OPTIMIZATION OF POSTAL ROUTES BY GENETIC ALGORITHM FOR SOLVING THE MULTIPLE TRAVELING SALESMAN PROBLEM

Martin Macík[1], Jozef Štefunko[2]

Abstract: High level of competition on postal market increases demands on reliability of postal services and lowering of transport costs. This can be achieved by optimizing the routing of postal vehicles. The article discusses the possibilities of such optimization by using graph theory. It describes basic methods of finding optimal routes using a graph. The approach, used in this article, assesses the possibility of applying meta-heuristic solution to the traveling salesman problem in the postal sector. Simulation of methods described has been applied on a regional postal network. Results showed that the software used proves to be sufficiently functional for the field of postal transport networks.

JEL Classification Numbers: L87, R53, **DOI:** http://dx.doi.org/10.12955/cbup.v3.582

Keywords: Postal network, graph theory, heuristic methods, genetic algorithm, traveling salesman

Introduction

Nowadays, high demands are placed on transport of postal items. Customer requires efficiency, safety, and, especially, the speed of delivery. The mail processing time in sorting centers is affected by the use of appropriate techniques and technology. Except the optimization of submitting a processing phase prior and after transport, the transport process itself can be optimized as well.

The task of optimization (in transport process or in general) is to find an optimal solution (optimum). The optimal solution represents the best possible solution under given conditions. It should be noted, that the mere optimization of transport network is closely related to technological processing of postal items. But, this issue is not addressed in this research (Čorejová, Achimský, Fitzová, & Kajánek, 1995).

Postal transport network of a universal postal service provider has a hierarchical structure, given by historical development and determined by technological processes. Therefore, it is necessary to decide at what level to perform the optimization. This is closely related to the selection of appropriate algorithms for solving optimization problems at a given network level. After such decision, the process is the following:

1. constructing the transport network graph;
2. finding the matrix of minimum distances from each node to all other nodes;
3. finding the shortest closed path passing through all nodes in the graph.

Construction of the transport network graph

The first step in this type of task is to abstract the transport network into a graph. Graph G is arranged with two sets $G = (V, E)$; where V is a non-empty finite set of vertices/nodes (the regional postal processing centers), and E is a non-empty finite set of edges (communication links).

Shortest paths

Finding the shortest connection—*Walk* (in graph theory) is one of tasks to be addressed in the optimization of the mail transport process. This task is taken into account when dealing with distribution routes, picking up mail from mailboxes, delivery of telegrams, or running errands in districts. The basic starting point is the structure that includes distances from each vertex in the graph.

[1] Ing. Martin Macík, PhD. Candidate, Faculty of Operation and Economics of Transport and Communications, University of Žilina, Slovakia, martin.macik@fpedas.uniza.sk

[2] Ing. Jozef Štefunko, PhD. Candidate, Faculty of Operation and Economics of Transport and Communications, University of Žilina, Slovakia, jozef.stefunko@fpedas.uniza.sk

This structure is a matrix compiled of lengths of the shortest walks from each node to every other node in the graph (Palúch, 2001). To create a matrix of the shortest distances, we used a script that retrieves data from Google Maps and directly produces the desired matrix. However, this process has certain limitations, such as the allowed number of script requirements per day.

Eulerian path

These tasks are aimed at finding the shortest closed path, passing through all nodes of the graph. The most important of these tasks are distribution of the press, picking up mail from mailboxes, or creating distribution routes in transportation networks.

In practice, the entire task is usually difficult because solutions must satisfy various requirements, such as the latest possible arrivals to selected nodes, number of available vehicles, or need to find multiple simultaneous routes on the network. Therefore, the solution to large-scale networks is usually derived from the use of approximate methods that although does not guarantee finding the optimal solution, but the obtained solution is often satisfactory. The fundamental problem in many approximate methods is that it is not possible to determine how much the obtained solution differs from the optimal solution. We can subdivide used methods into two categories (Janáček, 2006):

- In terms of solution exactness:
    - exact – suboptimal and optimal,
    - heuristic – approximate.
- In terms of construction:
    - gradual construction of circuit,
    - improvement of existing solution,
    - elimination of non-optimal circuits.

Metaheuristic methods

Metaheuristic is the superset of heuristic solution. Greek prefix *meta* is used for marking the overlap of the root concept (heuristics) and represents something beyond on an upper level. Metaheuristic has various definitions in literature. Osman (1996) defines *metaheuristics* as "an iterative process that controls and changing operations of subordinate heuristics to efficiently create high-quality solutions. It can handle the complete (or incomplete) single solution or a whole set of solutions during each iteration. Subordinate heuristics can be high (or low) level procedures, simple local search, or construction methods."

The most common source of inspiration for invention of new metaheuristics is the living nature. The authors of algorithms refer to genetic patterns, behavior of ants, and evolutionary principles. The method of genetic algorithms, described in this article, also referred to such principles.

Genetic algorithm method

Obtaining solution using genetic algorithm (GA) is a relatively simple task. The biggest problem is suitable encoding of the individual subject, which in our case is the node of the graph. Initial experiments dealt with binary representation, but in practical terms, it is used for encoding chromosome in numbers, e.g. in the range from 0 to 9. The simplest method is the encoding method of individuals with natural representation of the path. For example, a chromosome is composed of numbers from 0 to 9, wherein each number represents the node, through which leads the path. Therefore, the *chromosome (6, 7, 4, 2, 5, 3)* represents the path *6 - 7 - 4 - 2 - 5 - 3 - 6*. This procedure establishes the whole population of individuals, which is a random solution of eulerian path by topology. The individuals are then valued by the path distance (weight) that every individual traveled. So, we have a population of very weak individuals with heavy weight (distance) and very strong individuals with light weight. Sequential crossover and mutation produce new individuals who should have heavier weight than the previous individuals with very light weight. It is possible that there is an individual in the very first generation with the shortest path in the entire topology (Hynek, 2008).

GA is an iterative process, which maintains a population with the generally fixed number of individuals or, in our case, nodes. In each iteration (called generation), the configuration of the current population is evaluated and based on this result; new population is created. When we define variable *S* as population of individuals, we can write the following procedure:

1) *Initialize S* – create a population of *k* initial individuals $S = \{S_1, \ldots, S_k\}$. The choice of individuals, which will form the initial population, is usually conducted randomly from addressed environment.
2) *Evaluate S* – assign every individual with weight representing the importance against all other individuals (fitness). This will increase the chance that weaker (inferior) individuals will not get to the next population and will be replaced by better individuals that will continue to spread their genes. Thus, this genetic operator simulates the mechanism of natural selection. The average value of the entire population fitness is also calculated in this step.
3) S*election of S* (if the result does not converge) – select from the population such individuals who will create a basis for the next generation. It is essential that the number of individuals with above average quality in each new generation was higher than in the previous one. There are various methods to select the individuals, e.g. roulette wheel method, tournament selection method, and probabilistic tournament.
4) *Recombination of S* - selected individuals are then used in recombination of S, which means to use the group selected in the previous step to create the following new population (the next generation). In this step, GA uses the techniques known as “the crossover” (an operation in which two or more selected individuals create a new individual – the characteristics of a new individual is simply a combination of the parental characteristics) and “the mutation” (which is the random variation in selected individuals).

After application of mutation operator on individuals, the old generation is destroyed and replaced with a new one. Then, we evaluate the new generation and repeat the steps of the iteration process. It is necessary to remember the best individual found yet. When the new generation is evaluated, and if a better individual is found, we replace him. This ensure the saving of the best solution found yet, in case that all the following solutions would only be worse. We have to decide when we assume that *S* can no longer be improved, and thus when we terminate the algorithm. There are at least three possibilities in this regard. One is to make a large number of iterations, ensuring the highest quality grouping of individuals. In this case, it is necessary to decide in advance what will be the number of conducted generations. The disadvantage is that more generations must be created than necessary. The second way is to decide on the process termination, based on some criteria resulting from the configuration of current population. In this case, the termination criteria have to be selected, and the computing time needed has to be added to each generation in order to verify whether this criterion is met or not. The third way is based on an observation that if, for instance, the last 10 generations did not result in an improvement, then we have to terminate the algorithm (Hynek, 2008).

Solving the multiple traveling salesmen problem using genetic algorithm

When solving multiple traveling salesmen problem (MTSP), it is possible to use the above described algorithm. There are specific restrictions for such optimization problem:

1. Each traveling salesman will return to the initial node of his path.
2. Each traveling salesman will visit a unique set of nodes.
3. Each node is visited by exact one traveling salesman, except the initial node.

Problem is defined by given *n > 1* nodes and *m* salesmen located at a single initial node (depot). The remaining nodes that are to be visited are called “intermediate nodes.” The MTSP consists of finding routes for *m* salesmen, who all start and end at the depot, while the traveled path is minimized. The *n* nodes in MTSP is partitioned into *m* routes, where each route corresponds to a single traveling salesman problem (TSP). The MTSP is more difficult compared to the TSP, since it is necessary to assign nodes to each traveling salesman as well as the optimal arrangement of nodes (Seidighpour & Darani, 2011).

Mathematical model

Let *G(V,E)* be a complete undirected connected graph with a vertex (node) set $V = \{0, 1, ..., n\}$ and an edge set $E = \{(i, j) : i, j \in V , i \neq j\}$. If the graph is not complete, the lack of any edge is replaced with an edge that has an infinite size. For presenting the integer linear programming model for MTSP, the following variables are introduced (Seidighpour & Darani, 2011):

n – number of nodes for each instance,

$x_{ij} = \begin{cases} 1 \\ 0 \end{cases}$ *m* – number of salesmen used for each instance,

C – distance matrix on graph *G;* firstly, symmetric; secondly, it is true in triangle inequality. It means that $c_{ij} = c_{ji}$ and $c_{ij} + c_{jk} \geq c_{ik,}$.

For each (*i, j ,k = 1, 2, ..., n*), if the salesman travels directly from *i* to *j* otherwise.

Hence, one of the common integer programming formulations for the MTSP can be written as follows:

$$\min = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \cdot x_{ij} \tag{1}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \text{ , } where\ \ j = 2, ..., n \tag{2}$$

$$\sum_{i=1}^{n} x_{ij} = m \text{ , } where\ \ j = 1 \tag{3}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \text{, } where\ i = 2, ..., n \tag{4}$$

$$\sum_{j=1}^{n} x_{ij} = m \text{, } where\ i = 1 \tag{5}$$

$$\sum_{i \in S} \sum_{j \in N-S} x_{ij} \geq 1(\phi \neq S \subset N = \{2,...,n\}, |S| \geq 2 \tag{6}$$

$$\sum_{i \in N-S} \sum_{j \in S} x_{ij} \geq 1(\phi \neq S \subset N = \{2,...,n\}, |S| \geq 2 \tag{7}$$

$$x_{ij} \in \{0,1\} \tag{8}$$

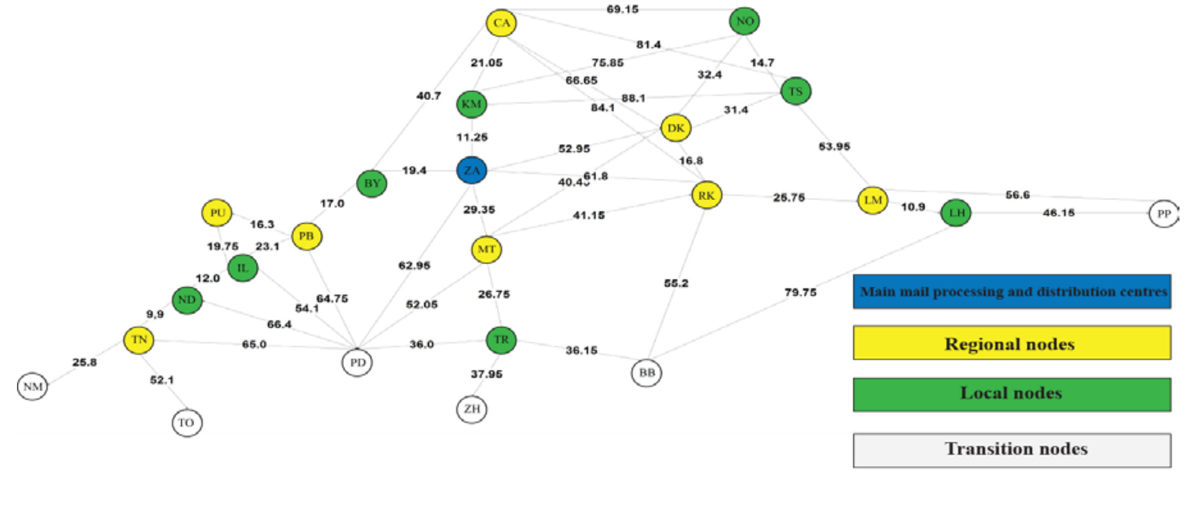
The objective function (1) minimizes the total distance traveled in one route. Constraint sets (2) and (3) ensure that the salesmen arrive once at each node and *m* times at the depot. Constraint sets (4) and

(5) ensure that the salesmen leave each node once and the depot *m* times. Constraint sets (6) and (7) are to avoid the presence of sub-routes for each salesman. Finally, constraint set (8) defines binary conditions on the variables (Seidighpour & Darani, 2011).

Solving MTSP in the region covered by main processing and distribution center in Žilina

The aim of the research was to test the use of GA in optimizing the transport network in the postal environment. The task is specified for the regional postal transport network subordinated to the main mail processing and distribution center in Žilina. This subset of Slovak postal network was chosen because this area is currently undergoing major changes in network construction, and there are also planned changes for transport routes. Abstraction of selected network to a graph is shown in Figure 1.

Figure 1: Graph of current regional postal transport network in Žilina



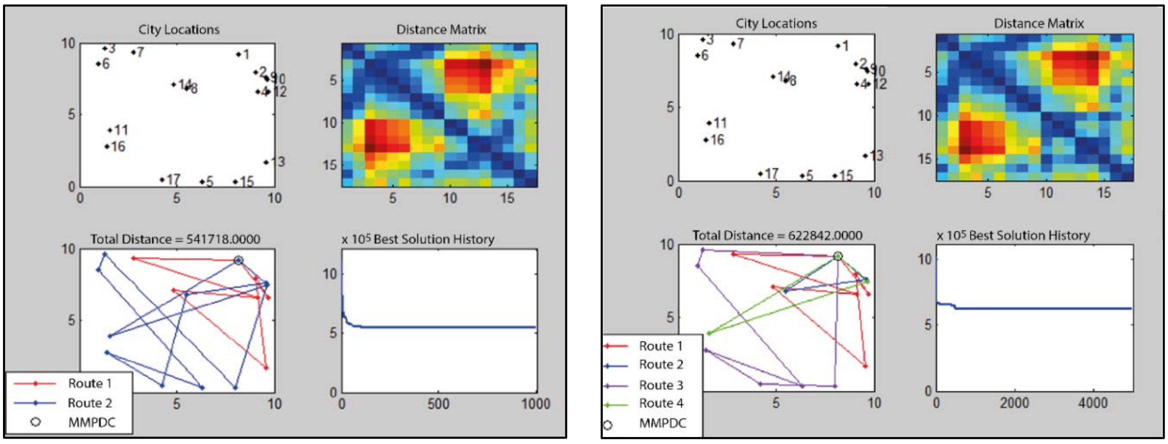
Source: Authors

The initial distance matrix is obtained by using a script that utilizes Google Maps and application Distance Matrix API. The nodes are exact geographical locations of regional and local processing centers. The solution requires the transfer of distances into 2-D coordinates using MATLAB functions. Therefore, it is necessary to adjust the matrix values to make the matrix symmetrical. The simulation is based on iteration to obtain the optimal solution, where the optimum is the sum of distances traveled by all traveling salesmen reduced to a minimum. The number of iterations is fixed prior the specified simulation, and approximation to the optimal solution (in terms of distance) can be observed on the graph. The problem is simulated for the cases of two and four traveling salesmen. The number of traveling salesmen is the only input parameter that changes.

Evaluation of simulation

The goal of the simulation was to test the possibility of optimizing postal routes by using genetic algorithm for solving the multiple traveling salesman problem and to provide a review of applicability for postal operators. Graphical and mathematical simulation outputs for two and four traveling salesmen for solved region are shown in Figure 2. The results provide a sufficient overview of the program functionality in the field of transport networks. Time demands on calculation increase radically with the difficulty of obtaining a solution, which can be observed in the increasing number of iterations when adding traveling salesmen (Table 1). However, difficulty also drastically increases with growing number of nodes. But, such simulation is not addressed in the article.

Figure 2: Simulation output of solution for two and four traveling salesmen



Source: Authors

Simulation reduces the total distance traveled for a set number of traveling salesmen to the minimum. This solution is theoretically correct because the reduction of the minimum distance should reduce costs. However, in practice, this is not an ideal solution because it would be economically correct to divide the distance traveled between traveling salesmen equally to achieve maximum usability of the vehicle fleet. The entire task is usually difficult in practice because solutions must satisfy various requirements, such as the latest possible arrivals to selected nodes or number and capacity of available vehicles.

Research on optimization of postal transport routes will continue using Clark-Wright's method, used for calculating eulerian path. Such method is based on the initial inefficient solutions, where each route will cover demands for one node. This method can easily meet other global requirements, such as the maximum length of the route, duration of travel, number of visited nodes, and other similar conditions where the controlled variable of the new travel depends only on the corresponding variables of connecting travels (the connected nodes).

Table 1: Simulation results

	Two traveling salesmen	**Four traveling salesmen**
Route 1 [km]	185.666	261.035
Route 2 [km]	356.052	113.475
Route 3 [km]	-	186.864
Route 4 [km]	-	61.468
Total distance [km]	**541.718**	**622.842**
Number of iterations	**88**	**876**

Source: Authors

Acknowledgement

This contribution was undertaken, as part of the research project KEGA 030ŽU-4/2015 Transformation of education in the field of postal technologies, networks and services.

References

Čorejová, T., Achimský, K., Fitzová, M., & Kajánek, B. (1995). *Projektovanie sietí v pošte I.* [Designing networks in postal industry I.]. Žilina, Slovakia: Edičné stredisko VŠDS. 101-143

Hynek, J. (2008). *Genetické algoritmy a genetické programovaní* [Genetic algorithms and genetic programing]. (Vol. 1). Praha, Czech Republic.

Janáček, J. (2006). Optimalizace na dopravních sítích [Optimization of transport networks]. Žilina, Slovakia: EDIS.

Osman, I. (1996). Meta-heuristics: Theory & applications. Boston: Kluwer Academic.

Palúch, S. (2001). *Teória grafov* [Graph theory]. Žilina, Slovakia: EDIS. 18-39

Seidighpour, M., & Darani, M. N. (2011). An Effective Genetic Algorithm for Solving the Multiple Traveling Salesman Problem. *Journal of Optimization in Industrial Engineering,* 8, 73-79.