Filomat 35:2 (2018), 1697–1702
https://doi.org/10.2298/FIL1805697W

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

A Quick Method to Compute Sparse Graphs for Traveling Salesman Problem Using Random Frequency Quadrilaterals

Yong Wang[a]

[a]*School of Renewable Energy, North China Electric Power University, Beijing, 102206, China*

Abstract. Traveling salesman problem (*TSP*) is extensively studied in combinatorial optimization and computer science. This paper gives a quick method to compute the sparse graphs for *TSP* based on the random frequency quadrilaterals so as to reduce the *TSP* on the complete graph to the *TSP* on the sparse graphs. When we choose N frequency quadrilaterals containing an edge e to compute its total frequency, the frequency of e in the optimal Hamiltonian cycle will be bigger than that of most of the other edges. We fix N to compute the frequency of each edge and the computation time of the quick method is $O(n^2)$. We suggest two frequency thresholds to trim the edges with the frequency below the two frequency thresholds and generate the sparse graphs for *TSP*. The experimental results show we compute the sparse graphs for these *TSP* instances in the *TSPLIB*.

1. Introduction

Traveling salesman problem (*TSP*) is one typical NP-hard problem in combinatorial optimization and it is NP-complete in computational complexity [1]. Given a set of n points $\{1, 2, \cdots, n\}$ and the distance function $d(u, v) > 0$ for two points $u, v \in \{1, 2, \cdots, n\}$ and uv, a salesman wants to find the shortest cycle, namely the optimal Hamiltonian cycle (*OHC*), that visits each of the n points exactly once. A cycle visiting each of the points exactly once is a Hamiltonian cycle (*HC*) noted as a sequence $\sigma = (v_1, v_2, \cdots, v_n)$ of the n points where $v_k \in \{1, 2, \cdots, n\} (1 \leq k \leq n)$. The distance of σ is computed as $d(\sigma) = d(v_1, v_n) + \sum_{i=1}^{n-1} d(v_i, v_{i+1})$. The *OHC* has the minimum distance. Due to its close relationships with many industrial problems, such as circuit design, vehicle routing, network optimization, machine scheduling, etc., the methods for *TSP* have been extensively studied by the researchers in the related areas [2].

Most people resolve the *TSP* on the complete graph K_n. The accurate methods for *TSP* include the integer programming, branch-and-bound, cutting plane and their variations [3]. Held and Karp [4], and independently Bellman [5] gave the dynamic programming to resolve *TSP* in $O(n^2 2^n)$ time. Until 2010, this best computation time was updated to $O(1.657^n)$ by Björklund [6] using a Monte Carlo method. In theory, these exact algorithms need the exponential computation time for resolving the worst case of *TSP*. The experiments illustrated that the exact algorithms usually consumed long time to tackle the large scale of *TSP* instances [7]. Under the assumption of *PNP*, many researchers turned to the approximation algorithms

2010 *Mathematics Subject Classification*. 05-xx; 05Cxx, 60Bxx
Keywords. Traveling salesman problem; Sparse graph, Frequency quadrilateral.
Received: 9 October 2017; Accepted: 17 November 2017
Communicated by Hari M. Srivastava
Research supported by the Fundamental Research Funds for the Central Universities (No.2015ZD10) and NSFC (No.51205129)
Email address: yongwang@ncepu.edu.cn (Yong Wang)

for *TSP*. In1998, Arora discovered the polynomial-time approximation scheme for the Euclidean *TSP*. However, there is no constant-ratio approximation algorithm for the metric *TSP*, needless to say for the general *TSP* [8].

Comparing the algorithms for *TSP* on the K_n, the exact algorithms for the *TSP* on the sparse graphs have the relatively smaller computation time. Furthermore, the approximation ratio of the approximation algorithms for the *TSP* on the sparse graphs is reduced to some extent. Given a sparse graph of average degree d, the number of the *HC*s is no more than $e(\frac{d}{2})^n$ owing to Sharir and Welzl [9] where e is the base of the natural logarithm. In addition, Björklund, Husfeldt, Kaski and Koivisto [10] proved that the *TSP* can be resolved in $O((2-\varepsilon)^n)$ computation time where ε depends on the maximum degree of the vertex in the sparse graph. Given a 3-regular graph, *TSP* can be resolved in $O(1.2312^n)$ time due to Xiao and Nagamochi [11]. In addition, there exists the polynomial-time approximation scheme for general *TSP* on the bounded genus graph, see the work of Borradaile, Demaine and Tazari [12]. Thus, whether one researches the exact algorithms or studies the approximation algorithms for *TSP*, one will obtain better results on the sparse graphs rather than on the complete graphs of *TSP*.

The following question is how to reduce the K_n to a sparse graph for *TSP*. Hougardy and Schroeder [13] gave a kind of sufficient condition according to $k-opt$ move to discern many edges out of the *OHC*. Their algorithm requires the $O(n^2 \log n)$ computation time. We used the frequency graphs as another way to compute the sparse graphs for *TSP*. The frequency graphs are initially computed with the optimal k-vertex paths with the specified endpoints [14]. In the following, Yong and Remmel [15, 15] discovered the frequency quadrilaterals and gave a binomial distribution model to reduce the number of edges that one has to consider for resolving the *TSP*. They chose $O(n^2)$ frequency quadrilaterals for each edge to compute the frequency graphs and derived the lower bound of the frequency of the *OHC* edges. The method to compute the frequency graphs needs $O(n^4)$ time, which is hard to execute for the big scale of *TSP* instances. Different from the above research, we compute the frequency of edges with a small number of random frequency quadrilaterals where the time complexity of the method is reduced to $O(n^2)$. The quick method will have more broad applications for the real-world *TSP* instances. Moreover, we analyze the change of the minimum frequency of the *OHC* edge according to n and suggest the frequency threshold to compute the sparse graphs containing the *OHC*.

This paper is organized as follows. The frequency quadrilateral is briefly introduced in section 1. The quick method to compute the sparse graphs is given in section 2. In section 3, we shall test the quick method with the *TSP* instances in *TSPLIB* at Heidelberg University and compare the results with the previous research. The conclusions are drawn in the last section.

2. The frequency quadrilateral

Given a quadrilateral *ABCD* in K_n, the distances of the six edges (A,B), (A,C), (A,D), (B,C), (B,D) and (C,D) are noted as $d(A,B)$, $d(A,C)$, $d(A,D)$, $d(B,C)$, $d(B,D)$ and $d(C,D)$, respectively. According to the distances of the six edges, we compute the six optimal 4-vertex paths in the quadrilateral *ABCD*. The frequency quadrilateral *ABCD* is computed with the six optimal 4-vertex paths. There are six frequency quadrilaterals *ABCD* for a quadrilateral *ABCD* [15]. Based on the six frequency quadrilaterals *ABCD*, Wang and Remmel found the frequency of an edge $e \in \{AB, AC, AD, BC, BD, CD\}$ had the frequency $f = 1, 3$ and 5 twice, respectively. Given any one quadrilateral *ABCD* in the K_n, the corresponding frequency quadrilateral *ABCD* will be one of the six frequency quadrilaterals. Thus, they assumed the probability that an edge e has the frequency $f = 1, 3$ and 5 in a frequency quadrilateral containing e is $p_1(e) = p_3(e) = p_5(e) = 1/3$.

For the *OHC* edges, they discovered some special frequency quadrilaterals where the frequency of their frequency f is 3 and 5 rather than 1. For each of the *OHC* edges, there are at least $n-3$ such frequency quadrilaterals. It mentions that every edge is contained in $\binom{n-2}{2}$ frequency quadrilaterals in the K_n. Based on the findings, they formulated the probability model for the *OHC* edges as $p_5(e) = \frac{1}{3} + \frac{1}{3(n-2)}$, $p_3(e) = \frac{1}{3} + \frac{1}{3(n-2)}$ and $p_1(e) = \frac{1}{3} - \frac{2}{3(n-2)}$. We see the probability $p_3(e)$, $p_5(e)$ of the *OHC* edges is bigger than those of the general edges. When we choose N frequency quadrilaterals for each edge to compute their total frequency $F(e)$, the expected frequency of the *OHC* edge e will be $\left(3 + \frac{2}{n-2}\right)N$ which is bigger than the average frequency $3N$ of a general edge.

3. The quick method to compute the sparse graphs

In fact, the probability $p_5(e)$ or $p_3(e)$ is too conservative for the *OHC* edges. One sees the probability $p_1(e)$, $p_3(e)$ and $p_5(e)$ will be equal as n is big enough. When we compute the total frequency $F(e)$ of the edges e according to the probability, there is no difference between the *OHC* edges and the other edges according to their frequency. Our intuition is that the *OHC* will have more intersections of edges with the optimal 4-vertex paths so they should have the bigger $p_3(e)$ and $p_5(e)$ according to n [14–16]. The proof is long and it is still in the preparation process. In theory, the $p_3(e)$ or $p_5(e)$ will reach 1 as n is big enough. This conclusion has been verified by some *TSP* instances where the $p_3(e)$ and $p_5(e)$ are computed with all of the frequency quadrilaterals in K_n [15]. However, the previous methods have the $O(n^4)$ computation time. Here we shall use a small number N of random frequency quadrilaterals to compute the $p_3(e)$ and $p_5(e)$ to reduce the time complexity and verify the performance of the quick method.

There are total $\binom{n}{4}$ quadrilaterals in the K_n. Each edge e is contained in the $\binom{n-2}{2}$ quadrilaterals. Meanwhile, every edge is contained in the corresponding $\binom{n-2}{2}$ frequency quadrilaterals. In each of the frequency quadrilaterals, the frequency of e will be 1 or 3 or 5. Here we choose N frequency quadrilaterals containing e to compute its total frequency $F(e)$. Therefore, the method to compute the frequency of all of edges becomes $O(Nn^2)$. Moreover, we fix N for every *TSP* instances so that the time complexity is $O(n^2)$. With this quick method, one will see we can still compute the sparse graphs for *TSP* with a given frequency threshold. In addition, one will also see the change of the minimum frequency of the *OHC* edges according to n for the general *TSP* instances.

For the *OHC* edges e, if the $p_5(e) = 1$, it means the edge e has the frequency $f = 5$ in each of the frequency quadrilaterals. When we choose N frequency quadrilaterals containing e to compute its total frequency, $F(e) = 5N$. Therefore, we can use a frequency threshold close to $5N$ to trim the edges with the frequency below the frequency threshold and preserve a sparse graph for *TSP*. In real applications, the $p_5(e)$ of some *OHC* edges will be less than 1 as n is small. Before $p_5(e) \approx 1$, the $p_3(e) + p_5(e)$ approaches 1 first as n is big enough. For some small and medium *TSP* instances, the $p_3(e) + p_5(e)$ will tends to 1 first. For the medium and big scale of *TSP* instances, $p_5(e)$ will approach 1 as well as $p_3(e) + p_5(e)$. When we choose N frequency quadrilaterals containing an edge e, the number of the frequency quadrilaterals where e has the frequency $f = 3$ and 5 is $[Np_3(e)]$ and $[Np_5(e)]$, respectively. For the *OHC* edges of small and medium *TSP* instances, $[Np_3(e)] + [Np_5(e)]$ will tend to N. In the worst cases, $p_5(e) = 0$ and the minimum frequency of the *OHC* edge is $F(e) = 3N$. In average case, $p_3(e) = p_5(e) = \frac{1}{2}$ so $F(e) = 4N$. For the *OHC* edges of big scale of *TSP*, $[Np_5(e)]$ will approach N and the $F(e)$ tends to $5N$.

Since the *OHC* edges e have the big probability $p_3(e)$ and $p_5(e)$, we give two methods to compute the frequency of edges. The first method is to compute the frequency of e with the frequency quadrilaterals where it has the frequency $f = 3$ and 5 among the N random frequency quadrilaterals. The second method is to compute the frequency of an edge e with the frequency quadrilaterals where it has the frequency $f = 5$ among the N random frequency quadrilaterals. We also suggest two frequency thresholds to eliminate the edges with the frequency below the two frequency thresholds. For the small and medium scale of *TSP* instances, we use the frequency threshold $F \approx 4N$. For large scale of *TSP* instances, the frequ3ncy threshold $F \approx 5N$ is taken as the frequency threshold.

There is one problem we should explain. Given a quadrilateral *ABCD*, we assume the three sum distances $d(A,B) + d(C,D)$, $d(A,C) + d(B,D)$ and $d(A,D) + d(B,C)$ are unequal to compute the 6 optimal 4-vertex paths. We may have the equal sum distances $d(A,B)+d(C,D) = d(A,C)+d(B,D) = d(A,D)+d(B,C)$ for many quadrilaterals *ABCD* for some *TSP* instances. In this case, the quadrilateral *ABCD* contains 12 optimal 4-vertex paths. It is hard to choose the right 6 optimal 4-vertex paths to compute the frequency quadrilaterals *ABCD* for the *OHC* edges. If the wrong optimal 4-vertex paths are selected, we will use the mistaken frequency quadrilaterals to compute the frequency of the edges. The frequency of the *OHC* edges will become smaller due to the restrictions of the frequency quadrilaterals. To avoid such cases, the small random distances are added to the distances of edges [15]. This method can make the three sum distances $d(A,B)+d(C,D)$, $d(A,C)+d(B,D)$, $d(A,D)+d(B,C)$ unequal as most as possible. The disadvantage is also obvious that the random distances cannot guarantee to compute the big frequency $F(e)$ for the *OHC* edges in every experiment.

4. The experiments and analysis

We shall do experiments to compute the sparse graphs for several Euclidean *TSP* instances in *TSPLIB* [17]. It mentions that the method can be applied to the other kinds of *TSP* instances as well. Given an edge e in K_n, we choose N random quadrilaterals containing e to compute its frequency $F(e)$. If the frequency $F(e)$ is bigger than the given frequency threshold F, it will be preserved. Otherwise, we will neglect it. After all the $\binom{n}{2}$ edges are checked, a sparse graph will be computed. The *OHC*s of these *TSP* instances are computed first with the Concorde Online [18]. The *OHC* is used to verify whether the preserved graphs lose some *OHC* edges. For different *TSP* instances, the frequency threshold F may be different to compute the sparse graphs because they include different percentage of frequency quadrilaterals $ABCD$ where the three sum distances $d(A,B)+d(C,D)$, $d(A,C)+d(B,D)$ and $d(A,D)+d(B,C)$ are equal. The equal sum distances $d(A,B)+d(C,D)$, $d(A,C)+d(B,D)$ and $d(A,D)+d(B,C)$ will reduce the frequency of some *OHC* edges if the wrong optimal 4-vertex paths are largely used.

For each *TSP* instance, we choose the frequency threshold F to compute the sparse graphs containing the *OHC* or losing at most one *OHC* edge. Due to the randomness of the frequency quadrilaterals, the results in various experiments will be different. To show the robustness of the quick method, we did three experiments for every *TSP* instance. The results are shown in Table 1. $N = 100$ is the number of the random frequency quadrilaterals containing each edge. *TH* represents the frequency threshold we used. S_i and l_i means the number of edges in the sparse graphs and the number of the lost *OHC* edges in the i^{th} experiment where $1 \leq i \leq 3$. $\bar{S} = \frac{S_1+S_2+S_3}{3}$ is the average number of the edges in the three sparse graphs. The computation time is trivial for these examples and it is not shown. In Table 1, we compute the frequency of an edge e with the 100 frequency quadrilaterals where its frequency $f = 3$ and 5. In Table 2, the frequency of e is computed with the frequency quadrilaterals where e has the frequency $f = 5$. Therefore, the frequency thresholds *TH* in the two Tables are different.

In Table 1, the *TH* is bigger than $3N$ and it tends to $4N$ according to n. It means the *OHC* edges have more and more percentage of the frequency quadrilaterals where their frequency $f = 3$ and 5 as n rises. In Table 2, we also see the frequency threshold *TH* increases according to n. Although the percentage of the frequency quadrilaterals where the *OHC* edges have the frequency $f = 5$ is smaller than that where they have the frequency $f = 3$ and 5, the percentage of the frequency quadrilaterals where the *OHC* edges have the frequency $f = 5$ is obvious increasing according to n.

We compute the sparse graphs with the proper frequency thresholds *TH* for every *TSP* instance. In Table 1, the number of the edges is reduced more than 4 times as that of the complete graph for the small *TSP* instances. For the big *TSP* instances, it is reduced more than 8 times. The number of the edges will be reduced more times for the bigger *TSP* instances. The number of the edges in the sparse graphs in Table 2 is smaller than that in Table 1 for these *TSP* instances. It says the second method (and frequency threshold) is better than the first method (and frequency threshold) to compute the sparse graphs for *TSP*. The computation time of the quick method is $O(n^2)$. It means the probability model and binomial distribution model [15] still works well for small N so we can design the quick algorithms to compute the sparse graphs for *TSP*.

In the end, we compare the experimental results for D657 and Rat783 in Table 2 with those in the other two papers [15, 16], see Table 3. One sees the quick method computed the better results for D657 but worse results for Rat783. The reason mainly depends on the computation of the frequency $F(e)$ and the frequency threshold *TH*. In addition, the percentage of the frequency quadrilaterals where the *OHC* edges have the frequency 1, 3 and 5 are not considered. We will compute the frequency $F(e)$ with the frequency quadrilaterals where the edges e has the frequency 1, 3 and 5, respectively. Then, give the three frequency thresholds *TH* for comparison to eliminate the edges. We will explore the suitable frequency thresholds in the future work.

Table 1: The experimental results for TSP instances according to the first frequency threshold.

TSP	n	N	TH/N	$S_1 \setminus l_1$	$S_2 \setminus l_2$	$S_3 \setminus l_3$	$\bar{S}/n$
Berlin52	52	100	3.4	289 \ 0	298 \ 0	289 \ 1	6
A280	280	100	3.4	8616 \ 1	8620 \ 1	8608 \ 1	31
D493	493	100	3.5	23331 \ 1	23303 \ 0	23420 \ 1	47
D657	657	100	3.6	38618 \ 0	38494 \ 0	38560 \ 0	58
Rat783	783	100	3.6	51323 \ 1	51236 \ 0	51164 \ 1	65
Pcb1173	1173	100	3.7	104379 \ 1	104809 \ 1	104344 \ 0	89
U1432	1432	100	3.9	118843 \ 1	118885 \ 0	118858 \ 1	83

Table 2: The experimental results for TSP instances according to the second frequency threshold.

TSP	n	N	TH/N	$S_1 \setminus l_1$	$S_2 \setminus l_2$	$S_3 \setminus l_3$	$\bar{S}/n$
Berlin52	52	100	2.7	253 \ 0	253 \ 1	244 \ 1	5
A280	280	100	2.8	6583 \ 0	6494 \ 1	6441 \ 0	23
D493	493	100	2.9	17906 \ 1	17924 \ 1	17950 \ 1	36
D657	657	100	3.0	30638 \ 0	30477 \ 1	30515 \ 1	46
Rat783	783	100	3.0	40735 \ 1	40755 \ 0	40812 \ 0	52
Pcb1173	1173	100	3.2	77291 \ 1	77384 \ 1	77456 \ 1	65
U1432	1432	100	3.3	107204 \ 1	107097 \ 1	107338 \ 1	74

Table 3: The comparisons of the three methods for D657 and Rat783.

TSP	n	N	N[15]	N[16]	$\bar{S}$	$\bar{S}$[15]	$\bar{S}$[16]
D657	657	100	215496	107748	30222	32404	32509
Rat783	783	100	306153	153076	40716	32582	35732

5. Conclusion

We gave a quick method to compute the sparse graphs for *TSP* based on random frequency quadrilaterals. The computation time is $O(n^2)$. Through experiments, we found the *OHC* edges have more and more percentage of frequency quadrilaterals where they have the frequency $f = 3$ or 5 as n rises. Thus, we can increase the frequency threshold for larger scale of *TSP* to compute the sparse graphs containing the *OHC*. On the other hand, if we use the bigger number N, the quick method may compute the better results whereas it needs more computation time. In the future, we will try different parameters N to compute the sparse graphs and the frequency thresholds will be explored. In addition, we will apply the method to the other kind of *TSP* instances for reducing the computation time of the algorithms for resolving them.

6. Acknowledgement

The authors acknowledge the anonymous referees for their suggestions to improve the paper. We also thank the support of Beijing Key Laboratory of Energy Safety and Clean Utilization.

References

[1] R.M. Karp, On the computational complexity of combinatorial problems, Networks(USA) 5(1975) 45–68.
[2] D. Manerba, R. Mansini, and J. Riera-Ledesma, The Traveling Purchaser Problem and its variants, European Journal of Operational Research 259 (2017) 1–18.
[3] W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, A. Schrijver, Combinatorial Optimization, (1st edition), John Wiley & Sons, New York, 1998.
[4] M. Held and R.M. Karp, A dynamic programming approach to sequencing problems, Journal of the Society for Industrial and Applied Mathematics 10(1962) 196–210.

[5] R.E. Bellman, Dynamic programming treatment of the travelling salesman problem, Journal of the ACM 9 (1962) 61–63.
[6] A. Björklund, Determinant sums for undirected Hamiltonicity, Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, Las Vegas, USA, 23-26 Oct. 2010, pp.173–182.
[7] M.S. Levine, Finding the right cutting planes for the TSP, ACM Journal of Experimental Algorithmics 5(2000), Article No.6.
[8] S. Sahni and T. Gonzales, P-complete approximation problem, Journal of the ACM 23(1976) 555–565.
[9] M. Sharir and E. Welzl, On the number of crossing-free matchings, cycles, and partitions, SIAM Journal on Computing 36(2006) 695–720.
[10] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto, The traveling salesman problem in bounded degree graphs, ACM Transactions on Algorithms 8(2012), Article No.18.
[11] M. Xiao and H. Nagamochi, An exact algorithm for TSP in degree-3 graphs via circuit procedure and amortization on connectivity structure, Algorithmica 74(2016) 713–741.
[12] G. Borradaile, E.D. Demaine, and S. Tazari, Polynomial-time approximation schemes for subset-connectivity problems in bounded-genus graphs, Algorithmica 68(2014) 287–311.
[13] S. Hougardy and R.T. Schroeder, Edges elimination in TSP instances, Graph-Theoretic Concepts in Computer Science, WG2014, D. Kratsch, I. Todinca (Eds), Lecture Notes in Computer Science, Vol. 8747, Springer, Berlin, Heidelberg, 2014, pp.275–286.
[14] Y. Wang, An approximate method to compute a sparse graph for traveling salesman problem, Expert Systems with Applications 42(2015) 5150–5162.
[15] Y. Wang and J. B. Remmel, A binomial distribution model for the traveling salesman problem based on frequency quadrilaterals, Journal of Graph Algorithms & Applications 20(2016) 411–434.
[16] P.Y. Wang, A Probability Model Based on Frequency Quadrilaterals for Travelling Salesman Problem, International Conference on Applied Mathematics, Simulation and Modelling (AMSM 2016), Beijing, China, 28-29 May, 2016, pp.28–31.
[17] G. Reinelt, http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/.
[18] H. Mittelmann, NEOS Server for Concorde. http://neos-server.org/neos/solvers/co:concorde/TSP.html.