IMPACT OF PARAMETERS α, β AND ρ ON QUALITY OF SOLUTION TO TRAVELLING SALESMAN PROBLEM BY ANT COLONY OPTIMIZATION ALGORITHM

Ondřej Míča

University of Pardubice, Faculty of Transport Engineering
Studentská 95
Pardubice, 532 10, Czech Republic
+420 466 036 428
ondrej.mica@upce.cz

Abstract: *The travelling salesman problem is a well-known and popular optimization problem. Because it is an NP-hard problem, the number of permissible solutions is very high – it grows with the number of nodes in the transport network. So even with nowadays computers, it takes very large amount of time to solve TSP with exact methods.*
Ant colony optimization is metaheuristic algorithm inspired by nature. It has several parameters, which have to be set before the first run of algorithm and their correct setting is important for finding a satisficing solution. This paper deals with the effect of setting these parameters on the quality of the solution found.

Keywords: *ant colony optimization, metaheuristic, optimization, travelling salesman problem*

1 Introduction

Although computing performance continues to grow, some optimization tasks especially discrete combinatorial problems cannot be resolved within a reasonable time period for their computational complexity over a larger scale. That is why complex sophisticated methods, commonly called metaheuristics, are increasingly being promoted, which offer a quality (sub) optimal solution in a relatively short time.

2 Traveling salesman problem

Traveling salesman problem (TSP) is one of the basic tasks of operational research. The goal of TSP is to visit all nodes (cities) of the graph exactly once with the possible shortest route and return to the origin node (i.e. to find Hamiltonian cycle called after the Irish mathematician William Rowan Hamilton, who is considered to formulate TSP in 19th century). [1][1]

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 a complete graph (i.e. each pair of nodes is connected by an edge) very often. Or when the graph is not complete, fictive edge between two unconnected nodes with length of shortest path between these two nodes can be added without affecting the optimal solution. [2][2]

TSP in general can be represented with following mathematical model: [3]

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

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

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ij} = 1 \quad \text{for } i = 1 \dots n \tag{3}$$

$$y_i - y_j + n x_{ij} \leq n - 1 \quad \text{for } 1 \leq i \neq j \leq n \tag{4}$$

$$x_{ij} \in \{0, 1\} \quad \text{for } i, j = 1 \dots n \qquad y_i \in \mathbb{N}_0 \tag{5}$$

Where n is a number of nodes, d_{ij} is a distance between nodes *i* and *j*, and variable x_{ij} is 1 when the edge between nodes *i* and *j* belongs to the Hamiltonian cycle, otherwise it is 0.

The equalities (2) and (3) provide that each node can be entered and left only once. The constraint (4) provides that Hamiltonian cycle is only one tour, not a number of smaller cycles. [3]

There are several cases of TSP (not all are mentioned): [4]

- 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.

Only a symmetric travelling salesman problem is considered further in this paper.

Since TSP is NP-hard problem [1][1] – the number of possible solutions for graph with n nodes is $(n-1)!/2$.

This means that the exact algorithm can be used only for small number of nodes.

3 Ant colony optimization algorithm

Ant colony optimization algorithm (ACO) is young probabilistic technique of discrete optimization. It was first presented by Marco Dorigo in his doctoral thesis in 1992 [5]. It was originally designed for finding the shortest path between two nodes of graph. Later, the ACO method was extended and now is used for solving a large amount of optimization problems. [6]

ACO belongs in nature-based algorithms, as e.g. Bat algorithm, Bees colony algorithm, Particle swarm optimization, etc. It is based on examining ant colonies and studying cooperation and communication of ants when they are searching for food. Ants belong to social insect; it means that the prosperity of whole colony is much more important than surviving of individual ant.

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. [5]

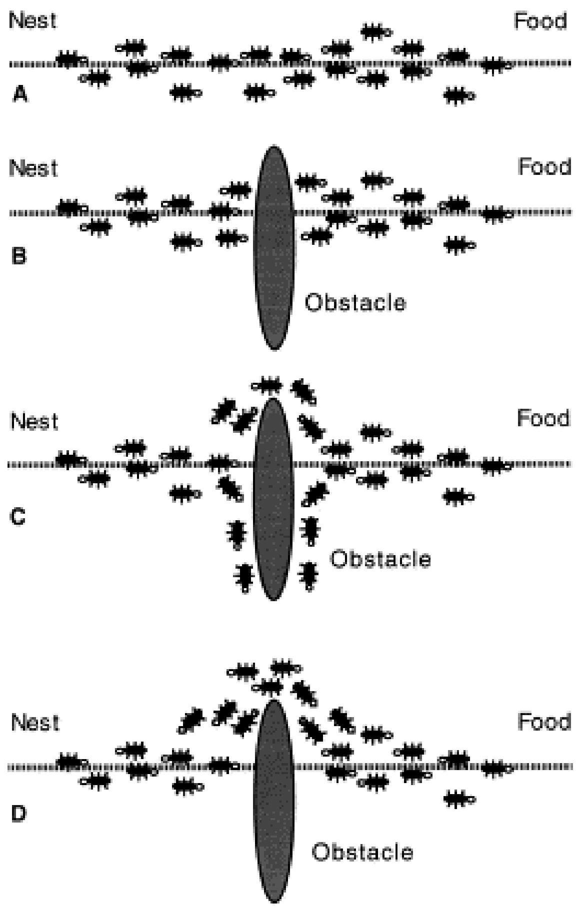

Figure 1: Reaction of ants to obstacle at their path [6]

The communication between ants by pheromones is shown at figure 1. At first (figure 1A), there is a pheromone marked path between the colony and the source of food. When a new obstacle appeared at the path (figure 1B), ants start to find a new path randomly choosing turn left or right (we can assume with the same probability) (figure 1C). At the shorter path, there is less time for pheromones to evaporate, so the density becomes higher and during a time, all ants will choose the shorter path (figure 1D). [6][6]

4 Finding solution using Ant colony optimization

When applying an ant colony method to an optimization task, an artificial ant passes through a graph and searches for a solution with the best value of objective function. The problem is, that it is not possible to determine whether the solution found is of sufficient quality (or even optimal), so the algorithm has to be defined with a stopping criterion. Typically, this is a time limit, the number of iterations without changing the best solution found, etc.

On its way through a graph, the ant moves discreetly from the node *i* to the node *j* from the neighbourhood *N(i)* (not staying at the edges) and remembers the already visited nodes.

The artificial ant decides to transition between edges *i* and *j* with a probability determined by the formula (6). The artificial ant can put the pheromone information on the edges both in the transition and the return phase when it already knows the value of the purpose function of the solution found. [7]

$$p_{ij} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{k \in N_k}(\tau_{ik})^{\alpha}(\eta_{ik})^{\beta}} \qquad (6)$$

Where τ_{ij} is amount of pheromone on the edge between nodes *i* and *j*, η_{ij} is an attractiveness of each edge according to formula (7), α and β are parameters and N_k is set of unvisited nodes for k^{th} ant.

$$\eta_{ij} = \frac{1}{d_{ij}} \qquad (7)$$

Where d_{ij} is a distance between nodes *i* and *j*.

The pseudocode for the implementation the basic ACO method is very simple: [8]

```
AntColony()
  repeat
    BuildSolution(k)
    PheromoneUpdate()
    PheromoneEvaporation()
  until StoppingCriterion
end;
```

In the procedure `BuildSolution`, *k* ants look for solutions according to the principle described above. When the search ends, it will be checked whether the value of the objective function of the best solution found by them is less than the value of the objective function of the best solution found so far. If so, the newly found solution will be saved.

The procedure `PheromoneUpdate` will modify the pheromone trace on the edges contained in the best solution found in the current iteration according to formula (8).

$$\tau_{ij} = \tau_{ij} + \Delta\tau \tag{8}$$

Where $\Delta\tau$ can be represented as a constant, or it may depend on the quality of the solution found.

The procedure `PheromoneEvaporation` will ensure the evaporation of a certain amount of pheromones throughout the graph, preventing the accumulation of pheromones at the edges of the graph and thus converging to disadvantageous paths. Evaporation of pheromones is governed by a parameter called *a coefficient of evaporation* ρ from the interval (0; 1⟩ whose value is chosen at the start of the algorithm run, according to the formula (9). [9]

$$\tau_{ij} = (1 - \rho)\tau_{ij} \tag{9}$$

In general, ACO is a complex algorithm that is very well-suited for calculating tasks whose solution is represented by a sequence of objects (eg the travelling salesman problem or the search for the shortest path between the two graph nodes).

This is a method sensitive to the correct setting of parameter values. But with good settings, it is capable of delivering high-quality solutions. Due to the great popularity of this method, there are also quite a number of modifications of this algorithm.

5 Experiment results

As mentioned in the previous chapter, the Ant Colony Method has three parameters:

- evaporation coefficient ρ,
- influence of pheromones on ant decision α,
- influence of heuristic information on ant decision β.

The TSPLIB library [10], run by Ruprecht-Karls-Universität in Heidelberg, was used to experimentally verify the effect of parameters setting on the quality of solving TSP by ACO. This is a set of 112 examples of travelling salesman problem along with their optimal solutions, many of which are based on real world issues. This library is publicly accessible and serves researchers from around the world to compare their own results.

For the solution, the Ant Colony Method was implemented in the Java programming language, and 6 implementations presented in TSPLIB, each with a different number of nodes (51, 76, 96, 130, 159 and 198 nodes), were solved with this implementation.

For each instance of TSP, the value of the monitored parameter gradually increased, the other parameters remained constant. Parameters α and β were monitored at interval ⟨0.5; 5⟩ with increase of 0.5; the coefficient ρ was monitored in the interval ⟨0.1; 1⟩ with increase of 0.1. In total, 30,000 TSP iterations were calculated. For each monitored combination of parameters, the best calculated solution was compared with the best solution reported in TSPLIB [10].

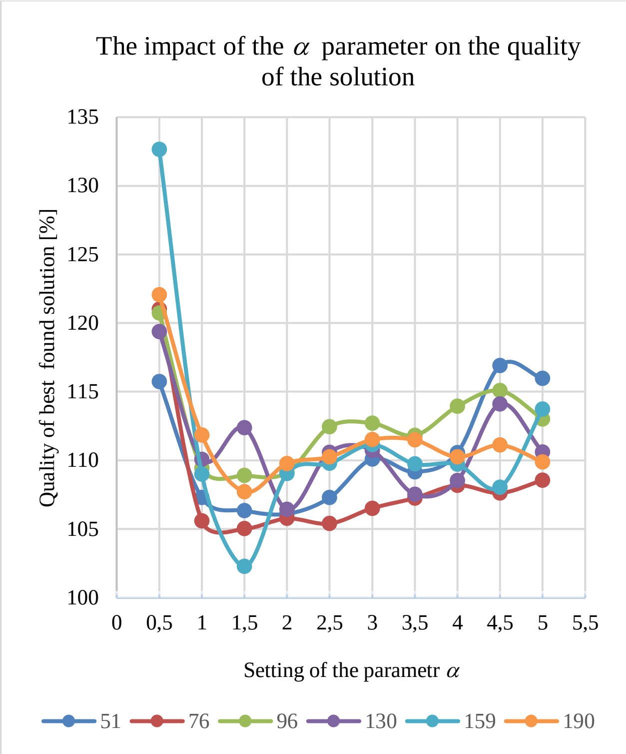

Figure 2: The impact of the α parameter on the quality of the solution

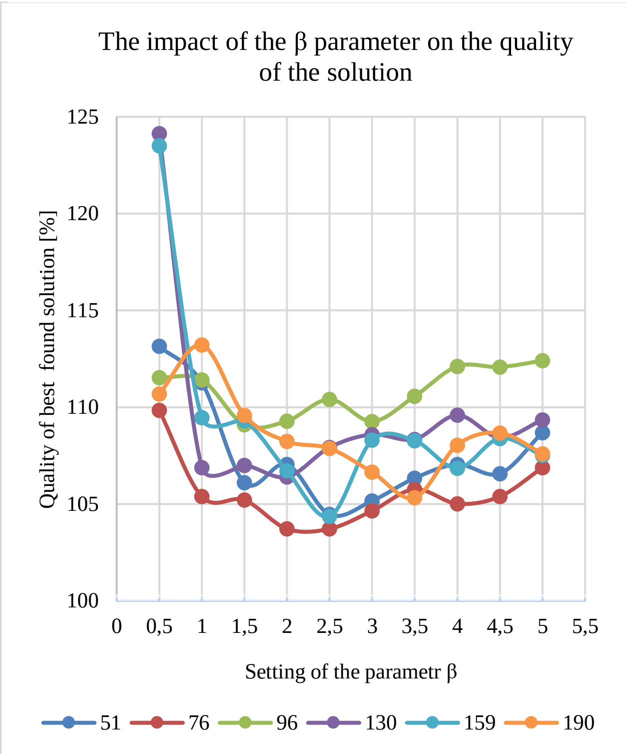

Figure 3: The impact of the β parameter on the quality of the solution

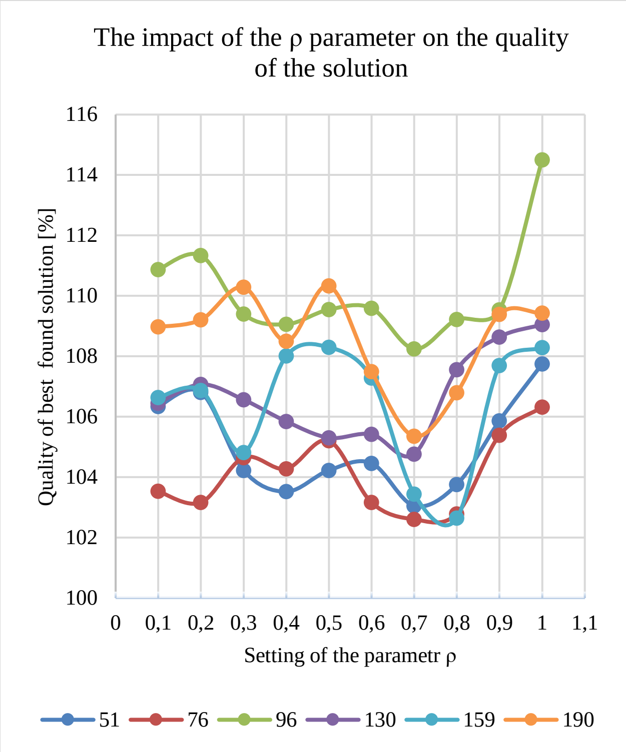

Figure 4: The impact of the ρ parameter on the quality of the solution

Figures 3 to 5 illustrate the effect of setting the individual parameters α, β and ρ on the quality of the best found solution.

6 Conclusions

This paper deals with the problem of correct parameter setting when solving the travelling salesman problem by metaheuristic method of an ant colony. By experimental verification on 6 different TSP instances, the recommended values of individual parameters were proposed, namely: parameter α in interval ⟨1; 2⟩, parameter β in interval ⟨2; 3⟩ and parameter ρ in interval ⟨0.7; 0.8⟩.

References

[1] CENEK, P. and JÁNOŠÍKOVÁ, Ľ.: Models and optimisation in transports and logistics. EDIS – vydavatel'stvo ŽU: Žilina. 2008. ISBN 978-80-8070-951-8.

[2] VOLEK, J.: Operační výzkum I. Vydání 2., nezměn. Pardubice: Univerzita Pardubice, 2008, ISBN 978-80-7395-073-6.

[3] MILLER, C. E., TUCKER, A. W. and ZEMLIN, R. A.: Integer Programming Formulation of Traveling Salesman Problems. In: Journal of the ACM. DOI: 10.1145/321043.321046. ISSN 00045411.

[4] APPLEGATE, D. L.: The traveling salesman problem: a computational study. Oxford: Princeton University Press, c2006, ix, Princeton series in applied mathematics. ISBN 0-691-12993-2.

[5] DORIGO, M.: Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italy, 1992.

[6] DORIGO, M., GAMBARDELLA, L. M. and VECCHI, M. P.: Ant colonies for the travelling salesman problem. In: Biosystems. 1997, p. 73-81. DOI: 10.1016/S0303-2647(97)01708-5. ISSN 0303264

[7] JANÁČEK, J. and BUZNA, Ľ, Optimization in networks. Žilina: University of Žilina, 2009. ISBN 978-80-8070-985-3.

[8] DORIGO, M. and STÜTZLE, T., Ant colony optimization. Cambridge: MIT Press. 2004. ISBN 0-262-04219-3.

[9] GENDREAU, M. and POTVIN, J. Y., Handbook of metaheuristics. 2nd ed. New York: Springer, 2010. ISBN 978-1-4419-1663-1.

[10] TSPLIB. *Ruprecht-Karls-Universität Heidelberg* [online]. [accessed 2018-08-20]. Available from: https://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/