A Multi-Goal Path Planning for Goal Regions in the Polygonal Domain

Jan Faigl*, Vojtěch Vonásek, Libor Přeučil
*Center for Applied Cybernetics, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University in Prague
{xfaigl,vonasek,preucil}@labe.felk.cvut.cz

***Abstract*— This paper concerns a variant of the multi-goal path planning problem in which goals may be polygonal regions. 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, which is extended to the polygonal goals. Neurons' weights are considered as nodes inside the polygonal domain and connected nodes represent a path that evolves according to the proposed adaptation rules. Performance of the rules is evaluated in a set of problems including an instance of the watchman route problem with restricted visibility range. Regarding the experimental results the proposed algorithm provides flexible approach to solve various NP-hard routing problems in polygonal maps.**

I. Introduction

The multi-goal path planning problem (MTP) stands to find a shortest path connecting a given set of goals located in a robot working environment. The environment can be represented by the polygonal domain $\mathcal{W}$ and the goals may be sensing locations in the inspection task. Such point goals guaranteeing the whole environment would be covered using a sensor with limited sensing range can be found by a sensor placement algorithm [8]. The MTP with point goals can be formulated as the *Traveling Salesman Problem* (TSP) [16], e.g., using all shortest path between goals found in a visibility graph by Dijkstra's algorithm. Then, the MTP is a combinatorial optimization problem to find a sequence of goals' visits.

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" the requested object. 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* [12], which can be solved in $O(n^3)$ [17]. 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 [1]. However, both problems are NP-hard in general.

Routing problems with polygonal goals can be considered as variants of the TSP with neighborhoods (TSPN) [10]. 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 [5, 3]; however, the TSPN is APX-hard and cannot be approximated to within a factor $2 - \epsilon$, where $\epsilon > 0$, unless P=NP [13].

A combinatorial approach [14] 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 cannot be used for continuous sets because of too many possibilities how to connect the goals. This is also the case of the *watchman route problem* (WRP) in which goals are not explicitly prescribed. The WRP stands to find a closed shortest path such that all points of $\mathcal{W}$ are visible from at least one point [11]. Although polynomial algorithms have been proposed for restricted class of polygons [2], the WRP is NP-hard for the polygonal domain.

In this paper, a self-organizing map (SOM) algorithm for the TSP in $\mathcal{W}$ [9] is modified to deal with a general variant of the MTP. Contrary to combinatorial approaches, a geometrical interpretation of SOM evolution in $\mathcal{W}$ allows easy and straightforward extensions to deal with polygonal goals. To demonstrate geometric relation between the learning network and polygonal goals several modifications of the adaptation rules are proposed and evaluated in a set of problems. 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 $\mathcal{W}$.

The rest of this paper is organized as follows. The addressed problem formulation is presented in the next section. The proposed algorithms are based on the SOM adaptation schema for the TSP in $\mathcal{W}$, and therefore, a brief overview of the schema is presented in Section III. The proposed modifications of the adaptation rules for polygonal goals are presented in Section IV. Experimental evaluation of the proposed algorithm variants is presented in Section V. Concluding remarks are presented in Section VI.

II. 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 [12]; 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 $\mathcal{W}$ [6].

III. SOM Algorithm for the TSP 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 neurons' weights. So, the nodes form a ring in $\mathcal{W}$ because of the uni-dimensional structure of the output layer, see Fig. 1.

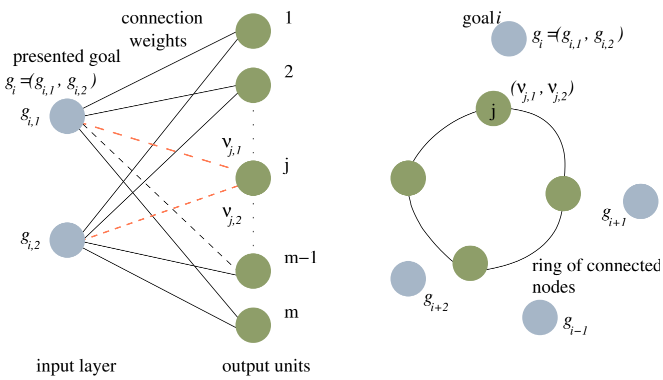


Fig. 1: A schema of the two-layered neural network and associated geometric representation.

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.
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:

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

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 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.

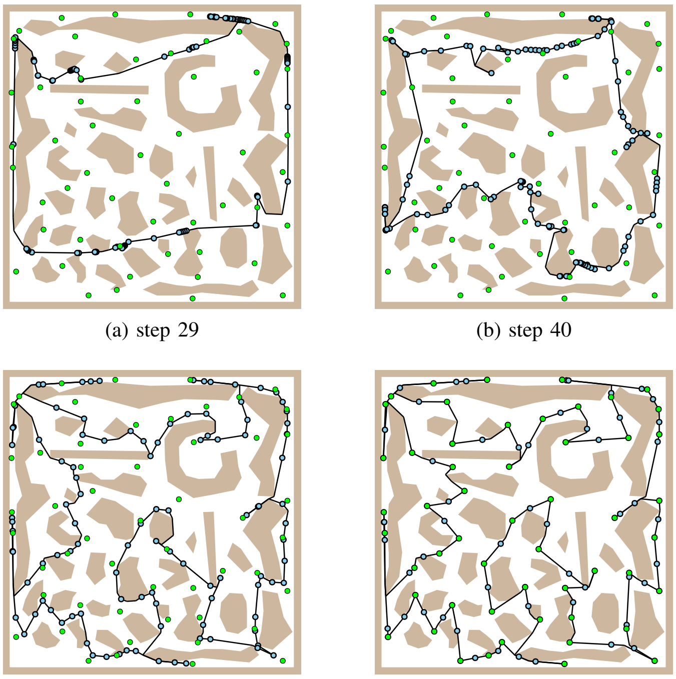
The computational burden of the adaptation procedure depends on determination of the shortest path in $\mathcal{W}$, because $2n^2$ node–goal distance queries (Step 4) and $(0.8n+1)n$ node–goal path queries (Step 5) have to be resolved in each adaptation step. Therefore, an approximate shortest path 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 [7], 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 $\mathcal{W}$.

IV. Adaptation Rules for Polygonal Goals

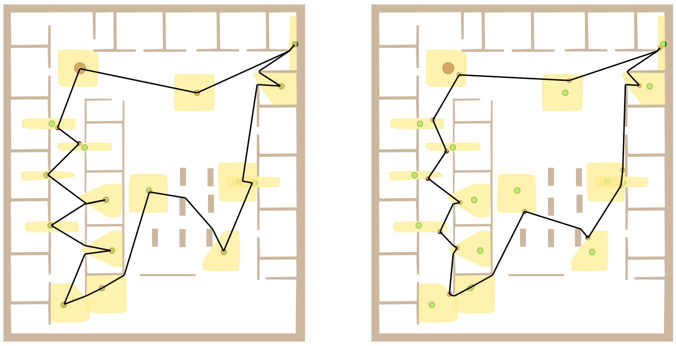
Although it is obvious that a polygonal goal can be sampled into a finite set of points and the problem can be solved as the MTP with partitioned goals, the aforementioned SOM procedure can be straightforwardly extended to sample the goals during the self-adaptation. Thus, instead of explicit sampling 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 the SOM for



(a) step 29 (b) step 40

(c) step 58 (d) step 78

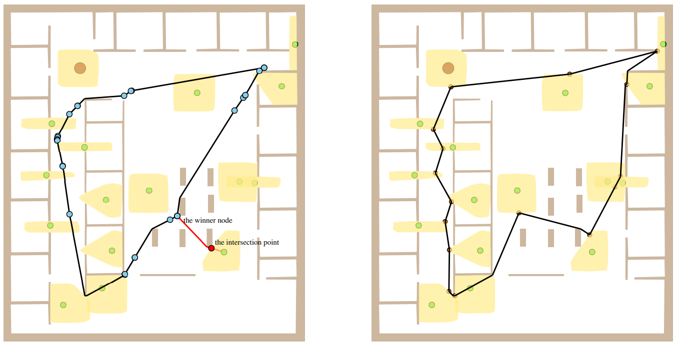
Fig. 2: An example of ring evolution in a polygonal map for the MTP with point goals, small green disks represent goals and blue disks are nodes.



(a) a found path for termination of the adaptation if all winners are inside the goals, L=84.3 m

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

Fig. 3: Examples of found paths without and with consideration of winners inside the goals. Goals are represented by yellow regions with small 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, and the length of the path connecting the centroids is L_{ref}=85.9 m.



(a) an intersection point

(b) a found path, L=59.7 m

Fig. 4: Examples of an intersection point and a found path using the *attraction* algorithm variant.

the TSP using centroids of the polygonal goals as point goals. 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 [6]).

A. 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 goals is shown in Fig. 3.

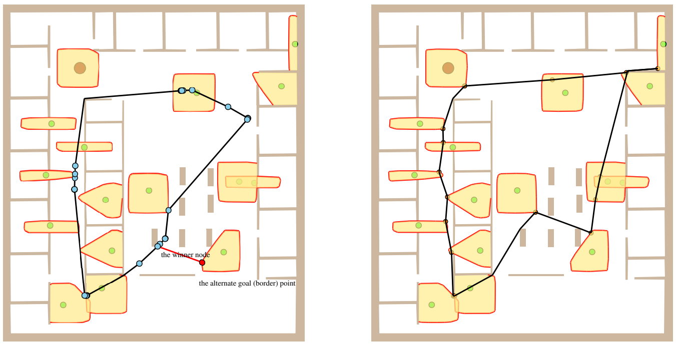
B. Attraction Point

The strategy described above can be extended by determination of a new attraction point at the border of the polygonal goal. 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 the point goal to adapt the winner and its neighbouring nodes. This modification is denoted as *attraction* in the rest of this paper.

An example of determined intersection point p and the final found path is shown in Fig. 4. The found 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, therefore an experimental evaluation of the proposed algorithm variants is presented in Section V.

C. 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 adaptation, therefore this modification is denoted as *alternate goal*.



(a) an alternate goal point (b) a found path, L=56.9 m

Fig. 5: An example of the alternate goal point and the final found path. Red straight line segments around the goal regions denote parts of the goal border inside the free space of $\mathcal{W}$.

Determination of the exact shortest point–segment path can be too computationally demanding, therefore the following approximation is proposed. 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 IV-B, it provides sampling of the goal boundary with a less distance of the goal point to the winner node; thus, a shorter final path can be found. An example of found alternate goal point and the found path is shown in Fig. 5.

V. Experimental Results

The proposed adaptation rules in Section IV 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 maps of real and artificial environments. An overview of the basic properties of the environments is shown in Table I.

TABLE I: Properties of environments and their polygonal representation

Map	Dimensions [m × m]	No. vertices	No. holes	No. convex polygons
jh	20.6 × 23.2	196	9	77
pb	133.3 × 104.8	89	3	41
h2	84.9 × 49.7	1 061	34	476
dense	21.0 × 21.5	288	32	150
potholes	20.0 × 20.0	153	23	75

The last column shows the number of convex polygons of the supporting convex polygon partition utilized in the approximation of the shortest path. The partition is found by Seidel's algorithm [15]. Maps *jh*, *pb*, and *h2* represent real environments (building plans), and maps *dense* and *potholes* are artificial environments with many obstacles.

Sets of polygonal goals 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 a particular problem variant, i.e., 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 are in mutual distance less than d. An unrestricted visibility range is considered in problems without the subscript.

Three proposed variants of the SOM based algorithm for the MTP with polygonal goals have been experimentally evaluated within the set of problems. The algorithms are randomized, therefore twenty solutions of each problem have been found by each algorithm variant. The average length of the path L, the minimal found path length L_{min}, and the standard deviation in percents of L (denoted as $s_L\%$) are used as the quality metrics. All presented length values are in meters. The experimental results are shown in Table II, where n is the number of goals. The best found solutions are shown in Fig. 6. From the visualized solutions, one can assume that high quality solutions are found for all problems.

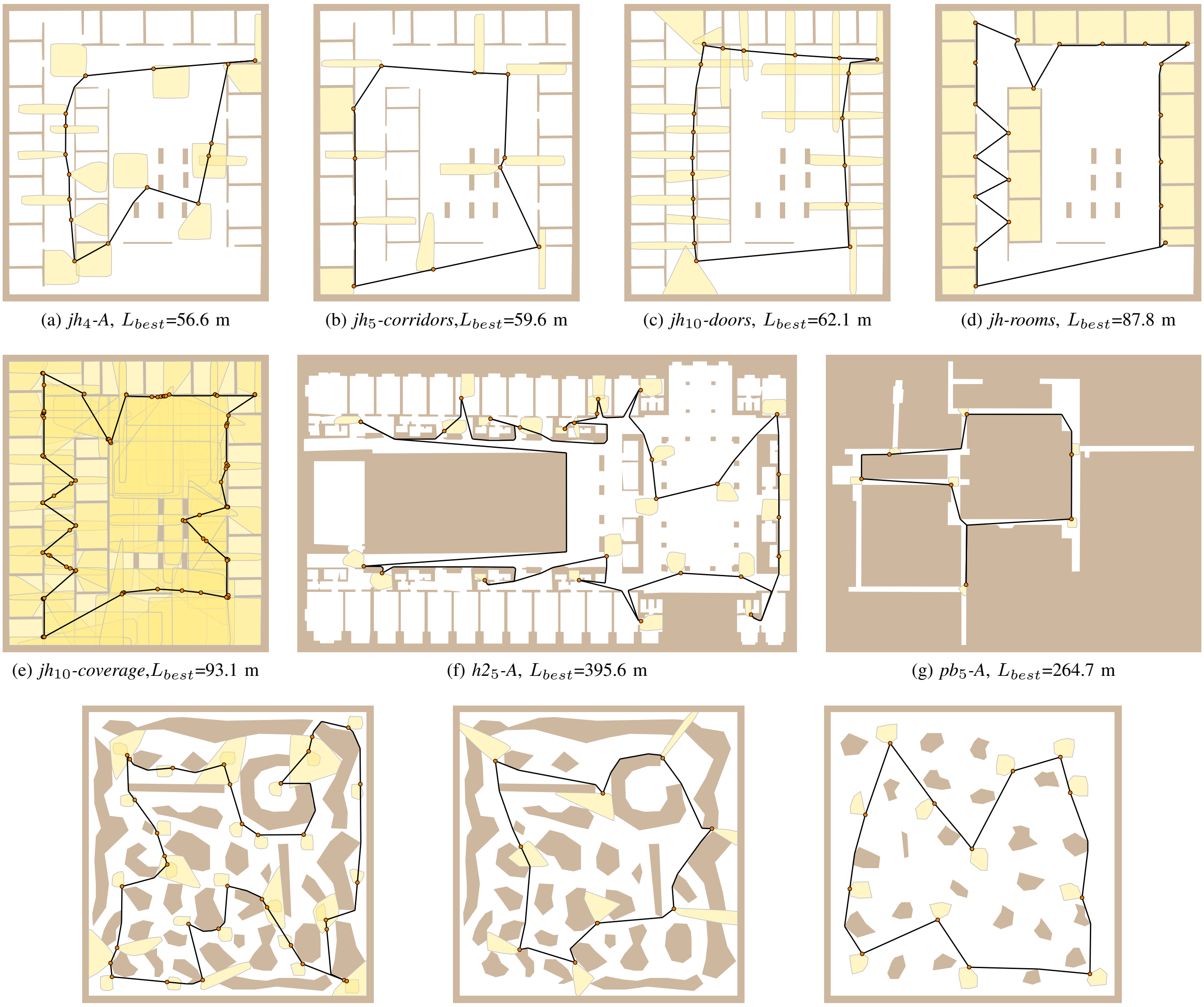
The required computational time is presented in the column T. All algorithms have been implemented in C++, compiled by the G++ version 4.2 with -O2 optimization, and executed within the same computational environment using a single core of 2 GHz CPU. Therefore, the presented required computational times can be directly compared. The time includes determination of all shortest path between map vertices (used in the path approximation) and the adaptation time. The supporting convex partition and the complete visibility graph are found in tens of milliseconds, and therefore, these times are negligible regarding the time of the adaptation procedure and determination of the shortest paths.

Discussion

The presented results provide 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 the *alternate goal* is sampling of goals' borders. Even though a simple approximation of the shortest path between

TABLE II: Experimental results

Problem	n	goal interior				attraction				alternate goal			
		L [m]	s_L%	L_{min}	T [s]	L [m]	s_L%	L_{min}	T [s]	L [m]	s_L%	L_{min}	T [s]
dense-small	35	119.5	3.21	113.51	0.85	114.4	3.47	108.64	0.96	111.8	3.09	106.11	1.20
dense$_5$-A	9	68.2	1.75	66.21	0.41	62.4	2.19	60.60	0.43	58.7	1.75	58.06	0.44
h2$_5$-A	26	425.3	1.42	416.22	3.56	405.9	1.25	399.62	3.54	402.1	0.97	396.46	3.75
jh-rooms	21	103.6	1.49	101.37	0.28	88.2	0.22	87.83	0.33	88.1	0.26	87.79	0.34
jh$_{10}$-doors	21	71.5	3.24	67.48	0.47	68.0	1.52	66.11	0.46	62.2	0.25	62.06	0.57
jh$_{10}$-coverage	106	134.2	2.52	128.31	3.64	106.6	1.59	101.19	4.19	93.8	0.47	93.12	6.22
jh$_4$-A	16	66.6	2.97	64.00	0.44	61.5	2.74	59.10	0.45	57.3	1.02	56.80	0.51
jh$_5$-corridors	11	69.6	2.05	66.96	0.38	66.0	1.23	64.83	0.39	60.0	0.56	59.60	0.41
pb$_5$-A	7	277.8	3.21	268.61	0.84	273.8	4.55	265.56	0.88	270.1	2.66	264.91	0.86
potholes$_2$-A	13	73.9	2.09	72.01	0.13	72.0	1.95	70.42	0.14	72.0	1.78	70.32	0.16



(a) *jh$_4$-A*, L_{best}=56.6 m (b) *jh$_5$-corridors*,L_{best}=59.6 m (c) *jh$_{10}$-doors*, L_{best}=62.1 m (d) *jh-rooms*, L_{best}=87.8 m

(e) *jh$_{10}$-coverage*,L_{best}=93.1 m (f) *h2$_5$-A*, L_{best}=395.6 m (g) *pb$_5$-A*, L_{best}=264.7 m

(h) *dense-small*, L_{best}=102.8 m (i) *dense$_5$-A*, L_{best}=58.0 m (j) *potholes$_2$-A*, L_{best}=68.7 m

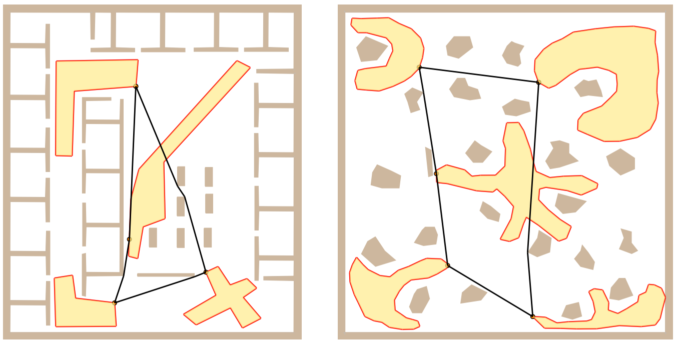
Fig. 6: The best found solutions.

a node (point) and goal's segment is used, a precision of the approximation increases with 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.

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.

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. 7).



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

Fig. 7: Found solutions of problems with non-convex goals by the *alternate goal* algorithm variant.

Regarding the problems with disjoint convex goals that are relatively far from each other, a sequence of goals' visits can be found as a solution of the TSP for centroids of the goals. Then, having the sequence of polygonal goals the problem of finding the path connecting them can be formulated as the *touring polygons problem* (TPP) if start and end points are given. A polynomial algorithm exists for the TPP with convex goals lying inside a simple polygon [4]; however, the TPP is NP-hard in general. Therefore, the proposed *alternate goal* algorithm seems be more practical due to its flexibility.

VI. 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 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. From the practical point of view, the proposed SOM algorithm is based on relatively simple algorithms and supporting structures, which is an additional benefit.

SOM is not a typical technique used for routing problems motivated by robotic 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.

Acknowledgment

This work has been supported by the Grant Agency of the Czech Technical University in Prague under grant No. SGS10/185, by the Ministry of Education of the Czech Republic under Projects No. 1M0567 and No. LH11053 and partially by Project No. 7E08006, and by the EU Project No. 216342.

References

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

[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] Mark de Berg, Joachim Gudmundsson, Matthew J. Katz, Christos Levcopoulos, Mark H. Overmars, and A. Frank van der Stappen. Tsp with neighborhoods of varying size. *Journal of Algorithms*, 57(1):22–36, 2005.

[4] Moshe Dror, Alon Efrat, Anna Lubiw, and Joseph S. B. Mitchell. Touring a sequence of polygons. In *STOC '03: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 473–482, San Diego, CA, USA, 2003. ACM Press.

[5] Adrian Dumitrescu and Joseph S. B. Mitchell. Approximation algorithms for tsp with neighborhoods in the plane. *Journal of Algorithms*, 48(1):135 – 159, 2003.

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

[7] Jan Faigl. On the performance of self-organizing maps for the non-euclidean traveling salesman problem in the polygonal domain. *Information Sciences*, 181(19):4214–4229, 2011.

[8] Jan Faigl, Miroslav Kulich, and Libor Přeučil. A sensor placement algorithm for a mobile robot inspection planning. *Journal of Intelligent & Robotic Systems*, 62(3-4):329–353, 2011.

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

[10] Jacob E. Goodman and Joseph O'Rourke, editors. *Handbook of Discrete and Computational Geometry*. CRC Press LLC, Boca Raton, FL, 2004.

[11] Fajie Li and Reinhard Klette. An approximate algorithm for solving the watchman route problem. In *RobVis'08: Proceedings of the 2nd international conference on Robot vision*, pages 189–206, Berlin, Heidelberg, 2008. Springer-Verlag.

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

[13] Shmuel Safra and Oded Schwartz. On the complexity of approximating tsp with neighborhoods and related problems. *Comput. Complex.*, 14(4):281–307, 2006.

[14] Mitul Saha, Tim Roughgarden, Jean-Claude Latombe, and Gildardo Sánchez-Ante. Planning Tours of Robotic Arms among Partitioned Goals. *Int. J. Rob. Res.*, 25(3):207–223, 2006.

[15] Raimund Seidel. A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons. *Comput. Geom. Theory Appl.*, 1(1):51–64, 1991.

[16] Steven N. Spitz and Aristides A. G. Requicha. Multiple-Goals Path Planning for Coordinate Measuring Machines. In *ICRA*, pages 2322–2327, 2000.

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