ARTIFICIAL INTELLIGENCE IN MODELING AND SIMULATION

DECOMPOSITION METHODS FOR LARGE-SCALE TSP

Roman Bazylevych, Marek Pałasiński, Roman Kutelmakh, Bohdan Kuz, Lubov Bazylevych

***Abstract**: Decomposition methods for solving large-scale Traveling Salesman Problem (TSP) are presented. Three approaches are proposed: macromodeling for clustered TSP as well as extending and "ring" methods for arbitrary points' distribution. Four stages of the problem solving include partitioning of the input set of points into small subsets, finding the partial high quality solutions in the subsets, merging the partial solutions into the complete initial solution and optimizing the final solution. Experimental investigations as well as the comparative analysis of the results and their effectiveness estimation in terms of quality and running time were conducted. The suggested approaches provide substantial reduction in the running time in comparison with the existing heuristic algorithms. The quality loss is small. The problem instances up to 200,000 points were investigated. The TSP is extensively applied in transportation systems analysis, printed circuit boards, VLSI, SoC and NoC computer-aided design, testing and manufacturing, laser cutting of plastics and metals, protein structure research, continuous line drawings, X-ray crystallography as well as in number of other fields.*

***Keywords**: traveling salesman problem, combinatorial NP-hard problems, decomposition, large-scale.*

***ACM Classification Keywords**: G.2.1 Combinatorics - Combinatorial algorithms; I.2.8 Problem Solving, Control Methods, and Search - Heuristic methods.*

Introduction

The Traveling Salesman Problem (TSP) belongs to the class of intractable combinatorial ones (NP-hard). It consists in finding the shortest route through the set of points provided that each point is visited one time. The complexity of the TSP is *O(n!)*. The largest problem for which the optimal solution was found consists of 85900 points [Applegate, 2009]. The computations, though, required about 136 years of CPU time. The proposed approaches are developed for the large-scale TSPs to receive the solutions close to optimal in a reasonable time. The problem under discussion is tightly connected with such areas as logistics, scheduling, robot management systems, analysis and synthesis of chemical structures, continuous line drawings, integrated circuit design, manufacturing, etc.

The existing effective heuristic approaches are characterized by the time complexity not less than $O(n^2)$. The most advanced software application to date that allows finding the optimal route is Concorde [Concorde]. The route close to optimal may be obtained through the Lin-Kernighan-Helsgaun (LKH) algorithm [Helsgaun, 2000, 2006]. Efficient way to solve the large-scale TSP is to use the data decomposition approaches [Reinelt, 1994, Rohe, 1997], [Yll Haxhimusa, 2009]. Some decomposition methods are proposed in [Bazylevych, 2007, 2008, 2009, 2011].

Decomposition approaches

The developed decomposition algorithms for the large-scale TSP solving consist of four main stages:

1. Input data set decomposition into subsets.
2. Finding the partial solutions for subsets.
3. Merging the partial solutions into the one complete initial solution.
4. Applying the optimization algorithms to the initial solution.

Data set decomposition allows splitting a huge problem into the set of the smaller ones. Small problems may be easily solved using the known approaches that guarantee getting high quality results. Each subset has a limited number of points. This number depends on the compromise between quality and runtime. The TSP for every cluster is solved separately by the chosen basic algorithm which provides a high quality solution. The exact algorithms, such as branch-and-bound, could be used at this stage.

For the clustered TSP, macromodeling approach is proposed in [Bazylevych, 2007]. Every cluster is approximated by the macromodel as one point. The macroroute for a set of such points as well as the routes (partial solutions) for all clusters are found by the chosen basic algorithm. All partial solutions are concatenated at the next stage with the creation of the complete initial solution.

For the arbitrary TSP, we split a huge problem into some set of the smaller ones, which have a limited number of points (subsets), and then find the partial solutions for them [Bazylevych, 2008, 2009, 2011]. These processes could be executed in parallel. As a result of this stage, the number of partial tours is obtained, each tour per subset.

At the third stage we merge the partial tours into one complete tour – the initial solution. Two approaches we proposed. As for the first one, merging process consists in extending the partial TSP solutions [Bazylevych, 2008, 2009]. As for the second approach, merging is performed over the regions called the “rings” that are formed from the border points of given and adjacent subsets [Bazylevych, 2011]. The result of this stage is the tour which passes through all the points. The complete initial solution is also received by independently merging the tour pieces for every “ring”.

The quality of the complete initial solution is improved by applying some specially developed optimization methods [Bazylevych, 2007, 2008, 2009, 2011].

Methods for initial solution

For receiving the initial solution we developed such methods:

1. Macromodeling for clustered TSP.
2. The extension method.
3. The “Ring” method.

Macromodeling for clustered TSP

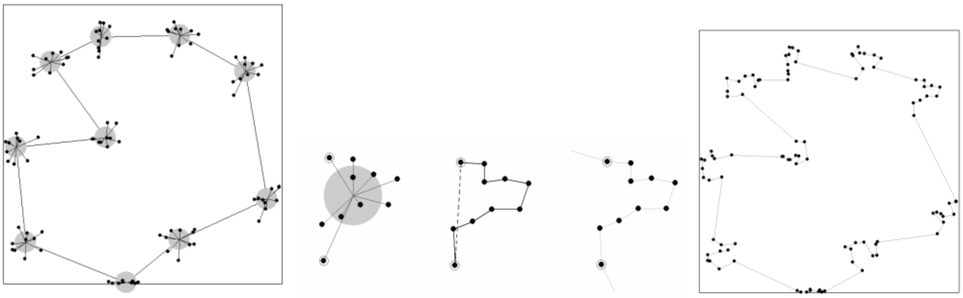
The main steps of the method [Bazylevych, 2007] developed for the clustered TSP are (Figure 1):

a) Macromodeling by finding the minimal length macroroute which passes through every cluster only once. Every cluster is approximated by one point (Figure 1a).

b) Micromodeling by finding the partial solutions in every cluster. This process consists of two steps:
   - finding the shortest route between the adjacent clusters and setting the border points for every cluster;
   - finding the shortest route between the border points for every cluster (Figure 1b).

c) Finding the complete initial solution by concatenating the routes between the clusters with partial routes for all clusters (Figure 1c).

d) Optimizing the solution.



a) Macroroute b) Internal subroute in the cluster c) Complete initial solution

Fig. 1. Macromodeling for Clustered TSP

The extension method

The main steps of the method [Bazylevych, 2008, 2009] developed for the arbitrary TSP are (Figure 2):

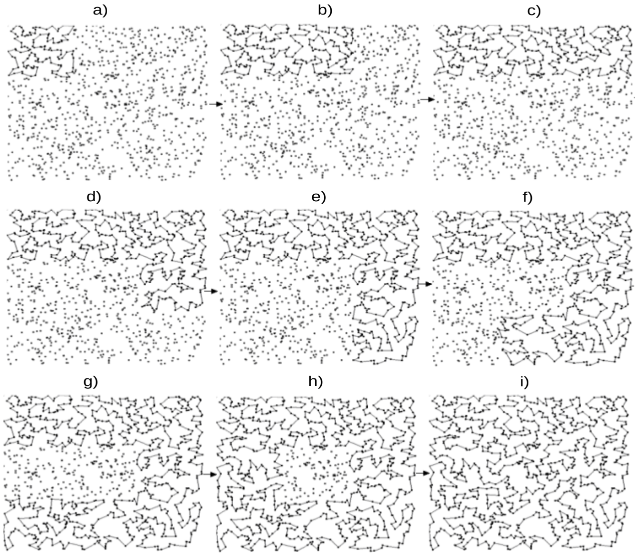


Fig. 2. Example of extending the partial solutions in spiral inner order

a) Choosing a small geometrical area from the whole problem surface in which it would be possible to obtain the high quality TSP solution. It could be the area from the corner (arbitrary) or from the center of the surface.

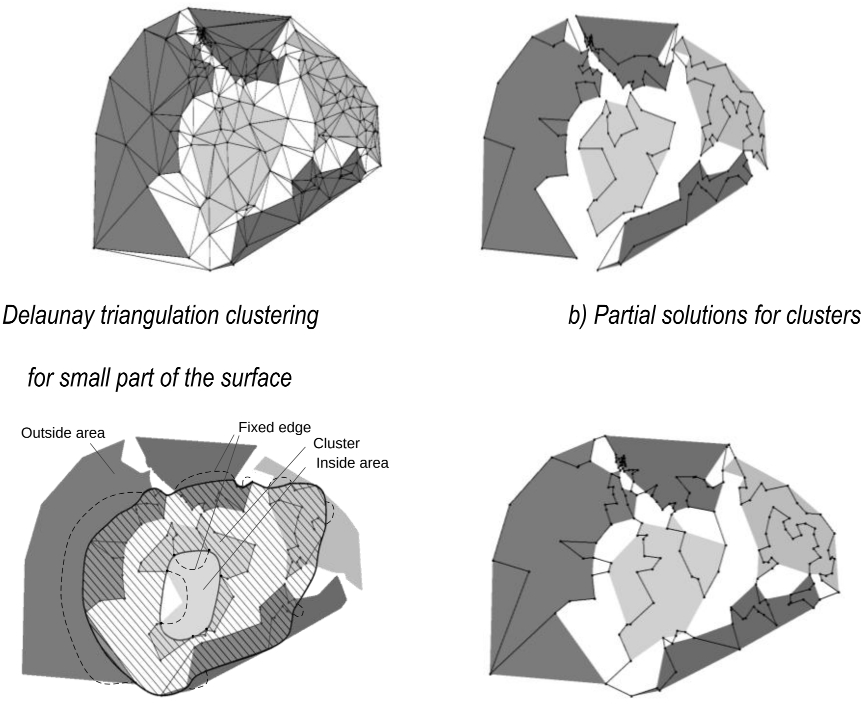
b) Choosing some neighboring area of nearly the same number of points of the previously chosen area (with given percentage of overlapping points, for example, 20% – 50 %).

c) Solving the TSP for a newly chosen area by replacing the rest pieces of already existing route with short (fixed) connections.

d) Finding the complete initial solution by sequential merging of one subset with the one adjacent to it. The extension (sequential merging) can be done in different ways – from the left to the right, from the top to the bottom, inner or outer spiral, etc.

e) Optimizing the solution.

The "Ring" method



Delaunay triangulation clustering for small part of the surface

b) Partial solutions for clusters

c) "Ring" for one cluster

d) Complete initial solution

Fig. 3. "Ring" method for TSP

The main steps of the "Ring" method [Bazylevych, 2011] developed for wide parallelization are (Figure 3, only small part of the surface with a few clusters is considered):

a) Delaunay triangulating the set of points while considering the distances between them.

b) Clustering the set of all points using the wave propagation by triangles. Creating the clusters with given number of points.

c) Finding the partial solutions for all clusters.

d) Finding the complete initial solution by merging the partial solutions of all adjacent clusters together. For every cluster is formed a "ring" from it's and border points of all adjacent clusters. The pieces of routes with non-considered points are replaced with the "fixed" routes.

e) Optimizing the solution.

The steps b), c) and d) can be executed in parallel.

Methods for solution optimization

Further optimization of the initial solution can be achieved by reducing the route length in the Local Optimization Areas (LOAs) [Bazylevych, 2007, 2008, 2009, 2011]. Every LOA consists of a small number of points located in a close proximity. For every LOA the solution can be obtained in a short amount of time. We developed a few optimization methods:

1. Scanning along the route (Figure 4).
2. Geometrical scanning of the whole surface (Figure 5).
3. Scanning around the cluster perimeters (Figure 6).
4. For selected "critical" areas (Figure 7).

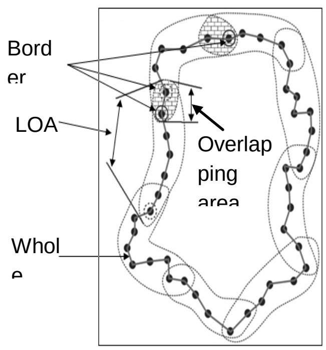


Fig. 4. Scanning along the route

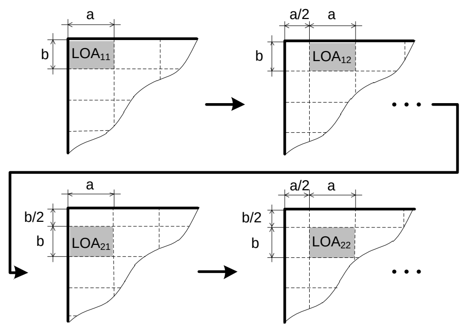


Fig. 5. Geometrical scanning of the whole surface

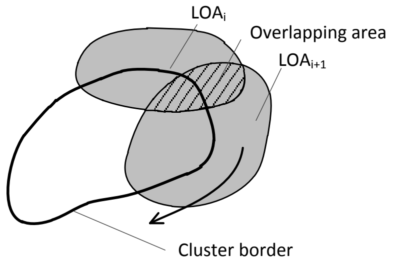


Fig. 6. Scanning around the clusters perimeters

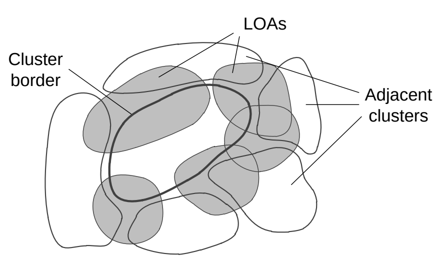


Fig. 7. Selected "critical" areas

The possible detailed changes to the route position due to optimization in LOAs are presented in the Figures 8 and 9.

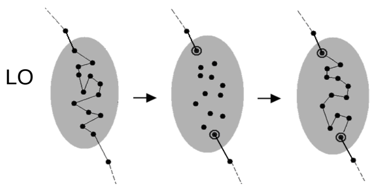


Fig. 8. Optimization in the LOA by scanning along the route

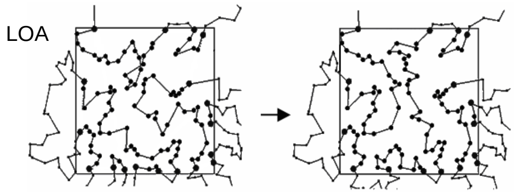


Fig. 9. Optimization in the LOA by geometrical scanning

A very important feature of the proposed optimization methods is that the adjacent LOAs have common overlapping areas. The quality of solution grows with increasing their sizes. The Scanning along the route method has two options. For the first one we consider only the rout points in the given LOA, as well as for improving the solution in the second variant - more qualitative, we also consider adjacent points with forming an enlarged zone [Bazylevych, 2009]. The quality of results depends from the LOAs and overlapping areas sizes. We studied the results depending on the sizes of overlapping areas within 20% - 80% of the LOAs sizes.

Experimental results

The proposed approach was investigated to study the solution quality and runtime. The TSP instances were taken from [TSP Art Instances]. Experiments were conducted on a PC with Athlon II X2 240 processor, 2.8 GHz CPU, and 2 GB RAM. The following parameters were used in our experiments:

- the number of points in a cluster is 800-900;

- the internal depth of a "ring" is 10 triangles (the ring covered 10 Delaunay triangles while the wave propagating inside the given cluster);
- the external depth of a "ring" is 15 triangles (the ring covered 15 Delaunay triangles while the wave propagating outside the given cluster);
- the number of points in the local optimization area is 800;
- the number of points in the overlapping area is 400.

The results are provided in Table 1. All four optimization strategies were used sequentially, i.e., one strategy after the other. According to this table, the "Tour quality %" column shows the divergence in the solution quality between the developed approach and the best known. "Time" column shows the runtime required to find the optimized complete solution. A significant feature of the developed approach is its computational complexity, which is close to linear. These results demonstrate that the developed approach can be used for large-scale problems to get high quality solutions in a reasonable amount of time. For the TSP with 200K points, the solution quality is only 0.03463% less comparatively with the best known. All experiments were executed using the LKH algorithm [Helsgaun, 2000, 2006] as the basis for all subsets.

Table 1. Experimental results

Test-case	Problem size (number of points)	Length of the initial solution	Length of the optimized solution	Time (minutes)	Length of the best known solution	Tour quality %
mona-lisa100K	100000	5 758 988	5 757 516	121	5 757 191	- 0.00565
vangogh120K	120000	6 545 620	6 544 127	178	6 543 610	- 0.00790
venus140K	140000	6 812 666	6 811 271	213	6 810 654	- 0.00905
pareja160K	160000	7 622 498	7 620 636	229	7 619 953	- 0.00896
courbet180K	180000	7 891 519	7 889 462	280	7 888 733	- 0.00924
earring200K	200000	8 174 726	8 174 507	295	8 171 677	- 0.03463

Conclusion

The large-scale intractable combinatorial NP-hard problems must have specially developed approaches to receive the high quality solutions. It was proposed to divide the problem solving into two main stages: receiving the initial solution and its optimizing. Four decomposition methods were proposed to get the initial solutions: the macromodeling for the clustered TSP, the extending partial solutions method, and the "Ring" method for the arbitrary TSP. Last of them is appropriate for wide parallelization. A few methods for solution optimization were developed: scanning along the route, geometrical scanning of the whole surface, scanning around the clusters perimeters, and for selected “critical” areas. Suggested approaches have computation complexity close to linear what makes them suitable for large-scale problems. They provide substantial reduction in running time in comparison with the best currently known TSP heuristics. The quality loss is small (≈ 0,006% – 0,03% for 100,000 - 200,000-points instances).

Further work will be directed towards improving the solutions quality, reducing the running time by using better basic algorithms as well as towards developing some new efficient methodologies, especially for the parallel optimization algorithms.

Bibliography

[Applegate, 2009] D. Applegate, R. Bixby, V. Chvátal, W. Cook, D. Espinoza, M. Goycoolea, K Helsgaun: Certification of an Optimal TSP Tour through 85,900 Cities, Operations Research Letters, No. 37, pp. 11-15, 2009.

[Bazylevych, 2007] 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.

[Bazylevych, 2008] R. Bazylevych, R. Kutelmakh, B. Prasad, L. Bazylevych: Decomposition and Scanning Optimization algorithms for TSP, Proceedings of the International Conference on Theoretical and Mathematical Foundations of Computer Science, Orlando, USA, pp. 110-116, 2008.

[Bazylevych, 2009] 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.

[Bazylevych, 2011] R. Bazylevych, R. Dupas, B. Prasad, B. Kuz, R. Kutelmakh, L. Bazylevych: A Parallel Approach for Solving a Large-Scale Traveling Salesman Problem. Proc. of the 5-th Indian Intern. Conf. on Artificial Intelligence, IICAI-2011, India, Dec., 2011, pp. 566- 579.

[Concorde] http://www.tsp.gatech.edu/concorde.html

[Delaunay, 1934] B. Delaunay: Sur la sphère vide, Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk, No 7: pp. 793–800, 1934.

[Helsgaun, 2000] K. Helsgaun: An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic, European Journal of Operational Research 126 (1), pp. 106-130, 2000.

[Helsgaun, 2006] K. Helsgaun: An Effective Implementation of k-Opt Moves for the Lin–Kernighan TSP Heuristic, Datalogiske Skrifter (Writings on Computer Science), No. 109, Roskilde University, 2006.

[Reinelt, 1994] G. Reinelt: The Traveling Salesman Problem: Computational Solutions for TSP Applications // Lecture Notes in Computer Science. – 840, Springer-Verlag. – Berlin. – 1994.

[Rohe, 1997] A. Rohe: Parallel Lower and Upper Bounds for Large TSPs // ZAMM. – 1997. – 77(2). – P. 429-432.

[TSP Art Instances] http://www.tsp.gatech.edu/data/art/index.html

[Yll Haxhimusa, 2009] Yll Haxhimusa, Walter G. Kropatsch, Zygmunt Pizlo, and Adrian Ion. Approximative Graph Pyramid Solution of the E-TSP. Image and Vision Computing, 27(7):887–896, 2009.

Authors' Information

Roman Bazylevych – *Prof. dr.hab., Mathematics and Computer Science Foundations, University of Information Technology and Management in Rzeszow;*

email:rbazylevych@wsiz.rzeszow.pl;

Major Fields of Scientific Research: Computer science, Design automation, Algorithms, Combinatorial optimization

Marek Pałasiński – *Prof. nadzw. dr.hab., Chair of Mathematics and Computer Science Foundations, University of Information Technology and Management in Rzeszow;*

e-mail: mpalasinski@wsiz.rzeszow.pl;

Major Fields of Scientific Research: Theoretical computer science, Theory of algorithms, Graph theory, Data mining and Algebraic logic

Roman Kutelmakh – *Assistant Professor, Ph.D., Software Engineering Department, Lviv Polytechnic National University, 12 S.Bandery Str., Lviv, 79013, Ukraine;*

e-mail: rkutelmakh@ua.fm;

Major Fields of Scientific Research: Software technologies, Combinatorial optimization

Bohdan Kuz – *PhD student, Software Engineering Department, Lviv Polytechnic National University, 12 S.Bandery Str., Lviv, 79013, Ukraine;*

e-mail: bohdankuz@gmail.com;

Major Fields of Scientific Research: Software technologies, Combinatorial optimization

Lubov Bazylevych – *senior scientist, Institute of Mechanical and Mathematical Applied Problems of the Ukrainian National Academy of Sciences;*

e-mail: lbaz@iapmm.lviv.ua;

Major Fields of Scientific Research: Applied mathematics and mechanics, Combinatorial optimization, Computer science