Inspection Planning in the Polygonal Domain by Self-Organizing Map

Jan Faigl

Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, 166 27, Prague 6, Czech Republic

Abstract

Inspection planning is a problem of finding a (closed) shortest path from which a robot "sees" the whole workspace. The problem is closely related to the Traveling Salesman Problem (TSP) if the discrete sensing is performed only at the finite number of sensing locations. For the continuous sensing, the problem can be formulated as the Watchman Route Problem (WRP), which is known to be NP-hard for the polygonal representation of the robot workspace. Although several Self-Organizing Map (SOM) approaches have been proposed for the TSP, they are strictly focused to the Euclidean TSP, which is not the case of the inspection path planning in the polygonal domain. In this paper, a novel SOM adaptation schema is proposed to address both variants of the inspection planning with discrete and continuous sensing in the polygonal domain. The schema is compared with the state of the art SOM schema for the TSP in a set of multi-goal path planning problems and WRPs. The proposed algorithms are less computationally intensive (in order of tens) and provide better or competitive solutions.

Keywords: Inspection Planning, Multi-Goal Path Planning, Self-Organizing Map (SOM), Traveling Salesman Problem (TSP), Watchman Route Problem (WRP), Polygonal Domain

1. Introduction

Inspection planning deals with finding a shortest inspection path such that all points of the workspace $\mathcal{W}$ are "seen" from the path. The problem is studied in mobile robotics in which a path for a mobile robot performing the inspection is planned in a priori known map of the environment. The map can be represented by the polygonal domain, which makes the problem close to computational geometry. A practical consideration of mobile robot sensing capabilities leads to two types of sensing: discrete and continuous. These sensing models are motivated by the cost of sensing and the cost of motions. The continuous sensing is suitable for problems where the cost of the sensing is relatively cheap in comparison to the cost of the motion. Contrary to the discrete sensing model, where the cost of the sensing is dominant, and the cost of the motion can be ignored. The combination of both costs is a difficult problem and it remains largely unexplored [1].

The inspection planning problem formulations can be found in computational geometry. The problem with continuous sensing can be formulated as the *Watchman Route Problem* (WRP) [2]. The WRP is a problem of finding a closed shortest path in the polygonal domain $\mathcal{W}$ such that all points of $\mathcal{W}$ are visible from at least one point at the path. Even though optimal algorithms for restricted class of polygons have been proposed, the WRP is NP-hard for $\mathcal{W}$, and probably the first heuristic approach has been introduced in [3].

A decoupled approach can be used to address the inspection planning with discrete sensing. The problem is decomposed into the set cover problem and the consecutive multi-goal path planning problem. For the polygonal domain $\mathcal{W}$, the set cover problem can be formulated as the *Art Gallery Problem* (AGP). The AGP stands to find a minimal number of guards to cover $\mathcal{W}$. The guards represent sensing locations, and each guard covers a part of the environment by its star-shaped visibility polygon. The AGP is NP-hard even for a polygon without holes [4]. The multi-goal path planning problem can be formulated as the well-known *Traveling Salesman Problem* (TSP) if all paths between sensing locations are known [5, 6].

The AGP and WRP are studied in the computational geometry domain for an unrestricted model of visibility. However, sensing (visibility) of real sensors (cameras or range finders) is limited, e.g., in sensing range and frequency. To distinguish the restricted visibility, authors of [1] call the problem of finding sensing locations *sensor placement* rather than the AGP. Similarly, the WRP with restricted visibility range to a distance d is called d-*Watchman Route Problem* (d-WRP) [7]. These variants of the problems with the restricted visibility range also belongs to the NP class; thus, approximate solutions are more suitable for real application to get "good" solutions with "reasonable" computational requirements.

The decoupled approach provides a feasible solution of the inspection planning, and has been used in robotic tasks [8, 9]. Sensing locations for restricted visibility range can be found by different techniques [10]. The TSP can

be solved by various approaches from the operational research [11], or by soft computing techniques like ant colony system [12], self-organizing map (SOM) approaches [13], or immune system [14]. On the other hand, the WRP in $\mathcal{W}$ has been addressed (to the best of our knowledge) only by the heuristic approach presented in [3]. The algorithm is based on a set of static guards that are used to determine the minimum spanning tree from the pairwise shortest paths between guards. The tree is split to construct a route that is shortened by vertex substitutions and removing of redundant vertices. Even though the approach is based on guards, solutions have been presented only for an unrestricted visibility range. In [15], a SOM based approach for the d-WRP has been presented; thus, SOM provides solutions for both inspection planning variants. However, the main difficulty of SOM application in the polygonal domain is determination of the shortest paths among obstacles, which is more computationally demanding than a pure computation of the Euclidean distances between neurons' weights and an input vector.

In this paper, SOM is applied to the inspection planning problem with discrete and continuous sensing in the polygonal domain. A new adaptation schema is proposed and compared with an already available schema for the TSP [16] in a set of problems created from a map of real environments and several visibility ranges. The main contribution of this paper is new adaptation schema for the multi-goal path planning problem in the polygonal domain that can be used to addressed the non-Euclidean TSP and d-WRP, i.e., inspection planning with continuous sensing.

The rest of this paper is organized as follows. The next section provides overview of the addressed problem. The related work is presented in Section 3. The proposed adaptation schema for the multi-goal path planning problem is presented in Section 4. Then, the schema is applied to the inspection planning with continuous sensing in Section 5. Experimental results of the proposed algorithms are presented in Section 6. Concluding discussion and remarks of the future work are presented in Section 7. The list of the used symbols is presented in Section 8.

2. Problem Statement

An environment to be inspected by a mobile robot is a priori known, and a polygonal map of the environment is available. The robot is equipped with an omnidirectional sensor with a sensing range restricted to a distance d. The notion of d-visibility is assumed as follows. Two points p and q in a polygon $\boldsymbol{P}$ are called *d-visible*, if the line segment joining them is contained in $\boldsymbol{P}$, and if the segment length is less or equal to d. The sensor coverage is modeled by a disk with the radius d. A point robot is assumed, and a path between two points in the polygonal domain $\mathcal{W}$ consists of sequence of straight line segments joining the points and vertices of $\mathcal{W}$, and all segments are entirely inside $\mathcal{W}$. The addressed variants of the inspection planning are following.

Discrete Sensing - The whole environment is covered by performing a finite number of measurements with the range d at sensing locations. A set of such sensing locations $\boldsymbol{G}$ is given, and all locations are reachable by the mobile robot. The problem is to find a closed path (possibly a shortest one) connecting all sensing locations. The problem is the multi-goal path planning problem that is considered as the non-Euclidean TSP in the polygonal domain.

Continuous Sensing - Measurements can be taken along a path in the continuous sensing problem variant, therefore, sensing locations are not explicitly prescribed. The problem is to find a closed (possibly a shortest one) path such that each point of the environment is d-visible from some point of the path. The problem is formulated as the d-WRP in the polygonal domain.

Even though the cost of the sensing and the cost of the motion can be considered in the inspection planning, only the length of the inspection path is used as the quality metric in this paper. Mainly because a SOM approach for the multi-goal path planning provides an approximate solution of the related TSP, and a shorter path is a plus. Besides, the decoupled approach can also be used for the d-WRP. Having a prescribed set of sensing locations, eventually the smallest set, only the length of the path can be minimized. Therefore, the length of the inspection path as the only metric makes sense for both inspection variants.

3. Related Work

3.1. Reference Algorithm

To compare the solution quality of the examined algorithms the following decoupled approach is used to find a reference solution. A deterministic sensor placement algorithm [17] is used to find a set of sensing locations. The algorithm is based on a decomposition of $\mathcal{W}$ into a set of convex polygons. First, Seidel's algorithm [18] is used to find the primal convex partition. Convex polygons of the partition are eventually divided into convex sub-polygons if a convex polygon cannot be covered from one point with the d-visibility. The required computational time is proportional to the number of found sensing locations [17].

The inspection path is found as the solution of the TSP on a graph $G(V, E)$, where V stands for sensing locations, and E is the set of edges with costs computed as the length of the shortest path between the sensing locations. The paths are found by Dijkstra's algorithm in $O(nn_e \log(n+v))$ on the visibility graph, which is found in $O((n+v)^2)$ [19], where v denotes the number of polygon vertices, n is the number of sensing locations, and n_e is the number of edges of the visibility graph. Without loss of generality $G(V, E)$ is assumed to be complete. An optimal solution of the TSP is found by the `concorde` solver [20].

3.2. SOM Procedures for the TSP

The basic idea of SOM for the TSP is based on Kohonen's two-layered unsupervised neural network in which

the first layer represents coordinates of the presented goals to the network. The second layer consists of neurons organized in a cycle (ring), and each neuron is connected with the first layer. The weights of the connections represent coordinates of the node, see Fig. 1. The adaptation schema is an iterative two phases procedure. At each iteration, goals are presented to the network in a random order, and a winner node is found for each goal in the competitive phase. The winner selection uses the Euclidean distance of a node to the goal. The winner node and its neighbouring nodes are adapted towards the goal in the cooperative phase. The adaptation process is typically terminated if winner nodes are sufficiently close to the goals.

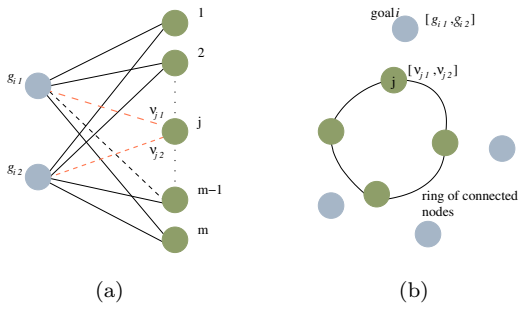


Figure 1: Schema of the two-layered neural network and associated geometric representation.

The TSP has been addressed by several SOM approaches during the last two decades. The pioneering work of Angéniol et at. [21] and Fort [22] in 1988 has been followed by particular improvements in the quality of found solutions and required computational time. An inhibition mechanism, which prevents nodes to win too often, has been used in [16]. In [23], authors consider creation/deletion of nodes, and selection of winners based on the shortest path to the segment joining two neighbouring nodes. Several nodes initializations have been examined in [24] and [25].

Probably the most complex algorithm is the Co-adaptive net that is extensively evaluated in [13]. The authors of the Co-adaptive net reported that their approach together with [16] provide superior results in various instances of the TSP from the TSPLIB [26]. However, the high number of the Co-adaptive net parameters can be consider as a drawback.

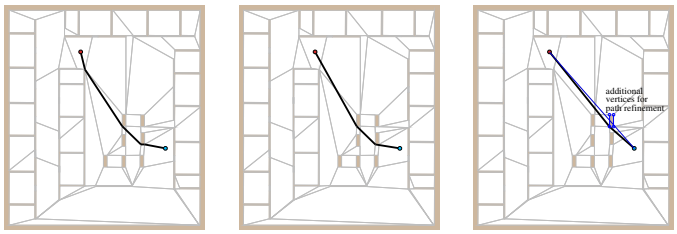
The algorithm proposed by Somhom et al. [16], denoted as SME in this paper, is particularly interesting. It provides competitive quality of solutions to the Co-adaptive net algorithm, but it is less complex, and it depends on a less number of parameters. In this paper, the SME algorithm is considered as the reference adaptation schema for the initial application of SOM to the inspection planning in $\mathcal{W}$. The algorithm works as follows. Nodes are initialized as a small ring around the center of the goals. The winner node is selected according to $\nu^\star = \text{argmin}_\nu |g, \nu|$, where $|.,.|$ denotes the Euclidean distance between the goal g and the node ν for the Euclidean TSP. Each node can be a winner only once in each adaptation step; thus, a winner is inhibited after its selection. The inhibition is cleaned at the begin of the next iteration, i.e., new presentation of goals to the network. The adaptation rule moves the winner node and its neighbouring nodes towards the presenting goal g according to $\nu'_j = \nu_j + \mu f(\sigma, l)(g - \nu_j)$, where μ is the fractional learning rate. The neighbouring function is $f(\sigma, l) = exp(-l^2/\sigma^2)$ for $l < \delta$, and $f(\sigma, l) = 0$ otherwise, where σ is the gain parameter, l is the distance in the number of nodes measured along the ring, δ is the size of the winner node neighbourhood that is set to $\delta = 0.2m$, where m is the number of nodes set to $m = 2n$ for n goals. The initial value of σ is set proportionally to the problem size $\sigma_0 = 0.06 + 12.41n$, and it is decreased at the end of each adaptation step according to $\sigma = \sigma(1-\alpha)$, where α is the gain-decreasing rate. The values of learning and decreasing rates are $\mu = 0.6$ and $\alpha = 0.1$ [27], respectively. The adaptation is terminated if all winners are in a distance less than $\epsilon = 0.001$.

Regarding the number of parameters authors of [28] proposed alternative adaptation rules to Kohonen's exponential evolution. To avoid initial values of the learning and gain-decreasing rates, the authors proposed simplified adaptation rules based only on the number of performed adaptation steps k. The learning rate is defined as $\mu = 1/\sqrt[4]{k}$, and the learning gain as $\sigma = \sigma(1-0.01k)$. The initial value of the gain is $\sigma_0 = 10$. For small values of σ, the value of the neighbouring function is very small; thus, the neighbouring nodes are negligibly moved. To decrease the computational burden, the authors recommended to gradually decrease the neighbourhood of the winner node after each adaptation step. The recommended initial value of the neighbourhood is $\delta = 0.4m$ that is decreased according to $\delta = 0.98\delta$ at the end of each adaptation step.

3.3. Approximation of the Shortest Path in $\mathcal{W}$

The main difficulty of SOM application to problems in the polygonal domain $\mathcal{W}$ is a determination of the shortest path among obstacles, which can be computationally intensive. In [29], a simple, yet sufficient approximation has been applied to the self-organizing adaptation procedure. It is based on a convex partition of $\mathcal{W}$. The partition $\boldsymbol{P}$ is a set of disjoint convex cells $\boldsymbol{P} = \{C_1, C_2, \ldots, C_k\}$ such that the union of the cells is $\mathcal{W}$. The cells are induced by the diagonals of $\mathcal{W}$, and each cell is formed from a sequence of $\mathcal{W}$ vertices. During the adaptation, nodes are inside $\mathcal{W}$, and therefore, they are always inside some cell. A collision free path for two points p_1 and p_2 that are inside cells $p_1 \in C_1$ and $p_2 \in C_2$ can be found as a path over the cells' vertices $v_1 \in C_1$ and $v_2 \in C_2$. The vertices are selected to minimize the length of the path $|p_1, v_1| + |S(v_1, v_2)| + |v_2, p_2|$, where $|.,.|$ denotes the Euclidean distance of two points, and $|S(.,.)|$ is the length of the shortest path between two vertices. Such a path can be further refined by consideration of direct visibility from the

particular point to a vertex of the path. The used direct visibility test is similar to the method [30], a convex partition is used rather than a triangulation. An additional improvement of the approximate path can be achieved if vertices of obstacle edges that intersect the direct line segment from p_1 to p_2 are considered in the construction of the primal path. An example of the primal path and its refined variants is shown in Fig. 2.



(a) (b) (c)

Figure 2: Approximate path between two points; (**a**) rough approximate path over cells' vertices, (**b**) refined path, (**c**) refined path with consideration of detected obstacle's vertices.

The used supporting structures are a convex partition of $\mathcal{W}$, and all shortest path between v vertices of $\mathcal{W}$. A convex partition can be found in $O(v \log v)$ [18]. The shortest path can be pre-computed by Dijkstra's algorithm using the visibility graph. The problem of finding the cell C_ν is the point-location problem, which can be solved in $O(\log v)$. Besides, the cell can be determined during the node movement towards the goal by the walking technique similar to [31]. The complexity of such cell determination is bounded by $O(\log n_d)$, where n_d is the number of passed diagonals of the used convex polygon partition.

In the TSP, goals are fixed, and therefore, all shortest path from the map vertices to the goals can be pre-computed, which reduces the required computational time for the adaptation at the cost of higher memory requirements. For large sets of goals, the approach visualized in Fig. 2 can be more appropriate, as it provides approximate path for two arbitrarily placed points in $\mathcal{W}$, and its space requirements depends only on the number of $\mathcal{W}$'s vertices.

3.4. SOM Procedure for the TSP in $\mathcal{W}$

An application of the SME adaptation schema to the TSP in $\mathcal{W}$ has been presented in [32]. The main difference to the algorithm for the Euclidean TSP is in consideration of the approximate path found by the above described procedure, in particular using the pre-computed shortest paths from the vertices to the goals. The path is used in the select winner part to determine distance of the node to the presented goal, and during the adaptation when nodes are moved towards the goal. Besides, the following modifications have been applied.

The termination condition also considers a maximal number of adaptation steps. An error of the path approximation can cause that the winner–goal distance is not effectively decreased during the adaptation, which can lead to a distance higher than the required ϵ. Even though such a convergence issue has been observed only for the rough path approximation, the maximal number of the iterations is advantageous as it guarantees termination of the algorithm.

A practical implementation of the select winner procedure can utilize the Euclidean distance to inform the winner searching process, and to decrease the computational burden. During the searching, all non inhibited nodes are examined, and the closest node is selected as the winner. Let ν_g be an actual winner candidate to the presented goal g. A distance of a node ν to g as a length of the path among obstacles is determined only if the Euclidean distance $|\nu, g|$ is shorter than $|\nu_g, g|$. This technical improvement does not affect the quality of solution. However, it has been observed that it provides solution up to two times faster for the SME adaptation schema.

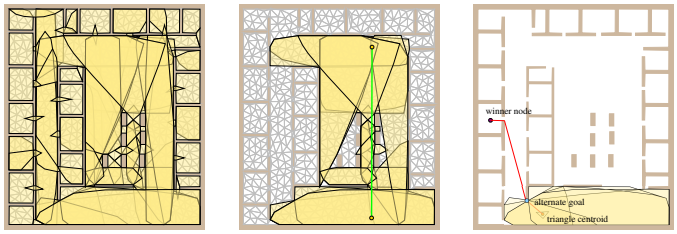
In the original SME algorithm [16], the initial values of nodes are placed around a center of goals, which cannot be used in the $\mathcal{W}$ because such a center can be in an obstacle. Based on experimental results with various initialization it has been observed that the SME schema is insensitive to the initial point around which the ring is created. Thus, to ensure that nodes are placed in $\mathcal{W}$ they are placed around the first goal as a small ring with the radius 0.5 cm. The sufficient free space around the first goal is assumed. In the case of inspection planning the space around the goal is ensured by the sensor placement algorithm.

3.5. SOM Procedure for the WRP in $\mathcal{W}$

The SOM procedure for the WRP has been presented in [15]. The main idea of the procedure is that the ring of nodes represents the watchman route itself, and the nodes are adapted towards uncovered parts of $\mathcal{W}$. Determination of the ring coverage is based on approximation of the continuous sensing along a straight line segment using a convex cover set. The cover set consists of a set (possibly overlapping) convex polygons, which dimensions are restricted to respect the limited visibility range d. A triangular mesh of $\mathcal{W}$ is used to support fast determination of incident convex polygons with a segment. A convex cover set is found on top of the mesh, i.e., a convex polygon of the set consists of mesh vertices, see Fig. 3a.

It is not required to have a minimal number of the convex polygons, because the cover set is used as follows. Each triangle is associated with at least one convex polygon, and each convex polygon has associated a set of triangles that are entirely inside the polygon. For a straight line segment lying in $\mathcal{W}$, all incident triangles are found by the walking in triangulation technique [31]. From these triangles, all associated incident convex polygons are found, and the coverage along the segment is determined as a union of all triangles associated to the incident convex polygons, see Fig. 3b. The coverage of the ring is determined from the

sequence of straight line segments joining each neighbouring nodes found by the approximation of the shortest path between two points in $\mathcal{W}$.



(a) (b) (c)

Figure 3: Supporting structures for the WRP; (**a**) a convex cover set and an underlying triangular mesh, (**b**) an incident convex polygons with a straight line segment, (**c**) an alternate goal.

The adaptation procedure follows the SME schema for the TSP in $\mathcal{W}$. Centroids of the mesh triangles are used as goals presented to the network. However, a winner node is found only for triangles that are not covered. A coverage from the ring is determined at the beginning of each adaptation step. During the adaptation, the coverage is updated by adding all triangles associated to the convex polygons that are incident with the presented triangle (centroid) after the winner node adaptation towards the centroid. The additional modification of the adaptation rule relates to the visibility nature of the WRP. To see the presented triangle from the ring, it is sufficient if the winner reaches some of the incident convex polygons associated to the triangle. Thus, an alternate goal is found from the intersection of the path from the node to the triangle centroid with the incident convex polygons, see Fig. 3c.

The adaptation is terminated if the ring covers all triangles or after 180 adaptation steps, which can lead to an incomplete coverage.

4. Proposed Adaptation Schema for the Multi-Goal Path Planning

In this section, a new adaptation schema for the multi-goal path planning problem is proposed. The schema follows the SME adaptation schema, particularly the algorithm described in Section 3.4, but the main difference is in the winner selection rule. The selection utilizes a creation/deletion mechanism, which is similar to the one used in [21], nevertheless it is also inspired by the approach [23]. Beside the selection rule, particular parts of the algorithm have been improved considering modifications of the aforementioned approaches proposed by several authors, and experimental evaluation of adaptation parameters settings. The proposed selection rule together with the improvements lead to a new adaptation schema for the inspection planning with discrete sensing that provides better solutions, and has lower computational requirements than the former schema. To provide an overview of the proposed algorithm the adaptation schema is depicted in Algorithm 1. The selection rule and the particular improvements are described in the following subsections.

Algorithm 1: SOM Adaptation Schema for the TSP

Input: $\boldsymbol{G} = \{g_1, \ldots, g_n\}$ - a set of goals
Input: $(m, \sigma, \mu, \alpha, \delta, s)$ - the adaptation parameters
Input: ϵ – the maximal allowable error
Input: σ_{min} – the minimal allowable σ

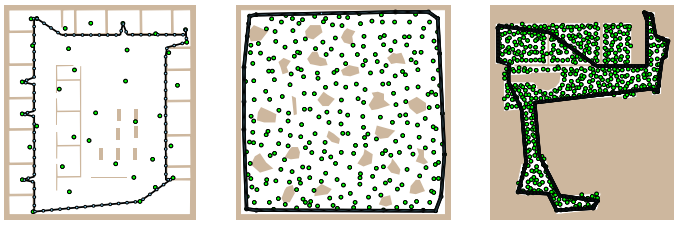
```
init(ν_1, ..., ν_M)   // initial set of neurons weights
i ← 0                 // set the adaptation step
repeat
    error ← 0
    I ← ∅             // set of inhibited nodes
    Π(G) ← a random permutation of goals
    foreach g ∈ Π(G) do
        ν* ← select winner(node to g), ν* ∉ I)
        error ← max{error, |ν*, g|}
        adapt(ν*, g)      // call the adapt procedure
        I ← I ∪ {ν*}      // inhibit winner node
    k ← k + 1             // increment the adaptation step
    update_adaptation_parameters(σ, δ)
until error < ε or σ < σ_min
```

4.1. Initialization

Several initialization of the neurons weights have been proposed by various authors, e.g., a small ring around centroid of the goals [16], a tour found by the nearest neighbourhood [25], or convex hull of the goals [33]. In the polygonal domain, these initialization methods cannot be directly used, as the initial weights must be inside $\mathcal{W}$. Based on experimental results, superior solutions have been achieved by the following modifications of the convex hull initialization. First, a convex hull of the goals is found without consideration of obstacles. After that, goals forming the hull, i.e., goals that are at the hull border, are connected by the shortest paths found using the visibility graph. So, a tour over the forming goals is constructed. Finally, nodes are equidistantly placed at the tour, starting at a random point of the tour. Examples of connected initial rings of nodes are shown in Fig. 4.

4.2. Winner Selection

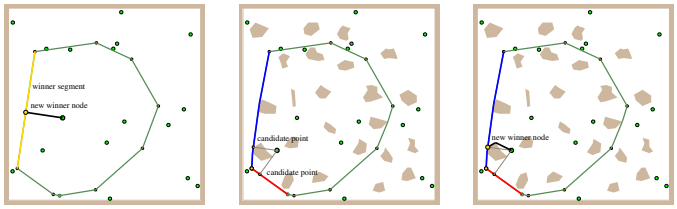
An idea behind the proposed winner selection method is based on consideration of a path between two nodes. The principle is shown in Fig. 5a for a problem without obstacles. The closest segment connecting two nodes is found instead of the closest winner. Then, the closest point at the segment is determined. If the point is different from the segment endpoints a new node is created with the point coordinates and added to the ring. Otherwise the closest



(a) *jh*, 36 goals (b) *potholes*, 281 goals (c) *ta*, 574 goals

Figure 4: Examples of initial rings of nodes in environments *jh*, *potholes* and *ta*; green disks represent goals and blue disks are nodes.

node is a candidate to be the winner. If the winner candidate is inhibited a new node is created with the identical values of the winner weights. The newly created node becomes the winner of the current selection. The winner node is then adapted towards the presented goal.



(a) winner segment (b) candidate points (c) winner node

Figure 5: A principle of the proposed winner selection rule.

The above described procedure can be effectively used for problems without obstacles, but determination of the closest segment to a point in $\mathcal{W}$ is more complex. That is why the following approximation is used. A regular winner node candidate is found using the approximation of the shortest path from a node to the goal. Then, two paths connecting the winner with its neighbouring nodes are determined as approximate paths between two points in $\mathcal{W}$. For each of the paths, the closest segment point to the goal is determined using the Euclidean distance. These two points become candidates to be a winner of the current selection, see Fig. 5b. Due to obstacles, the candidate points can be farther than the winner candidate. Therefore a path from each candidate point to the goal is determined. If the length of the path for one of the points is shorter than the winner candidate distance to the goal, the corresponding point is used to create a new node. If it is not the case a new node is created if the winner node candidate is inhibited. The newly created node is the winner, otherwise the winner node candidate becomes the winner.

New nodes are created during the selection of winners, which can increase the computational burden. To remove unnecessary nodes a deletion mechanism is based on moving activity of nodes. The nodes that are not moved (adapted) in the last s adaptation steps are removed from the ring.

4.3. Adaptation

A winner node and its neighbouring nodes are moved towards the goal in the `adapt` procedure. A node ν is moved along the approximation of the shortest path $S(\nu, g)$ to the goal g by a distance $\beta|S(\nu, g)|$, where $\beta = \mu f(\sigma, l)$. The value of $f(\sigma, l)$ decreases with increasing distance of the node from the winner (in the number of nodes) and the number of adaptation steps, as σ is decreased. In final adaptation steps, the value of β is very small, and the movement can be negligible. Considering this observation the neighbouring nodes of the winner node are moved towards the goal only if $\beta > 10^{-5}$. This adaptation rule is denoted as $\beta - condition$ in this paper.

Performed experiments show that this modification does not decrease the solution quality and increases speed of the algorithm two times for problems with about five hundreds goals. Even though a distance from a node to the goal is determined in the select winner procedure, the movement of the node is more computationally demanding. A path as a sequence of $\mathcal{W}$ vertices is not needed in the distance query, but it is required for the node movement, where a new node position at the path is determined. Therefore, the path is found in the adapt procedure before the node adaptation.

4.4. Adaptation Parameters

Authors of [28] proposed adaptation rules that are derived from the number of performed adaptation steps, which dramatically decrease the required number of adaptation steps to find a stable solution. However, for problems in $\mathcal{W}$ and with combination of the select winner procedure described in Section 4.2 better solutions are achieved with a slower decreasing σ. Also the solution quality is increased for a fixed value of the learning rate μ. The algorithm performance is also affected by the size of the winner node neighbourhood denoted as δ in this paper. Due to the used node creation/deletion mechanism, the number of nodes varies in each adaptation step. Rules that derive δ from the number of nodes can lead to a large winner's neighbourhood, and the proposed decreasing δ in [28] is not effective. So, the maximal value of δ is restricted to $\delta_m = 2n/8$, which corresponds to $m = 2n$ initial nodes and the neighbouring factor $f = 8$.

A summary of the used adaptation parameters is as follows. The initial values of the parameters are: σ=10, μ=0.6, m=2n, f=8, δ=m/f, α=0.1, s=8. Values of δ and σ are changed (in the procedure `update_adaptation_parameters`) after each complete presentation of goals as follows:

- $\sigma \leftarrow \sigma(1 - 0.001k)$ - decrease the learning gain,
- $d \leftarrow 0.99^k \min\{m/f, 2n/f\}$,

where m is the actual number of the nodes and k the actual number of the performed adaptation steps.

4.5. Termination Condition

The initial value of the learning gain σ is independent to the problem size. Therefore, instead of a maximal number of adaptation steps the adaptation procedure can be terminated if σ is below given threshold σ_{min}. The selected value is $\sigma_{min} = 10^{-4}$, for which the value of the neighbouring function is small, and the neighbouring nodes are practically not moved. This termination condition is more intuitive and problem size independent contrary to the used maximal number of steps for the SME schema. Even though the adaptation is terminated before *error* is below the selected ϵ, the inhibition mechanism guarantees that all goals have associated distinct nodes. So, the final inspection tour over the goals is retrieved by traversing the ring.

4.6. Discussion

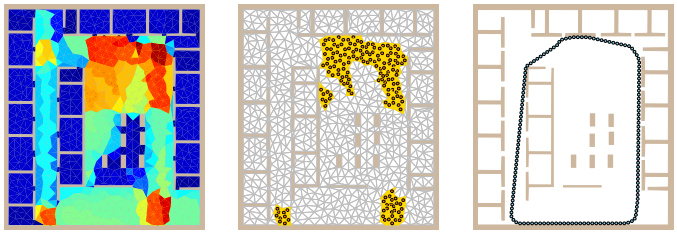
A collection of the presented adaptation rules provides new adaptation schema in which the number of nodes is not explicitly restricted, which is one of the benefit over the SME adaptation schema. Here, it should be mentioned that the particular rule (modification) can be used in other SOM approaches, e.g., the proposed $\beta - condition$. The rules can decrease the computational burden; however, they do not necessary improve the solution quality. During the evaluation of the rules, it has been observed that the proposed *hull* initialization does not improve solutions if the SME adaptation parameters are used. Moreover, the SME schema seems to be insensitive to the initial values of neurons weights. The evaluation has been performed in a set of 21 inspection problems that represent instances of the non-Euclidean TSP. After this evaluation, the parameters presented in Section 4.4 have been selected.

5. Adaptation Schema for the Inspection Planning with Continuous Sensing

The adaptation schema proposed in Section 4 has been applied to the algorithm for the d-WRP [15] briefly described in Section 3.5. The proposed schema has to be modified, because the solution of the d-WRP is represented by the ring of nodes itself, i.e. a sequence of straight line segments connecting the nodes. This aspect is considered in the following schema adjustments.

5.1. Initialization

Triangles, or more concretely their centroids, of the supporting triangular mesh are used as goals presented to the network. The nodes can be very close to the border polygon of $\mathcal{W}$ if a convex hull of the all centroids is used for the initial construction of the ring. It is because small triangles are typically located at corners. In addition, once a part of $\mathcal{W}$ is covered by some nodes, a winner is not selected to the particular goal, and the nodes lying in the part are moved only as neighbourhoods of another winner node. Such initially placed nodes can lead to an unnecessary long inspection path. To avoid such initialization only selected triangles are considered for the convex hull construction. The selection is based on an idea that if a ring starts from parts that are visible from large portion of $\mathcal{W}$, then nodes will be attracted to other locations, and the parts will be covered by the segments connecting two neighbouring nodes.



(a) (b) (c)

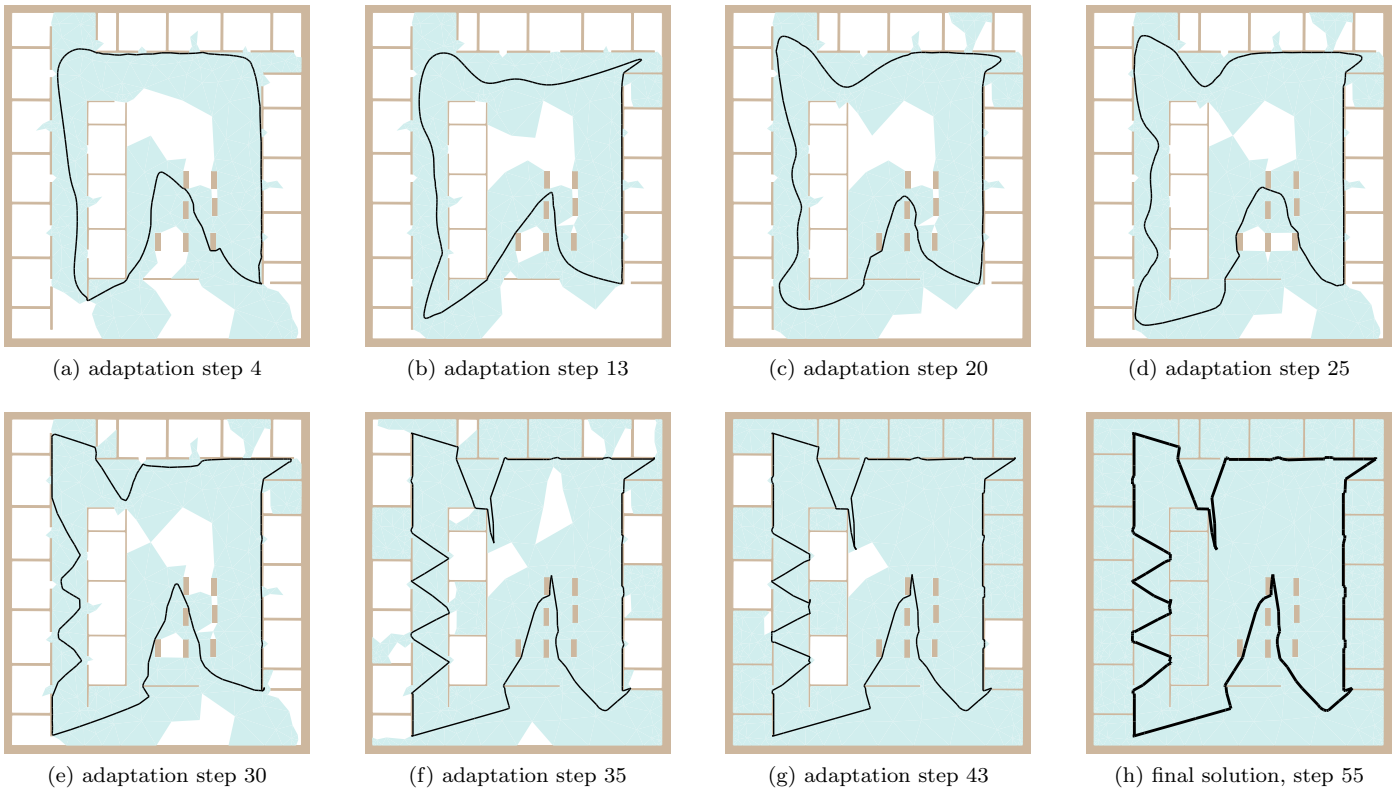
Figure 6: An example of nodes initialization; (**a**) visualization of the triangle coverage, the highest coverage is in red (light), while the triangles that are incident with the smallest number of convex polygons are in blue (dark); (**b**) selected triangles (centroids) with highest coverage; (**c**) initial ring created from the convex hull of the selected triangles.

First, for each triangle an area visible from the triangle is determined from the associated convex polygons of the cover set. The area is the sum of the areas of all triangles associated to the polygons. A visualization of the triangles visible areas is shown in Fig. 6a. Centroids of triangles with the largest visible area are selected for the convex hull construction, see Fig. 6b. Let the visible area of the i^{th} triangle be a_i, and a_{max} be the largest visible area. All triangles with $a_i \geq a_{max} - a_t$ are selected. After the selection, a convex hull of the triangles' centroids is created, and the centroids at the hull border are connected by the approximate shortest paths. Then, nodes are placed at the paths like in Section 4.1, see Fig. 6c.

The threshold value a_t can be set individually for a particular problem, as the triangular mesh provides only approximation of the coverage. However, the median of the visible areas provided the best results in the experimental evaluation. The initial number of nodes m is set to $m = 0.1n$, where n is the number of centroids (triangles), because n is typically much higher than the number of sensing locations in the discrete inspection planning.

5.2. Three Phases Adaptation

The WRP algorithm tries to cover all triangles, and the adaptation is terminated if all triangles are covered. The adaptation procedure avoids selection of winner nodes for triangles that are already covered by nodes, or are covered from a path connecting the nodes. This is an important distinction in comparison to the multi-goal path planning. Also nodes deletion can decrease the ring coverage, and



(a) adaptation step 4 (b) adaptation step 13 (c) adaptation step 20 (d) adaptation step 25

(e) adaptation step 30 (f) adaptation step 35 (g) adaptation step 43 (h) final solution, step 55

Figure 7: An example of ring evolution and its coverage in the WRP, environment *jh* and visibility range 5 m

in a consequence it can lead to a convergence issue. In final adaptation steps, σ becomes low, and if some nodes are deleted the network does not effectively adapt to cover all triangles, because the adaptation is terminated due to σ_{min}. Therefore the deletion rule cannot be simply used during the whole adaptation.

The following three phases adaptation is proposed to increase coverage of the final solution:

1. the creation/deletion rule described in Section 4.2 is used until 85% of triangles are covered, then
2. the creation rule is used without the deletion until 95% coverage is reached.
3. The only winner node is selected without node creation in the final adaptation steps.

In the first phase, the ring is spread around the environment to cover most of the space while only the active nodes are preserved. The second phase is typically active only for several adaptation steps in which the number of nodes is increased. To decrease the computational burden, the number of neighbouring nodes δ is not increased in the second phase. The value of δ is computed as $\delta_I = m_I/f$, where m_I is the number of nodes in the last step of the first phase after the deletion. Nevertheless, δ is regularly decreased after each adaptation step by the rule $\delta = 0.99^k\delta_I$. The newly created nodes in the second phase support local searching that is finalized in the third stage.

An example of the ring evolution and the ring coverage is depicted in Fig. 7. Notice that the space is almost covered in the step 43; however, additional 12 steps are needed to achieve the full coverage. The advantage of the proposed adaptation rules is that these steps are performed very quickly, because the network adapts only to the uncovered triangles.

6. Experiments

The proposed SOM adaptation schema has been evaluated in a set of inspection planning problems for two types of sensing. The discrete sensing is considered as the multi-goal path planning problem formulated as the TSP, and the d-WRP formulation is used for the continuous sensing. The proposed algorithms are compared with the SME adaptation schema, in particular the schema is used in the TSP algorithm [32] and the d-WRP algorithm [15]. However, the Euclidean pre-selection (Section 3.4) and the $\beta-condition$ (Section 4.3) are considered in the algorithms to decrease the computational burden without noticeble changes to the solution quality. Besides, the reference solutions of the examined problems are found as solutions of the decoupled approach using the `concorde` solver, see Section 3.1. The full path refinement of the approximate shortest path in $\mathcal{W}$ is used in all algorithms; the node-goal paths are used for the TSP while approximate shortest paths between two points are used for the d-WRP.

The SOM algorithms are randomized, and therefore, twenty solutions are found for each problem and particular algorithm. The quality of solution is measured as the percent deviation to the reference path length of the mean solution value, $PDM = (\overline{L} - L_{ref})/L_{ref} \cdot 100\%$, and as the percent deviation from the reference of the best solution value (PDB), where L_{ref} is the length of the reference path. Besides, $s_L\%$ is denoted to the percent sample variance of the path length to the mean solution value.

All algorithms have been implemented in C++ and compiled by the G++ 4.2 with the -O2 optimization flag. All results have been obtained within the same computational environment using single core of the Athlon X2 5050e CPU at 2.6 GHz and 2 GB RAM running FreeBSD 8.1. Thus, all presented required computational times can be directly compared.

The proposed adaptation schema has been studied in a set of multi-goal path planning problems, the experimental results for the final found parameters are presented in the next subsection. Then, the parameters found in the evaluation of the multi-goal path planning have been used in the experimental evaluation of the inspection planning problems for various visibility ranges. The results are presented in Section 6.2.

6.1. Multi-Goal Path Planning - the non-Euclidean TSP

The examined multi-goal path planning problems consist of polygonal maps and a set of goals (sensing locations). The maps represent real and artificial environments used for examination of path and motion planning approaches. The used approximation of the shortest path depends on the numbers of vertices and convex polygons, therefore, to provide an overview of maps' relation to the algorithm performance, the basic maps properties are depicted in Table 1. Moreover, all paths from vertices to goals are pre-computed for the node–goal path approximation.

Detailed experimental results of the SME schema and the proposed adaptation schema are shown in Table 2. The most time consuming preparation step is computation of all shortest paths from vertices to all goals, the time is denoted as T_{init} in the table. Construction of the supporting convex partition, and the visibility graph is negligible in comparison to the required computation time of the adaptation T_a. For the largest problem $h2_2$ the convex partition is found in 220 milliseconds, and the visibility graph is found in 150 milliseconds.

The proposed adaptation schema provides solutions with higher quality than the SME schema. The required computational times are not significantly different for small problems, but for larger problems the proposed algorithm provides better solution in less computational time. An overview of the algorithms performance as average values of the solution quality measured by the PDM, and average values of the required computational time including T_{init} are shown in Fig. 8 and Fig. 9 as histograms

Table 1: Map Properties

Name	**Dimensions** [m × m]	**Area** [m^2]	v	h	p
jari	4.5 × 4.9	20	48	1	14
complex2	20.0 × 20.0	322	40	3	21
m1	4.8 × 4.8	20	51	4	26
m2	4.8 × 4.8	15	51	6	20
map	4.8 × 4.8	14	68	8	36
potholes	20.0 × 20.0	367	153	23	75
rooms	20.0 × 20.0	351	80	0	33
a	8.9 × 14.1	71	99	6	22
dense	21.0 × 21.5	299	288	32	150
m3	4.8 × 4.8	17	308	50	120
warehouse	40.0 × 40.0	1192	142	24	83
jh	20.6 × 23.2	455	196	9	77
pb	133.3 × 104.8	1453	89	3	41
ta	39.6 × 46.8	731	74	2	30
h2	84.9 × 49.7	2816	2062	34	476

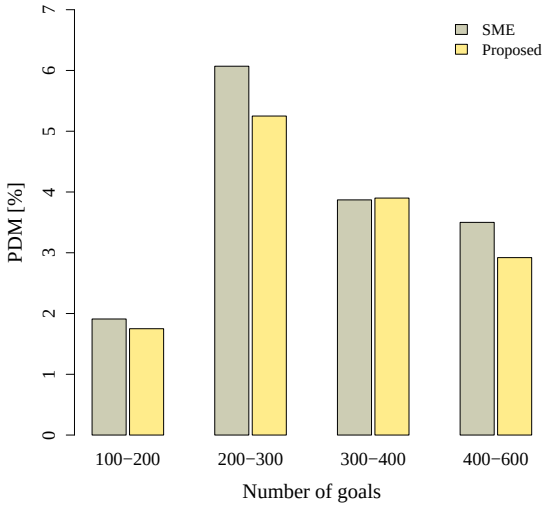


Figure 8: Average values of the solution quality for the multi-goal path planning problems.

for the number of goals. Examples of found solutions are depicted in Fig. 10.

Here, it should be noted that for SME schema various initialization have been considered. Also the number of nodes has been increased up to $m = 3n$. However, the changes of the solution quality are below s_L, and only the computational requirements are increased for higher values of m. Based on these observations, $2n$ nodes are initialized as a small ring around the first goal for the SME algorithm in the all presented experimental results.

Table 2: Experimental results for the Multi-Goal Path Planning

Problem	n	L_{opt} [m]	T_{init} [s]	SME m	SME PDM	SME PDB	SME s_L%	SME T_a [s]	Proposed m	Proposed PDM	Proposed PDB	Proposed s_L%	Proposed T_a [s]
jari	6	13.6	0.001	12	0.00	0.00	0.00	0.007	6	0.81	0.00	1.38	0.002
complex2	8	58.5	0.003	16	0.00	0.00	0.00	0.014	19	0.00	0.00	0.00	0.005
m1	13	17.1	0.006	26	0.03	0.00	0.09	0.029	17	0.07	0.00	0.14	0.016
m2	14	19.4	0.005	28	7.29	0.00	3.25	0.035	19	9.16	6.02	1.86	0.018
map	17	26.5	0.010	34	2.13	0.00	2.61	0.062	29	3.08	0.00	2.15	0.029
potholes	17	88.5	0.046	34	0.83	0.00	0.85	0.074	31	0.75	0.00	1.66	0.034
a	22	52.7	0.022	44	0.11	0.00	0.25	0.118	36	0.04	0.00	0.15	0.047
rooms	22	165.9	0.016	44	0.83	0.11	0.70	0.141	25	1.36	0.17	0.79	0.058
dense$_4$	53	179.1	0.198	106	13.76	5.85	3.77	1.092	188	8.85	4.85	2.02	0.299
potholes$_2$	68	154.5	0.092	136	5.61	2.75	1.20	1.461	154	4.43	2.37	1.13	0.424
m3$_1$	71	39.0	0.245	142	7.06	4.34	1.21	3.327	396	5.82	4.51	1.11	0.456
warehouse$_4$	79	369.2	0.074	158	6.27	2.20	2.39	2.091	233	4.57	2.28	1.27	0.534
jh$_2$	80	201.9	0.116	160	1.63	0.35	0.67	2.122	228	1.48	0.43	0.71	0.534
pb$_4$	104	654.6	0.043	208	0.64	0.05	0.28	2.846	443	0.10	0.00	0.11	0.578
ta$_2$	141	328.0	0.070	282	3.04	2.11	0.49	5.154	421	3.21	2.14	0.65	0.918
h2$_5$	168	943.0	3.413	336	2.06	1.19	0.52	28.972	263	1.95	1.03	0.63	2.999
potholes$_1$	282	277.3	0.498	564	6.07	4.75	0.65	28.475	814	5.25	3.82	0.73	2.563
jh$_1$	356	363.7	0.644	712	3.87	2.84	0.36	49.823	1 066	3.90	2.76	0.69	3.410
pb$_{1.5}$	415	839.6	0.446	830	2.21	1.21	1.50	50.553	1 436	1.43	0.91	0.24	3.360
h2$_2$	568	1 316.2	6.107	1 136	2.69	2.00	0.43	337.014	1 352	2.26	1.61	0.43	9.096
ta$_1$	574	541.1	1.065	1 148	5.59	4.55	0.56	100.442	1 367	5.08	4.35	0.54	4.195

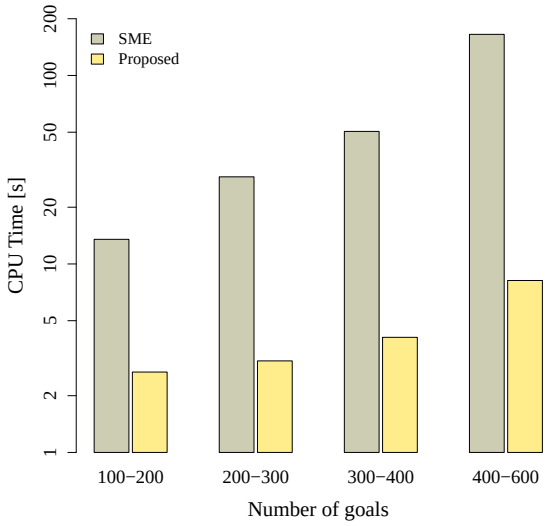
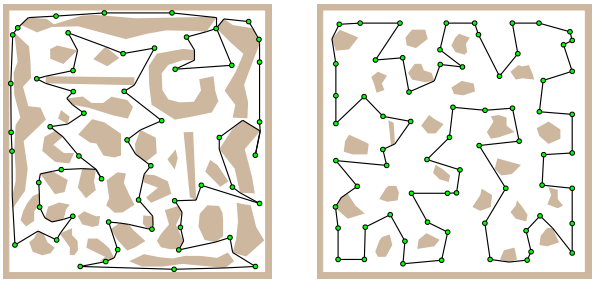


Figure 9: Average values of the required computational time for the multi-goal path planning problems.



(a) dense$_4$, L=187.8 m (b) potholes$_2$, L=158.2 m

Figure 10: Examples of the best solutions of the multi-goal path planning problem found by the proposed adaptation schema.

6.2. Inspection Planning with Restricted Visibility Range

Three maps of real environments[1] denoted as *jh*, *ta* and *pb* are used for discrete and continuous sensing evaluation. The examined visibility ranges are from the set $\{inf, 10.0, 5.0, 4.0, 3.0, 2.0, 1.5, 1.0\}$ meters, where inf denotes the unrestricted visibility range. The experimental results of the proposed algorithms are presented in the following subsections.

[1]The maps represent real testing environments of the search and rescue mission experiments of the IST-2001-FET project number 38873 - PeLoTe - Building Presence through Localization for Hybrid Telematic Systems [34].

Table 3: Experimental results for the inspection planning with discrete sensing

Map	d [m]	n	L_{ref} [m]	T_{init} [s]	**SME** m	**PDM**	**PDB**	s_L%	T_a [s]	**Proposed** m	**PDM**	**PDB**	s_L%	T_a [s]
jh	*inf*	77	193.3	0.12	154	1.55	0.51	0.57	2.0	213	1.04	0.59	0.47	0.6
jh	10.0	78	194.6	0.12	156	2.07	0.94	0.91	2.1	216	1.67	0.51	0.80	0.6
jh	5.0	85	204.3	0.12	170	1.98	0.81	0.67	2.6	254	1.64	0.84	0.61	0.7
jh	4.0	89	207.9	0.14	178	2.32	1.02	0.84	2.8	248	1.55	0.59	0.60	0.7
jh	3.0	100	215.5	0.14	200	2.09	0.96	0.75	3.6	297	2.04	0.42	1.40	0.8
jh	2.0	180	295.5	0.23	360	3.12	2.12	0.48	11.7	445	2.84	1.62	0.70	1.8
jh	1.5	282	359.0	0.44	564	3.47	2.63	0.48	29.1	673	2.78	1.88	0.55	2.7
jh	1.0	563	485.0	1.48	1 126	4.10	3.54	0.34	125.5	1 678	4.19	3.54	0.46	5.7
ta	*inf*	46	215.6	0.03	92	1.74	0.34	1.61	0.6	137	1.17	0.00	1.37	0.2
ta	10.0	47	216.9	0.03	94	2.74	0.66	2.20	0.6	131	1.05	0.11	0.78	0.2
ta	5.0	70	256.8	0.04	140	1.52	0.44	0.83	1.5	210	0.57	0.19	0.34	0.4
ta	4.0	93	291.3	0.06	186	2.19	1.56	0.53	2.6	238	2.79	1.46	1.06	0.6
ta	3.0	138	335.6	0.08	276	1.42	1.00	0.32	6.1	397	1.86	0.98	0.73	1.2
ta	2.0	255	427.0	0.21	510	3.95	3.36	0.37	22.0	707	4.20	2.67	0.52	2.0
ta	1.5	432	538.5	0.64	864	5.56	4.65	0.54	62.0	1 068	5.40	4.65	0.45	3.2
ta	1.0	934	774.3	3.48	1 868	6.24	5.48	0.36	327.1	2 207	5.77	4.61	0.45	8.9
pb	*inf*	50	554.9	0.05	100	3.81	0.10	4.19	0.7	150	2.34	0.00	3.48	0.2
pb	10.0	76	615.9	0.07	152	1.06	0.29	2.11	1.6	235	0.37	0.18	0.09	0.4
pb	5.0	134	687.2	0.11	268	0.68	0.37	0.19	5.0	650	0.15	0.00	0.10	0.9
pb	4.0	165	721.9	0.13	330	2.15	0.78	2.29	7.8	526	0.73	0.42	0.31	1.2
pb	3.0	244	781.6	0.21	488	1.80	0.52	2.12	17.6	732	0.76	0.53	0.15	1.7
pb	2.0	473	919.0	0.68	946	2.54	1.42	2.09	68.3	1 418	1.62	1.33	0.17	3.7
pb	1.5	870	1 158.2	2.30	1 740	2.79	2.41	0.63	241.5	2 581	2.45	1.97	0.26	8.2
pb	1.0	1 845	1 606.6	13.33	3 690	4.15	3.71	0.22	1186.4	4 368	3.78	3.17	0.33	41.4

6.2.1. Inspection Planning with Discrete Sensing

A set of sensing locations is found by the sensor placement algorithm [17] for each polygonal map and the visibility range d. The algorithm has been selected mainly due to its similarity to the used supporting triangular mesh in the d-WRP algorithm. The sensing locations are goals in the multi-goal path planning problem that is solved as the TSP like in the previous experiments.

Detail experimental results are presented in Table 3. Notice the higher number of nodes m for the proposed schema. However, the required computational time is approximately thirty times lower for the largest problem. Moreover, the solution quality is better or competitive to the SME schema.

The results show that the proposed adaptation schema provides better results in less computational time. Even though the solution quality improvements are only in units of percents, the speedup improvements are in tens. The proposed schema typically finished the adaptation with a higher number of nodes (about more than 20 %) than the SME algorithm. The higher number of nodes together with lower computational requirements indicate that a lower number of nodes is actually adapted. The proposed adaptation rules with decreasing σ and δ based on the number of performed adaptation steps decrease the computational burden. However, the rules also decrease the solution quality that is "compensated" by the proposed winner selection method.

The utilized approximate shortest path uses precomputed paths from map vertices to the all goals. The required memory footprint of the algorithm is about 330 MB for the largest problem *pb* with d=1 m. A typical value of the required memory by the program without the precomputed paths is about 20 MB, which provides an estimation of the real space requirements of the supporting structures.

6.2.2. d-WRP - Inspection Planning with Continuous Sensing

The proposed d-WRP algorithm utilizes a triangular mesh and a convex cover set build on top of the mesh triangles. The number of triangles and convex polygons of the cover set has influence to the algorithm performance. For each map and a particular visibility range a triangular mesh has been created individually by the quality mesh generator `triangle` [35] for the required minimal angle

Table 4: Experimental results for the inspection planning with continuous sensing - d-WRP

Map	d	L_{ref}	SME					Proposed				
	[m]	[m]	m	**PDM**	**PDB**	s_L%	T [s]	m	**PDM**	**PDB**	s_L%	T [s]
jh	*inf*	193.3	87	-48.99	-49.79	1.88	2.33	129	-45.29	-49.58	9.05	0.31
jh	10.0	194.6	87	-48.96	-50.03	3.04	2.36	128	-46.97	-49.65	4.40	0.31
jh	5.0	204.3	87	-46.95	-49.32	3.22	2.59	154	-43.23	-48.33	5.45	0.34
jh	4.0	207.9	174	-40.62	-44.40	2.83	6.25	206	-33.73	-38.65	5.57	0.40
jh	3.0	215.5	186	-24.43	-25.94	1.21	11.02	321	-17.75	-21.95	4.19	1.29
jh	2.0	295.5	381	-13.30	-14.91	1.08	45.27	831	-7.54	-12.20	3.64	6.44
jh	1.5	359.0	682	-6.20	-7.57	0.85	146.43	1 585	-1.75	-5.40	2.90	17.95
jh	1.0	485.0	1 701	0.96	-0.29	0.79	948.87	3 684	2.33	-0.74	2.80	114.40
ta	*inf*	215.6	101	-34.77	-35.23	0.45	0.53	112	-33.27	-34.59	3.45	0.08
ta	10.0	216.9	101	-32.84	-33.11	0.20	0.79	103	-32.03	-33.01	0.66	0.08
ta	5.0	256.8	101	-16.65	-18.64	1.84	2.40	120	-14.86	-17.96	2.73	0.24
ta	4.0	291.3	203	-13.23	-16.96	1.55	7.70	227	-9.81	-13.82	2.61	0.40
ta	3.0	335.6	376	-11.56	-14.14	1.65	26.21	427	-4.78	-9.08	3.05	1.38
ta	2.0	427.0	778	-3.25	-4.63	0.96	132.46	1 263	0.29	-3.31	1.78	9.90
ta	1.5	538.5	1 247	-2.05	-3.98	1.06	408.80	2 419	-1.64	-4.40	1.09	48.29
ta	1.0	774.3	3 522	1.07	0.20	0.74	2 993.48	5 570	-0.13	-1.47	0.68	347.47
pb	*inf*	554.9	240	-22.25	-23.73	3.82	2.87	254	-20.51	-22.49	4.66	0.59
pb	10.0	615.9	240	-13.08	-15.11	2.75	5.26	244	-14.63	-16.69	1.17	0.48
pb	5.0	687.2	240	-7.92	-9.33	1.63	11.55	267	-7.66	-9.31	1.46	1.23
pb	4.0	721.9	481	-5.92	-8.31	3.19	35.91	318	-6.82	-7.87	0.72	2.51
pb	3.0	781.6	616	-6.38	-7.34	0.61	86.75	811	-6.27	-7.02	0.56	10.22
pb	2.0	919.0	1 408	-4.73	-5.37	0.43	503.42	2 145	-5.40	-6.47	0.56	56.10
pb	1.5	1 158.2	2 858	-2.86	-4.24	0.61	2 092.47	4 614	-3.25	-4.06	0.51	282.68
pb	1.0	1 606.6	5 785	-0.08	-0.75	0.40	11 357.65	11 316	-1.40	-2.05	0.44	2 239.71

32.5°, and 25.0° for the map *jh*, and a selected maximum triangle area. The area is experimentally set according to the circumscribed circle of the triangle, which radius is derived from the restricted visibility range d. Particular properties of the used meshes are depicted in Table A.5.

In the d-WRP algorithm, path queries are resolved by the approximation of the shortest path between two points in $\mathcal{W}$ using the convex partition of $\mathcal{W}$. Because of the relatively small number of vertices in the examined maps, the initialization of the shortest path between map vertices is not computationally demanding like in the TSP algorithm. Nevertheless, the time is included in the presented results like in the discrete sensing. Construction of the triangular mesh, the convex polygon partition, and the visibility graphs is done in a fraction of second. Also a convex cover set for the largest problem, regarding the number of triangles, is found in hundreds of milliseconds. In comparison to the required computational time of the adaptation procedure the required times to create the supporting structures are negligible. Moreover, in comparison to the decoupled approach of the inspection planning the problem of determining a set of sensing locations can be more computationally demanding [10].

In the algorithm based on the SME schema, the number of nodes is set individually according to the number of triangles of used triangular mesh, see Table A.5 and m in Table 4. For the proposed adaptation schema, the initial number of nodes is set to the tenth of the number of triangles, $m = 0.1N_T$.

Detail experimental results are presented in Table 4. The proposed adaptation schema provides solution in a less computational time. However, in several cases, the found solutions have worse quality than solutions provided by the SME schema. Regarding the PDB the proposed algorithm provides shorter inspection paths than the reference solutions in all cases. Also for small visibility ranges the proposed adaptation schema provides better results than the SME schema. In all cases, the found solutions provide full coverage of $\mathcal{W}$, and the convergence issue has not been observed.

Examples of the best found solutions for the selected visibility ranges d are presented in Fig. 11. Notice the self-crossing route in Fig. 11h that is caused due to avoidance of the winner node selection to the already covered area. Once the corner is covered, the network does not adapt to that part. Also, such a crossing can be caused by the

deletion of inactive nodes, because the shape of the ring can be significantly changed after removing the nodes, and self-crossing can suddenly occur.

The memory footprint of the d-WRP algorithm is smaller than for the multi-goal path planning, because only the vertex–vertex paths are pre-computed. The required memory is about 24 MB for the problem *pb* with d=1 m.

During the experimental verification of the algorithm, a sensitivity to the initialization of the nodes has been observed for the proposed d-WRP algorithm. An initialization as a small ring around a point gives similar results, however in several cases the found solutions were worse than the reference solutions in units of percents. Although the proposed hull initialization provides overall best results, it can also stick the found route in a local solution. The reason for that is similar to self-crossings. Once triangles are covered, the restricted set of the neighbouring nodes does not spread nodes to other parts; thus, the nodes remain close to their previous positions. In the presented results, this can be observed for small visibility ranges in the maps *jh* and *ta*, which contain several rooms, and does not occur in the map *pb*. Despite this issue, the found solutions are competitive with the SME schema.

6.3. Comparison of Discrete and Continuous Sensing

An overall comparison of the solution quality for the discrete and continuous sensing approaches is presented in Fig. 12 as a histogram of average values for the visibility distances. Due to shorter paths of the d-WRP solutions than the reference solutions, the PDM is increased about 100%, i.e., 100% is the length of the reference path. The discrete sensing inspection solved as the multi-goal path planning is denoted to the TSP in the figure. Even though the histogram bins represent average values over all maps and particular selected visibility ranges, the histogram shows increasing quality of the d-WRP solution over the TSP for higher visibility ranges. With regard to the required computational time, see Fig. 13, the proposed d-WRP algorithm provides the best results (according to the PDM) in hundreds of milliseconds for high values of d. For small visibility ranges, a density of the sensing locations in the map is high, and d-WRP solutions are only about units of percents shorter.

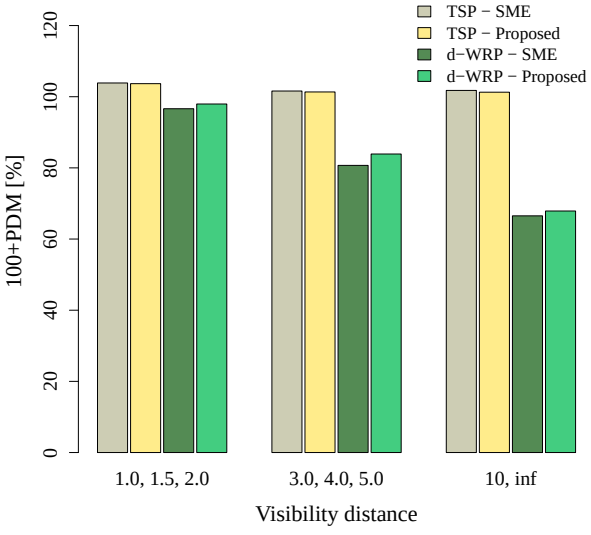


Figure 12: Average values of the solution quality for the multi-goal path planning problems.

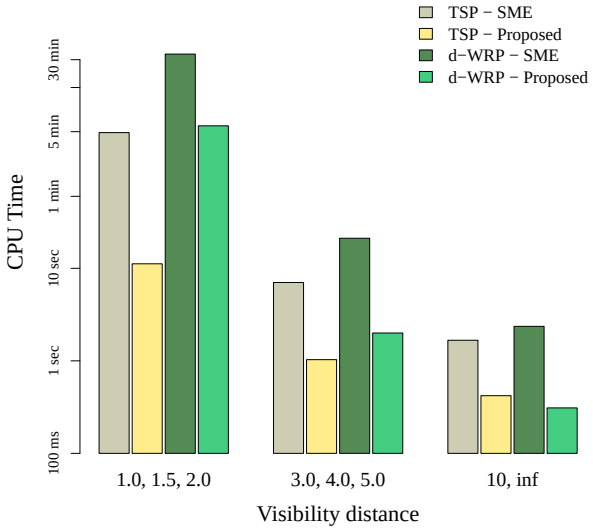
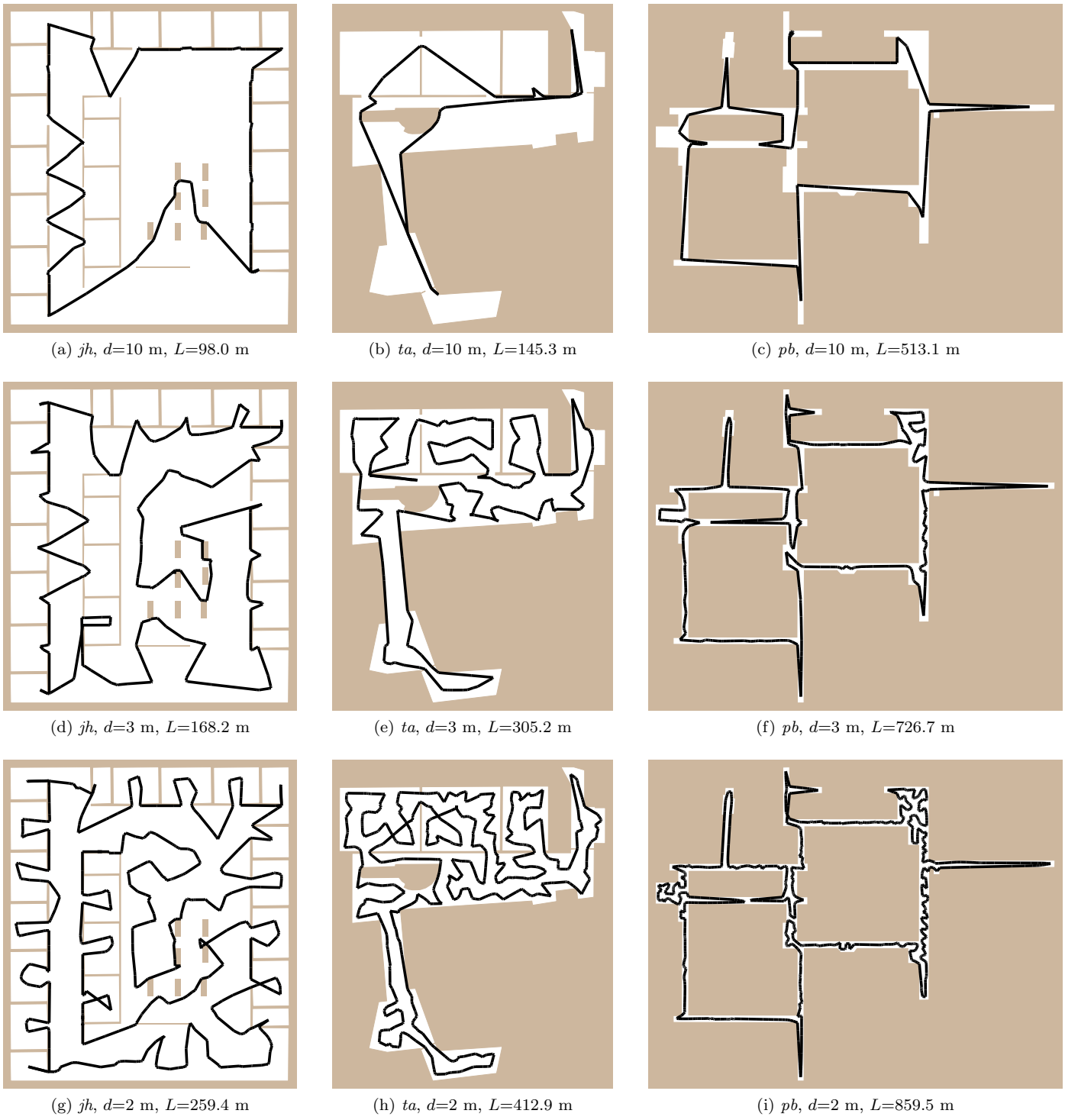


Figure 13: Average values of the required computational time for the multi-goal path planning problems.

6.4. Discussion

Regarding the experimental results the found inspection paths for the d-WRP are shorter than for the decoupled approach with the same visibility range d. However, it cannot be clearly stated that the continuous sensing is better than the discrete approach, because the sensing and motion costs have to be taken into account. Considering the available d-WRP algorithms and the presented experimental results, it seems that for small visibility ranges the decoupled approach is appropriate. Moreover, a more sophisticated sensor placement algorithm can provide lower number of sensing locations leading to shorter inspection paths [10]. For higher visibility ranges, the proposed d-WRP algorithm may be used for finding a solution of the sensor placement. Because the found path of the d-WRP is shorter than for the TSP, it is expected that such a solution provides overall better solution for both costs (sensing and motion). The final position of the winner nodes can be used as primal sensing locations, and to achieve complete coverage points at the final ring can be selected.

The problem of selection of the smallest set of points at the watchman route is called Vision Points problem in computational geometry. A combination of the proposed d-WRP algorithm and selection of the sensing locations can provide a suitable mechanism to combine the cost of



(a) *jh*, d=10 m, L=98.0 m (b) *ta*, d=10 m, L=145.3 m (c) *pb*, d=10 m, L=513.1 m

(d) *jh*, d=3 m, L=168.2 m (e) *ta*, d=3 m, L=305.2 m (f) *pb*, d=3 m, L=726.7 m

(g) *jh*, d=2 m, L=259.4 m (h) *ta*, d=2 m, L=412.9 m (i) *pb*, d=2 m, L=859.5 m

Figure 11: Best found d-WRP solutions by the proposed adaptation schema.

motion with the cost of sensing based on SOM. This problem is tightly related with the problem of nodes deletion in the proposed adaptation schema, because nodes that may not be deleted are eventual candidates to be sensing locations. Even though the proposed three phases adaptation provides "good" solutions in less computational time, this problem needs future investigations that can open new application areas of SOM in the field of visibility problems studied in computational geometry.

7. Conclusion

New adaptation schema for the inspection planning has been presented in this paper. The schema uses new winner selection rule that considers a path between two nodes utilizing a node creation/deletion mechanism. Besides, the schema comprises particular SOM improvements proposed by several authors. The schema has been applied to the discrete and continuous sensing variants of the inspection planning. Both sensing variants are considered with the restricted visibility range. The discrete variant is the multi-goal path planning formulated as the non-Euclidean TSP, and the continuous sensing variant is formulated as the d-WRP.

The schema has been experimentally verified in a set of problems representing the non-Euclidean TSP and the d-WRP. The presented experimental results show that the proposed adaptation schema is faster than the SME schema, it provides better solutions for discrete sensing, and competitive solutions for the d-WRP.

For high visibility ranges, the proposed d-WRP algorithm provides significantly shorter inspection paths in comparison with the solutions for the discrete sensing. During the solution of the d-WRP the winner nodes can be considered as the sensing locations that makes the SOM algorithm applicable to the inspection planning with combination of the sensing and motion costs. Such a combination opens future applications of SOM principles to the similar visibility based routing problems.

8. Nomenclature

$\mathcal{W}$	the polygonal domain representing the robot workspace, $\mathcal{W} \subset \mathbb{R}^2$
v	the number of vertices of $\mathcal{W}$
h	the number of holes of $\mathcal{W}$
p	the number of convex polygons of the convex partition of $\mathcal{W}$
n	the number of goals
d	the visibility range
N_V	the number of trianguar mesh vertices
N_T	the number of triangles
N_C	the number of convex polygons of the cover set of $\mathcal{W}$
a_i	the visible area from a triangle
g	a goal, $g \in \mathcal{W}$
ν	a node (neuron weights), $\nu \in \mathcal{W}$
$S(\nu, g)$	an approximate path from ν to g
m	the number of nodes (neurons)
f	the neighbouring factor
δ	the size of the winner node neighbourhood
σ	the learning gain (neighbouring function variance)
$f(\sigma, l)$	the neighbouring function
α	the gain-decreasing rate
μ	learning rate
s	the node moving activity threshold
ϵ	the minimal allowable error

Acknowledgement

The work has been supported by the Ministry of Education of the Czech Republic under the program "National research program II" by Project No. 2C06005 and partially by Project No. 7E08006 and EU project No. 216342.

Appendix A. Parameters of the support triangular meshs for the WRP

References

[1] H.H. González-Baños, D. Hsu, and J.-C. Latombe. Motion planning: Recent developments. In S.S. Ge and F.L. Lewis, editors, *Autonomous Mobile Robots: Sensing, Control, Decision-Making and Applications*, chapter 10. CRC Press, 2006.

[2] Wei-Pang Chin and Simeon Ntafos. Optimum watchman routes. In *SCG '86: Proceedings of the second annual symposium on Computational geometry*, pages 24–33, Yorktown Heights, New York, United States, 1986. ACM Press.

[3] Eli Packer. *Robust Geometric Computing and Optimal Visibility Coverage*. PhD thesis, Stony Brook University, New York, 2008.

[4] Joseph C. Culberson and Robert A. Reckhow. Covering polygons is hard. *Journal of Algorithms*, 17(1):2–44, 1994.

[5] Steven N. Spitz and Aristides A. G. Requicha. Multiple-Goals Path Planning for Coordinate Measuring Machines. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2322–2327, 2000.

Table A.5: Parameters of Supporting Structures for the WRP

d	**Map *jh***			**Map *ta***			**Map *pb***		
[m]	N_N	N_T	N_C	N_N	N_T	N_C	N_N	N_T	N_C
inf	576	872	100	638	1 014	46	1 630	2 403	52
10.0	576	872	108	638	1 014	70	1 630	2 403	111
5.0	576	872	130	638	1 014	152	1 630	2 403	262
4.0	576	872	169	638	1 014	209	1 630	2 403	373
3.0	608	931	258	776	1 252	357	2 018	3 078	714
2.0	1 183	1 904	480	1 151	1 944	757	2 955	4 692	1 564
1.5	1 392	2 272	852	1 788	3 117	1 320	4 319	7 144	2 787
1.0	1 988	3 401	1 800	3 849	7 044	2 955	8 250	14 462	6 188

[6] Mitul Saha, Gildardo Sánchez-Ante, and Jean-Claude Latombe. Planning multigoal tours for robot arms. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 3797–3803, 2003.

[7] Xuehou Tan and Tomio Hirata. Finding shortest safari routes in simple polygons. *Information Processing Letters*, 87(4):179–186, 2003.

[8] Tim Danner and Lydia E. Kavraki. Randomized Planning for Short Inspection Paths. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 971–976, San Fransisco, CA, April 2000. IEEE Press.

[9] Jan Faigl and Miroslav Kulich. Sensing locations positioning for multi-robot inspection planning. In *DIS '06: Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications*, pages 79–84, Washington, DC, USA, June 2006. IEEE Computer Society.

[10] Jan Faigl, Miroslav Kulich, and Libor Přeučil. A sensor placement algorithm for a mobile robot inspection planning. *Journal of Intelligent & Robotic Systems*, 2010. doi: 10.1007/s10846-010-9449-0.

[11] Vašek Chvátal, William Cook, George B. Dantzig, Delbert R. Fulkerson, and Selmer M. Johnson. *Solution of a Large-Scale Traveling-Salesman Problem*, chapter 1, pages 7–28. Springer Berlin Heidelberg, 2010.

[12] Soheil Ghafurian and Nikbakhsh Javadian. An ant colony algorithm for solving fixed destination multi-depot multiple traveling salesmen problems. *Applied Soft Computing*, 11(1):1256–1262, January 2011.

[13] E. M. Cochrane and J. E. Beasley. The co-adaptive neural network approach to the Euclidean travelling salesman problem. *Neural Networks*, 16(10):1499–1525, 2003.

[14] Thiago A. S. Masutti and Leandro N. de Castro. A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem. *Information Sciences*, 179(10):1454–1468, 2009.

[15] Jan Faigl. Approximate Solution of the Multiple Watchman Routes Problem with Restricted Visibility Range. *IEEE Transactions on Neural Networks*, 21(10):1668–1679, 2010.

[16] Samerkae Somhom, Abdolhamid Modares, and Takao Enkawa. A self-organising model for the travelling salesman problem. *Journal of the Operational Research Society*, pages 919–928, 1997.

[17] Giorgos D. Kazazakis and Antonis A. Argyros. Fast positioning of limited visibility guards for the inspection of 2d workspaces. In *Proceedings of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland, September 2002.

[18] Raimund Seidel. A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons. *Computational Geometry*, 1(1):51–64, 1991.

[19] M. H. Overmars and E. Welzl. New methods for computing visibility graphs. In *SCG '88: Proceedings of the fourth annual symposium on Computational geometry*, pages 164–171, New York, NY, USA, 1988. ACM.

[20] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. CONCORDE TSP Solver. `http://www.tsp.gatech.edu/concorde.html`, 2003. [cited 8 Jul 2010].

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

[22] J. C. Fort. Solving a combinatorial problem via self-organizing process: An application of the Kohonen algorithm to the traveling salesman problem. *Biological Cybernetics*, 59(1):33–40, 1988.

[23] Alessio Plebe and Angelo Marcello Anile. A neural-network-based approach to the double traveling salesman problem. *Neural Comput.*, 14(2):437–471, 2002.

[24] Frederico Carvalho Vieira, Adri ao Duarte Dória Neto, and José Alfredo Ferreira Costa. An Efficient Approach to the Travelling Salesman Problem Using Self-Organizing Maps. *International Journal of Neural Systems*, 13(2):59–66, 2003.

[25] Yanping Bai, Wendong Zhang, and Zhen Jin. An new self-organizing maps strategy for solving the traveling salesman problem. *Chaos, Solitons & Fractals*, 28(4):1082 – 1089, 2006.

[26] Gerhard Reinelt. TSPLIB– A Traveling Salesman Problem Library. *Journal on Computing*, 3(4):376–384, 1991.

[27] Samerkae Somhom, Abdolhamid Modares, and Takao Enkawa. Competition-based neural network for the multiple travelling salesmen problem with minmax objective. *Computers and Operations Research*, 26(4):395–407, 1999.

[28] Wendong Zhang, Yanping Bai, and Hong Ping Hu. The incorporation of an efficient initialization method and parameter adaptation using self-organizing maps to solve the TSP. *Applied Mathematics and Computation*, 172(1):603–623, 2006.

[29] Jan Faigl. *Multi-Goal Path Planning for Cooperative Sensing*. PhD thesis, Czech Technical University in Prague, 2010.

[30] Marcelo Kallmann. Path Planning in Triangulations. In *Proceedings of the IJCAI Workshop on Reasoning, Representation, and Learning in Computer Games*, Edinburgh, Scotland, July 31 2005.

[31] Olivier Devillers, Sylvain Pion, Monique Teillaud, and Projets Prisme. Walking in a Triangulation. *International Journal of Foundations of Computer Science*, 13:106–114, 2001.

[32] Jan Faigl, Miroslav Kulich, Vojtěch Vonásek, and Libor Přeučil. An Application of Self-Organizing Map in the non-Euclidean Traveling Salesman Problem. *Neurocomputing*, 74(5):671–679, 2011.

[33] Laura Burke. "Conscientious" neural nets for tour construction in the traveling salesman problem: the vigilant net. *Computers and Operations Research*, 23(2):121–129, 1996.

[34] L. Přeučil, J. Pavlíček, R. Mázl, F. Driewer, and K. Schilling.

Next Generation Human-Robot Telematic Teams. In *Multidisciplinary Collaboration for Socially Assistive Robotics*, pages 65–69, Menlo Park, California, 2007. AAAI Press.

[35] Jonathan Richard Shewchuk. Delaunay Refinement Algorithms for Triangular Mesh Generation. *Computational Geometry: Theory and Applications*, 22:1–3, 2001.