

Speed-up of Self-Organizing Networks for Routing Problems in a Polygonal Domain

Miroslav Kulich, Roman Sushkov, Libor Přeučil

Abstract—Routing problems are optimization problems that consider a set of goals in a graph to be visited by a vehicle (or a fleet of them) in an optimal way, while numerous constraints have to be satisfied. We present a solution based on multidimensional scaling which significantly reduces computational time of a self-organizing neural network solving a typical routing problem – the Travelling Salesman Problem (TSP) in a polygonal domain, i.e. in a space where obstacles are represented by polygons. The preliminary results show feasibility of the proposed approach and although the results are presented only for TSP, the method is general so it can be used also for other variants of routing problems.

I. INTRODUCTION

Mobile robots, as their autonomy increases, can be employed in more and more complex scenarios of e.g. inspection, reconnaissance or surveillance. A mission goal in these scenarios is not specified in the form 'go from A to B'. Instead, the task incorporates specification of particular sub-goals to be reached by a robot successively as well as determination of the order in which to reach the sub-goals. Finding an optimal solution leads frequently to some kind of combinatorial optimization problems like the TSP, the Vehicle Routing Problem or their variants called routing problems, which are NP-hard in general.

Although powerful heuristics for TSP providing near optimal solutions in a reasonable time exist, they have several drawbacks. These heuristics are not general so they cannot be used for other variants of problems, for multi-robot cases, and incorporation of additional constrains (e.g. assuming heterogeneous robots) is not straightforward or even possible. Therefore, other approaches like genetic algorithms, ant colony optimization, integer linear programming, self-organizing maps and others have an important role in situations where heuristics can not be applied.

A self-organizing map (SOM) as a popular approach to routing problems [1], [2], [3] is a competitive learning network trained using iterative unsupervised learning. Assuming the TSP as the problem of finding the shortest closed path between a defined set of goals in 2D, a standard variant of SOM is expressed as a ring of neurons where two neighboring neurons are connected and neurons are represented by their coordinates in a plane. The learning procedure takes

The authors are with Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, Zikova 1903/4, 166 36 Prague, Czech Republic e-mail: kulich@cvut.cz, sushkrom@fel.cvut.cz, preucil@cvut.cz

This work has been supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 688117 and the Technology Agency of the Czech Republic under the project no. TE01020197 "Centre for Applied Cybernetics".

one input goal in each iteration and finds the nearest neuron to it. Positions of all neurons are then adapted based on their ring distance to the winning neuron (i.e. the number of neurons between the neuron and the winning neuron), iteration number, and possibly other criteria. The process finishes when some stopping condition (e.g., a number of iterations) is satisfied.

This approach works fast in an empty space, where goal-neuron distance needed to determine the nearest neuron is determined as the Euclidean distance. The situation is more complicated in the real world when obstacles are present, as a shortest collision-free path between a goal and a neuron has to be computed, which is a computationally intensive task.

We presented several approaches to fast determination of a neuron-goal path in our previous work considering a polygonal representation of obstacles which are based on shortest path maps [4] or convex polygon partitioning [1]. The idea presented in this paper is different: a polygonal domain representing the working environment with obstacles is transformed into a higher-dimensional space without obstacles so that distances in this space are (ideally) the same as those in the input space.

II. ALGORITHM

Assume a polygonal workspace $\mathcal{W} \subset \mathbb{R}^2$ and a set of goals \mathcal{G} . The task is to find a shortest path visiting all goals from \mathcal{G} while respecting obstacles of \mathcal{W} . The proposed Transformation Based Algorithm (TBA) works in the following steps:

- 1) Compute a distance matrix for all vertices of obstacles in \mathcal{W} . To do this, a visibility graph of the vertices is constructed on which Johnson's algorithm to find the shortest paths between all pairs of vertices is run.
- 2) Use multidimensional scaling to transform the vertices into \mathbb{S}^m , a high-dimensional space without obstacles, where their mutual distances are preserved.
- 3) Compute constrained Delaunay triangulation (CDT) of the original polygonal vertices.
- 4) Approximate coordinates of the goals. This is done in several steps:
 - a) Determine a triangle of CDT, the goal lies in.
 - b) Find a linear combination of vertices of the triangle which represents the goal in \mathcal{W} .
 - c) Apply the same linear combination to determine a position of the goal in \mathbb{S}^m .
- 5) Run SOM in \mathbb{S}^m (neuron-goal distances are computed as Euclidean distances in \mathbb{S}^m).

Problem	n	L_{opt}	Transformation based					Shortest Paths (va-10)		
			PDM	PDB	$T(s)$	$T_{MDS}(s)$	$T_{SOM}(s)$	PDM	PDB	$T(s)$
<i>jari</i> *	6	13.6	4.41	1.47	0.01	0.0	0.0	0.21	0.0	0.08
<i>complex2</i>	8	58.5	0.0	0.0	0.03	0.02	0.0	0.29	0.0	0.1
<i>m1</i> *	13	17.1	0.0	0.0	0.01	0.0	0.0	0.14	0.0	0.16
<i>m2</i>	14	19.4	9.79	9.28	0.03	0.02	0.01	12.51	8.95	0.17
<i>map</i> *	17	26.5	5.66	5.28	0.02	0.0	0.01	3.85	0.0	0.25
<i>potholes</i>	17	88.5	1.58	0.0	0.37	0.28	0.01	1.48	0.0	0.33
<i>a</i> *	22	52.7	0.19	0.0	0.03	0.0	0.01	0.28	0.0	0.4
<i>rooms</i>	22	165.9	6.21	4.04	0.09	0.06	0.02	1.01	0.17	0.44
<i>dense</i> ₄ *	53	179.1	14.35	7.98	0.18	0.03	0.07	14.14	6.86	2.71
<i>potholes</i> ₂	68	154.5	5.05	3.24	0.56	0.28	0.15	5.06	3.09	3.54
<i>jh</i> ₂	80	201.9	5.45	1.98	0.68	0.35	0.21	2.04	0.43	4.96
<i>pb</i> ₄	104	654.6	3.67	1.56	0.44	0.04	0.35	1.1	0.03	6.88
<i>ta</i> ₂ *	141	328.0	17.01	13.26	0.86	0.37	0.44	2.96	2.13	11.94
<i>h2</i> ₅ *	168	943.0	12.26	9.05	2.71	1.04	0.61	1.7	1.16	67.68
<i>potholes</i> ₁	282	277.3	8.26	6.02	3.44	0.26	2.44	6.54	3.75	64.6
<i>pb</i> _{1.5}	415	839.6	8.83	7.36	5.62	0.04	5.17	2.17	1.07	123.92
<i>h2</i> ₂ *	568	1316.2	28.66	24.94	24.81	13.47	7.23	2.38	1.75	686.24
<i>ta</i> ₁ *	574	541.1	18.85	14.64	15.42	6.98	7.33	5.46	4.77	225.98

TABLE I

COMPARISON OF TBA AND THE APPROACH IN [1] THE QUALITY OF SOLUTIONS IS EVALUATED AS THE PERCENT DEVIATION TO THE OPTIMUM TOUR LENGTH OF THE MEAN SOLUTION VALUE, $PDM = (L - L_{opt})/L_{opt} \cdot 100\%$, AND AS THE PERCENT DEVIATION FROM THE OPTIMUM OF THE BEST SOLUTION VALUE (PDB), WHERE L_{opt} IS THE LENGTH OF THE OPTIMAL SOLUTION FOUND BY THE CONCORDE SOLVER, SEE [1]. DIMENSION OF THE TRANSFORMED SPACE IS $m = 5$ OR $m = 10$ (THOSE ARE MARKED BY AN ASTERISK).

- 6) Use the solution (ordering of the goals) found \mathbb{S}^m as the results of the problem in the original space.

The key component of the algorithm is multidimensional scaling (MDS) [5] – a method that maps a set of samples in some space into some other space based on their similarity which is usually specified by a positive symmetric distance matrix. MDS is traditionally used for visualization of multidimensional data in two or three-dimensional space (or, mapping multidimensional data into low-dimensional space). In the produced output, similar objects are mapped close to each other forming clusters. Contrary to the traditional use, the objects (vertices of obstacles) are mapped into a high-dimensional space in our case as our goal is to preserve the distances between the objects while abolishing the obstacles and not to visualise their similarity.

III. PRELIMINARY RESULTS

To demonstrate feasibility of the proposed approach, experiments were performed with Scaling by Maximizing a Convex Function [5] employed for MDS and ORC-SOM [6] as a variant of SOM and results were compared with those in [1], which is one of the best SOM-based approaches. The preliminary results presented in Table I show that TBA is by one or two orders faster, while quality of generated solutions is slightly worse. Note also that MDS-based transformation does not depend on goals as it is computed from vertices of obstacles only. This implies that it can be precomputed once a map is available and T_{SOM} is thus relevant if several TSP problems are to be solved for a given map.

The current and future research is focused on study of properties of MDS and a polygonal domain in order to find better approximations of the polygonal domain and thus produce high-quality results.

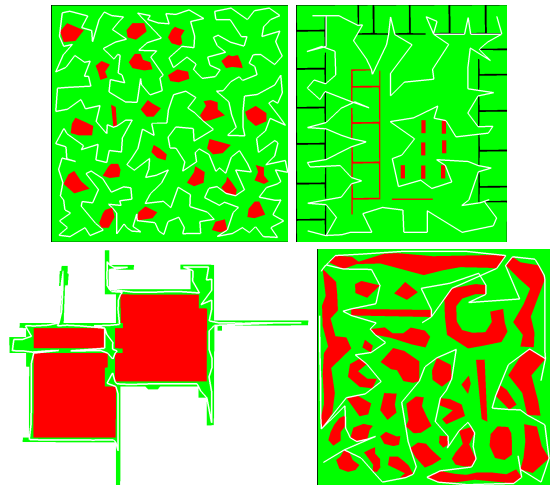


Fig. 1. Examples of solutions of some problems (goals are placed in all turning points of the polylines).

REFERENCES

- [1] J. Faigl, M. Kulich, V. Vonásek, and L. Přeučil, “An application of the self-organizing map in the non-Euclidean Traveling Salesman Problem,” *Neurocomputing*, vol. 74, no. 5, pp. 671–679, 2011.
- [2] J. Faigl and G. A. Hollinger, *Self-Organizing Map for the Prize-Collecting Traveling Salesman Problem*. Cham: Springer International Publishing, 2014, pp. 281–291.
- [3] E. Cochrane and J. Beasley, “The co-adaptive neural network approach to the euclidean travelling salesman problem,” *Neural Networks*, vol. 16, no. 10, pp. 1499 – 1525, 2003.
- [4] M. Kulich, J. Faigl, and L. Přeučil, “Cooperative planning for heterogeneous teams in rescue operations,” in *IEEE International Safety, Security and Rescue Robotics, Workshop, 2005.*, June 2005, pp. 230–235.
- [5] I. Borg and P. J. Groenen, *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.
- [6] J. Zhang, X. Feng, B. Zhou, and D. Ren, “An overall-regional competitive self-organizing map neural network for the euclidean traveling salesman problem,” *Neurocomputing*, vol. 89, pp. 1–11, 2012.