Approximation Algorithms - Tools and Techniques	**Spring 2002**

Lecture 7 : Feb 1, 2002

Lecturer: Kamesh Munagala [1] *Scribe: Siddartha Y R*[2]

Euclidean TSP - Karp's Partitioning Scheme

The Metric TSP

Given n points and distances $d(i,j),\ i,j \in \{1,\ldots,n\}$ satisfying the triangular inequality $d(i,j) \leq d(i,k) + d(k,j),\ \forall i,j,k$; To find the minimum cost tour which visits all the n points (i.e., a permutation of the n points).

The Euclidean TSP

The metric space is Euclidean, i.e., the points $\{p_1, p_2, \ldots, p_n\}$ are in $\Re^m$ and specified as m-tuples $(x_1, x_2, \ldots, x_m)$; the distance between points $x = (x_1, x_2, \ldots, x_m)$ and $y = (y_1, y_2, \ldots, y_m)$ is given by

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + \ldots + (x_i - y_i)^2 + \ldots + (x_m - y_m)^2}.$$

Karp's Partitioning Scheme

Provides an approximation which is $O(\sqrt{n})$ in the worst case but almost optimal (in the average) if the points are chosen independantly and uniformly at random from $[0,1]^2$. The restriction to $[0,1]^2$ does not imply any loss of generality since any problem can be scaled to satisfy this condition. The scaling is made such that the points "just fit" into $[0,1]^2$ - as shown in Figure. 1 - points are of the form that either A and B or C and D are present in the scaled problem. This form of scaling immediately gives a lower bound on the tour of 2.

[1] Dept of Comp. Sci., Stanford Univ.
[2] Dept of Comp. Sci. and Automation, Indian Institute of Science, Bangalore.
`siddu@csa.iisc.ernet.in`

A Naive Algorithm

Divide the problem space $P = [0,1]^2$ into a number of strips, and do a "raster scan" across the space, visiting each point from the nearest point along the raster. This is illustrated in Figure. 2 If P is divided into l strips, each of width $1/l$, then the cost of the tour is (in the worst case)

$$\begin{array}{rcl} = & \text{cost of scan lines} & +n\times \text{ deviation from scan line to point} \\ = & \text{vertical } + \text{ horizontal traversals} & +n\times 2\times 1/2l \\ = & 2+(l+1) & +\dfrac{n}{l} \\ = & l+3+\frac{n}{l}. & \end{array}$$

Choosing l to minimise this cost gives $l = \sqrt{n}$. This gives the worst case cost of the tour as

$$C_{NAIVE} = 2\sqrt{n} + O(1) \quad (1)$$
$$= O(\sqrt{n}). \quad (2)$$

Given a TSP instance with a constant number of points, the brute force method of trying all possible tours and picking the least cost one is itself poly-time. The running time for s points is then $O(s!)$. One approach to solving a general TSP is to partition the n points into small sets of bounded size, obtain tours for each of these subsets in poly-time, and combine these tours to obtain the complete tour. Choosing a linear bound for the running time for solving the subproblems, we obtain the required bound on the size of the subsets as

$$\begin{array}{rl} & s! \le n \\ \Rightarrow & s \le \frac{\log n}{\log\log n}. \end{array}$$

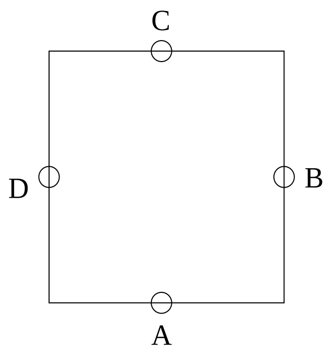

Figure 1: "Just-Fit" Scaling of a problem instance to $[0,1]^2$

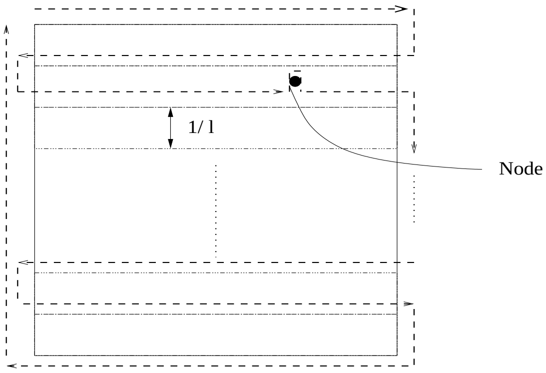

Figure 2: A Naive Algorithm

Partitioning the points into n/s subsets[3] of size s gives the running time as

$$\frac{n}{s} \times O(n).$$

For the euclidean space where $m = 2$ (which is the case we will consider in the rest of this lecture), this partitioning can be done easily - divide P into $\sqrt{\frac{n}{s}}$ vertical (say) strips, each containing a set of $\sqrt{ns}$ points. Again, divide each of these into $\sqrt{\frac{n}{s}}$ horizontal strips such that each partition P_i contains s points. Find the optimal tour inside each of these partitions by brute-force in $O(n)$ time. These optimal tours are then joined by randomly picking a vertex in each P_i, running NaiveTSP on this set, and interleaving the optimal subtours with the "scan" returned by NaiveTSP; giving a complete tour as shown in Figure. 3. It is found that randomly picking the vertex from each partition gives a acceptable aproximation with little computational effort.

Algorithm 1 KarpTSP

Input : n points in $[0,1]^2$ Euclidean space
Output : A tour over the n points whose cost exceeds the optimal by atmost $(1 + O\left(\sqrt{\frac{n}{s}}\right)$

1: $s \leftarrow \frac{\log n}{\log \log n}$
2: divide $[0,1]^2$ into $\sqrt{\frac{n}{s}}$ vertical strips, each containing $\sqrt{ns}$ points.
3: divide each strip into $\sqrt{\frac{n}{s}}$ horizontal strips, each resulting partition containing s points.
4: set $Q \leftarrow \phi$

[3]in the following, note that rounding down of fractional values to their floor has nott been explicitly shown to preserve a cleaner exposition.

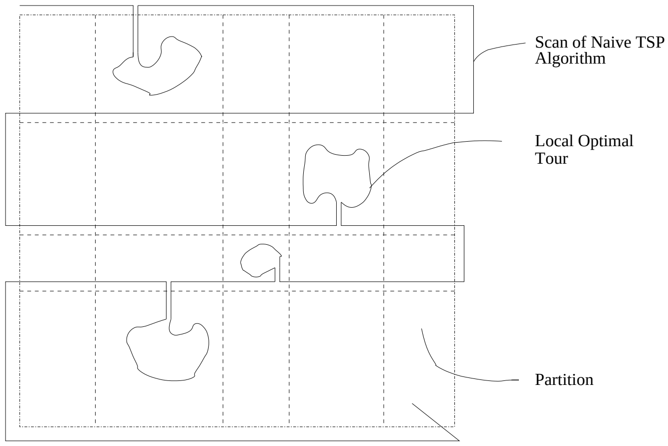

Figure 3: Karp's Partitioning Scheme

for each $i \in \{1, \ldots, \frac{n}{s}\}$ **do**
  form set P_i each containing the points in the $i\acute{t}h$ partition,
  find T_i, the optimal tour in P_i by brute force,
  $Q = Q \cup \{x \mid x \text{ randomly selected from } P_i\}$
end for
tour $T' \leftarrow$ NaiveTSP(Q)
for each $x \in Q \mid x \in P_i$ **do**
  tour $T \leftarrow$ interleave T_i to T' at x
end for
return T
done

The tour returned by KarpTSP has a cost C_{KARP} given by

$$C_{KARP} = \frac{n}{s} \times C_s^* + C_{NAIVE,n/s}$$

where C_s^* is the cost of an optimal TSP over s points of a partition, and $C_{NAIVE,n/s}$ is the cost of tour over $\frac{n}{s}$ points, returned by NaiveTSP.

We know that NaiveTSP on a set of $\frac{n}{s}$ points yields a tour of cost

$$2\sqrt{\frac{n}{s}} + O(1).$$

Now we establish a relation between the cost of the optimal tour (of the s points) in a partition P_i, call it z_i^*; and the cost of that section the optimal tour (over n points) which passes through

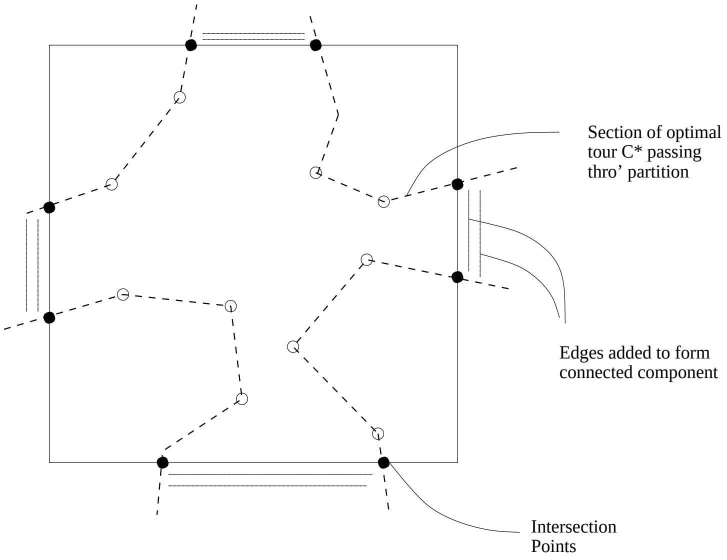

Figure 4: Relating z_i and z_i^* in a partition

P_i, call it z_i. We have the optimal tour over n points

$$C^* = \sum_{P_i} z_i$$

and the tour returned by our algorithm

$$C_{KARP} = \sum_{P_i} z_i^* + 2\sqrt{\frac{n}{s}} + O(1)\,. \tag{3}$$

We show next that

Claim 1 *$z_i^* \le z_i + 2p(P_i)$, where $p(P_i)$ is the perimeter of the partition P_i.*

Proof: Let S_i be the set

$$S_i = \{s \text{ points of } P_i\} \cup \{\text{points on the intersection of perimeter of } P_i \text{ and the optimal tour}\}$$

By showing that an optimal tour over S_i satisfies the required bound, we will have shown that z_i^* also satisfies the same. The graph formed by the optimal tour over S_i may be disconnected. By connecting adjacent boundary points, a connected graph results. The points in P_i are of even degree in this graph, since the optimal tour enters and leaves this partition. However, the introduced boundary vertices could be of odd degree. By repeating the required newly introduces edges, all the points can be made to be of even degree. Then, the total length of all these edges put together is

$$\begin{aligned} z_i^* \leq & \quad z_i + 2p(P_i) \\ \leq & \quad z_i + 2\left(4\sqrt{\tfrac{n}{s}}\right). \end{aligned}$$

gives us the required result. ■

Now, we have from Equation. 3,

$$\begin{aligned} C_{KARP} &\leq \sum_{P_i} z_i^* + 2\sqrt{\frac{n}{s}} + O(1) \\ &\leq C^* + 2\sum_{P_i} p(P_i) + 2\sqrt{\frac{n}{s}} + O(1) \\ &\leq C^* + 2\left(4\sqrt{\frac{n}{s}}\right) + 2\sqrt{\frac{n}{s}} + O(1) \\ \Rightarrow \quad C_{KARP} &\leq C^* + O\left(\sqrt{\frac{n}{s}}\right). \end{aligned}$$

where in the third step we sum over the perimeters of all the partitions and where $s = \frac{\log n}{\log \log n}$. In the next lecture we show that if the points are chosen independently and uniformly at random then the expected value of C^* itself is large - $O(\sqrt{n})$ and hence C_{KARP} is a sufficiently close approximation to C^*.