A Heuristic Algorithm to Compute a Subgraph for TSP Based on Frequency Quadrilaterals

Qirong Deng¹, Yong Wang¹²,∗ and Liping Chen²

¹School of New Energy, North China Electric Power University, Beijing, China
²School of Information Engineering, Tarim University, Alar, Xinjiang, China

*Corresponding author email: yongwang@ncepu.edu.cn

Abstract. A heuristic algorithm was presented to compute certain subgraphs for symmetric traveling salesman problem using frequency quadrilaterals. If we choose a set of frequency quadrilaterals containing an edge to compute the frequency for it, the frequency of an edge contained within the optimal Hamiltonian cycle approaches an extreme value as the city number is big enough. A heuristic algorithm was designed to compute a frequency graph with randomly selected frequency quadrilaterals. A subgraph containing the optimal solution was computed according to a given frequency threshold. The experimental results illustrated that in certain instances for Euclidean TSP with scale above 1000, the number of edges in complete graph is reduced by more than 30 times. Moreover, the bigger the number of frequency quadrilaterals is used, the higher the frequency threshold is and the smaller the number of edges in the subgraphs is kept.

Keywords: Traveling salesman problem; Frequency graph; Frequency Quadrilateral; Heuristic algorithm; Subgraph.

1. Introduction

TSP is a well-known NP-hard problem in combinatorial optimization. The problem can be described as given a set of n vertices \( \{v_1, v_2, ..., v_n\} \) in complete graph \( K_n \), there is a distance \( d(u, v) > 0 \) for two vertices \( u, v \in \{v_1, v_2, ..., v_n\} \) and \( u \neq v \). For symmetric TSP, \( d(u, v) = d(v, u) \). The goal of TSP is to find an optimal Hamiltonian cycle (OHC) which is a permutation \( \sigma = (v_1', v_2', ..., v_n') \) of the \( n \) vertices \( v'_i \in \{v_1, v_2, ..., v_n\} \) \((1 \leq i \leq n)\) make the distance \( d(\sigma) = d(v'_1, v'_n) + \sum_{i=1}^{n-1} d(v'_i, v'_{i+1}) \) is the minimum. In real-world applications, the vehicle routing problem, machine scheduling, large scale integration design, manufacturing path optimization, etc., can be transformed into various TSP. Therefore, TSP is widely studied in OR and computer science in order to find the efficient methods[1]. Data mining aims to find the useful and significant knowledge from data sets[2]. Many structural and semi-structural datum are usually represented as graphs. In recent years, subgraph mining becomes popular as a branch of data mining for cultivating useful subgraphs from set of graphs or a given large graph[3]. The frequent subgraph mining is concerned by researchers who regard the frequent subgraphs occurring above a frequency threshold in a given graph set[4]. The subgraphs usually have special properties which benefit the science research and engineering applications. Topological relationships between vertices are broadly considered to mine useful frequent graphs.

This research focuses on computing a subgraph for TSP on \( K_n \). That is, given a complete graph \( K_n \) of an instance for TSP, we will compute a subgraph with a small number of edges for it. This topic is interesting because TSP on \( K_n \) and subgraphs has different complexities. In 1962, Held and Karp[5], independently with Bellman[6] designed the dynamic programming algorithm to resolve TSP in
According to the binomial distribution, the number of edges with higher frequency above the frequency threshold is estimated in this paper. Furthermore, the selection of frequency quadrilaterals is improved.

The other parts of the paper are arranged as follows. In section 2, the frequency graph for TSP is introduced briefly. In section 3, we shall review frequency quadrilaterals computed by optimal 4-vertex paths and a probability model established with frequency quadrilaterals. In the preserved subgraphs, the possible number of edges is estimated in section 4. In section 5, the heuristic algorithm is designed to compute subgraphs from the constructed subgraphs. If the extracted subgraphs have some special properties, the concerned subgraphs also have the special properties. Furthermore, the subgraphs with a big frequency will intensively show the special properties of the extracted subgraphs. For example, we extract \( \binom{n}{2} \) i-vertex complete subgraphs \( K_i \) from \( K_n \) and compute the minimum spanning tree for each of the subgraphs. After we enumerate the frequency of edges from the \( \binom{n}{2} \) minimum spanning subtrees, it is obvious the edges in minimum spanning tree of \( K_n \) have the maximum frequency.

2. Frequency Graph for TSP

We can extract all kinds of subgraphs from \( K_n \). Moreover, we can count the frequency of the concerned subgraphs from the extracted subgraphs. If the extracted subgraphs have some special properties, the concerned subgraphs also have the special properties. Furthermore, the subgraphs with a big frequency will intensively show the special properties of the extracted subgraphs. For example, we extract \( \binom{n}{2} \) i-vertex complete subgraphs \( K_i \) from \( K_n \) and compute the minimum spanning tree for each of the subgraphs. After we enumerate the frequency of edges from the \( \binom{n}{2} \) minimum spanning subtrees, it is obvious the edges in minimum spanning tree of \( K_n \) have the maximum frequency.
OHC is one kind of spanning subgraphs of $K_n$. As we compute the high frequency of the subgraphs in OHC, we must rely on the subgraphs close to the subgraphs in OHC. Two close or similar subgraphs will have the following characters: (1) They have the similar topological structure. (2) They have the similar or same special properties. We analyse the characters of the subgraphs in OHC: (1) They are $k$-vertex paths where $k \geq 2$, and (2) each $k$-vertex path is the shortest whatever the intermediate vertices are exchanged. For the first character, we may find the $k$-vertex paths in $K_n$ as the initial subgraphs. To meet the second property, we will compute a kind of special $k$-vertex paths for $k$ vertices with two given endpoints and the distance is the minimum. The basic idea to compute the special $k$-vertex paths close to the subgraphs in OHC is given as follows.

Given a set $\{v_1, v_2, ..., v_k\}$ of $k$ vertices in $K_n$ where $k \geq 4$, let arrange $v = (v'_1, v'_2, ..., v'_k)$ of the $k$ vertices with $v'_1 = v_1$ and $v'_k = v_k$. The distance of $v$ is computed as $d(v) = \sum_{i=1}^{k-1} d(v'_i, v'_{i+1})$. We assume all the paths have different distances. There is one path $\nu$ whose distance $d(\nu)$ is the minimum.

We call this path the optimal $k$-vertex path for end vertices $v_1, v_k$. There are $\binom{k}{2}$ ways to select the end vertices from $\{v_1, v_2, ..., v_k\}$ so that there are $\binom{k}{2}$ optimal $k$-vertex paths.

Now, we find the optimal $k$-vertex paths which have the second property as that of the paths in OHC. The next work is to compute the optimal $k$-vertex paths and enumerate the frequency of the $i$-vertex paths where $i < k$. Since the optimal $k$-vertex paths have the same properties as the paths in OHC, the $i$-vertex paths in OHC will have high frequency after each of them is enumerated from the optimal $k$-vertex paths.

In $K_n$, there are $\binom{n}{k}$ $k$-vertex sets. Thus, there are total $\binom{k}{2}\binom{n}{k}$ optimal $k$-vertex paths. As $n$ is fixed, the number of optimal $k$-vertex paths changes as a combinatorial number according to $k$. Wang has computed the total number of optimal $k$-vertex paths in $K_n$ as $\binom{n}{2}2^{n-2}$ [14]. But it is still difficult to find an OHC through traversing all of the optimal $k$-vertex paths.

One possible method is to compute the frequency of optimal $i$-vertex paths with optimal $k$-vertex paths where $i < k$. The optimal $i$-vertex paths with big frequency are chosen as the candidate paths for OHC. The values of $k$ and $i$ should be determined before computing. The time complexity to compute the optimal $k$-vertex paths is $O(k^n)$. In general, the bigger the $k$ is, the closer the optimal $k$-vertex paths approach the paths in OHC. We will obtain the outstanding frequency for the $i$-vertex paths in OHC according to the optimal $k$-vertex paths where $k$ is big. As well as, it will spend much time to compute such optimal $k$-vertex paths. In theory, Section 3 shows the $i$-vertex paths in OHC will have the biggest frequency as $n$ is large enough and $k$ is larger than 3. In order to reduce the computation time, we compute the optimal 4-vertex paths and count the frequency of the optimal $i$-vertex paths ($i < 4$). For small or medium scale TSP instances, the frequency of all OHC $i$-vertex paths will not be outstanding. The merits are also attractive since we will save much computation time. Moreover, the frequency of most $i$-vertex paths in OHC will be very high. The optimal 4-vertex paths are composed of 3-edge or 2-edge paths ($i=3$ and 2). For convenience, we enumerate the frequency of edges from the optimal 4-vertex paths.

Given a set of optimal 4-vertex paths, the frequency of each edge is counted. The edges and their frequencies from a frequency graph. In such frequency graphs, the frequency of the OHC edges is generally much bigger than that of most of the other edges. In a previous paper [14], we computed the frequency graphs with optimal 4-vertex paths and filter out many useless edges according to a given frequency threshold. The time complexity of the previous algorithm is $O(n^4)$ which is difficult to apply to large scale TSP. In this paper, we compute the frequency graphs with randomly selected frequency quadrilaterals. Furthermore, we design a heuristic algorithm which needs $O(Nn^2)$ time to obtain a subgraph having less edges for TSP, where $N$ is much smaller than $n$.  

3
3. The Probability Model Based on Frequency Quadrilaterals

3.1. Frequency Quadrilaterals

Paper [13] have introduced the frequency quadrilaterals. Given four vertices \( \{A, B, C, D\} \) in \( K_n \) which shall form a quadrilateral ABCD. ABCD includes six optimal 4-vertex paths. Due to the simplicity of quadrilateral, the optimal 4-vertex paths can be computed with a four-vertex and three-line inequality. Then, the corresponding frequency quadrilateral is computed based on the six optimal 4-vertex paths. The following gave the method to compute a frequency quadrilateral.

A weighted quadrilateral ABCD is shown in Figure 1. The distances of edges \( d_{AB}, d_{AC}, d_{AD}, d_{BC}, d_{BD} \) and \( d_{CD} \) are noted aside the edges where \( d_{AB} = d(A, B), d_{AC} = d(A, C), d_{AD} = d(A, D), d_{BC} = d(B, C), d_{BD} = d(B, D) \) and \( d_{CD} = d(C, D) \).

\[
\begin{array}{|c|c|}
\hline
\text{No.} & \text{Inequality array} & \text{Six optimal 4-vertex paths} \\
\hline
(1) & d_{AB} + d_{CD} < d_{AD} + d_{BD} < d_{AC} + d_{BC} & (A,D,C,B), (A,B,D,C), (A,B,C,D), (B,A,D,C), (B,A,C,D) \\
(2) & d_{AB} + d_{CD} < d_{AC} + d_{BD} < d_{AD} + d_{BC} & (A,C,D,B), (A,B,D,C), (A,B,C,D), (B,A,D,C), (B,A,C,D) \\
(3) & d_{AD} + d_{BC} < d_{AB} + d_{CD} < d_{AC} + d_{BD} & (A,D,C,B), (A,D,B,C), (A,B,C,D), (B,A,D,C), (B,A,C,D) \\
(4) & d_{AD} + d_{BC} < d_{AC} + d_{BD} < d_{AB} + d_{CD} & (A,D,C,B), (A,D,B,C), (A,B,C,D), (B,D,A,C), (B,C,A,D), (C,B,A,D) \\
(5) & d_{AC} + d_{BD} < d_{AB} + d_{CD} < d_{AD} + d_{BC} & (A,C,D,B), (A,B,D,C), (A,C,B,D), (B,D,A,C), (B,A,D,C), (C,A,B,D) \\
(6) & d_{AC} + d_{BD} < d_{AD} + d_{BC} < d_{AB} + d_{CD} & (A,C,D,B), (A,D,B,C), (A,C,B,D), (B,D,A,C), (B,A,C,D), (C,A,B,D) \\
\hline
\end{array}
\]

Figure 1. A weighted quadrilateral ABCD.

It is the three sum distances related to the three pairs of opposite edges \( d_{AB} + d_{CD}, d_{AC} + d_{BD} \) and \( d_{AD} + d_{BC} \) that determine the six optimal 4-vertex paths in ABCD. We assume the three sum distances \( d_{AB} + d_{CD}, d_{AC} + d_{BD} \) and \( d_{AD} + d_{BC} \) are unequal to compute the six optimal 4-vertex paths. Since there are three sum distances, they naturally have six different permutations and each permutation constitutes an inequality array. The six inequality arrays are given in Table 1. Each independent inequality array is used to compute six optimal 4-vertex paths in ABCD. Then, the six optimal 4-vertex paths are used to compute the corresponding frequency quadrilateral. The six different frequency quadrilaterals are shown in Figure 2. Given an arbitrary weighted quadrilateral ABCD in \( K_n \), the corresponding frequency quadrilateral will be one of the frequency quadrilaterals in Figure 2.

Let’s see the frequency of edges in the quadrilaterals. For an edge \( e \in \{AB, AC, AD, BC, BD, CD\} \), the frequency \( f \) of \( e \) is 5, 3 and 1, respectively. In addition, each of the frequencies 5, 3 and 1 occurs twice in the six frequency quadrilaterals, respectively. For \( i \in \{1,3,5\} \), \( p_i(e) \) is the probability refers to \( e \) contained in a frequency quadrilateral has frequency \( i \). Thus, the probability \( p_1(e) = p_3(e) = p_5(e) = \frac{1}{3} \) according to the six frequency quadrilaterals. In \( K_n \), each ABCD has the equal probability to be one of the six frequency quadrilaterals in Figure 2. Given \( N \) frequency quadrilaterals containing \( e \), there will be \( \left[ \frac{N}{3} \right] \) frequency quadrilaterals in which \( e \) has frequency \( f = 1, 3 \) and 5, respectively.

Table 1. The six inequality arrays and the corresponding optimal paths.
3.2. A theorem Based on Frequency Quadrilaterals

We use the frequency quadrilaterals instead of optimal 4-vertex paths to compute the frequency of edges. Given \( N \) frequency quadrilaterals containing edge \( e \), the frequency of \( e \) is computed as \( \mathcal{F}(e) = \sum_{i=1}^{N} f_i \), where \( f_i \) is the frequency of \( e \) in the \( i \)th frequency quadrilateral. Since frequency quadrilaterals are computed based on optimal 4-vertex paths, the frequency of edges computed by frequency quadrilaterals has the same performance as that computed by the optimal 4-vertex paths. Differently, Wang and Remmel can estimate \( \mathcal{F}(e) \) of the OHC edges with a binomial distribution model based on frequency quadrilaterals\(^{[13]}\).

In \( K_n \), each edge \( e \) is contained in \( \binom{n-2}{2} \) quadrilaterals so that there are the same number of frequency quadrilaterals containing \( e \). When they choose \( N \) frequency quadrilaterals with edge \( e \) to compute \( \mathcal{F}(e) \), \( \mathcal{F}(e) = 3N \) according to the probability \( p_4(e) = p_5(e) = \frac{2}{3} \) which is the average frequency for all of edges. For an OHC edge \( e \), Wang and Remmel found some special frequency quadrilaterals where the frequency of \( e \) is 5 or 3 rather than 1\(^{[13]}\). They gave the probability model (1) for the edges in OHC, where they combined \( p_5(e) \) and \( p_5(e) \) together as \( p_{3,5}(e) = \frac{2}{3} + \frac{2}{3(n-2)} \).

\[
\begin{align*}
\begin{cases}
    p_4(e) = \frac{1}{3} - \frac{2}{3(n-2)} \\
p_{3,5}(e) = \frac{2}{3} + \frac{2}{3(n-2)}
\end{cases}
\end{align*}
\]  

(1)

When they use \( N \) frequency quadrilaterals containing an OHC edge \( e \) to compute, its frequency \( \mathcal{F}(e) = (3 + \frac{2}{n-2})N \) which is bigger than \( 3N \). We did experiments for certain TSP instances. It illustrated that the probability model (1) was too conservative for the OHC edges. For the OHC edges, the minimum frequency is much bigger than \( (3 + \frac{2}{n-2})N \) for almost all TSP instances. If we use \( (3 + \frac{2}{n-2})N \) as the frequency threshold to trim the edges with a frequency \( \mathcal{F}(e) < (3 + \frac{2}{n-2})N \), we will compute a subgraph with less than \( \frac{1}{2} \binom{n}{2} \) edges. To compute a subgraph as sparse as possible, we expect the probability \( p_{3,5}(e) \) for OHC edges is as big as possible so that we can use a big frequency threshold \( F \) to eliminate more edges out of OHC.

According to the probability model (1), we know \( p_{3,5}(e) \gt \frac{2}{3} \) for an OHC edge. Thus, we assume \( p_{3,5}(e) = \frac{2}{3} + \epsilon \) for an OHC edge where \( \epsilon \gt 0 \). We are interested in the change of \( p_{3,5}(e) = \frac{2}{3} + \epsilon \) according to the scale of TSP. In paper \(^{[15]}\), Wang proven the following theorem for cultivating \( p_{3,5}(e) \) for an OHC edge according to \( n \).
Theorem 1: If \( p_{3,5}(e) = \frac{2}{3} + \epsilon \) is the minimum probability of an OHC edge in a frequency quadrilateral in \( K_n \), \( p_{3,5}(e) = \frac{2}{3} + \epsilon \) tends to 1 as \( n \) is big enough.

For an OHC edge, the minimum probability \( p_{3,5}(e) = \frac{2}{3} + \epsilon \) will increase according to \( n \) before it reaches the maximum value 1. In this case, we shall assume \( e \) has the probability \( p_3(e) = p_5(e) = \frac{1}{3} \) in each of the \( N \) frequency quadrilaterals. The minimum frequency of an OHC edge computed based on \( N \) frequency quadrilaterals will be equal to \( 4N \). According to the similar proof process, we can prove \( p_5(e) \rightarrow 1 \) for an OHC edge as \( n \) is big enough under the assumption of \( p_5(e) = \frac{1}{3} + \epsilon \) and \( \epsilon > 0 \). In the ultimate case, the frequency \( F(e) \) of each OHC edge will increase to \( 5N \) as \( n \) is sufficiently large.

For TSP instances with large scale, we can improve the frequency threshold to eliminate more edges which with a lower frequency.

If every quadrilateral includes exactly six optimal 4-vertex paths, edges in OHC will have the probability which with a lower frequency.

For TSP instances with large scale, we can improve the frequency threshold to eliminate more edges which with a lower frequency.

The next questions are how big the frequency threshold is and how many edges the subgraph contains if we use a given frequency threshold for eliminating the edges with small frequency. In practice, the scale of most TSP instances is limitative. \( p_{3,5}(e) \) and \( p_5(e) \) of some OHC edges are less than 1. In this case, \( F \) close to \( 5N \) will be too big to be the frequency threshold.

\( F \) can be estimated according to the probability \( p_{3,5}(e) = \frac{2}{3} + \epsilon \) and \( p_3(e) = \frac{1}{3} - \epsilon \). We assume \( p_3(e) = p_5(e) = \frac{1}{3} + \frac{\epsilon}{2} \) so the frequency threshold is equal to \( F = 3(1 + \epsilon)N \). \( F \) depends on the parameter \( \epsilon \). The bigger the scale \( n \) is, the bigger the \( \epsilon \) is as well as the frequency threshold \( F \). When \( \epsilon \) is small, there will be some other edges whose probability \( p_{3,5}(e) \) is equal to or above \( \frac{2}{3} + \epsilon \). We assume there are \( K \) edges with probability \( p_{3,5}(e) \geq \frac{2}{3} + \epsilon \) in \( K_n \). On the average, an edge \( e \) has the probability \( p_3(e) = p_5(e) = \frac{1}{3} \) in a given frequency quadrilateral containing \( e \). Further, when 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 frequency 3 or 5 is computed as \( \left( \frac{N}{3} \right)^i \left( \frac{2}{3} + \epsilon \right)^N \left( \frac{1}{3} \right)^{N-i} \). It is a value of the binomial distribution \( P(X = i) = \binom{N}{i} \left( \frac{2}{3} + \epsilon \right)^i \left( \frac{1}{3} \right)^{N-i} \) where \( X \) is the random variable denoting the number of frequency quadrilaterals where frequency of \( e \) is 3 or 5. We will reserve the edges with a probability \( p_{3,5}(e) \geq \frac{2}{3} + \epsilon \). The accumulative probability is \( \Pr \left( X \geq \left[ \left( \frac{2}{3} + \epsilon \right)N \right] \right) = \sum_{i=\left[ \left( \frac{2}{3} + \epsilon \right)N \right]}^{N} \binom{N}{i} \left( \frac{2}{3} \right)^i \left( \frac{1}{3} \right)^{N-i} \). Considering the \( \binom{N}{2} \) edges, the number \( K \) of preserved edges will be the formula (2) as \( F = 3(1 + \epsilon)N \) is used as the frequency threshold.

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

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 \( i = \frac{2}{3}(N + 1) - 1 \) or \( i = \frac{2}{3}(N + 1) \). After that, it decreases exponentially from \( i = \frac{2}{3}(N + 1) \) or \( i = \frac{2}{3}(N + 1) + 1 \)
to $N$. Once $\varepsilon > \frac{1}{N}$, $\left(\frac{2}{3} + \varepsilon\right)N > \frac{2}{3}(N + 1)$ or $\left(\frac{2}{3} + \varepsilon\right)N > \frac{2}{3}(N + 1) + 1$, the value

$\left(\frac{2}{3} + \varepsilon\right)N \left(\frac{2}{3} + \varepsilon\right)N \left(\frac{1}{3} + \varepsilon\right)N \left(\frac{1}{3} + \varepsilon\right)N$ deviates from the maximum probability and tends to zero quickly.

Even though $\varepsilon$ takes the conservative value $\frac{2}{3(n-2)}$ in formula (1), $K$ will be very small when $N = n\log n$ or a bigger value. In theory, we can compute the subgraphs with a small number of edges for TSP if the OHC edges have $p_{3,5}(e) \geq \frac{2}{3} + \varepsilon$ based on frequency quadrilaterals. It is granted especially for big scale of TSP.

$K$ depends on the parameter $\varepsilon$. The bigger the $\varepsilon$ is, the smaller the $K$ is. For the edges with $p_{3,5}(e) < \frac{2}{3} + \varepsilon$ (or $p_5(e) < \frac{1}{3} + \varepsilon$), there is a small probability for them to have $\left[\left(\frac{2}{3} + \varepsilon\right)N\right]$ (or $\left[\frac{1}{3} + \varepsilon\right]N$) frequency quadrilaterals where their frequency is 3 or 5 (or only 5) among $N$ frequency quadrilaterals. These edges will be trimmed according to the frequency threshold. For the other edges with $p_{3,5}(e) \geq \frac{2}{3} + \varepsilon$ (or $p_5(e) \geq \frac{1}{3} + \varepsilon$), there is a big probability that they have $\left[\left(\frac{2}{3} + \varepsilon\right)N\right]$ (or $\left[\frac{1}{3} + \varepsilon\right]N$) frequency quadrilaterals where their frequency is 3 or 5 (or only 5). However, the number of these edges is very small according to formula (2). Most edges will have approximate $\left[\frac{2}{3}N\right]$ (or $\left[\frac{1}{3}N\right]$) frequency quadrilaterals where their frequency is 3 or 5 (or only 5) among $N$ frequency quadrilaterals. These edges of small $F(e)$ will be neglected. Thus, we will preserve a small number $K$ of edges in the subgraphs. Through experiments, we can approximately determine the parameter values of $N$ and $\varepsilon$ for various TSP instances. In the next section, we will design a heuristic algorithm in order to compute a subgraph for TSP based on the Theorem (1).

5. The Heuristic Algorithm

Some techniques need to tackle for computing a subgraph for TSP used frequency quadrilaterals. For an OHC edge, $p_{3,5}(e) \to 1$ or $p_5(e) \to 1$ as $n$ is big enough. In real-life applications, TSP scale is generally limited. Therefore, $p_{3,5}(e)$ or $p_5(e)$ of some OHC edges will be smaller than 1, especially for small and medium scale of TSP instances. If we select $N$ frequency quadrilaterals randomly for an edge $e$, $F(e)$ of some OHC edges will deviate from $4N$ or $5N$. In order to compute a big $F(e)$ for the edges in OHC, we should choose the frequency quadrilaterals where their frequency is 5 or 3. In a frequency quadrilateral, the two opposite edges having the minimum sum distance have the frequency quadrilaterals. We first order the quadrilaterals in order to compute a big frequency for some OHC edges. Once $e \in \left(\frac{n - 2}{2}\right)$ quadrilaterals in $K_n$. Given edge $e$, we have to order the $\left(\frac{n - 2}{2}\right)$ opposite edges according to their distances for constructing $N$ suitable quadrilaterals. The distances sorting for $\left(\frac{n}{2}\right)$ edges will consume $O(n^4 \log n)$ time. To save computation time, we use another way to choose the opposite edges for $e$ to build $N$ quadrilaterals. We first order the $\left(\frac{n}{2}\right)$ edges according to their distances from small to big values and an edge sequence $\left(e_1, e_2, ..., e_{\frac{n}{2}}\right)$ is obtained. The first edge $e_1$ has the minimum distance and the last edge $e_{\frac{n}{2}}$ is the longest. For an edge $e_i \in \left(e_1, e_2, ..., e_{\frac{n}{2}}\right)$ where $1 \leq i \leq \frac{n}{2}$, we choose its opposite edges $g$ from the edge subsequence $\left(e_1, e_2, ..., e_{i+m}\right)$ where $m$ is a given number and $i + m \leq \frac{n}{2}$. With this method, we only order the $\left(\frac{n}{2}\right)$ edges once and it needs $O(n^2 \log n)$ time if the quick sorting algorithm is adopted. The parameter $m$ is
close to $i$ to ensure that the distance $d(g)$ is not far bigger than $d(e_i)$. In addition, the previous several shortest edges $e_i$ are always maintained, such as $i < 10$. This treatment does not affect the computation time and results much.

The other question is how many frequency quadrilaterals we should choose for every edge $e_i$. If $N$ is too small, the method will lose the power of the probability model and binomial distribution model. The number $K$ computed with formula (2) will be big as we use a small frequency threshold $F$. Otherwise, we will lose some OHC edges in the subgraphs if the frequency threshold $F$ is too big. When a big number of frequency quadrilaterals are used, the OHC edges will be preserved even though a big frequency threshold is adopted. However, the algorithm will consume long computation time. The suitable parameter $N$ can be determined according to experimental results. We suggest $N < n$ in order to save the computation time.

Since we choose the quadrilaterals for $e$ where $d(e) + d(g)$ is small, the frequency of $e$ will be 5 or 3 in most of the $N$ corresponding frequency quadrilaterals if $e$ has a big probability $p_{3,5}$ or $p_5$ in $K_n$. Otherwise, $e$ will not have many such frequency quadrilaterals. To simplify the computation, we compute $F(e)$ based on the frequency quadrilaterals where edge $e$ has the frequency $f = 5$. The frequency threshold $F$ is assigned a value near to $5N$. The experiments in the next section showed that $F(e)$ of most of OHC edges is close to $5N$ for the TSP instances in TSPLIB[16].

Given a TSP instance on $n$ vertices, $\binom{n}{2}$ edges are ordered first to form an edge sequence $s = (e_1, e_2, ..., e_{\binom{n}{2}})$. We use the quick sorting algorithm and it needs $O(n \log n)$ time. The heuristic algorithm to compute a subgraph is given in Figure 3.

At the first step, the prepared datum is input. They are the vertex set, edges’ distances in $K_n$ and the ordered edge sequence $e_1, e_2, ..., e_{\binom{n}{2}}$. The three parameters $N$, $m$ and $F$ are also assigned at the first step. The following steps are the kernel computation process of the heuristic algorithm. For an edge $e_i$ ($1 \leq i \leq \binom{n}{2}$), it first chooses $N$ non-adjacent edges $g$ for $e_i$ from the edge subsequence $(e_1, e_2, ..., e_{i+m})$ where $i + m \leq \binom{n}{2}$. $m$ is small to meet the distance constraint $d(g) < cd(e_i)$. Each pair of edges $g$ and $e_i$ form one quadrilateral. For the short edges $e_i$, the sum distance $d(e_i) + d(g)$ in each of the $N$ quadrilaterals is small. Thus, the frequency of $e_i$ will be 5 in almost of all the $N$ frequency quadrilaterals. On the other side, this method cannot guarantee a long edge $e_i$ to have frequency 5 in most of the $N$ frequency quadrilaterals. After $N$ quadrilaterals containing $e_i$ and $g$ are determined, they are converted into the corresponding frequency quadrilaterals. Then, $F(e_i)$ is computed with the frequency quadrilaterals in which the frequency of $e_i$ is 5. For the edges in OHC, they will have frequency 5 in most of the $N$ frequency quadrilaterals. After $F(e_i)$ is calculated, we compare $F(e_i)$ with $F$. If $F(e_i) > F$, $e_i$ is preserved in the subgraph. Otherwise, $e_i$ is eliminated. The computation process runs until all of edges are traversed. The final step is to output the subgraph.
Figure 3. The heuristic algorithm to compute a subgraph for TSP. The computation time of the algorithm relies on two parameters: the scale $n$ of TSP and the number $N$ of used quadrilaterals. In the kernel computation process, there are two loops. In the outer loop, the index $i$ goes from 1 to $\binom{n}{2}$ for every edge in the edge sequence. In the inner loop, the index $k$ goes from 1 to $N$ to choose $N$ non-adjacent edges $g$ for each edge. Thus, the heuristic algorithm requires $O(n^2)$ computation time. The parameter $N$ can be adjusted to guarantee the accuracy and efficiency of the algorithm. In current stage, we suggest $N \ll n$ so the computation time of the heuristic algorithm is smaller than $O(n^2)$.

There are three parameters $m$, $N$, and $F$ in the algorithm. To save the computation time, $N$ should be as small as possible. However, the probability model does not work well as $N$ is too small, for example $N<30$. In general, the $N$ value in [100, 1000] is a good choice for moderate and large scale of TSP. A small value of $m$ is useful to compute the high frequency for the OHC edges. It should not be too small for choosing $N$ opposite edges for some very short edges. In applications, $m$ can be assigned a value close to the value of $N$. A big value of $F$ is useful to cut many edges of the lower frequency. According to Formula (1) and Theorem 1, the frequency of an OHC edge will bigger than that of a common edge. Plus the selection of frequency quadrilaterals for the OHC edges, the value of $F$ will tend to $5N$. Therefore, $F$ can be assigned a value above $4.9N$ and it will increase according to $n$.

6. Experiment and Analysis
In this section, we would do experiments for TSP instances to test the heuristic algorithm. The TSP instances are selected from TSPLIB. The heuristic algorithm is run on a laptop with a 2.3GHz CPU and 4.0GB inner memory. Before we execute the heuristic algorithm, OHCs of these TSP instances are computed with Concorde online[17]. The OHC is used to verify whether the preserved subgraphs contain
OHC or how many OHC edges are lost in the subgraphs under the assignments of the three parameters $N$, $m$ and $F$.

There is one problem we should explain before the experiments. For some TSP instances, they have many equal-weight edges. Given a quadrilateral ABCD, the three sum distances $d_{AB} + d_{CD}$, $d_{AD} + d_{BC}$, $d_{AC} + d_{BD}$ will usually equal. In this case, it is hard to compute the six optimal 4-vertex paths and the exact frequency quadrilateral. To compute one frequency quadrilateral for such ABCD, we add small random distances $rd \in [0,1]$ to the edges’ distances to make the three sum distances $d_{AB} + d_{CD}$, $d_{AD} + d_{BC}$, $d_{AC} + d_{BD}$ as unequal as possible. The function of small random distance $rd$ has been used in paper [13]. Due to the randomness of $rd$, some frequency quadrilaterals cannot be rightly computed with respect to OHC in one experiment. In probability, it will compute the useful frequency quadrilaterals in most experiments.

We first assign the three parameters $N$, $m$ and $F$ for the heuristic algorithm. For each TSP instance, the parameters may be different to compute the best subgraph due to the scale $n$ and different structures of TSP. We will choose the proper values for the three parameters for every TSP instance to compute the subgraphs.

### Table 2. The experimental results with fixed parameters $N$, $m$ and $F$.

| TSP  | $n$  | $N$ | $m$ | $F$  | Edges number $K/l$ | $d$ | $\bar{l}/s$ |
|------|------|-----|-----|------|--------------------|-----|----------|
| a280 | 280  | 100 | 100 | 4.9N | 1815               | 1800| 1846     | 13 | 0.3  |
| gr431| 431  | 600 | 100 | 4.9N | 5818/4             | 5826/3| 5863/4 | 27 | 3.5  |
| att532| 532  | 300 | 100 | 4.9N | 7988               | 7974/1| 7949/1 | 30 | 2.9  |
| u574 | 574  | 100 | 100 | 4.9N | 9997               | 10039| 9893    | 25 | 4.5  |
| rat783| 783  | 200 | 100 | 4.92N| 11508/1            | 11604/1| 11521/1 | 23 | 7.8  |
| pr1002| 1002 | 200 | 100 | 4.93N| 12226/2            | 12047/1| 12104/3 | 23 | 8.1  |
| si1032| 1032 | 200 | 100 | 4.94N| 17470             | 17565| 17513   | 30 | 11.3 |
| pcb1173| 1173| 200 | 100 | 4.94N| 17470             | 17565| 17513   | 30 | 11.3 |
| d1655| 1655 | 200 | 100 | 4.94N| 45864/1            | 45961| 45743   | 55 | 27   |
| u1817| 1817 | 200 | 100 | 4.94N| 54555             | 54778| 54882   | 60 | 33.7 |
| u2152| 2152 | 250 | 100 | 4.95N| 64520/1            | 64288| 64562   | 60 | 61.7 |
| pr2392| 2392| 250 | 100 | 4.95N| 94586             | 94760| 94078/1 | 79 | 79.9 |
| pcb3038| 3038| 300 | 100 | 4.95N| 148365/1           | 148402/1| 148379/3 | 98 | 164.1 |
| rl5934| 5934| 500 | 100 | 4.97N| 569592/1          | 570698/1| 570049  | 192| 1277.8|

In the first round of our experiment, we take $m=100$. $N$ and $F$ are not fixed. For each TSP instance, we first assign the number $N$ and then the frequency threshold $F$ is tried. For the big TSP instances, we use the $N$ and $F$ of relatively big values. We tried the frequency threshold $F$ to compute the subgraphs containing OHC or the subgraphs only lose a few OHC edges. Four types of TSP instances are used to verify the heuristic algorithm. For different types of TSP, they usually have different inherent structures. According to the preserved subgraphs, we can estimate the performance of the heuristic algorithm for different types of TSP. Since we add random distances with small value to edges’ distances, the results in experiments will be different. Thus, we give three experimental results for each TSP instance. The experimental results are demonstrated in Table 2. Under the assignments of parameters $N$, $m$ and $F$, we compute three subgraphs. $K$ denotes the number of edges. $l$ is the number of the lost OHC edges in each subgraph, which is behind $K$. $l=0$ is not shown in the table if the subgraph contains OHC. The average degree $\bar{d}$ for vertices in the three subgraphs is computed according to the three numbers $K$. The average computation time $\bar{l}$ of the three experiments is also recorded for each TSP instance.

With the assignments of $N$, $m$ and $F$, the heuristic algorithm computes the subgraphs for all the TSP instances. For each TSP instance, the subgraphs usually contain OHC and sometimes they lose a few OHC edges in some experiments. The probability that the subgraphs contain OHC is very high. It validates the theory in Sections 3 and 4. Many useless edges are trimmed from $K_n$ for all these TSP instances. In view of $K$ values, the number of edges is reduced by more than 20 times for all the Euclidean TSP instances. Moreover, the number of edges is reduced by more than 30 times for the
Euclidean TSP instances of big scale above 783. We also see the frequency threshold $F \geq 4.9N$ for these TSP instances. It means the frequency of each OHC edges is 5 in most of the $N$ selected frequency quadrilaterals. Thus, a big probability of the OHC edges $p_{3.5}(e) = \frac{2}{3} + \epsilon$ (or $p_{5}(e) = \frac{1}{3} + \epsilon$) base on such frequency quadrilaterals. In addition, $F$ increases following the incremental scale of TSP. For the other kinds of TSP instances, such as GEO (gr431) and ATT (att532) TSP, the number of edges in the subgraphs is reduced by more than 15 times. However, for the MATRIX TSP (si1032), the number of edges in the subgraphs is reduced by more than 44 times. It says the heuristic algorithm is useful to compute a subgraph either for Euclidean TSP instances or for the other types of TSP instances.

In [14], they used a set of shortest optimal four-vertex paths containing each edge to compute the frequency of edges and then eliminated the edges according to their frequency. In the paper [18], one iterative algorithm is used to eliminate the edges using all frequency quadrilaterals. At each iteration, one-third of the edges are cut according to the frequency of edges. In the above papers, the algorithms require $O(n^4)$ time and it is hard to apply to the big size of TSP instances. Comparing with the previous algorithms based on frequency quadrilaterals, the heuristic algorithm eliminates more edges at one step. Moreover, it requires $O(n^2)$ time.

It mentions that there are many quadrilaterals ABCD where $d_{AB} + d_{CD} = d_{AD} + d_{BC}$, $d_{AC} + d_{BD}$ are equal for some of these TSP instances, especially for ATT, GEO and MATRIX TSP instances. The small random distances $rd$ cannot guarantee to compute the biggest frequency for the OHC edges in every experiment. If we did more experiments for these examples, the better subgraphs will be computed.

### Table 3. The experimental results with various $N$ and $F$ for rat783 when $m=100$.

| $F$  | 50  | 100 | 150 | 200 | 250 | 300 | 350 | 400 | 450 | 500 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 4.9N| 18712 | 14344 | 12981 | 12258 | 11789 | 11453 | 11298 | 11167 | 11053 | 10979 |
| 4.91N| 9772/14 | 9905/4 | 9966 | 9940 | 10109 | 10038 | 10002 | 10007 | 9976 | 9221 |
| 4.92N| 9818/11 | 9788 | 9959 | 9939 | 9998 | 8516/1 | 8798 | 8960 | 9052 | 9130 |
| 4.93N| 9902/10 | 9978/3 | 9951/1 | 8321 | 8459 | 7477/1 | 7735 | 8040 | 7382 |
| 4.94N| 9867/8 | 9922/2 | 6951/3 | 7691/1 | 8184 | 7015 | 7430 | 6630 | 7043 | 7296 |
| 4.95N| 9832/13 | 5402/21 | 7021/5 | 5485/4 | 6469/4 | 5492/1 | 6188 | 5529 | 5988 | 5562/2 |
| 4.96N| 9882/10 | 5405/21 | 7005/3 | 5554/5 | 6429/1 | 5523/5 | 4828/5 | 5495/2 | 5019/2 | 5612 |
| 4.97N| 9811/7 | 5345/24 | 3867/23 | 5489/7 | 4595/10 | 3965/17 | 4833/4 | 4345/6 | 3994/4 | 3620/9 |
| 4.98N| 9759/12 | 5438/20 | 3920/29 | 3132/41 | 4619/10 | 3946/7 | 3458/15 | 3209/21 | 2913/27 | 3689/5 |
| 4.99N| 9918/9 | 5453/19 | 3912/39 | 3159/30 | 2714/39 | 2384/51 | 2157/57 | 1970/52 | 1834/55 | 1752/73 |

To show the change of results according to $N$ and $F$, we did the second round of experiments for some TSP instances. In this experiment, we change the parameters $N$ and $F$ to compute subgraphs for rat783. The parameter $m=100$. The number $K$ of edges and the number $l$ of the lost OHC edges in the subgraphs are illustrated in Table 3. $N$ changes from 50 to 500 and the difference between each two adjacent $N$s is 50. Given a specific $N$, the frequency threshold varies from $4.9N$ to $4.99N$ and the interval is 0.01N. For a fixed $N$, the bigger the frequency threshold $F$ is, the smaller the number $K$ is. Meanwhile, more OHC edges will be lost in the subgraphs as $F$ becomes bigger. If we use a small number $N$, the method will lose the power of the probability model and binomial distribution model. As $N$ is smaller, such as $N=50$, 100, etc., the smaller $F$ is suitable to compute a subgraph with more edges containing OHC. As $N$ rises, we can use the relatively bigger frequency threshold $F$ so that the sparser subgraphs are computed. Thus, the big value of $N$ is a good choice to compute the sparse subgraphs. The disadvantage is that the algorithm will require more computation time.
Figure 4. The preserved edges according to \( N, F \) and fixed \( m \)

The number of preserved edges according to \( N, F \) and \( m=100 \) is shown in Figure 4. One sees that for a specific value of \( F \), the bigger the \( N \) is, the smaller the number of edges in the subgraphs is. For example, the subgraphs computed according to \( N=500 \) is generally sparser than that computed according to \( N<500 \) if the same \( F \) value is used. It means a big \( N \) value is helpful to compute a sparse subgraph for TSP. This reason can be explained with formula (2). As \( N \) becomes bigger, the probability

\[
\left( \frac{2}{3} + \frac{1}{3} \right)^N \left( \frac{2}{3} \right)^N \left( \frac{1}{3} \right)^N
\]

will become smaller for a given parameter \( e \) since \( \left[ \frac{2}{3} + \frac{1}{3} \right] \) deviates from \( \frac{2}{3} \) further. Thus, \( K \) value computed with formula (2) decreases according to \( N \). In the heuristic algorithm, the number of edges with frequency \( F(e_1) > F \) becomes less as \( N \) rises. In addition, the OHC edges will have more frequency quadrilaterals where their frequency is 5 due to their big probability \( p_{3,5}(e) \) or \( p_5(e) \). Thus, \( F \) is improved and the subgraph with a small number of edges is computed. In general, \( F \) will approach 5N based on the big value of \( N \) and small value of \( m \). However, it will consume much computation time if \( N \) is very big.

\( m \) is important to choose the frequency quadrilaterals for an edge where it has the frequency 5. When \( m \) is small, the distances \( d(g) \) are less than \( d(e) \) in most of the selected quadrilaterals containing \( e \) and \( g \). If \( d(e) + d(g) \) is very small, the frequency of \( e \) will be 5 in most of these corresponding frequency quadrilaterals. Thus, \( F(e) \) will approach 5N. On the other hand, if \( m \) is too big, we will have a big probability to obtain \( d(g) > d(e) \) in the constructed quadrilaterals. In this case, \( d(e) + d(g) \) may be bigger than the sum distances of the other two pairs of opposite edges. There will be more frequency quadrilaterals where the frequency of \( e \) is 1 and 3. In such cases, a small \( F \) should be considered to compute a subgraph containing OHC but with more number of edges.

We did the third round of experiment for rat783 to see the preserved subgraphs according to changes of \( m \) and \( N \). \( F \) is assigned as 4.96N and \( N \) still changes from 50 to 500. We give three values below and
above 100 for \( m \), respectively. The minimum value of \( m \) is 10 and the maximum value of \( m \) is 1000. The number \( K \) of edges and the number \( l \) of lost OHC edges in the subgraphs are given in Table 4.

**Table 4.** The experimental results with various \( N \) and \( m \) for rat783 when \( F=4.96N \).

| \( m \) | 50   | 100  | 150  | 200  | 250  | 300  | 350  | 400  | 450  | 500  |
|--------|------|------|------|------|------|------|------|------|------|------|
| \( N \) | 100  | 150  | 200  | 250  | 300  | 350  | 400  | 450  | 500  | 550  |
| 10     | 9854/5 | 5429/13 | 6976/6 | 5480/4 | 6468 | 5507/2 | 4958/3 | 5521/1 | 5010/2 | 5599 |
| 30     | 9822/12 | 5517/13 | 7012/1 | 5504/4 | 6468/1 | 5466 | 4898/8 | 5498/2 | 5068/1 | 5506 |
| 50     | 9831/14 | 5415/19 | 6971/3 | 5506/3 | 6454/1 | 5557 | 4885/1 | 5560/1 | 5019/2 | 5572 |
| 70     | 9914/7 | 5469/13 | 7044/3 | 5491/8 | 6377 | 5463/1 | 4861/1 | 5557/2 | 4977/3 | 5584 |
| 200    | 9800/11 | 5377/21 | 6965/3 | 5449/5 | 6314/1 | 5429/3 | 4811/3 | 5467/2 | 4949 | 5465/2 |
| 500    | 9737/14 | 5275/25 | 6832/7 | 5376/4 | 6271/4 | 5467/2 | 4738/4 | 5387/1 | 4879 | 5359/1 |
| 1000   | 9380/21 | 5056/42 | 6672/4 | 5166/6 | 5989/3 | 5211/2 | 4548/10 | 5138/2 | 5110 | 4646/2 |

**Figure 5.** The preserved edges according to \( N, m \) and fixed \( F \)

The number of preserved edges according to \( N, m \) and \( F=4.96N \) is given in Figure 5. If \( N \) is fixed, the number of edges becomes smaller as \( m \) rises, see Figure 5. It means we construct less percent of the frequency quadrilaterals where the frequency of \( e \) is 5 among the \( N \) frequency quadrilaterals. \( F = 4.96N \) is bigger than \( F(e) \) for more OHC edges as \( m \) rises. As \( N \) is small (\( N\geq50, 100, 150, etc. \)), it is not suitable to use a big \( m \). Otherwise, the subgraphs will lose quite a few OHC edges if we use a big frequency threshold. As \( N \) is big (\( N\geq300 \)), the preserved graphs include nearly all of OHC edges whatever \( m \) changes. We could compute the subgraphs containing OHC according to a big \( N \) and the effect of \( m \) can be neglected. \( F(e) \) of the edges in OHC are determined by the structure of \( K_m \). The edges in OHC have the big probability \( p_{3,5}(e) \) or \( p_{5}(e) \) in their frequency quadrilaterals. One sees \( K \) becomes smaller as \( N \) and \( m \) become bigger. It implies us to compute a sparse subgraph with the big values of \( m \) and \( N \). In this case, the frequency threshold \( F \) also rises to \( 5N \).

Through the experiments, \( N \) plays the first important role to compute a sparse subgraph containing OHC for TSP. The bigger the \( N \) is, the bigger the \( F \) will be. \( m \) is the second useful parameter to compute a satisfactory subgraph. The big \( m \) is helpful to compute a sparse subgraph as parameter \( N \) is big enough. As \( N \) is small, we generally use a small \( m \). The third parameter \( F \) relies on \( N \). The smaller the \( N \) is, the smaller the \( F \) is and the bigger the \( K \) is. In addition, the scale of TSP is not neglected when we assign the three parameters. According to the results in the Table 2, 3 and 4, the three parameters \( N, m \) and \( F \) should be given the bigger values as \( n \) rises. According to the experimental results, we suggest \( N < n \),
\( m = o(n) \) and \( F \) is close to \( 5N \). When \( m \) is too big and \( i + m \) approaches \( \binom{m}{2} \), we will need a lot of frequency quadrilaterals for each edge to compute a sparse subgraph. If \( N \) is too big, we will compute the satisfactory subgraphs whereas the heuristic algorithm will consume long computation time, especially for large \( n \).

7. Conclusions

A subgraph is computed for TSP based on frequency quadrilaterals. The probability that the frequency of an OHC edge is 3 or 5 is bigger than \( 2/3 \) according to frequency quadrilaterals. This probability tends to 1 as the scale of TSP is big enough. The number of edges in the subgraphs is derived based on a binomial distribution model. A heuristic algorithm containing three parameters is designed to compute a subgraph for TSP. The experiments showed the number of edges in the subgraphs is reduced by more than 30 times for large scale of Euclidean TSP. If we use the proper parameter values, the better results will be computed. The heuristic algorithm based on the frequency quadrilaterals has much space to improve. In the future, we will study what kind of subgraphs are computed if the procedure is iterated.

Acknowledgements

The authors acknowledge the funds supported by the Fundamental Research Funds for the Central Universities (Nos. 2018ZD09 and 2018MS039) and NSFC (No.51205129). The work benefits from the facilities of 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] G. Gutin, A. Punnen. The traveling salesman problem and its variations. Paradigms of Combinatorial Optimization Problems & New Approaches, 2007, 4(2):193-205.
[2] J. Han, K. Micheline. Data Mining: Concepts and Techniques. Data Mining Concepts Models Methods & Algorithms Second Edition, 2006, 5(4):1-18.
[3] C. Jiang, F. Coenen, M. Zito. A survey of frequent subgraph mining algorithms. The Knowledge Engineering Review, 2013, 28(1):75-105.
[4] D.J. Cook, L.B. Holder. Substructure discovery using minimum description length and background knowledge. Journal of Artificial Intelligence Research, 1994, 231–255.
[5] M. Held, R.M. Karp. A dynamic programming approach to sequencing problems. Journal of the Society for Industrial & Applied Mathematics, 1962, 10(1):196-210.
[6] R. Bellman. Dynamic programming treatment of the travelling salesman problem. Journal of the ACM, 1962, Vol.9: 61–63.
[7] A. BjöJrklund. Determinant sums for undirected Hamiltonicity. Computer Science, 2010.
[8] M. Xiao, H. Nagamochi. An Exact Algorithm for TSP in Degree-3 Graphs Via Circuit Procedure and Amortization on Connectivity Structure. Algorithmica, 2016, 74(2):713-741.
[9] H. Blum. A transformation for extracting new descriptors of shape. Models for the Perception of Speech and Visual Form. Cambridge: MIT Press, 1967.362-380
[10] K.D. Boese. Cost versus distance in the traveling salesman problem, Technical Report CSD-950018, 1995
[11] W.X. Zhang, M. Looks. A novel local search algorithm for the traveling salesman problem that exploit backbones. Proceedings of the 19th International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, 2005, 343–350.
[12] Z.L. Ren, H. Jiang, J.F. Xuan, Z.X. Luo. An Accelerated-Limit-Crossing-Based Multilevel Algorithm for the p-Median Problem. IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society, 2012, 42(4), 1187–1202.
[13] Y. Wang, J.B. Remmel. A binomial distribution model for the traveling salesman problem based on frequency quadrilaterals. Journal of Graph Algorithms & Applications, 2016, 20(2), 411–434.
[14] Y. Wang. An approximate method to compute a sparse graph for traveling salesman problem. Expert Systems with Applications, 2015, 42(12):5150-5162.
[15] Y. Wang. Sufficient and necessary conditions for an edge in the optimal Hamiltonian cycle based on frequency quadrilaterals, Journal of Optimization Theory and Applications, 2019, 181, 671–683.

[16] G. Reinelt. TSPLIB (1995). http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

[17] H. Mittelmann. NEOS Server for Concorde (2015). http://neos-server.org/neos/solvers/co:concorde/TSP.html

[18] Y. Wang, J.B. Remmel. A method to compute the sparse graphs for traveling salesman problem based on frequency quadrilaterals. International Workshop on Frontiers in Algorithmics (FAW 2018), Guang zhou, China, 2018, 286-299.