Path Planning On Hierarchical Bundles With Differential Evolution

Victor Parque, Tomoyuki Miyashita

Waseda University,
3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555, Japan
parque@aoni.waseda.jp

Abstract. Computing hierarchical routing networks in polygonal maps is significant to realize the efficient coordination of agents, robots and systems in general; and the fact of considering obstacles in the map, makes the computation of efficient networks a relevant need for cluttered environments. In this paper, we present an approach to compute the minimal-length hierarchical topologies in polygonal maps by Differential Evolution and Route Bundling Concepts. Our computational experiments in scenarios considering convex and non-convex configuration of polygonal maps show the feasibility of the proposed approach.

Keywords: route bundling, hierarchical network design, minimal trees

1 Introduction

Over the last decade research on Internet of Things and collaborative robots has made clear that optimal and robust routing in networks are significant to realize the effective coordination and communication of multi-agent systems; and the fact of having obstacles over the map, makes the computation of collision-free routing a relevant need in cluttered environments [1–5].

Research in route planning has its origins in the mid 60's, and since the seminal work of Lozano-Perez in 1979 [6], the problem has been extensively studied in the literature. For recent reviews, see [7, 8]. Often, collision-free trajectories are computed considering the optimality of navigation in the free space. And well-known methods such as RRT [9, 10] and PRM [11] guarantee probabilistic completeness, while RRT* guarantees asymptotic optimality. Also, approaches based on sampling and optimization with gradient-based approaches are used, such as CHOMP, STOMP, and TrajOpt. However, these methods are sensitive to initial conditions (initial trajectory). Also, path planning based on geometric information has been argued to be accurate [12–16], in which finding optimal origin-destination is usually based on the triangulation of the free space. Also, online and approximation approaches have been proposed as well, e. g. the Potential Field method [17], and the Cell Decomposition method [18]. Furthermore, heuristic approaches have been used to achieve optimality of routes in the global sense, and examples include nature inspired approaches such as Neural

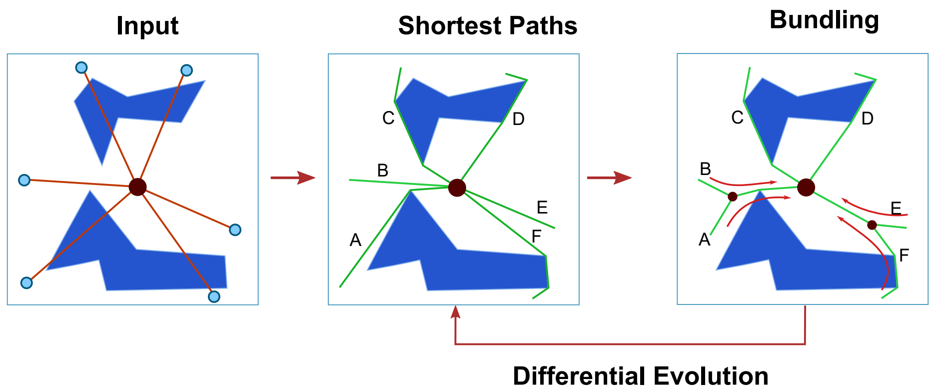


Fig. 1. Basic concept of the proposed approach

Networks [19, 20], Genetic Algorithms [21], Differential Evolution [22–26] and Particle Swarm Optimization (PSO) [27, 28].

Being related to the *Steiner tree* problem, path planning on hierarchical bundles is key to allow efficient distribution and communication of sparsely distributed nodes. Having started in the 30's [29], the *Steiner tree* problem was popularized in the 40's [30]. In practical settings, the Obstacle-Avoiding Rectilinear Steiner (OARST) given n nodes in a polygonal map has received recent attention in VLSI systems [1–5]; and there exists a polynomial-time approximation of the more general Obstacle-Avoiding Steiner Tree (OAST) with $O(nlog^2n)$ (n is number of terminals and obstacle vertices) [31]. However, the exhaustive study of global optimization and gradient-free approaches on Minimum Steiner Trees in polygonal maps has received little attention.

Other related works to path planning on hierarchical bundles involve the edge bundling in network visualization [32–36], the path bundling in bipartite networks [25, 26, 37], and the minimal trees in n-star networks with fixed roots [38]. However, path planning considering minimal Steiner trees and flexible root configuration has received little examination in the literature.

In this paper, in order to fill the above gaps, we propose an approach to compute minimal trees given n points with a star topology and flexible configuration of the root, wherein the goal is to generate topologically compact and minimal trees being free of clutter and easy to visualize. The basic idea of our approach, depicted by Fig. 1, is to allow routes to be bundled by using a hierarchical configuration, and optimize the location of the root by Differential Evolution. Our results by using a diverse set of polygonal map configurations show the feasibility to compute minimal trees in the plane.

In the rest of this article, after describing the key components in our proposed approach, we discuss our findings though our computational experiments, and finally summarize our insights and future work.

2 Path Planning on Hierarchical Bundles

The basic outline of our algorithm is depicted by Fig. 1, in the following we briefly describe the key components and dynamics.

2.1 Preliminaries

The *input* in our algorithm is the set of terminal nodes V and a polygonal map P; and the *output* is a tree layout aiming at minimizing the total tree length, while not only preserving connectivity from the root r towards the nodes in the terminal set V, but also avoiding the obstacles in map P.

2.2 Shortest Paths

The route r is known a-priori and its location is an interior point of the convex hull of V. The shortest paths are computed from the root r towards each node in the terminal V by using the A* algorithm with visibility graphs, rendering the set ρ of *shortest routes* from the source r to each terminal node in V.

2.3 Route Bundling

Then, shortest routes are clustered by the hierarchical agglomerative approach with complete and Euclidean metric, which renders a *dendogram* $\mathbf{Z} = [z_{ij}] \in \mathbb{R}^{|\rho| \times 2}$ denoting the ordering of path bundling in which the rows of the matrix $\mathbf{Z}$ are configured in ascending order, with *similar* (*different*) routes being located first (last). For clustering and similarity computation, the distance between two routes is computed by the following metric:

$$d(\rho_i, \rho_j) = \left(\sum_{k=1}^{SP} ||\rho_i^k - \rho_j^k||^2 \right) . \left(cos^{-1} \Big(\frac{\mathbf{a_i}.\mathbf{a_j}}{|\mathbf{a_i}||\mathbf{a_j}|} \Big) \right) \tag{1}$$

, where $\rho_i \in \rho$, ρ_i^k is the k-th sampled point along the route ρ_i, SP is the number of equally-separated interpolated points along the route ρ_i, and $\mathbf{a_i} = \rho_i^{end} - \rho_i^{init}$ in which $\rho_i^{init}, \rho_i^{end} \in \mathbb{R}^2$ are the *starting* and the *end* coordinates of the route ρ_i, respectively. The main rationale of using the above distance metric is due to its key benefit of measuring not only *piecewise gaps* (due to difference in topology), but also *orientation gaps* (due to arbitrariness of location of end nodes). Furthermore, note that the above distance metric is able to be computed under parallelization schemes, bringing benefits in scalability for large-scale path planning applications.

By using the order of the *dendogram* (hierarchical clustering), routes are bundled by a nature-inspired approach which considers the *merging*, the *expansion* and the *shrinkage* of leaves [38]. The bundle process is executed in *bottom-up* approach (from terminal nodes to root), followed by a *top-down* approach (from root to terminal nodes), which ensures co-adaptation while searching for optimal topologies.

– *Merging* occurs when the anchoring node x is far from the root r, yet close to either routes u or v. Farness of node x to r, and closeness of x to u, v is computed by $\ell(x, r) > \delta_1$ and $D(r, u, v) < \delta_2$, respectively, where:

$$D(r, u, v) = min(\ell(r, u), \ell(r, v)) \tag{2}$$

, where $\ell(x, w)$ is the length of the *shortest route* from node x to node w along the polygonal map P. The role of using the user-defined thresholds δ_1 and δ_2 is to allow flexibility and granularity when designing and generating minimal trees: smaller (larger) values of $\delta_1(\delta_2)$ creates more (less) intermediate nodes x, thus the global tree length is expected to be small (large). Then the *farthest* leaf (u or v) is merged to the *closest* one (u or v), in which the *closest* leaf is computed by the following metric:

$$closest = \begin{cases} u, & \text{for } \ell(x, u) < \ell(x, v) \\ v, & \text{for } \ell(x, u) > \ell(x, v) \end{cases} \tag{3}$$

As a natural consequence, the *farthest* node is the opposite of the above.
– *Expansion* occurs when the anchoring node x is *far* from the root r, and *far* from leaves u, v.
– *Shrinkage* occurs either when the tree has a single leaf, or when the anchoring node x is *close* to the root r and *close* to either u or v.

The above bundling operations are guided by the *dendogram* **Z**; in which some of the edges of the T are compounded and some intermediate nodes are inserted due to the *expand* operation. Note that tree operations are performed recursively.

2.4 Optimizing the Root of Trees

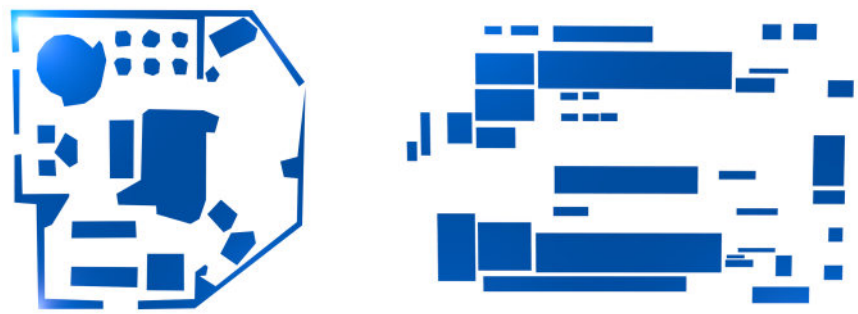
The root in the tree T is allowed to be flexible, and its location is optimized by minimizing:

$$J(T, r) = \sum_{s \in leaves(T)} \ell(r, s) \tag{4}$$

, where r is the root of tree T. Note that the above definition is *recursive*. Due to the nature of handling obstacles with arbitrary geometry, the optimization of the above cost function is realized by Differential Evolution with Neighborhood and Convex Encoding [25, 26], which is used due to its advantages to not only balance the *exploration* and the *exploitation* while searching for the optimal location of the root, but also to render feasible root coordinates by using a triangular encoding, which allows to sample obstacle-avoiding coordinates in polygonal maps efficiently.

3 Computational Experiments

In order to evaluate the performance of our proposed path bundling algorithm, we performed computational experiments in diverse scenarios.



(a) Map 1 (b) Map 2

Fig. 2. Polygonal Maps

3.1 Settings

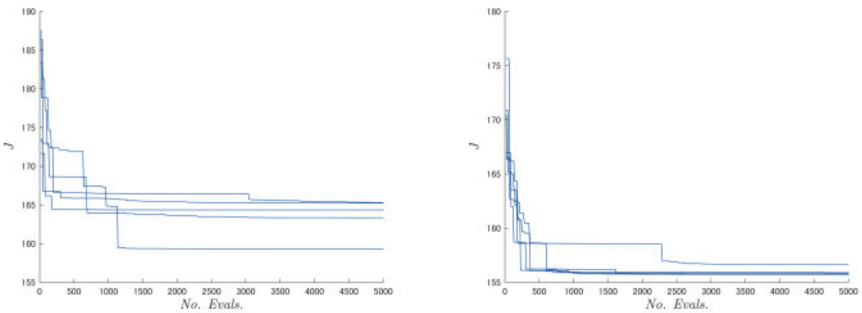
Our computing environment was an Intel i7-4930K @ 3.4GHz, Matlab 2016a. To evaluate our approach in diverse scenarios, we used polygonal maps with convex and non-convex polygonal configurations, as shown by Fig. 2 [39, 40]. Also, for each configuration, 5 independent runs for path planning in origin-destination pairs consisting of 20 edges in a star-topology was performed. The main motivation of using the above is due to our foci on scenarios being close to indoor environments, where complexity is controlled by the convexity and the configuration of the polygonal map.

As for parameters in Differential Evolution, we used: probability of crossover $CR = 0.5$, scaling factor $\alpha = \beta = |ln(U(0, 1))/2|$, population size with 10 individuals, neighborhood ratio $\eta = 0.2$, and termination criterion is 5000 function evaluations. The main reason of using crossover probability $CR = 0.5$ is to give the same importance to the sampling with historical search vectors, and with local and global interpolations. The scaling factors α, β allow to search in small steps when computing the self-adaptive directions. Furthermore, small values of population size and neighborhood factor enable efficient sampling within the local neighborhood [25]. Fine tuning of the above is out of our scope.

3.2 Results

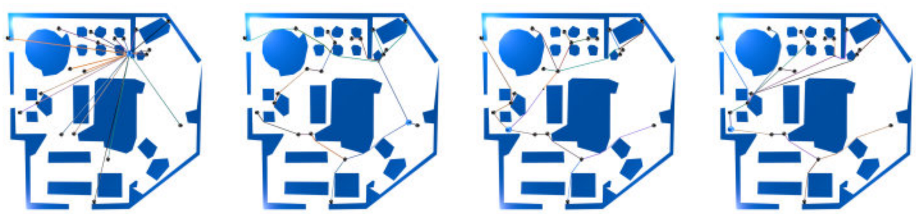
In order to show the learning performance of our proposed approach, Fig. 3 shows the convergence behaviour as a function of the number of evaluations over independent runs. By observing Fig. 3, we can confirm that computing minimal trees becomes possible within a 1000 - 1500 function evaluations.

Also, in order to show the evolvability performance of our proposed approach, Fig. 4 and Fig. 5 show the elite solutions generating the path planners in hierarchical bundles after E function evaluations. Here $E = 0$ denotes the input (as

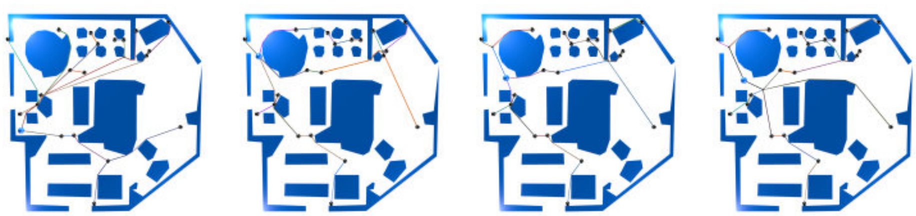


(a) Convergence in Map 1 (b) Convergence in Map 2

Fig. 3. Convergence over different Maps



(a) $E = 0$ (b) $E = 20$ (c) $E = 40$ (d) $E = 290$



(e) $E = 420$ (f) $E = 900$ (g) $E = 1180$ (h) $E = 5000$

Fig. 4. Convergence examples in Map 1 after E function evaluations.

portrayed by the basic concept in Fig. 1), and $E = 5000$, denotes the converged solution. By looking at the generated topologies in Fig. 4 and Fig. 5, we can observe that it is possible to compute the optimal location of the roots (which is different from the initial solution), and that larger changes in topology occur at earlier stages of the learning algorithm. We believe this fact occurs due to the highly explorative (exploitative) nature in earlier (later) generations, which induces in large (small) changes in the nature of the topology of minimal trees.

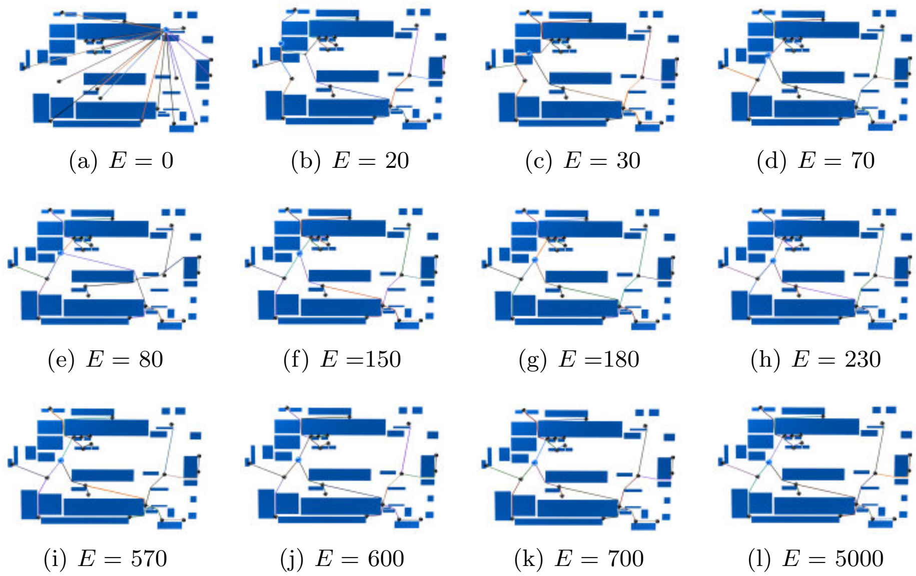


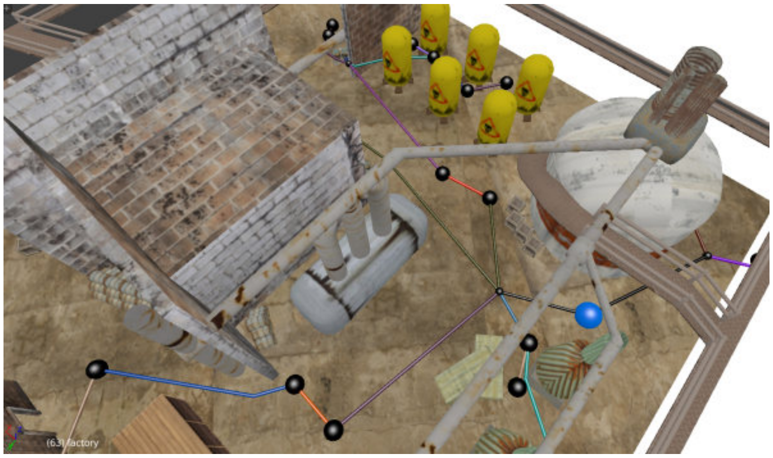
Fig. 5. Convergence examples in Map 2 after E function evaluations.

Finally, to visualize deployment, Fig. 6 shows the generated topologies of the minimal trees in their respective environments [39, 40].

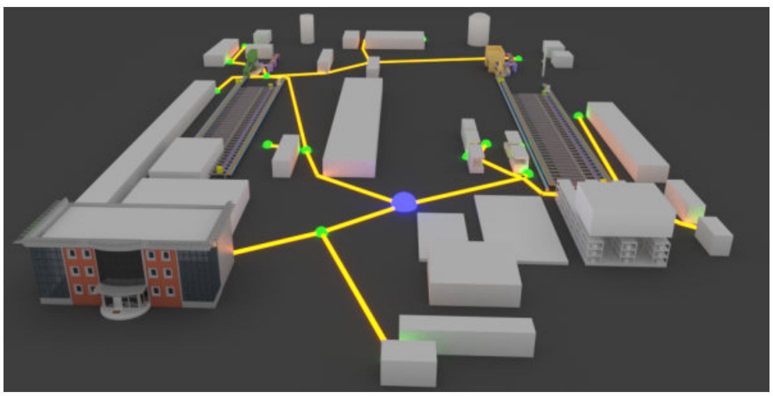
We believe that our obtained results are building blocks to further advance path planning in hierarchical networks in the presence of polygonal obstacles. Investigating the learning performance with canonical encodings in directed graphs [41] and undirected graphs [42], the use of concurrency concepts in networks [44] and in exploration-exploitation [43], as well as the formation of modules in hierarchical bundles by succinct subset partitions [45] are in our agenda.

4 Conclusion

We proposed a method to compute hierarchical networks in polygonal maps given n points configured in an n-star topology with flexible root location. The basic idea of our approach is based on path bundling to find minimal trees while avoiding obstacles, and evolution to compute the optimal location of the roots in the minimal tree. Our computational experiments involving convex and non-convex polygonal map scenarios confirm the feasibility to compute obstacle-avoiding minimal trees, and the efficiency to converge to optimal solutions within 1000-1500 function evaluations. In our future work, we aim at exploring the learning performance of minimal topologies by using succinct encodings of graphs, concurrency and combinatorial subset formation. We believe our approach may find uses in Operations Research, Communications and Multi-Agent Systems.



(a) Minimal Tree in Map 1



(b) Minimal Tree in Map 2

Fig. 6. Minimal Trees.

References

1. Zhang, H., Ye, D., Guo, W.: A heuristic for constructing a rectilinear Steiner tree by reusing routing resources over obstacles. INTEGRATION, the VLSI journal, Vol. 55, pp. 162-175 (2016)
2. Winter, P., Zachariasen, M., Nielsen, J.: Short trees in Polygons. Discrete Applied Mathematics, Vol. 118, pp. 55-72 (2002)
3. Jing, T.T., Hu, Y., Feng, Z., Hong, X., Hu, X., Yan, G.: A full-scale solution to the rectilinear obstacle-avoiding Steiner problem. INTEGRATION, the VLSI journal, Vol. 41, pp. 413-425 (2008)
4. Winter, P.: Euclidean Steiner minimal trees with obstacles and Steiner visibility graphs. Discrete Applied Mathematics, Vol. 47, pp. 187-206 (1993)

5. Chow, W., Li, L., Young, E., Sham, C.: Obstacle-avoiding rectilinear Steiner tree construction in sequential and parallel approach. INTEGRATION, the VLSI journal, Vol. 47, pp. 105-114 (2014)
6. Lozano-Pérez, T., Wesley, M.A.: An algorithm for planning collision-free paths among polyhedral obstacles. Commun. ACM **22**(10) (October 1979) 560–570
7. Souissi, O., Benatitallah, R., Duvivier, D., Artiba, A., Belanger, N., Feyzeau, P.: Path planning: A 2013 survey. In: Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM). (Oct 2013) 1–8
8. Mohanan, M., Salgoankar, A.: A survey of robotic motion planning in dynamic environments. Robotics and Autonomous Systems **100** (2018) 171 – 185
9. LaValle, S.M.: Rapidly-exploring random trees: A new tool for path planning. Technical Report. Computer Science Department, Iowa State University (TR 98-11)
10. LaValle, S.M., Kuffner, J.J., Jr.: Rapidly-exploring random trees: Progress and prospects (2000)
11. Kavraki, L.E., Svestka, P., Latombe, J.C., Overmars, M.H.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation **12**(4) (Aug 1996) 566–580
12. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik,1:269-271 (1959)
13. P.E. Hart, N.J. Nilsson, B.R.: A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics, 4(2):100-107 (1968)
14. Chazelle, B.: A Theorem on Polygon Cutting with Applications. In Proc. 23rd IEEE Symposium on Foundations of Computer Science, pp. 339-349 (1982)
15. Cormen, T., Leiserson, C., Rivest, R.: Introduction to Algorithms. MIT Press, Cambridge, MA (1993)
16. Lee, D.T., Preparata, F.P.: Euclidean Shortest Paths in the Presence of rectilinear barriers. Networks. 14(3), pp. 393-410 (1984)
17. Chiang, H.T., Malone, N., Lesser, K., Oishi, M., Tapia, L.: Path-guided artificial potential fields with stochastic reachable sets for motion planning in highly dynamic environments. (May 2015) 2347–2354
18. Ghita, N., Kloetzer, M.: Trajectory planning for a car-like robot by environment abstraction. Robotics and Autonomous Systems **60**(4) (2012) 609 – 619
19. S., B.: An integrated learning approach to environment modelling in mobile robot navigation. Neurocomputing **57** (2004) 215 – 238 New Aspects in Neurocomputing: 10th European Symposium on Artificial Neural Networks 2002.
20. Duan, H., Huang, L.: Imperialist competitive algorithm optimized artificial neural networks for ucav global path planning. Neurocomputing **125** (2014) 166 – 171 Advances in Neural Network Research and Applications Advances in Bio-Inspired Computing: Techniques and Applications.
21. Davoodi, M., Panahi, F., Mohades, A., Hashemi, S.N.: Clear and smooth path planning. Applied Soft Computing **32** (2015) 568 – 579
22. Wang, M., Luo, J., Fang, J., Yuan, J.: Optimal trajectory planning of free-floating space manipulator using differential evolution algorithm. Advances in Space Research **61**(6) (2018) 1525 – 1536
23. Zhang, X., Chen, J., Xin, B., Fang, H.: Online path planning for uav using an improved differential evolution algorithm. IFAC Proceedings Volumes **44**(1) (2011) 6349 – 6354 18th IFAC World Congress.

24. Parque, V., Miura, S., Miyashita, T.: Computing Path Bundles in Bipartite Networks. 7th Int. Conf. on Simulation and Modelling Methodologies, Technologies and Applications, pp. 422-427, Madrid, Spain (2017)
25. Parque, V., Miura, S., Miyashita, T.: Route bundling in polygonal domains using differential evolution. Robotics and Biomimetics **4**(1) (Dec 2017) 22
26. Parque, V., Miura, S., Miyashita, T.: Optimization of route bundling via differential evolution with a convex representation. In: 2017 IEEE Int. Conf. on Real-time Computing and Robotics (RCAR). (July 2017) 727–732
27. Zhang, Y., Gong, D.W., Zhang, J.H.: Robot path planning in uncertain environment using multi-objective particle swarm optimization. Neurocomputing **103** (2013) 172 – 185
28. Mac, T.T., Copot, C., Tran, D.T., Keyser, R.D.: A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization. Applied Soft Computing **59** (2017) 68 – 76
29. Vojtěch, J., Kössler, M.: O minimálních grafech, obsahujících n daných bodů. Časopis pro pěstování matematiky a fysiky 063.8, pp. 223-235 (1934)
30. Robbins, H., Courant, R.: What is Mathematics? Oxford University Press (1941)
31. Müller-Hannemann, M., , Tazari, S.: A near linear time approximation scheme for Steiner tree among obstacles in the plane. Computational Geometry: Theory and Applications, Vol. 43, pp. 395-409 (2010)
32. Holten, D., van Wijk, J.J.: Force-Directed Edge Bundling for Graph Visualization. Eurographics, Symp. on Visualization (2009)
33. Selassie, D., Heller, B., Heer, J.: Divided Edge Bundling for Directional Network Data. IEEE Transactions on Visualization and Computer Graphics, Vol. 17, No. 12, pp. 2354 - 2363 (2011)
34. Cui, W., Zhou, H., H. Qu, P.C.W., Li, X.: Geometry-based edge clustering for graph visualization. IEEE Transactions on Visualization and Computer Graphics, Vol. 14, pp. 1277 - 1284 (2008)
35. Gansner, E.R., Y. Hu, S.N., Scheidegger, C.: Multilevel agglomerative edge bundling for visualizing large graphs. IEEE Pacific Visualization Symposium, pp. 187 - 194 (2011)
36. Holten, D.: Hirerarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. IEEE Pacific Visualization Symposium, pp. 187 - 194 (2006)
37. Parque, V., Kobayashi, M., Higashi, M.: Optimisation of Bundled Routes. 16th Int. Conf. on Geometry and Graphics, pp. 893-902 (2014)
38. Parque, V., Miyashita, T.: Bundling n-Stars in Polygonal Maps. 29th IEEE Int. Conf. on Tools with Artificial Intelligence, Nov. 6-8, Boston, U.S. (2017)
39. LWP23D: Game map: Factory, https://www.blendswap.com/blends/view/81600
40. exedesign: Factory, http://www.blendswap.com/blends/view/55233
41. Parque, V., Miyashita, T.: On Succinct Representation of Directed Graphs. IEEE Int. Conf. on Big Data and Smart Computing, pp. 199-205 (2017)
42. Parque, V., Kobayashi, M., Higashi, M.: Bijections for the numeric representation of labeled graphs. IEEE Int. Conf. on Systems, Man and Cybernetics, pp. 447-452 (2014)
43. Parque, V., Kobayashi, M., Higashi, M.: Searching for Machine Modularity using Explorit. IEEE Int. Conf. on Systems, Man and Cybernetics, pp. 2599-2604 (2014)
44. Parque, V., Kobayashi, M., Higashi, M.: Neural computing with concurrent synchrony. In: Neural Information Processing, Cham, Springer International Publishing (2014) 304–311
45. Parque, V., Miyashita, T.: On k-subset sum using enumerative encoding. In: IEEE Int. Symp. on Signal Processing and Information Technology. (2016) 81–86