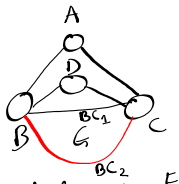
27 April 2023
MAT 2250

Ack.

Example.



Here the graph G has 2 vertices of odd degree ⇒ Hence cannot have an Euler circuit (Thm later).

But it has an Euler path.

Just add one more edge between the two vertices of odd degree. Now everything has even degree.

We can apply the algorithm with $u_0 = A$.

G': A → B → D → C → (not: now CA is a cut edge in the remaining edges) $CB_1 \to BC_2 \to CA$

To get back to G, need to take out CB_2 it leaves:

AB, BD, DC, CB_1, CA

⇒ this is not legit because CB_1 & CA has no common vertex. What to do is to make a cyclic permutation, starting from one of the vertex with odd degree:

CA, AB, BD, DC, CB, BC_1, CD, DB, BA, AC

Also the algorithm can be extended to directed multigraphs.

Thm 8.19: A connected graph $G = (V, E)$ has an Euler circuit if and only if all vertices of G has even degree.

Proof:

(⇒) Assume that G has an Euler circuit.

$u_0 \to u_1 \to \dots \to u_0$

Degree of u_i, $i \neq 0$ is obviously even

Because this is a circuit, we can make a shift to have u_1 at the beginning part ⇒ u_0 also has even degree.

(⇐) Assume that G is connected & every vertex has even degree. We will now show that the algorithm gives us an Euler circuit.

Now apply the algorithm & assume that it produces a path $W = u_0 \to \dots \to u_p$

First: $u_p = u_0$. (Because if not, then the degree of u_p in W is odd: obvious?) & since degree of u_p in G is even, we could add one more edge to W and hence make W bigger, a contradiction.

So:

$W = u_0 \to u_1 \to \dots \to u_p = u_0$.

Now we show that W contains every edges in G, and hence an Euler circuit for G.

$G_p = (V, E \setminus \{k_1, \dots, k_p\})$

Assume W is not an Euler circuit ⇒ $G \cap \{k_1, \dots, k_p\} \neq \emptyset$.

Consider:

$S = \{v \in V:$ degree of v in G_p is positive$\}$

$T = \{v \in V:$ degree of v in G_p is $0\}$

$S(W) = S \cap \{\text{vertices in } W\}$

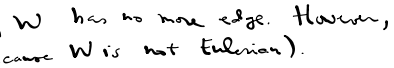
$T(W) = T \cap \{\text{vertices in } W\}$

Claim: $u_p \in T(W)$. Hence $T(W) \neq \emptyset$.
(Because W cannot be extended)

Claim: $S(W) \neq \emptyset$. If $S(W) = \emptyset \Rightarrow$ every vertex in W has no more edge. However, $S \neq \emptyset$ (Because W is not Eulerian).

Since G is connected, we can find one path between a vertex in $T(W)$ to a vertex in $S \Rightarrow W$ can be extended.

So $S(W), T(W) \neq \emptyset$.
$u_p \in T(W)$. So by go backward

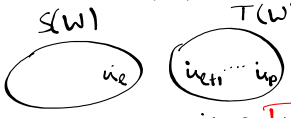


i.e. $u_1 \cdots u_\ell \cdots u_p$

From u_p, we find a number ℓ with the following property:

$u_\ell \in S(W)$

$u_{\ell+1}, \ldots, u_p \in T(W)$.

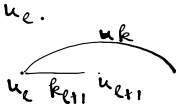


Most difficult part in the argument

Claim: $u_\ell u_{\ell+1}$ is a cut edge in G_ℓ (it is a bridge between S & T in G_ℓ)

Otherwise we could choose one of these $u_{\ell+1}, \ldots, u_p$ to be in S, so a contradiction (or we can add more edge).

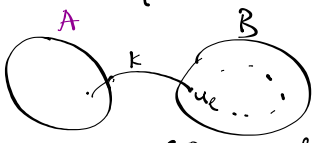
Since the degree of u_ℓ is positive in $G_p \Rightarrow$ also positive in G_ℓ (edges of $G_p \subseteq$ edges of G_ℓ)
$\Rightarrow$ We can find another edge k in G_ℓ incident to u_ℓ.



k must be also a cut edge in G_ℓ
(since otherwise we could choose k instead of $k_{\ell+1}$ by the rules of the algorithm.). k is different from all edges in W. $\Rightarrow k$ is also a cut edge in G_p.



Let B be a component of $G_p \setminus k$ which contains u_p.



Degree of u_ℓ in B is = degree of u_ℓ in $G_p - 1$ = odd number. (Because W is a circuit, & $G_p = E \setminus$ edges in W)

For other points in B, the degree in B = degree in G_p = even number.

But we know that the number of vertices of odd degree in the graph B must be an even number (Cor 6.2) $\Rightarrow$ contradiction. □

Corollary: G (connected) has an Euler path iff G has exactly 2 vertices of odd degrees.

Proof: Illustrated in the first example of today. □

More generally, we can consider the situation where G does not satisfy the conditions for to have an Euler circuit or Euler path.

Chinese post man problem: $G = (V, E)$ a connected multigraph. We want to find a route which visits each of the edges at least 1, so that the cost is minimum. (For each edge we give a positive weight.)
Solution is given in the book, but we won't cover more in this course.

Lemma: The algorithm to find an Euler path/circuit has polynomial complexity.

Proof:
In the algorithm, we need to add, at each step i, one more edge. There are $O(|E|)$ of them.
At each step i: We need to look at remaining edges which connect to i. There are $O(|E|)$ of them.
For each of the above edges, say k, want to find if it is a cut edge of G_i. That is want to find if $G_i \setminus \{k\}$ has more connected components than G_i. So just need to count the number of connected components of a graph.
The complexity is polynomial.

How to find the number of connected components of a graph.?

Algorithm:
Start from $u_0 \in V$.
Find all $v \in V$ which connect to u_0.
(for example by using Breadth first search.)



If we covered all the vertices $\Rightarrow$ STOP.
If not choose one vertex that is not covered, & start again. □
HW: look at the Breadth first search again & have a more precise estimate for time complexity.

Hamiltonian path: A path which visits each vertex exactly once.
Hamiltonian circuit: A circuit which visits each vertex exactly once.
Rmk: Thomas Kirkman worked with this subject 1 year before Hamilton. On chessboard it was studied even earlier in the 9th century – for knight tours – by Indian & Islamic mathematicians.
Rmk: It looks similar to Euler path/circuit problem (incidence) The difference is that an edge connects exactly 2 vertices, while a vertex can be incident to any number of edges.

Thm: The Hamiltonian path/circuit problem is NP-complete.

Proof: Look at references on the Wikipedia page. □

Hamilton path/circuit problem is a special case of the famous Traveling Salesman Problem (TSP): next time.

Move now to §8.5: P vs NP.

P: A problem which can be solved in polynomial complexity.

E.g. Euler path problem.

NP: (non-deterministic polynomial):

A problem, which for every proposed solution, we can check it in polynomial time.

Ex. Hamiltonian path problem.

(Proof: A Hamiltonian path problem will involve $|V|$ of edges, which is polynomial complexity). Rmk: NP does not mean non-poly nomial.

P vs NP problem: Is $P = NP$?

1 million USD problem.

The most important & useful in Mathematics & Computer Science.

People believe that $P \neq NP$.

NP-hard: A problem P is NP-hard if $\forall$ NP problem L, there is a polynomial algorithm (many-one reduction) to reduce L to P.

E.g. TSP is NP-hard.

NP-complete: A problem P is NP-complete if $\forall$ NP problem L, there is a polynomial algorithm to reduce L to P.

E.g.: Hamiltonian path problem is NP complete.

Obvious(?): $P \subseteq NP$

History: John Nash (1955) is the first person to mention P vs NP in a letter to NSA, in which he applies to cryptography.