Self-Organizing Map for the Multi-Goal Path Planning with Polygonal Goals

Jan Faigl and Libor Přeučil

Czech Technical University in Prague - Department of Cybernetics
{xfaigl,preucil}@labe.felk.cvut.cz

Abstract. This paper presents a self-organizing map approach for the multi-goal path planning problem with polygonal goals. The problem is to find a shortest closed collision free path for a mobile robot operating in a planar environment represented by a polygonal map $\mathcal{W}$. The requested path has to visit a given set of areas where the robot takes measurements in order to find an object of interest. Neurons' weights are considered as points in $\mathcal{W}$ and the solution is found as approximate shortest paths connecting the points (weights). The proposed self-organizing map has less number of parameters than a previous approach based on the self-organizing map for the traveling salesman problem. Moreover, the proposed algorithm provides better solutions within less computational time for problems with high number of polygonal goals.

1 Introduction

A problem of finding a collision-free path for a mobile robot such that the robot visits a given set of goals is called the multi-goal path planning problem (MTP). The problem arises in various robotic tasks and one of them is an inspection task in which model of the robot work space is a priori known. A model can be a building plan that can be represented as the polygonal domain, i.e., a polygonal map with obstacles. In such a map, a goal can be a single point or a polygonal region. Goals represent places in the environment where a mobile robot takes measurements. A practical motivation for this type of problems are searching missions where a mobile robot has to inspect the environment to find an object of interest, e.g., victims in search&rescue missions [7].

The planning problem for point goals can be formulated as the well-known traveling salesman problem (TSP), and for which many self-organizing map (SOM) approaches have been proposed since the first work of Angéniol and Fort. In the case of polygonal goals, the problem formulation can be found as the safari route problem [8], or the zookeeper problem [2]. These problems can be solved in a polynomial time for particular restricted problem formulations, e.g., problems without obstacles, with a given starting point, and polygonal goals attached to the boundary. However, these problem variants can be formulated as the traveling salesman problem with neighborhoods (TSPN) [6]. Although approximation algorithms for restricted variants of the TSPN exist [3,1], in general, the TSPN is APX-hard and cannot be approximated with a factor $2 - \epsilon$, where $\epsilon > 0$, unless P=NP [9].

T. Honkela et al. (Eds.): ICANN 2011, Part I, LNCS 6791, pp. 85–92, 2011.

Here, it is worth to mention that SOM approaches for the TSP are focused on its Euclidean variant, i.e., distances between nodes and goals are determined as the Euclidean distances between two points. The main difference of the MTP is that a path between two goals (or node–goal path) has to be collision free; thus, geodesic paths (distances) avoiding the collision with obstacles have to be considered in the self-organizing procedure, which increases the complexity of the adaptation process.

In this paper, new SOM adaptation procedure for the MTP with polygonal goals is proposed. The approach follows standard SOM adaptation schema for the TSP that has been extended to the polygonal domain using approximate shortest path in [5]. The adaptation uses new winner selection procedure that finds and creates new neurons using a distance to a segment of the goal. Moreover, practical aspects of the adaptation process in the polygonal map are considered to decrease the computation burden of the adaptation. In addition, simplified adaptation rules based on [11] are used and together with the novel winner selection procedure they lead to less number of adaptation parameters. The proposed procedure is also able to deal with point goals. As such, it provides a unified way to solve various modifications of the MTP, which includes safari route problem and also the watchman route problem as a variant of the MTP where goals are polygons of a convex cover set of $\mathcal{W}$ [4].

2 Self-Organizing Map for Multi-Goal Path Planning with Polygonal Goals

The problem addressed in this paper can be defined as follows. Having a polygonal map $\mathcal{W}$ and a set of goals $\mathbf{G} = \{g_1, \ldots, g_n\}$, the problem is to find a closed shortest path such that the path visits at least one point of each goal $g_i \in \mathbf{G}$. A goal can be a single point, or a polygonal region, and all goals entirely lie in $\mathcal{W}$. A polygonal goal g is represented as a sequence of points $g = (p_1^g, \ldots, p_k^g)$, which forms a border of g represented as a set of straight line segments $\delta g = \{s_1^g, s_2^g, \ldots, s_k^g\}$, where s_i^g is a straight line segment inside $\mathcal{W}$, $s_i^g = (p_i^g, p_{i+1}^g)$ for $0 \le i < k$, and $s_k^g = (p_k^g, p_1^g)$.

The proposed adaptation procedure is based on two-layered competitive neural network. The input layer consists of two dimensional input vector. An array of output units is the second layer, and it forms a uni-dimensional ordered structure. The neuron's weights represent coordinates of a point in $\mathcal{W}$, which is called node, and denoted as ν in this paper. Connected nodes form a ring that represents the requested path. In SOM for the TSP (for example [10]), goals are presented to the network in a random order and neurons compete to be the winner using the Euclidean distances between them and the goal. Then, the winner node is adapted towards the presented goal. However, in the MTP, a collision free path has to be determined because of obstacles in $\mathcal{W}$. The adaptation process may be considered as a node movement along the node–goal path towards the goal, i.e., the node (neuron's weights) is placed on the path closer to the goal while it travels distance according to the neighbouring function f.

An approximation of the shortest path may be used for the node–goal path determination [5].

Novel winner selection procedure is proposed to address polygonal goals. The procedure is based on consideration of the ring as a sequence of straight line segments in $\mathcal{W}$. Again, due to obstacles in $\mathcal{W}$, such a sequence is found using an approximate shortest path between two points (point–point path) in $\mathcal{W}$ [4].

Let the ring r be a sequence of line segments $r = (s_1^r, s_2^r, \ldots, s_l^r)$. The winner node is found as a "closest" point of the ring to the set of segments representing the goal g. The exact shortest path between two segments in $\mathcal{W}$ is substituted by the following approximation. First, the Euclidean distance between the segments s_i^r and s_j^g is determined; thus, two points on the segments are found, $p_r \in s_i^r$ and $p_g \in s_j^g$. The point–point path for these points is found to approximate the shortest path between two segments in $\mathcal{W}$. So, a pair (p_r, p_g) with the minimal length of the approximate shortest path between p_r and p_g is the result of the winner selection procedure. The point p_r is used for creating new node if a node with the same coordinates is not already in the ring. The found point p_g at the goal segment is used as a point goal towards which nodes are adapted using the point–point path. In the case of a point goal g, a similar procedure is used for approximating shortest segment–point path and p_g is the point goal itself.

The adaptation is an iterative stochastic procedure starting with an initial creation of m nodes, where $m = 2n$ and n is the number of goals. The neurons' weights are set to form a small circle around the first goal g_1, or around the centroid of g_1 for the polygonal goal. The used neighbouring function is $f(\sigma, d) = exp(-d^2/\sigma^2)$ for $d < 0.2m$, and $f(\sigma, d) = 0$ otherwise, where σ is the learning gain (the neighbouring function variance) and d is the distance of the adapted node from the winner node measured in the number of nodes (the cardinal distance). The adaptation process performs as follows.

1. *Initialization:* For a set of n goals $\mathbf{G}$ and a polygonal map $\mathcal{W}$, create $2n$ nodes around the centroid of the first goal. Let the initial value of the leaning gain be $\sigma = 10$, and adaptation parameters be $\mu = 1$, $\beta = 10^{-5}$, and $i = 1$.
2. *Randomizing:* Create a random permutation of goals $\Pi(\mathbf{G})$.
3. *Winner Selection:* For a goal $g \in \Pi(\mathbf{G})$ and the current ring r as a path in $\mathcal{W}$ find the pair (p_r, p_g) using the proposed winner selection procedure. Create a new node ν with coordinates p_r if such a node does not already exist. A node at the coordinates p_r is the winner node $\nu^\star$.
4. *Adapt:* If g is a point goal or $\nu^\star$ is not inside the polygonal goal g:
   - Let the current number of nodes be m, and N be a set of $\nu^\star$'s neighborhoods in the cardinal distance less than or equal to $0.2m$.
   - Move $\nu^\star$ along approximate shortest path $S(\nu^\star, p_r)$ towards p_r by the distance $|S(\nu^\star, p_r)|\mu$, where $|S(.,.)|$ is the length of the approximate path.
   - Move nodes $\nu \in N$ for which $\mu f(\sigma, d) < \beta$ towards p_r along $S(\nu, p_r)$ by the distance $|S(\nu, p_r)|\mu f(\sigma, d)$, where f is the neighbouring function and d is the cardinal distance of ν to $\nu^\star$.

   Remove g from the permutation, $\Pi(\mathbf{G}) = \Pi(\mathbf{G}) \setminus \{g\}$, and if $|\Pi(\mathbf{G})| > 0$ go to Step 3.

5. *Ring regeneration:* Create a new ring as a path in $\mathcal{W}$ using only the winner nodes of the current adaptation step, i.e., remove all other nodes. Make nodes from the endpoints of $s^r \in r$ that do not correspond to the winners, i.e., nodes correspond to the sequence of path's vertices.
6. *Update adaptation parameters:* Set $i = i+1$, $\sigma = (1-0.001i)\sigma$, and $\mu = 1/\sqrt[4]{i}$.
7. *Termination condition:* If all polygonal goals have particular winner inside the polygonal goal, and if all point 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 the final path using point–point approximate path in $\mathcal{W}$.

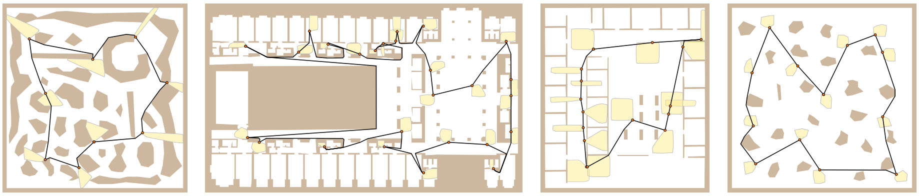
It is clear that the proposed adaptation procedure considering ring as a collision free path in $\mathcal{W}$ with the closest ring–goal segments selection is more computationally demanding than a consideration of node–goal points, which does not require determination of shortest path between two nodes. The adaptation performed only if $\mu f(\sigma, d) < \beta$ (called $\beta - condition$ rule) decreases the computational burden without significant influence to the solution quality. Also the used evolution of σ, μ [11] provides fast convergence. However, it decreases the solution quality in few cases in comparison to Somhom's parameters [10] used in [5,4]. An experimental comparison of these algorithms is presented in Section 3.

Regarding the necessary parameters settings the main advantage of the proposed procedure is that it does not require specific parameters tuning. Based on several experiments the procedure seems to be insensitive to changes of the initial values of σ and μ. Also the used size of the winner neighborhood ($0.2m$) provides the best trade-off between the solution quality and computational time.

It is worth to mention that the used approximation of the shortest path between two points (described in [4]) is more computationally demanding, and it is less precise than the node–goal path approximation. However, it requires less memory. It is because precomputed shortest paths from all map vertices to the goals are used in the node–goal path queries. Thus, lower memory requirements and a faster initialization are additional advantages of the proposed method.

3 Experiments

The proposed adaptation procedure has been experimentally verified in two sets of problems with polygonal goals, and compared with the SOM approach for the watchman route problem (WRP) [4]. The first set represents a "generalized" safari route problem, where convex polygonal goals, possibly overlapping each other, are placed in $\mathcal{W}$. The second set represents the WRP with restricted visibility range presented in [4]. Moreover, the proposed procedure has been compared with the SOM approach for the TSP in $\mathcal{W}$ [5] where goals are points.



(a) *dense*$_5$*-A* (b) *h2*$_5$*-A* (c) *jh*$_4$*-A* (d) *potholes*$_2$*-A*

Fig. 1. Selected solutions of the safari route problems, light polygons are goals, small disk at convex goal are the last winner nodes, black lines are found paths

The WRP algorithm adapts nodes towards centroids of the convex polygonal goals[1]. An alternate point is determined at the polygon border using node–centroid path to avoid placement of nodes too close to the polygon centroid, i.e., the node movement towards the centroid is stopped at the border. For the safari route problem, the WRP algorithm has been modified to do not consider the ring coverage, and to adapt nodes towards the determined alternate points. Besides, the WRP and the TSP algorithms has been modified to use the $\beta - condition$ rule and the Euclidean distance for pre-selection of winner nodes candidates, i.e., approximate node–goal path is determined only if the Euclidean node–goal distance is less than the distance of the current winner node candidate to the goal. These two modifications are technical, as they do not affect the solution quality; however, they decrease the computational burden several times.

The examined algorithms have been implemented in C++, compiled by the G++ 4.2.1 with the -O2 optimization, and executed within the same computational environment using single core of the i7-970 CPU at 3.2 GHz, and 64-bit version of the FreeBSD 8.2. Thus, the presented average values of the required computational times T can be directly compared.

The SOM algorithms are randomized, and therefore, each problem has been solved 50 times, and 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. Reference solutions from [4,5] are used for the WRPs and the TSPs, and the solution quality is measured as the percent deviation to the reference path length of the average path length, $PDM = (L-L_{ref})/L_{ref} \cdot 100\%$, and as the percent deviation from the reference of the best solution, $PDB = (L_{min}-L_{ref})/L_{ref} \cdot 100\%$. All presented length values are in meters. The number of goals is denoted as n in the presented tables.

The experimental results for the safari route problems are presented in Table 1 and selected best solutions found by the proposed algorithm are depicted in Figure 1. The proposed procedure provides better solutions for most of the problems. The procedure is more computationally demanding for complex environments like the problem h2$_5$-A because shortest paths have many segments. This is also the case of the jh$_{10}$-coverage problem, which is an instance of the WRP with many overlapping convex goals.

[1] In [4], triangles of a triangular mesh are used to support determination of ring coverage, which is not necessary for safari route problems.

Table 1. Experimental results for the safari route problems

Problem	n	SOM for WRP [4]				Proposed			
		L	s_L%	L_{min}	T [s]	L	s_L%	L_{min}	T [s]
dense-small	35	114.2	3.45	105.63	0.34	113.7	3.99	102.80	0.98
$dense_5$-A	9	62.6	1.96	60.66	0.14	59.0	2.77	58.05	0.23
$h2_5$-A	26	407.2	0.98	399.34	1.22	405.2	0.88	396.07	2.12
jh-rooms	21	88.3	0.76	87.84	0.13	88.1	0.10	87.83	0.15
jh_{10}-doors	21	67.6	1.34	66.11	0.16	63.7	1.43	61.99	0.15
jh_{10}-coverage	106	106.9	1.34	103.89	1.49	97.9	6.20	92.99	2.66
jh_4-A	16	61.1	1.86	58.71	0.33	57.3	1.32	56.59	0.32
jh_5-corridors	11	65.8	1.87	62.77	0.14	59.7	0.35	59.53	0.20
pb_5-A	7	275.8	4.47	265.29	0.31	271.7	4.36	264.70	0.31
$potholes_2$-A	13	71.9	1.91	70.37	0.04	71.6	2.08	70.09	0.08

Table 2. Experimental results for the WRP

Map	d [m]	n	L_{ref} [m]	SOM for the WRP [4]				Proposed				
				PDM	**PDB**	s_L%	T [s]	**PDM**	**PDB**	L_{min}	s_L%	T [s]
jh	inf	100	207.8	-52.67	-53.39	1.53	1.45	-53.71	-54.17	95.27	2.78	2.40
jh	10.0	108	207.3	-51.84	-53.02	3.27	1.95	-50.65	-54.02	95.30	6.89	2.64
jh	5.0	130	216.4	-48.67	-51.75	3.39	1.27	-51.54	-53.06	101.56	4.18	5.75
jh	4.0	169	219.9	-43.48	-46.34	3.22	2.97	-48.38	-49.42	111.22	2.71	9.21
jh	3.0	258	225.5	-27.92	-30.60	1.61	5.18	-35.04	-37.04	142.01	2.27	13.12
jh	2.0	480	281.9	-8.99	-11.09	1.06	20.68	-17.25	-19.85	225.91	1.64	23.16
jh	1.5	852	350.3	-3.81	-5.51	1.02	109.81	-14.56	-15.68	295.39	0.74	100.40
jh	1.0	1800	470.8	3.96	2.36	0.59	430.88	-9.06	-10.25	422.50	0.55	452.03
pb	inf	52	533.3	-18.11	-22.26	4.98	1.44	-16.18	-23.18	409.69	5.70	1.28
pb	10.0	111	612.7	-12.48	-14.86	3.92	2.57	-15.46	-17.94	502.78	4.73	3.46
pb	5.0	262	682.9	-7.35	-9.34	2.45	5.56	-7.01	-10.62	610.38	4.45	15.23
pb	4.0	373	720.1	-6.17	-8.78	3.25	16.80	-7.46	-10.09	647.41	3.16	20.37
pb	3.0	714	774.8	-5.62	-6.72	0.55	42.52	-3.04	-9.54	700.81	6.95	114.08
pb	2.0	1564	901.9	-2.88	-4.41	1.02	244.72	-0.30	-9.40	817.12	4.53	373.74
pb	1.5	2787	1115.9	1.03	0.07	0.54	997.68	-9.12	-12.12	980.59	2.27	1078.42
pb	1.0	6188	1564.2	2.55	1.90	0.41	5651.06	-12.52	-13.89	1346.87	0.78	3276.43
ta	inf	46	203.6	-30.99	-31.48	0.52	0.28	-33.67	-33.94	134.52	1.69	0.76
ta	10.0	70	202.6	-28.11	-28.80	0.28	0.41	-28.36	-28.89	144.08	1.45	1.63
ta	5.0	152	254.1	-15.68	-17.97	1.81	1.26	-19.61	-20.35	202.39	0.83	6.39
ta	4.0	209	272.2	-7.39	-9.91	1.36	3.69	-15.70	-16.65	226.90	0.85	11.64
ta	3.0	357	315.0	-6.28	-8.75	1.61	12.61	-13.46	-14.42	269.57	1.30	15.58
ta	2.0	757	408.3	1.09	-1.20	0.87	66.48	-10.97	-12.52	357.18	1.00	59.00
ta	1.5	1320	522.1	1.06	-1.18	0.97	194.25	-12.81	-13.63	450.92	0.59	251.08
ta	1.0	2955	743.6	5.21	3.80	0.57	1398.71	-12.45	-13.54	642.89	0.57	987.77

The results for the WRP are presented in Table 2, where d denotes the restricted visibility range. Also in this type of problems, the proposed procedure provides better solutions. Although the procedure is more computationally demanding for small problems, it provides significantly better results with less required computational time for problems with d=1 m, which have many convex polygons. The results indicate that the proposed procedure scales better with

Table 3. Experimental results for the TSP

Problem	n	L_{ref} [m]	**SOM for the TSP** [5]				**Proposed**			
			PDM	**PDB**	s_L%	T [s]	**PDM**	**PDB**	s_L%	T [s]
jari	6	13.6	0.36	0.00	0.55	0.01	0.23	0.00	0.15	0.01
complex2	8	58.5	-0.00	-0.00	0.00	0.01	0.47	-0.00	1.60	0.02
m1	13	17.1	0.31	0.00	1.15	0.02	0.17	0.00	0.20	0.03
m2	14	19.4	9.52	0.00	3.50	0.03	10.76	5.32	3.16	0.04
map	17	26.5	5.92	0.73	4.39	0.05	6.87	0.73	4.37	0.07
potholes	17	88.5	4.58	2.37	2.17	0.06	5.56	2.37	2.48	0.06
a	22	52.7	0.89	0.31	1.00	0.09	1.58	0.31	2.37	0.11
rooms	22	165.9	1.02	0.00	0.86	0.11	0.12	0.00	0.11	0.15
dense_4	53	179.1	15.04	8.33	3.16	0.68	18.17	9.00	2.38	0.68
potholes_2	68	154.5	6.12	2.50	2.01	0.65	7.54	3.11	2.23	0.35
m3_1	71	39.0	6.71	2.29	1.53	1.41	8.72	4.80	1.64	1.00
warehouse_4	79	369.2	5.97	2.42	2.13	1.92	8.47	2.87	2.68	0.81
jh_2	80	201.9	1.94	0.48	0.64	0.95	2.04	0.67	0.66	0.71
pb_4	104	654.6	1.06	0.01	1.34	1.53	1.95	0.51	3.05	0.84
ta_2	141	328.0	2.97	1.69	0.69	2.27	3.69	2.19	0.75	1.11
h2_5	168	943.0	2.85	2.00	0.60	8.75	2.42	1.65	0.53	6.70
potholes_1	282	277.3	6.84	4.91	1.02	10.47	6.97	4.19	0.91	2.71
jh_1	356	363.7	4.02	2.74	0.56	22.29	4.32	3.23	0.46	7.05
$\text{pb}_{1.5}$	415	839.6	2.60	1.12	2.25	24.13	10.40	1.47	5.21	6.62
h2_2	568	1 316.2	2.81	1.87	0.51	87.61	3.00	1.97	0.46	32.19
ta_1	574	541.1	5.51	4.63	0.41	38.11	6.39	4.88	0.73	10.86

increasing number of goals. The reason for this is in the number of involved neurons. While the algorithm [4] derives the number from the number of goals, the proposed procedure dynamically adapts the number of neurons using shortest path in $\mathcal{W}$. Thus, for very large problems in the same map, additional neurons do not provide any benefit, and only increase the computational burden. The worse average results for the map *pb*, d=3 and d=2 are caused by the used point–point shortest path approximation, which provides unnecessary long paths in several cases. Nevertheless, the proposed procedure is able to find significantly better solutions, regarding the PDB, than the WRP algorithm [4].

The results for the TSP are presented in Table 3. The proposed procedure provides competitive results to the algorithm [5]. Worse average solutions are found for several problems. In these cases, the point–point path approximation provides longer paths than the point–goal path used in the TSP algorithm. The used schema of parameters evolution [11] leads to faster convergence, which "compensates" the more complex winner selection. However, the schema is the main reason for the worse performance of the proposed procedure than the TSP algorithm [5] with parameters' evolution [10].

4 Conclusion

Novel winner selection procedure for self-organizing maps has been proposed in this paper. The proposed adaptation procedure is able to deal with variants of the multi-goal path planning problem including the TSP, the WRP and the safari route problem. Moreover, the procedure can be considered as parameterless, as the number of neurons is determined during the adaptation process. It provides a unified approach to solve various routing problems in the polygonal domain $\mathcal{W}$.

Although the proposed algorithm provides outstanding results in many cases, both the required computational time and the solution quality may be improved as the former algorithms for the WRP and the TSP provide better results in particular problems. Both these aspects are related to the evolution of the adaptation parameters, e.g., σ, μ, or size of the winner node neighborhood. Besides, the utilized approximation may be improved. Shortest path approximation and investigation of adaptation schemata with different evolution of parameters are subjects of the further work.

Acknowledgments. The work presented in this paper has been supported by the Ministry of Education of the Czech Republic under the program "National research program II" by Project No. 2C06005.

References

1. de Berg, M., Gudmundsson, J., Katz, M.J., Levcopoulos, C., Overmars, M.H., van der Stappen, A.F.: Tsp with neighborhoods of varying size. Journal of Algorithms 57(1), 22–36 (2005)
2. Chin, W.-P., Ntafos, S.: The zookeeper route problem. Information Sciences: an International Journal 63(3), 245–259 (1992)
3. Dumitrescu, A., Mitchell, J.S.B.: Approximation algorithms for tsp with neighborhoods in the plane. Journal of Algorithms 48(1), 135–159 (2003)
4. Faigl, J.: Approximate Solution of the Multiple Watchman Routes Problem with Restricted Visibility Range. IEEE Transactions on Neural Networks 21(10), 1668–1679 (2010)
5. Faigl, J., Kulich, M., Vonásek, V., Přeučil, L.: An Application of Self-Organizing Map in the non-Euclidean Traveling Salesman Problem. Neurocomputing 74, 671–679 (2011)
6. Goodman, J.E., O'Rourke, J. (eds.): Handbook of Discrete and Computational Geometry. CRC Press LLC, Boca Raton (2004)
7. Kulich, M., Faigl, J., Přeučil, L.: Cooperative planning for heterogeneous teams in rescue operations. In: IEEE International Workshop on Safety, Security and Rescue Robotics (2005)
8. Ntafos, S.: Watchman routes under limited visibility. Computational Geometry: Theory and Applications 1(3), 149–170 (1992)
9. Safra, S., Schwartz, O.: On the complexity of approximating tsp with neighborhoods and related problems. Computational Complexity 14(4), 281–307 (2006)
10. Somhom, S., Modares, A., Enkawa, T.: A self-organising model for the travelling salesman problem. Journal of the Operational Research Society, 919–928 (1997)
11. Zhang, W., Bai, Y., Hu, H.P.: The incorporation of an efficient initialization method and parameter adaptation using self-organizing maps to solve the TSP. Applied Mathematics and Computation 172(1), 603–623 (2006)