8th. World Congress on Computational Mechanics (WCCM8)
5th. European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2008)
June 30 – July 5, 2008
Venice, Italy

A NOVEL IMPLEMENTATION OF ARORA´S ALGORITHM FOR THE EUCLIDEAN TSP

Barbara M. Rodeker[1], María V. Cifuentes[2] and Liliana Favre[3]

[1]Universidad Nacional del Centro de la Provincia de Buenos Aires
brodeker@alumnos.exa.unicen.edu.ar

[2]Universidad Nacional del Centro de la Provincia de Buenos Aires
CIC
cifuente@exa.unicen.edu.ar

[3]Universidad Nacional del Centro de la Provincia de Buenos Aires
CIC
lfavre@exa.unicen.edu.ar

Key Words: *Geometric optimization Problem, Approximation Algorithms, Heuristic Algorithms, Travelling Salesman Problem, C++.*

ABSTRACT

The travelling salesperson problem (TSP) is a famous combinatorial optimization problem in the set NP-Hard. The significance of the TSP is that includes many problems that are natural applications in computer sciences and engineering. However, none of the known TSP solutions are of polynomial-time complexity.

Given n nodes in R^2 (more generally in R^d), TSP is to find the minimum length path that visits each node exactly once. If distance is computed using distance between nodes then the problem is called Euclidean TSP. This problem can be used as a model for the study of general methods that can be applied to a wide range of geometric optimization problems.

An innovative Polynomial-Time Approximation Scheme (PTAS) for the Euclidean TSP was given by Arora [1]. Given n nodes in R^2, and for every fixed $c > 1$, a randomized version of the scheme finds a $(1+1/c)$-approximation to the optimum traveling salesperson tour in $O\left(n(\log n)^{O(c)}\right)$ time [2]. The main idea is to recursively divide the problem into weakly dependent subproblems that are solved *bottom-up*, using dynamic programming to find a tour that crosses each line of the partition at most $O(c)$ times. So therefore, the solution involves four steps: *perturbation*, *construction of dissection*, *dynamic programming* and *path´s trimming*.

Arora´s proposal concerns only with the existence of a PTAS and hence not with practical implementations. In light of this, we propose an implementation of the Euclidean TSP that is based on the essential steps of Arora´s algorithm and adds some heuristics to improve the running time. Next, we describe the essential steps of the Arora´s solution in the context of our approach.

In the perturbation step, a proposed regular grid classifies geometrically nodes in the plane. The cell size c is the minimum distance between nodes. In that case, the node coordinates and cell size are modified by a factor equal to $2c$ to ensure that the minimum distance is two. Finally, to avoid overlaps in the next phase, a new shift puts in coordinates in odd positions. Following, the dissection step divides the *enclosing box:* the smallest square containing all nodes whose size is 2^k. The successive subdivisions generate a quadtree hierarchy until every quadtree region has only one node inside.

At the next step, the adjacent quadtree regions are communicated via portals. The portals for a dissection are a set of points on the square edges. On each edge a square has at least one portal to communicate with every adjacent terminal square (neighbor). This portals configuration improves the execution time. Finally the path is computed using dynamic programming and the reconstruction is made in the path´s trimming step, starting from the information registered by the dynamic programming table. The following figures summarize the differences between Arora´s proposal (Fig. 1) and our implementation (Fig. 2) regarding the steps described before.

Figure 1. Arora´s proposal

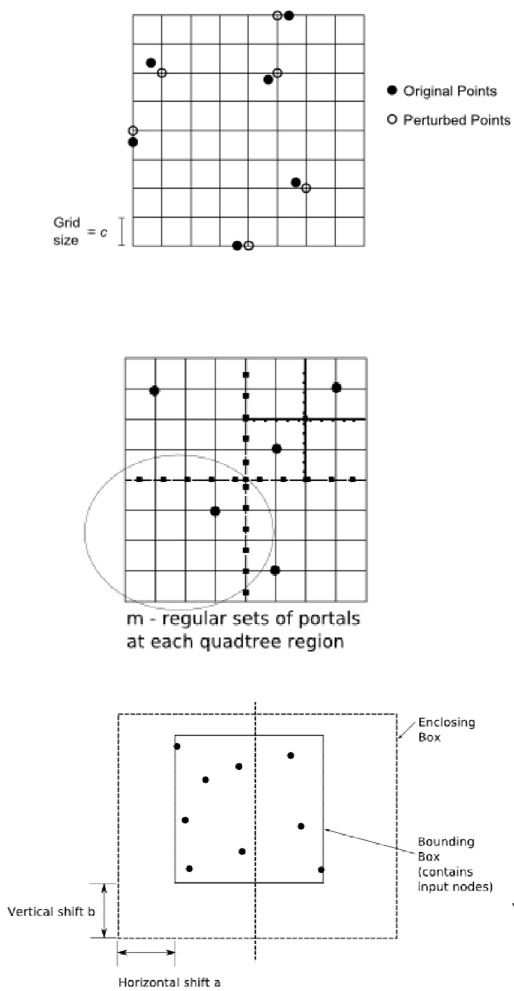
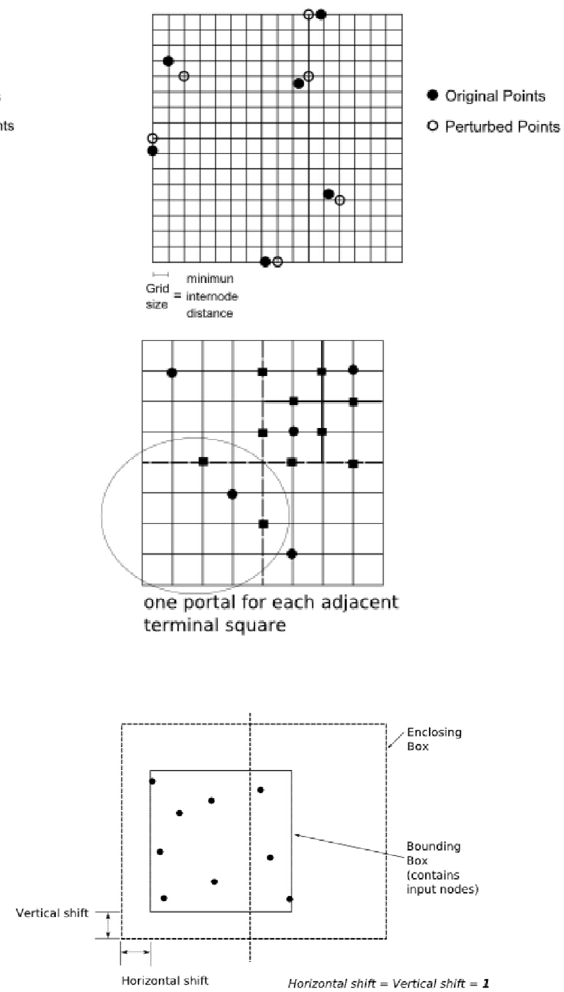


Figure 2. A modified Euclidean TSP



A TSP implementation in C++ is presented and discussed along with some results on several moderate sized problems. We have compared the performance of our implementation with other algorithms included in Concorde Solver Software [3]. For comparative analysis we have used TSP instances from TSPLIB [4].

REFERENCES

[1] S. Arora. Approximation schemes for NP-hard geometric optimization problems: A survey. Math Programming, 97 (1,2), 2003

[2] S. Arora. *Polynomial Time Approximation Schemes for Euclidean Travelling Salesman and Other Geometric Problems*. Princeton University, 1996.

[3] Concorde Software page. http://www.tsp.gatech.edu/concorde/index.html

[4] TSPLIB, Traveling salesman problem library, http://www.iwr.uni-heidelberg.de/iwr/comopt/software/TSPLIB95/