On Self-Organizing Map and Rapidly-Exploring Random Graph in Multi-Goal Planning

Jan Faigl

Abstract This paper reports on ongoing work towards an extension of the self-organizing maps for the traveling salesman problem to more challenging problems of multi-goal trajectory planning for complex robots with a high-dimensional configuration space. The main challenge of this problem is that the distance function needed to find a sequence of the visits to the goals is not known a priori and it is not easy to compute. To address this challenge, we propose to utilize the unsupervised learning in a trade-off between the exploration of the distance function and exploitation of its current model. The proposed approach is based on steering the sampling process in a randomized sampling-based motion planning technique to create a suitable motion planning roadmap, which represents the required distance function. The presented results shows the proposed approach quickly provides an admissible solution, which may be further improved by additional samples of the configuration space.

1 Introduction

Self-Organizing Map (SOM) is a type of neural network that can provide a non-linear mapping of a high dimensional input space into a lower dimensional output space. In addition to data processing, visualization, and classification, it has also been successfully applied in optimization routing problems, in particular, the Traveling Salesman Problem (TSP). The TSP is a well-defined optimization problem arising from many practical scenarios and several SOM-based approaches have been proposed, e.g., see [2, 14]. In our case, the TSP is a problem formulation for robotic tasks like inspection, surveillance, and data collection where a mobile robot is requested to visit a set of locations, e.g., to perform an operation or take a sensor measurement [3, 4, 8, 11].

J. Faigl (✉)
Czech Technical University in Prague, Department of Computer Science,
Technická 2, 166 27 Prague, Czech Republic
e-mail: faiglj@fel.cvut.cz

E. Merényi et al. (eds.), *Advances in Self-Organizing Maps and Learning Vector Quantization*, Advances in Intelligent Systems and Computing 428,
DOI 10.1007/978-3-319-28518-4_12

The most straightforward application of SOM to the TSP is in Euclidean instances, where the problem stands to find a closed shortest tour connecting a given set of goal locations (cities). In robotics, the problem is to find a shortest path connecting the locations such that the path is collision free. This make an application of SOM to the TSP a bit more challenging because a pure Euclidean distance cannot be simply used in the computation of distances between neuron weights and the presented goal location (signal) to the network; otherwise a poor solution would be found [5]. The distance corresponds to the length of the shortest path between two locations, which can be PSPACE-hard in 3D environment. Hence, the problem is called the *Multi-Goal Motion Planning* (MGMP) problem rather than the TSP to emphasize difficulty of distance queries.

Randomized sampling-based approaches are motion planning techniques for planning in high-dimensional configuration space $\mathcal{C}$ that provide the so-called motion planning roadmap, which is a graph representing collision free configurations in $\mathcal{C}$ [9]. A combination of the roadmap with SOM for a graph input [13] has been proposed in [6] to solve the MGMP by SOM. In this decoupled approach, the roadmap (graph) is constructed independently on the planning problem, and therefore, a complete graph is unnecessarily dense.

In this paper, we report our recent results on application of SOM in the roadmap generation and solution of the MGMP problem. The main idea of the proposed approach is based on combining principles of the optimal motion planning algorithm called *Rapidly-exploring Random Graph* (RRG) [7] with the SOM adaptation principles to simultaneously determine the sequence of the goal visits together with trajectories connecting the goals in the tour. The core of the proposed approach is a utilization of the SOM adaptation to steer a randomized sampling of $\mathcal{C}$ to increase the number of samples in the most promising areas to quickly find a solution and eventually improve quality of the final trajectory.

A feasibility of this idea has been reported in [12], where it has been employed in finding multi-goal trajectories for a hexapod walking robot. The proposed SOM-based algorithm needs a lower number of the roadmap expansions to find a first feasible solution of the MGMP problem in comparison to a straightforward MGMP solver based on a given sequence of visits to the goal locations.

Here, we focus on two main aspects of the proposed approach: (1) a detailed evaluation of the idea of SOM-based expansion of the roadmap to find an initial solution of the MGMP; and (2) improving the quality of the final solution with increasing number of the roadmap expansions. Based on the evaluation, we propose a hybrid approach that consists of the initial construction of the roadmap by SOM to find the first feasible solution followed by a consecutive roadmap improvement to find a shorter trajectory.

The paper is organized in the following way. The problem statement, notion of the configuration space $\mathcal{C}$, and related background is presented in the next section. The key idea of the SOM-based steering of the roadmap expansions using the RRG is briefly described in Sect. 3. Considered MGMP solvers are presented in Sect. 4 and results of their evaluation are in Sect. 5. Concluding remarks and future work are summarized in Sect. 6.

2 Problem Statement

The problem addressed by the proposed approach is motivated by autonomous data collection with a hexapod walking robot operating in a rough environment to collect samples, e.g., images, of the requested areas of interest, see Fig. 1. The robot has six legs, each with three joints that gives 18 control degrees of freedom, which together with the robot position and orientation in the 3D environment gives 24 dimensional vector fully describing the position of the robot body in the environment. Therefore, it is controlled by designed gait patterns and a set of motion primitives to simplify the motion control and planning [12]. In addition and without loss of generality, the robot pose (x, y, θ) is considered as the robot position on a surface x, y with orientation θ.

The working environment $\mathcal{W} \subset \mathbb{R}^3$ is represented as a set of obstacles $\mathcal{O} \subset \mathcal{W}$. The configuration space $\mathcal{C}$ describes all possible configurations of the robot in $\mathcal{W}$ and can be defined as follows. Let the robot body at q be $\mathcal{A}(q)$, then the configuration q is a collision free if $\mathcal{A}(q) \cap \mathcal{O} = \emptyset$. All configurations for which the robot is in a collision with the obstacles $\mathcal{O}$ are denoted as $\mathcal{C}_{obst}$, $\mathcal{C}_{obst} \subseteq \mathcal{C}$. The point of our interest to find a solution of the MGMP is a collision free part of $\mathcal{C}$, which can be denoted as $\mathcal{C}_{free} = \mathrm{cl}(\mathcal{C} \setminus \mathcal{C}_{obst})$, where cl(.) is the set closure.

A collision free path from some starting configuration q_{start} to a goal configuration q_{goal} is a continuous curve κ in $\mathcal{C}_{free}$, such that $\kappa : [0, 1] \rightarrow \mathcal{C}_{free}$ with $\kappa(0) = q_{start}$ and $d(\kappa(1), q_{end}) < \epsilon$. The end point $\kappa(1)$ of the path found by a motion planner will unlikely be exactly the requested goal location, and therefore, we rather admit an admissible distance ϵ of the path to the requested goal [7], e.g., 5 cm. Then, such a collision free path is called an admissible path.

Similarly to a simple trajectory, a multi-goal trajectory visiting a set of n goal locations $\mathcal{G} = (g_1, \ldots, g_n)$ can be defined as follows. Let the sequence of the visits to the locations be $(v_1, v_2, \ldots, v_n)$ for which $v_i \in \mathcal{G}$ and $\bigcup_{1<i\leq n} v_i = \mathcal{G}$. Then, an admissible multi-goal trajectory is a closed trajectory $\tau : [0, 1] \rightarrow \mathcal{C}_{free}$ such that $\tau(0) = \tau(1) = q_{start}$ and for which there exists n points on τ such that $0 \leq t_1 \leq t_2 \leq \cdots \leq t_n$ and $d(\tau(t_i), v_i) < \epsilon$.

Having the aforementioned preliminaries, the MGMP problem can be formulated as follows: *For the given goal locations* $\mathcal{G}$*, configuration space* $\mathcal{C}$*, an admissible distance* ϵ*, and a monotonic, bounded, and strictly positive cost*

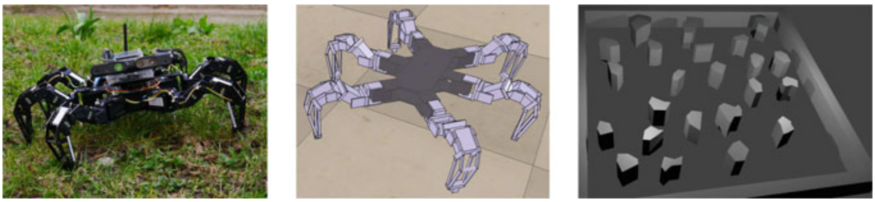

Fig. 1 Robot, its geometrical model, and visualized 3D environment

function c: find an admissible (according to ϵ *) trajectory* τ^* *such that* $c(\tau^*) = \min\{c(\tau) \mid \tau$ *is admissible multi-goal trajectory*$\}$.

2.1 Randomized Sampling-Based Motion Planners

Sampling based motion planning techniques have been proposed to address difficulty of explicit representation of $\mathcal{C}_{free}$ for a complex shape of the robot body and its high-dimensional $\mathcal{C}$ [9]. These techniques sample $\mathcal{C}_{free}$ into a finite number of configurations that are connected into a graph, where an edge represents a collision free trajectory between two configurations. Hence, $\mathcal{C}_{free}$ is represented by a graph and the key problem is how to efficiently create the graph (roadmap) in which the requested trajectory can be found, e.g., by a graph search technique.

In this work, we consider RRG [7] to create a graph $\mathbf{G}_{RRG} = (\mathbf{V}_{RRG}, \mathbf{E}_{RRG})$, which represents the motion planning roadmap. The set of vertices $\mathbf{V}_{RRG}$ are particular configurations of the robot $q \in \mathcal{C}_{free}$ and an edge $e \in \mathbf{E}_{RRG}$ describes a feasible collision free motion between two configurations $v_i, v_j \in \mathbf{V}_{RRG}, i \neq j$. The graph is incrementally constructed by the RRG algorithm as a result of the graph expansion from the nearest vertex of the graph towards a random sample by applying a particular control command. The main steps of the RRG expansion are depicted in Fig. 2, further details can be found in [7].

2.2 Basic Background of Self-Organizing Map for the TSP

The proposed MGMP solvers are based on SOM for the TSP, in particular, a variant for a graph input [13]. The neural network is structured in two layers. The first layer servers for presenting goal locations to be visited and towards which the network is

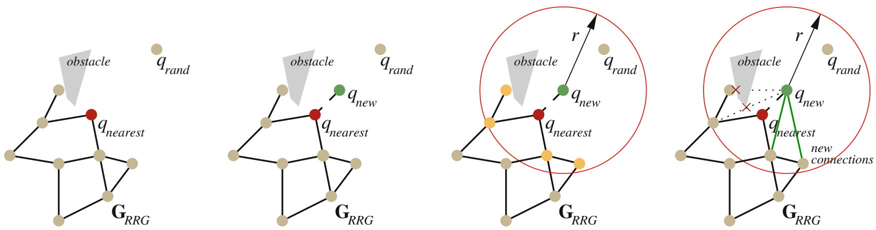

Fig. 2 An expansion of the RRG roadmap (from *left* to *right*): First, a random (collision free) configuration q_{rand} is sampled and the nearest vertex $q_{nearest} \in \mathbf{V}_{RRG}$ is determined; Then, the most suited control command is applied to expand the roadmap towards q_{rand} by a collision free trajectory and a new configuration q_{new} is added to the roadmap; To further improve the roadmap, all vertices within a ball with a particular radius r (see [7]) centered on q_{new} are connected with q_{new} by a collision free trajectory

adapted using the self-organizing principles. The output layer consists of m units, $\mathcal{N} = \{\nu_1, \ldots, \nu_m\}$, which represent neurons weights, where m is set according to the number of goal locations n, e.g., $m = 2.5n$. The units are organized into one-dimensional array that represents a sequence of configurations in $\mathcal{C}_{free}$. The learning procedure can be summarized as follows:

1. *Initialization*—Create a ring of connected neurons $\mathcal{N} = \{\nu_1, \ldots, \nu_m\}$.
2. *Randomization*—Create a random permutation of goals $\Pi(\mathcal{G}) \leftarrow \text{permute}(\mathcal{G})$.
3. *Winner selection*—Select the best matching neuron ν^* to the currently presented goal $g \in \Pi(\mathcal{G})$; $\nu^* \leftarrow \text{argmin}_{\nu \in \mathcal{N}} d(\nu, g)$.
4. *Adaptation*—Adapt the winner ν^* and its neighbouring nodes ν_j within the distance k (in the number of nodes) using the neighbouring function $f(\sigma, k) = \mu e^{(-k^2/\sigma^2)}$ for $k < 0.2m$ and $f(\sigma, 0) = 0$ otherwise. Remove g from the permutation, $\Pi(\mathcal{G}) \leftarrow \Pi(\mathcal{G}) \setminus \{g\}$, and `If` $|\Pi(\mathcal{G})| > 0$ go to Step 3.
5. *Update* the number of the learning epochs and neighbouring function variance.
6. *Termination condition*—`If` termination condition is met, stop the adaptation. `Otherwise` go to Step 2.
7. *Final tour construction:*—Traverse the output layer and use the associated goals to the last winners to construct the final goal tour.

The adaptation of neurons can be imagined as a movement of the neurons towards the presented goal location. For a graph input, the neurons weights are restricted to be at the graph edges or vertices and the adaptation can be imagined as neurons movements along the graph edges [13]. Thus, for an adaptation in the roadmap $\mathbf{G}_{RRG}$ with spatially close vertices (such that provided by the RRG), we can consider the neuron weights as a particular configuration represented by the closest vertex from V_{RRG}.

Notice, even though we can use SOM to find a solution of the MGMP on $\mathbf{G}_{RRG}$ like in [6]; here, we are rather interested in employing the adaptation procedure to grow and improve the roadmap $\mathbf{G}_{RRG}$ by the RRG expansions.

3 SOM-based Steering of Randomized Sampling in RRG

The fundamental issue of applying SOM to the given problem is that the selection of the winner node to a presented location g is based on computing a distance $d(\nu, g)$ between nodes $\nu \in \mathcal{N}$ and g. Such a distance corresponds to the length of the trajectory from ν to g, which is obviously not known due to a sparse coverage of $\mathcal{C}$ by $\mathbf{G}_{RRG}$, especially at the beginning of the learning. In [12], we propose to address this issue by the approximation that combines Euclidean distance and the current knowledge about $\mathcal{C}_{free}$ stored in the incrementally built $\mathbf{G}_{RRG}$.

Regarding a collision free and feasible trajectory in $\mathcal{W}$, the current roadmap $\mathbf{G}_{RRG}$ provides a much more realistic estimation of the expected distance $d(\nu, g)$ than a pure Euclidean distance. Therefore, a part of $d(\nu, g)$ is based on a trajectory in $\mathbf{G}_{RRG}$ from ν towards the vertex $w_{\nu,g}$ that is found as

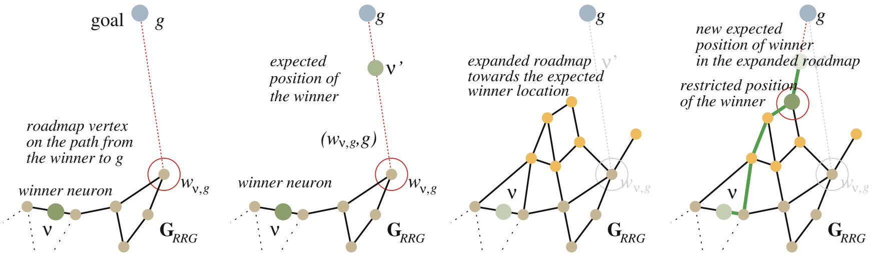

Fig. 3 SOM adaptation with $\mathbf{G}_{RRG}$ expansions, from *left* to *right*: First, vertex $w_{\nu,g}$ is found in $\mathbf{R}_{RRG}$ using (1) for the current winner ν (green disc); The expected position ν' of the neuron after the adaptation is determined; which is then utilized together with g in the RRG expansion of $\mathbf{G}_{RRG}$; Finally, new $w_{\nu,g}$ is found and ν is updated to the nearest vertex to the expected position of ν

$$w_{\nu,g} = \text{argmin}_{v \in \mathbf{V}_{RRG}} (c(\kappa_{\nu,v}) + |(v, g)|^2), \tag{1}$$

where $c(\kappa_{\nu,v})$ is the trajectory cost from the intermediate vertex w determined in $\mathbf{G}_{RRG}$ and $|(v, g)|$ is the Euclidean distance from v to g. Thus, the path from ν to g consists of the trajectory $\kappa_{\nu,w_{\nu,g}}$ in $\mathbf{G}_{RRG}$ and a straight line segment from $w_{\nu,g}$ to g. Notice, the cost found in the roadmap should be preferred and the influence of the Euclidean distance should be suppressed, that is why it is in power of two in (1). The found path is utilized in adaptation of neurons to g.

However, the path over the vertex $w_{\nu,g}$ cannot be directly used for a new position of the adapted neuron because the expected position of the neuron may be out of the current roadmap $\mathbf{G}_{RRG}$. Therefore, the expected position of the neuron after the adaptation is determined and the roadmap is expanded towards it and the location g using the RRG expansion accompanied by the goal bias and goal zooming techniques [10] (in which a random sample is substituted by the given location and sampled around the location, respectively). Then, the vertex $w_{\nu,g}$ is determined again in the updated roadmap and a new expected position of the neuron being adapted is restricted to the nearest vertex of $\mathbf{G}_{RRG}$. Hence, the approximation together with the proposed adaptation of neurons turns out to a steering strategy to randomized sampling in the RRG. The process is schematically visualized in Fig. 3.

4 Solvers for the Multi-Goal Motion Planning Problem

The proposed approach to solve the MGMP problem consists of two steps. First, a roadmap $\mathbf{G}_{RRG}$ is created. An admissible solution of the MGMP problem is found if all locations $g \in \mathcal{G}$ have its corresponding (nearest) configuration $v_g \in \mathbf{G}_{RRG}$ in less than ϵ distance from the particular g and there exists a trajectory in $\mathbf{G}_{RRG}$ that connects all the locations $\mathcal{G}$. The final shortest multi-goal trajectory is found in $\mathbf{G}_{RRG}$ as a solution of the TSP using Chained Lin-Kernighan heuristic [1].

An admissible trajectory can be found in $\mathbf{G}_{RRG}$ if all vertices representing the goal locations are connected. The quality of the final trajectory depends on the roadmap and basically a denser roadmap may provide shorter trajectories at the cost of more demanding computations. The key to efficiently find a good trajectory is in the construction of the roadmap. Various methods how to steer the expansion of $\mathbf{G}_{RRG}$ can be proposed. The SOM-based steering of the RRG has been firstly introduced in [12]. The idea has been further investigated and the improved method is presented here. Moreover, we considered the proposed idea utilized in SOM steering also in a direct construction of the roadmap to verify the added value of the unsupervised learning. The proposed roadmap construction methods are briefly summarized in the following paragraphs.

Naive construction of the roadmap is based on iterative roadmap expansions towards the locations $\mathcal{G}$ that are alternating in a sequence found as a solution of the Euclidean TSP. Each location is iteratively used in the goal zooming technique for 5 expansions and the process is repeated until the maximum number of expansions M is not reached. The ball expansions of the RRG are activated after 100 alternations of the whole sequence, to reduce the computational burden and improve convergence of the roadmap to an admissible solution.

SOM expansion is based on the steering strategy described in Sect. 3 that is accompanied by additional expansions towards the presented location $g \in \mathcal{G}$ to the network, which support a fast convergence of the roadmap to $\mathcal{G}$. If g is not yet connected with the roadmap, 20 expansions towards g are performed using g in goal zooming prior adaptation of the winner neuron towards g. After that, the proposed SOM steering is employed. Similarly to *Naive* method, the ball expansions of the RRG are suppressed for the first 10 learning epochs.

Rand variant of the roadmap construction is based on additional expansions to $\mathcal{G}$ used in the SOM method. It is similar to the *Naive* method, but the sequence of locations $\mathcal{G}$ is a random permutation as in SOM. Each location $g \in \mathcal{G}$ is used in goal zooming for 20 expansions. Then, the algorithm continues with the next location in the sequence. Once all locations are used, a new permutation of $\mathcal{G}$ is created and the process is repeated up to M roadmap expansions are performed.

MST method represents an existing approach for the MGMP [11] based on an iterative determination of the Minimum Spanning Tree (MST) as approximation of the TSP. The MST is initially determined using Euclidean distances that is iteratively refined using an "optimal" motion planner to find corresponding trajectories for all MST edges until all the edges represent admissible trajectories. An optimal motion planning is too computationaly demanding for the hexapod robot, and therefore, the MST is used to steer roadmap expansion. For each MST edge without a corresponding trajectory in the roadmap, 20 expansions towards the edge's endpoints are performed for every iteration of the MST refinement. This is repeated until an admissible multi-goal trajectory is found.

Because the SOM method provides a first admissible solution very quickly, two hybrid approaches are proposed: **Naive-SOM** and **Rand-SOM**. The SOM method is utilized to find the first admissible solution. Then, the *Naive* and *Rand* approaches are used up to M expansions of the roadmap, respectively. In a similar way, the MST is utilized in the **Rand-MST** to further improve initial solution provided by the MST-based method.

5 Evaluation Results

The roadmap expansion strategies have been evaluated for a hexapod walking robot and several scenarios of the MGMP problem in the environment called *potholes*, see Fig. 1. A particular difficulty of the problem depends on spatial distribution of the goal locations in the environment. Therefore, 20 random problem instances are created in the given environment. Each instance is solved 20 times by each particular algorithm because all algorithms are stochastic, and the results are presented as average values accompanied by standard deviations. We considered problems with 10 goal locations ($n = 10$) as sufficient to demonstrate difficulty of constructing roadmap for the multi-goal trajectory planning. Particular algorithms have been evaluated for different parameters; however, only selected results are presented because of the space limit. The total number of the evaluated scenarios was more than twenty thousands. Examples of constructed roadmaps, the first admissible solution found by SOM, and the final found solution found by the Rand variant are shown in Fig. 4.

The most time consuming step in the solution of the MGMP problem is a single roadmap expansion, which, in the case of the RRG, is a more computationally demanding with increasing number of roadmap vertices. Moreover, it is even more demanding in the improving phase, where expansions are performed for vertices in the ball around the last added vertex to the roadmap. Therefore, the number of performed roadmap expansions is the main performance indicator.

The first evaluation is focused on the performance of the roadmap expansion strategies in finding the first admissible solution with the maximal number of expansions restricted to 100000. The results for 400 trials on 20 problems solved by each approach are depicted in Table 1 (values are computed from admissible solutions).

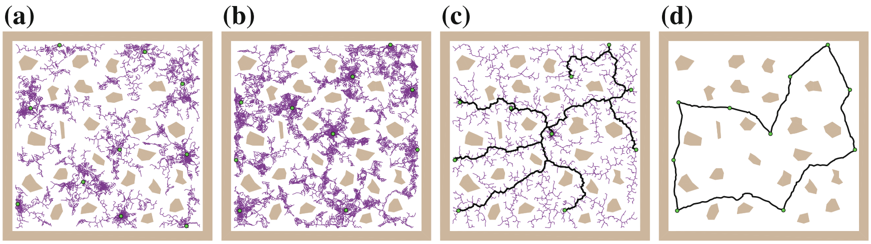

Fig. 4 Build roadmaps by *Naive* and *SOM-based* approaches after performing M expansions. A path found by the *Rand* approach after 204 357 expansions. Obstacles are in brown, goals are represented as green discs, roadmap edges are purple segments, and a multi-goal trajectory is in black. **a** Naive, M = 10000. **b** Naive, M = 20000. **c** SOM, M = 597. **d** Rand, M = 204357

Table 1 Roadmap construction for determining a first admissible solution

Method	Naive	SOM	Rand	MST
Success rate	54 %	**93** %	61 %	45 %
Average number of the RRG expansions	85 468	**14 258**	66 375	70 241
Average number of the roadmap vertices	24 698	**5 662**	25 781	37 815
Average number of the roadmap edges	142 472	**16 218**	109 096	84 372
Average required CPU time [s]*	39	**13**	56	209

*Indicative values because several machines of different configurations have been used

Here, we can observe that the same randomized schema utilized in SOM and Rand strategies provide different performance. Moreover, the MST-based approach proposed in [11] does not provide significant advantage over Rand and its more demanding because of determination of the MST. The results support the evidence that the proposed SOM-based steering significantly improves the performance in finding the first admissible solution. The main results is that SOM provides fastest admissible solutions with a high success rate.

Notice, the number of the roadmap vertices is always lower than the number of expansions. A higher number of vertices indicates a successful expansion of the roadmap and similarly a higher number of edges indicates a denser roadmap as a result of the improving step of the RRG.

The next evaluation has been focused on the quality improvement of the found multi-goal trajectory according to increasing maximal number of the performed RRG expansions. We found out that the proposed SOM improves solutions only slowly with more expansions, and therefore, we consider it only in finding the first admissible solution in the hybrid approaches *Naive-SOM* and *Rand-SOM*. The quality of the trajectory is considered as a ratio of the trajectory length to the best found solution for the particular problem determined from all the performed trials. This allows to aggregate results for various problem instances, for which trajectories may be significantly different. Thus, values of the ratio close to 1 indicate the particular approach provides relatively high quality solutions among the evaluated algorithms. The results for increasing number of roadmap expansions are depicted in Fig. 5.

Discussion—Based on the performed evaluation of the steering strategies of the randomized sampling in the RRG, the results support that the proposed SOM-based strategy provides the first admissible solution with a significantly less number of expansions than other strategies. However, the solution quality does not improve with more expansions and thus the current form of the strategy is suitable only for finding an admissible solution. On the other hand, the proposed combination of the SOM and randomized expansions in the hybrid solvers provide benefits of the both approaches and it seems to be a suitable technique to provide the first solution quickly and further quality improvements.

An important lesson learned from the presented evaluation is that the way how the roadmap is initially created significantly affects the ability to find an admissible solution quickly. Here, the SOM adaptation provides an efficient trade-off between

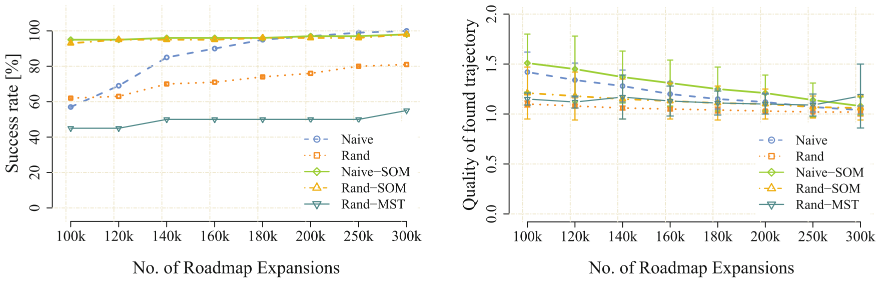

Fig. 5 Success rate and quality of the found trajectories

exploration of $\mathcal{C}$ and exploitation of the current $\mathbf{G}_{RRG}$ towards connecting the required goal locations. However, once the locations are connected in the roadmap, the adaptation process only moves neurons along the roadmap and does not explore possible shortcuts to improve the solution.

6 Conclusion

An evaluation of four multi-goal trajectory planners is presented in this paper. The results indicate the proposed SOM-based roadmap expansion improves finding the first admissible solution. However, a planner solely based on the SOM strategy does not improve the found solution, but the solution can be improved by additional expansions of the roadmap. Although the current achieved results does not meet the expectation of a motion planner solely based on SOM, it support feasibility of the SOM-based simultaneous building of the distance function approximation together with its utilization in the multi-goal trajectory planning.

Regarding the applied SOM based principles, the whole graph $\mathbf{G}_{RRG}$ can be considered as a growing neural network, where the adaptation rules can be used to remove not promising configurations and thus reduce the number of vertices of the graph. Besides, they can also be utilized to further exploration of the configuration space to improve quality of the found solution. Consideration of these extensions is a subject of our further work.

Acknowledgments The presented work is supported by the Czech Science Foundation (GAČR) under research project No. 13-18316P. Computational resources were provided by the MetaCentrum under the program LM2010005 and the CERIT-SC under the program Centre CERIT Scientific Cloud, part of the Operational Program Research and Development for Innovations, Reg. No. CZ.1.05/3.2.00/08.0144.

References

1. Applegate, D., Cook, W., Rohe, A.: Chained Lin-Kernighan for large traveling salesman problems. Informs J. Comput **15**(1), 82–92 (2003)
2. Créput, J.C., Koukam, A.: A memetic neural network for the Euclidean traveling salesman problem. Neurocomputing **72**(4–6), 1250–1264 (2009)
3. Elinas, P.: Multi-goal planning for an autonomous blasthole drill. In: Gerevini, A., Howe, A.E., Cesta, A., Refanidis, I. (eds.) ICAPS. AAAI (2009)
4. Faigl, J., Hollinger, G.: Unifying multi-goal path planning for autonomous data collection. In: IROS, pp. 2937–2942 (2014)
5. Faigl, J., Kulich, M., Vonásek, V., Přeučil, L.: An application of self-organizing map in the non-euclidean traveling salesman problem. Neurocomputing **74**(5), 671–679 (2011)
6. Janoušek, P., Faigl, J.: Speeding up coverage queries in 3d multi-goal path planning. In: ICRA, pp. 5067–5072 (2013)
7. Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning. Int. J. Rob. Res. **30**(7), 846–894 (2011)
8. Lattanzi, L., Cristalli, C.: An efficient motion planning algorithm for robot multi-goal tasks. In: ISIE, pp. 1–6. IEEE (2013)
9. Lavalle, S.M.: Planning Algorithms. Cambridge University Press (2006)
10. Lavalle, S.M., Kuffner, J.J.: Rapidly-exploring random trees: progress and prospects. Algorithmic and Computational Robotics: New Directions, pp. 293–308 (2001)
11. Saha, M., Roughgarden, T., Latombe, J.C., Sánchez-Ante, G.: Planning tours of robotic arms among partitioned goals. Int. J. Rob. Res. **25**(3), 207–223 (2006)
12. Vanek, P., Faigl, J., Masri, D.: Multi-goal trajectory planning with motion primitives for hexapod walking robot. In: ICINCO, pp. 599–604 (2014)
13. Yamakawa, T., Horio, K., Hoshino, M.: Self-organizing map with input data represented as graph. In: Neural Information Processing, pp. 907–914 (2006)
14. Zhang, J., Feng, X., Zhou, B., Ren, D.: An overall-regional competitive self-organizing map neural network for the euclidean traveling salesman problem. Neurocomputing **89**, 1–11 (2012)