UDC 621.382

Bazylevych R., Kutelmakh R., Tomchuk A.

Lviv Polytechnic National University,

Software department

SOLVING LARGE-SCALE TRAVELING SALESMAN PROBLEM BY THE "COMMON EDGES" METHOD

Existing heuristic algorithms for solving traveling salesman problem, such as Nearest Neighbor, 2-Opt, 3-Opt, Lin-Kernighan and LKH have been investigated in this work. Mentioned algorithms have been compared in terms of calculation time and quality of found solution. Decomposition approach, based on using common edges in multiple solutions, has been proposed.

Keywords — traveling salesman problem, NN, 2-Opt, 3-Opt, Lin-Kernighan, LKH

Introduction

The problem of constructing optimal routes through a given set of points in the plane or in space arises in many areas of human activity. For example, this problem can come in solving problems of planning and logistics; in the manufacture of printed circuit boards; while minimizing movements in robotics; in the analysis of the structure of DNA, etc. In essence, all these areas close to solving the traveling salesman problem. There is a great interest in the problem associated with using it as a platform to study general methods of combinatorial optimization [1]. Relatively new practical application of the traveling salesman problem is continuous line drawing. Robert Bosch and Adrian Herman in 2003 suggested the use of a geometric interpretation of the traveling salesman problem, which means image generation using continuous line [2].

There are many algorithms for solving the traveling salesman problem, but most of them are not applicable to the large scale problems, as the complexity of these algorithms increases exponentially. So, we often have to choose between time of the algorithm and the quality of the solution, a compromise is usually found by combining methods of finding the initial solution and consistent optimization [3].

Problem

Traveling salesman problem is formulated as follows: given a set of cities and the distance between all possible pairs of cities. Need to find a path that passes through all cities, and returns to the original. The total length of the traversed path should be minimal. The problem can be represented as a graph model. There are different versions of the problem, the most common are symmetric and asymmetric problems.

In the case of symmetric problems all pairs of edges between the same vertices have the same weight, i.e., weights of edges $d_{ij}$and d_{ji} are the same. As a result, all the routes have the same length in both directions. In the symmetric case, the number of possible routes twice less than in the asymmetric case. Symmetric problem is modeled as undirected graph [1].

In this paper the symmetric traveling salesman problem is investigated, where $V = v_1,, v_n$- set points, $A = (r, s): r, s \in V$ - the set of edges, where $d_{rs} = d_{sr}$ is the cost (weight), associated with an edge. To solve the traveling salesman problem, it is needed to find the minimal closed path passing through all vertices only once [4]. Points $v_i \in V$ are coordinates (x_i, y_j) in two-dimensional space and the distance between any pair of points is represented as Euclidean distance between r and s. Often this problem is called Euclidean traveling salesman problem.

Review of existing algorithms

All existing methods for solving the traveling salesman problem can be divided into two groups:

- methods for finding optimal routes;
- approximate methods [5].

Nowadays, there are many well-studied algorithms for finding optimal solution of the traveling salesman problem, in particular, such as branch and bound or cutting plane methods, progressive methods based on linear programming techniques. All of them are not applicable to large-scale problems, because computational complexity of these algorithms increases exponentially.

One of the simplest heuristic algorithms is the nearest neighbor algorithm. The algorithm can be described as a set of steps:

1. Choose an arbitrary starting point.
2. Find the nearest unvisited point and go for it.
3. If there are some unvisited points, go to step 2.
4. Return to the starting point.

Computational complexity of the algorithm is equal $O(n^2)$. The quality of the obtained solution may be worse than the optimal by 25-30%, but the solution will be obtained with minimal time-cost [6]. The widest use of such algorithm is for generating the initial solution for further optimization of route by 2-Opt and 3-Opt (this family of algorithms is often called the k-Opt algorithm) [Fig. 1].

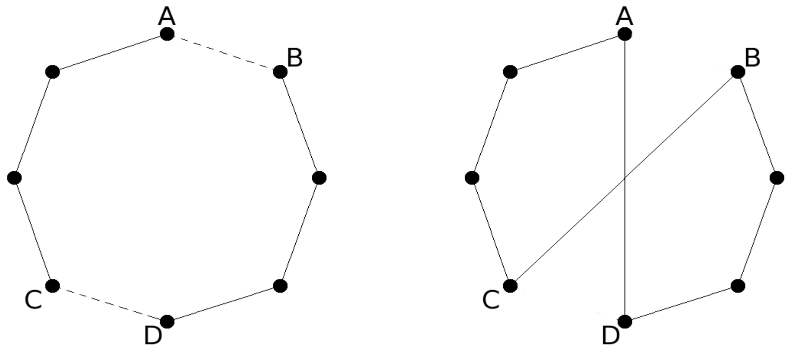


Fig. 1. Changing edges by using 2-Opt algorithm

2-Opt algorithm removes the two edges of the existing path (obtained using some constructive algorithm) and replaces them by two new ones, shorter in total. This action is repeated for each pair of edges. It should be noted that replacing these two edges can be only one way, as shown in Fig. 1, otherwise it will form two cycles (Fig. 2). Computational complexity of the algorithm is $O(n^2)$.

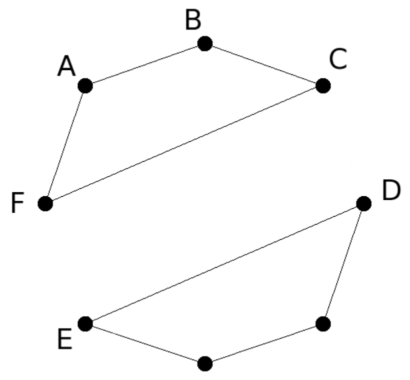


Fig. 2. Formation of loops in the process of algorithm 2-Opt

The resulting optimization solution is approximately 5-15% worse than optimal. Similarly, by using 3-Opt algorithm, 3 edges are replaced. In this case two options are possible, as shown on Fig. 3 In practice, this algorithm is often implemented as two or three sequential applications of 2-Opt. The result is worse than the optimal by 3-4%. Computational complexity of the algorithm is $O(n^3)$.

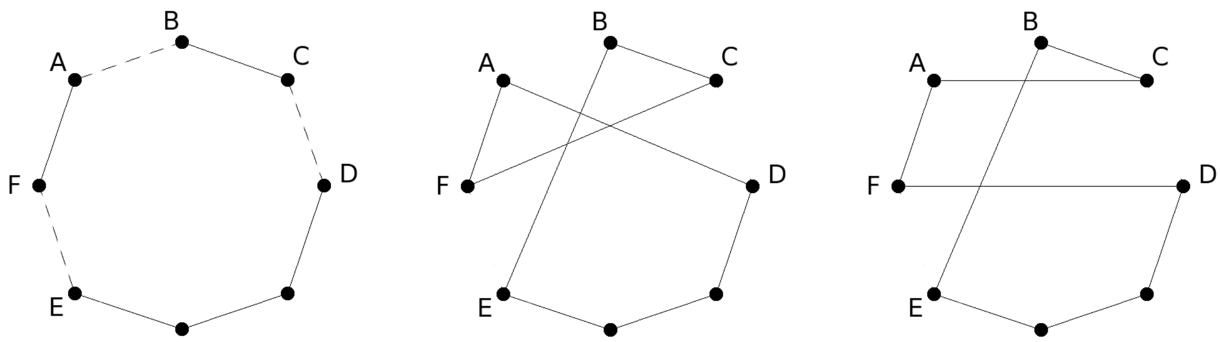


Fig. 3. Possible permutations using 3-Opt algorithm

The logical extension of the described methods is to use route optimization with "bigger *k*", i.e,. 4, 5 or more. With an increasing number of removed edges, increases the time of the algorithm and makes it unacceptable for large-scale problems. Studies of such optimization led to the creation Lin-Kernighan algorithm. This method involves the use of an initial suboptimal routes and it optimization. Number of edges (number *k*), which will be replaced at each step of the algorithm changes dynamically.

The feature of this algorithm is that it ensures that the solution for the problem of dimension of *N* points will be no 4 times longer than the optimum, and is never worse than the original route. The method proposed in 1973 by Lin and Kernighan. Most aspects of the implementation described in the original work.

Computational complexity of the method is O ($n^{2,2}$). The result usually worse than optimal by 1-2%. Nowadays, many modifications of Lin-Kernighan algorithm are designed: multilevel Lin- Kernighan [8], chained Lin-Kernighan [9], a parallel Lin-Kernighan [10], but the most effective modification is Lin-Kernighan-Helsgaun algorithm, which improves the rules of selecting the edges to be replaced [7].

Experimental results

In order to compare the described algorithms, experiments were carried out. The test data included problem instances with dimension of 100, 500, 1000, 5000, 10000, 50000, 100000 points. As a source, the well-known Mona Lisa test problem was used and its data was modified to generate test instances. The testing process consisted of sequential processing of test data using each of the described algorithms, as a result received time of the algorithm for the corresponding set of input data and solution of the problem. Research conducted on PC with Intel Celeron M220 CPU. Figure 4 shows the results of quality comparison of solutions obtained with the algorithm LKH. As a benchmark was chosen LKH, as it is the closest to the optimum. 3-Opt algorithm provides a worse than the best solution within 10%, depending on the dimension of the problem. The result of 2-Opt algorithm is worse than the best within 15%, and the worst result was obtained using the nearest neighbor algorithm, which is worse than LKH within 30%. Figure 4 and Table 1 shows the results.

Table 1

Running time for nearest neighbor, 2-opt, 3-opt and LKH algorithms for

problems with the dimension of 10000-100000 points (seconds)

Dimension	**10000**	**50000**	**100000**
NN	0,01	0,09	0,2
2-opt	0,04	0,37	1,3
3-opt	0,09	0,72	2,5
LKH	15,6	90,3	237,3

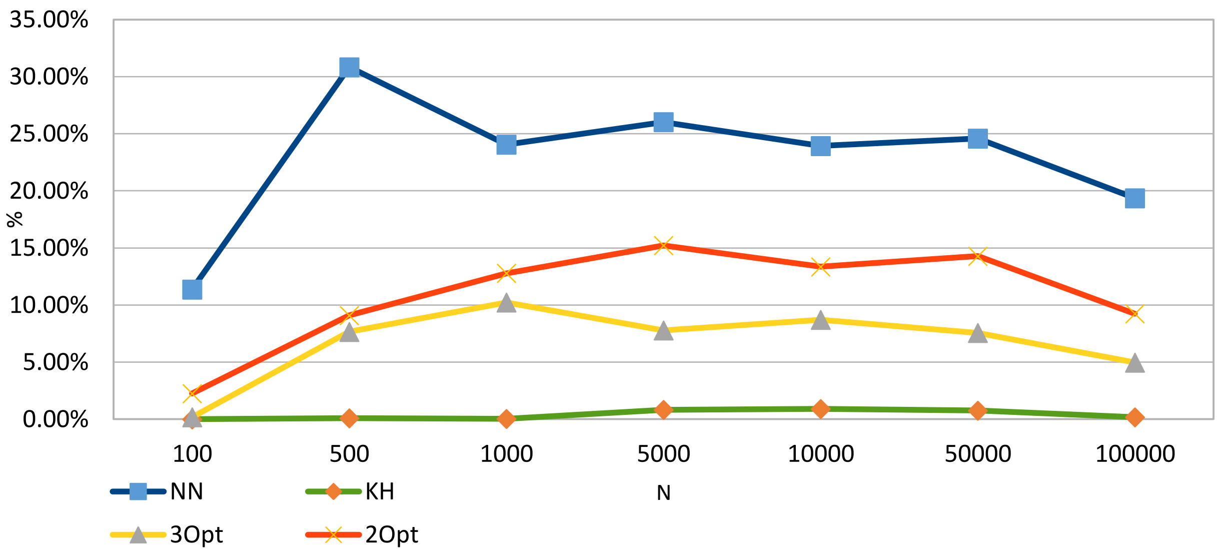


Figure 4. Comparison of solution quality obtained by LKH algorithm with solutions, obtained by other algorithms

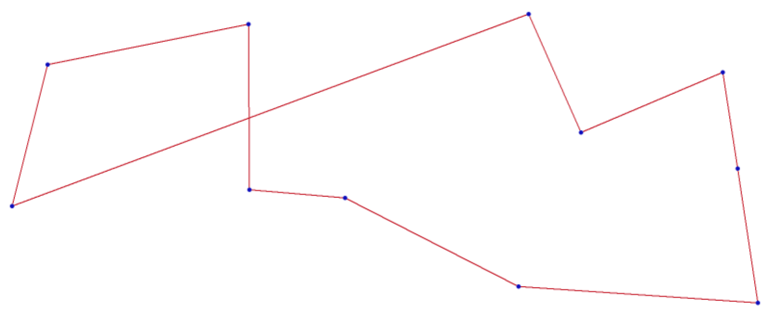
Improved approach

By comparing the performance and quality of the solutions we can conclude significant time and computing needs in finding solutions as close to optimal. It is therefore advisable to consider alternative approaches to solving problems. In particular, the proposed idea is to find a solution to the problem by combining the solutions obtained using algorithms with less computational complexity. The steps of the algorithm:

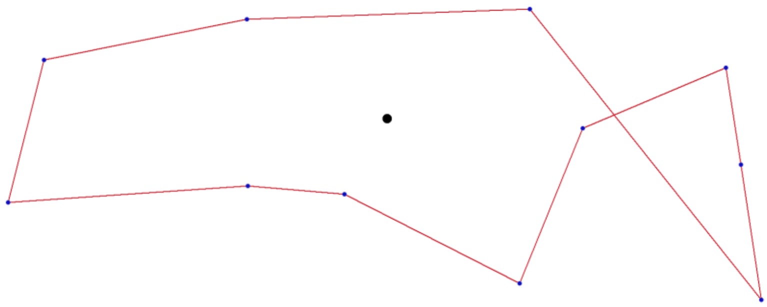
1. Selecting the set P consisting N points.
2. Solving traveling salesman problem with initial point belonging to the set P. As a result, N solutions S_i, where $i \in 1..N$, are obtained with different starting points (fig. 7).
3. Selecting edges from N routes, which are common to all of the routes. (fig. 8).
4. Merging all the edges into the whole solution of the problem.

Let's consider how the proposed approach works on example (fig. 5). Among the available set of points select five random, and use them as the initial point for finding the solution. The result is obtained in few ways (Fig. 7).

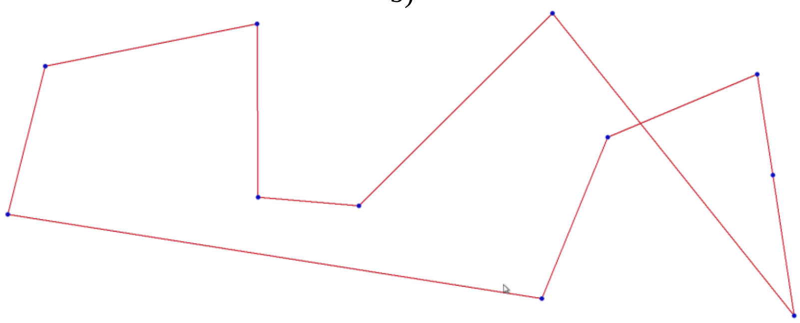
Figure 5. Initial point set



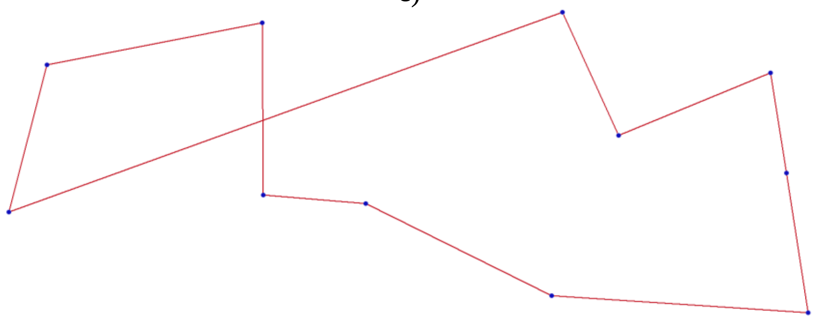
a)



b)



c)



d)

Fig. 6. Results of selecting different starting points

As a result of selecting only common edges, which are in all solutions we can form set of them (fig. 7). The next step is the calculating the nearest distances between the edges or route segments and forming the solution of the whole problem. (fig. 8).

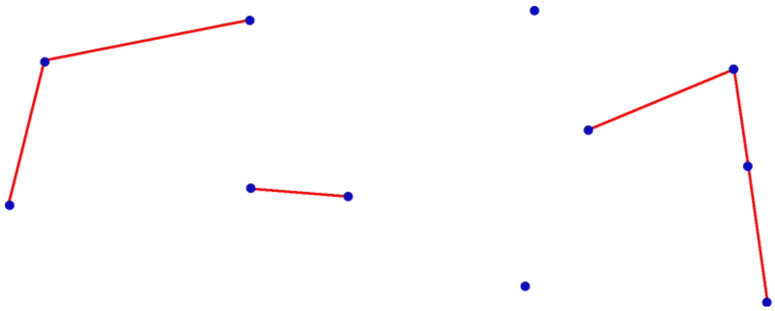


Fig. 7. Common edges

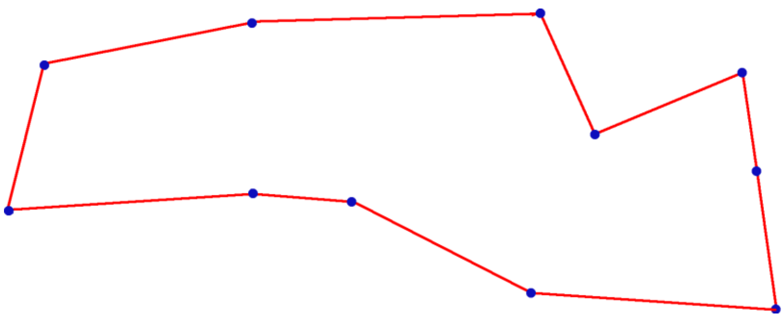


Fig. 8. Solution of the whole problem

The proposed method was investigated with the 264-point problem instance. As initial, 6 different routes, found by 2-opt algorithm were used (the length of each of them was 13% worse in comparison with optimal solution). Also, in order to compare with existing methods, the problem was solved with nearest neighbor algorithms and LKH. The results are shown in the table 2.

Table 2

The length of the 264-point problem solution, obtained by the "common edges" method and some existing methods in comparison with the optimal

Nearest neighbor algorithm	**2-opt**	***"Common edges" algorithm***	**LKH**
28.7%	13%	6.6%	Optimal

The solution, obtained with the "common edges" method (with using 2-opt algorithm for merging parts of the route) is 6,6% worse than optimal. Figure 9 shows common edges and solution of the 264-point problem, obtained with the "common edges" method.

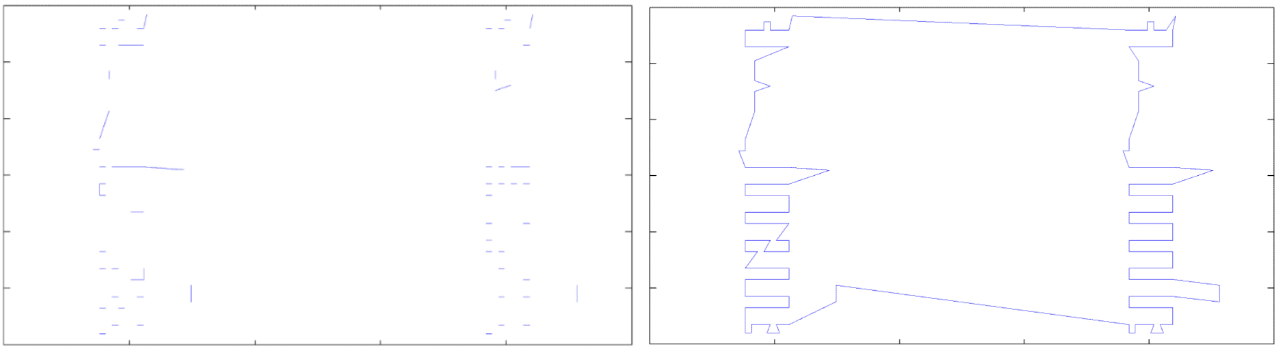


Fig. 9. Common edges and solution of the problem with dimension of 264 points

The proposed method can be applied for solving large scale problems, for example in finding partial solutions in local areas (in extending partial solution method, fig. 10).

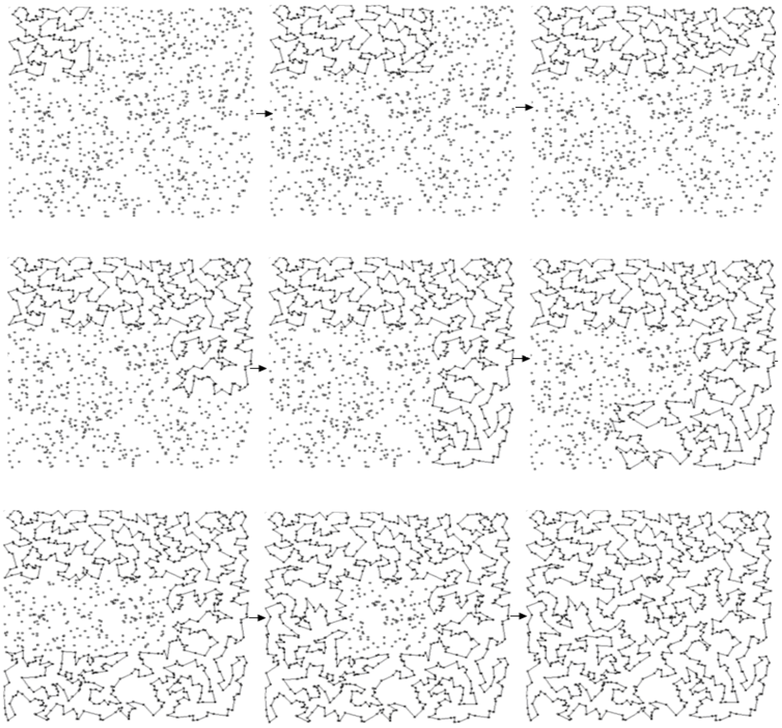


Fig. 10. Finding the whole solution of the problem by extending partial solution method

Conclusions

As a result, the investigations of existing nearest neighbor, 2-Opt, 3-Opt, Lin-Kernighan and Lin-Kernighan-Helsgaun algorithms were performed. Analysis of the results leads to the assumption that the optimal choice of algorithm to solve the traveling salesman problem depends on the requirements for the quality of the solution and running time. Where appropriate solution need to find quickly, you should use constructive algorithms, such as NN. On the other hand, if you want to find the most accurate, the closest to the optimal solution, it is feasible to use the Lin-Kernighan algorithm or its improved version, LKH. The results also show that the algorithms that yield results close to the optimal require significant time and computational cost. They are “hard to parallelize” also. The decomposition method, based on “common edges” was proposed. The method can be applied to large scale problems. Experimental results show that the solution , obtained using the proposed algorithm, is worse than optimal by 6.6%. Method improvement and research will be our further work.

1. R. Matai, S.P. Singh, M.L. Mittal, “Traveling Salesman Problem: An Overview of Applications, Formulations, and Solution Approaches ”, Jaipur, India, 2010, 2. 2. R. Bosch and A. Herman, “Continuous line drawings via the traveling salesman problem,” Operations Research Letters 3 (2004) 302-303, 3. P. Базилевич, Р. Кутельмах // Комп'ютерні науки та інформаційні технології [Текст] : [зб. наук. пр.] / відп. ред. Ю. М. Рашкевич. - Л. : Видавництво Національного університету "Львівська політехніка", 2009. - 279 с. : іл. - (Вісник / Національний університет "Львівська політехніка" ; № 650). - С. 235-244, 4.E. Nood and J. Been, An Efficient Transformation of the Generalized Travaling Salesman Problem, October 1991, 5. D.S. Johnson and L.A. McGeoch, “The Traveling Salesman Problem: A Case Study in Local Optimization”, November 20, 1995., 6. Christian Nilsson , Heuristics for the Traveling Salesman Problem, Linkoping Univercity, 7. Keld Helsgaun , General k-opt submoves for the Lin–Kernighan TSP veling heuristic Keld Helsgaun, 1 July 2009, 8. Chris Walsaw, A multilevel lin-kernigan-helsgaun algorithm for the treveling salesman problem, 9. D.Applegate, W.Cook, A.Rohe, Chained Lin-Kernigan for large traveling salesman problems, Algorithms and Optimization Departament, AT&T Labs-Research 10.Verhoeven, M.G.A. & Aarts, E.H.L. A parallel Lin-Kernighan algorithm for the traveling salesman problem, 1994.