Vertex Penalty Algorithm for the Traveling Salesman Problem

This Hypercard stack was prepared by:
Dennis L. Bricker,
Dept. of Industrial Engineering,
University of Iowa,
Iowa City, Iowa 52242
E-mail: dennis-bricker@uiowa.edu

Symmetric TSP

\[
\text{Minimize } \sum_{(i,j) \in A} C_{ij}X_{ij} + \sum_{i \in N} u_i \left( \sum_{(i,j) \in A} X_{ij} - 2 \right)
\]

subject to

\[
X \in T_f = \text{set of all spanning 1-trees of network}
\]

\[
\Phi(u) = \text{Minimum } \sum_{X \in T_f} \left( C_{ij} + u_i + u_j \right) X_{ij} - 2 \sum_{i \in N} u_i
\]

Lagrangian Relaxation

For fixed $u$, this is a "minimum spanning 1-tree" problem!

Finding the Minimum Spanning 1-Tree

Select an arbitrary city, e.g., city #1

Further restrict the set of 1-trees to include only those for which node 1 has degree 2 and lies on a cycle!

Finding the Minimum Spanning 1-Tree

Select an arbitrary city, e.g., city #1

Find the minimum spanning tree of the set of cities excluding the selected city.

Finding the Minimum Spanning 1-Tree

Select an arbitrary city, e.g., city #1

Find the minimum spanning tree of the set of cities excluding the selected city.

Finding the Minimum Spanning 1-Tree

Find the two nearest neighbors of the selected city, and add the two corresponding edges.

Lagrangian Dual

For each choice of vector $u$, the value of the Lagrangian relaxation $\Phi(u)$ provides us with a lower bound on the optimum TSP tour.

The Lagrangian Dual problem is to...

Maximize $\Phi(u)$
\[ \Phi(u) = -2 \sum_{i \in N} u_i + \text{Minimum} \sum_{X \in \mathcal{T}_I} \sum_{(i,j) \in X} (c_{ij} + u_i + u_j) x_{ij} \]

For the purpose of finding the minimum spanning 1-tree, the length of edge \((i,j)\) is
\[ c_{ij} + u_i + u_j \]
i.e., the edge length is increased by the 'penalties' of the 2 end vertices.

Suppose that we were to enumerate the (finitely many) 1-trees of the network:
\[ \hat{x}_k^b \in \mathcal{T}_I, k=1,2,...,K \]
Then
\[ \Phi(u) = \text{Minimum} \sum_{l \in k} \sum_{(i,j) \in A} (c_{ij} + u_i + u_j) \hat{x}_{ij}^k - 2 \sum_{i \in N} u_i \]
i.e., \( \Phi \) is the lower envelope of a finite set of linear functions of \( u \) (& is therefore concave piecewise linear).

If \( \hat{x}_k \) is optimal in the evaluation of \( \Phi(u) \), then
\[ \Phi(u) = \sum_{(i,j) \in A} (c_{ij} + u_i + u_j) \hat{x}_{ij}^k - 2 \sum_{i \in N} u_i \]
and the vector \( \gamma \)
with \[ \gamma_i = \sum_{(i,j) \in A} \hat{x}_{ij}^k - 2 \]
is a subgradient of \( \Phi \) at \( u \).
To adjust \( u \), then, step in the direction \( \gamma \).

**Finding Minimum Spanning 1-tree**
- select an arbitrary node \( k \)
- find minum spanning tree of the network with node \( k \) deleted
- add to the minimum spanning tree the 2 shortest edges incident to node

**Example**
Random Symmetric TSP
(seed = 135393)

**Iteration 1**
Lower Bound: 260
Sum of excess degrees: 3

Subgradient direction:
\[ \gamma_i = +1 \text{ for } i=2,4,9 \]
\[ \gamma_i = -1 \text{ for } i=10,11,12 \]
\[ \gamma_i = 0 \text{ otherwise } \]
New minimum spanning 1-tree:

Note that the degrees of nodes 2, 4, and 9 were decreased because of the penalties...

Iteration 2
Lower Bound: 259.4666667
Sum of excess degrees: 4

Subgradient direction:
γ_i = +2 for i = 5
γ_i = -1 for i = 3,10
γ_i = -1 for i = 4,9,11,12
γ_i = 0 otherwise

Iteration 3
Lower Bound: 261.0711111
Sum of excess degrees: 3

Subgradient direction:
γ_i = +1 for i = 2,10,12
γ_i = -1 for i = 4,9,11
γ_i = 0 otherwise

Iteration 4
Lower Bound: 274.5712593
Sum of excess degrees: 3

Subgradient direction:
γ_i = +1 for i = 3,6,9
γ_i = -1 for i = 10,11,12
γ_i = 0 otherwise

Failed to converge.
Greatest Lower Bound on tour length is 220.7662064

Final 1-tree
Final Penalties

The optimal tour length is 321, so the lower bound is quite "tight!"