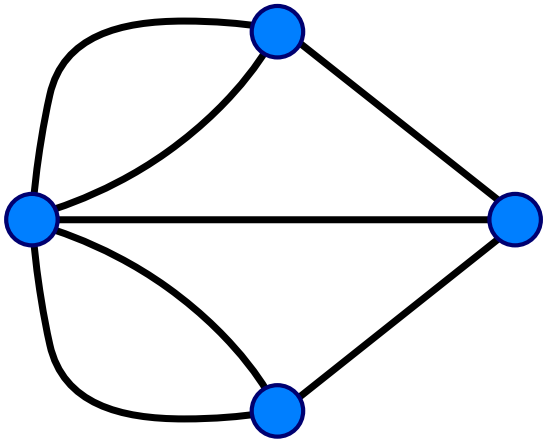
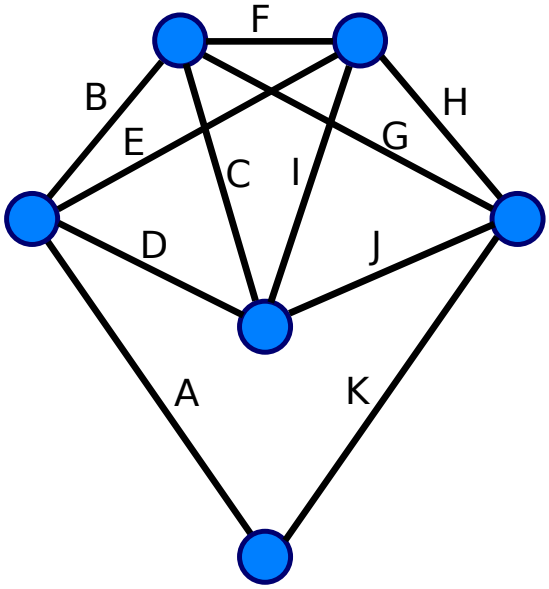
Eulerian path



The Königsberg Bridges graph. This graph is not Eulerian, therefore, a solution does not exist.



Every vertex of this graph has an even degree. Therefore, this is an Eulerian graph. Following the edges in alphabetical order gives an Eulerian circuit/cycle.

In graph theory, an **Eulerian trail** (or **Eulerian path**) is a trail in a graph which visits every edge exactly once. Similarly, an **Eulerian circuit** or **Eulerian cycle** is an Eulerian trail which starts and ends on the same vertex. They were first discussed by Leonhard Euler while solving the famous Seven Bridges of Königsberg problem in 1736. The problem can be stated mathematically like this:

> Given the graph in the image, is it possible to construct a path (or a cycle, i.e. a path starting and ending on the same vertex) which visits each edge exactly once?

Euler proved that a necessary condition for the existence of Eulerian circuits is that all vertices in the graph have an even degree, and stated without proof that connected graphs with all vertices of even degree have an Eulerian circuit. The first complete proof of this latter claim was published posthumously in 1873 by Carl Hierholzer.[1]

The term **Eulerian graph** has two common meanings in graph theory. One meaning is a graph with an Eulerian circuit, and the other is a graph with every vertex of even degree. These definitions coincide for connected graphs.[2]

For the existence of Eulerian trails it is necessary that zero or two vertices have an odd degree; this means the Königsberg graph is *not* Eulerian. If there are no vertices of odd degree, all Eulerian trails are circuits. If there are exactly two vertices of odd degree, all Eulerian trails start at one of them and end at the other. A graph that has an Eulerian trail but not an Eulerian circuit is called **semi-Eulerian**.

1 Definition

An **Eulerian trail**,[3] or **Euler walk** in an undirected graph is a walk that uses each edge exactly once. If such a walk exists, the graph is called **traversable** or **semi-eulerian**.[4]

An **Eulerian cycle**,[3] **Eulerian circuit** or **Euler tour** in an undirected graph is a cycle that uses each edge exactly once. If such a cycle exists, the graph is called **Eulerian** or **unicursal**.[5] The term "Eulerian graph" is also sometimes used in a weaker sense to denote a graph where every vertex has even degree. For finite connected graphs the two definitions are equivalent, while a possibly unconnected graph is Eulerian in the weaker sense if and only if each connected component has an Eulerian cycle.

For directed graphs, "path" has to be replaced with *directed path* and "cycle" with *directed cycle*.

The definition and properties of Eulerian trails, cycles and graphs are valid for multigraphs as well.

An **Eulerian orientation** of an undirected graph G is an assignment of a direction to each edge of G such that, at each vertex v, the indegree of v equals the outdegree of

v. Such an orientation exists for any undirected graph in which every vertex has even degree, and may be found by constructing an Euler tour in each connected component of G and then orienting the edges according to the tour.[6] Every Eulerian orientation of a connected graph is a strong orientation, an orientation that makes the resulting directed graph strongly connected.

2 Properties

- An undirected graph has an Eulerian cycle if and only if every vertex has even degree, and all of its vertices with nonzero degree belong to a single connected component.
- An undirected graph can be decomposed into edge-disjoint cycles if and only if all of its vertices have even degree. So, a graph has an Eulerian cycle if and only if it can be decomposed into edge-disjoint cycles and its nonzero-degree vertices belong to a single connected component.
- An undirected graph has an Eulerian trail if and only if exactly zero or two vertices have odd degree, and if all of its vertices with nonzero degree belong to a single connected component.
- A directed graph has an Eulerian cycle if and only if every vertex has equal in degree and out degree, and all of its vertices with nonzero degree belong to a single strongly connected component. Equivalently, a directed graph has an Eulerian cycle if and only if it can be decomposed into edge-disjoint directed cycles and all of its vertices with nonzero degree belong to a single strongly connected component.
- A directed graph has an Eulerian trail if and only if at most one vertex has (out-degree) − (in-degree) = 1, at most one vertex has (in-degree) − (out-degree) = 1, every other vertex has equal in-degree and out-degree, and all of its vertices with nonzero degree belong to a single connected component of the underlying undirected graph.

3 Constructing Eulerian trails and circuits

3.1 Fleury's algorithm

Fleury's algorithm is an elegant but inefficient algorithm which dates to 1883.[7] Consider a graph known to have all edges in the same component and at most two vertices of odd degree. The algorithm starts at a vertex of odd degree, or, if the graph has none, it starts with an arbitrarily chosen vertex. At each step it chooses the next edge in the path to be one whose deletion would not disconnect the graph, unless there is no such edge, in which case it picks the remaining edge left at the current vertex. It then moves to the other endpoint of that vertex and deletes the chosen edge. At the end of the algorithm there are no edges left, and the sequence from which the edges were chosen forms an Eulerian cycle if the graph has no vertices of odd degree, or an Eulerian trail if there are exactly two vertices of odd degree.

While the *graph traversal* in Fleury's algorithm is linear in the number of edges, i.e. $O(|E|)$, we also need to factor in the complexity of detecting bridges. If we are to re-run Tarjan's linear time bridge-finding algorithm after the removal of every edge, Fleury's algorithm will have a time complexity of $O(|E|^2)$. A dynamic bridge-finding algorithm of Thorup (2000) allows this to be improved to $O(|E| \log^3 |E| \log \log |E|)$ but this is still significantly slower than alternative algorithms.

3.2 Hierholzer's algorithm

Hierholzer's 1873 paper provides a different method for finding Euler cycles that is more efficient than Fleury's algorithm:

- Choose any starting vertex v, and follow a trail of edges from that vertex until returning to v. It is not possible to get stuck at any vertex other than v, because the even degree of all vertices ensures that, when the trail enters another vertex w there must be an unused edge leaving w. The tour formed in this way is a closed tour, but may not cover all the vertices and edges of the initial graph.
- As long as there exists a vertex u that belongs to the current tour but that has adjacent edges not part of the tour, start another trail from u, following unused edges until returning to u, and join the tour formed in this way to the previous tour.

By using a data structure such as a doubly linked list to maintain the set of unused edges incident to each vertex, to maintain the list of vertices on the current tour that have unused edges, and to maintain the tour itself, the individual operations of the algorithm (finding unused edges exiting each vertex, finding a new starting vertex for a tour, and connecting two tours that share a vertex) may be performed in constant time each, so the overall algorithm takes linear time, $O(|E|)$.[8]

4 Counting Eulerian circuits

4.1 Complexity issues

The number of Eulerian circuits in *digraphs* can be calculated using the so-called **BEST theorem**, named after de

Bruijn, van Aardenne-**E**hrenfest, **S**mith and **T**utte. The formula states that the number of Eulerian circuits in a digraph is the product of certain degree factorials and the number of rooted arborescences. The latter can be computed as a determinant, by the matrix tree theorem, giving a polynomial time algorithm.

BEST theorem is first stated in this form in a "note added in proof" to the Aardenne-Ehrenfest and de Bruijn paper (1951). The original proof was bijective and generalized the de Bruijn sequences. It is a variation on an earlier result by Smith and Tutte (1941).

Counting the number of Eulerian circuits on *undirected* graphs is much more difficult. This problem is known to be #P-complete.[9] In a positive direction, a Markov chain Monte Carlo approach, via the *Kotzig transformations* (introduced by Anton Kotzig in 1968) is believed to give a sharp approximation for the number of Eulerian circuits in a graph, though as yet there is no proof of this fact (even for graphs of bounded degree).

4.2 Special cases

The asymptotic formula for the number of Eulerian circuits in the complete graphs was determined by McKay and Robinson (1995):[10]

$$ec(K_n) = 2^{(n+1)/2}\pi^{1/2}e^{-n^2/2+11/12}n^{(n-2)(n+1)/2}\left(1+O(n^{-1/2+\epsilon})\right).$$

A similar formula was later obtained by M.I. Isaev (2009) for complete bipartite graphs:[11]

$$ec(K_{n,n}) = (n/2-1)!^{2n}2^{n^2-n+1/2}\pi^{-n+1/2}n^{n-1}\left(1+O(n^{-1/2+\epsilon})\right).$$

5 Applications

Eulerian trails are used in bioinformatics to reconstruct the DNA sequence from its fragments.[12] They are also used in CMOS circuit design to find an optimal logic gate ordering.[13] There are some algorithms for processing trees that rely on an Euler tour of the tree (where each edge is treated as a pair of arcs).[14][15]

6 See also

- Eulerian matroid, an abstract generalization of Eulerian graphs
- Five room puzzle
- Handshaking lemma, proven by Euler in his original paper, showing that any undirected connected graph has an even number of odd-degree vertices
- Hamiltonian path – a path that visits each *vertex* exactly once.
- Route inspection problem, search for the shortest path that visits all edges, possibly repeating edges if an Eulerian path does not exist.
- Veblen's theorem, that graphs with even vertex degree can be partitioned into edge-disjoint cycles regardless of their connectivity

7 Notes

[1] N. L. Biggs, E. K. Lloyd and R. J. Wilson, Graph Theory 1736–1936, Clarendon Press, Oxford, 1976, 8–9, ISBN 0-19-853901-0.

[2] C. L. Mallows, N. J. A. Sloane (1975). "Two-graphs, switching classes and Euler graphs are equal in number". *SIAM Journal on Applied Mathematics*. **28** (4): 876–880. doi:10.1137/0128070. JSTOR 2100368.

[3] Some people reserve the terms *path* and *cycle* to mean *non-self-intersecting* path and cycle. A (potentially) self-intersecting path is known as a **trail** or an **open walk**; and a (potentially) self-intersecting cycle, a **circuit** or a **closed walk**. This ambiguity can be avoided by using the terms Eulerian trail and Eulerian circuit when self-intersection is allowed.

[4] Jun-ichi Yamaguchi, Introduction of Graph Theory.

[5] Schaum's outline of theory and problems of graph theory By V. K. Balakrishnan .

[6] Schrijver, A. (1983), "Bounds on the number of Eulerian orientations", *Combinatorica*, **3** (3-4): 375–380, doi:10.1007/BF02579193, MR 729790.

[7] Fleury, M. (1883), "Deux problèmes de Géométrie de situation", *Journal de mathématiques élémentaires*, 2nd ser. (in French), **2**: 257–261.

[8] Fleischner, Herbert (1991), "X.1 Algorithms for Eulerian Trails", *Eulerian Graphs and Related Topics: Part 1, Volume 2*, Annals of Discrete Mathematics, **50**, Elsevier, pp. X.1–13, ISBN 978-0-444-89110-5.

[9] Brightwell and Winkler, "Note on Counting Eulerian Circuits", 2004.

[10] Brendan McKay and Robert W. Robinson, Asymptotic enumeration of eulerian circuits in the complete graph, *Combinatorica*, 10 (1995), no. 4, 367–377.

[11] M.I. Isaev (2009). "Asymptotic number of Eulerian circuits in complete bipartite graphs". *Proc. 52-nd MFTI Conference* (in Russian). Moscow: 111–114.

[12] Pevzner, Pavel A.; Tang, Haixu; Waterman, Michael S. (2001). "An Eulerian trail approach to DNA fragment assembly". *Proceedings of the National Academy of Sciences of the United States of America*. **98** (17): 9748–9753. Bibcode:2001PNAS...98.9748P. doi:10.1073/pnas.171285098. PMC 55524. PMID 11504945.

[13] Roy, Kuntal (2007). "Optimum Gate Ordering of CMOS Logic Gates Using Euler Path Approach: Some Insights and Explanations". *Journal of Computing and Information Technology*. **15** (1): 85–92. doi:10.2498/cit.1000731.

[14] Tarjan, Robert E.; Vishkin, Uzi (1985). "An efficient parallel biconnectivity algorithm". *SIAM Journal on Computing*. **14** (4): 862–874. doi:10.1137/0214061.

[15] Berkman, Omer; Vishkin, Uzi (Apr 1994). "Finding level-ancestors in trees". *J. Comput. Syst. Sci.* 2. **48**: 214–230. doi:10.1016/S0022-0000(05)80002-9.

8 References

- Euler, L., "Solutio problematis ad geometriam situs pertinentis", *Comment. Academiae Sci. I. Petropolitanae* **8** (1736), 128–140.
- Hierholzer, Carl (1873), "Ueber die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechung zu umfahren", *Mathematische Annalen*, **6** (1): 30–32, doi:10.1007/BF01442866.
- Lucas, E., *Récréations Mathématiques IV*, Paris, 1921.
- Fleury, "Deux problemes de geometrie de situation", *Journal de mathematiques elementaires* (1883), 257–261.
- T. van Aardenne-Ehrenfest and N. G. de Bruijn (1951) "Circuits and trees in oriented linear graphs", Simon Stevin 28: 203–217.
- Thorup, Mikkel (2000), "Near-optimal fully-dynamic graph connectivity", *Proc. 32nd ACM Symposium on Theory of Computing*, pp. 343–350, doi:10.1145/335305.335345
- W. T. Tutte and C. A. B. Smith (1941) "On Unicursal Paths in a Network of Degree 4", American Mathematical Monthly 48: 233–237.

9 External links

- Discussion of early mentions of Fleury's algorithm.
- *Euler tour* at Encyclopedia of Mathematics.

10 Text and image sources, contributors, and licenses

10.1 Text

- **Eulerian path** *Source:* https://en.wikipedia.org/wiki/Eulerian_path?oldid=749191639 *Contributors:* Michael Hardy, Chris-martin, Bueller 007, Charles Matthews, Dysprosia, Doradus, Taxman, Zero0000, McKay, Robbot, Fredrik, Altenmann, MathMartin, Iosif~enwiki, Modeha, Giftlite, Herbee, Jason Quinn, Peter Kwok, Hanru~enwiki, Mh, Bender235, Rgdboer, GTubio, Obradovic Goran, Jumbuck, Arthena, Hippophaë~enwiki, Igorpak, Dan100, Brookie, Oliphaunt, Jacobolus, Pixeltoo, Trevor Andersen, Audiovideo, Graham87, Magister Mathematicae, Rjwilmsi, Nguyen Thanh Quang, Maxal, Vonkje, Eythian, Uni4dfx, PonyToast, John Sheu, Bota47, Arthur Rubin, Ilmari Karonen, Owl-syme, SmackBot, Danyluis, Gelingvistoj, Chris the speller, Bluebot, DHN-bot~enwiki, Mhym, Armend, Henning Makholm, Vina-iwbot~enwiki, Rschwieb, Frazz, Courcelles, CRGreathouse, Anakata, Myasuda, Eesnyder, Gogo Dodo, Boemanneke, Perebot~enwiki, David Eppstein, Martynas Patasius, Ricardogpn, CommonsDelinker, Ekarulf, Policron, Idioma-bot, Fr33kman, VolkovBot, Maxtremus, TXiKiBoT, Ohiostandard, Powerpanda, ClueBot, Jonasft, Alexbot, AmirOnWiki, SilvonenBot, Addbot, Dwyerj, Friginator, Tide rolls, Luckas-bot, Kilom691, KamikazeBot, Citation bot, Twri, Xqbot, Kinewma, Omnipaedista, FrescoBot, Artem M. Pelenitsyn, Lunae, DoostdarWKP, Ricardo Ferreira de Oliveira, At-par, RjwilmsiBot, Cbarrick1, Delio.mugnolo, WikitanvirBot, T3dkjn89q00vl02Cxp1kqs3x7, Bugthink, Tommy2010, Haterade111, CtrlAltDel121, DiSkip, Llaesrever, Red Bulls Fan, ClueBot NG, Wcherowi, El Roih, Rompelstompel777, Widr, Helpful Pixie Bot, Yatin.mirakhur, Arya84, Ynaamad, Odenkos, Chenzhixin1994, Markonator, Thakkarparth007 and Anonymous: 115

10.2 Images

- **File:Commons-logo.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/4/4a/Commons-logo.svg *License:* CC-BY-SA-3.0 *Contributors:* ? *Original artist:* ?
- **File:Königsberg_graph.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/9/96/K%C3%B6nigsberg_graph.svg *License:* CC-BY-SA-3.0 *Contributors:* ? *Original artist:* ?
- **File:Labelled_Eulergraph.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/7/72/Labelled_Eulergraph.svg *License:* Public domain *Contributors:* Own work *Original artist:* S Sepp
- **File:Lock-green.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/6/65/Lock-green.svg *License:* CC0 *Contributors:* en:File:Free-to-read_lock_75.svg *Original artist:* User:Trappist the monk

10.3 Content license

- Creative Commons Attribution-Share Alike 3.0