

Transformation of the Mixed Chinese Postman Problem in multigraph into the Asymmetric Travelling Salesman Problem

Maria K. Gordenko, Sergey M. Avdoshin

Abstract— The Mixed Chinese Postman Problem (MCPP) is to find a minimum shortest tour of given graph or multigraph traversed each edge or arc at least once. The problem is NP-hard. However, mixed case of the problem has many potentially useful applications, including delivering of something, robot exploration, web site usability, etc. In this article, we propose to solve the problem using the graph transformation and solving well-known Asymmetric Travelling Salesman Problem (ATSP). The algorithm for transforming the MCPP in multigraph into ATSP is pointed out.

Keywords— Mixed Chinese Postman Problem, Arc Routing Problem, graph transformation, Traveling Salesman Problem.

I. INTRODUCTION

In the Mixed Chinese Postman Problem in multigraph, we are given a set of vertices (cities) and arcs and edges, that connect the vertices and define non-negative costs between them. The problem was originally studied by the Chinese mathematician Kwan Mei-Ko in 1962 [1]. A problem is called the Chinese Postman Problem (CPP) after him [2].

Apart from the traditional application of the CPP to solving the routing problems such as path planning of snowplows or serving teams, there is a wide range of applications including robot exploration, testing web site usability and finding broken links [3].

There are many various classifications of CPP. Let us consider some of them.

A. Classification based on graph type

Depending on the graph type, the CPP can be following:

- *Undirected Chinese Postman Problem (UCPP, UPP)*. The problem is defined on undirected graph $G = \langle V, E, C \rangle$, where V is the set of vertices, E is the set of edges, $C: E \rightarrow R_+$ – cost function giving non-negative weights of arcs and edges between vertices. Each edge can be traversed in both directions. The problem can be solved exactly in polynomial time. Notice that between two vertices can be defined more than one edge. In this case, the

multigraph

$G = \langle V, \tilde{E}, C \rangle$ is given, where V is the set of multigraph's vertices, \tilde{E} is the *multiset* of edges [4].

- *Directed Chinese Postman Problem (DCPP, DPP)*. In this case, the directed graph $G = \langle V, A, C \rangle$ is given, where E is the set of edges. Each arc has a direction and can be traversed in one way. The problem also may be determined on multigraph, where between a single pair of vertices can be more than one arc. The exact polynomial solution was provided [3].
- *Mixed Chinese Postman Problem (MCP, MPP)*. The initial graph $G = \langle V, E \cup A, C \rangle$ consists of directed and undirected edges, simultaneously [5].

The MCPP is a simply-stated problem, which has many potentially useful applications, however, has no exact algorithms [6].

B. Classification based on tour type

According to type of tour, the CPP can be following:

- *Closed CPP*. The solution tour should be closed (start and end vertex are equal). It means that postman should begin and end at the same point [3].
- *Open CPP*. The postman should not return to the starting point [3].

C. Classification associated with arcs and edges

There are many modifications of CPP based on arcs and/or edges' parameters variation. Below some of them are presented.

- *Windy Chinese Postman Problem*. The cost between vertices v_i and v_j depends on traverse direction ($c_{ij} \neq c_{ji}$) [7].
- *Rural Chinese Postman Problem*. The set of edges E in undirected graph, the set of arcs A in directed graph or the set $|A \cup E|$ in mixed graph consists from two subsets: required and non-required. Required arcs/edges should appear in solution at least once, non-required edges can absent in tour solution [8].
- *The Hierarchical Postman Problem*. The set of edges E in undirected graph, the set of arcs A in directed graph or the set $|A \cup E|$ in mixed graph partitioned into clusters and on each clusters the priority of the traversing is determined [9].

Above the main classifications and typical representatives of the problem are represented. However, the list of variations of the problem is not exhaustive.

In this paper, we are talking about the MCPP in multigraph. Despite the fact that the problem has many applications, there is no exact solution. There are many

Manuscript received April 20, 2017.

Maria K. Gordenko is with the Department of Software Engineering, Faculty of Computer Science, National Research University Higher School of Economics, Moscow, Russia (mkgordenko@edu.hse.ru).

Sergey M. Avdoshin is with the Department of Software Engineering, Faculty of Computer Science, National Research University Higher School of Economics, Moscow, Russia (savgoshin@hse.ru).

heuristic algorithms for solving the problem in a mixed graph [6]. However, it was not possible to find an algorithm for solving the problem in a mixed *multigraph*. To obtain a solution in the mixed multigraph, the methods of transforming the problem into equivalent arc routing problem Asymmetric Travelling Salesman Problem (ATSP) through the transformation of initial problem into the Asymmetric Generalized Travelling Salesman Problem (GTSP) were examined [10–12].

The paper is organized as follows. First, the mathematical formulation of the problem is pointed out. The next section is dedicated to related works. In the following parts the description of implemented algorithm and methods of the research are presented. Then already obtained results are revealed. In the final part, the future directions of research are described.

II. MATHEMATICAL FORMULATION OF THE MCPP

The weight strongly connected mixed multigraph $G = \langle V, E \cup A, C \rangle$ is given, where V is the set of multigraph's vertices, E is the multiset of edges, A is the multiset of arcs, $C: E \cup A \rightarrow R_+$ cost function giving non-negative weights of arcs and edges between vertices [6].

Let $I = \{1, 2, \dots, |E + A|\}$, $L = \{1, 2, \dots, |V|\}$.

Indexation on the set of vertices V is defined as $inv: V \rightarrow L$, $\forall v_i \in V \forall v_j \in V v_i \neq v_j \Rightarrow i \neq j$, $i = inv(v_i)$. On the multiset $E \cup A$ indexation is defined as $inea: E \cup A \rightarrow I$, $\forall e_i \in (E \cup A) \forall e_j \in (E \cup A) e_i \neq e_j \Rightarrow i \neq j$, $i = inea(e_i)$.

$$\mu = (e_{p_1}, e_{p_2}, \dots, e_{p_k}) \quad (1)$$

Route (1) is a solution of the MCPP that satisfies the following properties.

- $v^-(e_{p_1}) = v^+(e_{p_k})$,
- $\forall i \in \{1, 2, \dots, k-1\} v^+(e_{p_i}) = v^-(e_{p_{i+1}})$, where $v^-(e)$ is the start vertex of arc or edge e , $v^+(e)$ is the end vertices of arc or edge e .
- $E \cup A \setminus \{e_{p_1}, e_{p_2}, \dots, e_{p_k}\} = \emptyset$.

Let $C(\mu) = \sum_{i=1}^k C(e_i)$ is the cost of the route.

Let \mathcal{M} be a set of solutions of the MCPP. It is necessary to find a route $\mu_0 \in \mathcal{M}$ that satisfies the following property $\forall \mu \in \mathcal{M} C(\mu_0) \leq C(\mu)$ or $C(\mu_0) = \min_{\mu \in \mathcal{M}} (C(\mu))$.

III. RELATED WORKS

Number of articles and the research on the CPP issue have been found.

The most popular classifications and variations of the CPP was found in the paper by D. Mohammaditabar [10].

The mathematical formulations of problems can be found in the Optimization and Operation Research book. It shows that the UCPP and DCPD has exact solution and can be solved in polynomial time. The MCPP is NP-hard [11].

In the work of Harold Thimbleby from University College London, the polynomial algorithm and executable code on Java for DCPD in multigraph were represented. The article reveals possible applications CPP. Great attention is paid to the application of the solution of the problem to testing web-site, broken links and etc. [3].

In many researches, the algorithm for solving UCPP is presented. The algorithm is based on making the Eulerian graph and finding the Eulerian cycle in it.

In the paper by G. Laporte the exact algorithm for an undirected and directed graph is covered and the formulation of the problem in the mixed graph is shown [12]. The T. Ralph's paper contains formulation of CPP on the mixed graph in terms of integer linear programming [6].

For the Mixed case, the many heuristics are existed. However, for the MCPP in multigraph finding the algorithms description was failed [6].

It was shown that several arc routing problems (ARP) can be converted into equivalent ARP. It opens the possibility to solve ARP, including the MCPP, as equivalent of another arc routing problem [11–13].

The paper gives grounds to assume the MCPP can be converted into Asymmetric Travelling Salesman Problem (ATSP).

IV. TRANSFORMATION THE MCPP INTO ASYMMETRIC GTSP

The MCPP can be transforms into an equivalent arc routing problem. When problem defined in directed multigraph (DCPP), it can be transformed into ATSP. When problem defined in mixed or undirected multigraph, it can be at first transformed into an asymmetric GTSP, and after that into ATSP [11–12]. The proof of correctness of this algorithm was given by M. Blais [13].

A. Description of transformation algorithm

The process of transformation the MCPP to GTSP is to transform the original graph $G = \langle V, E \cup A, C \rangle$ into equivalent problem on complete graph $\tilde{G} = \langle \tilde{V}, \tilde{A}, \tilde{C} \rangle$.

Each arc $a_{ij}^k \in A$ between to vertices $v_i, v_j \in V$ is represented as vertex $v_{ij}^k \in \tilde{V}$, which must be used in the solution at least once, where k is the serial number of parallel arc.

Each edge $e_{ij}^k \in \tilde{E}$ between to $v_i, v_j \in V$ is represented as two vertices $v_{ij}^{k_1}, v_{ij}^{k_2} \in \tilde{V}$, one of which must be used in the solution, another may not be used, where k is a serial number of parallel edge, k_1, k_2 are the serial numbers of parallel arcs between two vertices.

After replacing the arcs and edges in vertices, the cost from each pair of vertex $v_{ab}^{k_1}, v_{cd}^{k_2} \in \tilde{V}$ in graph \tilde{G} compute, as

$$\tilde{c}_{ij} = d_{bc} + c_{cd}^{k_2} \quad (2)$$

where d_{bc} is the shortest distance between vertices $v_b, v_c \in V$ in original multigraph G .

Then, the complete graph \tilde{G} is partitioned into clusters as follows: each arc and each edge is separate clusters. The number of clusters is equals to $|A \cup E|$. The graph partitioned into clusters because edge can be traverse in two ways, for solving the MCPP any way is appropriate and the problem transforms into asymmetric GTSP [13].

The GTSP is a variation of the Traveling Salesman Problem in which all vertices are divided into clusters, and solution consist from one vertices from each cluster.

TABLE I. FORMULAS FOR COMPUTING ARC COSTS OF ASYMMETRIC GTSP

		1	2	3	4	5	6	7	8
		v_{12}^1	v_{13}^1	v_{13}^2	v_{21}^1	v_{23}^1	v_{23}^2	v_{31}^1	v_{32}^1
1	v_{12}^1	-	$s_{21} + c_{13}^1$	$s_{21} + c_{13}^2$	$s_{22} + c_{21}^1$	$s_{22} + c_{23}^1$	$s_{22} + c_{23}^2$	$s_{23} + c_{31}^1$	$s_{23} + c_{32}^1$
2	v_{13}^1	$s_{31} + c_{12}^1$	-	$s_{31} + c_{13}^2$	$s_{32} + c_{21}^1$	$s_{32} + c_{23}^1$	$s_{32} + c_{23}^2$	$s_{33} + c_{31}^1$	$s_{33} + c_{32}^1$
3	v_{13}^2	$s_{31} + c_{12}^1$	$s_{31} + c_{13}^1$	-	$s_{32} + c_{21}^1$	$s_{32} + c_{23}^1$	$s_{32} + c_{23}^2$	$s_{33} + c_{31}^1$	$s_{33} + c_{32}^1$
4	v_{21}^1	$s_{11} + c_{12}^1$	$s_{11} + c_{13}^1$	$s_{11} + c_{13}^2$	-	$s_{12} + c_{23}^1$	$s_{12} + c_{23}^2$	$s_{13} + c_{31}^1$	$s_{13} + c_{32}^1$
5	v_{23}^1	$s_{31} + c_{12}^1$	$s_{31} + c_{13}^1$	$s_{31} + c_{13}^2$	$s_{32} + c_{21}^1$	-	$s_{32} + c_{23}^2$	$s_{33} + c_{31}^1$	$s_{33} + c_{32}^1$
6	v_{23}^2	$s_{31} + c_{12}^1$	$s_{31} + c_{13}^1$	$s_{31} + c_{13}^2$	$s_{32} + c_{21}^1$	$s_{32} + c_{23}^1$	-	$s_{33} + c_{31}^1$	$s_{33} + c_{32}^1$
7	v_{31}^1	$s_{11} + c_{12}^1$	$s_{11} + c_{13}^1$	$s_{11} + c_{13}^2$	$s_{12} + c_{21}^1$	$s_{12} + c_{23}^1$	$s_{12} + c_{23}^2$	-	$s_{13} + c_{32}^1$
8	v_{32}^1	$s_{21} + c_{12}^1$	$s_{21} + c_{13}^1$	$s_{21} + c_{13}^2$	$s_{22} + c_{21}^1$	$s_{22} + c_{23}^1$	$s_{22} + c_{23}^2$	$s_{23} + c_{31}^1$	-

B. The example of transformation

The example of original multigraph is shown on Fig. 1. Each arc and edge has the cost of traverse. Each vertex has a serial number.

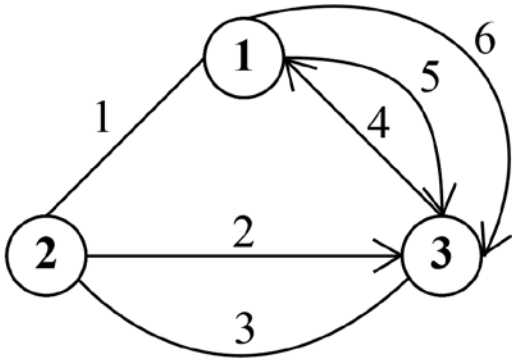


Fig. 1. The original MCPP problem in multigraph

We replace each edge by a pair of two oppositely directed arcs and specify the numbering of parallel arcs between each pair of vertices (see Fig. 2). In multigraph only one arc a_{12}^1 or a_{21}^1 is required, because these arcs represent one edge. The same applies to arc a_{23}^2 or a_{32}^1 .

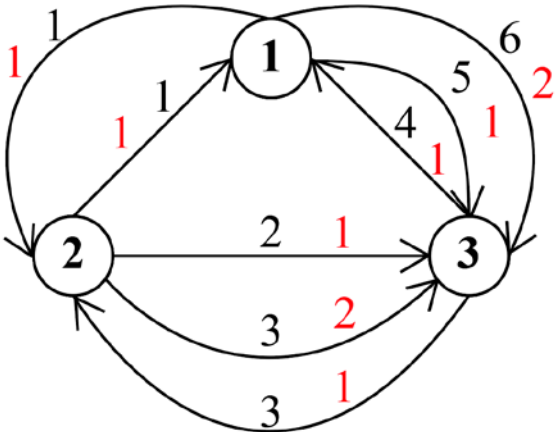


Fig. 2. The results of numbering each parallel arc

After that, should replace each arc and edge as vertex. We received new graph \tilde{G} with 8 vertices. The \tilde{V} can be calculates as formula:

$$|\tilde{V}| = |\tilde{A}| + 2|\tilde{E}| \quad (3)$$

The cost from each pair of vertices is calculated by formulas (see Table 1, see Table 2). The vertices represent the edge are marked with a color in the table (different colors for different edges).

TABLE II. THE COST MATRIX

		1	2	3	4	5	6	7	8
		v_{12}^1	v_{13}^1	v_{13}^2	v_{21}^1	v_{23}^1	v_{23}^2	v_{31}^1	v_{32}^1
1	v_{12}^1	-	6	7	1	2	3	6	5
2	v_{13}^1	5	-	10	4	5	6	4	3
3	v_{13}^2	5	9	-	4	5	6	4	3
4	v_{21}^1	1	5	6	-	3	4	7	6
5	v_{23}^1	5	9	10	4	-	6	4	3
6	v_{23}^2	5	9	10	4	5	-	4	3
7	v_{31}^1	1	5	6	2	3	4	-	6
8	v_{32}^1	2	6	7	1	2	3	6	-

Then vertices from \tilde{V} are partitioned into clusters. Fig. 3 depicts the vertices and clusters of transformed graph.

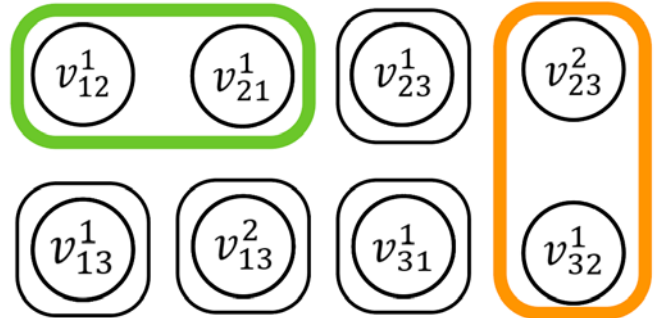


Fig. 3. The vertices and clusters of transformed problems

The transformation of the MCPP into Asymmetric GTSP is received. After transformation of the original MCPP into the GTSP, the existing algorithm for GTSP can be applied. However, we try to transform received Asymmetric GTSP into ATSP.

V. TRANSFORMATION THE ASYMMETRIC GTSP INTO ATSP

The Asymmetric GTSP can be transformed into ATSP with the same number of vertices. Proof that the algorithm is applicable to any GTSP is given in the paper by G. Laporte [13]. Below the description of transformation algorithm is described [12, 16].

A. Description of transformation algorithm

The steps of transformation:

- Define the set of vertices \tilde{V} , which is equal \tilde{V} .
- Within each cluster, set the cost of the arcs between each pair of vertices in the cluster equal to 0. $\tilde{c}_{ij} = 0$,

where i, j – the number of vertices belonging to the clusters.

- If two vertices $v_i, v_j \in \tilde{V}$ are in different clusters, the cost calculate as:

$$\tilde{c}_{ij} = \tilde{c}_{i_j} \quad (4)$$

where v_l is another vertex from cluster with v_i .

The based idea of this transformation that salesman, firstly visit all vertices into clusters and then go into another cluster. The GTSP tour can be easily find from TSP tour, by extracting first vertex from each cluster from TSP tour [14].

B. The example of transformation

Define the auxiliary tables.

Table 3 shows the distribution of vertices over clusters. While Table 4 shows the distribution of clusters over vertices.

TABLE III. THE CLUSTERS

Cl	v
1	$v_{12}^1; v_{21}^1$
2	v_{13}^1
3	v_{13}^2
4	v_{23}^1
5	$v_{23}^2; v_{32}^1$
6	v_{31}^1

TABLE IV. THE VERTICES INTO CLUSTERS

N	1	2	3	4	5	6	7	8
v	v_{12}^1	v_{13}^1	v_{13}^2	v_{21}^1	v_{23}^1	v_{23}^2	v_{31}^1	v_{32}^1
Cl	1	2	3	1	4	5	6	5

Table 5 shows the cost matrix of equivalent ATSP. Costs were calculated in accordance with subsection A.

TABLE V. THE ATSP COST MATRIX

		1	2	3	4	5	6	7	8
		v_{12}^1	v_{13}^1	v_{13}^2	v_{21}^1	v_{23}^1	v_{23}^2	v_{31}^1	v_{32}^1
1	v_{12}^1	-	5	6	0	3	4	7	6
2	v_{13}^1	5	-	10	4	5	6	4	3
3	v_{13}^2	5	9	-	4	5	6	4	3
4	v_{21}^1	0	6	7	-	2	3	6	5
5	v_{23}^1	5	9	10	4	-	6	4	3
6	v_{23}^2	2	6	7	1	2	-	6	0
7	v_{31}^1	1	5	6	2	3	4	-	6
8	v_{32}^1	5	9	10	4	5	0	4	-

VI. METHODS OF TESTING

The code for graph transformations was written on C++ in MS Visual Studio 2015. For solving ATSP the library LKH by Keld Helsdgaun was used as a black box [17].

To test all algorithms, two test databases were used. For multigraph, the test data sets were not found. However, graph is a special case of multigraph (without parallel arcs and edges) and algorithm can be tested on graph data sets. In Bönisch’s database the input data for 50, 100, 200 vertices in graph are presented. For each dimension, there is 25 different tasks. In Corberan’s database the input data for 500, 1000, 1500, 2000, 3000 vertices in graph are presented. For each dimension, there is 24 different tasks. [15, 16].

The time performance of the two transformations (the

MCCP into GTSP, the GTSP into ATSP) were measured as follows:

- Test data were loaded in console program;
- The measurements for each input data set were carried out 10 times. The results of computational time were obtained as the average of 10 runs of the program:

$$T_{av} = \frac{T_1 + \dots + T_{10}}{10} \quad (5)$$

At first, the MCCP was transformed into the GTSP, then the GTSP was converted into ATSP.

The error and time performance of the LKH [16, 17] algorithm was measured on the converted data, as follows:

- Test converted data of ATSP were loaded in console program;
- The measurements for each input data set were carried out 10 times. The results of time performance were obtained as the average of 10 runs of the program (according to the formula (5));
- Error rate of the algorithms was evaluated according to the formula:

$$Error = \frac{C(\mu) - C(\mu_0)}{C(\mu_0)} \quad (6)$$

where $C(\mu)$ is the resulting length of the route of the MCCP, $C(\mu_0)$ is the optimum length of the route of the MCCP given in input data.

All test provides on Mac Book Pro 13 retina 2014 (Intel Core i5, 2.6 GHz).

VII. OBTAINED RESULTS

The test results are divided into three subsections.

At first, the time performance of the MCCP into the GTSP transformation was received. Second, the testing of transformation the GTSP into ATSP was conducted. Finally, the LKH [18, 19] algorithm was applied to the transformed data and its time performance and error rate were measured.

A. Results of testing transformation the MCCP into the GTSP

The time performance of algorithm’s work most depends on number of vertex $|\tilde{V}|$ in transformed problems. The graphic for $|V| = 500$ is presented on Fig.4. The results for all test data are presented in Table 6.

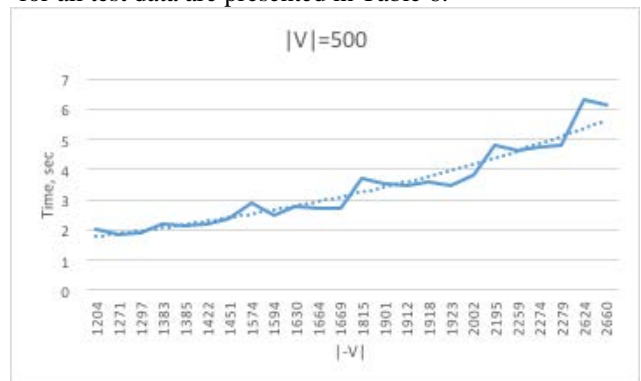


Fig. 4. The time performance of transforming the MCCP into the GTSP, $|V|=500$

Relatively long time of work is because that during the calculation of the transformed weights for the original matrix, it is necessary to find the matrix of shortest paths between all pairs of vertices (using the Floyd Warshall Algorithm, computational complexity - $O(V^3)$).

TABLE VI. THE RESULT OF TESTING TRANSFORMING THE MCPP INTO THE GTSP

$ V $	$ A _{av.}$	$ E _{av.}$	$ \tilde{V} _{av.}$	Average time, sec	Time Standard Deviation
50	73	59	190	0,060	0,007
100	139	118	374	0,162	0,020
200	282	224	730	0,695	0,067
500	563	621	1804	3,380	0,432
1000	1133	1248	3628	13,956	1,323
1500	1689	1866	5421	37,620	3,766
2000	2256	2481	7218	72,198	7,540
3000	3319	3726	10770	196,126	16,769

B. Results of testing transformation the GTSP into the ATSP

The time of algorithm's work most depends on number of vertex $|\tilde{V}|$ in transformed problems. The graphic for $|V| = 500$ is presented on Fig.5. The results for all test data are presented in Table 7.

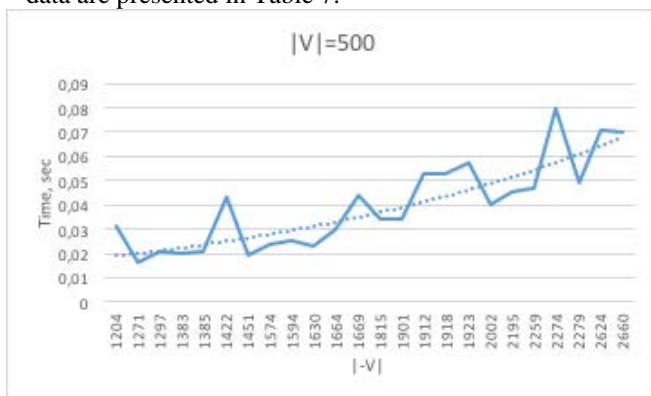


Fig. 5. The time performance of transforming the GTSP into the ATSP, $|V|=500$

TABLE VII. THE RESULT OF TESTING TRANSFORMIN THE GTSP INTO THE ATSP

$ V $	$ A _{av.}$	$ E _{av.}$	$ \tilde{V} _{av.}$	Average time, sec	Time Standard Deviation
50	73	59	190	0,001	0,001
100	139	118	374	0,003	0,001
200	282	224	730	0,008	0,001
500	563	621	1804	0,039	0,001
1000	1133	1248	3628	0,176	0,021
1500	1689	1866	5421	0,905	0,097
2000	2256	2481	7218	3,617	0,411
3000	3319	3726	10770	6,086	0,703

It can be seen from the table that the transformation takes place quickly.

C. Results of applying the LKH to solving the MCPP

For transforming data set the LKH was applied. We did not make out the principle of the algorithm, but applied it as a black box with standard perimetries, suggested by the author [17].

The results of average computing time and error rate is represented in Table 8. The applying algorithm for solving the MCPP as ATSP throw the transformations opens many opportunities for research. As seen from Table 8, the results of such approach are good. Using confidence intervals, the accuracy of the time estimate was determined, which was at least 87%.

VIII. CONCLUSION

The way of solving the MCPP in multigraph were proposed. Previously, the MCPP were solved only in a mixed graph.

The algorithm for transforming the MCPP in multigraph into an equivalent arc routing problem the GTSP, and then in the ATSP was implemented in C++ and tested.

The approach to transforming the MCPP in mixed multigraph into equivalent ARP was evaluated on test data from the open databases. Testing has shown that this approach can be applied to the solve the MCPP in multigraph and should be further investigated with various algorithms for solving the ATSP.

TABLE VIII. THE RESULT OF TESTING LKH TO SOLVING THE TRANSFORMED MCPP

	Average time, sec	Time Standard Deviation	Average error, %	Error Standard Deviation
$ V =50$	28,854	2,987	0,04	0,001
$ V =100$	60,345	5,301	0,07	0,001
$ V =200$	112,980	10,439	0,12	0,002
$ V =500$	171,713	17,106	0,58	0,005
$ V =1000$	500,729	62,606	0,92	0,007
$ V =1500$	836,333	89,486	1,05	0,006
$ V =2000$	1139,467	113,358	1,19	0,006
$ V =3000$	2403,892	240,273	1,20	0,008

In our future work, we are going to fine-tune parameters of LKH methods using genetic search optimization algorithms. Further, it is possible to apply and investigated other existing algorithms for ATSP. Next, we are going to develop (prepare) test data for the mixed multigraph.

REFERENCES

- [1] V. Bryant and P. David, Decision Maths 1, London: Heinemann Educational Publishers, 2004.
- [2] P. Vreda and P. Black, Dictionary of Algorithms and Data Structures, National Institute of Standards and Technology, 2014.
- [3] H. Thimbleby, "The directed Chinese Postman Problem," *Software Practice and Experience*, vol. 33, no. 11, pp. 1081-1096, 2003.
- [4] G. Laporte, "The Undirected Chinese Postman Problem," in *Arc Routing*, 2013, pp. 53-64.
- [5] J. Edmonds and E. Johnson, "Matching, Euler tours and the Chinese postman," *Mathematical Programming*, vol. 5, no. 1, pp. 88-124, 1973.
- [6] T. K. Ralphs, "On the Mixed Chinese Postman Problem," *School of Operations Research and Industrial Engineering*, pp. 123-127, 1993.
- [7] M. Guan, "On the windy postman problem," *Discrete Applied Mathematics*, vol. 9, pp. 41-46, September 1984.
- [8] A. M. Rodrigues and J. S. Ferreira, "MIC'2001 - 4th Metaheuristics International Conference," in *Solving the Rural Postman Problem by Memetic Algorithms*, Porto, Portugal, 2001.
- [9] G. Ghiani and G. Improta, "An algorithm for the hierarchical

- Chinese postman problem," *Operations Research Letters*, no. 26, pp. 27-32, 2000.
- [10] D. Mohammaditabar, "Chinese Postman Problem," pp. 224-225, 2013.
- [11] U. Derigs, *Optimization and Operations Research*, Singapore: EOLSS Publications, 2009.
- [12] G. Laporte and M. Gendreau, "Arc Routing Problems, Part I: The Chinese Postman Problem," *Operations Research*, vol. 43, no. 2, pp. 131-242, April 1993.
- [13] G. Laporte and M. Blais, "Exact Solution of the Generalized Routing Problem through Graph Transformations," *Operations Research*, vol. 54, no. 8, pp. 906-910, 2003.
- [14] C. E. Noon and J. C. Bean, "An Efficient Transformation Of The Generalized Traveling Salesman Problem," *INFOR Information Systems and Operational Research*, vol. 31, no. 1, February 1993.
- [15] S. Bönisch, "Implementierung der Edmonds-Johnson Heuristik für das Mixed Chinese Postman Problem," 21 December 1999. [Online]. Available: <http://comopt.ifi.uni-heidelberg.de/teaching/praktikum/projekte/mcppHeuristikSebastianBoenisch.tar.gz>. [Accessed 20 April 2017].
- [21] R. J. Roy S., "The Capacitated Canadian Postman Problem," *INFOR*, vol. 29, no. 1, pp. 58-73, 1989.
- [16] A. Corberan, "Arc Routing Problems: Data Instances," [Online]. Available: <http://www.uv.es/corberan/instancias.htm>. [Accessed 3 April 2017].
- [17] K. Helsgaun, "LKH," [Online]. Available: <http://akira.ruc.dk/~keld/research/LKH/>. [Accessed 9 April 2017].
- [18] K. Helsgaun, "An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic," *European Journal of Operational Research*, vol. 126, no. 1, pp. 106-130, 2000.
- [19] K. Helsgaun, "Solving Arc Routing Problems Using the Lin-Kernighan- Helsgaun Algorithm," Roskilde University, 2014.
- [20] K. Helsgaun, "Solving the Equality Generalized Traveling Salesman Problem Using the Lin-Kernighan-Helsgaun Algorithm," Roskilde University, 2014.