Research Paper Logistics

Application of Genetic Algorithm to The TSP Problem

Ing. Alena Rybickova CTU in Prague,Faculty of Transportation Sciences, Department of Logistics and Management of Transport Department, Konviktská 20, 110 00 Praha 1, Czech Republic

Ing. Denisa Mockova CTU in Prague,Faculty of Transportation Sciences, Department of Logistics and Management of Transport Department, Konviktská 20, 110 00 Praha 1, Czech Republic

Ing.Bc. Adela Karaskova CTU in Prague,Faculty of Transportation Sciences, Department of Logistics and Management of Transport Department, Konviktská 20, 110 00 Praha 1, Czech Republic

ABSTRACT

The article concerns the proposal of the solution of genetic operators of a genetic algorithm for calculation of a transportation task in the travelling salesman problem. Travelling salesman problem belongs to NP-hard problems with significant use in transportation. Application of genetic algorithm to a specific problem includes mainly design of encoding and three operators of selection, crossover and mutation. For all these operators several variants were proposed with regard to travelling salesman problem specifics as well as some other adjustments to the problem.

KEYWORDS TSP problem, genetic algorithm, genetics operators

INTRODUCTION

One of the basic transportation problems is the travelling salesman problem (TSP). The aim of this problem is to find the minimum Hamiltonian cycle in the graph, i.e. to go through all the vertices of the graph just once, with an exception of the initial vertex that is also the terminal vertex, and in the shortest possible way. In this problem the set of all the admissible solutions is so extensive that it is impossible to find an optimal solution within an acceptable time horizon. TSP is an NP-hard problem. For this reason methods, known as metaheuristics, have been developed that operate on the principles of combinatorial optimisation and seek solutions in the limited area of all admissible solutions. An advantage of metaheuristics is the calculation speed; while on the other hand, a disadvantage is that they do not guarantee finding an optimal solution.

There are many types of metaheuristics, the most frequently used and examined are genetic algorithms (GA) [1], due to their relative simplicity, and which we will use in solving the given problem. The problem will be appropriately encoded and variant genetic operators (GO) will be proposed.

METHODS OF SOLVING THE PROBLEM

The problem can be solved by various methods, i.e. exactly, heuristically or by way of a metaheuristic.

- The brute-force approach (permutation solution) – at the end of this approach we will gain the real optimal solution, however, with an increasing number of vertices this approach becomes very time consuming (Table 1).
- Little's algorithm, based on the principle of the Branch and Bound Approach, providing an optimal solution [2].
- Integer linear programming – namely binary programming. [3].
- Heuristic and metaheuristic methods – these methods in general need not provide an optimal solution to the problem, but they do provide a solution within a reasonable time. They include methods intended only for a specific problem as well as more generally applicable schemes. An example ofheuristic methods specific for a travelling salesman problem is iterative improvement methods [4].

TABLE – 1 EXAMPLES OF THE NUMBER OF POSSIBLE ROUTES FOR A VARIOUS NUMBER OF GRAPH VERTICES

Number of vertices in the graph	Number of possible routes	Number of vertices in the graph	Number of possible routes
3	2	40	$2.03.10^{46}$
10	362,880	72	$8.5.10^{101}$
13	479 001 600	80	$8.94.10^{116}$
20	$1.21.10^{17}$	100	$9.22.10^{155}$

Source: author

GENETIC ALGORITHM

The genetic algorithm is one of the metaheuristic methods. It is an evolutionary algorithm, i.e. it takes advantage of procedures similar to those used in the field of genetics. The algorithm works with a population of individuals whose genetic information is passed onto the offsprings. In solving a problem the algorithm works with encoded strings (encoding) and with the quality of these strings (fitness functions). The whole algorithm consists of only three operators – reproduction, crossover, mutation. Each of these operators uses random selection. The whole algorithm works on the basis of creating, out of one generation, a new generation using these three operators. After many repetitions (e.g. 10 000 times) of creating an improved generation, the best string found is chosen as a solution to the problem. An improvement, such as elitism, may be also added; which will ensure that the best value found, will never be missed. The first to use the GA for solving a TSP was Brady [5], to be followed by Grefenstette et al. [6].

DESIGN OF THE GENETIC ALGORITHM FOR THE TSP PROBLEM

Encoding of the individual

We consider a complete graph. The individual was encoded by the permutation code [7] of a length of n, where n represents the number of vertices in the graph. Each gene thus represents a vertex we wish to visit at the given moment, and the whole individual thus represents the full path through the graph.

Fitness function

$$f(x) = \sum o(v_i, v_j) + o(v_k, v_p) \qquad (1)$$

where:

$o(v_i, v_j)$ ..... assessment of the edge between two vertices

$o(v_k, v_p)$ ..... assessment of the edge between the terminal and initial vertex

Selection

For this problem a tournament selection was chosen. Two types of such selection were chosen.

Binary - from the previous generation two individuals are randomly selected and of these two the one with a lower value of fitness function progresses on.

Tertiary – three individuals are randomly selected of which the best one (the one with the lowest value of fitness function) goes on in the algorithm.

After the selection we always obtain a set of two individuals, which progress into crossover and mutation. The already selected individuals are also available for further selection. This means that the set of two individuals can figure with the identical individuals. The reason for this is that we wish to carry on developing individuals from the best possible set of parents.

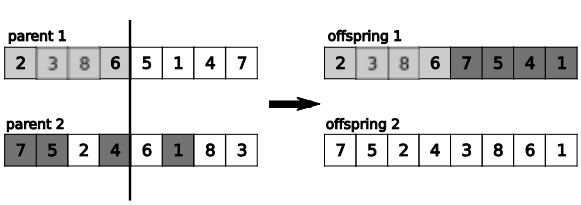
Crossover

From the previous step (selection) we get a set of two individuals that will be passed onto the crossover. The crossover will take place depending on the crossover probability, alternatively chosen in the programme in the interval of 0.7 – 0.9 (designated as P_C). If the individuals succeed in the crossover, they can crossover in three approaches, where each of these crossover approaches is supplemented with two variants of copying a part of the genetic information of parents (copying of the left or right side) and with two variants of supplements of an individual from the right or left of the remaining part of the genetic information of a parent. This way we get 4 crossover possibilities for each variant. Only one of the 4 possibilities will be shown in the respective picture. Individual crossover variants are based on the order crossover mechanism [8].

One-point crossover – Variant 1

The position where the individuals will cross over is chosen randomly, i.e. the offspring of these parents will exchange all the genes present prior to this position. This step is followed by supplementing of the offspring to become fully valid individuals, as shown in Fig. 1.

FIGURE - 1 EXAMPLE OF ONE-POINT CROSSOVER – VARIANT 1

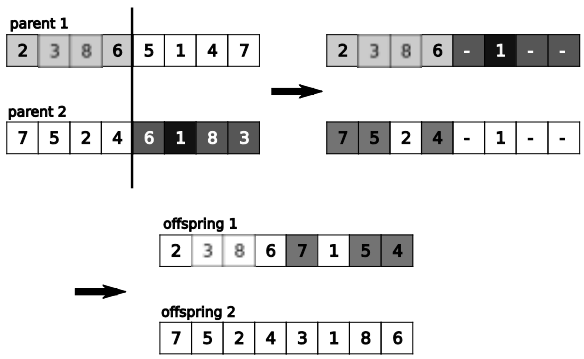


Source: author

One-point crossover – Variant 2

The position where the individuals will cross over is chosen randomly, i.e. the offspring of these parents will exchange all the genes present prior to this position. This step is followed by supplementing of the offspring to become fully valid individuals, as shown in Fig. 2.

FIGURE - 2 EXAMPLE OF ONE-POINT CROSSOVER – VARIANT 2

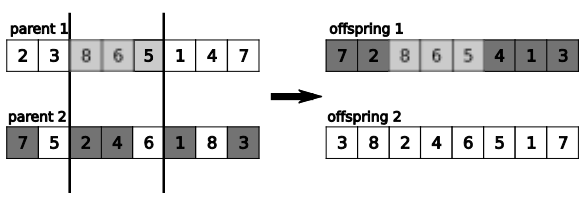


Source: author

Two-point crossover

Two positions between which the individuals will cross over are chosen randomly, i.e. the offspring of these parents will exchange all the genes present between these two positions. This step is followed by supplementing of the offspring to become fully valid individuals, as shown in Fig. 3.

FIGURE - 3 EXAMPLE OF TWO-POINT CROSSOVER –



Source: author

Solution variants:

- one-point crossover – variant 1 without elitism
- one-point crossover – variant 1 with elitism
- one-point crossover – variant 2 without elitism
- one-point crossover – variant 2 with elitism
- two-point crossover without elitism
- two-point crossover with elitism

Mutation

Another step in the cycle is mutation. This works with each particular individual from the set of individuals. Based on the mutation probability, chosen in the programme alternatively within the interval of 0.001 – 0.2 (referred to as the P_M) date individual will get mutated. Two types of mutation have been chosen.

Swapping two genes – this concerns the same procedure as above in swapping one gene, but this operation is performed twice in a row.

Elitism

The GA was supplemented with a calculation without elitism or a calculation with x-degree elitism. This concerns the quality that determines how many of the best individuals from the past generation are to pass on to the next generation without any changes, i.e. without selection, crossover or mutation. Such individuals are, though, still taken into consideration in the selection, so their crossed or mutated individuals have the chance to get into the next generation.

Calculation without elitism – in the transitions from one generation to another all the individuals arise by way of genetic operators; none of them is automatically copied without changes. Due to this we may lose the best individual.

Calculation with x-degree elitism - "x" here stands for the number of the best individuals (differing from one another) that will automatically be copied into the further generation without changes. In this way we can be sure that no degeneration will take place in the next generation. It does not though hold true that the greater degree of elitism the better the result attained. Therefore, the degree of elitism must be chosen with care, ideally within the range of 1-2.

Control of individuals in the set
Mutation is followed by a control check of individuals. The latter resolves the situation where any of the individuals would equal one of the individuals already assigned to the new generation. An equal individual means also an individual with an identical route, but a different initial vertex. Our requirement is that the new generation does not contain any such duplicities, therefore such individuals are discarded and another set of individuals is calculated until the new generation is filled with unique individuals.

CONCLUSIONS

Basic objective of this paper was the design of operators for a genetic algorithm for the TSP. The operators then had to be adapted to the quality of the problem.

Prior to the actual design of the GA and of its GOs it was necessary to choose the correct encoding of the individuals so that they are able to carry the necessary information (the graph's route) and so that the algorithm could work easily with it. Therefore we decided to use encoding with the aid of a permutation code, which is ideally suited to our requirement.

The selection's operator chose the tournament variants (due to their fast implementation), namely in two modifications – binary and tertiary. During the selection identical and similar individuals are controlled and removed.

The crossover's operators chose two types of crossover – one-point and two-point. With the one-point crossover, two variants were created. In total we thus use three crossover operators.

This design will be further used for testing and comparison of proposed operators.

ACKNOWLEDGEMENTS

This research was supported "Application of Exact and Metaheuristic Methods to VRP in Transportation", (FIS: 161-1611404F000).

REFERENCES

| [1] Holland JH. Adaptations in Natural and Artificial Systems. Ann Arbor, MI: University of Michigan Press; 1975. | [2] Little JDC, Murty KG, Sweeney DW, and Karel C. An algorithm for the travelling salesman problem. Operations Research 11, pp. 972-989. 1963. | [3] Dantzig GB. Application of the Simplex Method to a Transportation Problem, Activity Analysis of Production and Allocation. New York: T. C. Koopmans, Ed , Wiley; 1951. | [4] Karg RL and Thompson GL.A heuristic approach to solving travelling salesman problems. Management Science 10, pp. 225-248. 1964. | [5] Brady, R.M. Optimization Strategies Gleaned from Biological Evolution. Nature 317, pp. 804. 1985. | [6] Grefenstette J, Gopal R, Rosmaita R, Van Gucht D. Genetic Algorithms for the TSP. Proceedings of the First International Conference on Genetic Algorithms and Their Applications. NJ: Lawrence Erlbaum Associates, Hillsdale; 1985. | [7] Hynek J. Genetickéalgoritmy a geneticképrogramování. pp. 185.Praha: Grada Publishing; 2008. | [8] Davis L. Applying algorithms to epistatic domains, Proc. Int. Joint Conf. on Artifical Intelligence, pp. 162–164. 1985. |