Comparison of metaheuristic methods by solving travelling salesman problem

Ondřej Míča[1]

Abstract: *Travelling salesman problem (TSP) belongs in basic problems of operations research. It is a NP-hard problem. The number of possible solutions of this problem is very high – it increases with the factorial of the number of the nodes at the graph. So even with nowadays computers it takes very large amount of time to solve TSP with exact methods. Therefore TSP is now usually solved with a heuristic (or metaheuristic) techniques, which provides a satisfactory solution in real-time.*

This paper focuses on four classical metaheuristic methods: tabu search, simulated annealing, genetic algorithm and ant colony optimization algorithm, and compares all algorithms using difference between best given solution and optimal solution as evaluation criterion. Computational results on several standard instances of TSP show efficiency of all scrutinized methods.

Key words: Travelling salesman problem · Metaheuristic optimization · Tabu search · Simulated annealing · Genetic algorithm · Ant colony optimization algorithm

JEL Classification: C61 · C63

1. Introduction

Travelling salesman problem is very well known and popular optimization problem. The main issue in TSP is to find Hamiltonian cycle (called after the Irish mathematician William Rowan Hamilton, who is considered to formulate TSP in 19th century). It means to visit all nodes (cities) of the graph exactly once with the possible shortest route and return to the origin node. (Cenek, Jánošíková, 2008)

This seeking the Hamiltonian cycle is situated on a transportation network which can be described as graph G = (**N, E**), where **N** is set of *n* nodes (cities) and **E** is set of *m* edges (paths) between these nodes and each edge has its own length. The network is very often a complete graph (i.e. each pair of nodes is connected by an edge). Or when the graph is not complete, fictive edge between two unconnected nodes with infinite length can be added without affecting the optimal solution. (Volek, 2008)

TSP in general can be represented with following mathematical model:

$$min \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} d_{ij} x_{ij} \tag{1}$$

$$\sum_{\substack{i=1 \\ i \neq j}}^{n} x_{ij} = 1 \text{ for } j = 1 \dots n \qquad \sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ij} = 1 \text{ for } i = 1 \dots n \tag{2}$$

$$y_i - y_j + n x_{ij} \leq n - 1 \text{ for } 2 \leq i \neq j \leq n \tag{3}$$

$$x_{ij} \in \{0,1\} \text{ for } i, j = 1 \dots n \qquad y_i \in \mathbb{N}_0 \text{ for } 2 = 1 \dots n \tag{4}$$

[1] **Ing. Ondřej Míča**, University of Pardubice, Jan Perner Transport Faculty, Department of Informatics in Transport, Studentská 95, 532 10, Pardubice 2, Czech Republic. GSM.: +420 466 036 428, E-mail: on.mica@email.cz

where:

n number of nodes,

d_{ij} distance between nodes i and j,

x_{ij} variable, which is 1 when the edge between nodes i and j belongs to the Hamiltonian cycle, otherwise it is 0.

The equalities (2) provide that each node can be entered and left only once. The constraint (3) provides that Hamiltonian cycle is only one tour, not a number of smaller cycles. (Miller, Tucker, Zemlin, 1960)

There are several cases of TSP (not all are mentioned):

- Symmetric: distance between nodes i and j is the same as distance between j and i.
- Asymmetric: distance between nodes i and j is not the same as distance between j and i.
- With time windows: each node can be visited only in given amount of time.
- Sequential Ordering Problem: nodes can be visited only in given order.
- Etc.

Further in this paper is considered only a symmetric travelling salesman problem.

Since TSP is NP-hard problem – the number of possible solutions for graph with n nodes is $(n\text{-}1)!/2$ – the exact algorithm can be used only for small number of nodes. Therefore only heuristics or metaheuristics are used for solving large TSP. (Applegate, 2006)

2. Methods

2.1 Tabu search

Tabu search optimization method was designed by Fred Glover in late 1980's (Glover, 1986). It uses neighborhood search with best admissible selection of improved solution. To prevent sticking in local suboptimal solution tabu list is implemented. The tabu list is a short term memory structure which defines a set of solution explored in previous k iterations of the algorithm where k is parameter of the method also called the tabu tenure. The algorithm runs until some stopping criterion has been satisfied (amount of time, number of iterations without improving best solution, etc.) (Glover, 1990)

2.2 Simulated annealing

The inspiration for this algorithm is from metallurgy. The algorithm was described for first time by IBM research team (Kirkpatrick, Gelatt Jr, Vecchi, 1983). As tabu search algorithm, this approach also uses neighborhood search. The possible move is made with probability (5).

$$p(x, x', T) = \begin{cases} e^{\left(-\left(f(x') - f(x)\right)\right)/T} & \text{when } f(x') \geq f(x) \\ \\ 1 & \text{when } f(x') < f(x) \end{cases} \tag{5}$$

where:

$f(x)$ objective function of solution x,

T the parameter of the algorithm called temperature.

Each run of the algorithm stars with T set to high value and temperature is decreased with each next iteration of algorithm. The simulated annealing can be extended with heating – rising of the temperature. The algorithm stops when some stopping criterion is met (amount of time, number of iterations without improving best solution, etc.). (Kirkpatrick, Gelatt Jr, Vecchi, 1983)

2.3 Genetic algorithm

Genetic algorithm belongs to nature-based algorithms. The inspiration of this algorithm is a process of natural selection and evolution. This method was first designed by John Holland in the early 1970's. The goal of the genetic algorithm is to breed new population of candidate solutions from old population, as it goes in the nature. The population size k is the parameter of the method.

At the beginning of each iteration candidates for breeding are stochastically selected from old generation according to their fitness (usually objective function). Next generation of solution is generated from selected individuals using combination of genetic operators: crossover and mutation. The stopping conditions for genetic algorithm can be time, number of generated population, etc. (Gendreau, Potvin, 2010)

2.4 Ant colony optimization algorithm

This method is based on examining ant colonies and studying cooperation and communication of ants when they are searching for food. It was first presented by Marco Dorigo in his doctoral thesis. (Dorigo, 1992)

Ants communicate with each other using pheromone trails. The pheromones are chemical substances used by ants to mark their paths. Ants leave pheromone trails on a ground and other ants can scent the direction and intensity of those pheromones. Each ant which uses a marked path renews a pheromone trail because it evaporates during a time (loses its attractive strength). When the path is not used for some time, the pheromone trail slowly disappears.

At the beginning of the algorithm the m ants are placed into randomly selected nodes of the graph. At each step they move to the new node with probability given by the random proportional rule defined as

$$p_{ij} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{l \in N_k}(\tau_{il})^{\alpha}(\eta_{il})^{\beta}} \tag{6}$$

where:

τ_{ij} amount of pheromone on the edge between nodes i and j,

η_{ij} attractiveness of edge from i to j calculated as 1 divided by length of the edge,

α, β parameters of the method used to weigh the relative influence of the pheromone and the attractiveness of the edge,

N_k is set of unvisited nodes for k^{th} ant.

After visiting all nodes (the set of unvisited nodes is empty) the ants returns to their original nodes and the amount of pheromone on each edge is updated according to the following formula

$$\tau_{ij} = (1-\rho)\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{7}$$

where:

ρ evaporation parameter,

$\Delta\tau_{ij}^{k}$ is defined as

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k} \text{ if } k^{th} \text{ant uses edge } (i,j) \text{in its Hamiltonian cycle} \\ 0 \text{ otherwise} \end{cases} \tag{8}$$

where:

Q constant chosen before the run of the algorithm,

L_k length of found Hamiltonian cycle of k^{th} ant. (Dorigo, Gambardella, Vecchi, 1997)

3. Research results

Each of the listed metaheuristics was implemented in Java language. For benchmarking and testing examples was used TSPLIB (TSPLIB, 2001). It is a library of sample instances for the travelling salesman problem from various sources and of various types. This library nowadays contains 112 instances of symmetric TSPs with its optimal solutions. For this paper was chosen 8 instances of TSP with different numbers of nodes (51, 76, 96, 130, 159, 198, 225 and 262 nodes).

For each tested instance of TSP was made 500 runs of described methods, and each run was limited to 500 iterations. So together 16 000 runs of TSP were made for testing.

For each tested combination the best given solution was taken and this best solution was compared with the optimal solution of instance taken from (TSPLIB, 2001) and was determined the difference between the best and optimal solution.

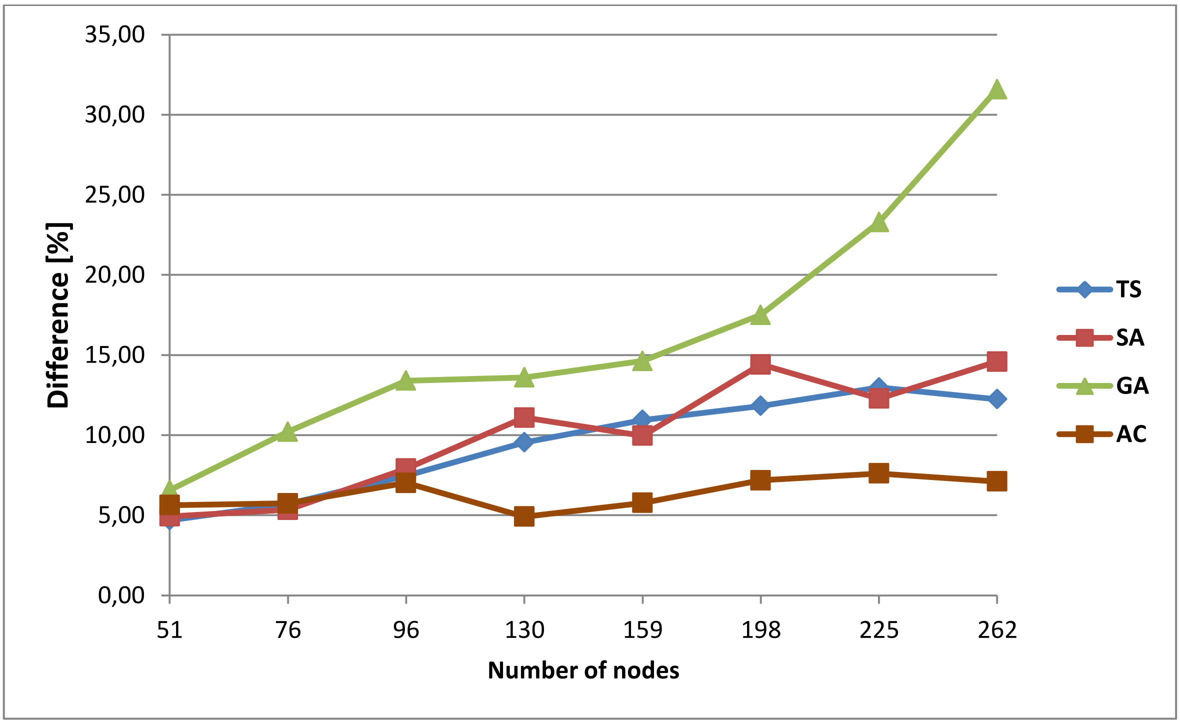
The gained results are shown in table 1 and figure 1.

Table 1: Difference between the best and optimal solution for each tested combination [%]

	Algorithm				
Number of nodes		**TS**	**SA**	**GA**	**ACO**
	51	4.69	4.93	6.57	5.63
	76	5.70	5.34	10.21	5.74
	96	7.46	7.91	13.40	7.02
	130	9.54	11.10	13.60	4.91
	159	10.94	9.97	14.63	5.77
	198	11.82	14.40	17.51	7.18
	225	12.97	12.28	23.29	7.61
	262	12.24	14.59	31.58	7.11

Source: Own processing

Figure 1: Difference between average and optimal solution



Source: Own processing

4. Conclusions

For TSP instance with fewer nodes all used methods gives solution with similar differences. But with rising number of nodes, the genetic algorithm becomes more and more inaccurate. Tabu search and simulated annealing methods give solution with very similar differences. A best result in almost all cases gives the ant colony optimization algorithm.

So the best result of solving travelling salesman problem from all compared methods gives Ant colony optimization. The disadvantage of this method is higher difficulty of implementing this algorithm. It is also important to say, that all methods are very sensitive to parameters setting. So with different setting they can give better or much worse results.

References

Applegate, D. L. (2006). The traveling salesman problem: a computational study. Oxford: Princeton University Press, 593 p. Princeton series in applied mathematics. ISBN: 0-691-12993-2.

Cenek, P., Jánošíková, Ľ. (2008). Models and optimisation in transports and logistics. EDIS – vydavatel'stvo ŽU: Žilina.. 186 p. ISBN: 978-80-8070-951-8.

Dorigo, M. (1992). Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italy.

Dorigo, M., Gambardella, L. M. and Vecchi, M. P. (1997). Ant colonies for the travelling salesman problem. *Biosystems*. 43(2), 73-81. doi: 10.1016/S0303-2647(97)01708-5. ISSN: 0303264.

Gendreau, M., & Potvin. J. Y. (2010). Handbook of metaheuristics. 2nd ed. New York: Springer, 648 p. ISBN 978-1-4419-1663-1.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5), 533-549. doi: 10.1016/0305-0548(86)90048-1.

Glover, F. (1990), Tabu Search: A Tutorial. *Interfaces.* 74-94. doi: 10.1287/inte.20.4.74.

Kirkpatrick, S., Gelatt Jr, C. D., and Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220 (4598), 671-680. doi: 10.1126/science.220.4598.671.

Miller, C. E., Tucker, A. W. and Zemlin, R. A. (1960). Integer Programming Formulation of Traveling Salesman Problems. *Journal of the ACM*, 7(4), 326-329. doi: 10.1145/321043.321046. ISSN: 00045411.

TSPLIB. Ruprecht-Karls-Universität Heidelberg [online]. 2001 [accessed 2015-10-05]. Available from: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

Volek, J. (2008). Operační výzkum I. Vydání 2., nezměn. Pardubice: Univerzita Pardubice, 111 p. ISBN 978-80-7395-073-6.