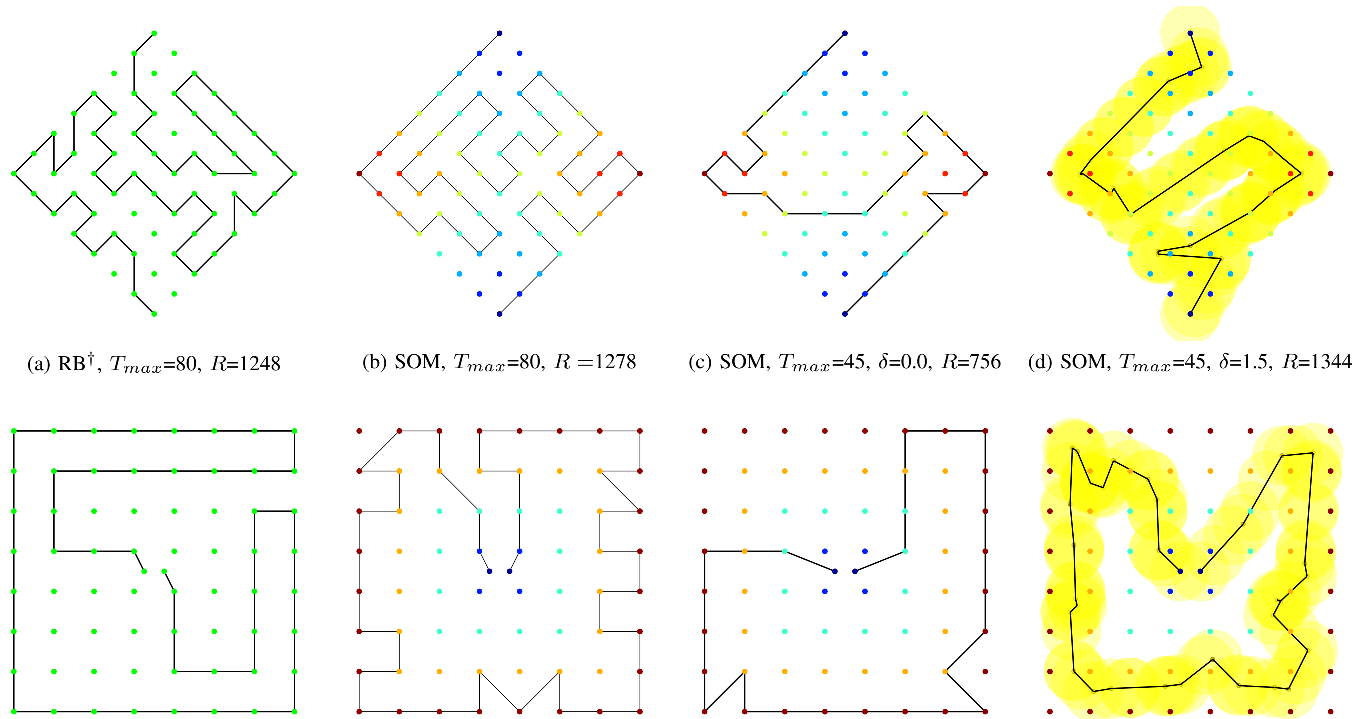


Fig. 3. Average sum of the collected rewards with increasing communication range δ. Standard deviations are shown as error bars, but are very small.

Acknowledgments – The presented work was supported by the Czech Science Foundation (GAČR) under research project No. 16-24206S. The work of Robert Pěnička was supported by the Grant Agency of the Czech Technical University in Prague,



(a) RB†, T_{max}=80, R=1248 (b) SOM, T_{max}=80, R =1278 (c) SOM, T_{max}=45, δ=0.0, R=756 (d) SOM, T_{max}=45, δ=1.5, R=1344

(e) RB†, T_{max}=95, R=1395 (f) SOM, T_{max}=95, R=1335 (g) SOM, T_{max}=60, δ=0.0, R=845 (h) SOM, T_{max}=60, δ=1.5, R=1600

Fig. 4. Found solutions for the diamond-shaped problem Set 64 (upper row) and square-shaped problem Set 66 (bottom row) found by the RB† and the proposed SOM algorithm for $\delta = 0.0$ and T_{max}. The effect of non-zero communication range is shown in (**d**) and (**h**) for a lower travel budget $T_{max} = 45$ (top) and $T_{max} = 60$ (bottom), where $\delta = 1.5$ allows to collect rewards higher than for $T_{max} = 80$ and $T_{max} = 95$, respectively. The superimposed yellow disks represent the communication radius and covered sensors.

grant No. SGS15/157/13. The work of Graeme Best was supported by the Australian Centre for Field Robotics, the New South Wales Government, and the Australian Research Council's Discovery Project funding scheme (No. DP140104203).

REFERENCES

[1] 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.

[2] R. N. Smith, J. Das, H. K. Heidarsson, A. A. Pereira, F. Arrichiello, I. Cetinic, L. Darjany, M.-E. Garneau, M.-D. Howard, C. Oberg, M. Ragan, E. Seubert, E. C. Smith, B. Stauffer, A. Schnetzer, G. Toro-Farmer, D. A. Caron, B. H. Jones, and G. S. Sukhatme, "USC CINAPS builds bridges: Observing and monitoring the Southern California Bight," *IEEE Robotics and Automation Mag.*, vol. 17, no. 1, pp. 20–30, Mar 2010.

[3] M. Bryson, A. Reid, F. Ramos, and S. Sukkarieh, "Airborne vision-based mapping and classification of large farmland environments," *Journal of Field Robotics*, vol. 27, no. 5, pp. 632–655, 2010.

[4] E. Balas, "The prize collecting traveling salesman problems," *Networks*, vol. 19, pp. 621–636, 1989.

[5] G. Hollinger, S. Choudhary, P. Qarabaqi, C. Murphy, U. Mitra, G. Sukhatme, M. Stojanovic, H. Singh, and F. Hover, "Underwater data collection using robotic sensor networks," *IEEE J. Selected Areas in Communications*, vol. 30, no. 5, pp. 899–911, 2012.

[6] G. Hollinger, U. Mitra, and G. Sukhatme, "Autonomous data collection from underwater sensor networks using acoustic communication," in *IROS*, 2011, pp. 3564–3570.

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

[8] 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.

[9] 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.

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

[11] A. H. Anthony Wren, "Computer scheduling of vehicles from one or more depots to a number of delivery points," *Operational Research Quarterly (1970-1977)*, vol. 23, no. 3, pp. 333–344, 1972.

[12] G. A. Croes, "A method for solving traveling-salesman problems," *Operations Research*, vol. 6, no. 6, pp. 791–812, 1958.

[13] 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.

[14] G. Best, J. Faigl, and R. Fitch, "Multi-robot path planning for budgeted active perception with self-organising maps," in *IROS*, 2016, (*to appear*).

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

[16] 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.

[17] T. Kohonen, M. R. Schroeder, and T. S. Huang, Eds., *Self-Organizing Maps*, 3rd ed. Springer-Verlag New York, Inc., 2001.

[18] S. Somhom, A. Modares, and T. Enkawa, "A self-organising model for the travelling salesman problem," *Journal of the Operational Research Society*, vol. 48, no. 9, pp. 919–928, 1997.

[19] J. Faigl, M. Kulich, V. Vonásek, and L. Přeučil, "An application of self-organizing map in the non-euclidean traveling salesman problem," *Neurocomputing*, vol. 74, no. 5, pp. 671–679, 2011.

[20] J. Faigl, "Approximate Solution of the Multiple Watchman Routes Problem with Restricted Visibility Range," *IEEE Transactions on Neural Networks*, vol. 21, no. 10, pp. 1668–1679, 2010.

[21] "The Orienteering Problem: Test Instances – Department of Mechanical Engineering," citied on 2016-04-09. [Online]. Available: http://www.mech.kuleuven.be/en/cib/op#section-0

[22] P. Vansteenwegen, W. Souffriau, G. V. Berghe, and D. V. Oudheusden, "A guided local search metaheuristic for the team orienteering problem," *European Journal of Operational Research*, vol. 196, no. 1, pp. 118–127, 2009.