IE 400 – Spring 2018

Project Assignment
Due Date: May 3, 2018 by 12:00 noon

Let $G = (N, A)$ where $A = \{(i, j) : i, j \in N, i \neq j\}$ be a given instance for TSP. Let c_{ij} be the travel time between node $i \in N$ and node $j \in N$. Consider the following compact model for solving TSP. Let,

$$x_{ij} = \begin{cases} 1 & \text{if salesperson goes from node } i \in N \text{ to node } j \in N \text{ in the tour} \\ 0 & \text{otherwise} \end{cases},$$

$u_i =$ visitation order of node $i \in N$

$$\begin{aligned}
\min & \sum_{(i,j)\in A} c_{ij}x_{ij} & \\
\text{s.t.} & & \\
& \sum_{j:(i,j)\in A} x_{ij} = 1 & i \in N \\
& \sum_{j:(j,i)\in A} x_{ji} = 1 & i \in N \\
& u_j \geq u_i + 1 - n(1 - x_{ij}) & (i,j) \in A \; j \neq 1 \\
& u_1 = 0 & \\
& x_{ij} \in \{0, 1\} & (i,j) \in A \\
& u_i \geq 0 & i \in N
\end{aligned}$$

There are different versions of TSP problems such as *Symmetric* TSP where travel times are symmetric and *Euclidean* TSP where travel times correspond to Euclidean distances. The above integer programming model will solve any instance of TSP exactly. There are also heuristic approaches such as *Nearest Neighbor* and *Minimum Spanning Tree* algorithms which do not guarantee the optimal solution but a sub-optimal heuristic solution in polynomial time. You could learn more about TSP from **here**. There are also some useful sources in the References. Your tasks for this project are the following:

1. Generate some random TSP instances corresponding to *Symmetric*, *Asymmetric*, *Euclidean* and *non-Euclidean* (where triangle inequality may not be satisfied) problems. One recommended way to generate the Euclidean distances is as follows: at first step generate random coordinates for each node and then calculate the distance between each pair of nodes in Euclidean manner. This guarantees that the triangle inequality is satisfied.

2. Solve each instance using an integer programming solver and using the above model. If you choose, you could research different and better integer programming formulations of TSP and use such formulations. You can choose any solver and any modelling language of your choice (Xpress, Cplex, GAMS, etc.). We recommend XPRESS which is available all through BCC labs. Attached please find some introductory remarks on Xpress solver. If you decided to use GAMS modeling language which works with CPLEX we can share GAMS/CPLEX license with you upon your request.

3. Write a program in your choice of programming language for any TSP heuristic (you can come up with a hybrid heuristic of your own) and solve each of the generated instances in Step 1 using this algorithm.

4. Prepare a written document to report your findings and explain the developed heuristic method. Generate as many instances as necessary to make reasonable arguments and answer the following types of questions. How far are your heuristic solutions from the optimal solution? Does your heuristic algorithm favor for example symmetric instances over non-symmetric ones, Euclidean ones over non-Euclidean ones? Which type of TSP instances are best solved with the heuristic? What dimensions (number of cities/nodes) are you capable of solving using the exact methodology?

5. Submit your report as well as your Xpress (or your choice of solver) and your code in electronic form to *arda.bala@ug.bilkent.edu.tr*

6. You will have an oral exam where you will be asked random question about using Xpress (or your choice of solver) and your heuristic code.

References

http://www.math.uwaterloo.ca/tsp/index.html, (There are many interesting materials about TS).

https://heuristicswiki.wikispaces.com/Travelling+salesman+problem

https://ocw.mit.edu/courses/sloan-school-of-management/15-053-optimization-methods-in-management-scienc
lecture-notes/MIT15_053S13_lec17.pdf