CS6100 Course Project
Report

Chriostofides' Short-cutting heuristics for Euclidean metric Travelling salesman problem

Submitted in partial fulfillment of
the requirements for the award of the degree of

Bachelor of Technology
in
Computer Science and Engineering

Submitted by

Roll No	Names of Students
CS17B005	A. Subhash
CS17B011	D. Varun Teja
CS17B012	D.M.S Krishna
CS17B021	P. Jaitesh

Under the guidance of
Prof. B. V. Raghavendra Rao

Department of Computer Science and Engineering
IIT Madras

Abstract

Several $O(n), O(n^2)$ short-cutting heuristics are described which are used in Christofides' algorithm for solving n-city travelling salesman problems whose cost matrix satisfies the triangularity condition. The Christofides' algorithm invovles computation of a shortest spanning tree of the graph G defining the TSP, and finding the minimum cost perfect matching of a certaing induced subgraph of G.

Contents

List of Figures

Chapter 1

Problem Definition

The goal is to find more efficient(in terms of solution cost) short-cutting heuristics which are used in the last step of Christofides' algorithm for finding the TSP tour.

Chapter 2

Introduction

2.1 Metric space

A metric d on a set X, also called a distance function, is a function that defines a distance between each pair of elements of the set. A set with a metric is called a **metric space**.

Formally, $d : X \times X \to R$ is a metric if it is a function satisfying the following properties $\forall x, y, z \epsilon X$:

1. Non-negativity : $d(x, y) \geq 0$
2. Indiscernability : $d(x, y) = 0$ iff $x = y$
3. Symmetry : $d(x, y) = d(y, x)$
4. Subadditivity : $d(x, y) + d(y, z) \geq d(x, z)$

2.2 The Christofides' algorithm

The Christofides' algorithm is an approximation algorithm for Metric TSP with an approximation ratio of 1.5. Let G be a graph with n points in the euclidean metric space. Below is a description of the steps involved in the Christofides' algorithm.

2.2.1 Pseudo code

Algorithm

1. Find an MST of G, say T.
2. Compute a minimum cost perfect matching, M, on the set of odd-degree vertices of T.
3. Add M to T and obtain an Eulerian multi-graph H.
4. Find an Euler tour, E of this graph.
5. Output the tour that visits vertices of G in order of their first appearance in E.

Figure 2.1: MST T of G

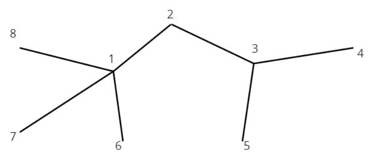


Figure 2.2: Multi-graph H of G

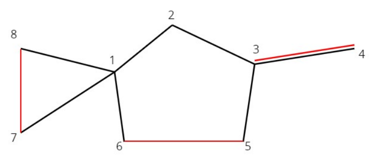
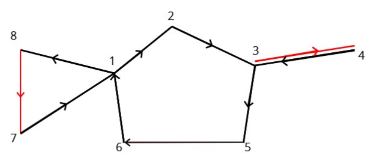


Figure 2.3: Euler tour E of H



Euler Tour: 1 8 7 1 2 3 4 3 5 6 1

2.2.2 Time complexity

1. Creating MST T of G : $O(nlogn)$
2. Finding the minimum cost perfect matching M : $O(n^3)$
3. Creating multi-graph H : $O(n)$
4. Finding Euler tour E in H : $O(n)$

Chapter 3

Short-cutting heuristics

3.1 Simple Heuristic

1. Remove repeated points in the Euler tour to get a Hamiltonian cycle.
2. Time complexity is $O(n)$

3.2 Tri-Opt Heuristic

1. In this hurestic we take a Euler tour and greedily remove the repeated vertices
2. The hurestic value used is sum of distances from the vertex to adjacent vertices minus distance between the adjacent vertices
3. We can see that this is better than previous one but we are performing it on a Euler tour, hence there is still room for imprevement
4. Time complexity is $O(n)$

3.3 Tri-Comp Heuristic

1. This hurestic is applied on Multi graph (H) instead of one Euler tour
2. Here we start with vertices of order greater than two and greedily remove its edges until its order is two
3. The idea is that in the final hamiltonian cycle which we need to arrive by short cutting has degree two for all the vertices

4. Two things we need to do are - pair up the free vertices formed greedily and make sure the process does not result in two disjoint components
5. The hurestic value here is sum of distances between paired up vertices and distace of two edges that remained with our vertex
6. Since our problem is in 2d space, each vertex in MST will have a degree of maximum 5, so our multi graph will have a maximum degree of 6. This property highly affect the theoritical complexity of our hurestic
7. Time complexity for checking the graph connectivity is $O(n)$ and for complete hurestic is $O(n^2)$

3.4 DIH-Tri-Comp Heuristic

1. This hurestic is DIH(Degree Increasing Heuristic) optimization applied on the previous one
2. We want to increase the order of a vertex in MST by adding the vertices of its children to itself
3. We can see that the tree is no longer MST but this process preserves the Euler tours i.e. the set of Euler tours of this tree is super set of that of the MST
4. This way we are applying hurestic on bigger space than that of the former one and have a chance of getting better results
5. DIH can be implemented with $O(n)$, so comp hurestic is the bottleneck here. Overall complexity is $O(n^2)$

Chapter 4

Future Work

¡Future work here¿

Chapter 5

Conclusion

¡Conclusion here¿

Acknowledgments

¡Acknowledgements here¿

¡Name here¿

¡Month and Year here¿
National Institute of Technology Calicut

References

[1] ¡Name of the reference here¿, `<urlhere>`

[2] ¡Name of the reference here¿, `<urlhere>`