1 Postman problem

In this problem, a postman starts at a post office and must deliver mail to all houses on all streets and come back to the post office using the minimum amount of time.

We assume all streets are lined with houses and are reachable from the post office. We can formalize the problem as follows.

Chinese postman
Input: A connected graph $G = (V, E)$, weights $w_e \geq 0$ for each $e \in E(G)$.
Output: A minimum weight closed walk containing all edges.

Note that since we must visit every edge at least once, we might as well only count the total weight of edges we encounter more than once. In particular, if G is contains an Eulerian cycle, this cycle is the optimal solution. The reformulated problem is as follows.

Chinese postman
Input: A connected graph $G = (V, E)$, weights $w_e \geq 0$ for each edge $e \in E$.
Output: A minimum weight set of edge of G that we need to "double" to make the graph Eulerian.

1.1 Solution

Find shortest paths between all pairs of odd degree vertices. Then find a minimum weight subset of the paths that contain every odd vertex once (as endpoint). I.e., find a maximum weighted matching in the complete graph where vertices are odd degree vertex and weights are the weight of shortest paths between them.

We can solve a more general problem and will prove the above solution is correct in that setting.

2 T-join

T-join
Input: A graph G and a subset T of vertices
Output: Determine the minimum weight spanning subgraph of G with odd degree for all vertices in T and even degree for all other vertices.

For the postman problem, T was all odd degree vertices of G. If all weights are non-negative, we can use the same solution as for the postman problem:

Find shortest paths between all pairs of vertices in T and find a maximum weighted matching in the complete graph with vertices set T and edges weighted by the shortest paths.

Lemma 2.1. *There is an optimal T-join J that is union of shortest paths and no edge is used more than once.*

Proof. The main tool we will use is the fact that for a cycle C, $J \Delta C$ is still a T-join.

Since all weights are non-negative, we can assume J contains no cycles.

We can assume J contains no multiple edges as they form cycles of length 2.

Start at an odd vertex and walk in J until we are stuck. We end up at an odd vertex. This walk is a simple path P since J contains no cycles. If P is not a shortest path, we can replace P by a shortest path P' and get a better T-join $J\Delta(E(P) \cup E(P'))$, a contradiction. So P is a shortest path.

Say the endpoints of P are a, b. By induction on $|T|$, if $J - P$ is a minimum $T\Delta\{a, b\}$-join then it is a union of shortest paths so J is union of shortest paths. If $J - P$ is not a minimum $T\Delta\{a, b\}$-join then adding P to a minimum $T\Delta\{a, b\}$-join gives a T-join of better value, contradicting the minimality of J. □

[Comb opt p170-171 for examples]

Running time analysis: One all pairs shortest paths and one maximum weighted matching (using $|V(G)|$ as an upper bound in each case). The second term is dominating.

2.1 Negative cost

Now suppose some edges are allowed to have negative weights.

Instead of starting from an empty solution, pretend we take all negative weight edges (but can choose to drop those edges from our solution at the cost of its weight is absolute value).

We need to update the parity requirements for this transformation. $T' = T\Delta T_{<0}$ where $T_{<0}$ is all odd vertices in the subgraph of all negative weight edges. This is a T-join problem! The edges we can drop simply have weight $|w_e|$ (if originally they had weight w_e).

But in this new problem, we only have non-negative weights!

2.2 Application: Negative cycles detection

Testing for negative cycles in a (undirected) graph: Set $T = \emptyset$ and ask for an $\emptyset$-join.