Autonomous Data Collection using a Self-Organizing Map

Jan Faigl, *Member, IEEE,* Geoffrey A. Hollinger, *Member, IEEE*

***Abstract*—The self-organizing map is an unsupervised learning technique providing a transformation of a high dimensional input space into a lower dimensional output space. In this paper, we utilize the self-organizing map for the traveling salesman problem to develop a solution to autonomous data collection. Autonomous data collection requires gathering data from pre-deployed sensors by moving within a limited communication radius. We propose a new growing self-organizing map that adapts the number of neurons during learning, which also allows our approach to apply in cases where some sensors can be ignored due to a lower priority. Based on a comparison with available combinatorial heuristic algorithms for relevant variants of the traveling salesman problem, the proposed approach demonstrates improved results, while also being less computationally demanding. Moreover, the proposed learning procedure can be extended to cases where particular sensors have varying communication radii, and it can also be extended to multi-vehicle planning.**

I. Introduction

The problem addressed in this paper is motivated by environmental monitoring and surveillance scenarios, where an autonomous vehicle is required to collect data from pre-deployed sensors. Examples include autonomous underwater vehicles assisting oceanographers to track harmful algae blooms [1], aerial vehicles collecting data about changing ecosystems [2], and ground vehicles monitoring volcanic activity [3]. In particular, the proposed approach aims to provide an efficient solution to autonomous data collection from sensors placed on an ocean floor [4]. A convenient method for data collection is to equip each sensor with wireless communication capability and retrieve data remotely. However, due to limited communication technology available in the underwater domain (e.g., wireless acoustic modems), a mobile underwater vehicle is needed to retrieve the data from deployed sensors [5].

Such an autonomous data collection mission to visit a set of pre-specified locations can be formulated as a variant of the *traveling salesman problem* (TSP). The TSP stands to determine a shortest tour visiting all the given locations (cities), such that each location is visited exactly once and the tour returns to the origin location. The TSP is a well studied problem in operations research and is known to be NP-hard (unless P=NP). Although efficient combinatorial heuristics [6] and probabilistic approaches [7] have been proposed for the TSP, the considered autonomous data collection problem is different from the standard TSP in two important aspects. First, it is not necessary to visit the sensors exactly. It is sufficient if the vehicle reaches the vicinity of each sensor to allow for a reliable transmission of the data from it. The second aspect is related to a lower priority of some sensors providing less important data. It may be better to ignore these sensors, rather than unnecessarily increasing the solution cost to visit them. Regarding existing variants of the TSP in literature, these two aspects of the studied data collection problem have appeared as TSP formulations called the *traveling salesman problem with neighborhoods* (TSPN) and the *prize-collecting traveling salesman problem* (PC-TSP) [8], respectively.

The PC-TSP and TSPN have been studied in the literature, and most prior approaches are based on combinatorial optimization techniques (including our own prior work [5], [9]). Existing frameworks are based on a combination of heuristics, used to determine which locations to visit, followed by a solution of the standard TSP with point cities. Specific algorithms have been proposed for particular problem variants, but the combinatorial heuristics do not address the entire spectrum of these optimization problems (TSP, TSPN, PC-TSP, and a combination of PC-TSPN) in a single unified way.

An alternative to the combinatorial optimization approach is to use a self-organizing map (SOM) to solve the TSP [10]. The SOM is able to solve problems where the locations to be visited are not explicitly prescribed (e.g., in the *watchman route problem* (WRP) [11]). A basic variant of the WRP is to find a shortest path for a mobile watchman to cover a whole environment using a sensor with limited sensing range. Solving this problem requires determining sensing locations to cover the environment and finding a shortest path connecting these locations. The main benefit of SOM-based approaches is an ability to address both sub-problems together and solve them simultaneously using unsupervised learning. New sensing locations are iteratively determined during the learning from the current route represented by the network.

The idea of SOM for the WRP has been applied to a robotic variant of the TSPN in [12], where the problem is to find a shortest path connecting a set of polygonal goals. These recent advancements of the SOM-based route planning motivate us to consider SOM-based unsupervised learning as a unifying technique to solve the entire class of the aforementioned problem formulations arising from autonomous data collection. Thus, we can bridge the gap between combinatorial heuristics that address only particular problem variants.

The presented work is supported by the Czech Science Foundation (GAČR) under research project No. 13-18316P and 15-09600Y. The support under NSF grant IIS-1317815 to Geoffrey Hollinger is also gratefully acknowledged. The support of the Fulbright commision is also gratefully acknowledged.

J. Faigl is with Department of Computer Science, Czech Technical University in Prague, Technická 2, 166 27 Prague, Czech Republic faiglj@fel.cvut.cz

G. Hollinger is with the School of Mechanical, Industrial, and Manufacturing Engineering, Oregon State University, 204 Rogers Hall, Corvallis, OR 97331 geoff.hollinger@oregonstate.edu

The main novelty of this paper is a new adaptation procedure of SOM that provides a unified architecture for solving variants of the TSP. To the best of our knowledge, the proposed solution is the first application of SOM to the TSP with both neighborhoods and prizes on the cities. Based on the presented results, the proposed SOM-based approach provides improved solutions in comparison with combinatorial heuristics in scenarios motivated by underwater data collection missions. Moreover, the SOM approach is less computationally demanding than approaches based on combinatorial solutions to the underlying TSP. The presented work is based on our early results reported in a prior conference paper [13], where we introduced the idea of considering SOM for the TSPN in data collection missions.

In the current paper, we present an improved adaptation procedure along with a detailed description of the identified parameters providing a trade-off between the solution quality and computational requirements. The particular improvements and extensions over our preliminary results are as follows:

- Improved ring regeneration technique to provide a better quality of solutions.
- Initial values of the unsupervised learning technique and their evolution during adaptation to decrease the computational time while maintaining the solution quality.
- More elaborate evaluation and comparison of the proposed technique versus existing solutions.
- Extension of the adaptation procedure to consider varying communication radii for individual sensor stations.
- Extension of the unsupervised learning to multi-vehicle missions and initial evaluation of found solutions.

The paper is organized as follows. A discussion of related work in autonomous data collection is presented in Section II to show the need for a unified architecture. Then, an overview of SOM-based routing techniques is presented. The problem statement and considered problem formulation are presented in Section III together with the used notation. In Section IV, the proposed SOM-based planner is presented. A generalization of the proposed planner to problems with varying communication radii for each sensor location and extensions to multi-vehicle missions with and without a common starting location (depot) are demonstrated in Section V. Results of a comparison with state-of-the-art combinatorial heuristic methods are presented in Section VI together with a discussion of found insights. Concluding remarks are presented in Section VII.

II. Related Work

In an autonomous data collection mission, a mobile vehicle is requested to collect data from a set of pre-deployed sensors. Hence, this mission planning problem is closely related to the TSP. In this case, wireless communication can be used to retrieve data from sensors, and the vehicle may not need to visit the location of the sensor exactly. Instead, it may be sufficient to move within a communication radius of the sensor. Moreover, it may be sufficient to move within proximity of a particular location (e.g., in view planning, coverage, or surveillance missions [14]) to collect the requested data. Finally, it can be difficult to achieve precise navigation to an exact point location with an autonomous mobile robot. Having multiple options to visit a goal may provide a solution with increased reliability [15]. Therefore, the TSP with neighborhoods (TSPN) is an important problem formulation used for planning efficient paths to visit a set of regions by a robot.

Even though approximate solutions of the restricted variants of the TSPN exist (e.g., a problem with arbitrary connected neighborhoods with similar diameters and disjoint unit disc neighborhoods [16] or disjoint convex fat neighborhoods [17]), in general, the TSPN is APX-hard [18] and cannot be approximated to within a factor $2-\epsilon$, where $\epsilon > 0$.

Another aspect of the autonomous data collection related to a practical deployment is the case where sensors provide information of different quality. This can be modeled by an association of different "prizes" (or corresponding penalty if the sensor is ignored), which leads to the Prize-Collecting TSP (PC-TSP) originally introduced by Balas in [19]. The original PC-TSP has been further extended to a number of related variants [20], which include variants of the type of required path, restrictions on the prizes, and the appearance of new locations during the tour [21].

An approximation algorithm for the PC-TSP has been proposed in [22]. The algorithm is based on an LP primal/dual scheme, which is guaranteed to provide a solution within an approximation factor of two. In [23], authors show a slightly better approximation guarantee is possible; however, at the cost of computational and implementation complexity.

The problem of planning a route for data collection missions has been studied in robotics in the context of robotic data mules. Approximation algorithms for multiple data mules collecting data from a sensor field have been proposed in [24]. In this problem, the time for communication with the sensor is considered as part of the tour. In addition, the communication radii around sensors are assumed to be uniform, fixed, and deterministic. Mobile and stationary nodes for data collection in underwater missions based on optical and acoustic communication has been presented in [25] together with description of the networking architecture and sensor specifications necessary for this type of missions.

Regarding our own prior work on robotic data collection, we proposed a combination of the PC-TSP and TSPN as a suitable problem formulation for autonomous data collection planning and a heuristic solution to this problem [5], [9]. The combinatorial solution is based on a decoupled approach, where sensing locations are determined prior finding a route as a solution of the TSP. A covering set of goals is generated according to the size of the neighborhoods, and the goals to be visited are found as a subset of the covering set by the approximation algorithm [22]. Finally, the TSP tour of this subset is found by the Concorde solver [6], [26]. In [5], we showed that this heuristic approach outperforms competing algorithms in autonomous data collection scenarios. However, the key drawbacks of this prior work are the need to use the TSP solver for the underlying routing problem and the heuristic nature of the subgoal selection. In the present paper, we propose using the unsupervised learning architecture of a self-organizing map (SOM) to eliminate these drawbacks, improve solution quality, and decrease the required computation time.

A. Self-Organizing Map for Route Planning

The self-organizing map (SOM) was originally proposed as a data visualization technique by T. Kohonen to provide a non-linear transformation (map) of a high-dimensional input space into a lower dimensional (usually 2D) discrete output space of cell lattices [27]. SOM is a two-layered artificial neural network accompanied by an adaptation procedure of the unsupervised learning that adapts neuron weights to represent the input space. The procedure is based on an iterative selection of the best matching neuron to the presented input signal and its adaptation, together with its neighboring neurons, according to the used neighbouring activation function. The adaptation provides the main feature of SOM, a preservation of topological properties of the input space in the output space. Based on this principle, SOM was successfully applied in many applications of clustering, classification, and data visualization problems [27]. In 1988, SOM was applied to solve NP-hard routing problems, in particular the TSP, independently by Angéniol [28] and Fort [29]. Since then, several approaches have been proposed to improve SOM performance in the TSP over the last decades (e.g., see overviews of approaches in [30], [31], [32], [33], [34]).

SOM for the TSP [10] is also a two layered network, but the output layer is one dimensional and forms an array of neuron weights, which is called a ring. The basic adaptation works as follows: 1. The input is a two dimensional vector for presenting 2D points that represent cities in the planar TSP; 2. The neuron weights also represent 2D points that are adapted to fit the input cities; 3. The unsupervised learning consists of iterative presentation of the cities to the network; 4. The algorithm selects the best matching neuron for each presented city and then adapts the winner and its neighbouring neurons to the city; 5. The adaptation is terminated when winners fit the cities (i.e., are negligibly close to them); 6. Then, the sequence of cities to be visited is retrieved by traversing the output layer (ring). Hence, the TSP tour can be obtained by connecting the cities (in the input space) associated to the winners.

III. Problem Statement

The studied problem of autonomous data collection is formulated as a variant of the traveling salesman problem (TSP), where data to be collected are retrieved from a pre-deployed set of n sensors. Two practical aspects of data collecting are considered in the proposed problem formulation. First, we consider importance of data following the *prize-collecting traveling salesman problem* (PC-TSP) formulation [8], where each city represents a prize that might be collected and each prize has associated penalty cost if it is not collected. This formulation leads to cases where the penalty is significantly lower than the travel cost to the sensor, which makes it more suitable to avoid visitation of the sensor by the tour. Hence, the problem is to find a cost-efficient tour collecting the most important prizes (sensor measurements). The optimal tour will have minimal total cost, computed as the sum of the tour length (cost), in addition to the sum of the penalties of all sensor measurements not collected along the tour.

The second aspect is related to the fact that data from the sensor can be retrieved using wireless communication, which can be considered reliable up to a specific communication radius ρ (e.g., based on the used technology and local conditions around each sensor station). Therefore, it is not necessary to visit the sensor exactly (only within the radius ρ from it). This aspect is formalized by the *traveling salesman problem with neighborhoods* (TSPN). In the case where data from all the sensors have to be collected and communication is reliable only for a very close distance to a sensor (i.e., $\rho = 0$), the problem becomes the standard TSP with point cities.

These two aspects are combined in a problem formulation called the *prize-collecting traveling salesman problem with neighborhoods* (PC-TSPN), which is a suitable problem formulation of the studied autonomous data collection planning. Thus, the problem is to find a cost efficient path to retrieve data from the given set of n sensor stations (not necessarily from all sensors). We formalize the problem as follows.

For simplicity, it is assumed the sensor stations are located in $\mathbb{R}^2$ with the positions $\mathbf{S} = \{s_1, \ldots, s_n\}$, $s_i \in \mathbb{R}^2$. Each sensor s_i has associated penalty $\zeta(s_i) \geq 0$ characterizing the additional cost if the data are not retrieved from s_i. A vehicle for collecting data from the sensors is operating in $\mathbb{R}^2$ with the traversal cost $c(p_1, p_2)$ defined for any two points $p_1, p_2 \in \mathbb{R}^2$. Data from a sensor station s_i can be retrieved by the vehicle located within ρ distance from s_i. Having these preliminaries, the PC-TSPN stands to find a set of unique goal locations $\mathbf{G} = \{g_1, \ldots, g_k\}$, $k \leq n$, $g_i \in \mathbb{R}^2$, and $g_i \neq g_j$ for all $g_j, g_i \in \mathbf{G}$, at which the sensor data readings are performed and to determine a cost efficient tour T to visit the locations $\mathbf{G}$ such that the total cost of the tour $\mathcal{C}(T)$ is minimal:

$$\mathcal{C}(T) = \sum_{(g_{l_i}, g_{l_{i+1}}) \in T} c(g_{l_i}, g_{l_{i+1}}) + \sum_{s \in \mathbf{S} \setminus \mathbf{S}_T} \zeta(s), \tag{1}$$

where $\mathbf{S}_T$ is the determined subset of the sensors $\mathbf{S}_T \subseteq \mathbf{S}$ from which data are collected at the goal locations $\mathbf{G}$, i.e., for each selected sensor $s \in \mathbf{S}_T$ there exists $g \in \mathbf{G}$ such that $|(s, g)| \leq \rho$. The tour T is a permutation of the determined goal locations $T = (g_{l_1}, \ldots, g_{l_{k-1}}, g_{l_k})$, where $g_{l_j} \in \mathbf{G}$, $1 \leq l_j \leq n$, and $g_{l_1} = g_{l_k}$. Notice, for a set of goal locations $\mathbf{G}$, the optimal tour T can be found as a solution of the TSP, e.g., using the Concorde solver [26]. However, the proposed SOM-based approach aims to determine $\mathbf{G}$ and T simultaneously during the unsupervised learning.

Regarding the motivational data collection scenario, the environment can be considered as planar and without obstacles. Therefore, to simplify description of the proposed approach, we consider the travel cost between two goals $c(g_i, g_j)$ as the Euclidean distance $|(g_i, g_j)|$.

Used notation and symbols are depicted in Table I, where the values of the particular parameters are recommended values used to obtain the results presented in Section VI.

A. Terminology Note

In the original formulation of the TSP, the location to be visited by the tour are called cities. In robotics, such a planning problem to visit a set of locations (cities) is called multi-goal path planning, where the goals represent particular navigational waypoints towards which the robot navigates

TABLE I
USED SYMBOLS

Symbol	Description
$\mathbb{R}^2$	Problem domain – 2D plane
ρ	Communication radius for reliable data transmission
$\mathbf{S}$	Set of sensor stations (positions)
s_i	Position of sensor station $s_i \in \mathbf{S}$
n	Number of sensor stations, $n = \lvert\mathbf{S}\rvert$
$\mathbf{G}$	Set of goal locations (positions) from which data are retrieved from the sensors
g_j	Position of the goal location $g_j \in \mathbf{G}$
k	Number of the determined goal locations, $k = \lvert\mathbf{G}\rvert$
$\zeta(s_i)$	Penalty cost for not retrieving data from the sensor s_i
T	Solution of the PC-TSPN – a sequence of goal visits (tour)
μ	Learning rate, $\mu = 0.99$
μ_t	Learning rate threshold to terminate the adaptation, μ_t=0.4
$c(p_i, p_j)$	Euclidean distance between points $p_i, p_j \in \mathbb{R}^2$
ν	Neuron weights, $\nu \in \mathbb{R}^2$
m	Current number of neurons in a ring
σ	Neighbouring function variance (learning gain)
σ_0	Initial value of neighbouring function variance, σ_0=10
α	Decreasing rate of neighbouring function variance, α=0.01
$\Pi(\mathbf{S})$	Random permutation of sensor stations
$f(\sigma, d)$	Neighbouring function, where d is a distance of the neuron being adapted from the winner in the output layer (i.e., distance in the number of neurons)
$\mathcal{C}(T)$	Cost of the solution (tour) T

during a mission execution. The problem addressed in this paper is to determine a cost efficient solution for collecting data from the given sensors, and thus we can consider sensors as the cities in the TSP. However, we are allowed to read data within a communication radius ρ, and we can imagine that data from a sensor s_i can be read from any location inside a disk with the radius ρ centered at s_i. Hence, the sensor, together with the communication radius, represent a goal region to be visited by a vehicle (similar to [35], [12]). In the addressed PC-TSPN, the problem is to determine the particular goal locations at which data from the selected sensors can be retrieved and the sequence of the goal locations.

IV. SOM-based Planner for the PC-TSPN

The proposed approach for the PC-TSPN follows the basic structure of the SOM for the TSP [10] accompanied by ideas for the TSPN [12] and PC-TSP [36]. We extend prior adaptation schema by providing a new way of determining a winner neuron and proposing an additional rule for considering the prize-collecting penalty associated to each sensor measurement. The penalty is addressed by a conditional adaptation of the winner neuron to a suitable goal location from which data can be read from the sensor if the distance (cost) of such a winner neuron to that location would be lower than the associated penalty of the sensor [36]. Therefore, prior to an actual winner selection and its adaptation, we need to determine the location of such a winner and the corresponding goal location to retrieve data from the sensor.

Due to the increased complexity of the conditional adaptation in comparison to a standard SOM, we first provide an overview of the used SOM for the TSP followed by the proposed learning parameters and their evaluation during the learning in Section IV-B. A detailed description of the proposed SOM for the PC-TSPN is presented in Section IV-C.

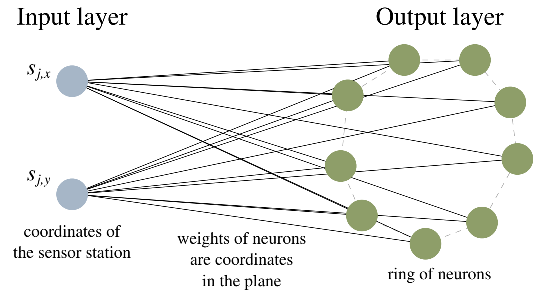


Fig. 1. Structure of the SOM for the TSP.

Comments on the computational complexity and solution quality are in Section IV-D and Section IV-E, respectively.

A. Overview of the used SOM for the TSP

The used neural network for the TSP is a two-layered network with one-dimensional output (see Fig. 1). The first layer consists of two inputs for presenting coordinates of the particular sensor in a plane, $s_j \in \mathbb{R}^2$. The neuron weights are also two dimensional coordinates in $\mathbb{R}^2$ that are adapted to particular sensor locations $s_j \in \mathbf{S}$ during the network learning. The output layer is organized into an array of output units. By connecting the sequence of the neuron weights by straight line segments, the output layer forms a ring of neurons that represents the requested data collection tour in the input space.

The learning of SOM for the TSP is an iterative process to adapt the network to the sensor locations (cities). The sensors are presented to the network in a random order to escape a local minima [30], [31], [10]), and for each sensor, the best matching neuron (further referred to as the winner neuron or just the winner) is determined according to its distance to the sensor. Then, the winner and its neighboring neurons are adapted towards the sensor. The adaptation of the neighborhoods is performed with a decreasing power defined by the neighbouring function f, which has the form $f(\sigma, d) = e^{(-d^2/\sigma^2)}$, where d is a distance of the adapted neighbouring neuron from the winner in the number of neurons in the output array. The learning is terminated after a given number of learning epochs or when each sensor has an associated winner at a distance less than a given threshold. Notice, in contrast to standard SOM for classification and other problems, here, a single learning epoch includes presentation of all sensors (cities) to the network (as in other SOM approaches for the TSP [31], [33]). This is beneficial because it makes the learning epoch counter independent of the problem size.

In a standard SOM [27], the number of neurons must be selected prior to learning, e.g., usually as 2–3 times the number of sensors (cities) for the TSP [33]. On the contrary, sensors are selected during the learning in the PC-TSPN. Therefore, it is desirable to adapt the number of neurons during the learning, which is a part of the proposed adaptation procedure.

B. Learning Parameters

The performance of the SOM adaptation depends on initialization of the network and the evolution of the learning parameters. This includes initialization of the neighbouring function variance σ, its decreasing rate α, learning rate μ, and their evolution schemata. Instead of initialization of the σ according to the number of cities [30] (e.g., used in our previous work [10], [11]), we follow the evolution schema proposed in [37] and set initial values of the parameters to $\sigma = 10$ and $\mu = 0.99$. These parameters are then updated after each learning epoch as follows:

$$\sigma_{i+1} = \sigma_i(1 - i\alpha) \tag{2}$$

and

$$\mu = \sqrt[4]{\frac{1}{i}}, \tag{3}$$

where i is the number of the current learning epoch. Notice, μ is independent of the number of neurons and sensor stations, and it is decreased after each learning epoch (presentation of all sensors). The learning rate μ is used in the adaptation, and new neuron weights ν' are set according to the distance of ν to g_s as $\nu' = |(\nu, g_s)|\mu f(\sigma, d)$. Therefore, for a very small μ and together with small $f(\sigma, d)$, the weights are not effectively changed after ten or more learning epochs. Thus, we can terminate the adaptation. In fact, we have observed the network is typically stabilized very quickly, and additional learning epochs do not improve the solution quality. Based on our experimental evaluation of different parameters, we recommend $\alpha = 0.01$ to provide a good trade-off between the solution quality and the computational burden. Therefore, we suggest terminating the adaptation when the network is stabilized, or after 40 learning epochs.[1]

C. Proposed SOM for the PC-TSPN

The main idea of the proposed SOM for the PC-TSPN is in a new winner selection based on the procedure introduced in [38]. In this procedure, a new neuron is added to the network at the closest point p of the current ring to the sensor s (see Fig. 2a). However, we need to trade off the cost to read data from s with the penalty for not collecting measurements from s in the PC-TSPN. Therefore, expected weights of such an eventual winner neuron (the point p) and the corresponding location p_s at which we can read data from s are determined first. If p and p_s satisfy the conditional adaptation for the PC-TSP proposed in [36], the coordinates of p are used as the neuron weights for the winner neuron ν^*, and p_s becomes the goal location towards which the network is adapted.

Contrary to [38] with polygonal goal regions, the point p_s is directly determined from the intersection of the straight line segment (p, s) formed from p and s with the ρ-radius disk goal region of s (see Fig. 2b). Thus, a possible location for reading data from s is a point p_s on (p, s) such that $|(p_s, s)| < \rho$.

The network is adapted to s only if the distance from p to the determined location p_s is shorter than the sensor penalty, i.e., $|(p, p_s)| \leq \zeta(s)$; otherwise both p and p_s are discarded, and the network is not adapted to s. If $|(p, p_s)| \leq \zeta(s)$ the weights of the winner neuron for s are set to the coordinates of p, with p_s as its associated goal location g_s.

[1] If a large number of learning epochs is needed, a low value of α, e.g., $\alpha = 10^{-4}$, may be used. Notice, for i_{max} learning epochs, α must be set to $\alpha < 1/i_{max}$ to keep σ_{i+1} positive.

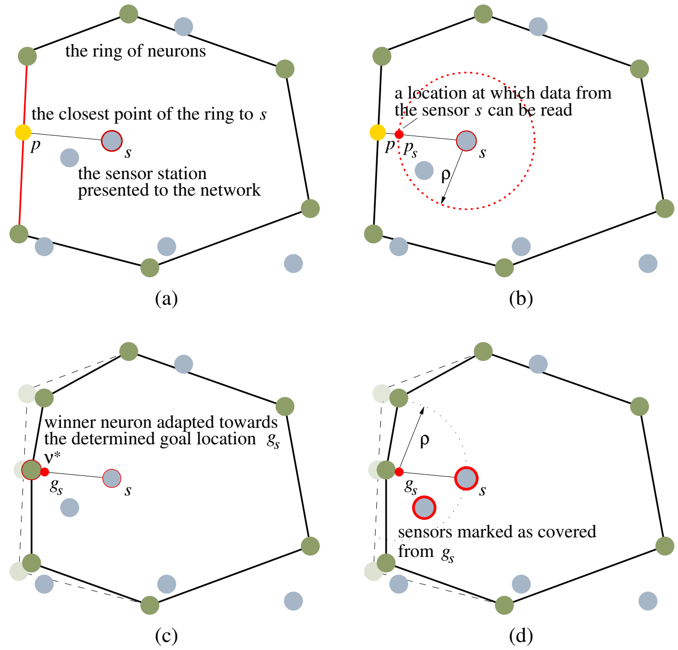


Fig. 2. Visualization of the determination of the winner neuron and goal location: (**a**) winner corresponds to the closest point (in yellow) of the ring to the presented sensor (blue disc with red outline); (**b**) determination of p_s (red small disc) at the ρ communication radius from the sensor s; (**c**) the current ring after adaptation of the winner and its neighbouring neurons to g_s, which is created from p_s; (**d**) sensors within the communication radius ρ from g_s. The blue discs denote sensor stations, neurons (their weights) are in green.

The determination of the winner neuron ν^* itself further distinguishes three cases because we aim to adjust the number of neurons during the learning process. Therefore, the ring is searched for a neuron ν with the identical weights as the coordinates of p first. If such ν is not found, a new neuron ν_{new} is added to the network with the weights set to the coordinates of p and positioned between the neurons that form the straight line segment on which p is laid (see Fig. 2a); and ν_{new} becomes the winner neuron. In the case ν is found, two additional cases are considered to follow the inhibition mechanism proposed in [30], which avoids selecting a single neuron to be winner neuron multiple times during a single presentation of all sensors to the network (a single learning epoch). If ν has been already selected as a winner neuron in the current epoch, it indicates there is a high demand to adapt the network in the area around the presented sensor s. Therefore, the neuron is duplicated to increase the number of neurons, and this new neuron is set as the winner neuron ν^*. Otherwise ν becomes the winner neuron ν^* for s.

The winner neuron ν^* and its neighbouring neurons are adapted to the goal location g_s, as in a standard SOM for the TSP (see Fig. 2c), and g_s becomes one of the goal locations determined in the current epoch. Notice, data from several sensors can potentially be read from g_s. Therefore, all sensors in the ρ-distance from g_s are marked as covered for the current epoch (see Fig. 2d), and only uncovered sensors are considered for the network adaptation.

During each learning epoch, all goal locations $\mathbf{G}$ to be visited are determined. However, $\mathbf{G}$ can vary during the learning, and therefore, the ring regeneration [38] is performed prior the next iteration of the learning procedure. In prior work, only winners are preserved and all other neurons are removed [38]. Here, additional neurons are placed between each preserved winner neurons to improve convergence of the network as follows. A sequence of neuron weights between two winners in the original ring forms a path. After removing the neurons in between the winners, a new neuron is placed at the location p_c corresponding to half of this path, i.e., its weights are set to the coordinates of the corresponding p_c.

The proposed SOM with all the recommend parameters and initialization is summarized in the following steps:

1) *Initialization:* For n sensor stations $\mathbf{S} = \{s_1, \ldots, s_n\}$, create $2n$ neurons around the first sensor s_1.[2] Set the neighbouring function variance $\sigma \leftarrow 10$, the learning rate $\mu \leftarrow 0.99$, and the learning epoch counter $i \leftarrow 1$.
2) *Randomizing:* Create a random permutation of the sensor stations $\Pi(\mathbf{S}) \leftarrow \mathrm{permute}(\mathbf{S})$ to avoid local minima.
3) *Clear goal locations* $\mathbf{G} \leftarrow \emptyset$ for the current epoch i.
4) *Determine the weights of the expected winner neuron* using the closest point p of the current ring to the presented sensor $s \in \Pi(\mathbf{S})$ (Fig. 2a).
5) *Determine a location p_s to read data from s* such that p_s lies on (p, s) and $|(p_s, s)| = \rho - 0.01$ (see Fig. 2b).
6) *Adapt:* `If` $i = 1$ `or` $|(p, p_s)| \leq \zeta(s)$ `Then`
   - Determine the winner neuron ν^* from p by the procedure described earlier in this section;
   - Associate p_s to ν^* as the goal location g_s;
   - Extend the set of the goal locations $\mathbf{G} \leftarrow \mathbf{G} \cup \{g_s\}$;
   - Adapt ν^* and its neighbouring neurons ν_j within the distance d (in the number of neurons) using the neighbouring function $f(\sigma, d) = \mu e^{(-d^2/\sigma^2)}$ for $d < 0.2m$ and $f(\sigma, d) = 0$ otherwise, where m is the current number of neurons;
   - Mark all reachable sensors from g_s as covered; $\Pi(\mathbf{S}) \leftarrow \Pi(\mathbf{S}) \setminus \{s_i : |(s_i, g_s)| \leq \rho, s_i \in \mathbf{S}\}$.

   `If` $|\Pi(\mathbf{S})| > 0$ go to Step 4 (all sensors evaluated).
7) *Ring regeneration:* Create a new ring using only the winners for the current epoch and add a new neuron between each two consecutive winners with the weights set to the point halfway along the path connecting them.
8) *Update the number of the learning epoch and neighbouring function variance:* $i \leftarrow i + 1$; $\sigma \leftarrow (1 - i\alpha)\sigma$; Update the learning rate μ according to (3).
9) *Termination condition:* `If` the distance of each winner to its associated goal location is less than 10^{-3} `Or` $i > 40$, stop the adaptation. `Otherwise` go to Step 2.
10) *Final tour construction:* Traverse the ring and use the sequence of winners in the ring to construct the final tour from the goal locations associated with the winners.

An evolution of the proposed self-organizing network is visualized in Fig. 3. An implementation of the described algorithm for the PC-TSPN is available at [39].

[2] Various methods of initialization may be considered. For simplicity and replicability we initialize the network around s_1.

D. Computational Complexity

The most time-consuming operations of the proposed learning procedure are the winner neuron selection, adaptation of neurons, and evaluation of sensors that are covered from the determined goal locations. The complexity of these operations depends on the number of sensor locations n for which the network can have up to $3n$ neurons in a single learning epoch. In the worst case, all neurons can be adapted and all sensors can be evaluated. Hence, the complexity of each learning epoch can be bounded by $(3n + 3n + n)n$.

The number of learning epochs can be restricted explicitly since the proposed unsupervised learning provides a solution at the end of each epoch. Moreover, the adaptation is stable because μ and the neighboring function f are always less than 1.0 [40]. The neuron weights are effectively changed only for a sufficiently high value of $f(\sigma, d)$. Since σ depends only on α, and it is decreasing after each epoch, the highest possible value of f is also decreasing. Hence, the neurons weights are stabilized in a constant number of epochs and the overall computational complexity can be bounded by $O(n^2)$.

E. Quality of the Solutions

The proposed unsupervised learning is a stochastic procedure to map the given input space to a one dimensional array of the determined goal locations $\mathbf{G}$ associated to the units of the SOM output layer. The learning procedure does not guarantee an optimal solution would be found in a finite number of learning epochs; hence, the procedure is a heuristic polynomial algorithm to address NP-hard routing problems. On the other hand, an initial guess about the solution quality is provided quickly since a feasible solution is available after each epoch. To the best of the authors' knowledge, there are no known approximation bounds for SOM-based TSP solvers. Therefore, an empirical evaluation is reported in Section VI.

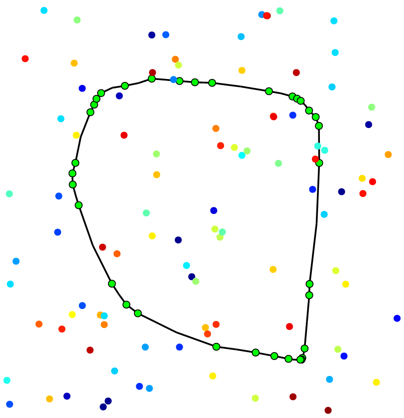
V. Extensions to Varying Communication Radius and Multi-Vehicle Missions

A. Varying Communication Radii

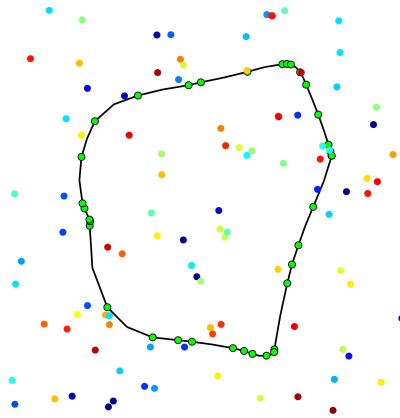
A flexibility of the self-organizing procedure allows varying communication radii for individual sensor stations (e.g., based on the local conditions influencing radio signal propagation). The only modification of the proposed adaptation procedure is to consider a particular communication radius ρ_s associated with each sensor station $s \in \mathbf{S}$ in the determination of the goal location to collect data from s. A visualization of such a problem and found solution is depicted in Fig. 4.

B. Multi-Vehicle Missions

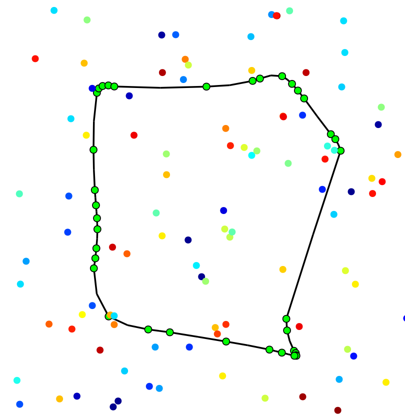
The underlying self-organizing map for the TSP is also flexible for addressing data-collecting problems with several vehicles. The idea is based on the extension of the SOM adaptation procedure for the TSP to the *multiple traveling salesman problem* (MTSP) by considering an individual ring of neurons for each salesman. The winner neuron is selected as the closest neuron to the presented city by considering all the rings. In the MTSP, all tours start and finish at the same (home)



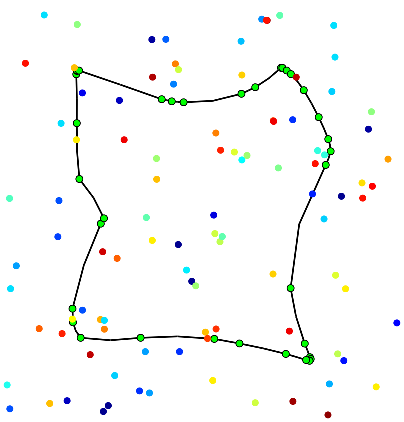
(a) Learning epoch 5



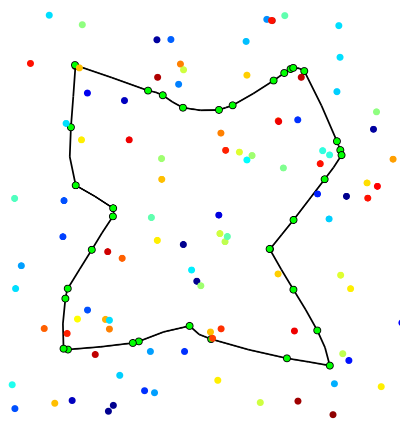
(b) Learning epoch 7



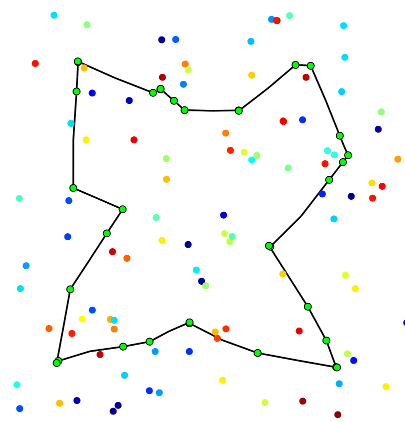
(c) Learning epoch 10



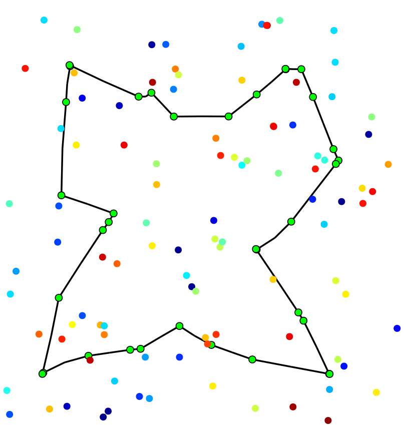
(d) Learning epoch 12



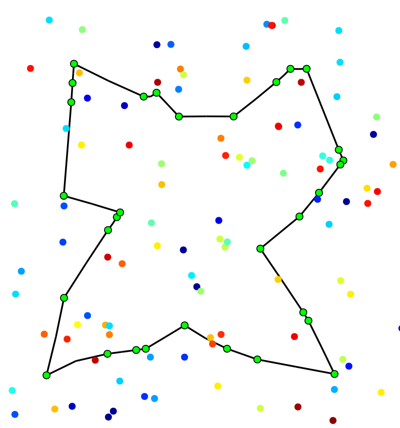
(e) Learning epoch 17



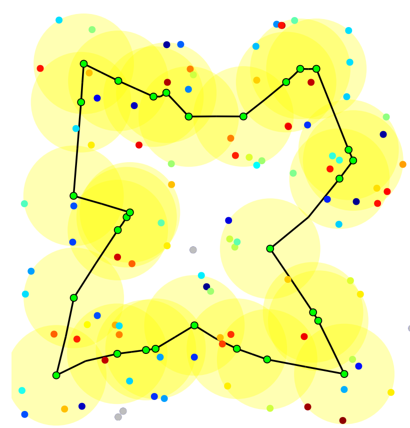
(f) Learning epoch 19



(g) Learning epoch 21



(h) Learning epoch 25



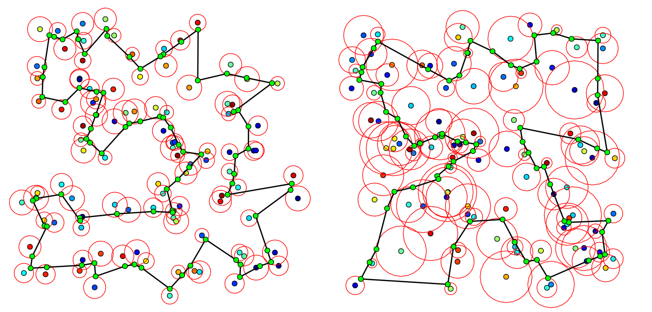
(i) Learning epoch 27 (final solution)

Fig. 3. Example of network evolution. Green connected discs represent neuron weights. Colored discs are sensors, where the more important sensors (higher penalty cost ζ) are in red while less important sensors are in blue. In the final solution, uncovered sensors are in gray.

location (called the depot in literature). In each learning epoch, the best matching neuron to the depot is found in each ring and adapted to the depot prior to selecting winners to other cities. Thus, the adaptation consists of two parts:

1) Select and adapt the winner from each ring to the depot;
2) For each not yet covered city, select a winner from all neurons (regardless of ring), and adapt it to the city.

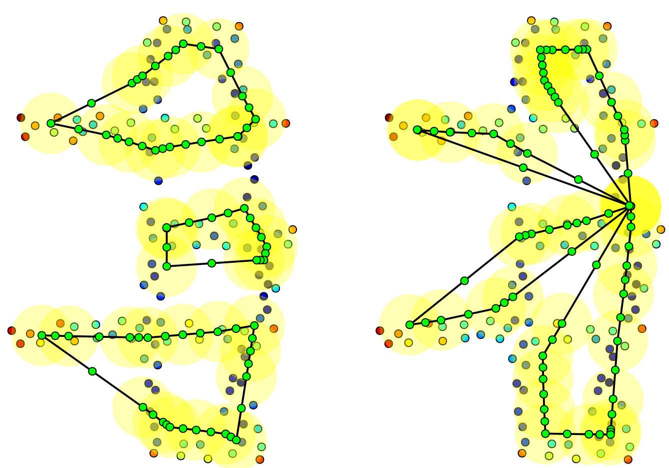
This extension was introduced in [41], where the authors prefer winners from any ring that is shorter than the average length of the rings. Therefore, the distance of the neuron ν to the presented city is penalized by the factor $1+(l_\nu - l_{avg})/l_{avg}$, where l_ν is the length of the ring to which the neuron ν belongs, and l_{avg} is the average length of the rings. The preference of winners from shorter rings is motivated by the desire to avoid expanding longer rings and thus minimizing the length of the longest tour. This technique provides a solution to the MTSP with a Minmax objective. If we avoid adaptation of the winners from each ring to the depot, patrolling routes



(a) 100 km × 100 km area (b) 200 km × 200 km area

Fig. 4. Solutions of problems with varying communication radii (particular values are drawn from a uniform distribution): red circles denote the communication radius of each sensor station; the found path is represented by the black straight line segments with small green disks denoting the determined goal locations; the other small disks are the sensor stations filled by the color corresponding to the importance of the data (high penalty data are in red while low penalty are in blue); all sensors are covered.

can be found, which has been shown in [11].



(a) Three vehicles, no depot (b) Four vehicles with a depot

Fig. 5. Example of found solutions with and without a common depot.

The same ideas can also be applied in the context of the data collection planning. Examples of multi-vehicle data collection paths are visualized in Fig. 5.

VI. Results

We have evaluated the proposed unifying data collection planning approach in several data collection problems. In this section, the performance of our SOM-based approach is compared with an existing deterministic combinatorial algorithm proposed in [9]. The combinatorial approximate PC-TSPN algorithm is based on decomposing the problem into the determination of a covering set and a consecutive solution to the TSP. A set of goal locations is determined from the neighbouring sensors defined by the communication radius ρ. A heuristic [22] is utilized to determine a subset of this covering set. The resulting TSP is then solved optimally using the Concorde solver [26]. We denote this solution from prior work as PC-TSPN in the presented results.

Two variants of the proposed unsupervised learning procedure are considered. The first variant uses the SOM-based method as a sole algorithm to simultaneously determine both the goal locations and a tour to visit these locations. In the second variant, the SOM adaptation is utilized only to determine the goal locations. Then, the TSP of these goal locations is solved optimally by the Concorde solver [26] (similarly as the combinatorial PC-TSPN algorithm). We denote these variants as SOM and SOM+TSP, respectively, in the rest of the paper.

All three algorithms are evaluated in scenarios with random deployments of sensors and varying ratios of mutual distances between them. The associated penalty for not collecting data from a sensor is also varied. In addition, the algorithms are compared in a realistic underwater monitoring deployment.

A. Random Deployments of Sensor Stations

In this evaluation, the solution quality provided by the algorithms is compared in random problem instances with different penalties and communication radius ρ. These instances include the standard TSP (for $\rho = 0$ and very high penalties), PC-TSP (for $\rho = 0$), and TSPN (for $\rho > 0$ and very high penalties). A penalty for each sensor station is drawn from a uniform distribution for *low*, *middle*, *high*, and *very high* values of the penalty (see Table II). The communication radius is set to the range $0 \leq \rho \leq 50$ km.

TABLE II
Considered values of penalties in the random deployments

Penalty Assignment Schema	Penalty Range
very high penalties	$0 \leq \zeta \leq 25000$
high penalties	$0 \leq \zeta \leq 250$
middle penalties	$0 \leq \zeta \leq 25$
low penalties	$0 \leq \zeta \leq 5$

Three data collection scenarios are considered for random deployments of sensors. The first scenario consists of 100 randomly placed sensors in a 100 km × 100 km area. To further study the influence of different penalty ratios and distances between the sensors, an additional scenario with 100 randomly placed sensors within a 200 km × 200 km large area was created. An average vehicle speed of 5 km per hour is assumed, and thus the sensors are effectively placed in 20 × 20 and 40 × 40 large squares, respectively. The cost between the locations is directly computed as their Euclidean distance (similarly to [9]). The scenarios are denoted as *100 km × 100 km area* and *200 km × 200 km area*, respectively. For each scenario, 50 random instances were created, and for each such an instance, the penalties were assigned according to the four different schemata depicted in Table II. The communication radius ρ is set to one of 11 different values from the range 0 to 50 km, which provides 4400 random problems.

Finally, we consider random values of the penalties in an additional sensor placement scenario (denoted as *OOI area*) taken from the Ocean Observatories Initiative (OOI) [42] (see Section VI-B for more detail) from which 50 random problems were created for the penalties assignment schemata from Table II. The same ρ values as for the prior scenarios are considered in these problems.

The total number of problem instances evaluated in this study is 5500. Only a single trial per each problem is solved by the deterministic PC-TSPN algorithm [9]. However, 50 trials are solved for each instance by the stochastic SOM algorithms, which gives 275,000 trials per algorithm variant.

To study the algorithms' performance using such a huge set of results, the solution quality is standardized as a ratio to a reference value. Using the ratio as the solution quality indicator, we can aggregate the results according to a particular penalty schema and communication radius ρ. Since optimal algorithms for the general TSPN and PC-TSP are not available, we instead use the optimal solution of the TSP as the reference value, which can be straightforwardly computed by any available TSP solver. The selected solution quality indicator is thus the ratio of the solution cost provided by the tested algorithm to the cost of the optimal solution of the related standard TSP, i.e., without penalties and consideration of the communication radius ρ. Such a reference solution is found for each problem instance by Concorde [26]. Having an optimal tour T_{TSP} of the underlying TSP, the ratio is computed as

$$R = \frac{\mathcal{C}(T)}{\mathcal{C}(T_{TSP})}. \tag{4}$$

The average results of R are presented in Fig. 6, Fig. 7, and Fig. 8, where the standard deviations are shown as error bars. Due to the way the ratio is computed, we note that it is expected an algorithm for the PC-TSPN outperforms the optimal solution of the underlying TSP because the solution of the PC-TSPN can benefit from consideration of the communication radii and penalties. Regarding the cost function (1) of the autonomous data collection mission, a lower value of R means a better solution, and thus an algorithm that provides lower R is a more suitable for these missions. In the *100 km × 100 km area* and *200 km × 200 km area* scenarios, the proposed SOM provides improved results in comparison with the combinatorial PC-TSP heuristic. The main differences between SOM and SOM+TSP approaches are for the problem instances with $\rho = 0$, which are solutions of the TSP (or PC-TSP). This is because the SOM provides only an approximate solution of the TSP. Regarding the particular results for these standard TSP instances, the average difference between the solution quality and the optimal solution is about 3–5%. Similar results can be observed in other SOM-based solvers for the standard TSP (e.g., see [31], [33]). Regarding the performance of the PC-TSP heuristic [9], the ratio R close to 1 indicates the sensor selection is not evident, as mostly all sensors are visited by the provided solution. A noticeable improvement by sensor selection can be seen in Fig. 7d, where SOM selects better sensors than the deterministic heuristic.

For the *OOI area* scenario with *low penalties*, the deterministic PC-TSPN algorithm provides lower solution cost than the SOM-based approaches for $5 \leq \rho \leq 30$ km. This is because the penalties with respect to distances between sensors are so low that it does not make sense to travel to different places, and solutions consist of data collection from a single goal location. In this case, the SOM adaptation algorithm simply performs a random selection of the location and overall provides worse solution than the deterministic algorithm. However, this situation can be easily detected without any significant computational cost as follows. If the total cost for such a single goal location solution is lower than the previous tour cost, it can be used as a new solution to the problem. Results for this modified algorithm (denoted as SOM+1GT in Fig. 8d) indicate that in these problems, the penalties are too low, and the PC-TSPN degrades to selection of the single best covering goal location.

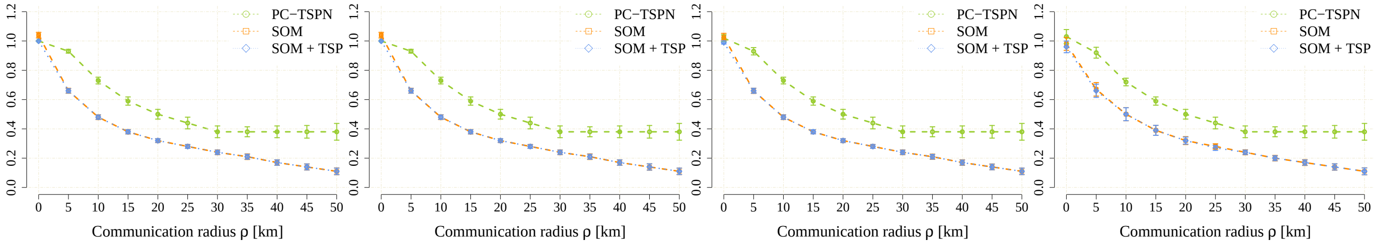
Required Computational Time: The required computational time mostly depends on the number goal locations, which is higher for high penalties and low ρ. Therefore, in Fig. 9, the computational time is shown only for the *100 km × 100 km area* scenario. The evaluated algorithms are implemented in C++, and all results have been computed using a single core of the iCore7 processor running at 3.4 GHz. The most time-consuming part of the algorithms is the optimal solution of the TSP using [26], which increases the computational time significantly for the deterministic PC-TSP heuristic algorithm [9]. On the other hand, the SOM algorithm provides solutions in tens of milliseconds (less than 20 milliseconds), which is about 5 times faster than our early results [13]. Additionally, the radius ρ is more important for computational considerations than the penalties because the adaptation is less demanding with increasing ρ than for decreasing penalty values.

B. *Underwater Monitoring Deployment*

The proposed SOM-based algorithm was also tested using example configurations from the Ocean Observatories Initiative (OOI) Endurance Array [42]. The OOI Endurance Array is a project to continuously maintain a presence of Autonomous Underwater Vehicles (AUVs) off the coast of the Pacific Northwest (Washington and Oregon) with the intention of monitoring important biological and physical measurements. In this paper, we primarily consider the case where a single AUV is deployed and focus on single vehicle missions.

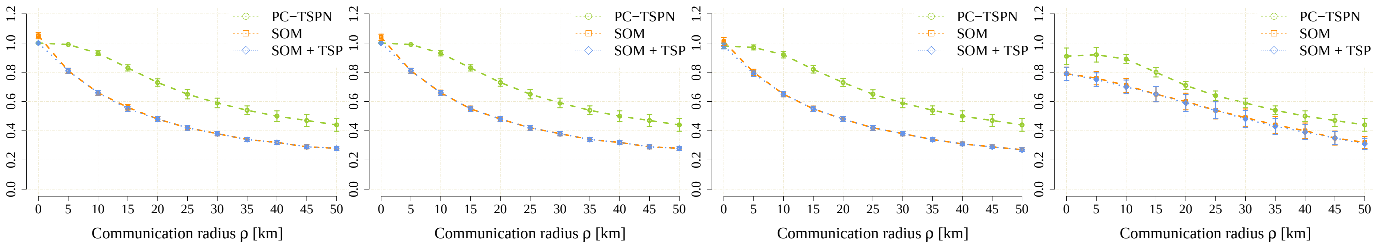
A sensor placement based on the OOI Endurance Array was used to evaluate performance of the SOM-based solver for the PC-TSPN in a more realistic case. In simulation, 100 sensors are placed along the planned AUV paths, and we assume that the sensors and AUV are equipped with acoustic modems for wireless communication [5]. The range of these modems can vary depending on power and environmental conditions, which corresponds to the size of the neighborhoods. Ranges up to 50 km are possible with existing technology [43]. The placement of the sensors is visualized in Fig. 10.

The penalty (prize) assigned to each sensor is based on the variance of a Gaussian Process (GP) [44]. A GP is a non-parametric Bayesian inference technique to estimate a quantity of interest (e.g., temperature, salinity, chlorophyll, or dissolved oxygen) and also predict the variance of those estimates. The output of the GP is an estimate of the quantity of interest at each point in space along with an uncertainty in each estimate. The GP has been employed in the literature as a way of measuring sensing performance because reducing the variance correlates with an improved estimate of a quantity of interest over a spatial field [45].



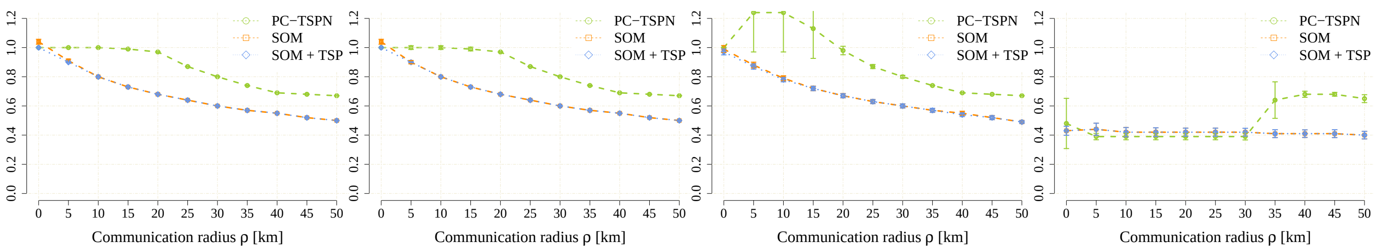
(a) *very high penalties* (b) *high penalties* (c) *middle penalties* (d) *low penalties*

Fig. 6. Mean ratios R of the solution cost relative to the optimal solutions of the related TSP in the 100 km $\times$ 100 km area scenario. Lower ratios correspond to better performance.



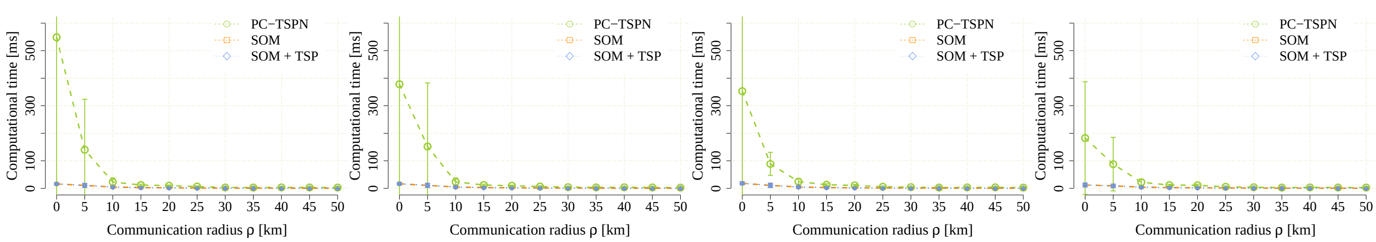
(a) *very high penalties* (b) *high penalties* (c) *middle penalties* (d) *low penalties*

Fig. 7. Mean ratios R of the solution cost relative to the optimal solutions of the related TSP in the 200 km $\times$ 200 km area scenario. Lower ratios correspond to better performance.



(a) *very high penalties* (b) *high penalties* (c) *middle penalties* (d) 111*low penalties*

Fig. 8. Mean ratios R of the solution cost relative to the optimal solutions of the related TSP in the *OOI area* scenario. Lower ratios correspond to better performance.



(a) *very high penalties* (b) *high penalties* (c) *middle penalties* (d) *low penalties*

Fig. 9. Mean computational time (in milliseconds) for the problems in the 100 km $\times$ 100 km area scenario.

In the simulations presented, we input sensor data of ocean temperature from the Regional Ocean Modeling System [46] into a GP. The commonly used squared exponential kernel is employed to model correlations between points. Then, hyperparameters were learned using conjugate gradient ascent on the marginal likelihood. The resulting variance was then utilized as the penalty for not collecting data from each sensor in the configuration in Fig. 10. This setup yields an instance of the PC-TSPN similar to those described throughout the paper. The mean costs of the solutions in the OOI scenario are depicted in Fig. 11. The SOM-based approach provides substantial improvement over the PC-TSPN heuristic in this real-world monitoring scenario.

An example of a solution in the OOI scenario is depicted in Fig. 12. In this example, the vehicle collects data from all sensors. The performance improvement of the SOM-based approach lies in the determination of goal locations. The deterministic PC-TSPN algorithm visits the sensor locations

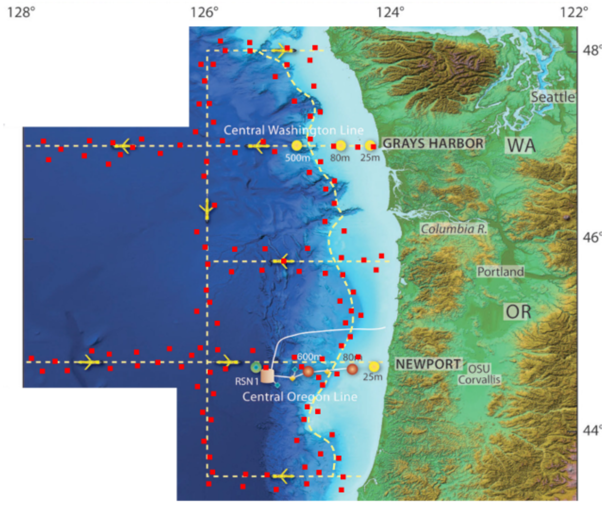


Fig. 10. OOI Endurance Array [42] of autonomous underwater vehicles monitoring the ocean. Simulated sensors are placed along the monitoring lines on the seafloor (red squares). Figure courtesy J. Barth and D. Reinert, Oregon State University.

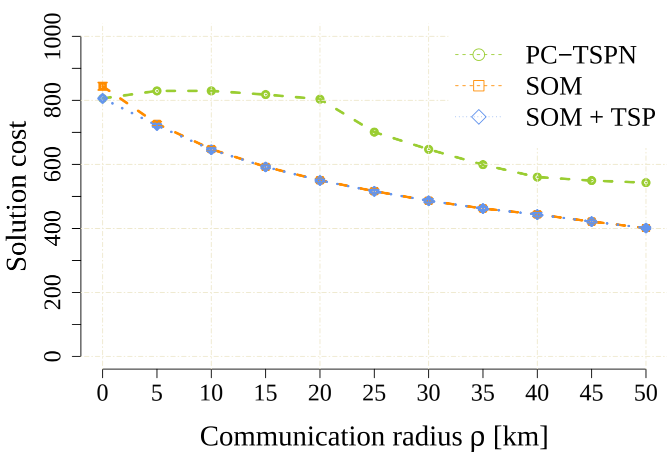


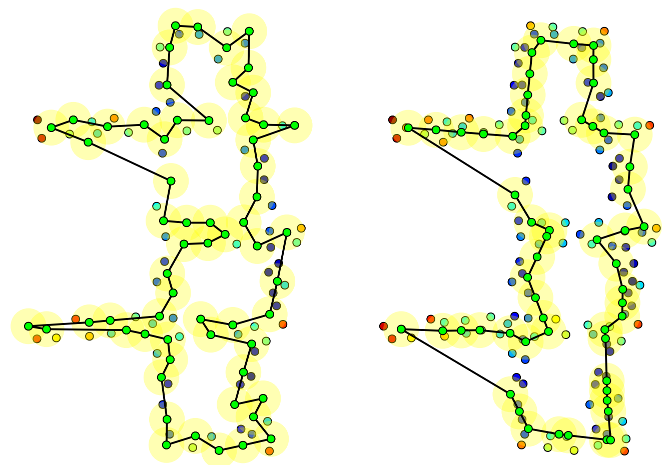
Fig. 11. Solution cost provided by the proposed SOM-based algorithms and the deterministic PC-TSPN heuristic in the OOI data collection scenario (see Fig. 10). The penalties correspond to the variance of a Gaussian Process.

themselves as a part of the tour, but the SOM-based approach is able to select new locations to visit and improve the tour cost. The most essential locations are shown by the red discs, which are found in the distant part of the environment. Data from a number of sensors are collected when the vehicle travels to these locations.

C. Multi-Vehicle Mission

The OOI scenario was considered for a preliminary evaluation of the proposed planning method in multi-vehicle missions. Solution costs for 1 to 10 vehicles for two types of the missions are depicted in Fig. 13. The first type is a mission with the common starting location (depot), and the second type is without it. The location of the depot is shown in the example of found solutions in Fig. 5.

In the OOI scenario, all sensors are covered, and the results indicate that a more than four vehicles do not provide a significant improvement if vehicles start and return at the



(a) PC-TSPN [9], $C = 647$ (b) Proposed SOM, $C = 484$

Fig. 12. Solutions of the OOI deployment scenario for the communication radius $\rho = 30$ km.

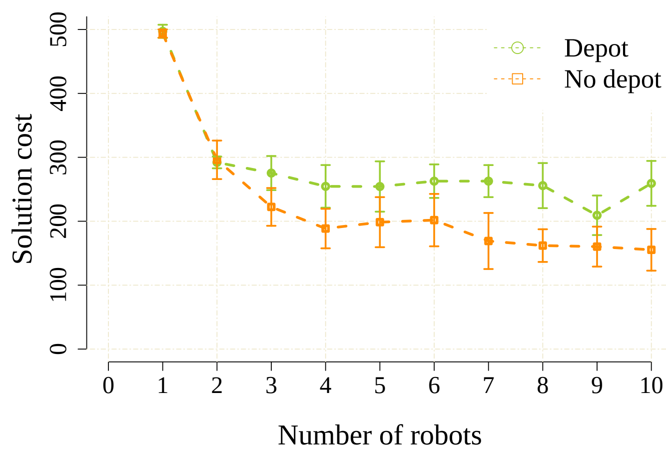


Fig. 13. Solution costs for multi-vehicle missions with and without a common depot in the OOI data collection scenario with $\rho = 30$ km.

common depot. This is caused by vehicles traveling to distant sensors and covering other sensors along the path. With increasing numbers of the vehicles, more and more vehicles have zero path length, which means they are not used in the data collection. It seems that just four vehicles are enough to retrieve data from all the sensors. This finding could prove valuable for oceanographers looking to determine how many AUVs to deploy in a given scenario. A further investigation of multi-vehicle missions is needed to study this behavior and to identify the most appropriate number of vehicles to collect the required data in an efficient way.

D. Discussion

The proposed SOM-based algorithm for data collection planning outperforms the deterministic heuristic for the PC-TSPN [9] in almost all the problem instances examined. The heuristic algorithm provides better solutions only for small communication radii (practically for $\rho = 0$) and problems with high penalties, where the data collection problem is close to the standard TSP. In these cases, the SOM-based solution of the combinatorial TSP is worse than the optimal solution found by Concorde. This drawback can be addressed by the determination of the goal locations by SOM and optimal solution of the TSP in the proposed SOM+TSP, at the cost of more computationally demanding algorithm.

The combinatorial heuristic approach also provides better results when the solution is a single goal location, i.e., a single location provides all the important measurements. In these situations, the proposed SOM provides a solution consisting of several goal locations, which are somewhat worse than the cost of the single goal solution provided by PC-TSPN heuristic. This is because the learning procedure starts with an empty set of covered sensors. Then, the sensors are presented in a random order, and the algorithm typically needs a couple of goal locations to create a coverage that cannot be improved by adaptation to other sensors. However, this case can be trivially checked, as it is demonstrated by the proposed post-processing procedure in the SOM+1GT variant. Moreover, the PC-TSPN heuristic does not appear to provide stable behavior since the average solution cost is increased for a higher neighborhood size (see Fig. 8d). Therefore, the proposed post-processing is competitive with the PC-TSPN heuristic.

Regarding instances with low penalties and a high neighborhood size (large communication radius), it is also worth mentioning the solutions form small tours that cover a small random subset of the sensors. This is an indication that penalties are not set appropriately or the problem does not fit the PC-TSP problem formulation well. According to our observations, we believe this is related to the assumptions of static penalties and independent sensor measurements. The proposed SOM-based algorithm's low computational requirements would allow for further extension to compute the penalty values during the learning process [47], which could improve the solution quality.

The main difference between the pure SOM and SOM+TSP algorithms is for the zero neighborhood size and high penalty, where the optimal solution of the TSP improves the quality of the solution noticeably. Hence, it seems that determining appropriate goal locations affects the solution cost more than providing the optimal tour. The results also indicate that for problems with $\rho > 0$, the SOM-based unsupervised learning procedure provides solutions to the underlying TSP that are competitive to the optimal tour found by Concorde. The slightly worse solution of the TSP provided by the pure SOM-based approach is often negligible relative to the overall solution cost improvement gained by the SOM selection of the subset of the sensors and determined goal locations.

The presented results for the examined single vehicle data collection missions provide supporting evidence that the SOM-based planning approach is competitive with the combinatorial heuristics. An additional benefit of the proposed SOM-based approach is the ability to address multi-vehicle missions, which has been briefly described in Section V. The preliminary results show that SOM can provide solutions to multi-vehicle problems; however, we found that the quality of solutions has a higher variance than in the presented single-vehicle missions. Regarding the scaling of the approach with the number of vehicles, the key aspect of provided solutions is the situation when more vehicles do not improve the mission performance. This may be caused by a characteristic of the problem or by a behavior of the proposed adaptation procedure. A study of these aspects and eventual improvement of the solution quality is a subject of our future work [48].

VII. Conclusion

This paper proposed a unifying approach for planning data collection missions using a multi-goal path planning framework. The particular class of data collection tasks consisted of problems where it is necessary to consider penalties on unvisited sensors (goals) and the sensors are surrounded by neighborhoods (goal regions). The presented results indicate the proposed SOM-based planning approach outperforms currently available combinatorial heuristic approaches and provides improved solutions in all the problem variants examined: the TSPN, PC-TSP, and PC-TSPN. Moreover, the proposed SOM-based approach has substantially lower computational requirements for the problems with a high number of determined goal locations.

Although the combinatorial heuristics for the TSP provide better (or even optimal) solutions of the standard TSP versus the applied SOM-based unsupervised learning in the data collection tasks, the main source of the solution improvement relies on the selection of the most appropriate goal locations within the sensor neighborhoods (i.e., considering communication radii) during the planning. Besides, once the goal locations are determined, the related TSP on such goal locations can always be solved optimally to further improve the solution, but at the cost of significantly higher computational requirements.

Based on the presented results of the algorithms' performance, it seems the main difficulty in solving the addressed data collection problems in the desired way relies on designing appropriate penalties according to the vehicle travel cost to move between the sensor stations. This difficulty stems from the penalty values being domain specific and depending on the phenomena studied. In addition, the current formulation of the problems assumes that sensors provide independent measurements. Therefore, online determination of the sensor penalties during the unsupervised learning is a subject of our future work to overcome this limitation.

References

[1] 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.

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

[3] S. L. Nooner and W. W. Chadwick Jr., "Volcanic inflation measured in the caldera of axial seamount: Implications for magma supply and future eruptions," *Geochemistry, Geophysics, Geosystems*, vol. 10, no. 2, 2009.

[4] M. Kohler, P. Lynett, M. Legg, and D. Weeraratne, "Importance of large-scale bathymetry features on 2011 Tohoku tsunami waveforms through comparison of simulations with the spatially dense ALBACORE OBS array data," in *Proc. AGU Fall Meeting*, 2012.

[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] D. Applegate, R. Bixby, V. Chvátal, and W. Cook, *The Traveling Salesman Problem: A Computational Study*. Princeton, NJ, USA: Princeton University Press, 2007.

[7] T. A. Lian, M. R. Llave, M. Goodwin, and N. Bouhmala, "Towards multilevel ant colony optimisation for the euclidean symmetric traveling salesman problem," in *IEA/AIE*, 2015, pp. 222–231.

[8] 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.

[9] G. Hollinger, U. Mitra, and G. Sukhatme, "Autonomous data collection from underwater sensor networks using acoustic communication," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2011, pp. 3564–3570.

[10] 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.

[11] 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.

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

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

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

[15] J. Faigl, T. Krajník, V. Vonásek, and L. Přeučil, "On Localization Uncertainty in an Autonomous Inspection," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012, pp. 1119–1124.

[16] A. Dumitrescu and J. Mitchell, "Approximation algorithms for TSP with neighborhoods in the plane," *J. Algorithms*, vol. 48, no. 1, pp. 135–159, 2003.

[17] M. de Berga, J. Gudmundssonb, M. J. Katzc, C. Levcopoulosd, M. H. Overmarse, and A. F. van der Stappen, "TSP with neighborhoods of varying size," *J. Algorithms*, vol. 57, no. 1, pp. 22–36, 2005.

[18] S. Safra and O. Schwartz, "On the complexity of approximating tsp with neighborhoods and related problems," *Computational Complexity*, vol. 14, no. 4, pp. 281–307, 2006.

[19] E. Balas, "The Prize Collecting Traveling Salesman Problem," *Networks*, vol. 19, pp. 621–636, 1989.

[20] G. Ausiello, V. Bonifaci, S. Leonardi, and A. Marchetti-Spaccamela, "Prize-Collecting Traveling Salesman and related problems," in *Handbook of Approximation Algorithms and Metaheuristics*, T. F. Gonzalez, Ed. CRC Press, 2007.

[21] G. Ausiello, V. Bonifaci, and L. Laura, "The Online Prize-Collecting Traveling Salesman Problem," *Information Processing Letters*, vol. 107, no. 6, pp. 199–204, 2008.

[22] M. Goemans and D. P. Williamson, "A general approximation technique for constrained forest problems," *SIAM J. Computing*, vol. 24, no. 2, pp. 296–317, 1995.

[23] A. Archer, M. H. Bateni, M. T. Hajiaghayi, and H. Karloff, "Improved approximation algorithms for Prize-Collecting Steiner Tree and TSP," in *Proc. IEEE Symp. Foundations of Computer Science*, 2009, pp. 427–436.

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

[25] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke, "Data collection, storage, and retrieval with an underwater sensor network," in *Proc. Int. Conf. Embedded Networked Sensor Systems*, 2005, pp. 154–165.

[26] D. Applegate, R. Bixby, V. Chvátal, and W. Cook, "CONCORDE TSP Solver," 2003, [cited 28 Aug 2014]. [Online]. Available: http://www.tsp.gatech.edu/concorde.html

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

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

[29] J. C. Fort, "Solving a combinatorial problem via self-organizing process: An application of the Kohonen algorithm to the traveling salesman problem," *Biological Cybernetics*, vol. 59, no. 1, pp. 33–40, 1988.

[30] S. Somhom, A. Modares, and T. Enkawa, "A self-organising model for the travelling salesman problem," *Journal of the Operational Research Society*, pp. 919–928, 1997.

[31] 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.

[32] 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.

[33] J. Faigl, "On the performance of self-organizing maps for the non-euclidean traveling salesman problem in the polygonal domain," *Information Sciences*, vol. 181, pp. 4214–4229, October 2011.

[34] J. Zhang, X. Feng, B. Zhou, and D. Ren, "An overall-regional competitive self-organizing map neural network for the euclidean traveling salesman problem," *Neurocomputing*, vol. 89, pp. 1–11, 2012.

[35] S. Alatartsev, S. Stellmacher, and F. Ortmeier, "Robotic task sequencing problem: A survey," *Journal of Intelligent & Robotic Systems*, vol. 80, no. 2, pp. 279–298, 2015.

[36] J. Faigl and G. Hollinger, "Self-organizing map for the prize-collecting traveling salesman problem," in *Advances in Self-Organizing Maps and Learning Vector Quantization: Proceedings of the 10th International Workshop (WSOM)*, 2014.

[37] W. Zhang, Y. Bai, and H. P. Hu, "The incorporation of an efficient initialization method and parameter adaptation using self-organizing maps to solve the TSP," *Applied Mathematics and Computation*, vol. 172, no. 1, pp. 603–623, 2006.

[38] J. Faigl and L. Přeučil, "Self-Organizing Map for the Multi-Goal Path Planning with Polygonal Goals," in *ICANN*, 2011, pp. 85–92.

[39] J. Faigl and G. Hollinger, "SOM for the PC-TSPN," [cited 30 Nov 2015]. [Online]. Available: http://purl.org/faigl/pctspn

[40] M. Tucci and M. Raugi, "Stability analysis of self-organizing maps and vector quantization algorithms," in *IJCNN*, 2010, pp. 1–5.

[41] S. Somhom, A. Modares, and T. Enkawa, "Competition-based neural network for the multiple travelling salesmen problem with minmax objective," *Computers and Operations Research*, vol. 26, no. 4, pp. 395–407, 1999.

[42] W. Ramos. (2014, Feb.) Ocean observatories initiative. [cited 10 Nov 2015]. [Online]. Available: http://oceanobservatories.org/

[43] L. Freitag, M. Stojanovic, M. Grund, and S. Singh, "Acoustic communications for regional undersea observatories," in *Proc. Oceanology International*, London, UK, 2002.

[44] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[45] G. Hollinger, B. Englot, F. Hover, U. Mitra, and G. Sukhatme, "Active planning for underwater inspection and the benefit of adaptivity," *Int. J. Robotics Research*, vol. 32, no. 1, pp. 3–18, 2013.

[46] A. F. Shchepetkin and J. C. McWilliams, "The Regional Oceanic Modeling System (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model," *Ocean Modelling*, vol. 9, no. 4, pp. 347–404, 2005.

[47] J. Faigl and P. Váňa, "Self-organizing map for data collection planning in persistent monitoring with spatial correlations," in *SMC*, 2016, (to appear).

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