CS 456 Project V:
Two Approximation Algorithms

Description: In this final project you are to implement and analyze two approximation algorithms for NP-complete problems, Euclidean Traveling Salesman Problem and Subset Sum Problem.

E-TSP
For the Euclidean TSP problem, you are to implement the 2-approximation algorithm that uses an Euler tour of the “doubled” Minimum Spanning Tree (MST) of the points. Your goal is to calculate and analyze the actual approximation factors you obtain in your approximation algorithm compared to the optimal TSP tour, which may certainly be less than 2. This means that you must also compute the optimal TSP tour on the same point sets via a brute force method. Therefore, restrict yourself to point sets of 10 – 15 nodes.

I suggest that you implement Prim's algorithm to compute the MST because you already implemented the structurally similar Dijkstra algorithm in your first project. You must implement Prim's algorithm yourself, although you may use a library for an efficient priority queue, and your implementation must run in O(E log V) time. You should also figure out how to extract an Euler tour of your computed MST in O(E + V) time after you have computed your MST. And, finally, it should be straightforward to you to construct your approximate TSP tour from that.

Subset-Sum
For the subset-sum problem, you have already seen the non-polynomial O(nW) time dynamic programming algorithm to solve it exactly. That is what you will use to obtain the optimal solution to an instance. To the same instances, you will also use the greedy ½ – approximation algorithm we saw in class based on sorting the items in order of descending weight. You should use an efficient sorting algorithm (e.g. MergeSort) so that your greedy approximation algorithm takes O(n log n) time. You may generate instances any way that you wish, retaining the understanding that your goal is to analyze what the actual approximation factors are.

Report: In your report, you must discuss the following issues supported by appropriate figures, tables, and plots extracted from your data:

• Compare the time complexities of your brute force and dynamic programming exact algorithms to the greedy approximate algorithms. This would be a good time to discuss any improvements you might have made for your brute force E-TSP algorithm if you made any, as well as how you obtained your Euler tour in linear time.

• For what types of point sets did you obtain the best (i.e. closest to 1) approximation ratios for your approximate E-TSP? And, for what point sets did you obtain the worst (i.e. closest to 2) approximation ratios? Discuss.

• For what weight-distributions did you obtain the best (i.e. closest to 1) approximation ratios for your approximate Subset-Sum? And, for what weight-distributions did you obtain the worst (i.e. closest to ½ ) approximation ratios? Discuss.

You may further discuss any issues that you have learned in the process, or anything else that you feel is important. Remember to put your plots/figures/tables INTO your report.

What to turn in: This part is the same as “What to turn in” for Project 1.

This assignment is due by MIDNIGHT of Sunday, April 28. No late submissions are accepted.