The approximate traveling salesperson tour algorithm

The Input: A complete graph with a symmetric ($c(u,v) = c(v,u)$) weight function satisfying the triangle inequality ($c(u,v) + c(v,w) \geq c(u,w)$).

The Algorithm:

1. We pick any vertex y_1 and define C_1 as the empty tour from y_1 to y_1.
2. We set z_2 as the vertex nearest to y_1 and define C_2 as the round-trip $y_1 - z_2 - y_1$. (This is the only phase when we use the same edge twice.)
3. While k is less than the number of vertices we repeat the following step. Select y_k and z_k to be a pair of vertices such that y_k is on C_k, z_k is not on C_k and $c(y_k, z_k)$ is minimal among all distances $c(y,z)$ such that y is on C_k and z is not on C_k. Let y'_k be the vertex immediately preceding y_k on the tour C_k. The tour C_{k+1} is obtained from C_k by inserting z_k between y'_k and y_k.
4. The output is the Hamilton cycle C_n, where n is the number of vertices.

Theorem 1 *The cost of the approximate traveling salesperson tour is at most twice the minimum cost.*

To prove this theorem, we create a sequence $S_1, S_2, \ldots, S_n$ of subgraphs with the following properties:

(S1) S_1 is a Hamiltonian path obtained from a cheapest Hamiltonian circuit C^* by removing one of the maximum weight edges.

(S2) For each $k \in \{1, \ldots, n-1\}$, S_{k+1} is obtained from S_k by removing one edge e_k.

(S3) For each $k \in \{1, \ldots, n-1\}$, the cost of C_{k+1} exceeds the cost of C_k by at most twice the cost of e_k.

If we are able to construct such a sequence of subgraphs we are done: the cost of C_n is at most the cost of the edges in S_1, which is less than twice the cost of C^*. In order to be able to show that we can find an appropriate S_{k+1} in each step, along the way we show that each S_k has the following properties:

(S4) Each connected component of S_k has exactly one vertex on C_k.

(S5) Each vertex that is not on C_k belongs to some connected component of S_k.

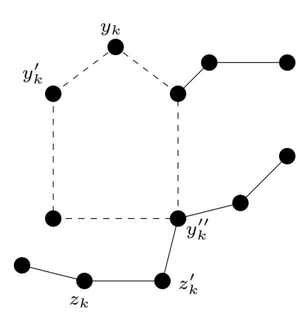

Figure 1: $C_k \cup S_k$

Note that, for $k = 1$, the graph C_1 is a single vertex, and the graph S_1 is a Hamiltonian path containing all vertices, so conditions (S4) and (S5) are satisfied. A typical situation is shown in Figure 1, where the edges of C_k are represented as dashed edges, and the edges of S_k are represented as continuous edges. The vertices y_k, y'_k and z_k are defined in the approximate traveling salesperson tour algorithm. The vertex z_k is not on C_k, but, by property (S5), it belongs to a connected component of S_k. This connected component is a path (obtained from the path S_1 after deleting some edges) which, by (S4), has exactly one vertex on C_k: let us call this vertex y''_k. In the connected component of S_k containing z_k, there is a unique path from z_k to y''_k. Let z'_k be the vertex adjacent to y''_k in this path. We now define S_{k+1} as the graph obtained from S_k by removing the edge $e_k = \{z'_k, y''_k\}$, see Figure 2 It is

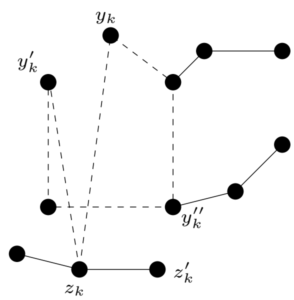

Figure 2: $C_{k+1} \cup S_{k+1}$

easy to see that S_{k+1} also satisfies the properties (S4) and (S5), and it is constructed obeying the rule (S2). We only need to show it satisfies (S3). When we create C_{k+1} from C_k, the weight changes by $c(y'_k, z_k) + c(z_k, y_k) - c(y'_k, y_k)$. By the triangle inequality, we have

$$c(y'_k, z_k) \leq c(z_k, y_k) + c(y'_k, y_k), \quad \text{implying} \quad c(y'_k, z_k) + c(z_k, y_k) - c(y'_k, y_k) \leq 2c(z_k, y_k).$$

Thus the weight of C_{k+1} exceeds the weight of C_k by at most $2c(z_k, y_k)$, which , by the selection rule for y_k and z_k, is at most $2c(z'_k, y''_k)$.