Chinese Postman Problem

Endre Boros
26:711:653: Discrete Optimization

Spring, 2019

1 Eulerian Tours and Paths

Leonard Euler lived in Königsberg in 1736 and liked to walk around the town. The river Pregel divided the town into four areas, the two sides and two islands, which can represent by the 4 vertices of the graph in figure 1. These areas were connected by seven bridges, which we can represent by the 7 edges of the graph in figure 1.

Euler liked to take long walks so that he could walk across all the bridges. Euler noticed that no matter how he walked he always had to cross some of the bridges twice. He realized that if he does not want to cross twice the bridges, then every time he enters an area, he needs another bridge form that area to leave, that is he will have to walk across an even number of bridges incident with a particular area. Since each area in Königsberg is incident with an odd number of bridges, such a walk is impossible.

In a graph a sequence of edges and vertices, such that consecutive ones are incident, is called a *walk*. It is a *closed walk* if the initial and terminal vertices are the same. Thus, Euler's desire was to find a walk in the graph in figure 1 that uses every edge exactly once. In memory of Euler, we call such a closed walk an Eulerian tour. Euler realized that a necessary condition for an Eulerian tour to exist is that all vertices must have an even degree. In fact he argued (and someone proved it later) that this is also sufficient.

Theorem 1 (Euler, 1736; Hierholtzer, 1871) *If $G = (V, E)$ is a connected graph, then it has an Eulerian tour if and only if $d_G(v)$ is even for all vertices $v \in V$. In fact, if this condition holds, one can find an Eulerian tour in $O(|E|)$ time.*

Proof: The key observation for the proof is that if we walk along the edges of the graph and we arrive to a vertex v different from our initial vertex then up to that point of the walk, we must have used an odd number of edges incident with that vertex. Since the degree of v is even, there must exists an unused edge incident with v, and thus we can continue our walk along that edge, without returning to a used edge.

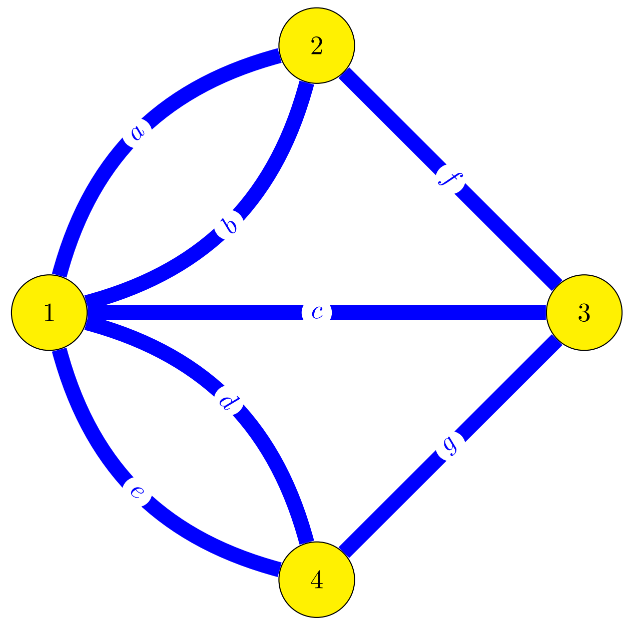

Figure 1: Bridges of Königsberg

Let us start at vertex $v_0 \in V$ and walk along the edges, using the above observation, always choosing an unused edge to leave the current vertex, until we arrive to a vertex v from which we cannot leave on an unused edge. Note that by the above argument we must have $v = v_0$, and we must have used all edges incident with v_0.

If there are still unused edges in the graph, then by the connectivity of G, we must have a vertex $v_1 \neq v_0$ that are incident with some used edges, and also has unused edges incident with it. Assume that edges e_k is on which we arrived in our walk to v_1 and e_{k+1} is the next edge on we left this vertex (note that k may not be unique, since we could have returned to v_1 several times; in this case we choose one of the moments when we pass through v_1). Let us now start a new walk on the unused edges of G from initial vertex v_1 and stop again when we cannot leave a vertex on an unused edge. Our argument above shows that we will get a closed walk that ends in v_1. Assume f_1 is the first edge on we left v_1, and f_p is the last edge on which we returned to v_1. Then we can combine the two walks, by inserting the new new walk between e_k and $e_k + 1$. That is, in our first walk when we arrive along e_k to v_1 then we continue on f_1, and follow the second walk, until we arrive back on f_p, after which we leave on e_{k+1} and follow again the first walk.

Now, if there are still unused edges, we must have a node v_2 different form v_1 and v_0 that are incident with both used and unused edges. Then we can start a new closed walk from v_2 on unused edges, until we have to stop. Then we had to arrive back to v_2. Now we can repeat the previous insertion step, and insert

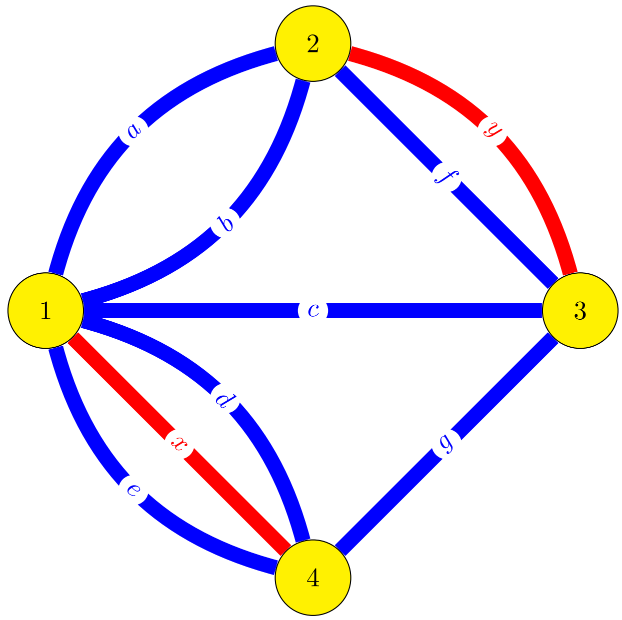

Figure 2: Königsberg with two new bridges

our new walk into our combined walk, etc.

If we store the edges in queues at the incident vertices, then the above procedure can be realized by working with every edge exactly once, that is in $O(|E|)$ time. □

In fact Euler also considered Eulerian paths, that is a walk that starts at a given vertex and ends at another given vertex and uses every edge exactly once.

Corollary 1 *Given a connected graph $G = (V, E)$ and two vertices $u, v \in V$, $u \neq v$. There exists an Eulerian path from u to v if and only if the degrees of vertices u and v are odd, and the degrees of all other vertices of G are even.*

Proof: Let us add a new edge (u, v) to G. Then all degrees become even, and thus by the previous theorem there exists an Eulerian tour. Let us go around the edges of this tour in such an order that we pass from v to u along the new ly added edge. Then, the rest of our walk, from u to back to v forms and Eulerian path. □

As an illustration, assume that in Königsberg we build two more bridges, as in figure 2, and assume that our initial vertex is area 3, i.e., $v_0 = 3$. Then in the first stage we may construct the walk c, d, g, f, y, after which we are back again in area 3 and cannot leave on an unused bridge. Now vertex area 1 that is $v_1 = 1$ is on this walk, and it is incident with some unused bridges. Starting a new walk form v_1 may give the closed walk a, b, x, e, starting and terminating in v_1. Since in the first walk edges c and d are incident with v_1 we can insert the second walk in between these two edges of the first walk, and obtain the closed walk $c, a, b, x, e, d, g, f, y$ which starts and ends in area 3. Since there are no more unused edges this is our Eulerian your with initial vertex area 3.

Try to create an Eulerian tour staring at area 2!

The Chinese Postman Problem

In 1960, the 26 years old Chinese mathematician, Mei-Ko Kwan, considered the following optimization problem. In rural areas at that time postman delivered mail by walking all day. To be able to get to every house, he had to walk along every street, starting from the post office in the morning and arriving back to the post office in the evening. The order in which he traversed the streets was not important. What is the best plan for him to walk the minimum distance daily?

In this problem we can represent the streets as edges of a graph $G = (V, E)$, where vertices represent crossings and important points, such as the post office. Every street segment connects two of these crossings/important points and has a given length. We denote by $\ell(e)$, $e \in E$ the given lengths of the edges.

Since the postman has to deliver letters, packages and newspapers to customers on every street, he has to walk through every street at least once. Thus, $\sum_{e \in E} \ell(e)$ is clearly a lower bound on the total distance he has to walk daily. Since he cannot fly or jump far, he has to follow a closed walk along the edges of G, starting and ending at the post office.

By our previous analysis, if G has an Eulerian tour, then his optimal plan is to follow that tour, and the above lower bound is exactly the optimal distance he has to walk daily.

So the question remains: what is his distance minimizing plan, if G is not Eulerian?

Jack Edmonds and Ellis L. Johnson answered this question in an elegant paper in 1973.

Theorem 2 (Edmonds and Johnson, 1973) *Given a connected graph $G = (V, E)$ on $n = |V|$ vertices, and a distance mapping $\ell : E \to \mathbb{R}_+$, one can find an optimal postman tour in $O(n^3)$ time.*

Proof: According to our analysis above, we can assume that G is not Eulerian, since otherwise an Eulerian tour is the optimal solution, and we know how to construct it in $O(|E|) = O(n^2)$ time.

Thus G has some vertices with odd degrees. Let us denote by $W \subseteq V$ the set of odd degree vertices, and note that $|W|$ must be even.

Note that the optimal plan of the postman is a closed tour of the edges of G that traverses every edge at least once, such that the total length of edges traversed more than once is as small as possible. If we consider every pass of an edge as a separate copy of that edge, then this extended version of G must be Eulerian (since our walk is a closed walk). In other words, as already Mei-Ko

Kwan observed in 1960, our task is to figure out how to add second, third, etc. copies of some of the edges to G so that it becomes Eulerian, and the total length of the added edges is as small as possible.

Edmonds and Johnson observed that this problem is equivalent with finding a perfect matching of the odd vertices between themselves via shortest paths in G such that the total length of the matching shortest paths is at minimum.

More precisely, let us denote by $P(u, v)$ the ℓ-shortest path between vertices $u, v \in W$ and define $\delta(u, v) = \sum_{e \in P(u,v)} \ell(u, v)$ for all $u, v \in W$, $u \neq v$. Consider the auxiliary complete graph $K(W)$ on vertex set W with weight function δ on its edges. Let us find a perfect matching M in $K(W)$ (—W— is even, and $K(W)$ is complete, thus such perfect matchings exists) that has the smallest δ-weight (solve a linear program over the perfect matching polytope).

Then add a new copy of each $(x, y) \in P(u, v)$ to G for all $(u, v) \in M$. The graph G extended this way becomes Eulerian. Any Euler tour in this extended graph is an optimal solution to the postman's problem.

In the above computations the most expensive steps are computing all pairs shortest paths between vertices of W (in $O(n^2 \log n)$ time) and then the computation of the minimum weight perfect matching in the auxiliary graph $K(W)$ (in $O(|W|^3) = O(n^3)$ time). This step is dominating the computations, thus the total complexity is bounded by $O(n^3)$.

Note that we added at most $(n-1)|W|/2$ edges to G, so the total number of edges of the extended graph is still $O(n^2)$, and hence the construction of the final Eulerian tour can be done in $O(n^2)$ time. $\square$

As an illustration consider the input graph in figure 3. The auxiliary graph is shown in figure 4. The minimum weight perfect matching in this auxiliary graph is shown in figure 5. Finally, the extended graph is shown in figure 6. The optimal postman tour is $(1, 2)$, $(2, 3)$, $(3, 4)$, $(4, 1)$, $(1, 5)$, $(5, 2)$, $(2, 3)$, $(3, 5)$, $(5, 4)$, $(4, 5)$, $(5, 1)$, and the total length of the optimal postman tour is 10 units more than the sum of the length of the edges in the input graph.

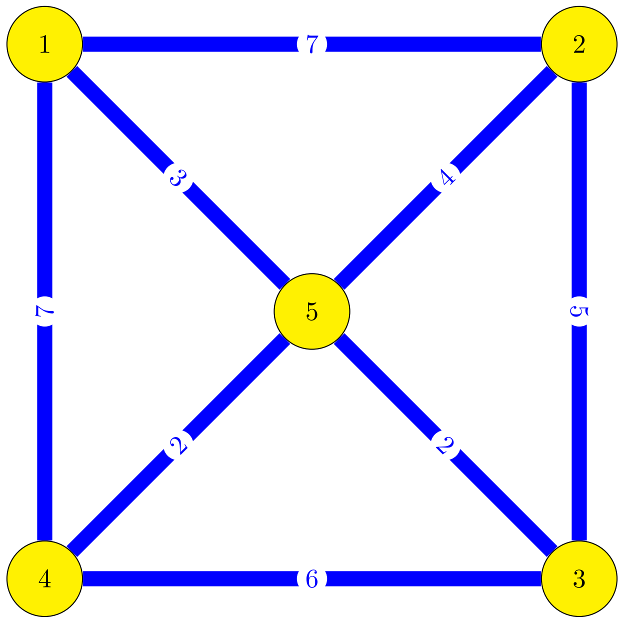

Figure 3: Chinese postman example

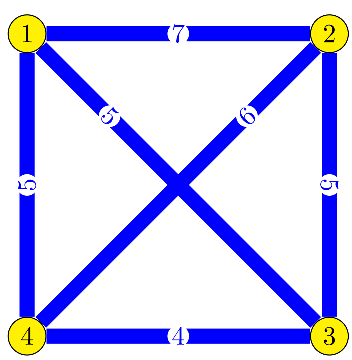

Figure 4: Chinese postman problem: auxiliary graph.

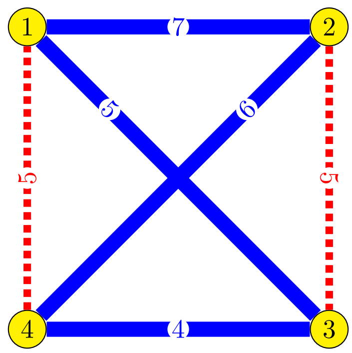

Figure 5: Chinese postman problem: minimum weight perfect matching in the auxiliary graph.

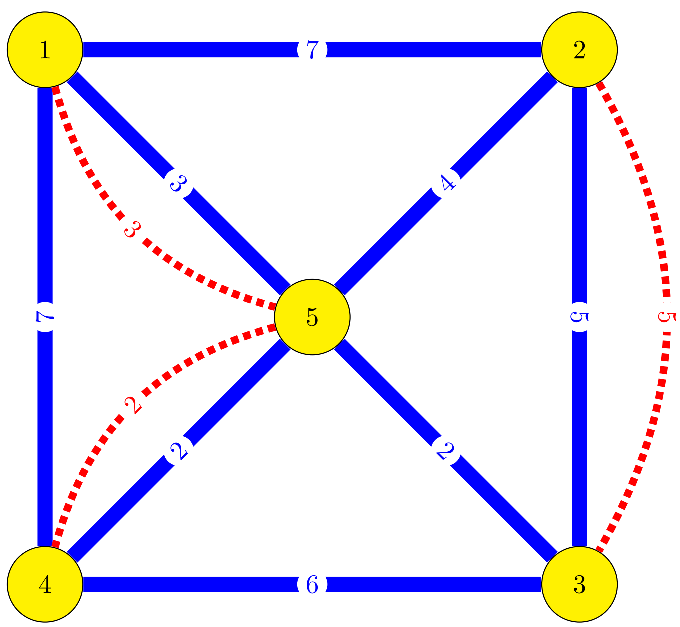

Figure 6: Chinese postman problem: shortest paths realizing minimum weight perfect matching in auxiliary graph.