SCIENCE CHINA
Information Sciences

• LETTER •

November 2019, Vol. 62 219103:1–219103:3
https://doi.org/10.1007/s11432-018-9829-5

On the neutrality of two symmetric TSP solvers toward instance specification

Gloria Cerasela CRIŞAN[1*], László Barna IANTOVICS[2*] & László KOVÁCS[3]

[1]*Faculty of Sciences, Vasile Alecsandri University, Bacău* 600115*, Romania;*
[2]*Faculty of Sciences and Letters, University of Medicine, Pharmacy, Sciences and Technology, Targu Mures* 540139*, Romania;*
[3]*Faculty of Mechanical Engineering and Informatics, University of Miskolc, Miskolc* H-3515*, Hungary*

Received 9 October 2018/Revised 28 November 2018/Accepted 15 March 2019/Published online 18 September 2019

Citation Crişan G C, Iantovics L B, Kovács L. On the neutrality of two symmetric TSP solvers toward instance specification. Sci China Inf Sci, 2019, 62(11): 219103, https://doi.org/10.1007/s11432-018-9829-5

Dear editor,
Currently, many computationally difficult problems can be solved using very efficient methods. Some of these state-of-the-art methods have online implementations. The research question addressed herein is how sensitive are such implementations if the input data are preprocessed in a specific manner? The symmetric traveling salesman problem (sTSP), which is an NP-hard problem with many real-life applications is studied. The proposed method includes systematic transformation using rotations and reflections of the vertex order of sTSP instances. This model was used for investigating the neutrality of Concorde [1] (currently the best exact sTSP solver) and Lin–Kernighan implementation [2], both from NEOS [3] (the state-of-the-art collection of online tools in computational optimization).

Methodology. Given a set of n vertices together with their complete set of distances, traveling salesman problem (TSP) seeks for a closed tour with the least length, and each vertex is visited exactly once. TSP is one of the most investigated problems in combinatorial optimization, with broad theoretical developments and many societal applications. In symmetric TSP (sTSP), the distance between any two cities is assumed to be the same in both directions. Exact solvers for sTSP (such as Concorde [1]) are very resource-intensive in most cases. Heuristic approaches (such as the Lin–Kernighan method [2]) provide rapid and often very good results. The implementations of these two methods (from NEOS [3]) require (among other data) the list of vertices and their coordinates. Our goal was to explore the behavior of the Concorde and Lin–Kernighan solvers when the list of vertices is systematically permuted.

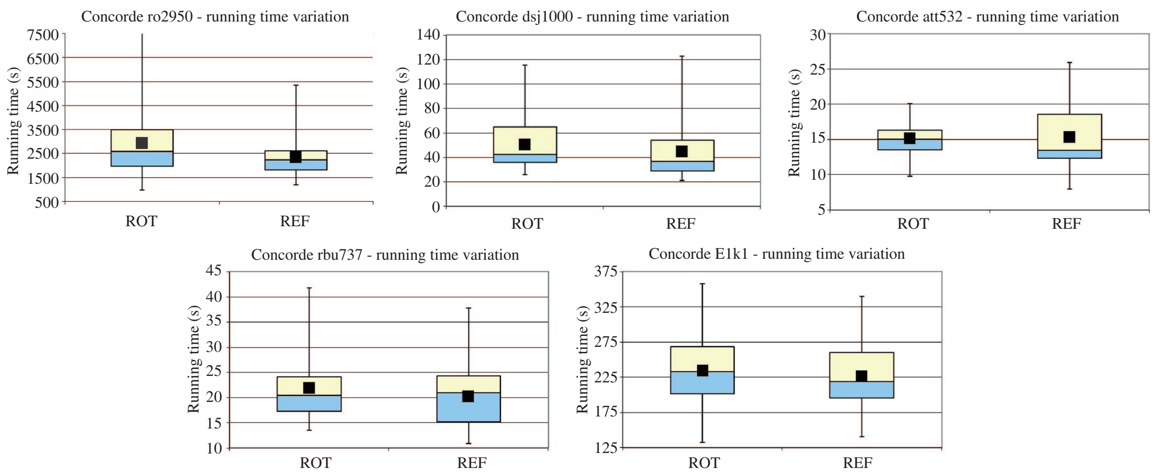
We selected five sTSP instances with different characteristics. One sTSP instance was romania2950.tsp, the first national set of 2950 real localities specified by geographic coordinates and great-circle distances in meters (the GEO norm) [4]. From [5] we chose dsj1000.tsp (a semi-structured, clustered random instance with CEIL_2D norm — the Euclidean 2D distance rounded up to the next integer) and att532.tsp (a real-world instance, with 532 cities from the USA; it has the ATT norm, which is a "special pseudo-Euclidean distance"). From [6] we focused on rbu737.tsp, a drilling VLSI, structured instance with 737 vertices and the Euclidean 2D norm (the nearest integer value). The last instance was E1k1 from [7]; it is a random sTSP instance with 1000 vertices and the Euclidean 2D norm.

Given a list of vertices, we decided to permute the list and derive two groups of files:

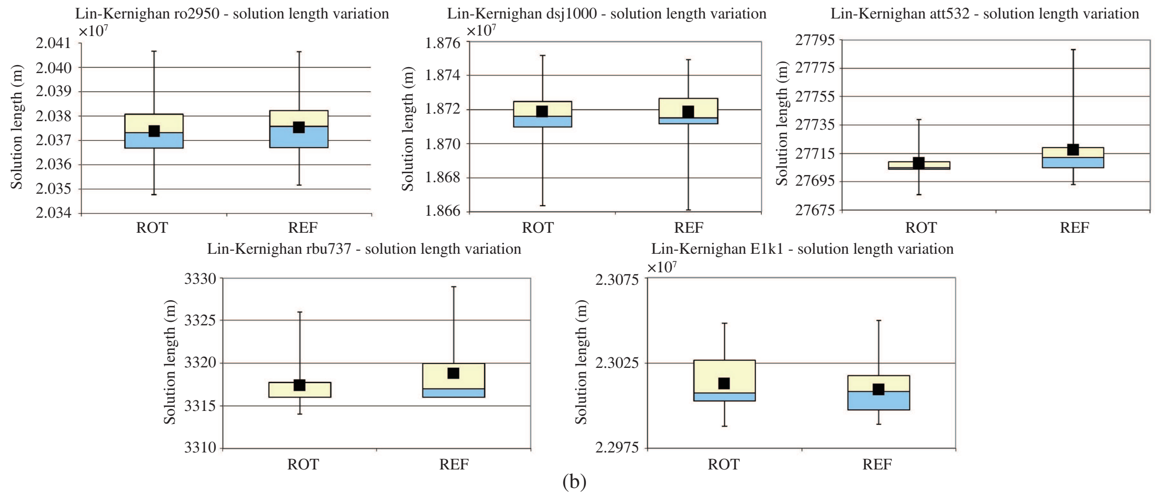
(1) The rotated files. The first block of 25 vertices is recursively moved to the end of the file.

(2) The reflected files. The list is first reverted

* Corresponding author (email: ceraselacrisan@ub.ro, ibarna@science.upm.ro)



(a)



(b)

Figure 1 (Color online) Result charts on rotated (ROT) vs. reflected (REF) files for Concorde and Lin–Kernighan. (a) Variation of the Concorde solving time, grouped by instance; (b) variation of the Lin–Kernighan solution length, grouped by instance.

and then the rotations are performed recursively, like for the rotated group.

The five chosen sTSP instances derived 492 files, which were uploaded on NEOS [3] and solved using both methods with implicit parameters. The total execution time was 180 h.

Results and discussion. For Concorde, we extracted the solving time (in seconds), the initial solution length (in meters), and the number of bbnodes (branching nodes). For the Lin–Kernighan solver, we gathered the initial solution length (obtained using the Quick Borůvka method) and the solution length (both in meters).

To test the neutrality of the Concorde and Lin–Kernighan solvers when the same sTSP instance is specified using block rotations or reflections of its vertices, we performed a characterization based on descriptive statistics. The results (rounded to integer values) for solving all 492 files are given in Appendix A. The charts that comparatively present the results for the ROT and REF files are provided in Figure 1.

The ranges of the results for Concorde show that this exact solver is not neutral to the order of vertices. For example, for romania2950, the ROT files execution time spanned from 16 min to 2 h. The time needed for the REF files was less dispersed: from 20 min to 1.5 h. The number of branching nodes exhibited a high variance. The initial solution length was stable, as its range was approximately 61 km, representing only 0.3% of the average value (20368 km). The REF specifications show a narrower range for Concorde. For the same instance, the Lin–Kernighan solver is stable, as its solution range was approximately 57 km, representing 0.3% of the average length (20374 km). This is achieved although it started from a broader range of initial solution lengths: 1830 km, representing 7.4% of the average length (24705 km). For the dsj1000 solving time, Concorde provided a large range: 90 s for the ROT files and 102 s for the REF files. The instance att532 was again solved using Concorde within a large range. The Lin–Kernighan method delivered lengths in a 0.4% range of the average value for dsj1000. The instance att532 produced results that started from a large range but ended in a very narrow region: the range was 53 km for the ROT files and 95 km for the REF files. It is interesting that the structured instance rbu737 and the random instance E1k1 manifested the smallest percent relative range.

Figure 1(a) shows a strong difference between the behaviors of the ROT and REF files for the largest instance. This empiric investigation reveals an interesting fact: the rotations preserve the neutrality degree. To modify the neutrality degree, one has to perform operations besides rotation on the vertex list. The random instances dsj1000 and E1k1 had more balanced behavior, with almost the same statistics for the ROT and REF files. The structured instance rbu737 and the real-world ones romania2950 and att532 exhibited biased behavior. As the categories ROT and REF are disjoint and can be reverted again, we can conclude that the neutrality degree is influenced by the structure of the sTSP instance. Figure 1(b) shows that the Lin–Kernighan implementation is neutral for larger instances, and the degree of neutrality is the same on ROT and REF groups. For att532, the smallest instance, the REF files produced a larger variation.

Conclusion. Extensive empirical investigations on five symmetric TSP instances prove that the execution time needed by Concorde is not neutral towards rotations and reflections of the vertex lists. The Lin–Kernighan heuristic method is neutral in terms of the quality of the solutions. The tests included random, semi-structured, or structured instances, with EUC_2D, ATT, CEIL_2D, or GEO norms. This study showed that intelligent data pre-processing can be very effective when large, difficult, real-life instances of NP-hard problems need to be solved quickly. These results are in line with other studies, for example [8].

The experimental results are publicly available at web[1)2)3)4)].

Supporting information Appendix A presents the descriptive characterizations for rotated and reflected files. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 University of Waterloo. Concorde TSP solver. 2015. https://uwaterloo.ca/
2 Lin S, Kernighan B W. An effective heuristic algorithm for the traveling-salesman problem. Oper Res, 1973, 21: 498–516
3 Neos Server. Concorde. 2019. https://neos-server.org/neos/
4 University of Bacau. Romanian TSP instance. 2017. http://cadredidactice.ub.ro/
5 University of Heidelberg. TSPLIB. 2018. https://www.uni-heidelberg.de/
6 University of Waterloo. VLSI TSP datasets. 2013. https://uwaterloo.ca/
7 Rutgers University. DIMACS generator for random TSP instances. 2013. http://archive.dimacs.rutgers.edu/
8 Mu Z, Dubois-Lacoste J, Hoos H H, et al. On the empirical scaling of running time for finding optimal solutions to the TSP. J Heuristics, 2018, 6: 879–898

1) DOI 10.13140/RG.2.2.12629.52964/1.
2) DOI 10.13140/RG.2.2.26051.30243/1.
3) DOI 10.13140/RG.2.2.11738.24000/1.
4) DOI 10.13140/RG.2.2.15984.97283/1.