Visiting Convex Regions in a Polygonal Map

Jan Faigl, Vojtěch Vonásek, Libor Přeučil

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

Abstract

This paper is concerned with a variant of the multi-goal path planning in which goals are represented as convex polygons. The problem is to find a closed shortest path in a polygonal map such that all goals are visited. The proposed solution is based on a self-organizing map algorithm for the traveling salesman problem. Neurons' weights are considered as nodes inside the polygonal domain and connected nodes represent a path that evolves according to the proposed adaptation rules. In addition, a reference algorithm based on the solution of the traveling salesman problem and the consecutive touring polygons problem is provided to find high quality solutions of the created set of problems. The problems are designed to represent various inspection and patrolling tasks and can form a kind of benchmark set for multi-goal path planning algorithms. The performance of the algorithms is examined in this problem set, which includes an instance of the watchman route problem with restricted visibility range. The proposed algorithms provide a unified approach to solve various visibility based routing problems in polygonal maps while they provide competitive quality of solutions to the reference algorithm with significantly lower computational requirements.

Keywords: Inspection Planning, Multi-Goal Path Planning, Self-Organizing Map, Safari Route Problem, Traveling Salesman Problem, Watchman Route Problem, Touring Polygons Problem, Polygonal Domain

1. Introduction

A problem to find a path visiting a set of given goals by a robot is called the *multi-goal path planning problem* (MTP). In particular, the MTP stands to find a shortest path connecting a given set of goals located in a robot working environment. Approaches for the MTP are motivated by practical problems that include planning for a robotic arm [1, 2], where the found path leads to minimization of the execution time providing a better utilization of the tools, or, in a case of a mobile robot, inspection planning [3], e.g., motivated by a search and rescue missions [4], where the time to find possible victims is critical.

A robot working environment can be represented by the polygonal domain $\mathcal{W}$ and goals may be represented by points. In such a case, the MTP can be formulated as the *traveling salesman problem* (TSP) [5]. Thus, the MTP becomes a combinatorial optimization problem to find a sequence of goals' visits, e.g., using all shortest paths between goals found in a visibility graph by Dijkstra's algorithm.

A more general variant of the MTP can be more appropriate if objects of interest may be located in certain regions of $\mathcal{W}$, e.g., when it is sufficient to reach a particular part of the environment to "see" or measure the requested object. A practical example of such a problem is collecting samples from particular areas, e.g., taking snapshots of objects or measuring concentration levels of substances' in regions or ponds, which are accessible from various directions.

In such a problem formulation, a goal is a polygonal region rather than a single point. Several algorithms addressing this problem can be found in literature; however, only for its particular restricted variant. For example goals form a disjoint set of convex polygons attached to a simple polygon in the *safari route problem* [6], which can be solved in $O(n^3)$ [7]. If the route enter to the convex goal is not allowed, the problem is called the *zoo-keeper problem*, which can be solved in $O(n \log n)$ for a given starting point and the full shortest path map [8]. However, both problems are NP-hard in general.

A combinatorial approach [2] can be used for the MTP with partitioned goals, where each goal is represented by a finite (small) set of point goals. However, combinatorial approaches are unsuitable for continuous sets because of too many possibilities how to connect the goals.

In this paper, we present a self-organizing map (SOM) based algorithm for the general variant of the MTP with polygonal goals. The algorithm is based on SOM for the TSP in $\mathcal{W}$ [9]. Contrary to combinatorial approaches or other soft-computing techniques [10], a geometrical interpretation of SOM evolution in $\mathcal{W}$ allows easy and straightforward extensions to deal with polygonal goals. To show flexibility of the SOM approach, several modifications of the adaptation rules are proposed and evaluated in a set of problems, which also demonstrates a geometric relation between the learning network and polygonal goals.

The main advantage of the proposed approach is ability to address general multi-goal path planning problems in $\mathcal{W}$ (not only in a simple polygon) and with goals not necessarily attached to $\mathcal{W}$; thus, the approach provides a unifying framework to solve various MTP variants.

Beside the SOM based algorithms, we present an alternative approach to address the MTP with polygonal goals that provides a reference solution of the problems solved. It is based on the solution of the TSP with point goals and a consecutive solution of the *touring polygons problem* (TPP). Although, a polynomial algorithm for the TPP in a simple polygon has been proposed in [11], our reference algorithm is able to solve problems in $\mathcal{W}$ (not only in a simple polygon) and it also does not require disjoint convex goals. In addition, it is also probably easier to implement; thus, it represents a suitable reference algorithm for a comparison.

The rest of this paper is organized as follows. The next section provides an overview of the related work and similar problem formulations. Besides, it also contains a brief description of the SOM adaptation schema for the TSP in $\mathcal{W}$, because the proposed algorithms for the MTP with polygonal goals are its extensions. The addressed problem formulation and evaluation methodology of the solution quality and algorithms' comparison is presented in Section 3. A reference algorithm based on the solution of the related TSP and the consecutive TPP is presented in Section 4. In Section 5, the proposed modifications of the SOM's adaptation rules to deal with the polygonal goals are presented. The results and comparisons of the proposed algorithms and discussion of the results achieved are presented in Section 6. Concluding remarks are presented in Section 7.

2. Related Work

In this section, we present an overview of approaches to address the visibility based routing problems, i.e., various formulations of the MTP. Mainly because once a point robot is assumed, $\mathcal{W}$ directly represents the robot configuration space and many visibility based approaches can be applied to solve such a variant of the MTP. As the proposed approach is based on the SOM algorithm for the TSP, its brief description is presented in Section 2.1.

The multi-goal path planning can be considered as a type of path planning under visibility constraints [12]. The goals in the MTP can represent sensing locations, where a robot takes measurements. Such locations can be found by a sensor placement algorithm that aims to find a minimal set of locations from which the whole $\mathcal{W}$ is covered ("*seen*") by a robot sensing device.

The sensor placement problem is related to the *art gallery problem* (AGP), which is a classical problem studied in the computational geometry. The AGP was posed by Klee in 1973 and its most basic form is [13]: "*What is the smallest number of guards needed to guard an art gallery?*". The guards are static in the AGP and several problem variants for segment guards (representing patrolling guards) have been proposed [14, 15]. Besides, a patrolling route for a single robot can be found as a solution of the *traveling salesman problem* (TSP) where guards locations become cities that have to be visited [5]; hence, the problem becomes the MTP. Even though optimal algorithms for the AGP have been proposed for a restricted class of polygons [16, 17], the AGP is known to be NP-hard for a polygon with holes [18], and therefore, approximate algorithms are preferred to find guards [19, 20]. Moreover, additional visibility constraints can be considered, which is the reason why authors of [12] call the problem the *sensor placement problem* rather than the AGP. The constraints can restrict the visibility to a distance d, or an incident angle that regards a situation where a guard prefers to watch a scene directly rather than under unsuitable angle [21]. Several approximate algorithms have been proposed to address the sensor placement problem, e.g., based on deterministic convex partitioning [22] or randomized approaches [23, 3]. Once a set of guards covering the environment is determined, the problem to be solved is (again) the TSP.

The aforementioned approaches (consisting of finding the guards and the consecutive solution of the TSP) represent the so-called decoupled approach of the inspection planning to cover the whole $\mathcal{W}$. In the decoupled approach, the sensing of the environment is performed at discrete places, i.e., at the guards positions, and therefore, the problem of guarding/searching $\mathcal{W}$ by one robot leads to minimize visiting period of the sensing places. Regarding the cost of sensing and the cost of motion the number of places is minimized in the AGP part while the length of the path is minimized in the TSP. Due to independent solutions of the AGP and TSP, the decoupled approach is suitable for cases where the sensing cost is dominant over the motion cost [12].

A continuous sensing can be assumed if the motion cost is dominant and the sensing cost is relatively cheap. For such a case the problem can be formulated as the *watchman route problem* (WRP) that is a problem to find a closed shortest path such that all points of $\mathcal{W}$ are visible from at least one point of the path [24]. The WRP is NP-hard for the polygonal domain and similarly to the AGP, polynomial algorithms have been proposed for restricted class of polygons [25]. The main difficulty of the problem is that the sensing locations are not explicitly prescribed, therefore approaches based on the TSP cannot be directly used as they will lead to the decoupled approach. Here, it is worth to mention that a problem of finding a minimal set of guards lying on the shortest watchman route is called the *vision points problem* and is NP-hard [26].

A multi-robot variant of the WRP is the *m-watchman routes problem* (MWRP) that aims to find a route for each of m watchmen such that each point of the polygon $\mathcal{W}$ is visible from at least one route. For m=1 the problem is the WRP and if m is so large that the total length of the routes is zero, the problem is the stationary AGP. Nilsson proved that MWRP is NP-hard even in simple polygons [27].

Probably the first heuristic approach for the MWRP in a polygon with holes has been proposed by Packer in [28]. The approach is based on a set of static guards S found by the heuristic A_1 of [20] and constructing the minimum spanning tree of S. Distances between two guards are found as the length of the shortest path from the visibility graph. The tree is split to m sub-trees (for m watchmen) and Hamiltonian routes on each sub-tree are independently constructed. Vertices along a route are substituted by others that shorten the length of the route and maintain the full coverage. Finally, redundant vertices of the route are removed. Although this approach is based on a solution of the AGP, only unrestricted visibility range has been considered by the author.

If a visibility range is restricted to a distance d, two variants of the WRP can be found in literature [7]. The *d-watchman route problem* is a variant to see only the boundary of the polygon, while the *d-sweeper route problem* aims to sweep a polygonal floor using a circular broom of radius d, so that the total travel of the broom is minimized [6]. Approximate algorithm for the MWRP with the d-visibility has been presented in [29].

The aforementioned *safari route* [6] and *zoo-keeper route* [30] problems introduced in Section 1 are also motivated by the WRP with restricted visibility range. These problems are variants of the MTP with polygonal goals, as in both of them the problem is to find a route inside a polygon P that visits a given collection of sub-polygons of P. Also in both original problem formulations the sub-polygons are convex and are entirely inside the polygon P. Although these problems are very close to the problem addressed in this paper, the main difference is that their original formulations are only for a simple polygon, for which the polynomial algorithms have been proposed, but the problems are NP-hard for the polygonal domain.

Routing problems with polygonal goals can be considered as variants of the TSP with neighborhoods (TSPN) [31]. The TSPN is studied for graphs or as a geometric variant in a plane but typically without obstacles. Approximate algorithms for restricted variants of the TSPN have been proposed, e.g., the TSPN with arbitrary connected neighborhoods with comparable diameters and for disjoint unit disk neighborhoods [32], or disjoint convex fat neighborhoods of arbitrary size [33]. However, the TSPN is APX-hard and cannot be approximated to within a factor $2 - \epsilon$, where $\epsilon > 0$, unless P=NP [34].

Having a sequence of polygonal goals $(P_1, P_2, \ldots, P_k)$, one can ask for a shortest path visiting in order at least one point of each polygon in the sequence. This problem is called *touring polygons problem* [31], and it is a strict generalization of the safari, zoo-keeper, and watchman route problem in a simple polygon [11]. In a case of convex polygons in a plane, and given start and target points, an $O(kn \log(n/k))$ algorithm for disjoint polygons has been proposed by the authors of [11], where n is the number of vertices specifying the polygons. In addition, the authors also proposed an $O(nk^2 \log n)$ algorithm for arbitrarily intersecting polygons lying in a simple polygon. If polygons are non-convex, the TPP is NP-hard [11].

In [35], an approximate algorithm for the TPP in a plane is proposed. The algorithm is based on an iterative procedure refining the path until the selected accuracy ϵ is achieved. In each iteration, a new point at a polygon p_i is eventually computed to shorten the path connecting three consecutive polygons p_{i-1}, p_i, and p_{i+1}. Once the length of new path is shorter than the previous path's length (about less than ϵ), the refinement is terminated. A proof that the algorithm finds a global solution of the TPP is based on an approximate algorithm for solving the Euclidean shortest path problem in a three dimensional polyhedral space presented in [36].

2.1. SOM for Routing Problems in $\mathcal{W}$

A SOM algorithm for routing problems, in particular the SOM for the TSP in $\mathcal{W}$ [9], is Kohonen's type of unsupervised two-layered learning neural network. The network contains two dimensional input vector and an array of output units that are organized into a uni-dimensional structure. An input vector represents coordinates of a point goal, and connections' weights (between the input and output units) represent coordinates of the output units. Connections' weights can be considered as nodes representing a path, which provides direct geometric interpretation of the neurons' weights. So, the nodes form a ring in $\mathcal{W}$ because of the uni-dimensional structure of the output layer, see Fig. 1.

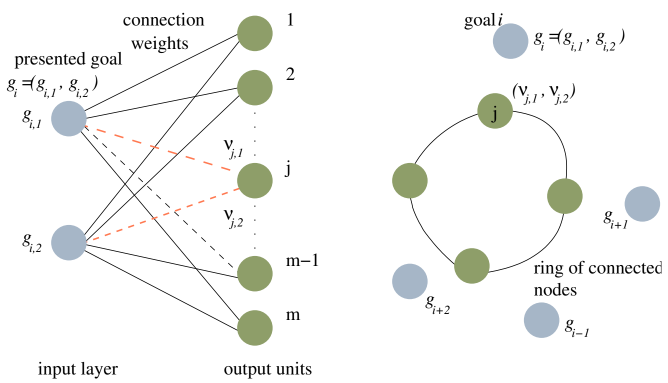
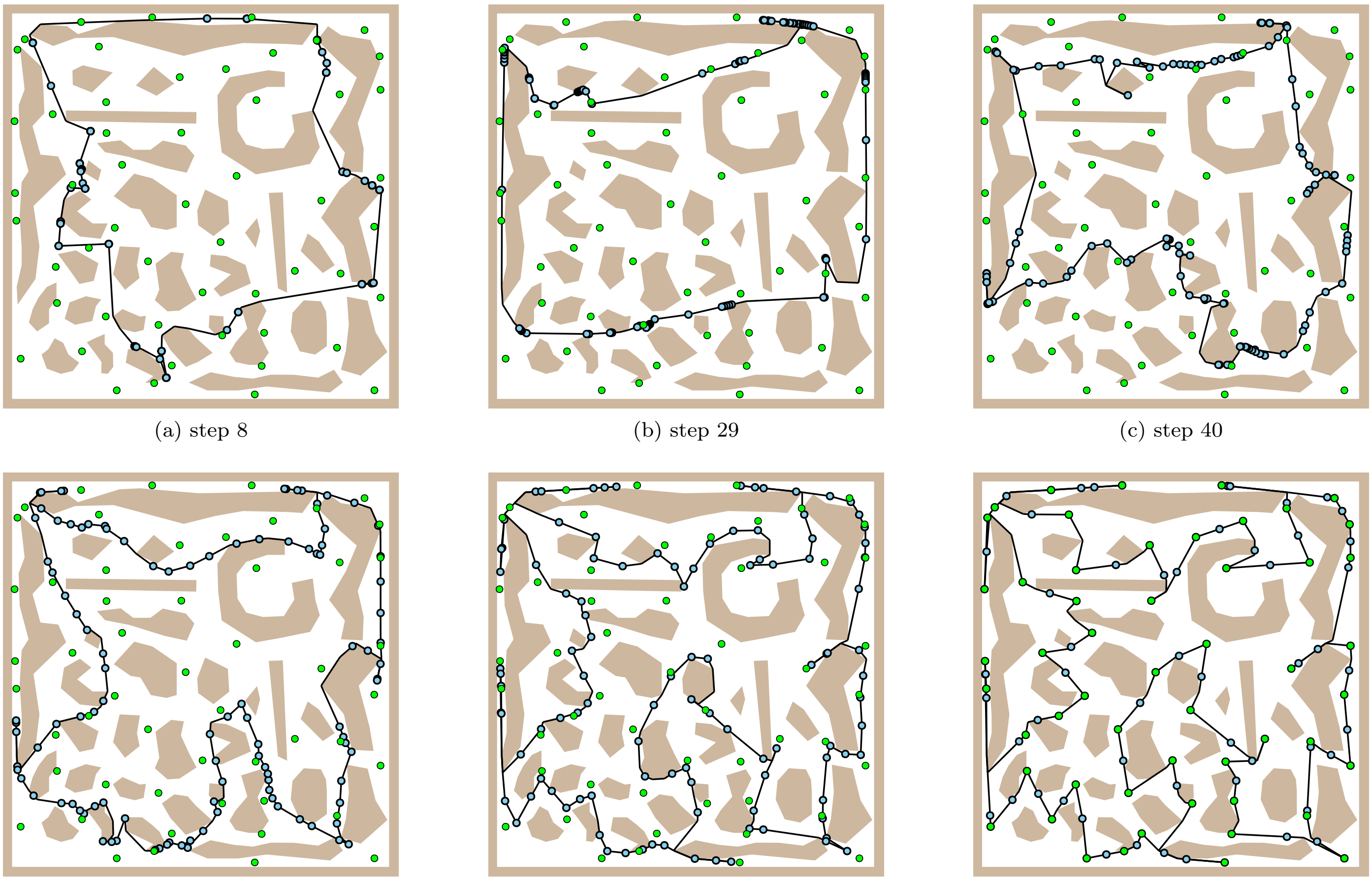


Figure 1: A schema of the two-layered neural network and the associated geometric representation.

The network learning process is an iterative stochastic procedure in which goals are presented to the network in a random order. The procedure basically consists of two phases: (1) selection of winner node to the presented goal; (2) adaptation of the winner and its neighbouring nodes toward the goal. The learning procedure works as follows.

1. *Initialization:* For a set of n goals $\boldsymbol{G}$ and a polygonal map $\mathcal{W}$, create $2n$ nodes $\mathcal{N}$ around the first goal. Let the initial value of the learning gain be σ=12.41n+0.06, and adaptation parameters be μ=0.6, α=0.1.



(a) step 8

(b) step 29

(c) step 40

(d) step 49

(e) step 58

(f) step 78

Figure 2: An example of the ring evolution in a polygonal map for the MTP with point goals, small green disks represent goals and blue disks are nodes.

2. *Randomizing:* Create a random permutation of goals $\Pi(\boldsymbol{G})$.
3. *Clear Inhibition:* $\boldsymbol{I} \leftarrow \emptyset$.
4. *Winner Selection:* Select the closest node $\nu^\star$ to the goal $g \in \Pi(\boldsymbol{G})$ according to:

$$\nu^\star \leftarrow \mathrm{argmin}_{\nu \in \mathcal{N}, \nu \notin \boldsymbol{I}} |S(\nu, g)|,$$

where $|S(\nu, g)|$ is the length of the shortest path among obstacles $S(\nu, g)$ from ν to g.
5. *Adapt:* Move $\nu^\star$ and its neighbouring nodes along a particular path toward g:
   - Let the current number of nodes be m, and N ($N \subseteq \mathcal{N}$) be a set of $\nu^\star$'s neighborhoods in the cardinal distance less than or equal to $0.2m$.
   - Move $\nu^\star$ along the shortest path $S(\nu^\star, g)$ toward g by the distance $|S(\nu^\star, g)|\mu$.
   - Move nodes $\nu \in N$ toward g along the path $S(\nu, g)$ by the distance $|S(\nu, g)|\mu f(\sigma, l)$, where f is the neighbouring function $f = \exp(-l^2/\sigma^2)$ and l is the cardinal distance of ν to $\nu^\star$.
   - Update the permutation: $\Pi(\boldsymbol{G}) \leftarrow \Pi(\boldsymbol{G}) \setminus \{g\}$.
   - Inhibit the winner: $\boldsymbol{I} \leftarrow \boldsymbol{I} \cup \{\nu^\star\}$.

   If $|\Pi(\mathbf{G})| > 0$ go to Step 4.
6. *Decrease the learning gain:* $\sigma \leftarrow (1 - \alpha)\sigma$.
7. *Termination condition:* If all goals have the winner in a sufficient distance, e.g., less than 10^{-3}, or $\sigma < 10^{-4}$ Stop the adaptation. Otherwise go to Step 2.
8. *Final path construction:* Use the last winners to determine a sequence of goals' visits.

The algorithm is terminated after a finite number of adaptation steps as σ is decreased after presentation of all goals to the network. Moreover, the inhibition of the winners guarantees that each goal has associated a distinct winner; thus, a sequence of all goals' visits can be obtained by traversing the ring at the end of each adaptation step.

The computational burden of the adaptation procedure depends on determination of the shortest path in $\mathcal{W}$, because $2n^2$ node–goal distance queries (Step 4) and $(0.8n + 1)n$ node–goal path queries (Step 5) have to be resolved in each adaptation step. Therefore, an approximate shortest path is considered using a supporting division of $\mathcal{W}$ into convex cells (convex partition of $\mathcal{W}$) and pre-computed all shortest path between map vertices to the point goals.

The approximate node–goal path is found as a path over vertices of the cells in which the points (node and goal) are located. Then, such a rough approximation is refined using a test of direct visibility from the node to the vertices of the path. Details and evaluation of refinement variants can be found in [9].

Beside the approximation, the computational burden can be decreased using the Euclidean pre-selection [37], because only the node with a shorter Euclidean distance to the goal than the distance (length of the approximate shortest path) of the current winner node candidate can become the winner.

In Fig. 2, a ring of nodes connected by an approximate shortest path between two points is shown to provide an overview of the ring evolution in $\mathcal{W}$.

3. Problem Statement

The problem addressed in this paper can be formulated as follows: *Find a closed shortest path visiting given set of goals represented as convex polygons (possibly overlapping each other) in a polygonal map $\mathcal{W}$.* The problem formulation is based on the safari route problem [6]; however, it is a more general in three aspects. First, polygons can be placed inside a polygon with holes. Also, it is not required that convex polygons are attached to the boundary of $\mathcal{W}$ like in the original safari route problem formulation. Finally, polygons can overlap each other, and therefore, such polygons can represent a polygonal goal of an arbitrary shape.

The proposed problem formulation comprises the WRP with restricted visibility range d. The set of goals can be found as a convex cover set of $\mathcal{W}$, i.e., a set of convex polygons whose union is $\mathcal{W}$. The advantage of an algorithm solving the formulated problem is that it is not required to have a minimal cover set. The restricted convex polygons to the size d can be found by a simple algorithm based on a triangular mesh of $\mathcal{W}$ [29].

3.1. The Quality of Solution

In this paper, the studied SOM based algorithms are randomized, and therefore, to examine the quality of found solutions 100 trials of each particular problem is considered. The solution quality can be then measured as the percent deviation from the reference path's length of the mean solution value (PDM)

$$PDM = \frac{L - L_{ref}}{L_{ref}} \cdot 100\%, \qquad (1)$$

and as the percent deviation from the reference of the best solution value (PDB)

$$PDB = \frac{L_s - L_{ref}}{L_{ref}} \cdot 100\%, \qquad (2)$$

where L_{ref} is the length of the reference path, and L and L_s are the average and the shortest path lengths from all the trials of the algorithm for the particular problem, respectively. The PDM and PDB provide an overview of the algorithm quality. The PDM can be interpreted as an expected solution quality and the PDB as what can be achieved by the algorithm.

It is expected that the reference value would always be lower than L and L_s; however, it may happen that it would not be the case, because the reference solution is also only approximation as an optimal algorithm for the general MTP is not available. Therefore, negative values of the PDB and PDM indicate that the evaluated algorithm provides better solutions than the selected reference algorithm. An algorithm providing the reference solution is described in the next section.

3.2. Algorithms Comparison

Due to randomized nature of the SOM based algorithm, it is desired to compare the performance and results of the proposed modifications of SOM for the TSP statistically. Therefore, for each modified variant of the SOM algorithm and each particular problem 100 trials are performed in order to obtain representative samples of two evaluated distributions. The distributions are the required computational time of the SOM adaptation procedure, and the length of the path found. The algorithm comparison is based on statistical tests using a null hypothesis H_0, i.e., H_0 represents that the algorithms provide statistically identical results (regarding the required computational time, or the path length), and the alternative hypothesis is the results are different.

The required computational time is evaluated using the *Wilcoxon* test of the null hypothesis, because the distributions of the time are not Gaussian. The algorithms are considered different, i.e., one is faster than the other, if the p-values obtained by the Wilcoxon test are less than 0.001. In such a case, the difference between the required computational times of the compared algorithms is statistically significant.

The comparison according to reference paths is evaluated in a different way. It is because the reference path is found using a deterministic algorithm, which always provides the same solution, while paths found by the SOM-based algorithm differ due to randomization. Therefore, the *One-sample Wilcoxon* test is used as it is suggested by the authors of [38]. Similarly to the above comparison, once the p-values of the test statistics are less than 0.001, the null hypothesis is rejected; thus, the paths provided by the algorithms differ.

It is expected, the SOM algorithm would provide a bit worse solution (in terms of path length) than the reference algorithm, and therefore, an interesting question is how much are the solutions found by SOM worse than the reference solution. The One-sample Wilcoxon test is used to find such a qualitative measure. It is performed as follows. The reference path's length is iteratively increased by a given percentage level p_L, and such a length is compared with the distribution of the paths found by the SOM algorithms. Once the null hypothesis is accepted, the iterative procedure is terminated, and the current value of p_L denotes the desired qualitative measure. In a case that a basic quality of solution indicator (e.g., PDM) is negative, the reference path's length is decreased in similar manner, and the qualitative measure p_L indicates how much the SOM based solution is better than the reference one.

4. Reference Algorithm

The multi-goal path planning problem is de facto the TSP once the paths between goals are known. Therefore, for a point goals, approximate algorithms guaranteeing the solution quality of the TSP can be used for a particular restricted problem variant of the MTP. However, the authors of [39] note that approximate factors characterize algorithms in the worst case, which are often several times worse than the optimal solution, and such loose bounds are not valuable in real-world situations.

Similarly, the main drawback of the approaches addressing the safari or zoo-keeper route problems is their focus only on the particular restricted problem variant. Also a more general MTP formulation as the TSPN does not really help due to the complexity of the general TSPN. For restricted variants of the TSPN, the situation is similar to the safari route problems, i.e., the approaches are considered only in a plane.

The aforementioned reasons lead us to propose a practical approach to find a reference solution of the MTP, which will provide solution "good" enough for comparing it with the SOM based approaches, and which will also be easy to implement. The main idea of the proposed reference algorithm is based on a transformation of the MTP to the TPP using the optimal solution of the underlying TSP. The algorithm is following.

4.1. Transformation of the MTP with polygonal goals to the TPP

The main difference between the MTP and the TPP is that in the TPP, the sequence of goals visits is known; however, the difficulty is how to connect the consecutive goals, i.e., which point of each goal has to be visited in order to minimize the total traveled distance. Therefore, the TPP is obtained by solving the MTP as the TSP with point goals. Thus, for each polygonal goal a single point representative is determined.

Convex goals are assumed in the addressed problem, therefore a centroid of each polygonal goal can be used as a point goal in the TSP. More formally, let $\boldsymbol{G} = \{g_1, g_2, \ldots, g_n\}$ be a set of n convex goals in the polygonal map $\mathcal{W}$ with v vertices, and $c(g)$ denotes the centroid of the goal g. Then, all shortest paths between the centroids are found using Dijkstra's algorithm and the complete visibility graph that is found in $O((n+v)^2)$ [40]. Such an instance of the TSP with n point goals is solved exactly by the `concorde` solver [41]. The found solution of the TSP is then used to retrieve a sequence of the goals' visits for the consecutive TPP.

4.2. Approximate solution of the TPP

Even though approaches for optimal [11] or approximate solution of the TPP [35] have been proposed, their main drawback is that they consider goals only in a plane or in a simple polygon. Therefore, a simple approximate algorithm to deal with goals in the polygonal domain is proposed here. The algorithm is inspired by the iterative procedure proposed in [35] while the obstacles are addressed by sampling the boundary of each polygonal goal into a finite set of points. For simplicity, the sequence of goals' visits obtained from the solution of the TSP is $(g_1, g_2, \ldots, g_n)$ in the rest of this section.

Having a given sampling distance ρ and each polygonal goal g represented by a set of straight line segments $\boldsymbol{S}_g = \{s_1, s_2, \ldots, s_k\}$, the sampling is performed as follows. First, for each goal only segments entirely lying inside $\mathcal{W}$ are considered, as the path never goes through an obstacle. Then, each such a segment s is sampled using its end points. If the length of the segment $|s|$ is less than 2ρ then a middle point is an additional representative point of s, otherwise additional points are sampled equidistantly using ρ. At the end of the sampling, each goal g_i has associated a set of the representative points $\boldsymbol{P}_i$.

The reference solution of the MTP with polygonal goals is found as a path over the goals using the sequence of the representative points. The path is found by the following refinement procedure.

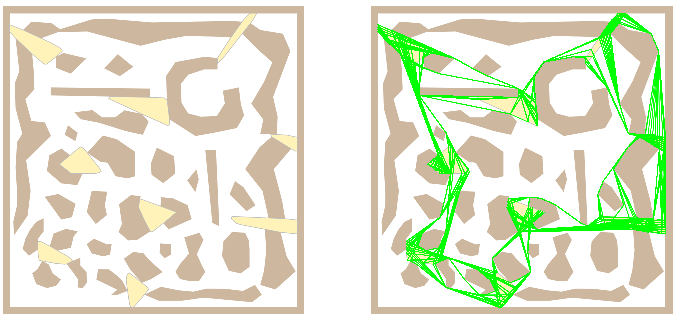
1. *Initialization:* Construct an initial touring polygons path using the first representative points of each polygonal goals. Let the path be defined by $Path = (p_1, p_2, \ldots, p_n)$, where $p_i \in \boldsymbol{P}_i$ is the selected representative point of the ith goal, and let $L = |Path|$ be the length of the shortest path induced by $Path$, e.g., found using the visibility graph of the points $\boldsymbol{P}_i$ in $\mathcal{W}$.
2. *Refinement:*
   - For $i = 1, 2, \ldots, n$
     - Find $p_i^* \in \boldsymbol{P}_i$ minimizing the length of the path $d(p_{i-1}, p_i^*) + d(p_i^*, p_{i+1})$, where $d(p_k, p_l)$ is the length of the shortest path (among obstacles) from p_k to p_l, $p_0 = p_n$, and $p_{n+1} = p_1$.
     - If the total length of the current path over point p_i^* is shorter than over p_i, replace the point p_i by p_i^*.
   - Compute new path length L_{new} using eventually refined representative points.
3. *Termination condition:* If $L_{new} - L < \epsilon$ Stop the refinement. Otherwise $L \leftarrow L_{new}$ and go to Step 2.
4. *Final path construction:* Use the last sequence of the representatives points of the goals and construct the path using the shortest paths among obstacles between two consecutive points.

The refinement procedure is repeated until the change of the path length is not significant (smaller than the value ϵ). The value of ϵ can be set arbitrarily, but it is clear a smaller value improves the solution quality. Similarly a smaller value of the sampling distance ρ can provide a better path; however, it also increases the number of representatives, thus increases the computational burden. The algorithm has been used to find a reference solution of the problem set presented in Section 6.1. During the computation, $\epsilon = 0$ has been used and it has also been observed that the quality of found solutions is almost independent of the value of ρ, e.g., for ρ less than 0.1 m (except the jh_{10}-*coverage* problem). It is caused mainly because the convex goals are formed from segments that typically contains several segments with length about 0.1 m, because the goals are created on top of a triangular mesh of the polygonal maps.

4.3. Comments

An eventual issue of the proposed reference algorithm could be high computational requirements for a high number of the representative points. This is mainly related to the computation of the visibility graph and determination of the shortest paths among obstacles. Moreover, if the distance matrix (or paths) are pre-computed and stored in a memory, the algorithm can be computationally infeasible due to memory requirements. In particular, this is the case of the jh_{10}-*coverage* problem, which represents an instance of the WRP.

These issues can be resolved using approximate shortest path between two points in $\mathcal{W}$ [29], which is principally similar to the node–goal path described in Section 2.1. For the problems examined in this paper the approximation provides the same paths, while it is up to two orders of magnitude faster than the pre-computation of the required shortest path and construction of the complete visibility graph using the approach [40], because of saved initialization phase.



(a) (b)

Figure 3: An example of the shortest between goals, (a) problem $dense_5$-A, (b) shortest paths between consecutive goals.

Here, it should be noted that the number of required paths is relatively small, as only paths between two consecutive goals need to be determined. In Fig. 3, polygonal goals of the $dense_5$-A problem and particular paths between each consecutive goals are presented.

The optimal solution of the TSP can be computationally demanding, therefore a heuristic algorithm like the Chained Lin-Kernighan approach [42] can be a more practical approach. However, the optimal solution together with the proposed solution of the TPP provide strictly deterministic approach, which does not require statistical evaluation of the reference solutions; thus, it simplifies the algorithms' comparison a bit.

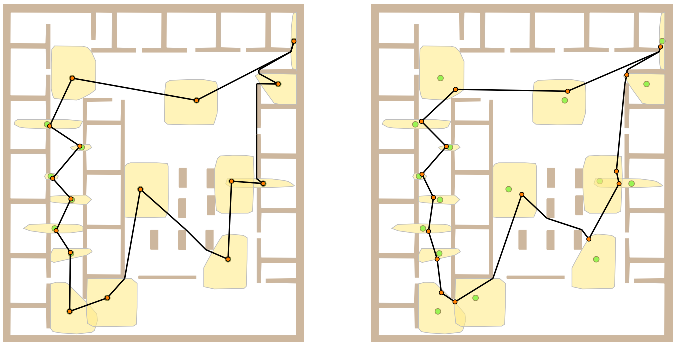
Convexity of the Goals and Randomization

Although the convexity of the goals is used for determining the representative points for the TSP as the centroids of the polygonal goals, the convexity is not mandatory. Alternatively any point can be used as a representative of the polygonal goal, because the sequence of polygons' visits is retrieved from the TSP; thus, each point is associated to the selected polygon.

The proposed reference algorithm is strictly deterministic; however, it can be straightforwardly randomized. First, the initial path can be created from a randomly selected sampled point of each polygon. In addition, each loop of the refinement can be started from a random goal. Such a randomization has been extensively evaluated and its significant benefit has not been observed as it mainly affects the number of required refinements to find the same final path. It is because the refinement itself is very fast, while the initialization phase using the visibility graph and pre-computed shortest path is computationally demanding.

5. Adaptation Rules for Polygonal Goals

Although it is obvious that a polygonal goal can be sampled into a finite set of points and the problem can be solved as the MTP with partitioned goals, the aforementioned SOM procedure can be straightforwardly extended to sample the goals during the self-adaptation. Thus, instead of explicit sampling of the goals three simple strategies how to deal with adaptation toward polygonal goals are presented in this section. The proposed algorithms are based on SOM for the TSP using centroids of the polygonal goals as point goals, see Section 2.1. However, the *select winner* and *adapt* phases are modified to find a more appropriate point of the polygonal goal and to avoid unnecessary movement into the goal. Therefore, a new point representing a polygonal goal is determined during the adaptation and used as a point goal, which leads to computation of a shortest path between two arbitrary points in $\mathcal{W}$. Similarly to the node–goal queries an approximate node–point path is considered to decrease the computational burden. The approximation is also based on a convex partition of $\mathcal{W}$ and the shortest path over cells' vertices (detailed description can be found in [29]).



(a) a path found using termination of the adaptation if all winners are inside goals, L=78.4 m

(b) a path found with avoiding adaptation of winners inside goals, L=65.0 m

Figure 4: Examples of found paths without and with consideration of winners inside the goals. Goals are represented by yellow regions with small green disks representing the centroids of the regions. Winner nodes are represented by small orange disks. The length of the found path is denoted as L.

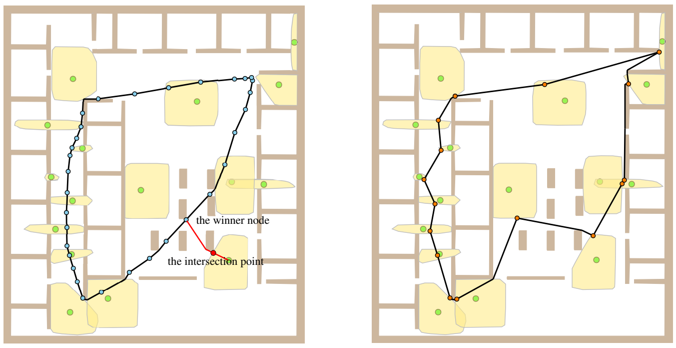
5.1. Interior of the Goal

Probably the simplest approach (called *goal interior* here) can be based on the regular adaptation to the centroids of the polygonal goals. However, the adaptation, i.e., the node movement toward the centroid, is performed only if the node is not inside the polygonal goal. Determination if a node is inside the polygonal goal with n vertices can be done in $O(n)$ computing the winding number or in $O(\log n)$ in the case of a convex goal. So, in this strategy, the centroids are more like attraction points toward which nodes are attracted because the adaptation process is terminated if all winner nodes are inside the particular polygonal goals. Then, the final path is constructed from a sequence of winner nodes using the approximate shortest node–node path. An example of solutions using the new termination condition and with the avoiding adaptation of winners inside the goals is shown in Fig. 4.

This adaptation strategy clearly demonstrates one of the SOM's advantages that is ability to reflect local properties of the environment during the adaptation, which, in this case, is the test if a node is inside the polygonal goal.

5.2. Attraction Point

The strategy described above can be extended by determination of a new attraction point at the border of the particular polygonal goal toward which is being adapted. First, a winner node $\nu^{\star}$ is found regarding its distance to the centroid $c(g)$ of the goal g. Then, an intersection point p of g with the path $S(\nu, c(g))$ is determined. The point p is used as a point goal to adapt the winner and its neighbouring nodes. This modification is denoted as *attraction* in the rest of this paper.



(a) an intersection point

(b) a path found, L=59.5 m

Figure 5: Examples of an intersection point and a path found using the *attraction* algorithm variant, blue disks are nodes.

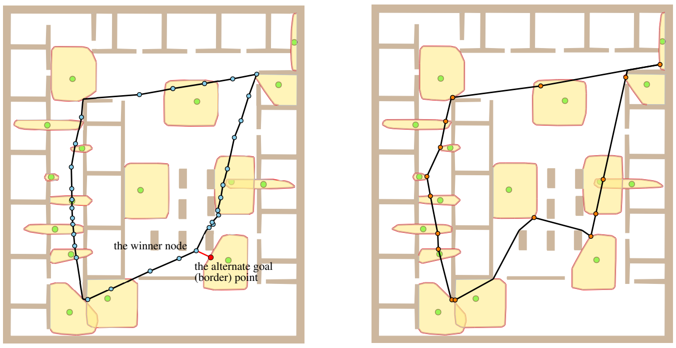
An example of determined intersection point p and the final path found is shown in Fig. 5. The path is about five meters shorter than a path found by avoiding adaptation of winner nodes inside the goals. Determination of the intersection point increases the computational burden, and therefore, an experimental evaluation of the proposed algorithm variants is presented in Section 6.

5.3. Selection of Alternate Goal Point

A polygonal goal can be visited using any point of its border. The closest point at the goal border to a node can be determined in the winner selection phase. To find such a point, straight line segments forming the goal are considered instead of the goal centroid. Moreover, a goal can be attached to the map, and therefore, only segments laying inside the free space of $\mathcal{W}$ are used. Let $\boldsymbol{S}_g = \{s_1, s_2, \ldots, s_k\}$ be the border segments of the polygonal goal g that are entirely inside $\mathcal{W}$. Then, the winner node ν^{*} is selected from a set of non-inhibited nodes regarding the shortest path $S(\nu, s)$ from a point ν to the segment $s, s \in \boldsymbol{S}_g$. Beside the winner node, a point p at the border of g is found in the winner selection procedure as a result of determination of $S(\nu, s)$. The border point p is then used as an alternate point goal for the adaptation, therefore this modification is denoted as *alternate goal*.

Determination of the exact shortest point–segment path can be too computationally demanding, therefore the following approximation is considered. First, the Euclidean distance between the node ν and the segment s is determined. If the distance is smaller than the distance of the current winner node candidate, then the resulting point p of s is used to determine an approximate path among obstacles between p and ν. If $|S(p, \nu)|$ is shorter than the path length of the current winner node candidate to its border point, ν becomes the new winner candidate and p is the current alternate goal (border) point.

Even though this modification is similar to the modification described in Section 5.2, it provides sampling of the goal boundary with a less distance of the goal point to the



(a) an alternate goal point (b) a path found, L=57.4 m

Figure 6: An example of the alternate goal point and the final path found. Red straight line segments around the goal regions denote parts of the goal border inside the free space of $\mathcal{W}$.

winner node; thus, a shorter final path can be found. An example of found alternate goal point and the path found is shown in Fig. 6.

6. Results

6.1. Problems Description

The proposed adaptation rules in Section 5 have been experimentally verified in a set of problems. Due to lack of commonly available multi-goal path planning problems with polygonal goals several problems have been created within a map of real and artificial environments[1]. An overview of the basic properties of the environments is shown in Table 1. Maps *jh*, *pb*, and *h2* represent real environments (building plans), and maps *dense* and *potholes* are artificial environments with many obstacles.

Table 1: Properties of environments and their polygonal representation

Map	**Dimensions** [m × m]	**No. vertices**	**No. holes**
jh	20.6 × 23.2	196	9
pb	133.3 × 104.8	89	3
h2	84.9 × 49.7	1 061	34
dense	21.0 × 21.5	288	32
potholes	20.0 × 20.0	153	23

Sets of polygonal obstacles have been placed within the maps in order to create representative multi-goal path planning problems. The name of the problem is derived from the name of the map, considered visibility range d in meters written as a subscript, and particular problem variant, e.g., the problem name is in a form map_d*-variant*. The value of d restricts the size of the convex polygonal goal, i.e., all vertices of each goal are closer than d. An unrestricted visibility range is considered in problems without the subscript. The convex polygonal goals are found on top of the triangular mesh, details about the used procedure can be found in [29].

The problems have been designed in order to create representative problems particularly focused on specific characteristics. Here, a short description of motivation behind their design is provided to present the main aim of the problems. The problems are visualized in Fig. 7, where the centroids of the convex goals and reference paths found are showed as well. The map *jh* represents an office-like environment with many rooms. Therefore, several problems within this map are created with a motivation of patrolling or inspection tasks. The jh_4*-A* problem variant is a general problem with goals in few rooms and partially trespassing to corridors. In the jh_5*-corridors* and *jh-rooms* problems, the expected path should not enter to rooms. Thus, the aim of these problems is to demonstrate the ability of the evaluated algorithms to take an advantage of the polygonal goals, as the centroids are located relatively far from the border, and visitations of the centroids unnecessary increase the path.

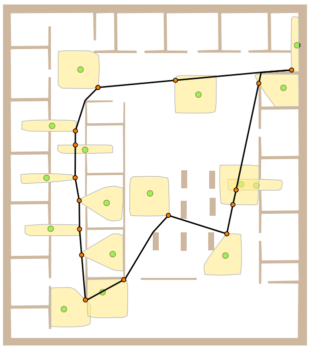
The jh_{10}*-coverage* problem represents an instance of the WRP with restricted visibility range, and therefore, the algorithm's performance in this problem can indicate a flexibility of the tested approach.

The problem $h2_5$*-A* is within a large map, and it is included in the problem set mainly because of the map's complexity; thus, it serves as a load and study of the algorithm's performance in maps with many vertices, e.g., to study the influence of the supporting algorithms like the approximate shortest path. The pb_5*-A* problem is a very simple problem; however, a solution can stuck in a local optima, due to the goal in the middle of the map. The *dense* map is a complicated environment, and therefore, several alternative paths connecting the goals exist, e.g., see Fig. 3. In addition, the *dense-small* problem contains several goals that are inside another goals. These goals can show the ability to avoid focus of the algorithm on the larger goals, as the visit of the inside goals is mandatory. Finally, the problem $potholes_2$*-A* contains many small obstacles, which are relatively sparse. The "right" sequence of goals visit is relatively easy to find; however, the final path length depends on a proper selection of the points at the border of the polygonal goals.

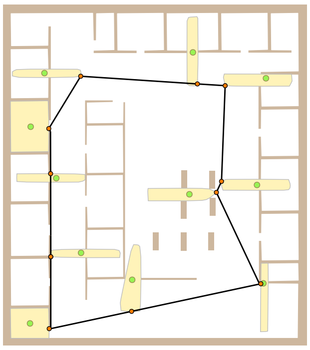
6.2. Results

Each problem of the aforementioned problem set has been solved using the reference algorithm described in Section 4 and three SOM based algorithms proposed in Section 5. Due to randomization of the SOM based algorithms, 100 trials have been performed for each algorithm and problem; thus, the total number of found solutions

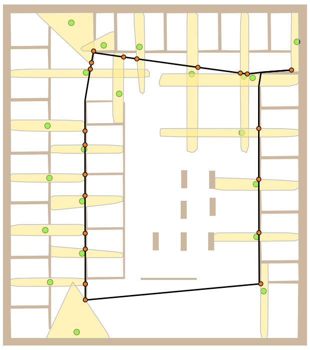
[1]All problems and supporting materials are available at `http://purl.org/faigl/safari`



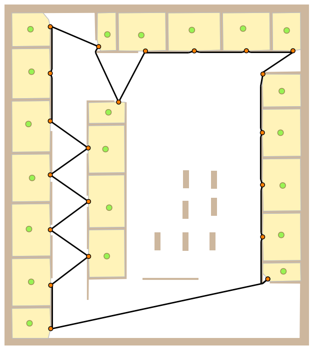
(a) jh_4-A, n=16, L_{ref}=56.6 m



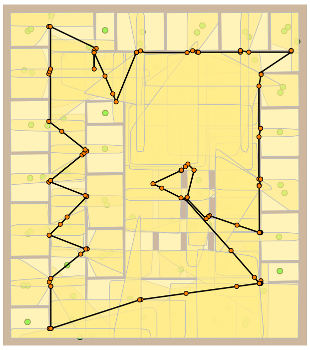
(b) jh_5-*corridors*, n=11, L_{ref}=59.5 m



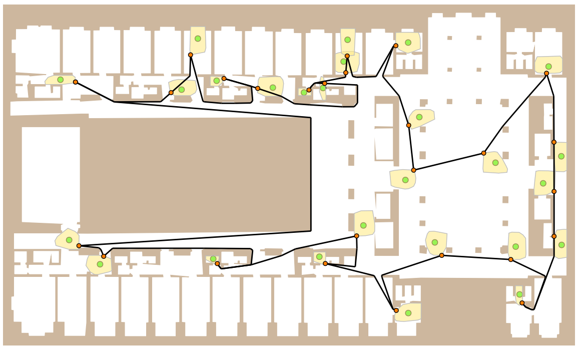
(c) jh_{10}-*doors*, n=21, L_{ref}=62.1 m



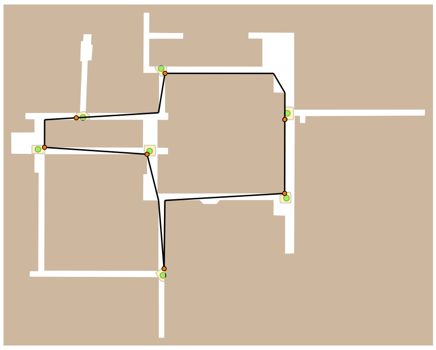
(d) *jh-rooms*, n=21, L_{ref}=87.6 m



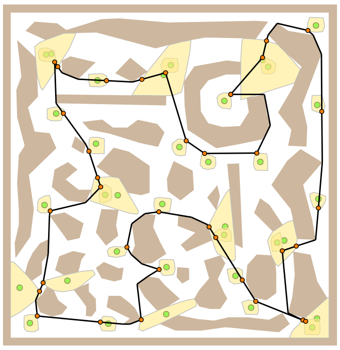
(e) jh_{10}-*coverage*, n=106, L_{ref}=108.9 m



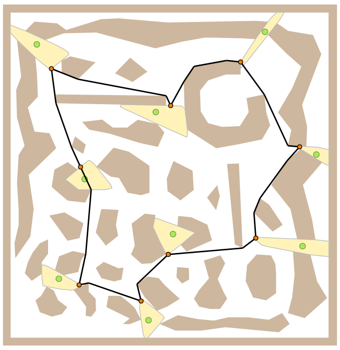
(f) $h2_5$-A, n=26, L_{ref}=395.0 m



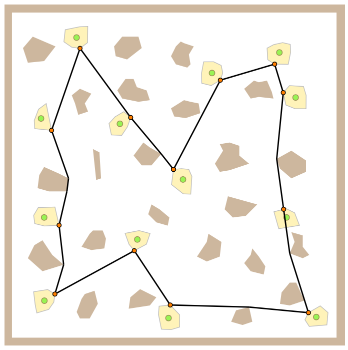
(g) pb_5-A, n=7, L_{ref}=263.7 m



(h) *dense-small*, n=35, L_{ref}=103.5 m



(i) $dense_5$-A, n=9, L_{ref}=57.9 m



(j) $potholes_2$-A, n=13, L_{ref}=68.4 m

Figure 7: The problems examined and the reference solutions found, n is the number of goals, L_{ref} is the length of the reference path, yellow regions are polygonal goals, and small green disks are the goals' centroids.

by the SOM's approaches is 3000 for all problems of the problem set.

All results have been obtained using the same computational environment consisting of a C++ implementation of the algorithms compiled by the G++ version 4.6 with -O2 optimization, a single core of the i7-970 CPU at 3.2 GHz, 12 GB RAM, and 64-bit version of the FreeBSD 8.2. Therefore, all the required computational times presented can be directly compared.

The basic quality indicators (described in Section 3.1) are presented in Table 2. Regarding the presented PDM values, it is clear that a more sophisticated adaptation rules provide better results. However, it seems it is not the case of the computational complexity indicated by the column T, where the average of the required computational time of the adaptation is presented. A statistical evaluation of the results is presented in Table 3, where two algorithms are compared using the null hypothesis approach described in Section 3.2. Once the null hypothesis is rejected (the statistics are not same), an additional null hypothesis is evaluated to determine if one algorithm is statistically better than the another one, i.e., using the average required computational time T_a and the average length of the path found L. The adaptation rules proposed

Table 2: Results of the proposed SOM adaptation rules

Problem	n	L_{ref} [m]	**goal interior** PDM	**goal interior** PDB	**goal interior** T_a [ms]	**attraction** PDM	**attraction** PDB	**attraction** T_a [ms]	**alternate goal** PDM	**alternate goal** PDB	**alternate goal** T_a [ms]
dense$_5$-A	9	57.9	17.30	10.87	12	7.11	3.62	16	1.81	0.24	23
dense-small	35	103.5	14.69	7.99	171	10.54	3.90	205	8.58	-0.13	274
h2$_5$-A	26	395.0	6.89	3.95	130	2.98	1.06	160	1.89	0.22	210
jh$_{10}$-coverage	106	108.9	22.91	15.40	872	-2.63	-7.62	1040	-13.75	-14.53	1578
jh$_{10}$-doors	21	62.1	14.86	8.51	35	8.79	5.53	38	0.38	-0.04	63
jh$_4$-A	16	56.6	16.95	11.34	24	7.43	3.18	28	0.69	0.18	45
jh$_5$-corridors	11	59.5	17.06	12.02	13	10.38	7.01	16	0.84	0.12	21
jh-rooms	21	87.6	17.75	13.82	53	0.78	0.18	68	0.61	0.21	73
pb$_5$-A	7	263.7	4.18	1.23	5	2.87	0.17	7	3.42	0.07	11
potholes$_2$-A	13	68.4	7.58	4.57	18	3.12	0.91	23	3.01	0.53	29

are incremental, and therefore, the *attraction* variant is compared against the *goal interior* variant, and similarly *alternate goal* is compared against *attraction*. Because all the p-values are very small, characters '-', '+', and '=' are used to denote that an algorithm is slower, a solution is better, or the performance indicators are statistically identical.

Table 3: Comparison of SOM based algorithms

Problem	a_1: **attraction** / a_2: **goal interior** *length*	*time*	a_1: **alternate goal** / a_2: **attraction** *length*	*time*
dense$_5$-A	+	-	+	-
dense-small	+	-	+	=
h2$_5$-A	+	-	+	-
jh$_{10}$-coverage	+	-	+	-
jh$_{10}$-doors	+	-	+	-
jh$_4$-A	+	-	+	=
jh$_5$-corridors	+	-	+	-
jh-rooms	+	-	=	-
pb$_5$-A	+	-	=	-
potholes$_2$-A	+	-	=	=

+ - the algorithm a_1 provides better paths than a_2.

The results show that the reference algorithm provides better solutions (except for the problem *jh*$_{10}$*-coverage*). Table 4 presents a deeper insight to the performance characteristics of the reference and SOM based algorithms. First, an estimation of the approximate factor of the SOM based algorithm is shown in the column $p_L\%$, viz Section 3.2. For the *alternate goal* algorithm, this factor is mostly less than one percent; however, the required computational time is significantly smaller (more than three orders of magnitude) than for the reference algorithm. Notice the time T of the reference algorithm does not include the time needed to find the optimal solution of the TSP. The computational burden of the reference algorithm is caused by the computation of the visibility graph and all shortest paths between points of two consecutive goals. The number of the used points is denoted by n_p in the table, and it is significantly higher than the number of goals due to sampling of the goals' borders. The refinement itself is very fast as all required distances are pre-computed, therefore only the total time to solve the TPP is presented in the column T. On the other hand, the required computational time T of the SOM based algorithms consists of the time to initialize the supporting structures (for the approximate shortest path) T_{init}, which is shown as a number of percentage points of T in the column $T_{init}\%$, and the adaptation time T_a. The initialization itself consists of construction of the convex partitioning and visibility graphs, and computation of all shortest paths between the map vertices (and centroids of the convex goals)[2]. The required computational times of the constructions are presented in Table 5 and are negligible regarding the time to compute the shortest path, which is indicated by T_{init}. The partition is found by Seidel's algorithm [43] and the number of convex polygons utilized in the approximation of the shortest path is presented in the second column. The visibility graph is found using [40].

The computational requirements of the reference algorithm using the pre-computed shortest paths are very high. This is especially significant for the problem *jh*$_{10}$*-coverage*. Beside the required computational time, the required memory footprint to store the pre-computed paths and distances is about 6.5 GB[3] for $\rho = 0.05$ and it rapidly increases for a lower ρ, which makes this approach un-

[2] Note these shortest paths are also required for the optimal solution of the TSP.

[3] Using a regular implementation of the distance matrix.

Table 4: Comparison of the reference algorithms with the SOM based algorithms

Problem	**reference (TPP part)**			**goal interior**			**attraction**			**alternate goal**		
	n_p	L [m]	T [s]	p_L%	T [s]	T_{init}%	p_L%	T [s]	T_{init}%	p_L%	T [s]	T_{init}%
dense$_5$-A	2 470	57.9	4.2	16.5	0.06	80	6.1	0.06	74	0.7	0.06	63
dense-small	7 857	103.5	91.7	13.9	0.22	22	9.3	0.25	19	8.0	0.33	17
h2$_5$-A	10 873	395.0	216.7	6.2	0.89	85	2.9	0.93	83	0.9	0.92	77
jh$_{10}$-coverage	23 720	108.9	3 033.4	22.1	0.90	3	-2.0	1.06	2	-13.8	1.60	1
jh$_{10}$-doors	12 733	62.1	552.0	13.5	0.06	39	7.9	0.06	38	0.3	0.08	19
jh$_4$-A	6 773	56.6	97.9	15.8	0.03	25	6.8	0.04	33	0.6	0.06	24
jh$_5$-corridors	4 757	59.5	42.4	16.3	0.03	54	9.5	0.04	59	0.7	0.04	51
jh-rooms	989	87.6	0.7	17.0	0.07	22	0.5	0.08	18	0.5	0.10	23
pb$_5$-A	3 664	263.7	11.1	2.3	0.01	46	0.8	0.01	37	0.5	0.01	27
potholes$_2$-A	3 050	68.4	16.3	6.5	0.03	43	2.0	0.04	38	1.9	0.04	33

Table 6: Comparison of the reference algorithms for the TPP

Problem	n	n_p	**visibility graph**			**approx. shortest path**		
			L [m]	T [s]	T_{init}%	L [m]	T [s]	T_{init}%
dense$_5$-A	9	546	57.9	0.35	100	57.9	0.08	76
dense-small	35	1 733	103.5	2.71	100	103.5	0.07	54
h2$_5$-A	26	2 285	395.0	7.34	100	395.0	0.86	92
jh$_{10}$-coverage	106	12 274	110.0	499.93	100	113.5	0.54	4
jh$_{10}$-doors	21	2 684	62.1	8.60	100	62.1	0.10	21
jh$_4$-A	16	1 458	56.6	2.14	100	56.6	0.05	45
jh$_5$-corridors	11	1 015	59.5	1.05	100	59.5	0.02	37
jh-rooms	21	227	87.7	0.08	99	87.7	0.03	84
pb$_5$-A	7	784	263.7	0.32	100	263.7	0.01	29
potholes$_2$-A	13	644	68.4	0.51	100	68.4	0.02	63

Table 5: Required computational times for preparing supporting structures

Map	**No. convex polygons**	$T_{partition}$ [ms]	$T_{visibility}$ [ms]
jh	77	12	4.0
pb	41	10	0.7
h2	476	65	24.0
dense	150	12	1.8
potholes	75	8	0.7

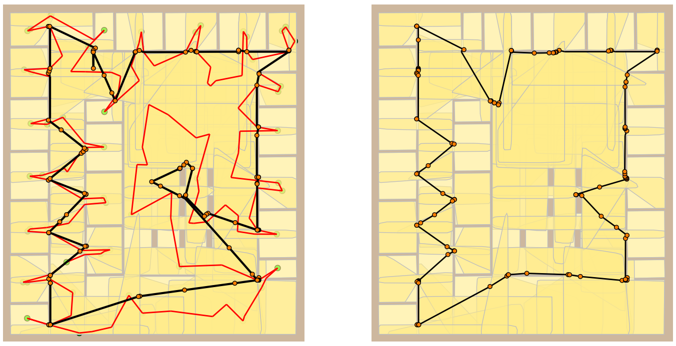
suitable for a high number of representative points. On the other hand, the approximation of the shortest paths used in the SOM based algorithms can also be used for the approximate TPP algorithm. In Table 6, basic performance indicators are presented for variants based on the exact shortest paths using the visibility graph and approximate shortest paths. In both variants, the sampling distance ρ is set to 0.1 m. Note the refinement itself is very fast, and it is done in a fraction of the initialization time, e.g., in units of microseconds. Although the approximation reduces the initialization time (indicated in the column T_{init}%), the initialization is still a significant part of the total required computational time as the initialization is a pre-computation of all shortest paths between map vertices and only several refinement steps are needed to find a final solution. The approximation provides same results as the exact shortest path except the problem *jh$_{10}$-coverage*, where the final path found is about three percentage points worse due to limited precision of the approximation. However, the total required computational time and also required memory are significantly smaller. Thus, the approximation seems be sufficient for these problems.

Based on the results, the best solutions found for each problems and over all approaches have been selected for

Table 7: Best solutions found

Problem	n	L_{best} [m]
dense$_5$-A	9	57.9
dense-small	35	*103.4
h2$_5$-A	26	395.0
jh$_{10}$-coverage	106	*93.1
jh$_{10}$-doors	21	*62.0
jh$_4$-A	16	56.6
jh$_5$-corridors	11	59.5
jh-rooms	21	87.6
pb$_5$-A	7	263.7
potholes$_2$-A	13	68.4

*The solution is found by the *alternate goal* algorithm.



(a) the reference algorithm, L=108.9 m

(b) the *alternate goal* algorithm, L_{best}=93.1 m

Figure 8: The best solutions of *jh$_{10}$-coverage* found by the reference and *alternate goal* algorithms, the optimal solution of the related TSP is in red.

(a) the reference algorithm, L_{ref}=103.5 m

(b) the *alternate goal* algorithm, L_{best}=103.4 m

Figure 9: The best solutions found for the problem *dense-small* by the reference and *alternate goal* algorithms.

a further comparison. The length of the best paths are depicted in Table 7.

6.3. Discussion

The presented results provide a performance overview of the proposed adaptation rules. The principle of the *attraction* and *alternate goal* algorithm variants are very similar; however, the *alternate goal* variant provides better results. The advantage of *alternate goal* is sampling of the goals' borders. Even though a simple approximation of the shortest path between a node (point) and the goal's segment is used, a precision of the approximation increases with the node movements toward the goal, and therefore, a better point of the goal is sampled. This is an import benefit of the SOM adaptation, which allows usage of a relatively rough approximation of the shortest path.

On the other hand, the *attraction* algorithm variant is a more straightforward, as the path to the centroid is utilized as a path to the fixed point goal. The fixed point goals allow to use pre-computed all shortest paths from map vertices to the goals, which improves precision of the approximate node–goal path. In addition, such approximation is less computationally intensive in the cost of higher memory requirements. However, this benefit is not evident from the results, because the *alternate goal* variant provides a faster convergence of the network.

The statistical comparison of the SOM-based algorithms provides a strong statistical evidence (as the p-values obtained by the Wilcoxon test are almost always less than 0.001) that the variants proposed are different. In particular, a more sophisticated rule provides better solutions. Even though the required computational times also increase, the differences between the *attraction* and *alternate goal* variants are small and in few cases statistically identical.

The reference algorithm provides better results than average solutions of the SOM based algorithms, except the problem *jh$_{10}$-coverage*, which is an instance of the WRP with a restricted visibility range. In this particular problem, the path based on the optimal solution of the related TSP does not provide competitive solution to the paths found by the SOM approach (regarding the PDM as well as p_L%), see Fig. 8. This indicates unsuitability of the pure combinatorial approaches for the WRP.

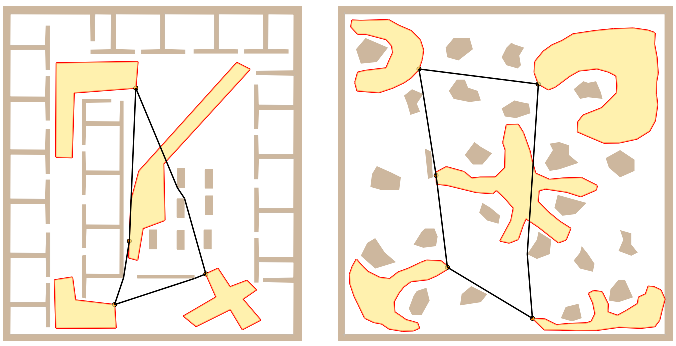
On the other hand, a worse average performance of the *alternate goal* algorithm is in the *dense-small* problem. In this problem, the SOM based solver stuck at a local optima due to many obstacles; however, the best solution found over 100 trials is better than the reference solution. The best solutions are pretty much similar as can be seen in Fig. 9. In other problems, the differences in the final path length are very small and they are caused by sampling of the convex goals' boundary, i.e., a small change of the final visiting point can decrease the path a bit. Besides, the final path of SOM based solutions are determined using the approximate shortest path; thus, the approximation can also affect the solution quality.

The above two examples of solutions demonstrate that better solutions are obtained for a different sequence of goals' visits than the sequence prescribed by the optimal solution of the TSP for the centroids. In the jh_{10}-*coverage* problem, many goals overlap each other, and therefore, centroids are not suitable representation; thus, it is not surprising the solution based on the TSP is worse. However, the *dense-small* problem also indicates that a bit better solution can be achieved for a different sequence. Notice, the solutions in Fig. 9 are almost identical, except the part approximately in the middle of the map. These examples demonstrate an advantage of the SOM based approach that includes the solution of the TSP and selection of the appropriate points of visits in a single unified way.

It should also be noted that an optimal solution of the TSP can be computationally demanding due to NP-hardness of the TSP. Therefore, regarding the results, the overall comparison of the solution quality, and the required computational time the *alternate goal* approach provides an acceptable trade-off between these two performance indicators. In addition, it also provides a greater flexibility than the reference algorithm based on a solution of the TPP, as it scales better for a more complex problems with many goals, and it also includes an approximate solution of the TSP.

6.3.1. Non-convex goals

Although convex goals are assumed in the problem formulation, the presented adaptation rules do not depend on the goal convexity. The convex goals are advantageous in visual inspection tasks (covering tasks), because the whole goal region is inspected by visiting the goal at any point of the goal. Also a point representative of the convex goal can be simply computed as the centroid. If a goal is not convex a point that is inside the goal has to be determined for the *goal interior* and *attraction* algorithms. Basically any point inside the goal can be used, but a bias toward the point can be expected. The *alternate goal* algorithm variant uses a set of segments representing the goal, and therefore, this algorithm can be directly used for problems with non-convex goals (see Fig. 10), which is an additional advantage of the SOM based approach for the MTP.



(a) the *jh* environment (b) the *potholes* environment

Figure 10: Solutions found by the *alternate goal* algorithm for problems with non-convex goals.

7. Conclusion

A self-organizing map based algorithm for the multi-goal path planning problem in the polygonal domain has been presented. Three variants of the algorithm addressing polygonal goals have been proposed and experimentally evaluated for a set of problems including an instance of the WRP with a restricted visibility range (jh_{10}-*coverage*). Even though the solution quality is not guaranteed because of SOM, regarding the experimental results the algorithms provide high quality solutions. The advantage of the proposed *alternate goal* algorithm is that it provides a flexible approach to solve various routing problems including the TSP, WRP, safari route problems, and their variants in the polygonal domain, and eventually with non-convex goals.

From a practical point of view, the SOM algorithms proposed are based on relatively simple algorithms and supporting structures, which is an additional benefit. The SOM adaptation schema is not a typical technique used for routing problems motivated by robotics applications. The presented results demonstrate flexibility of SOM based algorithm; thus, they may encourage roboticists to consider SOM as a suitable planning technique for other multi-goal path planning problems.

Beside the SOM approaches, a simple and straightforward reference algorithm has been presented. It provides an easily reproducible reference solutions of the examined problems with polygonal goals. Therefore additional problems can be proposed to create a set of problems for benchmarking further multi-goal path planning algorithms. An initial set of such problems are provided [44] together with the reference solutions found by the proposed approaches.

Although the proposed algorithms are able to deal with non-convex goals, the adaptation rules need a further development and an additional evaluation in such problems, which is a subject of our future work.

Acknowledgment

This presented work was supported by the Technology Agency of the Czech Republic under Project No. TE01020197 an by the Ministry of Education of the Czech Republic under Project No. LH11053 and partially by Project No. 7E08006, and by the EU Project No. 216342.

Appendix A. Nomenclature

$\mathcal{W}$	a polygonal map representing the robot workspace, $\mathcal{W} \subset \mathbb{R}^2$
$\boldsymbol{G}$	a set of (polygonal) goals to be visited
g	a goal, $g \subset \mathcal{W}$
n	a number of goals, $n = \lvert\boldsymbol{G}\rvert$
$\boldsymbol{S}_g$	segments forming the goal g
n_p	a number of points representing all goals (for the reference algorithm)
$c(g)$	a centroid of the goal g
$S(p, g)$	an approximate path from p to g
$\lvert S(p, g)\rvert$	a length of the approximate path $S(p, g)$
L	a length of the path found
L_{ref}	a length of the reference path
L_{best}	a length of the best path found
$\mathcal{N}$	a set of nodes
ν	a node (neuron weights), $\nu \in \mathcal{W}$, $\nu \in \mathcal{N}$
σ	a learning gain (neighbouring function variance)
$f(\sigma, l)$	a neighbouring function
α	a gain decreasing rate
μ	a learning rate
PDM	the percent deviation from the reference path's length of the mean solution value
PDB	the percent deviation from the reference of the best solution value
$p_L\%$	an estimation of the approximate factor of SOM based solutions to the reference solution
T	the required computational time (or its average value in the case of SOM algorithms)
T_a	the required computational of the adaptation phase
$T_{init}\%$	the part (in percents) of T spent in the initialization

References

[1] S. N. Spitz, A. A. G. Requicha, Multiple-Goals Path Planning for Coordinate Measuring Machines, in: ICRA, 2000, pp. 2322–2327.

[2] M. Saha, T. Roughgarden, J.-C. Latombe, G. Sánchez-Ante, Planning Tours of Robotic Arms among Partitioned Goals, Int. J. Rob. Res. 25 (3) (2006) 207–223.

[3] J. Faigl, M. Kulich, L. Přeučil, A sensor placement algorithm for a mobile robot inspection planning, Journal of Intelligent & Robotic Systems 62 (3) (2011) 329–353.

[4] M. Kulich, J. Faigl, L. Přeučil, Cooperative planning for heterogeneous teams in rescue operations, in: IEEE International Workshop on Safety, Security and Rescue Robotics, 2005, pp. 230–235.

[5] T. Danner, L. Kavraki, Randomized Planning for Short Inspection Paths, in: Proceedings of The IEEE International Conference on Robotics and Automation (ICRA), IEEE Press, San Fransisco, CA, 2000, pp. 971–976.

[6] S. Ntafos, Watchman routes under limited visibility, Computational Geometry: Theory and Applications 1 (3) (1992) 149–170.

[7] X. Tan, T. Hirata, Finding shortest safari routes in simple polygons, Information Processing Letters 87 (4) (2003) 179–186.

[8] S. Bespamyatnikh, An $o(n \log n)$ algorithm for the Zoo-keeper's problem, Computational Geometry: Theory and Applications 24 (2) (2002) 63–74.

[9] J. Faigl, M. Kulich, V. Vonásek, L. Přeučil, An Application of Self-Organizing Map in the non-Euclidean Traveling Salesman Problem, Neurocomputing 74 (2011) 671–679.

[10] J. Kubalík, J. Kléma, M. Kulich, Application of soft computing techniques to rescue operation planning., in: Artificial Intelligence and Soft Computing (ICAISC), Springer, 2004, pp. 897–902.

[11] M. Dror, A. Efrat, A. Lubiw, J. S. B. Mitchell, Touring a sequence of polygons, in: STOC '03: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing, ACM Press, San Diego, CA, USA, 2003, pp. 473–482.

[12] H. González-Baños, D. Hsu, J.-C. Latombe, Motion planning: Recent developments, in: S. Ge, F. Lewis (Eds.), Autonomous Mobile Robots: Sensing, Control, Decision-Making and Applications, CRC Press, 2006, Ch. 10.

[13] R. Honsberger, Mathematical Gems II, Mathematical Association of America, 1979.

[14] J. O'Rourke, Galleries need fewer mobile guards: a variation on Chvátal's theorem, Geometriae Dedicata 14 (1983) 273–283.

[15] T. Shermer, On recognizing unions of two convex polygons and related problems, Pattern Recognition Letters 14 (9) (1993) 737–745.

[16] S. Fisk, A short proof of Chvatal's watchman theorem, Journal of Combinatorial Theory Series B 24 (1978) 374+.

[17] M. C. Couto, C. C. S. de Souza, P. J. de Rezende, An Exact and Efficient Algorithm for the Orthogonal Art Gallery Problem, in: SIBGRAPI '07: Proceedings of the XX Brazilian Symposium on Computer Graphics and Image Processing, IEEE Computer Society, Washington, DC, USA, 2007, pp. 87–94.

[18] J. C. Culberson, R. A. Reckhow, Covering polygons is hard, Journal of Algorithms 17 (1) (1994) 2–44.

[19] M. C. Couto, P. J. de Rezende, C. C. de Souza, An IP solution to the art gallery problem, in: SCG '09: Proceedings of the 25th annual symposium on Computational geometry, ACM, New York, NY, USA, 2009, pp. 88–89.

[20] Y. Amit, J. S. B. Mitchell, E. Packer, Locating guards for visibility coverage of polygons, Int. J. Comput. Geometry Appl. 20 (5) (2010) 601–630.

[21] A. Tabatabaei, A. Mohades, Clarity Watchman Route, in: The Kyoto International Conference on Computational Geometry and Graph Theory, Kyoto, Japan, 2007.

[22] G. Kazazakis, 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 2002), Vol. 3, Lausanne, Switzerland, 2002, pp. 2843–2848.

[23] H. González-Baños, A randomized art-gallery algorithm for sensor placement, in: SCG '01: Proceedings of the seventeenth annual symposium on Computational geometry, ACM, New York, NY, USA, 2001, pp. 232–240.

[24] F. Li, R. Klette, An approximate algorithm for solving the watchman route problem, in: RobVis'08: Proceedings of the 2nd international conference on Robot vision, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 189–206.

[25] W.-P. Chin, S. Ntafos, Optimum watchman routes, in: SCG '86: Proceedings of the second annual symposium on Computational geometry, ACM Press, Yorktown Heights, New York, United States, 1986, pp. 24–33.

[26] S. Carlsson, B. J. Nilsson, S. C. Ntafos, Optimum Guard Covers and m-Watchmen Routes for Restricted Polygons, International Journal of Computational Geometry and Applications 3 (1) (1993) 85–105.

[27] B. J. Nilsson, Guarding Art Galleries: Methods for Mobile Guards, Ph.D. thesis, Lund University (1995).

[28] E. Packer, Robust geometric computing and optimal visibility coverage, Ph.D. thesis, Stony Brook University, New York (2008).

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

[30] W.-P. Chin, S. Ntafos, The zookeeper route problem, Informa-

tion Sciences: an International Journal 63 (3) (1992) 245–259.
[31] J. E. Goodman, J. O'Rourke (Eds.), Handbook of Discrete and Computational Geometry, CRC Press LLC, Boca Raton, FL, 2004.
[32] A. Dumitrescu, J. S. B. Mitchell, Approximation algorithms for tsp with neighborhoods in the plane, Journal of Algorithms 48 (1) (2003) 135 – 159.
[33] M. de Berg, J. Gudmundsson, M. J. Katz, C. Levcopoulos, M. H. Overmars, A. F. van der Stappen, Tsp with neighborhoods of varying size, Journal of Algorithms 57 (1) (2005) 22–36.
[34] S. Safra, O. Schwartz, On the complexity of approximating tsp with neighborhoods and related problems, Comput. Complex. 14 (4) (2006) 281–307.
[35] F. Li, R. Klette, Approximate algorithms for touring a sequence of polygons, MI-tech TR-24, The University of Auckland, Auckland, http://www.mi.auckland.ac.nz/tech-reports/MItech-TR-24.pdf (2008).
[36] F. Li, R. Klette, Approximate shortest paths in simple polyhedra, in: I. Debled-Rennesson, E. Domenjoud, B. Kerautret, P. Even (Eds.), Discrete Geometry for Computer Imagery, Vol. 6607 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2011, pp. 513–524.
[37] J. Faigl, On the performance of self-organizing maps for the non-euclidean traveling salesman problem in the polygonal domain, Information Sciences 181 (19) (2011) 4214–4229.
[38] A. Arcuri, L. Briand, A practical guide for using statistical tests to assess randomized algorithms in software engineering, in: Proceeding of the 33rd international conference on Software engineering, ICSE '11, ACM, New York, NY, USA, 2011, pp. 1–10.
[39] B. Yuan, M. Orlowska, S. Sadiq, Finding the optimal path in 3d spaces using edas – the wireless sensor networks scenario, in: ICANNGA '07: Proceedings of the 8th international conference on Adaptive and Natural Computing Algorithms, Part I, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 536–545.
[40] M. H. Overmars, E. Welzl, New methods for computing visibility graphs, in: SCG '88: Proceedings of the fourth annual symposium on Computational geometry, ACM, New York, NY, USA, 1988, pp. 164–171.
[41] D. Applegate, R. Bixby, V. Chvátal, W. Cook, CONCORDE TSP Solver, http://www.tsp.gatech.edu/concorde.html, [cited 8 Jul 2010] (2003).
[42] D. Applegate, W. Cook, A. Rohe, Chained lin-kernighan for large traveling salesman problems, INFORMS J. on Computing 15 (1) (2003) 82–92.
[43] R. Seidel, A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons, Comput. Geom. Theory Appl. 1 (1) (1991) 51–64.
[44] J. Faigl, Visiting convex regions in a polygonal map – problems, reference solutions, and supplemental data, http://purl.org/faigl/safari.