Advanced Algorithms and Data Structures
Hand-in 2

Philip Pickering
pgpick@gmx.at

Thomas Bracht Laumann Jespersen
ntl316@alumni.ku.dk

Prove that 1-tree is a lower bound

Consider the optimal tour C^* in a graph G and a vertex u. Included in the tour are two edges connecting u to the rest of the graph, e_1 and e_2. Let G' be the graph formed from G by removing u and all its incident edges from G. Then we can find a spanning tree $T^{G'}$ from the optimal solution, by removing e_1 and e_2 from C^*. The following holds:

$$c(C^*) = c(e_1) + c(e_2) + c(T^{G'}). \tag{1}$$

Next consider a minimum spanning tree M found in G', and the two lowest-cost edges e_1', e_2' connecting u to G'. It is trivial to see that:

$$c(e_1') + c(e_2') \leq c(e_1) + c(e_2), \tag{2}$$

and we can also state that:

$$c(M) \leq c(T^{G'}), \tag{3}$$

since M and $T^{G'}$ are both spanning trees in G'.

Combining equations (2) and (3) gives:

$$c(e_1') + c(e_2') + c(M) \leq c(e_1) + c(e_2) + c(T^{G'}) \overset{(1)}{=} c(C^*), \tag{4}$$

which proves that the 1-tree is indeed a lower bound on the optimal tour.

Zone LP

Given a graph $G = (V, E)$ with cost/distance function $c : V \times V \to \mathbb{R}$, we can write the following linear program to maximize the radii of non-overlapping circles drawn around each vertex (the vertices being the origin):

$$\begin{array}{lrcll} \text{max.:} & \sum_{i=1}^{|V|} r_i & & & \\ \text{s.t.:} & r_i + r_j & \leq & c(v_i, v_j), & \forall i, j \in \{1, \dots, |V|\}, v_i, v_j \in V, i \neq j \\ & r_i & \geq & 0, & \forall i \in \{1, \dots, |V|\} \end{array}$$

Zone lower bound

As is described in the exercise a tour visiting each city has to travel from somewhere on the circumference of each circle, to its center and back.

Let $r_1^*, \ldots, r_2^*$ be the optimal radii found by solving the liner program presented in the previous section. Then a lower bound on the optimal TSP solution is given by:

$$2 \sum_{i=0}^{|V|} r_i^*.$$

Because the circles are non-overlapping, the optimal solution C^* may have to travel outside the circles, this is indeed a lower bound.

ILP formulation of TSP

To formulate TSP as an *integer linear program* (ILP), we introduce decision variables for each edge x_{ij}, where $i, j \in \{1, \ldots, |V|\}$ and $i < j$, and demand that $x_{ij} \in \{0, 1\}$, indicating by zero that the edge is not included in a tour, and one if it is.

$$\text{min.:} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij} x_{ij}$$

$$\begin{array}{lrcll} \text{s.t.:} & \displaystyle\sum_{k=1}^{i-1} x_{ki} + \sum_{k=i+1}^{n} x_{ik} & = & 2, & i \in \{1, \ldots, |V|\} \\ & \displaystyle\sum_{i,j \in Z} x_{ij} & < & |Z|, & \varnothing \subset Z \subset V \\ & x_{ij} & \in & \{0, 1\}, & i, j \in \{1, \ldots, |V|\} \end{array}$$

The first constraint specifies that for every vertex i, exactly two edges incident to i must be included in the tour.

The second constraint is concerned with subtour elimination, demanding that for any non-empty $Z \subset V$, the number of edges that are entirely in Z (both endpoints in Z) must be strictly less than $|Z|$. Say we have five vertices in a subset Z. In order to form a tour amongst themselves, we would need use six edges, which is disallowed by this constraint.

The third constraint is our *integrality constraint*, demanding that we either include an edge completely or not at all.

Removing both integrality constraint and subtour constraint. => assignment problem...

1-tree description of algorithm

Zones

ILP Relaxation

	knollA		knollB		knollC	
	Training	Test	Training	Test	Training	Test
1-tree	0.01	0.03	0.40	0.49	0.01	0.03
Zones	-0.242763	-0.142026	-1.629922	-2.061411	-0.002435	-0.001425

Table 1: Error percentages and margins for each of the KNOLL problems

1-tree How it performed and reasoning

Zones

ILP Relaxation

- table of results 'inf' when no results
- relaxation of ILP (Thomas)
- Description of implementation
-

1. Prove that the 1-tree is a lower bound. ✓
2. Formulate a linear program that maximizes the radii of such non-overlapping circles. ✓
3. Describe how a lower bound can be computed from these circles. ✓
4. Write an integer linear program for TSP. ✓
5. Describe how you would relax the integer linear program of TSP to get a lower bound.
6. Implement each of the three lower bounds described above (1-tree, zones and relaxation) and use them on each of the three problems. For each problem, report the shortest tour and number of nodes visited for each lower bound method. If the program does not terminate within 10 minutes its fine to just write 'inf'. (There will be a prize for the group that solves the largest problem with the fewest number of nodes evaluated).
7. For each of the three lower bound methods, give a brief description why you believe it performed the way it did and how it could be improved.