Lecture 10

Eulerian Multigraphs

This section of the notes revisits the Königsberg Bridge Problem and generalises it to explore *Eulerian multigraphs:* those that contain a closed walk that traverses every edge exactly once.

Reading:
The material in today's lecture comes from Section 1.3 of

> Dieter Jungnickel (2013), *Graphs, Networks and Algorithms*, 4th edition, (available online via SpringerLink),

though his proof is somewhat more terse.

In Lecture 1 we used a proof by contradiction to demonstrate that there is no solution to the Königsberg Bridge Problem, which is illustrated in Figure 10.1. That is, it's not possible to find a walk that (a) crosses each of the city's seven bridges exactly once and (b) starts and finishes in the same place. Today we'll generalise the problem, then find a number of equivalent conditions that tell us when the corresponding closed walk exists.

First, recall that a *multigraph* $G(V, E)$ has the same definition as a graph, except that we allow parallel edges. That is, we allow pairs of vertices (u, v) to appear more than once in E. Because of this, people sometimes speak of the *edge list* of a multigraph, as opposed to the *edge set*.

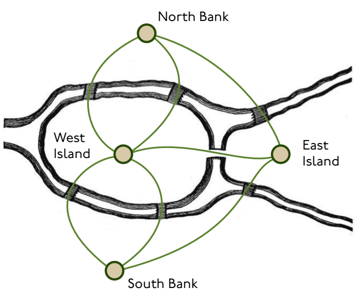

Figure 10.1: *We proved in the first lecture of the term that it is impossible to find a closed walk that traverses every edge in the graph above exactly once.*

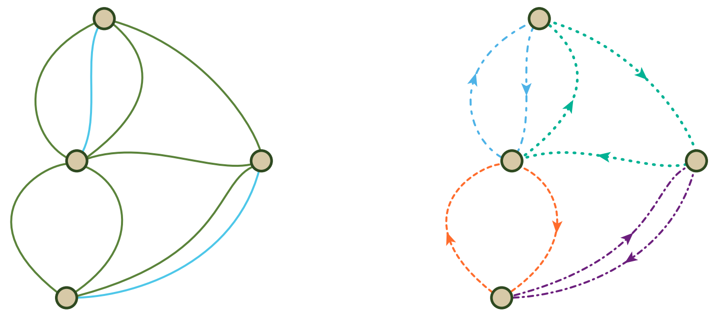

Figure 10.2: *The panel at left shows a graph produced by adding two edges (shown in blue) to the graph from the Königsberg Bridge Problem. These extra edges make the graph Eulerian and the panel at right illustrated the associated partition of the edge set into cycles*

The main theorem we'll prove today relies on the following definitions:

Definition 10.1. *An **Eulerian trail** in a multigraph $G(V, E)$ is a trail that includes each of the graph's edges exactly once.*

Definition 10.2. *An **Eulerian tour** in a multigraph $G(V, E)$ is an Eulerian trail that starts and finishes at the same vertex. Equivalently, it is a closed trail that traverses each of the graph's edges exactly once.*

Definition 10.3. *A multigraph that contains an Eulerian tour is said to be an **Eulerian multigraph**.*

Armed with these, it's then easy to formulate the following characterisation of Eulerian multigraphs:

Theorem 10.4 (Jungnickel's Theorem 1.3.1)**.** *Let G be a connected multigraph. Then the following statements are equivalent:*

(1) G is Eulerian.

(2) Each vertex of G has even degree.

(3) The edge set of G can be partitioned into cycles.

The last of these characterisations may be new to you: it means that it is possible to arrange the edges of G into a collection of disjoint cycles. Figure 10.2 shows an example of such a partition for a graph derived from the Königsberg Bridge multigraph by adding two extra edges, shown in blue at left. Adding these edges makes the graph Eulerian, and a decomposition of the edge set into cycles appears at right. Note that undirected multigraphs can contain cycles of length two that consist of a pair of parallel edges.

The proof of the theorem is simpler if one has the following lemma, whose proof I'll defer until after that of the main result. Note that the lemma, unlike the theorem, does not require the multigraph to be connected.

Lemma 10.5 (Vertices of even degree and cycles)**.** *If $G(V, E)$ is a multigraph with a nonempty edge set $E \neq \emptyset$ and the property that* $\deg(v)$ *is an even number for all $v \in V$, then G contains a cycle.*

Proof of Theorem 10.4. The theorem says these statements are all "equivalent", which encompasses a total of six implications[1] but we don't need to prove all of them: it's sufficient to prove, say, that (1) $\implies$ (2), (2) $\implies$ (3) and (3) $\implies$ (1). That is, it's sufficient to make a directed graph whose vertices are the statements and whose edges indicate implications. If this graph is strongly connected, so that one can get from any statement to any other by following a chain of implications, then the result is proven.

(1) $\implies$ (2):

Proof. We know G is Eulerian, so it has a closed trail that includes each edge exactly once. Imagine that this trail is specified by the following sequence of vertices

$$v_0, \ldots, v_m = v_0 \tag{10.1}$$

where $|E| = m$ and the v_j are the vertices encountered along the trail, so that some of them may appear more than once. In particular, $v_0 = v_m$ because the trail starts and finishes at the same vertex. As G is a connected multigraph, every vertex appears somewhere in the sequence (if not, the absent vertices would have degree zero and not be connected to any of the others).

Consider first some vertex $u \neq v_0$. It must appear one or more times in the sequence above and, each time, it appears in a pair of successive edges: if $u = v_j$ with $0 < j < m$, then these edges are (v_{j-1}, v_j) and (v_j, v_{j+1}). This means that $\deg(u)$ is a sum of 2's, with one term in the sum for each appearance of u in the sequence (10.1). A similar argument applies to v_0, save that the edge that forms a pair with (v_0, v_1) is $(v_{m-1}, v_m = v_0)$. □

(2) $\implies$ (3): The theorem requires this implication to hold for connected multigraphs, but this particular result is more general and applies to *any* multigraph in which all vertices have even degree. We'll prove this stronger version by induction on the number of edges. That is, we'll prove:

Proposition. *If $G(V, E)$ is a multigraph (whether connected or not) in which* $\deg(v)$ *is an even number for all vertices $v \in V$, then the edge set E can be partitioned into cycles.*

Proof. The base case is a multigraph with $|E| = 0$. Such a graph consists of one or more isolated vertices and, as the graph has no edges, $\deg(v) = 0$ (an even number) for all $v \in V$ and the (empty) edge set can clearly be partitioned into a union of zero cycles.

Now suppose the result is true for every multigraph $G(V, E)$ with $|E| \leq m_0$ edges whose vertices all have even degree. Consider such a multigraph with $|E| = m_0 + 1$: we need to demonstrate that the edge set of such a graph can be partitioned into

[1] (1) $\implies$ (2), (2) $\implies$ (1), (1) $\implies$ (3)...

cycles. We can use Lemma 10.5 to establish that we can find at least one cycle C contained in G. And then we can form a new graph $G'(V', E') = G \backslash C$ formed by removing C from G. This bit of graph surgery either leaves the degree of a vertex unchanged (if the vertex isn't part of C) or decreases it by two, but either way, all vertices in G' have even degree because the corresponding vertices in G do.

The cycle C will contain at least one edge (and, unless we permit self-loops, two or more) and so G' will have at most m_0 edges and so the inductive hypothesis will apply to it. This means that we can partition $E' = E \backslash C$ into cycles. But then we can add C to the partition of E' and so get a partition into cycles for E, completing the inductive step and so proving our result. □

(3) $\implies$ (1): Here we need to establish that if the edge set of a connected multigraph $G(V, E)$ consists of a union of cycles, then G contains an Eulerian tour. This result is trivial unless the partition of E involves at least two cycles, so we'll restrict attention to that case from now on.

The key observation is that we can always find two cycles that we can merge to produce a single, longer closed trail that includes all the edges from the two cycles. To see why, note that there must be a pair of cycles that share a vertex (if there weren't, all the cycles would all lie in distinct connected components, contradicting the connectedness of G). Suppose that the shared vertex is $v_\star$ and that the cycles are C_1 and C_2 given by the vertex sequences

$$C_1 = \{v_\star = v_0, v_1, \ldots, v_{\ell_1} = v_\star\} \qquad \text{and} \qquad C_2 = \{v_\star = u_0, u_1, \ldots, u_{\ell_2} = v_\star\}\,.$$

We can combine them, as illustrated in Figure 10.3 to make a closed trail given by the vertex sequence

$$\{v_\star = v_0, v_1, \ldots, v_{\ell_1} = v_\star = u_0, u_1, \ldots, u_{\ell_2} = v_\star\}\,.$$

Scrupulous readers may wish to use this observation as the basis of a proof by induction (on the number of elements in the partition of E) of the somewhat stronger result:

Proposition 10.6. *If $G(V, E)$ is a strongly-connected graph whose edge set can be partitioned as a union of disjoint, closed trails, then G is Eulerian.*

Then, as a cycle is a special case of a closed trail, we get the desired implication as an immediate corollary. □

I'd like to conclude by giving an algorithmic proof of Lemma 10.5. The idea is to choose some initial vertex u_0 and then construct a trail in the graph by following one of u_0's edges, then one of the edges of u_0's successor in the trail ... and so on until we revisit some vertex and thus discover a cycle. Provided that we can do as I say—always move on through the graph without ever tracing over some edge twice—this approach is bound to work because there are only finitely many vertices. The proof that follows formalises this approach by spelling out an explicit algortihm.

Proof of Lemma 10.5. Consider the following algorithmic process, which finds a cycle in a multigraph $G(V, E)$ for which $E \neq \emptyset$ and $\deg(v)$ is even for all $v \in V$.

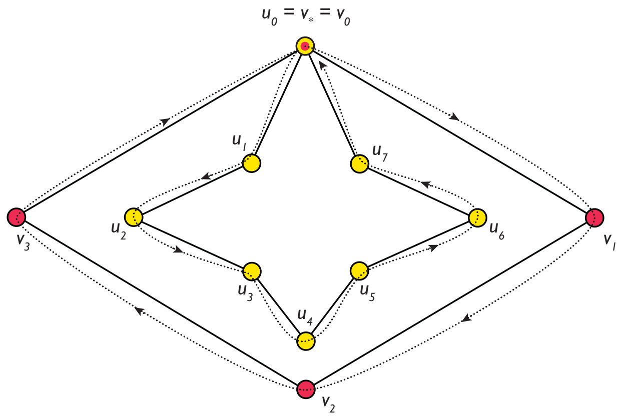

Figure 10.3: *The key step in the proof of the implication* (3) $\Longrightarrow$ (1) *in the proof of Theorem 10.4. The cycles* $C_1 = (v_\star = v_0, v_1, v_2, v_3, v_0)$, *whose vertices are shown in red, and* $C_2 = (v_\star = u_0, u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_0)$, *whose vertices are shown in yellow, may be merged to create the closed trail* $(v_\star = v_0, v_1, v_2, v_3, v_0 = v_\star = u_0, u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_0)$ *indicated by the dotted line.*

Algorithm 10.7 (Finding a cycle)**.**
Given a multigraph $G(V, E)$ *in which* $|E| > 0$ *and all vertices have even degree, construct a trail* T *given by a sequence of edges*

$$T = \{(u_0, u_1),\ (u_1, u_2),\ \ldots,\ (u_{\ell-1}, u_\ell)\}$$

that includes a cycle.

(1) Number the vertices, so that $V = \{v_1, \ldots, v_n\}$. *This is for the sake of concreteness: later in the algorithm, when we need to choose one of a set of vertices that have a particular property, we can choose the lowest-numbered one.*

(2) Initialize some things

- *Set a counter* $j \leftarrow 0$.
- *Choose the first vertex in the trail,* u_0, *to be the lowest-numbered vertex that has* $\deg(v_k) > 0$. *Such vertices exist, as we know* $|E| > 0$.
- *Initialise a list* A *(for "available") of edges that we have not yet included in* T. *At the outset we set* $A \leftarrow E$ *as we haven't used any edges yet.*

(3) Find the edge $(u_j, w) \in A$ *where* w *is the lowest-numbered neighbour of* u_j *whose edge we haven't yet used. The key to the algorithm's success is that this step is always possible. We chose* u_0 *with* $\deg(u_0) > 0$, *so this step is possible when* $j = 0$. *And when* $j > 0$, *the evenness of* $\deg(u_j)$ *means that if the trail we are constructing can arrive at* u_j, *then it must also be able to depart. The growing trail* T *either comes to a stop at* u_j *(see below) or uses a pair of the*

vertex's edges—one to arrive and another to depart—and so leaves an even number of unused edges behind.

We can thus always extend the trail T by one edge, modifying the list of unused edges A accordingly.

- $T \leftarrow T \cup \{(u_j, w)\}$
- $A \leftarrow A \backslash \{(u_j, w)\}$ (We've used (one copy of) the edge (u_j, w)).

(4) Are we finished? Does w already appear in the trail?

- *If yes, stop. The trail includes a cycle that starts and finishes at w.*
- *If no, set $u_{j+1} \leftarrow w$, then set $j \leftarrow j + 1$ and go to Step 3.*

The only way this process can stop is by revisiting a vertex and it must do this within $|V| = n$ steps. And once we've revisited a vertex, we've found a cycle and so are finished. $\square$