



HAL
open science

The Implementation of Eulerian Coverings of a Graph for Solving Routing Problems

Tatiana A. Makarovskikh

► **To cite this version:**

Tatiana A. Makarovskikh. The Implementation of Eulerian Coverings of a Graph for Solving Routing Problems. IFIP International Conference on Advances in Production Management Systems (APMS), Sep 2021, Nantes, France. pp.51-58, 10.1007/978-3-030-85914-5_6 . hal-03897879

HAL Id: hal-03897879

<https://inria.hal.science/hal-03897879>

Submitted on 14 Dec 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

The Implementation of Eulerian Coverings of a Graph for Solving Routing Problems^{*}

Tatiana A. Makarovskikh^{1[0000-0002-3656-9632]}

South Ural State University, 76, Lenin ave., Chelyabinsk, 454080, Russia
Makarovskikh.T.A@susu.ru

Abstract. In terms of graph theory, the urban transport network can be adequately represented as a graph. All crossroads in the city are considered to be the vertices of the graph, and streets are the edges. Using this model, one can set and effectively solve many routing problems: (1) constructing the routes between given starting and ending points, satisfying the rules of the road (allowed turns at crossroads and a given sequence of driving through the streets), (2) finding the way, passing through all the streets of the city in accordance with the rules of the road and a given sequence of turns at crossroads (an algorithm for constructing a permissible Eulerian chain), (3) constructing a minimal cardinality set of routes running through all the streets of the city (Eulerian cover of any graph with permissible chains). The paper provides a formalization of these problems and effective algorithms for solving them.

Keywords: Graphs · Trajectory planning · Polynomial algorithm · Transportation control.

1 Introduction

Various problems of management and automation design often lead to solving the routing problem. So, for example, one of the most actively explored areas is a mathematical model based on choosing the optimal route between different objects. These objects can be considered to be the vertices of a directed graph, and the optimal routes in this case will represent the shortest paths in the graph, which is a mathematical model of the considered problem. [1] considers the problem of modelling the closed queuing networks, for example, the placement of bicycle parking lots in the city, where parking lots are defined as the vertices of a directed graph, and possible paths between them be weighted arcs. When planning and managing the choice of a delivery route, it is possible to solve a problem based on representing a set of typical system states as graph vertices, the routes of this graph correspond to the control decisions of the fuzzy situational network.

Let's consider the solution of the routing problem solved by [2]. It requires the distribution of transport and passenger flows and takes into account the specifics

^{*} The work was supported by the Ministry of Science and Higher Education of the Russian Federation (government order FENU-2020-0022).

of driving in cities. To solve it, a correct description of the behaviour of the passenger (or the car) when choosing a route is required. Many factors influence his behaviour. For example, a special graph may be used to model the system of all possible paths within the city. Such a graph can be called communication graph. It is a superposition of the following subgraphs: subway routes, railway, pedestrian paths, highways, etc. Such a system of all possible displacements is not static, therefore it can be noted that all edges of the communication graph routes are characterized by the moments of creation and marking for deletion.

Routing problems complicated by various constraints are of the greatest theoretical interest. There are various methods for solving such problems and various constraints, for example, considered by [3], and [4].

In terms of graph theory, the transport network can be represented as a graph $G = (V, E)$, where the set of vertices V is the set of crossroads (cities, airports, warehouses, etc.), and the set of edges E is a set of connections between objects from V (roads, highways, flights, etc.). Thus, if there is an edge $e = \{v_1, v_2\}$, then there is a direct message between the objects corresponding to v_1 and v_2 . The solution of the routing problem in the transport network assumes (1) construction of routes between the given starting and ending points, satisfying the traffic rules (allowed turns at crossroads and a given sequence of travel along the streets), (2) finding a path passing through all the city streets in accordance with the traffic rules and a given sequence of turns at crossroads (an algorithm for constructing a permissible Euler chain), (3) constructing a minimal cardinality set of routes passing through all the streets of the city (Euler's cover of an arbitrary graph by permissible chains). As mentioned above, the most interesting are routing problems with constraints, since the constraints introduced allow us to describe a fairly narrow class of applied problems. Since the graph corresponding to the transport network has a sufficiently large dimension, the most urgent are algorithms that make it possible to obtain a solution (or a feasible solution) in the shortest possible time. The preferred algorithms are the ones that solve the problem in polynomial time (if they exist for the declared class of problems).

Constraints on constructing of routes in graphs can be classified as local (when a condition is set at a fixed vertex or edge) and global (a constraint is set for the graph as a whole). Problems with **local constraints** include the problem of constructing a route that satisfies a fixed system of allowed transitions. In terms of the transport routing problem, the problem statement can be interpreted as follows. All crossroads of the city roads are considered as the vertices of the graph, and streets are considered as edges. It is necessary to get a route between the starting and ending points that meets the rules of the road (allowed turns at crossroads) or the specified sequence of driving along the streets.

To solve these problems we define the following graphs: (1) the graph corresponding to the map of the area, in which each vertex $v \in V$ corresponds to a crossroad, (2) transition graphs $T_G(v)$ defined at each vertex $v \in V$. The vertices of the transition graphs are edges incident to $v \in V$, i.e. $V(T_G(v)) = E_G(v)$, and

the edges define the allowed transitions (the transitions correspond to the given traffic rules at a particular crossroad).

2 Constructing the Permissible Path between Two Vertices

The constraints on routes in the graph G can be formulated in terms of the graph of allowed transitions. These definitions are given by [4]. Let us formulate them for completeness.

Definition 1. *Let G be a graph. **Transitions graph** $T_G(v)$ for vertex $v \in V(G)$ be a graph, the vertices of which are the edges incident to vertex v , i.e. $V(T_G(v)) = E_G(v)$, and set of edges be the allowed transitions. **The system of allowed transitions** (or the **transitions system** for short) T_G be the set $\{T_G(v) \mid v \in V(G)\}$, where $T_G(v)$ be the transitions graph for v . Path $P = v_0e_1v_1 \dots e_kv_k$ in G be **T_G -compatible**, if $\{e_i, e_{i+1}\} \in E(T_G(v_i))$ for each i ($1 \leq i \leq k-1$).*

The following theorem proved by [5] holds for a problem of constructing the T_G -compatible chain, i.e. simple chain $C = v_0e_1v_1 \dots e_kv_k$ in graph G , for which $\{e_i, e_{i+1}\} \in E(T_G(v_i))$ for each i ($1 \leq i \leq k-1$).

Theorem 1. *. If all transition graphs belong to class M of complete multipartite graphs, or class P of matchings, then the problem of constructing the T_G -compatible chain may be solved by time $O(|E(G)|)$. Otherwise, this problem is \mathcal{NP} -complete.*

If transitions system for vertex $v \in V(G)$ is a matching then the problem can be reduced to a problem for graph

$$G' : V(G') = V(G) \setminus \{v\}, \\ E(G') = (E(G) \setminus E_G(v)) \cup \{\{v_i, v_j\} : \{\{v_i, v\}; \{v, v_j\}\} \in E(T_G(v))\}.$$

If $T_G(v)$ for any vertex $v \in V(G)$ is a complete multipartite graph then to get a permissible chain we may use algorithm **T_G -COMPATIBLE PATH** [4].

Note that in the general, the direct application of this algorithm does not allow to solve the problem of finding a T_G -compatible route containing the maximum number of edges. Note that **T_G -COMPATIBLE PATH** algorithm cannot be used to construct routes that cover all edges of graph G .

Indeed, a maximum cardinality matching in the graph G' cannot contain a pair of edges forming a forbidden transition, since they are incident to one common vertex of graph G' . At the same time, in the general, there may be a T_G -compatible route containing such a pair of edges. Note that [5] does not consider the issue of recognizing the multipartition of graphs $T_G(v)$, as well as the problem of constructing an permissible route or a set of routes covering all edges of the original graph. This recognition is rather trivial, and it is advisable to use the concept of a partition system introduced by [6] and [7].

3 Algorithm for Constructing the Permissible Eulerian Chain

As noted, algorithm by [5] generally does not allow constructing compatible chains of maximum length. Compatible Eulerian chains are of particular interest. A necessary and sufficient condition for the existence of P_G -compatible chains is given by [7]. The complexity of checking the existence of P_G -compatible Euler chain is not greater than $O(|E(G)|)$.

To define P_G -compatible chain in terms of forbidden transitions the term of partition system is used.

Definition 2. Let $G = (V, E)$ be a graph. Let $P_G(v)$ be some partition of set $E_G(v)$. **The partition system** of G be a system of sets $P_G := \{P_G(v) \mid v \in V(G)\}$. Let $p \in P_G(v)$, $\{e, f\} \in p$. Chain not containing the transitions $e \rightarrow v \rightarrow f$ and $f \rightarrow v \rightarrow e$ be called P_G -**compatible**, and transitions $e \rightarrow v \rightarrow f$ and $f \rightarrow v \rightarrow e$ be **forbidden**.

Note that the graph of allowed transitions $T_G(v)$ uniquely determines the graph of forbidden transitions $\overline{T_G}(v)$ and vice versa. Obviously, $T_G(v) \cup \overline{T_G}(v) = K_{|V(G)|}$, i.e. the forbidden transition graph is the complement of the allowed transition graph to a complete graph. Thus, using the definition of 1, one can pose a problem with any graph of allowed (forbidden) transitions.

On the contrary, the graph of allowed transitions, defined using the partition system P_G , cannot be arbitrary, but belongs to the class M of complete multipartite graphs: the elements of the partition $P_G(v)$ determine the parts of the graph $T_G(v) \in M$, and the set of its edges

$$E(T_G(v)) = \{e, f \in E_G(v) : (\forall p \in P_G(v)) \{e, f\} \not\subset p\}.$$

In this case, the graph of forbidden transitions $\overline{T_G}(v)$ will be a set of $|P_G(v)|$ cliques, this fact can be used to recognize the belonging of $T(v)$ to M .

Thus, a partitioning system into subsets can be given on $E(v)$ (the set of edges incident to the vertex v). If the edges e_1 and e_2 belong to the same subset, then the edge e_2 cannot follow the edge e_1 . The graph $G(V, E)$ will be defined as an adjacency list, the elements of which are structures. Each element of this structure consists of two fields: (1) numbers of the vertex v_i , adjacent to the current vertex; (2) the number c_i of the split element. Note that each edge e of the graph belongs to two adjacency lists: for vertices v_i and v_j , which are the ends of this edge. But for each vertex the edge e will belong to different partition systems.

To construct the compatible Eulerian chain we may use P_G -COMPATIBLE EULER CHAIN algorithm [4] that solves the problem of obtaining $P(G)$ -compatible Euler chain by time $O(|E(G)| \cdot |V(G)|)$. This algorithm can be easily implemented using standard computational tools.

4 Eulerian Cover of a Graph by Permissible Chains

Let us consider the problem of constructing an Euler covering of a graph with compatible chains. We assume that the transition system T_G contains only matchings and complete multipartite graphs [4].

Definition 3. *Let the Euler cover be the minimal cardinality set of edge-disjoint paths, covering the set of edges of a given graph.*

In this case, the graph is completed to be Euler. One way to reduce a graph to Euler is given by [4]. The considered algorithm 1 for constructing a covering by feasible chains consists in introducing a fictive vertex v^* and connecting it with all odd vertices with fictitious edges. [4] proved that in this case it is possible to solve the problem in linear time. As a result, $l + 1$ chain covering the original graph will be constructed. Here $l = \deg(v^*)$.

Algorithm 1 COVERING BY T_G -COMPATIBLE CHAINS

Require: : $G = (V, E)$, transitions graphs $T_G(v) \forall v \in V(G)$.

Ensure: : a set of chains $T^i, i = 1, 2, \dots, k$, covering graph $G, m = 2k$ is the number of odd vertices.

- 1: $U = \{v \in V(G) : T_G(v) \text{ is matching}\};$ ▷ **Step 1**
 - 2: $V(G') = V(G) \setminus U,$ ▷ Reduce G to G'
 - 3:

$$E(G') = \left(E(G) \setminus \bigcup_{v \in U} E_G(v) \right) \cup \left\{ \bigcup_{v \in U} \{ \{v_i v_j\} : \{v_i v, v v_j\} \in T_G(v) \} \right\};$$
 - 4: $T_{G'}(v) = \text{Reduce}(T_G(v), U)$ ▷ Replace all entries of $u \in U: vu, wu \in E_{T_G}(u)$ by w
 - 5: $G^* = \text{AddSupplementary}(G', v^*);$ ▷ **Step 2.**
 - 6: ▷ Introduce a supplementary vertex v^* adjacent to all odd vertices of G'
 - 7: $T_{G^*} = \text{Modify}(T_{G'}(v));$ ▷ Introduce vertex vv^* adjacent to all vertices in the graph $T_{G^*}(v)$ for all $v \in V'(G) : \deg(v) \equiv 1 \pmod{2}$ into the transition graph $T_{G^*}(v)$.
 - 8: **for all** $v \in V(G): \exists p \in P(v): |p| > \deg(v)/2$ **do** ▷ **Step 3.**
 - 9: **for all** $i = 1, 2, \dots, 2|p| - \deg(v)$ **do**
 - 10: **AddEdge** $(G^*, (vv^*)_i);$ ▷ introduce $2|p| - \deg(v)$ supplementary edges
 - 11: **Modify** $(T_{G^*}((vv^*)_i));$ ▷ Modify the transitions graph
 - 12: **end for**
 - 13: **end for**
 - 14: $T_{G^*} = P_{G^*}\text{-COMPATIBLE EULER CHAIN}(G^*)$ ▷ **Step 4.** Obtain compatible cycle
 - 15: **for all** $v \in V(G)$ **do** ▷ **Step 5**
 - 16: $T'(G') = \text{Delete}((vv^*))$ ▷ Delete supplementary edges from T^* and obtain T'
 - 17: **end for**
 - 18: **for all** $u \in U$ **do** ▷ **Step 6**
 - 19: $T = \text{Modify}(T');$ ▷ Modify paths by adding the vertices deleted at Step 1.
 - 20: **end for**
-

Theorem 2. *Algorithm 1 COVERING BY T_G -COMPATIBLE CHAINS correctly solves the problem of constructing the Euler cover of graph G by T_G -compatible chains. Its computing complexity is not greater than $O(|E(G)| \cdot |V(G)|)$.*

In terms of the transport routing problem, this problem can be interpreted in two ways: (1) the constructed cover is a set of permissible routes in the city, and the vertex v^* is a depot; (2) the constructed cover is a solution of well-known applied problems when certain constraints are put on the sequence of passed streets/corridors.

5 Example

Let's consider the simple but visual example of constructing the cover by T_G -compatible chains for graph in figure 1. The real life examples (maps of the cities where vertices are cross-roads, and edges are streets) have $10^4 - 10^9$ vertices and edges, and they are the subject of interest for another research concerning minimization of calculation time. So, the graph in figure 1 has odd vertices

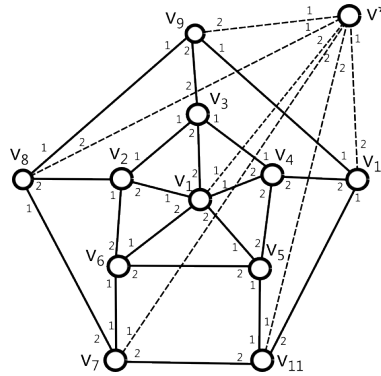


Fig. 1. Graph G with 6 odd vertices, partitions for each vertex, supplementary vertex v^* , and supplementary edges adjacent to it

$V_{odd} = \{v_1, v_7, v_8, v_9, v_{10}, v_{11}\}$. Hence, Euler cover contains three chains. Let us introduce the supplementary vertex v^* and construct the edges incident to it, the second ends of these edges be vertices

$$\begin{aligned} v \in V_{odd} : E_{odd} = \\ = \{ \{v^*, v_1\}, \{v^*, v_7\}, \{v^*, v_8\}, \{v^*, v_9\}, \{v^*, v_{10}\}, \{v^*, v_{11}\} \}. \end{aligned}$$

Thus, we obtain graph $G^* = G(V \cup V_{odd}, E \cup E_{odd})$. According to transitions system T_G we complete transitions system T_{G^*} according to Step 2 of algorithm

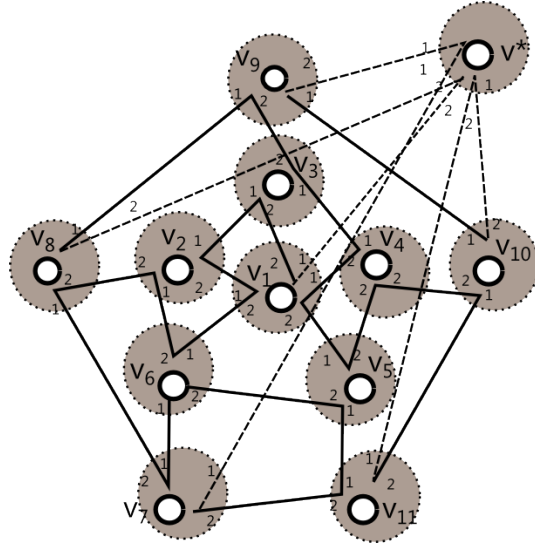


Fig. 2. Graph G^* all vertices of which are split according to the given transitions system. The gray circles round the split vertices

1. Splitting of graph G^* vertices according to T_{G^*} is given in figure 2. Hence, we obtain T_G -compatible Euler cycle for G^* by algorithm P_G -COMPATIBLE EULER CHAIN.

$$(G^*) = \{v_8 v_9 v_3 v_4 v_1 v_5 v_4 v_{10} v_{11} v^* v_{10} v_9 v^* v_7 v_{11} v_5 v_6 v_7 v_8 v_2 v_6 v_1 v_2 v_3 v_1 v^* v_8\}.$$

When we delete edges $e \in E_{odd}$ from $C(G^*)$ we get the cover of initial graph G by compatible chains

$$C_1 = \{v_8 v_9 v_3 v_4 v_1 v_5 v_4 v_{10} v_{11}\}, \quad C_2 = \{v_{10} v_9\}, \\ C_3 = \{v_7 v_{11} v_5 v_6 v_7 v_8 v_2 v_6 v_1 v_2 v_3 v_1\}.$$

Nevertheless, additional edges in the graph can also be introduced in accordance with additional constraints, for example, when solving a traffic problem, these edges may represent a planar extension and/or duplicate the existing edges. In this case, the number of constructed chains depends on the number of additional edges. So, if the graph from the considered example (Fig. 1) is a homeomorphic image of some area map, then the following auxiliary routes will be obtained between the chains C_i , $i = 1, 2, 3$: $C_{1,2} = v_{11} v_{10}$, $C_{2,3} = v_9 v_8 v_7$ (or $C_{2,3} = v_9 v_{10} v_{11} v_7$). In general, algorithm T_G -COMPATIBLE PATH to construct permissible chains between pairs of odd vertices.

In terms of the transport routing problem, this problem can be interpreted in two ways:

- the constructed cover is a set of permissible routes in the city, and the vertex v^* is a depot;
- the constructed cover is a solution of well-known applied problems when certain constraints are put on the sequence of passed streets/corridors.

6 Conclusion

The known algorithms make it possible to construct a simple chain between two different vertices that satisfies local constraints. The paper shows the possibility of recognizing a transition system, which allows solving the problem of constructing a compatible path in linear time. It is proved that using the developed algorithm *P_G -COMPATIBLE EULER CHAIN* for Euler graph G it is possible to obtain a P_G -compatible Euler cycle or to establish its absence in time $O(|V(G)| \cdot |E(G)|)$. Covering the graph G with compatible chains is also possible in time $O(|V(G)| \cdot |E(G)|)$ using algorithm *COVERING T_G -COMPATIBLE CHAINS*. The considered algorithms can be used, for example, to solve routing problems in transport, when the dimension of the problem is large enough and it is necessary to use algorithms that solve the problem in the shortest time. The directions for further research is the development of C++-library that implements the construction of routes with the local constraints discussed here.

References

1. Giuseppe, C., Portigliotti, C.F., Rizzo A.: A Network Model for an Urban Bike Sharing System. IFAC-PapersOnLine, vol. 50, issue 1, pp. 15633–15638.
2. Panyukov, A.V., Pivovarova, Yu.V., Chaloob, Kh.: Mathematical Model for Solving Operational Problem of Regional Cargo Transportation. Journal of Computational and Engineering Mathematics, vol. 6, pp. 68–73.
3. Chentsov, A.G., Chentsov, P.A.: Routing Under Constraints: Problem of Visit to Megalopolises. Automation and Remote Control, vol. 77, no. 11, pp. 1957–1974.
4. Panyukova, T., Alferov, I.: The Software for Constructing Trails with Local Restrictions in Graphs. Open Journal of Discrete Mathematics, vol. 3, no. 2, pp. 86–92.
5. Szeider, S.: Finding Paths in Graphs Avoiding Forbidden Transitions. Discrete Applied Mathematics, no. 126, pp. 261–273.
6. Fleischner, H.: Eulerian Graphs and Related Topics. Part 1, vol.1. Ann. Discrete Mathematics, no. 45 (1990).
7. Kotzig, A.: Moves Without Forbidden Transitions in a Graph. Mat.-Fiz. Casopis 18, no. 1, pp. 76–80.