The Mixed Chinese Postman Problem

M.K. Gordenko <mkgordenko@gmail.ru>
S.M. Avdoshin <savdoshin@edu.hse.ru>
Software Engineering School,
National Research University Higher School of Economics
20, Myasnitskaya, Moscow, 101000, Russia

Abstract. The routing problems are important for logistic and transport sphere. Basically, the routing problems related to determining the optimal set of routes in the multigraph. The Chinese postman problem (CPP) is a special case of the routing problem, which has many potential applications. We propose to solve the MCPP (special NP-hard case of CPP, which defined on mixed multigraph) using the reduction of the original problem into General Travelling Salesman Problem (GTSP). The variants of CPP are pointed out. The mathematical formulations of some problems are presented. The algorithm for reduction the MCPP in multigraph into GTSP is shown. The experimental results of solving MCPP in multigraph through the reduction into GTSP are presented.

Keywords: Mixed Chinese Postman Problem, Arc Routing Problem, heuristic algorithm, Traveling Salesman Problem

DOI: 10.15514/ISPRAS-2017-29(4)-7

For quoting: Gordenko M.K., Avdoshin S.M. The Mixed Chinese Postman Problem. *Trudy ISP RAN/Proc. ISP RAS*, vol. 29, issue 4, 2017, pp. 107-122. DOI: 10.15514/ISPRAS-2017-29(4)-7

1. Introduction

The Chinese Postman Problem (CPP) was originally studied by the Chinese mathematician Kwan Mei-Ko in 1962 on the example of the rural postman problem [1]. A problem is called the CPP after Kwan Mei-Ko [2].

In the modern world, the number of companies and industries that are interested in building an optimal route of product delivery is growing. For example, the postman delivering letters or leaflets wants to know the optimal route that traverses every street in the given area, starting and ending at the office [3].

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 various classifications of the CPP. This problem can be applied for a directed, undirected, mixed graph, or in a multigraph (a graph with parallel directed and undirected edges). The CPP can also be closed (the postman should return to the starting point) or open (starting and ending points can be different). The problem in directed or undirected graph has exact algorithms and may be solved in polynomial time. The mixed case is NP-hard and there are no polynomial-time algorithms for solving the CPP in mixed graph or multigraph exactly [4, 3].

In this paper, heuristic algorithms for the mixed case are described and assessed. The mixed CPP (MCPP) is a simply-stated problem, which has many useful applications, but has no exact algorithms [3].

The objective of the research is implementation and quality assessment of heuristic algorithms for the MCPP.

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 next part a brief description of implemented algorithms and methodology of the research are presented. Then, already obtained results are revealed. In the final part, the expected results and future directions of research are described.

In this article, in accordance with generally accepted definitions, under the understanding of an understanding directed edge, under the edge is an undirected edge.

2. The General Routing Problem

The routing is one of the most important problem in the optimization researches. The GRP is to define a minimum cost set of routes (one route also is possible) in a multigraph, that must include some required vertices and pass through some required edges and arcs of the original multigraph [1].

Formally, the GRP is defined on multigraph $G =< V, E, A, C >$, where

V is a set of vertices,

A is a multiset of directed edges (arc);

E is a multiset of undirected edges (edges);

C: $E \cup A \to R_+$ is a cost function giving non-negative weights of arcs and edges between vertices.

In the routing problems, it is not necessary to visit all vertices, edges and arcs of the multigraph. Two subsets of edges and arcs $A_R \subseteq A$ and $E_R \subseteq E$ are defined. The arcs and edges from A_R and E_R must necessarily appear in the solution. Let the subset of vertices $V_R \subseteq V$ consist of those vertices that must appear in the route.

The goal of all routing problems is to define a minimum cost set of routes, that traverses all the arcs and edges from the multisets A_R and E_R and includes all vertices of the set V_R.

3. The Vehicle Routing Problem

The VRP is a special case of the GRP with $A_R = \emptyset$ and $E_R = \emptyset$, i.e. the restrictions on the edges and arcs, which must necessarily appear in the route, are absent. The VRP is to determine the Hamiltonian cycle of minimum cost, which traverse all vertices of the subset V_R [1].

In the case, when $V_R = V$, the problem reduces to one of the most famous problem of combinatorial optimization – the classical Traveling Salesman Problem (TSP).

4. The Arc Routing Problem

Another special case of the GRP is the Arc Routing Problem (ARP), it is to determine the minimum cost set of routes, that traverses all required edges E_R and all required arcs A_R of original multigraph. In the ARP, there are no restrictions on the presence of vertices in the route, i.e. $V_R = \emptyset$. The CPP is the variant of ARP. In the original formulation, the CPP is the problem, where the postman should traverse through every street in the given area.

5. The Variants of Chinese Postman Problem

The CPP was originally studied by the Chinese mathematician Kwan Mei-Ko in 1960. A problem is called the Chinese Postman Problem after him. Kwan Mei-Ko defined the problem on undirected graph. Today, there are many various classifications of CPP, including classifications based on the graph type, on the type of solution route and other restrictions and additions [2].

In the modern world, the number of companies and industries that are interested in building an optimal route of product delivery is growing. For example, the postman delivering letters or leaflets wants to know the optimal route that traverses every street in the given area, starting and ending at the office. Apart from the traditional application of the CPP to solving the routing problems such as path planning of snowplows or serving teams, there are a wide range of applications including robot exploration, testing web site usability and finding broken links [3].

Below, the most popular variants of the CPP are presented.

5.1 The Undirected Chinese Postman Problem

The formulation of the Chinese Postman Problem in undirected graph (UCPP) is an original formulation of the CPP problem. The UCPP is a special case of ARP, where
$A = \emptyset$ and $E_R = E$. The UCPP belong to class of P problems.

5.2 The Directed Chinese Postman Problem

The Chinese Postman Problem in directed graph (DCPP) is the modification of UCPP, where every arc (directed edge) can be traversed in one direction. Another name of problem is the New York Street Sweeper Problem. The DCPP is a special case of ARP, where $A_R = A$ and $E = \emptyset$. The DCPP belong to class of P problems.

5.3 The Windy Chinese Postman Problem

The Windy Chinese Postman Problem (WCCP) is the interesting generalization of classical CPP, which has many real uses. In WCPP the cost of traversing some edge depends on way of traversing. The WCPP is a special case of ARP, where $E_R = E$, $A = \emptyset$ and at least for one edge the cost of traversing in direct way is differ from cost of traversing it in opposite way. The DCPP belong to class of NP-hard problems, which cannot be solved in polynomial time.

5.4 The Rural Chinese Postman Problem

The Rural Chinese Postman Problem (RCPP) is a special case of ARP, where $A_R \subseteq A$, $E_R \subseteq E$. Another name of RCPP is the Selecting Chinese Postman Problem. In all above defined CPP problems, it is necessary to find a closed shortest route, that traverses all edges or arcs of the original multigraph at least once. In the real world, it is not always necessary to traverse all roads (edges or arcs). It is enough to traverse only a set of requires arcs and edges (A_R and E_R). The RCPP belong to class of NP-hard problems, which cannot be solved in polynomial time.

5.5 The Mixed Chinese Postman Problem

The Mixed Chinese Postman Problem (MCPP) is a well-known version of the CPP, where multigraph contains both arcs and edges. The MCPP is a special case of ARP, where $E_R = E$ and $A_R = A$. The MCPP belong to class of NP-hard problems.There are other variants of the problem, such as Hierarchical Postman Problem (HCPP), k-Chinese Postman Problem (k-CPP), Chinese Postman Problem with Time Windows and others.

6. The Variants of Chinese Postman Problem

The MCPP is one of the most important problem of the ARP. The MCPP is a special case of ARP, for which $A_R = A \neq \emptyset$, $E_R = E \neq \emptyset$.

An edge $\{v_i, v_j\}$ (an unordered pair of vertices) from the set E is fixed. Let (v_i, v_j) is ordered pair of vertices (this mean, that should traverse $\{v_i, v_j\}$ from vertex v_i to vertex v_j). Note, that, $\forall\{v_i, v_j\} \in E$, $c((v_i, v_j)) = c(\{v_j, v_i\})$ and $C((v_j, v_i)) = C(\{v_j, v_i\})$, it means that the cost of traversing the edge in any directions is the same.

The mathematical formulation of the MCPP problem is presented below.

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

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

Route $\mu = (v_{l_1}, e_{p_1}, v_{l_2}, e_{p_2}, \dots, v_{l_n}, e_{p_k})$ is the solution of the MCPP that satisfies to the following properties:

- $e_{p_i} = (v_{l_i}, v_{l_{i+1}})$, $i = 1,2, \dots, k-1$;
- $e_{p_k} = (v_{l_k}, v_{l_1})$;
- $E \cup A \,/\{e_{p_1}, e_{p_2}, \dots, e_{p_k}\} = \emptyset$.

Let $C(\mu) = \sum_{i=1}^{k} C(e_{p_i})$ is the cost of the MCPP 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 $\forall \mu \in \mathcal{M} \ C(\mu_0) \leq \mathrm{C}(\mu)$ or $C(\mu_0) = \min_{\mu \in \mathcal{M}}(\mathrm{C}(\mu))$.

7. The reduction of Chinese Postman Problem

The MCPP can be reduced to an equivalent ARP. When problem defined in directed graph (DCPP), it can be reduced to the asymmetric TSP. When problem defined in mixed or undirected graph, it can be reduced to GTSP [4-6].

7.1 Description of reduction algorithm

Originally, the reduction algorithm was presented for the graph [3, 4]. The algorithm modification, applicable to the multigraph, is given below [7].

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

Table 1. 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_{31}^2	$s_{21} + c_{12}^1$	$s_{21} + c_{13}^1$	$s_{21} + c_{13}^2$	$s_{22} + c_{21}^1$	$s_{22} + c_{23}^1$	$s_{22} + c_{23}^2$	$s_{23} + c_{31}^1$	-

Each arc $a_{ij}^k \in A$ between to vertices $v_i \in V$, $v_j \in V$ 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 $v_i \in V$, $v_j \in V$ is represented as two vertices $v_{ij}^{k_1} \in \tilde{V}, v_{ij}^{k_2} \in \tilde{V}$, 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 $v_{ab}^{k_1} \in \tilde{V}, v_{cd}^{k_2} \in \tilde{V}$ in graph $\tilde{G}$ compute, as $\widetilde{c_{ij}} = d_{bc} + c_{cd}^{k_2}$, where d_{bc} is the shortest distance between vertices $v_b \in V$, $v_c \in V$ in original multigraph G. Then, the compete graph $\tilde{G}$ is partitioned into clusters as follows: each arc and each edge is separate cluster. The number of clusters is equal 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 GTSP [5-7].

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

7.2 The example of reduction

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

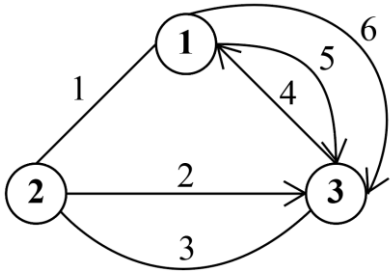

Fig. 1. The original MCPP problem in multigraph

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.

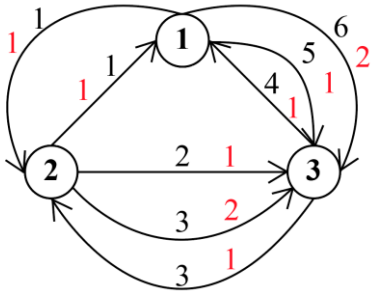

Fig. 2. The results of numbering each parallel arc

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 according to the formula $|\tilde{V}| = |\tilde{A}| + 2|\tilde{E}|$.

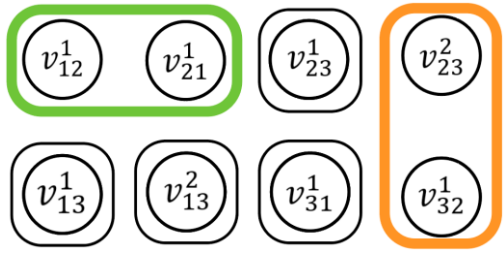

Fig. 3. The vertices and clusters of transformed problems

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 2. 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. The reduction of the MCPP to Asymmetric GTSP is received. After transformation of the original MCPP into the GTSP, the existing algorithm for GTSP can be applied.

8. Algorithms of GTSP Solving

In the work, the nearest neighbor heuristic algorithm (NN) and its modifications were applied to solve the GTSP problem.

8.1 Nearest Neighbor Heuristic (NN)

Nearest Neighbor heuristic belongs to the group of tour construction heuristics. In the tour construction methods, the route is built by adding new vertices at each step, according to some rules, while the already existing tour does not improve.

The algorithm starts building the route from some starting vertex, and then selects the nearest vertex from another cluster to the start point and adds it to the route. Then, the nearest vertex, belonging to an unused cluster, should appear in the route, until all the clusters are used. After adding the vertices, we return to the starting point.

Time complexity of the algorithm in the best and worst case – $O(|\tilde{\tilde{V}}|^2)$ [5].

8.2 Repetitive Nearest Neighbor Heuristic (RNN)

Since the length of the obtained route in NN depends on the considered starting vertex, another variant of the nearest-neighbor is the repeated nearest neighbor, which calculates the cost of the route, when NN is applied to each vertex as starting vertex, and chooses the best route among all.

Time complexity of the algorithm in the best and worst case – $O(|\tilde{\tilde{V}}|^3)$ [5].

8.3 Improved Nearest Neighbor Heuristic (INN)

Another modification of NN is the heuristic, in which the shortest edge between two vertices from different clusters is selected as the starting edge for the route, and then NN is applied to the found edge (the end vertex of shortest edge is the starting vertex for NN).

Time complexity of the algorithm in the best and worst case – $O(|\tilde{\tilde{V}}|^2)$ [5].

8.4 Repetitive Improved Nearest Neighbor Heuristic (RINN)

This method is a joint modification of the methods RNN and INN, which based on the fact, that in the problem there are several edges with a minimum weight. It is proposed to find the lengths of routes for all minimal edges [6].

Time complexity of the algorithm in the best case is $O(|\tilde{\tilde{V}}|^2)$ and in the worst case is $O(|\tilde{\tilde{V}}|^3)$ [6].

8.5 Double-Ended Nearest Neighbor Heuristic (DENN)

The algorithm starts building the route from some starting vertex, and, then, selects the nearest vertex to the start vertex and adds it to the route. Then the nearest vertex, belonging to an unused cluster, to the first vertex in the solution or the last is added, should appear in the route, until all the clusters are used. Thus, the route grows from both ends, the vertices can be added at the beginning of the route, and at the ending of the route. After adding all the vertices, we return to the starting point.

Time complexity of the algorithm in the best and worst case – $O(|\tilde{\tilde{V}}|^2)$ [5].

8.6 Loneliest Nearest Neighbor Heuristic (LNN)

The main idea of the heuristic is that the vertices most remote from the others should be paid special attention during the construction of the route to avoid their

later inclusion in the route with higher cost. To make such heuristic possible, the concept of "loneliness" of the city was introduced. Together with the distance to the nearest neighbor, the closeness of the nearest neighbors will also be the criteria for adding the next vertex to the route. "Lonely" neighbors, i.e. most remote, will be preferable to others. At the preprocessing stage, a new distance matrix is obtained, such that shorter new distances from the vertex to the others are a weighted function of short old distances to these vertices, and a higher loneliness of this city. Then NN is applied to the new matrix [5].

Time complexity of the algorithm in the best and worst case – $O(|\tilde{\tilde{V}}|^2)$.

8.7 Double-Ended Nearest Neighbor Heuristic (DENLN)

The heuristic is a modification of the NLN and DENLN heuristics, the weighted distance function is also calculated here, considering the "loneliness" of the city, but the DENLN algorithm is already applied in the next stages [5].

Time complexity of the algorithm in the best and worst case – $O(|\tilde{\tilde{V}}|^2)$ [6].

9. Methods of testing

The algorithm for graph transformations and solving GTSP was written on C++ in MS Visual Studio 2015.

To test all algorithms, the two 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 [8]. For each dimension, there are 75 different tasks. The test data from Angel Corberan web-site for 500, 1000, 1500, 2000 and 3000 vertices also was used [9]. For each dimension, there are 25 different tasks.

Table 3. The Pareto-optimal algorithms for solving MCPP through the reduction to GTSP

	$\|V\| = 50$	$\|V\| = 100$	$\|V\| = 200$	$\|V\| = 500$	$\|V\| = 1000$	$\|V\| = 1500$	$\|V\| = 2000$	$\|V\| = 3000$
NN			+		+	+		
INN	+	+	+	+	+	+	+	+
RINN	+							+
DENN								
NLN						+		
DENLN								
RNN	+	+	+	+	+	+	+	+

The time performance and error rate of proposed approach 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: $T_{av} = \frac{T_1 + \cdots + T_{10}}{10}$.
- Error rate of the TSP algorithms was evaluated according to the formula $Error = \frac{C(\mu) - C(\mu_0)}{C(\mu_0)}$, where $C(\mu)$ is the resulting length of the route of the MCPP, $C(\mu_0)$ is the optimum length of the route of the MCPP given in input data.

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

10. Obtained results

For all tests from test database the time and error rate were computed. On Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8, Fig. 9, Fig. 10, Fig. 11 the results are presented in the form of diagrams, where for each described above algorithm the average computational time and error rate are depicted.

Diagrams allow determine the Pareto-Optimal algorithms on two criteria: computational time and error rate. For all tested dimension, this groups are similar and contain RNN and INN algorithms.

The obtained results make it possible to conclude that the proposed approach is applicable to MCPP and gives good results in terms of computational time and error, taking into account the fact, that one of the simplest heuristics was used (the nearest neighbor heuristic).

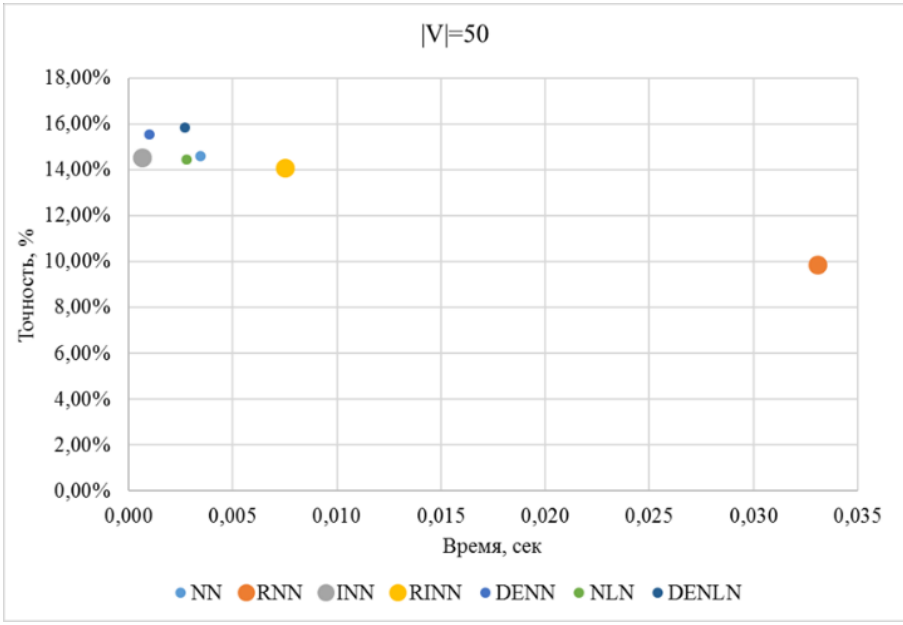

Fig. 4. The results for |V|=50

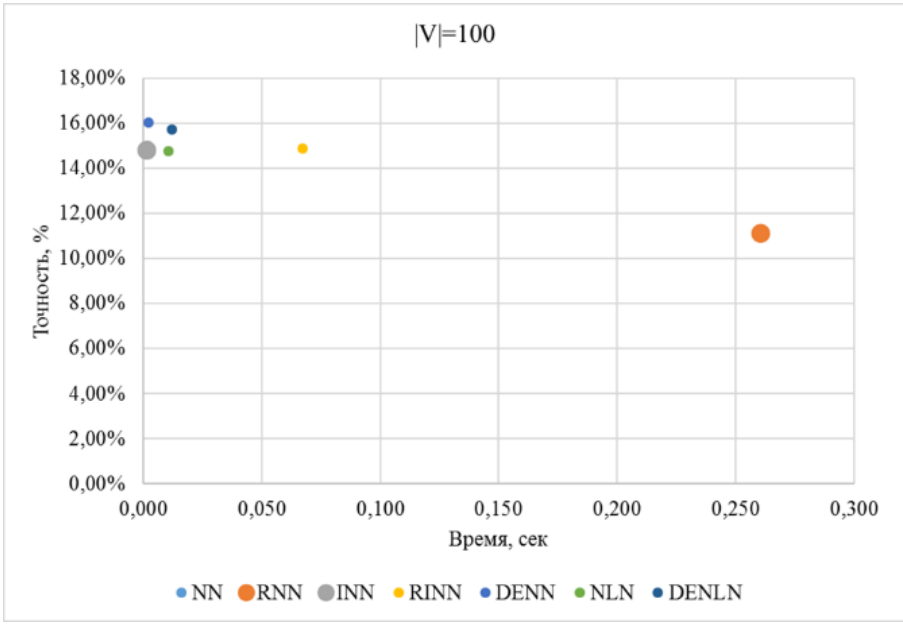

Fig. 5. The results for |V|=100

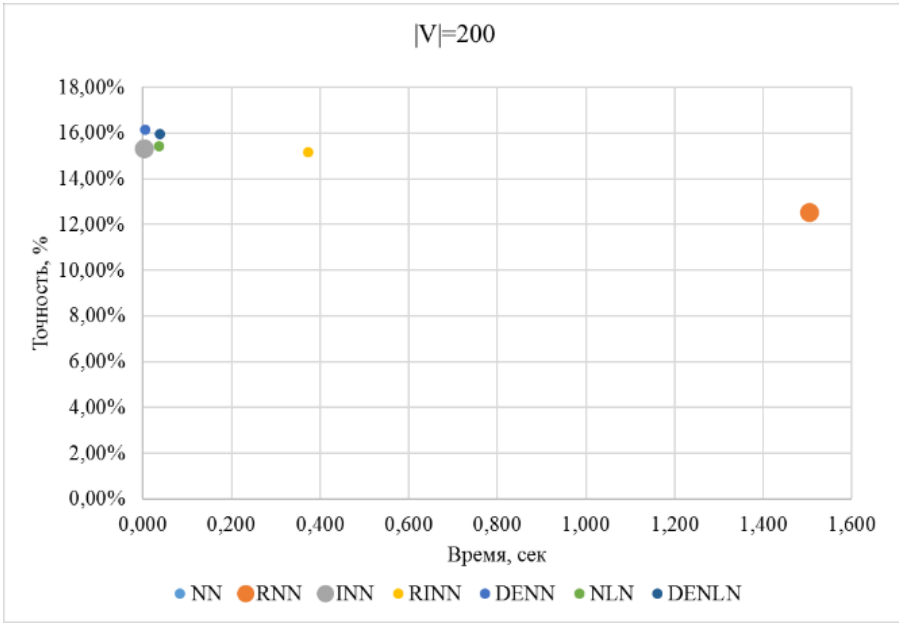

Fig. 6. The results for |V|=200

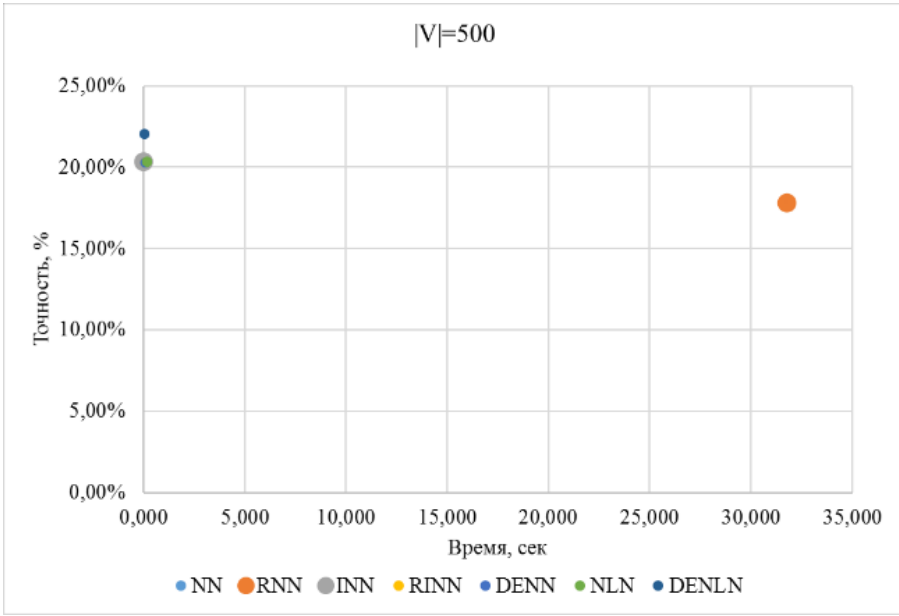

Fig. 7. The results for |V|=500

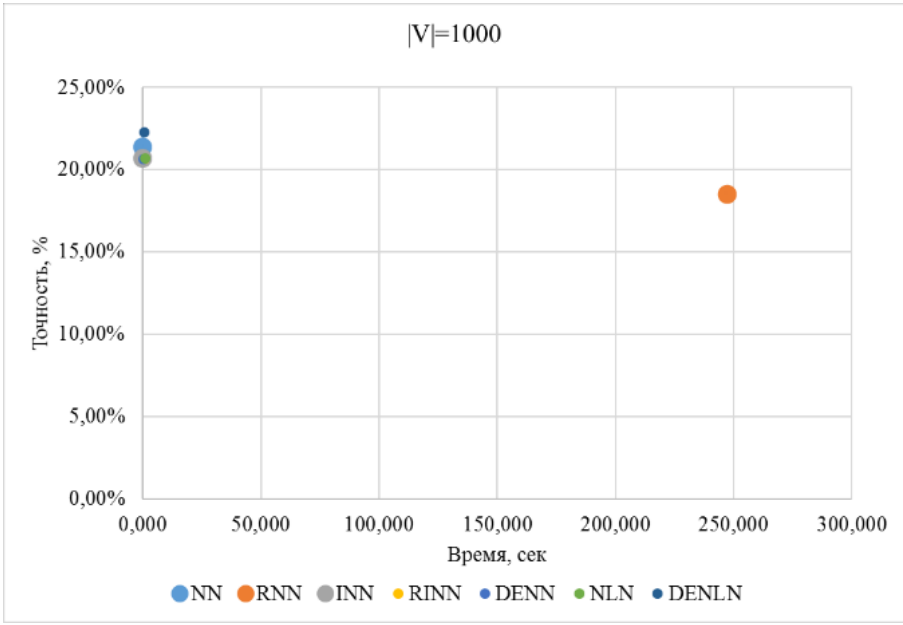

Fig. 8. The results for |V|=1000

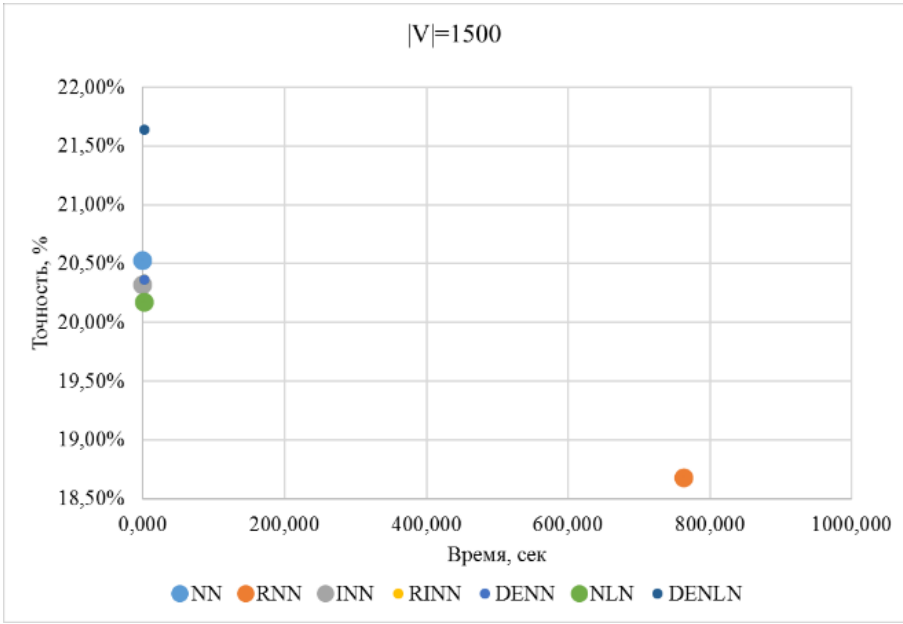

Fig. 9. The results for |V|=1500

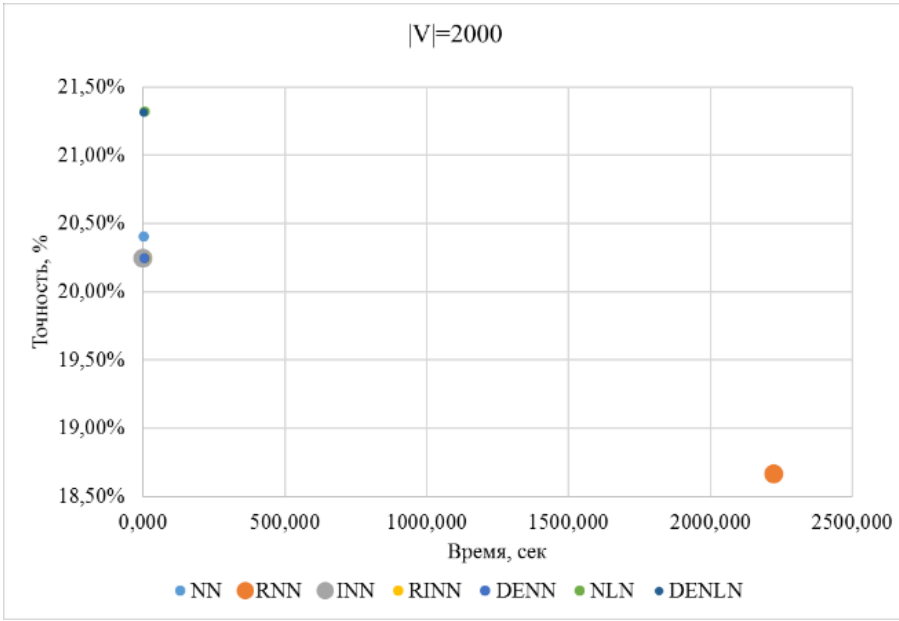

Fig. 10. The results for |V|=2000

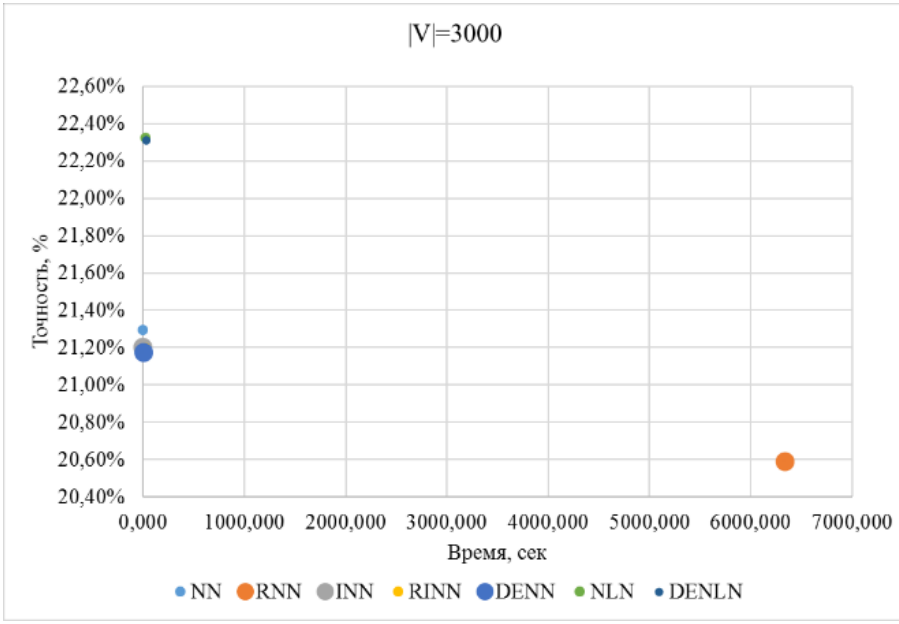

Fig. 11. The results for |V|=3000

References

[1]. 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.

[2]. H. Thimbleby, "The directed Chinese Postman Problem," Software Practice and Experience, vol. 33, no. 11, pp. 1081-1096, 2003.

[3]. Mei-Ko Kwan, «Graphic programming using odd or even points» Acta Mathematica Sinica, p. 263–266, 1960.

[4]. 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.

[5]. F. L. Pimentel, «Double-ended nearest and loneliest neighbour–a nearest neighbour heuristic variation for the travelling salesman problem» Revista de Ciências da Computação, т. 6, № 6, 2016.

[6]. P. Vreda and P. Black, Dictionary of Algorithms and Data Structures, National Institute of Standards and Technology, 2014.

[7]. G. Laporte, «Modeling and solving several classes of arc routing problems as traveling salesman problems» Computers & operations research, т. 24, № 11, pp. 1057-1061, 1997.

[8]. S. Bönisch, «Implementierung der Edmonds-Johnson Heuritik für das Mixed Chinese Postman Problem» 21 December 1999.

[9]. Á. Corberán, "Arc Routing Problems: Data Instances," [Online]. Available: http://www.uv.es/corberan/instancias.htm. [Accessed 3 April 2017].

Смешанная задача китайского почтальона

М.К. Горденко <mkgordenko@gmail.ru>

С.М. Авдошин <savdoshin@edu.hse.ru>

Департамент программной инженерии, Национальный исследовательский университет «Высшая школа экономики»,

101000, Москва, ул. Мясницкая, д. 20

Аннотация. Задачи маршрутизации важны для областей логистики и управления трансортом. Задачи маршрутизации в основном связаны с определением оптимального набора путей в мультиграфе. Задача китайского почтальона (CPP) является особым случаем задачи маршрутизации, имющим много потенциальных приложений. Мы предлагаем решение MCPP (специального NP-полного случая CPP на смешанном мультиграфе) с использованием редуцирования исходной задачи к обобщенной задаче коммивояжера (General Traveling Salesman Problem, GTSP). Указываются варианты CPP. Представлены математические формулировки некоторых проблем. Показан алгоритм редуцирования MCPP в мультиграфе к GTSP. Приводятся экспериментальные результаты решения MCPP в мультиграфе посредством редуцирования к GTSP.

Ключевые слова: смешанная задача китайского почтальона, задача маршрутизации, эвристический алгоритм, задача коммивояжера

DOI: 10.15514/ISPRAS-2017-29(4)-7

Для цитирования: Горденко М.К., Авдошин С.М. Смешанная задача китайского почтальона. *Труды ИСП РАН*, том 29, вып. 4, 2017 г., стр. 107-122 (на английском языке). DOI: 10.15514/ISPRAS-2017-29(4)-7

Список литературы

[1]. 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.

[2]. H. Thimbleby, "The directed Chinese Postman Problem," Software Practice and Experience, vol. 33, no. 11, pp. 1081-1096, 2003.

[3]. Mei-Ko Kwan, «Graphic programming using odd or even points» Acta Mathematica Sinica, p. 263–266, 1960.

[4]. 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.

[5]. F. L. Pimentel, «Double-ended nearest and loneliest neighbour–a nearest neighbour heuristic variation for the travelling salesman problem» Revista de Ciências da Computação, т. 6, № 6, 2016.

[6]. P. Vreda and P. Black, Dictionary of Algorithms and Data Structures, National Institute of Standards and Technology, 2014.

[7]. G. Laporte, «Modeling and solving several classes of arc routing problems as traveling salesman problems» Computers & operations research, т. 24, № 11, pp. 1057-1061, 1997.

[8]. S. Bönisch, «Implementierung der Edmonds-Johnson Heuritik für das Mixed Chinese Postman Problem» 21 December 1999.

[9]. Á. Corberán, "Arc Routing Problems: Data Instances". http://www.uv.es/corberan/instancias.htm. [Дата обращения 03.04.2017].