International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:12, No:3, 2018

# An Improved Method to Compute Sparse Graphs for Traveling Salesman Problem

Y. Wang

***Abstract*—**The Traveling salesman problem (TSP) is NP-hard in combinatorial optimization. The research shows the algorithms for TSP on the sparse graphs have the shorter computation time than those for TSP according to the complete graphs. We present an improved iterative algorithm to compute the sparse graphs for TSP by frequency graphs computed with frequency quadrilaterals. The iterative algorithm is enhanced by adjusting two parameters of the algorithm. The computation time of the algorithm is $O(CN_{\max}n^2)$ where $C$ is the iterations, $N_{\max}$ is the maximum number of frequency quadrilaterals containing each edge and $n$ is the scale of TSP. The experimental results showed the computed sparse graphs generally have less than $5n$ edges for most of these Euclidean instances. Moreover, the maximum degree and minimum degree of the vertices in the sparse graphs do not have much difference. Thus, the computation time of the methods to resolve the TSP on these sparse graphs will be greatly reduced.

***Keywords*—**Frequency quadrilateral, iterative algorithm, sparse graph, traveling salesman problem.

## I. Introduction

TSP is extensively studied in combinatorial optimization. Given a set of $n$ points $\{1,2,\cdots,n\}$ and the distances $d(u,v) > 0$ between each pair of points $u,v \in \{1,2,\cdots,n\}$ and $u \neq v$, a salesman expects to find the permutation $\boldsymbol{\sigma} = (\sigma_1,\sigma_1,\cdots,\sigma_n)$ of the $n$ points which makes the total distance $d(\boldsymbol{\sigma}) = d(\sigma_1,\sigma_n) + \sum_{i=1}^{n-1} d(\sigma_i,\sigma_{i+1})$ as small as possible, where $\sigma_i \in \{1,2,\cdots,n\}$. We call the cycle visiting each of the points exactly once the Hamiltonian cycle (HC) and the HC with the minimum distance the optimal Hamiltonian cycle (OHC). Karp [1] has shown that the TSP is NP-complete. It means that there is no exactly polynomial-time algorithms for TSP unless *P=NP*.

Most researchers resolve the TSP according to a given complete graph $K_n$. The complete graph $K_n$ contains $n$ vertices and $\binom{n}{2}$ edges. The primary idea is to choose $n$ edges from the $\binom{n}{2}$ edges to find the OHC under the given constraints. In 1962, Held and Karp [2], and independently Bellman [3] used the dynamic programming to resolve the TSP in $O(n^2 2^n)$ time. This is taken as the best computation time for more than half century. In 2010, the computation time was updated to $O(1.657^n)$ by Björklund [4] through an improved Monte Carlo algorithm. In real-world applications, the Concorde package based on the branch-and-bound and cutting plane methods resolved the large VLSI problem of 85,900 vertices [5]. It is the largest Euclidean TSP that was resolved until now. The experimental results also demonstrated that the computation time of the exact methods was hard to reduce for large scale of TSP.

Besides the exact methods or algorithms for TSP, the approximation algorithms are also focused on by many researchers. If the TSP obeys the triangle inequality, we have the 1.5-approximation Christofide's algorithm [6]. For the $d$-dimension Euclidean TSP, Arora [7] introduced the polynomial-time approximation schemes in 1998. For the Metric-TSP, Mömke and Svensson [8] gave the 1.461-approximation polynomial-time algorithm. According to the complete graph, whether the exact algorithms or the approximation algorithms are difficult to improve for the worst case of TSP. On the other hand, the computation time of the TSP on the sparse graphs has been reduced to some extent. Given the TSP on a bounded degree graph, it can be resolved in $O((2-\varepsilon)^n)$ computation time owing to Björklund, Husfeldt, Kaski and Koivisto [9] where $\varepsilon$ depends on the maximum degree of a vertex. In a special case, Xiao and Nagamochi [10] proposed the algorithm which has the $O(1.2312^n)$ computation time for TSP on cubic graphs. Besides the improvement of the exact methods for TSP on sparse graphs, Correa, Larŕ and Soto [11] proved there existed the strictly 4/3-approximation algorithm for TSP on a 3-regular graph. Moreover, there are polynomial-time approximation schemes for TSP on bounded genus graphs, see the literatures [12], [13]. For general TSP on the complete graphs $K_n$, we do not have such good exact or approximation algorithms.

One sees the sparse graph is helpful to reduce the computation time of the algorithms for the NP-hard TSP. To reduce the computation time of the methods for TSP, one feasible method is to reduce the TSP on $K_n$ to the TSP on the sparse graphs. In 2004, Hougardy and Schroeder [14] introduced a combinatorial algorithm based on the $k$-opt move to eliminate many edges excluding from the OHC. The experimental results showed the computation time of the Concorde package was much reduced to resolve the TSP on the sparse graphs they computed. In our previous research, we computed the sparse graphs for TSP through the frequency graphs computed with the specific optimal 4-vertex paths [15] or frequency quadrilaterals [16]. In these frequency graphs, the frequency of the OHC edges is generally much higher than those of most of the other edges. We have proven that when we choose $N$ frequency quadrilaterals containing an edge $e$ to compute its total frequency $F(e)$, the $F(e)$ of the OHC edges approach the maximum value $5N$ as $n$ is big enough (see the appendix). Thus, we can eliminate the edges with the small frequency below a given frequency threshold $F$ and preserve the subgraphs of $K_n$ containing the OHC for TSP. Based on the

Y. Wang is with the North China Electric Power University, Beijing, 102206 PRC (corresponding author, phone: 86-6177-2051; fax: 86-6177-2304; e-mail: yongwang@ ncepu.edu.cn).

The authors acknowledge the funds supported by the Fundamental Research Funds for the Central Universities (No.2015ZD10) and NSFC (No.51205129).

above work, we designed an iterative algorithm to compute the sparse graph from a given $K_n$ for TSP. The performance of the algorithm mainly depends on the two parameters *N* and *F* in the computation process. In theory, the bigger the parameter *F* is, the sparser the residual graph will be computed. In addition, the smaller the *N* is, the sparser the residual graph will be. However, the experimental results showed the residual graphs lost many OHC edges if we used the small number *N* and big value *F*. In this paper, we shall explore the methods to change the *N* and *F* in the computation process to improve the performance of the iterative algorithm. Our goal is to compute the sparse graph which loses the number of the OHC edges as small as possible.

The outline of the paper is given as follows. The improved iterative algorithm is introduced in Section II. The experiments and analysis of the improved iterative algorithm are executed in Section III. The conclusions are drawn in the last section as well as the possibilities of the future research.

## II. The Improved Iterative Algorithm

Given a quadrilateral ABCD in $K_n$, it has the six corresponding frequency quadrilaterals ABCD according to the orders of the three sum distances $d(\mathrm{A},\mathrm{B})+d(\mathrm{C},\mathrm{D})$, $d(\mathrm{A},\mathrm{C})+d(\mathrm{B},\mathrm{D})$ and $d(\mathrm{A},\mathrm{D})+d(\mathrm{B},\mathrm{C})$. For an edge $e \in \{\mathrm{AB},\mathrm{AC},\mathrm{AD},\mathrm{BC},\mathrm{BD},\mathrm{CD}\}$, it has the frequency 1, frequency 3 and frequency 5 in each of the six frequency quadrilaterals. In addition, the frequency 1, frequency 3, frequency 5 of *e* appears twice in the six frequency quadrilaterals, respectively. Based on the six frequency quadrilaterals ABCD, the probability $p_1(e)$, $p_3(e)$and $p_5(e)$ that *e* has the frequency 1, frequency 3 and frequency 5 in a given frequency quadrilateral containing *e* is computed as $\frac{1}{3}$. An edge $e \in \{\mathrm{AB},\mathrm{AC},\mathrm{AD},\mathrm{BC},\mathrm{BD},\mathrm{CD}\}$ in $K_n$ is included in $\binom{n-2}{2}$ quadrilaterals ABCD. There are corresponding $\binom{n-2}{2}$ frequency quadrilaterals ABCD containing *e*. When we choose *N* frequency quadrilaterals containing *e* to compute its total frequency *F*(*e*), the average value of *F*(*e*) will be 3*N*. For an OHC edge *e*, Wang and Remmel [16] constructed the special quadrilaterals ABCD where its frequency is 5 or 3. They gave the probability model for the OHC edges as (1).

$$\begin{cases} p_5(e) = \frac{1}{3} + \frac{1}{3(n-2)} \\ p_3(e) = \frac{1}{3} + \frac{1}{3(n-2)} \\ p_1(e) = \frac{1}{3} - \frac{2}{3(n-2)} \end{cases} \quad (1)$$

When we choose *N* frequency quadrilaterals containing the OHC edge *e* to compute its frequency *F*(*e*), the average value of *F*(*e*) will be $\left(3+\frac{2}{n-2}\right)N$ which is bigger than the average frequency 3*N* of a general edge.

The authors pointed out that the probability model (1) is very conservative for the OHC edges. As a matter of fact, the probability $p_{3,5}(e) = p_3(e) + p_5(e) \to 1$ and even $p_5(e) \to 1$ for the OHC edges as *n* is big enough. The appendix gives the proof of the probability $p_{3,5}(e) \to 1$ or $p_5(e) \to 1$ that an OHC edge *e* has frequency 3 and frequency 5 or only 5 in a frequency quadrilateral containing *e* in $K_n$ as *n* is big enough. When we choose *N* frequency quadrilaterals containing an OHC edge *e* to compute its total frequency *F*(*e*), the *F*(*e*) of the OHC edge will tend to 5*N* as *n* is big enough. Since the OHC edges have the big probability $p_{3,5}(e)$ and $p_5(e)$, most of the other edges will have the small probability $p_{3,5}(e)$ and $p_5(e)$ due to the restrictions of the frequency quadrilaterals. When we use a big frequency threshold *F* to eliminate the edges with low frequency *F*(*e*), most of the other edges will be discarded. Moreover, the bigger the frequency threshold is, the less the edges will be preserved. This fact has been verified by several families of TSP instances in TSPLIB [15], [16].

To compute the big *F*(*e*) for the OHC edges, the author gave an applicable method to choose the frequency quadrilaterals for edges. For an edge *e* with the distance *d*(*e*), they choose the *N* quadrilaterals ABCD containing the edges *e* and *g* where *g* is the opposite edge of *e* in ABCD and *d*(*g*)<*d*(*e*). This method guarantees the edge *e* will have the frequency 3 and frequency 5 in most of the corresponding frequency quadrilaterals ABCD if *e* has the big probability $p_{3,5}(e)$ and $p_5(e)$ in the graphs. Thus, we can use a bigger frequency threshold *F* to eliminate more other edges with small frequency and still keep the OHC edges intact. Based on the selection of the frequency quadrilaterals, an iterative algorithm was designed to eliminate the edges excluding from the OHC step by step. Through the experiments, the iterative algorithm is mainly influenced by the two parameters *N* and *F*. The bigger the parameter *F* is, the sparser the residual graph will be computed. And the smaller the *N* is, the sparser the residual graph will be. On the other hand, the residual graph will lose the OHC edges if the value *F* is too big. Meanwhile, the value *N* should not be too small. Otherwise, the probability model (1) will lose its power. To compute the sparse graph that includes the OHC edges as most as possible, we increase the parameters *F* by the small values *a* step by step until *F* approaches 5*N*. The frequency threshold *F* is used to eliminate more and more edges with the smaller *F*(*e*) below *F* in the computation process. Meanwhile, the parameter *N* is increased by a small value *b* gradually to choose more and more number of frequency quadrilaterals where an edge *e* has the big frequency 3 and frequency 5 if *e* has the big probability $p_{3,5}(e)$ and $p_5(e)$ in the preserved graphs. In addition, we maintain the first *n* edges with the smallest distances in the preserved graphs in the computation process. The improved iterative algorithm is given as Fig. 1.

We have seven parameters *N*, *F*, *t*, *d*, *r*, *a* and *b* in the iterative algorithm. The performance of the iterative algorithm mostly depends on the four parameters *N*, *F*, *a* and *b*. *N* represents the number of frequency quadrilaterals containing an edge *e*. *F* is the frequency threshold to eliminate the edges with *F*(*e*) below *F* at each computation cycle. The *F* and *N* are changing according to the computation cycle *C*.

*a* and *b* are the increments of *F* and *N* in the computation process, respectively. If *F* is too small for the edges *e* in the preserved graphs, few edges will be eliminated. In such cases, *F* is risen by adding the value *a*. Meanwhile, the *N* is added by the

number $b$. The parameters $t$, $d$ and $r$ correspond to three conditions. The iterative algorithm is controlled by the terminal conditions $d_{avg}^{in} - d_{avg}^{out} < t$ and $d_{\min} < d$. The $d_{avg}^{in}$ and $d_{avg}^{out}$ are the average degrees of the input graph and output graph, respectively. The $t$ is the parameter to evaluate the difference between $d_{avg}^{in}$ and $d_{avg}^{out}$. If $d_{avg}^{in} - d_{avg}^{out}$ is very small, it means that few edges are eliminated at the computation cycles and the algorithm converges to the current sparse graph. Thus, the $t$ is assigned a small value as one of the terminal conditions to prove the preserved graphs are nearly identical at the end of the computation process. The $d_{\min}$ is the minimum frequency of the vertices in the output graph. The parameter $d$ is used to let the algorithm compute the residual graph whose $d_{\min}$ is less than $d$. The parameter $r$ drives the algorithm to change the $F$ and $N$. If $d_{avg}^{in} - d_{avg}^{out} > r$, it means the current parameters $F$ and $N$ are feasible to eliminate many edges at these computation cycles. Thus, the $F$ and $N$ keep unchangeable. Once $d_{avg}^{in} - d_{avg}^{out} < r$, the parameter $F$ becomes small for the edges in the preserved graphs and few edges are eliminated according to the current $F$. It is the time to improve the $F$ by adding the value $a$ in the following computation cycles until the algorithm is able to eliminate many edges. Meanwhile, the parameter $N$ is added by the number $b$. In the improved algorithm, the parameter $a$ is assigned a small value, such as 0.001. The frequency threshold $F$ rises gradually to eliminate the edges with the frequency $F(e)$ below the $F$ step by step.

The $K_n$ of the TSP instance is input and the parameters are initialized. The iterative algorithm contains three loops. The outer loop is controlled by the terminal conditions $d_{avg}^{in} - d_{avg}^{out} < t$ and $d_{\min} < d$. In the outer loop, the $K$ edges in the input graph $I$ are ordered according to their distances from small to big values to form an edge sequence $\boldsymbol{s} = (e_1, e_2, \cdots, e_K)$. The average degree $d_{avg}^{in}$ of $I$ is computed. The next is the condition to judge whether the $F$ and $N$ should be improved or they are kept unchangeable. In addition, we preserve the first $n$ shortest edges in the edge sequence $\boldsymbol{s} = (e_1, e_2, \cdots, e_K)$. This strategy is helpful to maintain the OHC edges in the computation process. The following procedures are the middle and inner loops. For an edge $e_i (n < i \le K)$, $N$ opposite edges $g$ of $e_i$ in $N$ quadrilaterals ABCD are chosen at random from the subsequence $(e_1, e_2, \cdots, e_i)$ where $i < K$. The $N$ quadrilaterals containing $e_i$ and $g$ are converted into the corresponding frequency quadrilaterals. If the $e_i$ has the frequency $f_{ik}$ in the $k^{\text{th}}$ $(1 \le k \le N)$ frequency quadrilateral ABCD, the total frequency $F(e_i) = \sum_{k=1}^{N} f_{ik}$. At the end of the middle loop, we compare the $F(e_i)$ with the frequency threshold $F$. If $F(e_i) > F$, the edge $e_i$ is preserved. Otherwise, it will be eliminated. After all the $K$ edges in the input graph $I$ are traversed, we will obtain the output graph $O$ with a smaller number of edges. The average degree $d_{avg}^{out}$ of the output graph $O$ is computed and the input graph $I$ is replaced by the output graph $O$. The computation process is iterated until the output graph meets the terminal conditions. It mentions that the selection of the quadrilaterals ABCD containing $e_i$ guarantees the edge $e_i \in (e_1, e_2, \cdots, e_K)$ to have the frequency 3 and frequency 5 in most of the $N$ corresponding frequency quadrilaterals ABCD if $e_i$ has the big probability $p_{3,5}(e)$ or $p_5(e)$ in the preserved graphs. As we know, the OHC edges have the big probability $p_{3,5}(e)$ or $p_5(e)$ in the graphs as $n$ is big enough. It means the OHC edges will have the big probability $p_{3,5}(e)$ or $p_5(e)$ in some subgraphs in $K_n$. If the algorithm is feasible, it will preserve these subgraphs for the OHC edges in the computation process. Thus, we can use the bigger and bigger frequency threshold $F$ to eliminate the other edges with the frequency $F(e)$ below $F$ and obtain the sparser and sparser graphs containing the OHC.

The process to compute the sparse graph

**Input:** The vertex set $\{1,2,\cdots,n\}$ and distances $d(u,v)$ of edges $e = (u,v)$ in the complete graph $K_n$.
Give parameters $N$, $F$, $r$ and terminal conditions $d$, $t$, $d_{avg}^{in} = n - 1$, $d_{avg}^{out} = 0$.
**Repeat until $\boldsymbol{d_{avg}^{in} - d_{avg}^{out} < t}$ and $\boldsymbol{d_{\min} < d}$**
  Enumerate the number of edges $K$ in the input graph $I$.
  Compute the average degree $\boldsymbol{d_{avg}^{in}}$ of the input graph $I$.
  Order the $K$ edges according to their distances from small to big values to form the edge sequence $\boldsymbol{s} = (\boldsymbol{e_1}, \boldsymbol{e_2}, \cdots, \boldsymbol{e_K})$.
  **If** $(\boldsymbol{d_{avg}^{in} - d_{avg}^{out} > r})$ $N$:=$N$, $F$:=$F$.
  **Else** $F$:=$F$+$a$, $N$:=$N$+$b$.
  **End**.
  Preserve the first $n$ edges in the $\boldsymbol{s}$.
  index $i$:=$n$+1.
  **Repeat until $\boldsymbol{i := K}$**
    For $\boldsymbol{e_i}$, $\boldsymbol{F(e_i) := 0}$, index $k$ := 0.
    **Repeat until $\boldsymbol{k}$:=$\boldsymbol{N}$**
      Choose an opposite edge $g$ of $e_i$ at random from the edge subsequence $(\boldsymbol{e_1}, \boldsymbol{e_2}, \cdots, \boldsymbol{e_i})$ where $\boldsymbol{i \le K}$.
      Edge $e_i$ and $g$ compose the $k$th quadrilateral.
      Compute the frequency $f_{ik}$ of $e_i$ in the $k$th quadrilateral.
      Add $f_{ik}$ to $\boldsymbol{F(e_i)}$.
      $k$++.
    **End**.
    **If** $(\boldsymbol{F(e_i) > F})$ Preserve $\boldsymbol{e_i}$.
    **Else** Eliminate $\boldsymbol{e_i}$ and $K$--.
    **End**.
    $i$++.
  **End**.
  Compute the minimum degree $d_{\min}$ of the vertex in the output graph $O$.
  Compute the average degree $\boldsymbol{d_{avg}^{out}}$ of the output graph $O$.
  Assign the output graph to the input graph, i.e., $I$:=$O$.
**End**.
**Output**: The residual graph with $K$ edges.

Fig. 1 The improved iterative algorithm to compute the sparse graph

One sees the total computation time of the algorithm relies on the sorting of the $K$ edges and the computation of the $K$ frequency $F(e_i)$s. Given the computation cycles of the algorithm is $C$ and the maximum value of $N$ is $N_{\max}$, the maximum computation time of the improved iterative algorithm is $\max\{O(Cn^2\log n), O(CN_{\max}n^2)\}$. Since $N_{\max}$ is generally bigger than $\log n$ in the experiments, the maximum computation time of the algorithm will be $O(CN_{\max}n^2)$. In fact, the actual computation time will be less than the $O(CN_{\max}n^2)$ because the number of edges decreases according to the computation cycle $C$. After the first several computation cycles,

the number of the edges in the preserved graphs will decrease to $O(n\log n)$. Furthermore, the number $N$ in the previous computation cycles is much less than $N_{\max}$.

According to the appendix, the probability $p_{3,5}(e)$ of an edge $e$ is $\frac{2}{3}$ in $K_n$. As we know, the probability $p_{3,5}(e)$ of an OHC edge $e$ is bigger than $\frac{2}{3}+\frac{2}{3(n-2)}$ in $K_n$. In addition, the $p_{3,5}(e)$ of the OHC edge $e$ increases according to $n$. Given the probability $p_{3,5}(e)$ of the OHC edge $e$ is $\frac{2}{3}+\epsilon$, the $\epsilon$ will approach $\frac{1}{3}$ as $n$ is big enough. If we use the $\frac{2}{3}+\epsilon$ as the probability threshold to eliminate the edges with the probability $p_{3,5}(e)$ below $\frac{2}{3}+\epsilon$, the probability that an edge $e$ will be eliminated is $P\left(X<\left[\left(\frac{2}{3}+\epsilon\right)N\right]\right)=\sum_{i=1}^{\left[\left(\frac{2}{3}+\epsilon\right)N\right]}\binom{N}{i}\left(\frac{2}{3}\right)^{i}\left(\frac{1}{3}\right)^{N-i}$. Since $\epsilon$ tends to $\frac{1}{3}$ for the OHC edge $e$ as $n$ is big enough, the probability $P\left(X<\left[\left(\frac{2}{3}+\epsilon\right)N\right]\right)$ will approach 1. It means a lot of the other edges will be eliminated if use a big frequency threshold $F$ (or big probability threshold). In the computation process, the edges $e$ with the small probability $p_{3,5}(e)$ will have the small frequency $F(e)$ and they will be eliminated. According to the computation cycle $C$, the preserved edges in the graphs will have the bigger and bigger $p_{3,5}(e)$ and $F(e)$. The probability $p_1(e)$, $p_3(e)$ and $p_5(e)$ of edges will not conform to the probability model (1) in the preserved graphs. In this case, the number of frequency quadrilaterals where an edge $e$ has frequency $f$=3 and $f$=5 in the preserved graphs will violate the binomial distribution $P(X=i)=\binom{N}{i}\left(\frac{2}{3}\right)^{i}\left(\frac{1}{3}\right)^{N-i}$ in the appendix. At the final stages of the computation, nearly all of the edges will have the big probability $p_{3,5}(e)$ and frequency $F(e)$ so few edges are eliminated if the frequency threshold $F$ is not improved.

For the small and moderate scale of TSP instances, the probability $p_{3,5}(e)$ of some OHC edges $e$ will be much smaller than 1 in $K_n$. To preserve these OHC edges, the initial frequency threshold $F$ should not be assigned too big. Otherwise, these OHC edges will be trimmed as well as many other edges. If we compute the $F(e)$ of $e$ with $N$ random frequency quadrilaterals containing $e$, the initial frequency threshold $F$ can be assigned as $\left(3+\frac{2}{n-2}\right)N$. In the next computation cycles, we can use the bigger frequency threshold $F>\left(3+\frac{2}{n-2}\right)N$ to filter out the edges with the small frequency $F(e)$. At the end stages of the computation process, we can use the bigger frequency threshold $F$ to eliminate the other edges with the relatively smaller frequency $F(e)$. On the other hand, some OHC edges will be eliminated if the frequency threshold $F$ is too big at the end of the computation process. In this case, we will obtain the sparse graph which neglects a few OHC edges. If the residual graph is Hamiltonian, it will contain an approximation. The TSP solver will search the approximation according to the sparse graphs. The long edges excluding from the OHC usually have small frequency $F(e)$ due to the restriction of the optimal 4-vertex paths in the graphs. These long edges will be eliminated in the computation process. Since the long edges are deleted, the approximation will be composed of the short edges in the residual graph. Therefore, the approximation will tend to the OHC. In addition, if the residual graph is not Hamiltonian, we will find the segments of the OHC. The efficient algorithms, such as the $k$-opt move algorithm [17], can be applied to find the OHC or approximations based on these OHC segments.

## III. The Experiments and Analysis

We will use the TSP instances in the TSPLIB [18] to testify the improved iterative algorithm. The values of the seven parameters are given in Table I. The initial $N$ and $F$ are 50 and 4.80*50, respectively. If the $d_{avg}^{in}-d_{avg}^{out}<r=3$, the $F$ will be added by $a$ at each of the following computation cycles. Meanwhile, the $N$ will be increased by $b$ at the corresponding computation cycles. To accelerate the computation process, we take $a$=0.005 and $b$=20 if $F\in[4.80N, 4.94N]$. Once $F$ is bigger $4.94N$, we use the values $a$=0.001 and $b$=100. The upper bound of $F$ is $4.995N$. At the end of the computation process, if the output graph conforms to the conditions $d_{avg}^{in}-d_{avg}^{out}<0.006$ and $d_{\min}$<5 or 6, the iterative algorithm will be terminated and outputs the residual graph.

TABLE I
The Values of the Seven Parameters

| $N$ | $F$ | $t$ | $d$ | $r$ | $a$ | $b$ |
|---|---|---|---|---|---|---|
| [50, 6900] | [4.8,4.995]$N$ | 0.006 | 5 or 6 | 3 | 0.005 or 0.001 | 20 or 100 |

The initial $F$=4.8$N$ and the upper bound of $F$ is 5$N$. Most of the other edges will be eliminated according to the frequency threshold $F$ approaching 5$N$. We have done experiments with the parameters $a$=0.01 and $F\in[4.80N, 4.93N]$ to compute the sparse graphs for TSP. It found the algorithm converged to the solution quickly. Because the $F$ increases so quickly that many edges are eliminated within a small number of computation cycles. The drawback is that quite a few OHC edges are lost at the end of the computation process. The smaller parameters $a$=0.001 and bigger $F$ above 4.93$N$ will be tried in this research. The parameter $a$=0.001 guarantees the $F$ rises slowly in the computation process once $d_{avg}^{in}-d_{avg}^{out}<3$. The edges with the frequency $F(e)<F$ will be eliminated gradually. After many edges are trimmed, the maintained edges including the OHC edges will have the big probability $p_{3,5}(e)$ or $p_5(e)$ in the preserved graphs. Even though we use the big frequency threshold $F$, few OHC edges will be eliminated. According to the values $b$=20 and 100, the lower and upper bounds $F\in[4.80N, 4.995N]$ and $a$=0.005 and 0.001, the upper bound of $N$ is computed as 6900. The $N$=6900 may be still small for a few OHC edges for some TSP instances if $F$ is bigger than 4.99$N$. For such TSP instances, one can increase the number $N$ to

choose more number of frequency quadrilaterals where the OHC edges have the frequency 5 to compute the bigger frequency $F(e)$. In fact, the algorithm will jump out before the $N$ tends to 6900 or $F$ approaches $4.99N$ for most TSP instances. It mentions that if we increase the frequency threshold $F$, the number $N$ should be improved simultaneously. At the initial computation cycles, there are a lot of edges in the preserved graphs whereas the $N$ is small. At the end of the computation cycles, the $N$ becomes big whereas we have a small number of edges in the preserved graphs. The change of the parameter $N$ balances the computation time of the algorithm at the different computation cycles. The parameter $d$=5 or $d$=6 guarantees that the algorithm will compute the residual graph where the minimum degree of the vertices is less than 5 or 6.

Before the execution of the improved iterative algorithm, the OHCs of the selected TSP instances are computed with the Concorde online [19]. In view of the OHC, we will see if the preserved graphs contain the OHC or how many OHC edges are lost in the computation process. The algorithm is coded with C++ language and run on two computers. One is a laptop with Intel(R) Core(TM) i7-4712MQ CPU 2.3 GHz, inner memory 4.0GB and the other is a PC with Intel(R) Core(TM) i7-3770 CPU 3.4 GHz and inner memory 8 GB. The experiments for the small TSP instances of scale below 1400 are done on the laptop and those for the other big scales of TSP instances are implemented on the PC. In the experiments, we added the random small distances $rd \in [0,1]$ to the distances of edges before the computation of the frequency quadrilaterals ABCD. The random small distances are too small to change the OHC of these TSP instances. They make the three sum distances $d(\mathrm{A},\mathrm{B})+d(\mathrm{C},\mathrm{D})$, $d(\mathrm{A},\mathrm{C})+d(\mathrm{B},\mathrm{D})$ and $d(\mathrm{A},\mathrm{D})+d(\mathrm{B},\mathrm{C})$ in each quadrilateral ABCD as unequal as possible in $K_n$ so that we can compute the frequency quadrilaterals ABCD. The random small distances are generated with the random function provided by C++ function library.

The experimental results of the improved iterative algorithm are given in Table II. To show the robustness of the algorithm, we did three experiments for each TSP instance. The computation cycles $C$, the number $K$ of edges and the number $l$ of the lost OHC edges in the residual graphs are recorded in the three experiments. The maximum degree $d_{\max}$, average degree $d_{\mathrm{avg}}$ and minimum degree $d_{\min}$ of the vertices in the best and worst residual graphs are presented. The residual graph with the least average degree $d_{\mathrm{avg}}$ is taken as the best residual graph. On the contrary, it is the worst residual graph. The maximum computation time of the three experiments are recorded.

For each TSP instance, the maximum frequency threshold $F$ is different to compute their sparse graphs. We adjust the maximum frequency threshold $F$ for each TSP instance to compute their residual graphs. The residual graphs contain nearly all of the OHC edges and sometimes they lose 1, 2, or 3 OHC edges. If the OHC must be contained in the residual graphs, we can reduce the maximum frequency threshold $F$ by a small value whereas the residual graphs will include a mall number of more edges. In the experiments, we use the maximum frequency threshold $F$ which guarantees that the residual graphs lose no more than 3 OHC edges. The number $nlog_2n$ is computed for comparisons.

TABLE II
THE EXPERIMENTAL RESULTS OF THE IMPROVED ITERATIVE ALGORITHM

| TSP | $n$ | *Type* | $F/N$ | | $C$ | | | $K\backslash l$ | | | $nlog_2n$ | $d_{\max}$ | $d_{\mathrm{avg}}$ | $d_{\min}$ | $d_{\max}$ | $d_{\mathrm{avg}}$ | $d_{\min}$ | $t_{\max}$/s |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| d493 | 493 | Euc. | [4.80,4.980] | 85 | 89 | 89 | 1464\2 | 1456\2 | 1458\0 | 4410 | 13 | 6 | 3 | 13 | 6 | 2 | 10 |
| u574 | 574 | Euc. | [4.80,4.980] | 97 | 102 | 103 | 1829\3 | 1795\3 | 1801\3 | 5261 | 12 | 6 | 2 | 14 | 7 | 3 | 17 |
| rat575 | 575 | Euc. | [4.80,4.984] | 117 | 98 | 110 | 1636\1 | 1727\0 | 1672\1 | 5271 | 11 | 6 | 3 | 10 | 6 | 2 | 21 |
| d657 | 657 | Euc. | [4.80,4.980] | 94 | 100 | 118 | 2268\2 | 2229\3 | 2130\3 | 6149 | 14 | 7 | 3 | 13 | 7 | 3 | 22 |
| u724 | 724 | Euc. | [4.80,4.985] | 115 | 111 | 99 | 2220\2 | 2247\2 | 2318\1 | 6878 | 11 | 6 | 2 | 13 | 6 | 3 | 32 |
| rat783 | 783 | Euc. | [4.80,4.985] | 121 | 113 | 114 | 2493\1 | 2431\1 | 2487\0 | 7527 | 12 | 6 | 2 | 11 | 6 | 3 | 44 |
| pr1002 | 1002 | Euc. | [4.80,4.98] | 113 | 129 | 122 | 4180\1 | 4071\1 | 4010\2 | 9989 | 16 | 8 | 4 | 17 | 8 | 4 | 74 |
| u1060 | 1060 | Euc. | [4.80,4.984] | 106 | 124 | 113 | 4265\2 | 4416\2 | 4479\1 | 10653 | 19 | 8 | 3 | 20 | 8 | 3 | 98 |
| vm1084 | 1084 | Euc. | [4.80,4.975] | 113 | 108 | 108 | 5623\2 | 5637\1 | 5658\1 | 10929 | 23 | 10 | 4 | 21 | 10 | 4 | 108 |
| pcb1173 | 1173 | Euc. | [4.80,4.980] | 155 | 151 | 136 | 5362\2 | 3375\1 | 5488\1 | 11960 | 16 | 9 | 4 | 18 | 9 | 5 | 137 |
| d1291 | 1291 | Euc. | [4.80,4.980] | 163 | 164 | 162 | 5291\2 | 5360\1 | 5351\2 | 13342 | 19 | 8 | 3 | 18 | 8 | 3 | 162 |
| u1432 | 1432 | Euc. | [4.80,4.987] | 139 | 148 | 153 | 5671\1 | 5560\2 | 5673\0 | 15013 | 13 | 8 | 4 | 14 | 8 | 4 | 178 |
| fl1400 | 1400 | Euc. | [4.80,4.98] | 132 | 130 | 151 | 6837\2 | 6886\2 | 6647\2 | 14632 | 55 | 10 | 6 | 51 | 10 | 6 | 207 |
| d1655 | 1655 | Euc. | [4.80,4.98] | 168 | 202 | 186 | 8693\1 | 9089\1 | 9278\0 | 17696 | 20 | 11 | 5 | 18 | 10 | 5 | 325 |
| u1817 | 1817 | Euc. | [4.80,4.99] | 185 | 169 | 170 | 7100\0 | 7208\1 | 7197\1 | 19673 | 17 | 8 | 4 | 16 | 8 | 4 | 510 |
| rl1889 | 1889 | Euc. | [4.80,4.98] | 188 | 178 | 189 | 12559\2 | 12922\1 | 12644\2 | 20559 | 32 | 13 | 6 | 39 | 14 | 6 | 683 |
| d2103 | 2103 | Euc. | [4.80,4.985] | 182 | 191 | 336 | 13582\3 | 13632\3 | 12571\2 | 23213 | 20 | 13 | 5 | 18 | 12 | 6 | 1033 |
| u2152 | 2152 | Euc. | [4.80,4.99] | 182 | 207 | 188 | 9970\1 | 9658\2 | 9886\1 | 23826 | 16 | 9 | 4 | 15 | 9 | 4 | 819 |
| u2319 | 2319 | Euc. | [4.80,4.994] | 175 | 179 | 163 | 8098\0 | 8036\1 | 8221\1 | 25925 | 9 | 7 | 3 | 10 | 7 | 3 | 889 |
| pr2392 | 2392 | Euc. | [4.80,4.99] | 191 | 199 | 190 | 11178\0 | 11119\1 | 11246\1 | 26848 | 16 | 9 | 5 | 16 | 9 | 4 | 1152 |
| pcb3038 | 3038 | Euc. | [4.80,4.993] | 238 | 216 | 225 | 13040\1 | 13403\1 | 13289\1 | 35146 | 20 | 9 | 4 | 21 | 9 | 4 | 2744 |
| fnl4461 | 4461 | Euc. | [4.80,4.994] | 323 | 317 | 1500 | 18533\0 | 18685\1 | 18579\1 | 54081 | 19 | 8 | 3 | 17 | 8 | 4 | 13098 |

One sees the maximum frequency threshold $F$ increases according to $n$ in Table II. It means that we can use the bigger frequency threshold $F$ to compute the sparse graphs for the bigger scale of TSP instances. The experimental results comply with the theory in the appendix. Under the same terminal conditions, the algorithm computes the sparse graphs within a

limitative computation cycles. The number $K$ of edges in the residual graphs is much smaller than $nlog_2n$. The computation time is acceptable on the laptop and PC for these TSP instances. The minimum degree of the vertices in the residual graphs is less than 5 or 6. Moreover, the maximum degree of the vertices in the residual graphs is also restricted. It mentions that for the TSP instances $K_n$ containing many equal-weight edges, it is difficult to compute the appropriate frequency quadrilaterals ABCD even if we add the random small distances. When these inappropriate frequency quadrilaterals are used to compute the frequency of edges, the frequency of some OHC edges will become smaller. To guarantee the residual graph to contain these OHC edges, we have to use the relatively small frequency threshold to eliminate the edges with the even smaller frequency. Thus, the residual graphs with more edges are computed and the maximum degree of the residual graphs will become bigger. For example, the fl1400 and rl1889 contain many equal-weight edges. The maximum degrees of their residual graphs are 55 and 32, respectively. For the other examples with a small number of equal-weight edges, the maximum degree of the residual graphs is much smaller. For the TSP instances without many equal-weight edges, the maximum and average degrees of vertices in the residual graphs do not increase according to $n$. This means the improved iterative algorithm is able to compute the sparse graphs with good properties. In theory, the probability that an OHC edge is lost tends to zero according to the frequency threshold less than $5N$ as $n$ is big enough (see the appendix). In real-world applications, the scale of TSP is limitative. A few OHC edges may have the frequency $F(e) < 5N$ in the preserved graphs. These OHC edges may be lost if the maximum frequency threshold $F$ is close to $5N$. The random small distances also play a role to decrease the frequency of some OHC edges in the experiments, especially for the TSP with many equal-weight edges.

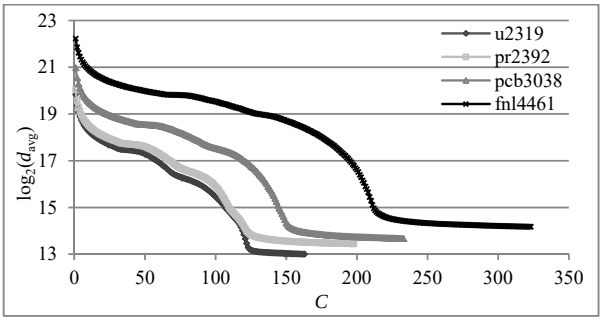


Fig. 2 The change of the $\log_2(d_{avg})$ of the four TSP instances

We give a picture to show the change of the $\log_2(d_{avg})$ according to the computation cycles for the four TSP instances u2319, pr2392, pcb3038 and fnl4461, see Fig. 2. One sees the $\log_2(d_{avg})$ decreases according to $C$ in the computation process. For the nearly same scale of TSP, such as u2319 and pr2392, their $\log_2(d_{avg})$s are almost equal at the same computation cycle under the same assignments of the set of parameters. It means the number of edges with the frequency above (or below) a special frequency threshold $F$ is nearly equal for the same scale of TSP instances at the same computation cycle. Moreover, the bigger scale of TSP instances consumes more number of iterations and eliminates more edges to compute the final sparse graphs. One sees the $\log_2(d_{avg})$s of the four instances drops sharply at the first several computation cycles. Furthermore, the $\log_2(d_{avg})$s of the pcb3038 and fnl4461 decreases in a relatively flat manner. It means that the parameters in Table I are suitable for u2319 and pr2392 to compute their sparse graphs quickly. To accelerate the algorithm to compute the sparse graphs for the bigger scale of TSP instances, we should increase the parameters $F$, $a$ and $b$ in the computation process. At the end of the computation process, the $\log_2(d_{avg})$ does not change much and few edges are eliminated. In such case, the improved iterative algorithm converges to the minima. Once the preserved graph conforms to the terminal conditions, the algorithms will stop and output the residual graphs.

In the next experiments, we slightly change the preconditions to compute the sparse graphs for the fnl4461. If $F$ is below $4.95N$, we use the assigned parameters in the Table I. As $F$ is bigger $4.95N$, the values $a$=0.001 and $b$=80 are used to compute the residual graphs. The six experimental results are given in Table III. Due to the random small distances, the experimental results are different in two or more experiments. One sees the computation cycles are lengthened when we change the preconditions. After we eliminate more edges with the frequency below the bigger frequency threshold $4.95N$, less number of edges are eliminated at each of the following computation cycles as the frequency threshold increases slowly unless the frequency threshold becomes big. Since more number of computation cycles are consumed, the residual graphs contain less number of edges than those results in Table II. Moreover, the number of edges in the residual graphs decreases according to the computation cycles $C$. The bigger the computation cycle $C$ is, the smaller the number of edges in the residual graph will be. Although the residual graphs contain different number of edges, the number of the lost OHC edges is nearly equal. It says most of the OHC edges have the big frequency in the preserved graphs in the whole computation process so they are preserved.

TABLE III
THE EXPERIMENTAL RESULTS FOR THE FNL4461

| No. | $C$ | $K$ | $l$ | $d_{max}$ | $d_{avg}$ | $d_{min}$ |
|---|---|---|---|---|---|---|
| 1 | 684 | 16778 | 1 | 18 | 7.5 | 2 |
| 2 | 1448 | 15528 | 3 | 15 | 6.9 | 2 |
| 3 | 678 | 16771 | 2 | 18 | 7.5 | 2 |
| 4 | 889 | 16208 | 3 | 17 | 7.2 | 2 |
| 5 | 1499 | 15492 | 3 | 17 | 6.9 | 3 |
| 6 | 1209 | 15844 | 2 | 17 | 7.1 | 2 |

The experiments illustrates that we can use the frequency threshold $F \in [4.80N, 4.995N]$ to compute the sparse graphs for most of the TSP instances. In addition, keeping the shortest $n$ edges at each cycle is useful to maintain the OHC edges in the residual graphs. The parameters in Table I may not suitable for the larger scale of TSP instances. To reduce the computation

time, one can improve the parameters $F$, $a$ and $b$ in the computation process.

In the end, we compare the improved iterative algorithm with the combinatorial algorithm owing to Hougardy and Schroeder [14]. According to the 3-opt move, the combinatorial algorithm finds some useless edges that are impossibly included in the OHC. There are three stages in the combinatorial algorithm and the third stage is time-consuming. They run the combinatorial algorithm on the computer with a single core of a 2.9 GHz Intel Xeon. The comparisons of the experimental results for the same examples are shown in Table IV. It mentions that the worst residual graphs computed with the improved iterative algorithm are chosen from Table II for comparisons. The worst residual graph contains the largest number of edges. One sees the improved iterative algorithm compute the sparser graphs for these TSP instances except for vm1084. The improved iterative algorithm outperforms the combinatorial algorithm in computing the sparse graphs. For some examples, such as d1291, d1655, u1817 and rl1889, the number of edges computed with the improved iterative algorithm is much smaller than that computed by the combinatorial algorithm. The shortcoming of the improved iterative algorithm is that a few OHC edges may be eliminated if the frequency threshold $F$ is too big. One also sees the improved iterative algorithm consumes less time to compute the sparse graphs. The computation time of our algorithm increases according to the scale of TSP instances. However, the combinatorial algorithm usually consumes more computation time for the smaller TSP instances than that for some bigger TSP instances.

TABLE IV
COMPARISON OF THE IMPROVED ITERATIVE ALGORITHM WITH THE COMBINATORIAL ALGORITHM [14]

| TSP | $n$ | Improved iterative algorithm | | | Combinatorial algorithm | |
|---|---|---|---|---|---|---|
| | | $K$ | $l$ | $t$(h:m:s) | $K$ | $t$(h:m:s) |
| pr1002 | 1002 | 4180 | 1 | 0:1:14 | 4521 | 2:30:21 |
| u1060 | 1060 | 4479 | 1 | 0:1:38 | 4619 | 3:33:13 |
| vm1084 | 1084 | 5658 | 1 | 0:1:48 | 4610 | 1:21:05 |
| pcb1173 | 1173 | 5488 | 1 | 0:2:17 | 6084 | 3:10:51 |
| d1291 | 1291 | 5360 | 1 | 0:2:42 | 11317 | 14:25:40 |
| u1432 | 1432 | 5673 | 0 | 0:2:58 | 6495 | 2:15:55 |
| d1655 | 1655 | 9278 | 0 | 0:5:25 | 12103 | 10:43:26 |
| u1817 | 1817 | 7208 | 1 | 0:8:30 | 11736 | 6:30:25 |
| rl1889 | 1889 | 12922 | 1 | 0:11:23 | 18673 | 28:34:41 |
| d2103 | 2103 | 13632 | 3 | 0:17:13 | 18105 | 18:19:34 |
| u2152 | 2152 | 9970 | 1 | 0:13:39 | 13170 | 7:19:08 |
| u2319 | 2319 | 8221 | 1 | 0:14:49 | 9473 | 1:42:22 |
| pr2392 | 2392 | 11246 | 1 | 0:19:12 | 12088 | 7:44:55 |
| pcb3038 | 3038 | 13403 | 1 | 0:45:44 | 14869 | 5:55:22 |
| fnl4461 | 4461 | 18685 | 1 | 3:40:18 | 19082 | 7:14:21 |

## IV. CONCLUSION

This research presents the improved iterative algorithm for reducing the TSP on the complete graph $K_n$ to the TSP on the sparse graphs. The computation time of the algorithm is $O(CN_{\max}n^2)$ which is acceptable for big scale of TSP instances. The experimental results showed that the improved algorithm is robust to compute the sparse graphs with less than $n\log_2 n$ edges containing nearly all of the OHC edges.

## APPENDIX

### A. The Probability $p_{3,5}(e)$ of the OHC edges

In $K_n$, an edge $e$ is contained in $\binom{n-2}{2}$ quadrilaterals. Each quadrilateral corresponds to one of the six frequency quadrilaterals ABCD [16]. $p_1(e), p_3(e)$ and $p_5(e)$ represent the probability that an edge $e$ has the frequency 1, frequency 3 and frequency 5 in a frequency quadrilateral containing $e$. $p_1(e) = p_3(e) = p_5(e) = \frac{1}{3}$ can be derived based on the six frequency quadrilaterals. When we choose $N$ frequency quadrilaterals containing an edge $e$ to compute its total frequency $F(e)$, the total frequency of $e$ is $F(e) = 3N$ according to the probability $p_1(e) = p_3(e) = p_5(e) = \frac{1}{3}$. This is the average frequency of all of edges. For the OHC edges $e$, Wang and Remmel constructed some special frequency quadrilaterals where the frequency of $e$ is 5 or 3 rather than 1. They gave the probability model (2) for the OHC edges where they combined the $p_3(e) = p_5(e) = \frac{1}{3} + \frac{2}{3(n-2)}$ together as $p_{3,5}(e) = \frac{2}{3} + \frac{2}{3(n-2)}$.

$$\begin{cases} p_1(e) = \frac{1}{3} - \frac{2}{3(n-2)} \\ p_{3,5}(e) = \frac{2}{3} + \frac{2}{3(n-2)} \end{cases} \quad (2)$$

When we use $N$ frequency quadrilaterals containing an OHC edge $e$ to compute its total frequency, the total frequency of $e$ becomes $F(e) = \left(3 + \frac{2}{n-2}\right)N$ which is bigger than the average frequency $3N$ of all of edges. The experiments for the TSP instances showed that the probability model (2) was too conservative for the OHC edges. The minimum frequency $F(e)$ of the OHC edge for most TSP instances is much bigger than $\left(3 + \frac{2}{n-2}\right)N$. If we use $\left(3 + \frac{2}{n-2}\right)N$ as the frequency threshold to trim the edges with frequency $F(e) < \left(3 + \frac{2}{n-2}\right)N$, we generally compute a subgraph with less than $\frac{1}{2}\binom{n}{2}$ edges.

According to the probability model (1), we assume $p_{3,5}(e) = \frac{2}{3} + \epsilon$ for the OHC edges where $\epsilon > 0$. In the following, we give a theorem to predict the change of $p_{3,5}(e)$ or $\epsilon$ for the OHC edges according to $n$.

**Theorem 1.** If $p_{3,5}(e) = \frac{2}{3} + \epsilon$ is the minimum probability that the OHC edge e has the frequency 3 or frequency 5 in a frequency quadrilateral in $K_n$, the $p_{3,5}(e) = \frac{2}{3} + \epsilon$ tends to 1 as n is big enough.

Some notations are given first for proof:

- $K_n$: The complete graph containing $n$ vertices.
- $K_{n+1}$: The complete graph containing $n$+1 vertices.
- $p_{3,5}^n = \frac{2}{3} + \epsilon_n$: The minimum probability that the OHC edge $e$ has the frequency $f$=3 or $f$=5 in a frequency quadrilateral in $K_n$ where $\epsilon_n \to 0$.
- $p_{3,5}^{n+1} = \frac{2}{3} + \epsilon_{n+1}$: The minimum probability that the OHC edge $e$ has the frequency $f$=3 or $f$=5 in a frequency

quadrilateral in $K_{n+1}$ where $\epsilon_{n+1} \to 0$.

The minimum $p_{3,5}^n$ and $p_{3,5}^{n+1}$ exists in the graph sets $K_n$ and $K_{n+1}$ according to the axiom of choice owing to Ernst Zermelo.

- $M_n = \left[p_{3,5}^n \binom{n-2}{2}\right]$: The minimum number of the frequency quadrilaterals where *e* has the frequency *f*=3 or *f*=5 in $K_n$.
- $M_{n+1} = \left[p_{3,5}^{n+1} \binom{n-1}{2}\right]$ : The minimum number of the frequency quadrilaterals where *e* has the frequency *f*=3 or *f*=5 in $K_{n+1}$.

Since $\epsilon_n \to 0$ and $\epsilon_{n+1} \to 0$ , we have the inequality $M_n \le M_{n+1}$. Thus, the inequality (3) is derived.

$$\frac{p_{3,5}^{n+1}}{p_{3,5}^n} \ge \frac{\binom{n-2}{2}}{\binom{n-1}{2}} = 1 - \frac{2}{n-1} \tag{3}$$

Given a big number *M*, we assume $\lim_{n\to M} \frac{2}{n-1} = 0$. If $n > M$ , $p_{3,5}^{n+1} \ge p_{3,5}^n$ and $\epsilon_{n+1} \ge \epsilon_n$ hold. It means the minimum probability $p_{3,5}(e) = \frac{2}{3} + \epsilon$ of the OHC edge *e* increases according to *n* until it reaches the maximum value 1. In this case, the $\epsilon$ approaches $\frac{1}{3}$. According to the similar proof process, we can prove the probability $p_5(e) \to 1$ for the OHC edges *e* as *n* is big enough under the assumption of $p_5(e) = \frac{1}{3} + \epsilon$ and $\epsilon > 0$. When we choose *N* frequency quadrilaterals containing *e* to compute the *F*(*e*), the $F(e)$ tends to 5*N* as *n* is sufficiently large.

*B. The Number of Edges in the Subgraphs*

We assume there are *K* edges *e* with the probability $p_{3,5}(e) \ge \frac{2}{3} + \epsilon$ in a given frequency quadrilateral in $K_n$. An edge *e* has the probability $p_1(e) = p_3(e) = p_5(e) = \frac{1}{3}$ in a given frequency quadrilateral containing *e*. The probability that *e* has the frequency *f*=3 or *f*=5 is $p_{3,5}(e) = \frac{2}{3}$ in a frequency quadrilateral. If we choose *N* frequency quadrilaterals containing *e*, the probability that there are $\left[\left(\frac{2}{3} + \epsilon\right)N\right]$ frequency quadrilaterals where *e* has the frequency *f*=3 or *f*=5 is computed as $\binom{N}{\left(\frac{2}{3}+\epsilon\right)N}\left(\frac{2}{3}\right)^{\left(\frac{2}{3}+\epsilon\right)N}\left(\frac{1}{3}\right)^{\left(\frac{1}{3}-\epsilon\right)N}$. It is a value of the binomial distribution $P(X = i) = \binom{N}{i}\left(\frac{2}{3}\right)^i\left(\frac{1}{3}\right)^{N-i}$ where *X* is the random variable representing the number of frequency quadrilaterals where *e* has the frequency *f*=3 or *f*=5. The edges with the probability $p_{3,5}(e) \ge \frac{2}{3} + \epsilon$ will be preserved as the candidate OHC edges. In this case, the number $i \ge \left[\left(\frac{2}{3} + \epsilon\right)N\right]$. The accumulative probability that $i \ge \left[\left(\frac{2}{3} + \epsilon\right)N\right]$ is $\left(X \ge \left[\left(\frac{2}{3} + \epsilon\right)N\right]\right) = \sum_{i=\left[\left(\frac{2}{3}+\epsilon\right)N\right]}^{N}\binom{N}{i}\left(\frac{2}{3}\right)^i\left(\frac{1}{3}\right)^{N-i}$. Considering the $\binom{n}{2}$ edges, the number of the edges *K* in the subgraph will be (4) if we use $\frac{2}{3} + \epsilon$ as the probability threshold.

$$K = \binom{n}{2}\sum_{i=\left[\left(\frac{2}{3}+\epsilon\right)N\right]}^{N}\binom{N}{i}\left(\frac{2}{3}\right)^i\left(\frac{1}{3}\right)^{N-i} \tag{4}$$

The binomial distribution $P(X = i) = \binom{N}{i}\left(\frac{2}{3}\right)^i\left(\frac{1}{3}\right)^{N-i}$ has the maximum probability at $i = \left\lfloor\frac{2}{3}(N+1)\right\rfloor - 1$ or $i = \left\lfloor\frac{2}{3}(N+1)\right\rfloor$. After that, it decreases exponentially from $i = \left\lfloor\frac{2}{3}(N+1)\right\rfloor$ or $i = \left\lfloor\frac{2}{3}(N+1)\right\rfloor + 1$ to *N*. Once $\epsilon > \frac{1}{N}$, $\left[\left(\frac{2}{3} + \epsilon\right)N\right] > \left\lfloor\frac{2}{3}(N+1)\right\rfloor$ or $\left[\left(\frac{2}{3} + \epsilon\right)N\right] > \left\lfloor\frac{2}{3}(N+1)\right\rfloor + 1$ , the value $\binom{N}{\left(\frac{2}{3}+\epsilon\right)N}\left(\frac{2}{3}\right)^{\left(\frac{2}{3}+\epsilon\right)N}\left(\frac{1}{3}\right)^{\left(\frac{1}{3}-\epsilon\right)N}$ deviates from the maximum probability and tens to zero quickly. Even though $\epsilon$ takes the conservative value $\frac{2}{3(n-2)}$ as in (1), the number *K* will be very small when $N \ge nlogn$. In theory, we can compute the subgraphs with a small number of edges for TSP once the OHC edges have the big probability $p_{3,5}(e) \ge \frac{2}{3} + \epsilon$ in $K_n$.

The number *K* depends on the parameter $\epsilon$. The bigger the value $\epsilon$ is, the smaller the number of edges *K* in the subgraph will be. For the edges *e* with the small probability $p_{3,5}(e) < \frac{2}{3} + \epsilon$ (or $p_5(e) < \frac{1}{3} + \epsilon$), we usually have less than $\left[\left(\frac{2}{3} + \epsilon\right)N\right]$ (or $\left[\left(\frac{1}{3} + \epsilon\right)N\right]$) frequency quadrilaterals where their frequency *f*=3 or *f*=5 (or only 5) among the *N* random frequency quadrilaterals containing each of them. These edges will be trimmed according to the probability threshold probability $\frac{2}{3} + \epsilon$. For the other edges with the big probability $p_{3,5}(e) \ge \frac{2}{3} + \epsilon$ (or $p_5(e) \ge \frac{1}{3} + \epsilon$ ), they usually have more than $\left[\left(\frac{2}{3} + \epsilon\right)N\right]$ (or $\left[\left(\frac{1}{3} + \epsilon\right)N\right]$) frequency quadrilaterals where their frequency *f*=3 or *f*=5 (or only 5). However, the number of these edges is very small according to (4). In theory, most edges will have less than $\left[\frac{2}{3}N\right]$ (or $\left[\frac{1}{3}N\right]$) frequency quadrilaterals where their frequency *f*=3 or *f*=5 (or only 5) among the *N* random frequency quadrilaterals containing each of them. These edges will be neglected with a big probability. Thus, we will preserve a small number of edges with the big probability $p_{3,5}(e) > \frac{2}{3} + \epsilon$ (or $p_5(e) > \frac{1}{3} + \epsilon$) in the subgraphs.

For the OHC edges, their probability $p_{3,5}(e) = \frac{2}{3} + \epsilon$ tends to 1 as *n* is big enough. We can use the big probability threshold $\frac{2}{3} + \epsilon$ to eliminate a lot of edges with the probability $p_{3,5}(e)$ below $\frac{2}{3} + \epsilon$ so a sparse graph is computed for TSP.

## ACKNOWLEDGMENT

The authors acknowledge Professor Jeffrey B. Remmel for the useful discussions on the frequency quadrilaterals. The authors also thank the support of the National Key Laboratory of New Energy Power System and the Beijing Key Laboratory of New and Renewable Energy, North China Electric Power University, Beijing, China.

## REFERENCES

[1] R. M. Karp, "On the computational complexity of combinatorial problems," *Networks*(USA), vol.5, no.1, pp. 45-68, 1975.

[2] M. Held, R. M. Karp, "A dynamic programming approach to sequencing problems," *Journal of the Society for Industrial and Applied Mathematics*, vol.10, no.1, pp.196–210, 1962.

[3] R. Bellman, "Dynamic programming treatment of the travelling salesman problem," *Journal of the ACM*, vol.9, no.1, pp.61–63, 1962.
[4] A. Björklund, "Determinant sums for undirected Hamiltonicity," in *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, Las Vegas 2010, pp.173-182.
[5] D. L. Applegate, R. E. Bixby, V. Chvátal, W. Cook, D. G. Espinoza, M. Goycoolea, and K. Helsgaun, "Certification of an optimal TSP tour through 85900 cities," *Operations Research Letters*, vol.37, no.1, pp.11–15, 2009.
[6] G. Gutin, A. P. Punnen, *The traveling salesman problem* and *its variations*. New York: Springer-Verlag, 2007.
[7] S. Arora, "Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems," *Journal of the ACM*, vol.45, no.5, pp.753–782, 1998.
[8] T. Moemke, O. Svensson, "Approximating graphic TSP by matchings," in *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, Palm Springs, CA, USA, 2011, pp. 560–569.
[9] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto, "The traveling salesman problem in bounded degree graphs," *ACM Transactions on Algorithms*, vol.8, no.2, Article No.18, 2012.
[10] M. Xiao, H. Nagamochi, "An Exact Algorithm for TSP in Degree-3 Graphs Via Circuit Procedure and Amortization on Connectivity Structure," *Algorithmica*, vol.74, no.2, pp.713–741, 2016.
[11] J. R. Correa, O. Larre, and J. A. Soto, "TSP tours in cubic graphs:beyond 4/3," *SIAM Journal on Discrete Mathematics*, vol.29, no.2, pp.915–939, 2015.
[12] G. Borradaile, E. D. Demaine, and S. Tazari, "Polynomial-time approximation schemes for subset-connectivity problems in bounded-genus graphs," *Algorithmica*, vol.68, no.2, pp.287–311, 2014.
[13] S. O. Gharan, A. Saberi, "The asymmetric traveling salesman problem on graphs with bounded genus," in *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete algorithms*, San Francisco, CA, USA, 2011, pp.967-975.
[14] S. Hougardy, R. S. Schroeder, "Edge elimination in TSP instances," in *International Workshop on Graph-Theoretic Concepts in Computer Science*, Nouan-le-Fuzelier, France, 2014, pp.275-286.
[15] Y. Wang, "An approximate method to compute a sparse graph for traveling salesman problem," *Expert Systems with Applications*, vol.42, no.12, pp.5150–5162, 2015.
[16] Y. Wang, J. B. Remmel, "A binomial distribution model for the traveling salesman problem based on frequency quadrilaterals," *Journal of Graph Algorithms & Applications*, vol.20, no.2, pp.411–434, 2016.
[17] K. Helsgaun, "An effective implementation of the Lin–Kernighan traveling salesman heuristic," *European Journal of Operational Research*, vol. 126 no.1, pp.106-130, 2000.
[18] G. Reinelt, http://comopt.ifi.uni-heidelberg.de/groups/comopt /software/TSPLIB95/, 2017.
[19] H. Mittelmann, http://neos-server.org/neos/solvers/co:concorde/ TSP.html, 2016.

**Y. Wang** was born in Henan province, China on February 28, 1978. Next, the author's educational background is listed. He got the bachelor degree on Aircraft Manufacturing at Nanjing University of Aeronautics and Astronautics, Nanjing, China in 2001. He got the master degree and PhD degree on Computer Aided Manufacturing at the Beihang University, Beijing, China in 2002 and 2009, respectively.

He now works at the Renewable Energy School of North China Electric Power University, Beijing, China. From 2009 to 2013, he was a lecturer and taught the discrete manufacturing. He has got the Associative Professor position since 2014. He was awarded the Teaching Prize of Beijing Education Committee in 2013. He published more than 20 papers in his research area. His research interests are broad which mainly include modeling and methods for combinatorial optimization problems, algorithms, heuristics and their real-world applications.