Parallel Approaches for Solving Large-scale Travelling Salesman Problem

Roman Bazylevych, Bohdan Kuz, Roman Kutelmakh

Lviv Polytechnic National University, 12 Bandera str., Lviv, Ukraine

rbaz@polynet.lviv.ua, rkutelmakh@ua.fm, bohdan.kuz@gmail.com

Abstract. *The Traveling Salesman Problem (TSP) belongs to class of NP-hard problems. Computations for large-scale problems require a lot of time. The article proposes parallel approach for solving such kind of problems. Solving is done in four stages: decomposition, finding partial solution, merging of partial solutions and optimization of complete solution. One of existing TSP algorithms is used as base heuristic for finding solutions in clusters as well as for joining them. The developed "ring" approach is used in the merging stage. Computations at the main stages can be done in parallel. Optimization stage is applied to improve quality of the complete initial solution. Experiments on multi-core and multi-processor systems are executed.*

Keywords

TSP, parallel algorithms, decomposition, MPI.

1 Introduction

The first mention of TSP can be found in 1832 [4]. Symmetric problem has computation complexity $O((n-1)!/2)$. Existing heuristic methods for TSP have complexity of $O(n^2)$ or higher and hence they are not efficient to the large-scale problems. Applegate et al. have calculated the optimal solution for the 85900-points TSP and it is the largest test-case that is solved optimally until now. This solution required nearly 136 years of CPU time [1].

2 Related works

Some TSP algorithms that can be applied to the large-scale problems and provide high quality solutions by using clusters for decompositions were developed in [6], [7]. The most recent paper [8] proposes a "ring" algorithm. Described approach aims to merge partial solutions into the complete one and can be executed in parallel. The goal of proposed logic is to decrease computation time using parallel algorithms on multi-processor systems.

3 Problem formulation

Given a set $P = \{p_1, p_2, \ldots, p_N\}$ of N points that are described by their coordinates $p_i = (x_i, y_i)$, the distance function $dist: P \times P \rightarrow R$ (where R is the set of real numbers), is defined by the Euclidean distance or by any other metric as follows:

$$r_{ij} = dist(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$
$$\forall i, j \in \{1, 2, \ldots, N\} \text{ and } r_{ij} \in R.$$

The problem is to minimize the length of the closed route, which visits all points of the set P only once and returns back to the starting point. That is:

$$M = \langle p_1, p_2, \ldots, p_N, p_1 \rangle, p_i \in P:$$
$$\text{length of } M = \sum_{i=1}^{N-1} distp(p_i, p_{i+1}) + dist(p_N, p_1) \rightarrow min.$$

This problem is symmetric if $dist(p_i, p_j) = dist(p_j, p_i)$, otherwise it is asymmetric. The main goal of this problem is to develop a methodology and an algorithm that can be applied for parallelization and finding the good quality solution.

The symmetric Euclidean TSP is used in further research described in that paper. Problem can be described as Hamilton cycle problem – finding a cycle path in an undirected graph that visits each vertex exactly once.
This work aims to create a parallel approach for solving a large-scale TSP. Area of problems that can be solved with proposed approach contains TSP with the number of points > 1000. Small problems can be solved by existing heuristics (Concorde [2], LKH [5] etc.) in a reasonable time and with high quality.

Currently the best approached for solving TSP are Concorde and LKH heuristics. A sequenced-like algorithms used in those heuristics don't allow to use such approached for a large-scale problems. There are two major approaches to TSP solving – genetic and heuristic, MPI is used for distributed calculations.

4 Main Stages

Finding the solution for a large-scale TSP consists of the following stages:

1. Decomposition of the set of given points P by clustering.
2. Finding the partial solutions for all clusters.
3. Merging the partial solutions to get a complete initial solution.
4. Optimizing the complete initial solution.

Clustering divides the input set partitioned into subsets with limited number of points. Several decomposition algorithms are proposed. The most important input parameter for all of them is maximum number of points allowed in one cluster. Decomposition algorithm isn't time critical and doesn't require multi-processor executions. In our experiments decomposition for model with 10^6 points requires 2.5 seconds.

The huge profit in computation time decreasing was achieved in second stage – finding the partial solutions. Solution for each cluster can be found independently. Maximum number of processors required in that stage is equal to number of generated clusters. In this stage possible to use any TSP algorithm, in our experiments we used Lin-Kernighan-Helsgaun (LKH) algorithm [5]. The result of that stage is partial solutions for each cluster. Each such tour comes through all points in its cluster.

4.1 Decomposition

As the first stage of the decomposition process, a Delaunay triangulation [3] of the input set P of points is performed. The result of the triangulation is a set of triangular faces. By having this triangulation, no point in P will be inside the circumcircle of any triangle. The total length of the Delaunay triangulation edges is minimal. The clustering process consists of the following steps:

1. Delaunay triangulation of the set of points P and form a set of the Delaunay triangles.
2. Build a list of Delaunay triangles sorted by their areas.
3. Take for the list p.2 a triangle with minimum area, remove it from this list and call it a fragment of cluster; else when this list is empty stop the clustering process.
4. Identify all the Delaunay triangles that are adjacent to the already formed fragment.
5. Take a triangle form the adjacent ones with minimum area and merge it with existing fragment. It is a step one of wave propagation.
6. Stop creating the cluster when number of it points is equal to the given quantity or all adjacent triangles belong to already generated clusters and go to p.3 to build next cluster. Else, go to p.4.

Algorithm: Data set decomposition
Input data: Set of points *S*, $N = \|S\|$, maximum and minimum number of points in cluster N_{max} and N_{min}
Output: Set of clusters *P*

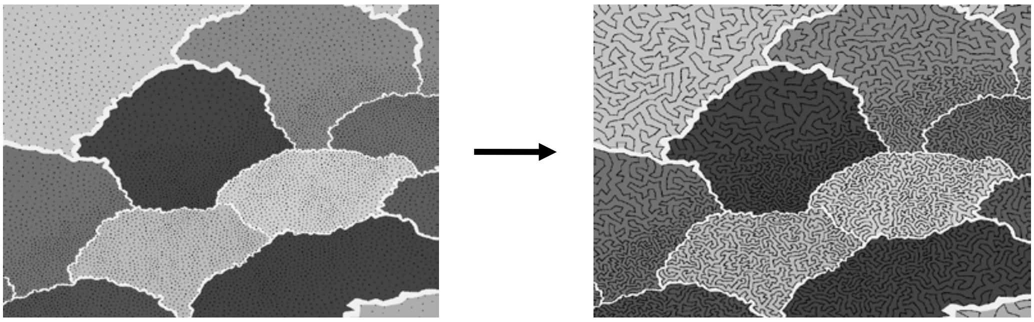
1. Build Delaunay triangulation
2. Sort created triangles by triangle area
3. While there is triangle which doesn't belong to cluster
    a. Take smallest triangle from the triangle sequence
    b. Create new cluster with limited number of points (not more than N_{max}) around selected triangle
4. Search for all points which doesn't belong to any cluster and add them to neighbor cluster
5. Search for all clusters that have number of points less than N_{min} and add them to neighbor clusters

Fig. 1. Decomposition algorithm

The border of each cluster consists of the Delaunay triangulation edges. In addition, the optimal solution for the TSP contains Delaunay triangulation edges. The main limitation parameter for clustering is the maximal number of points n_0 in each cluster. This stage creates the set of clusters.

4.2 Finding and Merging the Partial Solutions

For the second stage, separate TSP solutions are found for each cluster that is formed in the previous stage and these separate solutions will be merged together in stage 3 to form the complete initial solution for the whole problem. Fig. 1 contains the partial TSP solutions for some pieces of the Vangogh test-case [9]. In this stage, each partial solution can be found in parallel, as those calculations are independent.



a b

Fig. 2. Partial solutions

The “ring” method [8] is used to merge the partial solutions. For this method, the merging process consists of the following steps:

- Building an initial “ring”: Select any cluster at random. The border of the “ring” is the border points that belong to the Delaunay triangular edges of the selected cluster (let us call this border as e). Every border edge has two incident border triangles, one is internal and another one is external to the cluster. The incident border triangles of the edges of the “ring” will be considered for initial wave propagation.
- Building the final “ring”: The final “ring” consists of all the points that are obtained by wave propagation from the internal and external triangles (that share the border edges of the “ring”) to a given number of triangles, both internal and external zones of the cluster. If the number of steps of wave propagation is large enough, the final “ring” covers the entire cluster in which the ring is initially started. The number of steps is a parameter of the algorithm and depends on the number of points that must belong to the “ring”.
- Temporarily replacing some all “ring” external route segments by temporary single edges (b and c at Fig. 3): If there is a route segment, whose first and last (i.e., start and end) points are on the “ring” border (a at Fig. 3) and all other points of the route segment are external to the “ring” (i.e., not on the “ring” borders and not internal points of“ring”), then temporarily replace that route segment (let us call this route segment “temporary route segment”) with a single edge that connects the first and last points of the route segment.
- Solving the problem for “ring”: Apply base TSP algorithm on the “ring” points (including the border points d at the Fig. 3, if any also) in such a way that the solution include temporary edges, if any, formed in Step 4.
- Replacing the temporary single edges: replace each temporary single edge by the route segment that it replaced in Step 4.
- Repeating the process to form another “ring”: Repeating all previous steps by randomly selecting another cluster that is not already considered earlier. Stop the process if no more clusters are available.

As a result, we receive complete initial solution M_0 that passes through all points of the set P.

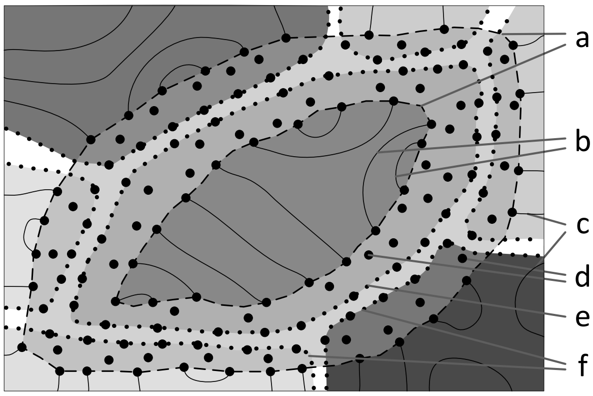


Fig. 3. Clusters merging zone – “ring” (a - “ring” borders, dashed line; b - internal route segments ; c - external route segments; d – points; e - considered cluster border; f - adjacent cluster borders).

Algorithm: Merging partial solutions

Input data: Set of partial solutions, “ring” internal depth G_{in} and “ring” external depth G_{out}

Output: Complete initial solution

1. For each cluster
   a. Create internal area using wave propagation on Gin triangles
   b. Create external area using wave propagation on Gout triangles
   c. Find all temporary single edges
   d. Find solution for created ring using base algorithm and considering temporary single edges
   e. Replace temporary single edges with parts of external tour

Fig. 4. Merging partial solutions

5 Experimental results

The parallel approach was investigated in terms of the solution quality and run time. The TSP test-cases were taken from the [9] and [10]. Experiments were conducted using Supercomputer SCIT4 of Glushkov Institute of Cybernetics of NAS of Ukraine and Cluster of Institute for Condensed Matter Physics of the NAS of Ukraine (results are scaled to processors Intel Xeon E5640 with CPU rate 2.67 GHz).

The following parameters were used:

- number of cluster points - 900.
- internal depth of a “ring” - 9 triangles (i.e., the “ring” covered 9 Delaunay triangles when propagating the wave inside to the given cluster).
- external depth of a “ring” - 24 triangles (i.e., the “ring” covered 24 Delaunay triangles when propagating the wave outside the given cluster).

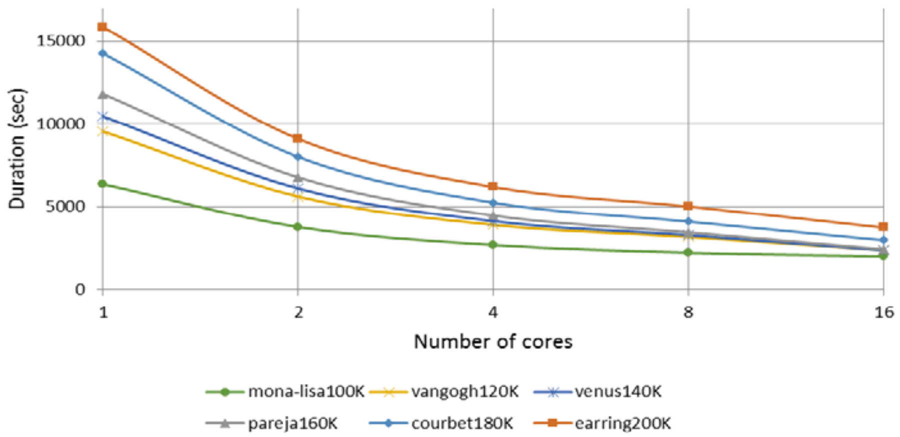
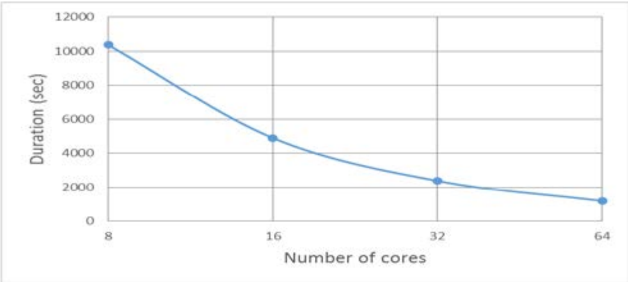
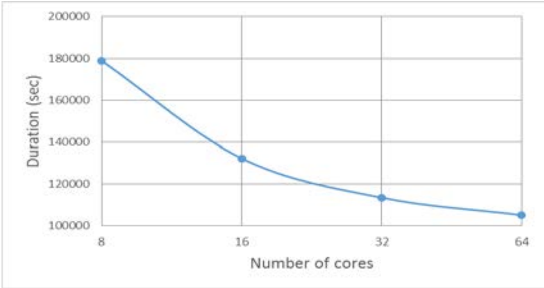


Fig. 5. Total duration vs. number of cores for TSP Art library.



a)



b)

Fig. 6. Partial solution (a) and merging (b) durations vs. number of cores for test E10M.0 from DIMACS library.

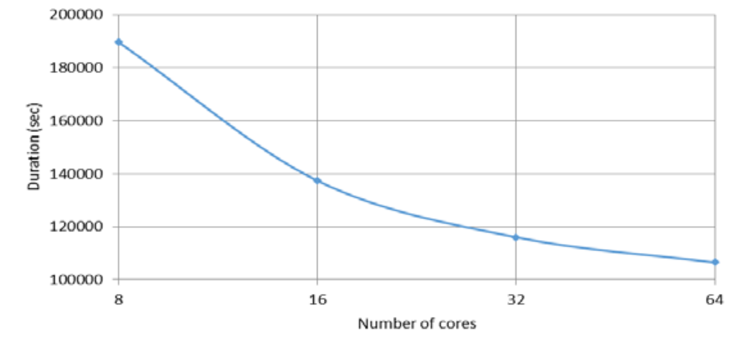


Fig. 7. Total complete solution duration vs. number of cores for test E10M.0 from DIMACS library.

6 Conclusion

Developed approach decreases running time for TSP. Speedup is achieved by decomposition in parallel approach. The well-known heuristic LKH method was used as the basic algorithm in each cluster and for merging stage. Computational complexity is close to linear that make it possible to use for large and very large-scale problems.

References

[1] Applegate D., Bixby R., Chvátal V., Cook W., Espinoza D., Goycoolea M., and Helsgaun K. Certification of an Optimal TSP Tour Through 85,900 Cities // *Operations Research Letters*, 37, 2009.
[2] Concorde website: http://www.tsp.gatech.edu/concorde/index.html
[3] B. Delaunay: Sur la sphère vide, *Izvestia Akademii Nauk SSSR*, *Otdelenie Matematicheskikh i Estestvennykh Nauk*, 7:793–800, 1934.
[4] Der Handlungsreisende – wie er sein soll und was er zu thun [sic] hat, um Aufträge zu erhalten und eines glücklichen Erfolgs in seinen Geschäften gewiß zu sein – von einem alten Commis-Voyageur, Ilmenau, 1832.
[5] K. Helsgaun: An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic, *European Journal of Operational Research 126 (1)*, pp. 106-130, 2000.
[6] R. Bazylevych, R. Kutelmakh, R. Dupas, L. Bazylevych: Decomposition Algorithms for Large-Scale Clustered TSP, *Proceedings of the 3rd Indian International Conference on Artificial Intelligence*, Pune, India, pp. 255-267, 2007.
[7] R. Bazylevych, B. Prasad, R. Kutelmakh, R. Dupas, L. Bazylevych: A Decomposition Algorithm for Uniform Traveling Salesman Problem, *Proceedings of the 4th Indian International Conference on Artificial Intelligence*, Tumkur, India, pp. 47 59, 2009.
[8] R. Bazylevych, R. Kutelmakh, B. Kuz: Algorithm for Solving Large Scale TSP Problem by "Ring" Method", *"Visnyk" of the Lviv Polytechnic National University "Computer Sciences and Information Technologies"*, No. 686, Lviv, pp. 179 – 182, 2010 (in Ukrainian).
[9] TSP Art test data: http://www.tsp.gatech.edu/data/art/index.html
[10] DIMACS test data: http://www2.research.att.com/~dsj/chtsp