A 3/4 Differential Approximation Algorithm for Traveling Salesman Problem

Yuki Amano\(^1\) and Kazuhisa Makino\(^1\)

\(^1\)Research Institute for Mathematical Sciences, Kyoto University, Kyoto, Japan. \{ukiamano, makino\}@kurims.kyoto-u.ac.jp

Abstract

In this paper, we consider differential approximability of the traveling salesman problem (TSP). We show that TSP is $3/4$-differential approximable, which improves the currently best known bound $3/4 - O(1/n)$ due to Escoffier and Monnot in 2008, where $n$ denotes the number of vertices in the given graph.

1 Introduction

The traveling salesman problem (TSP) finds a shortest Hamiltonian cycle in a given complete graph with edge length, when a cycle is called Hamiltonian (also called a tour) if it visits every vertex exactly once. TSP is one of the most fundamental NP-hard optimization problems in operations research and computer science, and has been intensively studied from both practical and theoretical viewpoints [7, 19, 21, 22]. It has a number of applications such as planning, logistics, and the manufacture of microchips [4, 11]. Because of these importance, many heuristics and exact algorithms have been proposed [3, 13, 14, 15]. From a viewpoint of computational complexity, TSP is NP-hard, even in the Euclidean case, which includes the metric case. It is known that metric TSP is approximable with factor 1.5 [6], and inapproximable with factor 117/116 [5]. Euclidean TSP admits a polynomial-time approximation scheme (PTAS), if the dimension of the Euclidean space is bounded by a constant [1]. We note that the approximation factors (i.e., ratios) above are widely used to analyze approximation algorithms.

Let $\Pi$ be an optimization problem, and let $I$ be an instance of $\Pi$. We denote by $\text{opt}(I)$ the value of an optimal solution to $I$. For an approximation algorithm $A$ for $\Pi$, we denote by $\text{apx}_A(I)$ the value of the approximate solution computed by $A$ for the instance $I$. Let

$$r_A(I) = \frac{\text{apx}_A(I)}{\text{opt}(I)},$$

and define the standard approximation ratio of $A$ by $\sup_{I \in \Pi} r_A(I)$, where we assume that $\Pi$ is a minimization problem. Although the standard approximation ratio is well-studied and an important concept in algorithm theory, it is not invariant under affine transformation of the objective function. Namely, if the objective function $f(x)$ is replaced by $a + bf(x)$ for some constant $a$ and $b$, which might depend on the instance $I$, the standard ratio is not preserved. For example, the vertex cover problem and the independent set problem have affinely dependent objective functions. However they have different characteristics in the standard approximation ratio. The vertex cover problem is 2-approximable [20], while the independent set problem is inapproximable within $O(n^{1-\epsilon})$ for any $\epsilon > 0$ [9], where $n$ denotes the number of vertices in a given graph. In order to remedy to this phenomenon, Demange and Paschos [8] proposed the differential approximation ratio defined by $\sup_{I \in \Pi} \rho_A(I)$, where

$$\rho_A(I) = \frac{\text{wor}(I) - \text{apx}_A(I)}{\text{wor}(I) - \text{opt}(I)}$$

1
and \( \text{wor}(I) \) denotes the value of a worst solution to \( I \). Note that for any instance \( I \) of \( \Pi \)
\[
\text{apx}_A(I) = \rho_A(I) \text{opt}(I) + (1 - \rho_A(I)) \text{wor}(I).
\]
Thus we have \( 0 \leq \rho_A(I) \leq 1 \) and the larger \( \rho_A(I) \) implies the better approximation for the instance \( I \).
Moreover, by definition, the differential approximation ratio remains invariant under affine transformation of the objective function. For this, it has been recently attracted much attention in approximation algorithm [2]. It is known [17] that TSP, metric TSP, max TSP, and max metric TSP are affinely equivalent, i.e., their objective functions are transferred to each other by affine transformations, where max TSP is the problem to find a longest Hamiltonian cycle and max metric TSP is max TSP, in which the input weighted graph satisfies the metric condition. Therefore, these problems have the identical differential approximation ratio.

Hassin and Khuller [12] first studied differential approximability of TSP, and showed that it is 2/3-differential approximable. Escoffier and Monnot [10] improved it to \( 3/4 - O(1/n) \), where \( n \) denotes the number of vertices of a given graph. Monnot et al. [16, 18] showed that TSP is 3/4-differential approximable if each edge length is restricted to one or two.

In this paper, we show that TSP is 3/4-differential approximable, which improves the currently best known results [10, 16, 18]. Our algorithm is based on an idea in [10] for the case in which a given graph \( G \) has an even number of vertices and a triangle (i.e., cycle with 3 edges) is contained in a minimum weighted 2-factor of \( G \). Their algorithm first computes minimum weighted 1- and 2-factors of a given graph, modify them to four path covers \( P_i \) (for \( i = 1, \ldots, 4 \)), and then extend each path cover \( P_i \) to a tour by adding edge set \( F_i \) to it in such a way that at least one of the tours guarantees 3/4-differential approximation ratio. Here the definitions of factor and path cover can be found in Section 2. We generalize their idea to the general even case. Note that \( \bigcup_{i=1}^4 F_i \) in their algorithm always forms a tour, where in general it does not. We show that there exists a way to construct path covers such that the length of \( \bigcup_{i=1}^4 F_i \) is at most the worst tour length. Our algorithm for odd case is much more involved. For each path with three edges, we first construct a 2-factor and two path covers of a given graph which has minimum length among all these which completely and partially contains the path, modify them to eight path covers, and then extend each path cover to a tour, in such a way that at least one of the eight tours guarantees 3/4-differential approximation ratio.

The rest of the paper is organized as follows. In Section 2, we define basic concepts of graphs and discuss some properties on 2-matchings, which will be used in the subsequent sections. In Sections 3 and 4, we provide an approximation algorithms for TSP in which a given graph \( G \) has even and odd numbers of vertices, respectively.

## 2 Preliminary

Let \( G = (V, E) \) be an undirected graph, where \( n \) and \( m \) denote the number of vertices and edges in \( G \), respectively. In this paper, we assume that a given graph \( G \) of TSP is complete, i.e., \( E = \binom{V}{2} \), and it has an edge length function \( \ell : E \to \mathbb{R}_+ \), where \( \mathbb{R}_+ \) denotes the set of nonnegative reals. For a set \( F \subseteq E \), let \( V(F) \) denote the set of vertices with incident edges in \( F \), i.e., \( V(F) = \{ v \in V \mid \exists (v, w) \in F \} \). A set \( F \subseteq E \) is called spanning if \( V(F) = V \), and acyclic if \( F \) contains no cycle. For a positive integer \( k \), a set \( F \subseteq E \) is called a \( k \)-matching (resp., \( k \)-factor) if each vertex has at most (resp., exactly) \( k \) incident edges in \( F \). Here 1-matching is simply called a matching. Note that an acyclic 2-matching \( F \) corresponds to a family of vertex-disjoint paths denoted by \( \mathcal{P}(F) \subseteq 2^E \). A 2-matching is called a path cover if it is spanning and acyclic. For a set \( F \subseteq E \), \( V_1(F) \) and \( V_2(F) \) respectively denote the sets of vertices with one and two incident edges in \( F \). For a set \( F \subseteq E \) and a vertex \( v \in V \), let \( \delta_F(v) = \{ e \in F \mid e \text{ is incident to } v \} \).

**Definition 1.** A pair of spanning 2-matchings \((S, T)\) is called valid if it satisfies the following three conditions:

1. \( T \) is acyclic.
2. \( \delta_T(v) = \delta_S(v) \) for any \( v \in V_2(S) \cap V_2(T) \).
3. \( V(C) \neq V(P) \) for any cycle \( C \subseteq S \) and any path \( P \subseteq \mathcal{P}(T) \).

## 2 Preliminary

Let \( G = (V, E) \) be an undirected graph, where \( n \) and \( m \) denote the number of vertices and edges in \( G \), respectively. In this paper, we assume that a given graph \( G \) of TSP is complete, i.e., \( E = \binom{V}{2} \), and it has an edge length function \( \ell : E \to \mathbb{R}_+ \), where \( \mathbb{R}_+ \) denotes the set of nonnegative reals. For a set \( F \subseteq E \), let \( V(F) \) denote the set of vertices with incident edges in \( F \), i.e., \( V(F) = \{ v \in V \mid \exists (v, w) \in F \} \). A set \( F \subseteq E \) is called spanning if \( V(F) = V \), and acyclic if \( F \) contains no cycle. For a positive integer \( k \), a set \( F \subseteq E \) is called a \( k \)-matching (resp., \( k \)-factor) if each vertex has at most (resp., exactly) \( k \) incident edges in \( F \). Here 1-matching is simply called a matching. Note that an acyclic 2-matching \( F \) corresponds to a family of vertex-disjoint paths denoted by \( \mathcal{P}(F) \subseteq 2^E \). A 2-matching is called a path cover if it is spanning and acyclic. For a set \( F \subseteq E \), \( V_1(F) \) and \( V_2(F) \) respectively denote the sets of vertices with one and two incident edges in \( F \). For a set \( F \subseteq E \) and a vertex \( v \in V \), let \( \delta_F(v) = \{ e \in F \mid e \text{ is incident to } v \} \).

**Definition 1.** A pair of spanning 2-matchings \((S, T)\) is called valid if it satisfies the following three conditions:

1. \( T \) is acyclic.
2. \( \delta_T(v) = \delta_S(v) \) for any \( v \in V_2(S) \cap V_2(T) \).
3. \( V(C) \neq V(P) \) for any cycle \( C \subseteq S \) and any path \( P \subseteq \mathcal{P}(T) \).
Figure 1: A valid pair \((S, T)\) of spanning 2-matchings.

Figure 1 shows a valid pair of spanning 2-matchings.

**Lemma 2.** Let \((S, T)\) be a valid pair of spanning 2-matchings. If \(S\) contains a cycle \(C\), then \(C\) contains two edges \(e_i\) for \(i = 1, 2\) such that \(S_i = S \setminus \{e_i\}\) and \(T_i = S \cup \{e_i\}\) satisfy the following three conditions:

\[
\begin{align*}
(S_i, T_i) & \text{ is valid for } i = 1, 2, \\
V_1(S_i) \cup V_1(T_i) & = V_1(S) \cup V_1(T) \quad \text{and} \\
V_1(S_i) \cap V_1(T_i) & = V_1(S) \cap V_1(T) \text{ for } i = 1, 2.
\end{align*}
\]

Proof. Let \(C = \{v_0, v_1, v_2, \ldots, v_k\}\) for \(k \geq 3\), where \(v_k = v_0\). If \(P(T)\) contains a \((s, t)\)-path \(P\) with \(s \in V(C)\) and \(t \notin V(C)\), then it follows from (1) that \(V(P) \cap V(C) = \{s\}\). We assume that \(s = v_1\) without loss of generality. Let \(e_1 = (v_0, v_1)\) and \(e_2 = (v_1, v_2)\). It is not difficult to see that \((S_i, T_i)\) is valid and \(P \cup \{e_i\}\) is a path for every \(i = 1, 2\). On the other hand, if \(P(T)\) contains a \((s, t)\)-path \(P\) with \(s, t \in V(C)\), we assume without loss of generality that \(s = v_1\), \(t = v_j\) for some \(j\) with \(2 \leq j \leq k - 1\), and \(P\) does not contain \((v_0, v_1)\). Note that such a path exists, since \(T\) is spanning. Define \(e_1 = (v_0, v_1)\) and \(e_2 = (v_j, v_{j+1})\). By (1) and (2), we can show that \((S_i, T_i)\) is valid and \(P \cup \{e_i\}\) is a path for every \(i = 1, 2\). This completes the proof.

Note that \((S_1, T_1)\) and \((S_2, T_2)\) in Lemma 2 satisfy

\[
S_i \cup T_i = S \cup T \quad \text{and} \quad S_i \cap T_i = S \cap T \quad \text{for } i = 1, 2,
\]

which immediately implies

\[
\ell(S_i) + \ell(T_i) = \ell(S) + \ell(T) \quad \text{for } i = 1, 2,
\]

where \(\ell(F) = \sum_{e \in F} \ell(e)\) for a set \(F \subseteq E\).

Figure 2 shows two pairs \((S \setminus \{e_c\}, T \cup \{e_c\})\) and \((S \setminus \{e_d\}, T \cup \{e_d\})\) satisfying (3), (4) and (5), which are obtained from \((S, T)\), \(e_1 = e_c\), and \(e_2 = e_d\) in Fig. 1.

### 3 Approximation for even instances

In this section, we construct an approximation algorithm for TSP in which a given graph has an even number of vertices. Our algorithm first constructs four path covers from minimum weighted 1- and 2-factors of a given graph \(G\), and then extend each path cover to a tour in such a way that at least one of the tours guarantees 3/4-differential approximation ratio.
Let us first describe the procedure \texttt{FourPathCovers}. Let \((S, T)\) be a valid pair of spanning 2-matchings of \((G, \ell)\) such that \(S\) is a 2-factor. The procedure computes from \((S, T)\) four path covers \(S_1, S_2, T_1,\) and \(T_2\) that satisfies (4), (6), and (9).

\begin{align*}
V_1(S_i) &\cup V_1(T_i) = V_1(T) \quad \text{and} \quad V_1(S_i) \cap V_1(T_i) = \emptyset \quad \text{for} \quad i = 1, 2, \quad (8) \\
\text{and} \\
\text{there exist} \; e_1, e_2 \in E \; \text{and} \; P \in \mathcal{P}(T_1 \cap T_2) \; \text{such that} \\
T_1 \setminus T_2 = \{e_1\}, \quad T_2 \setminus T_1 = \{e_2\}, \quad P \cup \{e_1\} \in \mathcal{P}(T_1), \quad \text{and} \quad P \cup \{e_2\} \in \mathcal{P}(T_2). \quad (9)
\end{align*}

In Fig. 3 we apply \texttt{Procedure FourPathCovers} to \((S, T)\) in Fig. 1.

\begin{algorithm}
\begin{algorithmic}
\Function{FourPathCovers}{$S, T$}
\State /*\((S, T)\) is a valid pair of spanning 2-matchings such that \(S\) has a cycle. The procedure returns 4 path covers \(S_1, S_2, T_1,\) and \(T_2\) that satisfies (4), (6), and (9).*\/
\If{\(S\) has exactly one cycle}
\State Take two edges \(e_1\) and \(e_2\) in Lemma 2.
\State \textbf{return} \(S_1 = S \setminus \{e_1\}, \; T_1 = T \cup \{e_1\}, \; S_2 = S \setminus \{e_2\},\) and \(T_2 = T \cup \{e_2\}\)
\Elses /*\(S\) has at least two cycles.*
\State Take an edge \(e_1\) in Lemma 2.
\State \textbf{return} \texttt{FourPathCovers} \((S \setminus \{e_1\}, T \cup \{e_1\})\)
\EndIf
\EndFunction
\end{algorithmic}
\end{algorithm}

\textbf{Lemma 3.} For a graph \(G = (V, E)\), let \((S, T)\) be a valid pair of spanning 2-matchings such that \(S\) has a cycle. Then \texttt{Procedure FourPathCovers} returns four path covers \(S_1, S_2, T_1,\) and \(T_2\) that satisfy (4), (6), and (9). Furthermore, if \(S\) is addition a 2-factor of \(G\), then the four path covers satisfy (8).

\textit{Proof.} By repeatedly applying Lemma 2 to \((S, T)\), we can see that four path covers \(S_1, S_2, T_1,\) and \(T_2\) returned by \texttt{Procedure FourPathCovers} satisfy (4), (6), and (9). Furthermore, if \(S\) is a 2-factor of \(G\), we have (8), since \(V_1(S) = \emptyset\). \hfill \Box

Note that \((S, T)\) is a valid and \(V_1(S) \cup V_1(T) = V\), if \(S\) and \(T\) are 2- and 1-factor of \(G\), respectively. Let \(S\) and \(T\) be 2- and 1- factors of \(G\), respectively. Note that our algorithm explain later makes use of minimum weighted 2-factor \(S\) and 1-factor \(T\) of \((G, \ell)\) which can be computed from \((G, \ell)\) in polynomial.
Figure 3: Two pairs \((S_1, T_1)\) and \((S_2, T_2)\) computed by Procedure \textbf{FourPathCovers} for a valid pair \((S, T)\), \(e_1^{(1)} = e_a, e_1^{(2)} = e_b, e_1^{(3)} = e_c\) and \(e_2^{(3)} = e_d\) in Fig. 1, where \(e_1^{(j)}\) denotes the edge chosen as \(e_j\) in the \(j\)-th round of the procedure.

Figure 4: Two cases \(p_2 = p_3\) and \(p_2 \neq p_3\) for path covers \(S_1\) and \(S_2\) returned by Procedure \textbf{FourPathCovers}(\(S, T\)).

Let \(e_1 = (p_1, p_2)\) and \(e_2 = (p_3, p_4)\) be edges in Lemma 3. Since \(e_1\) and \(e_2\) are chosen from a cycle \(C\), we can assume that \(p_1 \neq p_3, p_1 \neq p_4, p_2 \neq p_1, p_2 \neq p_3\), where \(p_2 = p_3\) might hold. We note that \(\mathcal{P}(S_1) \setminus \mathcal{P}(S_2)\) consists of a \((p_1, p_2)\)-path \(P_1 = C \setminus \{e_1\}\), and \(\mathcal{P}(S_2) \setminus \mathcal{P}(S_1)\) consists of a \((p_3, p_4)\)-path \(P_2 = C \setminus \{e_2\}\).

Let \(Q_i, (i = 1, \ldots, k)\) denote vertex-disjoint \((x_i, y_i)\)-paths such that \(\{Q_1, \ldots, Q_k\} = \mathcal{P}(S_1) \cap \mathcal{P}(S_2)\) and \(x_1\) and \(y_1\) satisfy

\[
\ell(p_2, x_1) + \ell(p_3, y_1) \leq \ell(p_2, y_1) + \ell(p_1, x_1).
\]

Figure 4 shows \(S_1\) and \(S_2\) computed by Procedure \textbf{FourPathCover}(\(S, T\)), where two cases \(p_2 = p_3\) and \(p_2 \neq p_3\) are separately described. Define \(A_1\) and \(A_2\) by

\[
A_1 = \{(p_2, x_1)\} \cup \{\{y_i, x_{i+1}\} | i = 1, \ldots, k-1\} \cup \{(y_k, p_1)\}
\]

\[
A_2 = \{(p_3, y_1)\} \cup \{\{x_i, y_{i+1}\} | i = 1, \ldots, k-1\} \cup \{(x_k, p_4)\}
\]
Lemma 4. Two sets $A_1$ and $A_2$ defined in (11) satisfy the following three conditions.

(i) $S_i \cup A_i$ is a tour of $G$ for $i = 1, 2$.

(ii) $V(A_i) = V_1(S_i)$ for $i = 1, 2$.

(iii) $A_1 \cap A_2 = \emptyset$ and $A_1 \cup A_2$ consists of

   (iii-1) a $(p_1, p_4)$-path if $p_2 = p_3$.

   (iii-2) vertex-disjoint $(p_1, p_3)$- and $(p_2, p_4)$-paths if $p_2 \neq p_3$ and $k$ is odd.

   (iii-3) vertex-disjoint $(p_1, p_2)$- and $(p_3, p_4)$-paths if $p_2 \neq p_3$ and $k$ is even.

Proof. Note that $\mathcal{P}(S_1) = \{Q_1, \ldots, Q_k\} \cup \{P_1\}$ and $\mathcal{P}(S_2) = \{Q_1, \ldots, Q_k\} \cup \{P_2\}$. Thus it follows from the definition of $A_1$ and $A_2$.

Figure 6 shows two edge sets $A_1$ and $A_2$ for $S_1$ and $S_2$ in Fig. 3.

where the illustration can be found in Fig. 5. Then we have the following lemma.
Let us next construct $B_1$ and $B_2$. Let $O_i$ ($i = 1, \ldots, d$) denote vertex-disjoint $(z_i, w_i)$-paths such that $\{O_1, \ldots, O_d\} = \mathcal{P}(T_1) \cap \mathcal{P}(T_2)$. Note that $\mathcal{P}(T_1) \cap \mathcal{P}(T_2) = \emptyset$ (i.e., $d = 0$) might hold. We separately consider the following four cases, where the illustration can be found in Fig. 7.

1. $p_2 = p_3$ and $\mathcal{P}(T_1 \cap T_2)$ contains a $(p_1, p_4)$-path.
2. $p_2 = p_3$ and $\mathcal{P}(T_1 \cap T_2)$ contains no $(p_1, p_4)$-path.
3. $p_2 \neq p_3$ and $\mathcal{P}(T_1 \cap T_2)$ contains $(p_1, p_4)$- and $(p_2, p_3)$-paths.
4. $p_2 \neq p_3$ and $\mathcal{P}(T_1 \cap T_2)$ contains a $(p_2, p_3)$-path and no $(p_1, p_4)$-path.

Here we recall that $e_1 = (p_1, p_2)$ and $e_2 = (p_3, p_4)$ satisfy Lemma 3.

**Case 1:** Let $R_1$ denote a $(p_1, p_4)$-path in $\mathcal{P}(T_1 \cap T_2)$, and for some vertex $q_2$, let $R_2$ denote $(p_2, q_2)$-path in $\mathcal{P}(T_1 \cap T_2)$. Then, we have

$$\mathcal{P}(T_1) = \{O_1, \ldots, O_d\} \cup \{R_1 \cup \{e_1\} \cup R_2\},$$

$$\mathcal{P}(T_2) = \{O_1, \ldots, O_d\} \cup \{R_1 \cup \{e_2\} \cup R_2\},$$

where $R_1 \cup \{e_1\} \cup R_2$ and $R_1 \cup \{e_2\} \cup R_2$ are $(p_4, q_2)$- and $(p_1, q_2)$-paths, respectively. Define $B_1$ and $B_2$ by

$$B_1 = \begin{cases} \{(q_2, p_4)\} & \text{if } d = 0 \\ \{(q_2, z_1)\} \cup \{(w_i, z_{i+1}) \mid i = 1, \ldots, d-1\} \cup \{(w_d, p_4)\} & \text{if } d \geq 1 \end{cases},$$

$$B_2 = \begin{cases} \{(q_2, p_1)\} & \text{if } d = 0 \\ \{(q_2, w_1)\} \cup \{(z_i, w_{i+1}) \mid i = 1, \ldots, d-1\} \cup \{(z_d, p_1)\} & \text{if } d \geq 1 \end{cases},$$

as illustrated in Fig. 8. By definition, we have

$$T_i \cup B_i$$

is a tour of $G$ for $i = 1, 2,$

$B_1 \cap B_2 = \emptyset$ and $V(B_i) = V_i(T_1)$ for $i = 1, 2,$ and

$B_1 \cup B_2$ is a $(p_1, p_4)$-path. \hspace{1cm} (13)

**Case 2:** For some vertices $q_1, q_2$ and $q_4$, let $R_1$, $R_2$ and $R_4$ respectively denote $(p_1, q_1)$-, $(p_2, q_2)$-, and $(p_4, q_4)$-paths in $\mathcal{P}(T_1 \cap T_2)$. Then, we have

$$\mathcal{P}(T_1) = \{O_1, \ldots, O_d\} \cup \{R_4, R_1 \cup \{e_1\} \cup R_2\},$$

$$\mathcal{P}(T_2) = \{O_1, \ldots, O_d\} \cup \{R_1, R_4 \cup \{e_2\} \cup R_2\},$$

7

Figure 6: Two edge sets $A_1$ and $A_2$ for path covers $S_1$ and $S_2$ in Fig. 3.
Figure 7: Four cases for path covers $T_1$ and $T_2$ returned by **Procedure** `FourPathCovers(S, T)`.

```plaintext
Figure 7: Four cases for path covers $T_1$ and $T_2$ returned by **Procedure** `FourPathCovers(S, T)`.
```
where \( R_1 \cup \{ e_1 \} \cup R_2 \) and \( R_4 \cup \{ e_2 \} \cup R_2 \) are \((q_1, q_2)\)- and \((q_4, q_2)\)-paths, respectively. Define \( B_1 \) and \( B_2 \) by

\[
\begin{align*}
B_1 &= \begin{cases} 
\{(q_2, q_4), (p_4, q_1)\} & \text{if } d = 0 \\
\{(q_2, z_1) \cup \{(w_i, z_{i+1}) \mid i = 1, \ldots, d-1\} \cup \{(w_d, q_4), (p_4, q_1)\} & \text{if } d \geq 1
\end{cases} \\
B_2 &= \begin{cases} 
\{(q_2, q_1), (p_1, q_1)\} & \text{if } d = 0 \\
\{(q_2, w_1) \cup \{(z_i, w_{i+1}) \mid i = 1, \ldots, d-1\} \cup \{(z_d, q_1), (p_1, q_4)\} & \text{if } d \geq 1,
\end{cases}
\end{align*}
\]

as illustrated in Fig. 9. Similarly to Case 1, we have (13), (14) and (15).

**Case 3:** Let \( R_1 \) and \( R_2 \) respectively denote \((p_1, p_4)\)- and \((p_2, p_3)\)-paths in \( \mathcal{P}(T_1 \cap T_2) \). Then, we have

\[
\begin{align*}
\mathcal{P}(T_1) &= \{O_1, \ldots, O_d\} \cup \{R_1 \cup \{e_1\} \cup R_2\} \\
\mathcal{P}(T_2) &= \{O_1, \ldots, O_d\} \cup \{R_1 \cup \{e_2\} \cup R_2\},
\end{align*}
\]

where \( R_1 \cup \{ e_1 \} \cup R_2 \) and \( R_1 \cup \{ e_2 \} \cup R_2 \) are \((p_3, p_4)\)- and \((p_1, p_2)\)-paths, respectively. Define \( B_1 \) and \( B_2 \) by

\[
\begin{align*}
B_1 &= \begin{cases} 
\{(p_3, p_4)\} & \text{if } d = 0 \\
\{(p_3, z_1) \cup \{(w_i, z_{i+1}) \mid i = 1, \ldots, d-1\} \cup \{(w_d, p_4)\} & \text{if } d \geq 1
\end{cases} \\
B_2 &= \begin{cases} 
\{(p_2, p_1)\} & \text{if } d = 0 \\
\{(p_2, w_1) \cup \{(z_i, w_{i+1}) \mid i = 1, \ldots, d-1\} \cup \{(z_d, p_1)\} & \text{if } d \geq 1,
\end{cases}
\end{align*}
\]

as illustrated in Fig. 10. Similarly to the previous cases, we have (13) and (14). Furthermore, \( B_1 \cup B_2 \) consist of vertex-disjoint \((p_1, p_2)\)- and \((p_3, p_4)\)-paths if \( d \) is even, and vertex-disjoint \((p_1, p_3)\)- and \((p_2, p_4)\)-paths if \( d \) is odd.

**Case 4:** Let \( R_2 \) denote \((p_2, p_3)\)-path in \( \mathcal{P}(T_1 \cap T_2) \), and for some vertices \( q_1 \) and \( q_4 \), let \( R_1 \) and \( R_4 \) respectively denote \((p_1, q_1)\)- and \((p_4, q_4)\)-paths in \( \mathcal{P}(T_1 \cap T_2) \). Then, we have

\[
\begin{align*}
\mathcal{P}(T_1) &= \{O_1, \ldots, O_d\} \cup \{R_4, R_1 \cup \{e_1\} \cup R_2\} \\
\mathcal{P}(T_2) &= \{O_1, \ldots, O_d\} \cup \{R_1, R_4 \cup \{e_2\} \cup R_2\},
\end{align*}
\]
Case 2

Figure 9: Two edge sets $B_1$ and $B_2$ for Case 2 (as illustrated in Fig. 7).

Case 3

Figure 10: Two edge sets $B_1$ and $B_2$ for Case 3 (as illustrated in Fig. 7).
Figure 11: Two edge sets $B_1$ and $B_2$ for Case 4 (as illustrated in Fig. 7).

where $R_1 \cup \{e_1\} \cup R_2$ and $R_3 \cup \{e_2\} \cup R_2$ are $(q_1, p_3)$- and $(q_4, p_2)$-paths, respectively. Define $B_1$ and $B_2$ by

$$B_1 = \begin{cases} 
\{(p_3, q_4), (p_4, q_1)\} & \text{if } d = 0 \\
\{(p_3, z_i)\} \cup \{(w_i, z_{i+1}) \mid i = 1, \ldots, d - 1\} \cup \{(w_d, q_4), (p_4, q_1)\} & \text{if } d \geq 1
\end{cases}$$

$$B_2 = \begin{cases} 
\{(p_2, q_1), (p_1, q_4)\} & \text{if } d = 0 \\
\{(p_2, w_1)\} \cup \{(z_i, w_{i+1}) \mid i = 1, \ldots, d - 1\} \cup \{(z_d, q_1), (p_1, q_4)\} & \text{if } d \geq 1
\end{cases}$$

as illustrated in Fig. 11. Similarly to the previous cases, we have (13) and (14). Furthermore, $B_1 \cup B_2$ consist of vertex-disjoint $(p_1, p_3)$- and $(p_2, p_4)$-paths if $d$ is even, and vertex-disjoint $(p_1, p_2)$- and $(p_3, p_4)$-paths if $d$ is odd.

In summary, we have the following lemma.

**Lemma 5.** Let $B_1$ and $B_2$ be two edge sets defined as above. Then they satisfy (13) and (14), and $B_1 \cup B_2$ consists of (i) a $(p_1, p_4)$-path if $q_2 = q_3$, and either (ii) vertex-disjoint $(p_1, p_2)$- and $(p_3, p_4)$-paths or (iii) vertex-disjoint $(p_1, p_3)$- and $(p_2, p_4)$-paths if $q_2 \neq q_3$.

Figure 12 shows two edge sets $B_1$ and $B_2$ for path covers $T_1$ and $T_2$ in Fig. 3.

Furthermore, $A_i$ and $B_i$ ($i = 1, 2$) satisfy the following properties.

**Lemma 6.** Let $A_1$, $A_2$, $B_1$, and $B_2$ be defined as above. Then they are all pairwise disjoint, and $C = A_1 \cup A_2 \cup B_1 \cup B_2$ is a 2-factor, consisting of either one or two cycles. Furthermore, there exists a tour $H$ of $G$ such that $\ell(H) \geq \ell(C)$.

**Proof.** It is not difficult to see that $A_1$, $A_2$, $B_1$, and $B_2$ are pairwise disjoint. Lemmas 3, 4, and 5 imply that $C = A_1 \cup A_2 \cup B_1 \cup B_2$ is a 2-factor consisting of either one or two cycles. Thus if $C$ is a 2-factor, the latter statement of the lemma holds. Assume that $C$ consists of two cycles. In this case, we can see that two edges $(p_2, x_1)$ and $(p_3, y_1)$ belong to different cycles by (11). Let $H =$
We are now ready to describe our approximation algorithm.

**Algorithm TourEven**

**Input:** A complete graph $G = (V, E)$ with even $|V|$, and an edge length function $\ell : E \rightarrow \mathbb{R}_+$.  

**Output:** A tour $T_{\text{apx}}$ in $G$.

Compute minimum weighted 2-factor $S$ and 1-factor $T$ of $(G, \ell)$.

**if** $S$ is a tour **then**

$T_{\text{apx}} := S$.

**else**

$S_1, T_1, S_2, T_2 := \text{FourPathCovers}(S, T)$.

Compute edge sets $A_1, A_2, B_1, B_2$ defined in (11), (12), (16), (17) and (18).

$T := \{S_1 \cup A_1, S_2 \cup A_2, T_1 \cup B_1, T_2 \cup B_2\}$.

$T_{\text{apx}} := \arg\min_{T \in T} \ell(T)$.

**end if**

Outputs $T_{\text{apx}}$ and halt.

**Theorem 7.** For a complete graph $G = (V, E)$ with an even number of vertices and an edge length function $\ell : E \rightarrow \mathbb{R}_+$, Algorithm **TourEven** computes a $3/4$-differential approximate tour of $(G, \ell)$ in polynomial time.
Proof. We show that Algorithm TourEven outputs a 3/4-differential approximate tour $T_{\text{apx}}$ in polynomial time. If a minimum weighted 2-factor $S$ of $(G, \ell)$ computed in the algorithm is a tour, then clearly $T_{\text{apx}} = S$ is an optimal tour. On the other hand, if $S$ is not a tour, then we have

$$4\ell(T_{\text{apx}}) \leq \ell(S_1 \cup A_1) + \ell(S_2 \cup A_2) + \ell(T_1 \cup B_1) + \ell(T_2 \cup B_2)$$

$$= 2(\ell(S) + \ell(T)) + \ell(A_1 \cup A_2 \cup B_1 \cup B_2)$$

$$\leq 3\text{opt}(G, \ell) + \text{wor}(G, \ell),$$

where the first equality follows from Lemmas 4, 5, and 6, and the last inequality follows from Lemma 6, and $\ell(S) \leq \text{opt}(G, \ell)$, and $2\ell(T) \leq \text{opt}(G, \ell)$. Thus $T_{\text{apx}}$ is a 3/4-differential approximate tour. Note that minimum weighted 1- and 2-factors can be computed in polynomial time, and $A_i$ and $B_i$ ($i = 1, 2$) can be computed in polynomial time. Thus Algorithm TourEven is polynomial, which completes the proof. \hfill $\square$

Before concluding the section, let us remark that 3/4-differential approximability is known for graph with an even number of vertices [10]. Different from the algorithm in [10], ours is constructed in a uniform framework, which can further be extended to the odd case.

4 Approximation for odd instances

In this section, we construct an approximation algorithm for TSP with an odd number of vertices. Our algorithm is much more involved than the even case. It first guesses a path $P$ with three edges in an optimal tour, constructs eight path covers based on $P$, and extend each path cover to a tour in such a way that at least one of the eight tours guarantees 3/4-differential approximation ratio.

More precisely, for each path $P$ with three edges, say, $P = \{(v_1, v_2), (v_2, v_3), (v_3, v_4)\}$ with all $v_i$‘s distinct, let $S$ be a minimum weighted 2-factor among those containing $P$, let $T$ be a minimum weighted path cover among those satisfying $(v_1, v_2), (v_2, v_3) \in T$ and $V(T) = V \setminus \{v_2\}$, and let $T'$ be a minimum weighted path cover among those satisfying $(v_2, v_3), (v_3, v_4) \in T'$ and $V(T') = V \setminus \{v_3\}$. Assume that $S$ is not a tour, i.e., it contains at least two cycles, since otherwise, is optimal, and hence ensures 3/4-differential approximability if some optimal tour contains $P$. We note that $(S, T)$ and $(S, T')$ are both valid pairs of spanning 2-matchings. We apply Procedure FourPathCovers to them, but not arbitrarily. Let us specify two cycles $C^*$ and $C^{**}$ in $S$ such that $P \subseteq C^*$ and $P \cap C^{**} = \emptyset$. We define two vertices $v_0$ and $v_5$ in $V(C^*)$ such that $v_0 \neq v_2$, $v_5 \neq v_3$, and $(v_0, v_1), (v_4, v_5) \in C^*$. By definition $v_0 = v_4$ and $v_5 = v_1$ hold if $|C^*| = 4$. Furthermore, we define two edges $f$ and $f'$ in $C^{**}$ that satisfy the properties in the next lemma.

**Lemma 8.** Let $C^{**}$, $T$ and $T'$ be defined as above. Then there exist two edges $f \in C^{**} \setminus T$ and $f' \in C^{**} \setminus T'$ such that

(i) they have a common endpoint $q$, and

(ii) $T \cup \{f\}$ and $T' \cup \{f'\}$ are path covers.

Proof. If $C^{**} \setminus (T \cup T') \neq \emptyset$, then arbitrarily take an edge $f = f'$ in $C^{**} \setminus (T \cup T')$. It is not difficult to see that (i) and (ii) in the lemma are satisfied. On the other hand, if $C^{**} \setminus (T \cup T') = \emptyset$. Then $C^{**}$ is even and it is covered with two matchings $C^{**} \cap T$ and $C^{**} \cap T'$. This again implies the existence of two edges. \hfill $\square$

We note that $f$ and $f'$ in Lemma 8 might be identical, and (ii) in Lemma 8 implies that two pairs $(S \setminus \{f\}, T \cup \{f\})$ and $(S \setminus \{f'\}, T' \cup \{f'\})$ are valid. Figure 14 shows an example of $S, T, T', f$ and $f'$.

Our algorithm uses Procedure FourPathCovers for $(S, T)$ defined as above in such a way that edge $e_1 = f$ is chosen in the first round and two edges $e_1 = (v_3, v_4)$ and $e_2 = (v_0, v_1)$ are chosen in the last round. Similarly, our algorithm uses Procedure FourPathCovers for $(S, T')$ defined as above in such a way that edge $e_1 = f'$ is chosen in the first round and two edges $e_1 = (v_1, v_2)$ and
Let $e_2 = (v_4, v_5)$ are chosen in the last round. Let $S_1$, $T_1$, $S_2$, and $T_2$ be four path covers obtained by Procedure FourPathCover($S, T$), and let $S'_1$, $T'_1$, $S'_2$, and $T'_2$ be four path covers returned by Procedure FourPathCover($S, T'$).

**Lemma 9.** Let $S$, $T$, $S_i$, and $T_i$ ($i = 1, 2$) be defined as above. Then $S_1$, $S_2$, $T_1$, and $T_2$ are path covers such that

\[
S_i \cup T_i = S \cup T \quad \text{and} \quad S_i \cap T_i = S \cap T \quad \text{for} \quad i = 1, 2, \\
V_1(S_i) \text{ and } V_1(T_i) \text{ is a partition of } V \setminus \{v_2\} \quad \text{for} \quad i = 1, 2, \\
T_1 \setminus T_2 = \{(v_3, v_4)\}, T_2 \setminus T_1 = \{(v_0, v_1)\}, \text{ and } \{(v_1, v_2), (v_2, v_3)\} \in \mathcal{P}(T_1 \cap T_2), \text{ and} \\
q \in V_1(S_1) \cap V_1(S_2),
\]

where $v_i \in V(C^*)$ ($i = 0, \ldots, 4$) are defined as above and $q$ is a common endpoint of $f$ and $f'$ in Lemma 8.

**Proof.** By definition, we have $V_1(S) = \emptyset$ and $V_1(T) = V \setminus \{v_2\}$. Moreover, since an edge $f$ in Lemma 8 is chosen in the first round of Procedure FourPathCovers($S, T$), and $(v_3, v_4)$ and $(v_0, v_1)$ are chosen in the last round of Procedure FourPathCovers($S, T$), Lemma 3 implies the statement of lemma. \hfill \Box

Figure 15 shows $(S_1, T_1)$ and $(S_2, T_2)$ computed by Procedure FourPathCovers for $(S, T)$ in Fig. 14.

Similarly, we have the following lemma.

**Lemma 10.** Let $S$, $T'$, $S'_i$, and $T'_i$ ($i = 1, 2$) be defined as above. Then $S'_1$, $S'_2$, $T'_1$, and $T'_2$ are path covers such that

\[
S'_i \cup T'_i = S \cup T' \quad \text{and} \quad S'_i \cap T'_i = S \cap T' \quad \text{for} \quad i = 1, 2, \\
V_1(S'_i) \text{ and } V_1(T'_i) \text{ is a partition of } V \setminus \{v_3\} \quad \text{for} \quad i = 1, 2, \\
T'_1 \setminus T'_2 = \{(v_1, v_2)\}, T'_2 \setminus T'_1 = \{(v_4, v_5)\}, \text{ and } \{(v_2, v_3), (v_3, v_4)\} \in \mathcal{P}(T'_1 \cap T'_2), \text{ and} \\
q \in V_1(S'_1) \cap V_1(S'_2),
\]

where $v_i \in V(C^*)$ ($i = 1, \ldots, 5$) are defined as above and $q$ is a common endpoint of $f$ and $f'$ in Lemma 8.
Figure 15: Two pairs \((S_1, T_1)\) and \((S_2, T_2)\) computed by **Procedure FourPathCovers** for \((S, T)\), \(e_1^{(1)} = f\), \(e_1^{(2)} = e\), \(e_1^{(3)} = (v_3, v_4)\) and \(e_2^{(3)} = (v_0, v_1)\) in Fig. 14, where \(e_i^{(j)}\) denotes the edge chosen as \(e_i\) in the \(j\)-th round of the procedure.

Figure 16: Two pairs \((S_1', T_1')\) and \((S_2', T_2')\) computed by **Procedure FourPathCovers** for \((S, T')\), \(e_1^{(1)} = f'\), \(e_1^{(2)} = e'\), \(e_1^{(3)} = (v_1, v_2)\) and \(e_2^{(3)} = (v_4, v_5)\) in Fig. 14, where \(e_i^{(j)}\) denotes the edge chosen as \(e_i\) in the \(j\)-th round of the procedure.
The definition of $A$

Note that $A$ are separately described. Define $A$ 

Figure 17: Two cases $|C^*| > 4$ and $|C^*| = 4$ for path covers $S_1$ and $S_2$ returned by **Procedure** 

FourPathCovers$(S, T)$.

Figure 16 shows $(S'_1, T'_1)$ and $(S'_2, T'_2)$ computed by **Procedure** FourPathCovers for $(S, T')$ in Fig. 14.

Let us then show how to construct edge sets $A_i^{(l)}$ and $B_i^{(l)}$ (for $i = 1, 2$), such that $S_i^{(l)} \cup A_i^{(l)}$ and $T_i^{(l)} \cup B_i^{(l)}$ (for $i = 1, 2$) are tours and

$$\ell(A_1) + \ell(A_2) + \ell(B_1) + \ell(B_2) + \ell(A'_1) + \ell(A'_2) + \ell(B'_1) + \ell(B'_2) \leq 2 \text{wor}(G, \ell) - 2\ell(v_2, v_3),$$

where $\text{wor}(G, \ell)$ denotes the length of a longest tour of $(G, \ell)$.

Let us first show how to construct $A_1$ and $A_2$. By definition, $P(S_1) \setminus P(S_2)$ consists of a $(v_4, v_3)$-path $P_1 = C^* \setminus \{v_4, v_3\}$, and $P(S_2) \setminus P(S_1)$ consists of a $(v_1, v_0)$-path $P_2 = C^* \setminus \{v_1, v_0\}$. Let $Q_i$ ($i = 1, \ldots, k$) denote $(x_i, y_i)$-paths such that $\{Q_1, \ldots, Q_k\} = P(S_1) \cap P(S_2)$, where $x_i = q$ in Lemma 9. Figure 17 shows $S_1$ and $S_2$ computed by **Procedure** FourPathCovers$(S, T)$, where two cases $|C^*| > 4$ and $|C^*| = 4$ are separately described. Define $A_1$ and $A_2$ by

$$A_1 = \{(v_3, x_1) \cup \{(y_i, x_{i+1}) \mid i = 1, \ldots, k - 1 \} \cup \{(y_k, v_4)\}$$

$$A_2 = \{(v_1, y_1) \cup \{(x_i, y_{i+1}) \mid i = 1, \ldots, k - 1 \} \cup \{(x_k, v_0)\},$$

as illustrated in Fig. 18. Then we have the following lemma.

**Lemma 11.** Two sets $A_1$ and $A_2$ defined in (27) satisfy the following three conditions.

(i) $S_i \cup A_i$ is a tour of $G$ for $i = 1, 2$.

(ii) $V(A_i) = V_i(S_i)$ for $i = 1, 2$.

(iii) $A_1 \cap A_2 = \emptyset$ and $A_1 \cup A_2$ consists of

(iii-1) a $(v_1, v_3)$-path if $|C^*| = 4$.

(iii-2) vertex-disjoint $(v_0, v_3)$- and $(v_1, v_4)$-paths if $|C^*| > 4$ and $k$ is odd.

(iii-3) vertex-disjoint $(v_0, v_1)$- and $(v_3, v_4)$-paths if $|C^*| > 4$ and $k$ is even.

**Proof.** Note that $P(S_1) = \{Q_1, \ldots, Q_k\} \cup \{P_1\}$ and $P(S_2) = \{Q_1, \ldots, Q_k\} \cup \{P_2\}$. Thus it follows from the definition of $A_1$ and $A_2$. 

16
Figure 18: Two edge sets $A_1$ and $A_2$ for path covers $S_1$ and $S_2$ (as illustrated in Fig. 17).
Similarly, let us define $A'_1$ and $A'_2$. Recall that $\mathcal{P}(S'_i) \setminus \mathcal{P}(S'_j)$ consists of a $(v_1, v_2)$-path $P'_1 = C^* \setminus \{(v_1, v_2)\}$, and $\mathcal{P}(S'_2) \setminus \mathcal{P}(S'_i)$ consists of a $(v_4, v_5)$-path $P'_2 = C^* \setminus \{(v_4, v_5)\}$. Let $Q'_i (i = 1, \ldots, k)$ denote $(x'_i, y'_i)$-paths such that $\{Q'_1, \ldots, Q'_k\} = \mathcal{P}(S'_1) \cap \mathcal{P}(S'_2)$, where $x'_i = q$ in Lemma 9. Define $A'_1$ and $A'_2$ by

$$A'_1 = \{(v_2, x'_1)\} \cup \{(y'_i, x'_{i+1}) \mid i = 1, \ldots, k-1\} \cup \{(y'_k, v_1)\}$$

$$A'_2 = \{(v_4, y'_1)\} \cup \{(x'_i, y'_{i+1}) \mid i = 1, \ldots, k-1\} \cup \{(x'_k, v_5)\}. \tag{28}$$

Then we have the following lemma.

**Lemma 12.** Two sets $A'_1$ and $A'_2$ defined in (28) satisfy the following three conditions.

(i) $S'_i \cup A'_i$ is a tour of $G$ for $i = 1, 2$.

(ii) $V(A'_i) = V(S'_i)$ for $i = 1, 2$.

(iii) $A'_1 \cap A'_2 = \emptyset$ and $A'_1 \cup A'_2$ consists of

(iii-1) a $(v_2, v_4)$-path if $|C^*| = 4$.

(iii-2) vertex-disjoint $(v_1, v_4)$ and $(v_2, v_5)$-paths if $|C^*| > 4$ and $k$ is odd.

(iii-3) vertex-disjoint $(v_1, v_2)$- and $(v_3, v_5)$-paths if $|C^*| > 4$ and $k$ is even.

**Proof.** Note that $\mathcal{P}(S'_1) = \{Q'_1, \ldots, Q'_k\} \cup \{P'_1\}$ and $\mathcal{P}(S'_2) = \{Q'_1, \ldots, Q'_k\} \cup \{P'_2\}$. Thus it follows from the definition of $A'_1$ and $A'_2$. \hfill \square

Figures 19 and 20 show an example of edge sets $A_1$, $A_2$, $A'_1$, and $A'_2$ for path covers $S_1$, $S_2$, $S'_1$, and $S'_2$ in Figs. 15 and 16. Let us next construct $B_1$, $B_2$, $B'_1$, and $B'_2$. Let $O_i (i = 1, \ldots, d)$ denote vertex-disjoint $(z_i, w_i)$-paths such that $\{O_1, \ldots, O_d\} = \mathcal{P}(T_1) \cap \mathcal{P}(T_2)$, where $z_1$ and $w_1$ satisfy

$$\ell(v_1, z_1) + \ell(v_3, w_1) \leq \ell(v_1, w_1) + \ell(v_3, z_1). \tag{29}$$

We remark that $d \geq 1$ (i.e., $\mathcal{P}(T_1) \cap \mathcal{P}(T_2) \neq \emptyset$) holds if $n \geq 16$. To see this, we have $|\mathcal{P}(T_1)| = \lfloor n/2 \rfloor - (k + 1)$, where $k + 1$ is equal to the number of cycles in $S$. Since each cycle in $S$ has size at least 3, $k + 1 \geq \lfloor n/3 \rfloor$ holds, which implies that $|\mathcal{P}(T_1)| \geq 3$ if $n \geq 16$. Since $|\mathcal{P}(T_1)| \neq 2$, we have $d = \mathcal{P}(T_1) \cap \mathcal{P}(T_2) \geq 1$ if $n \geq 16$. In the subsequent discussion, we assume that $n \geq 16$, and construct $B_1$ and $B_2$ by considering the following three cases (see in Fig. 21).
respectively denote \( v \) and \( B \) as illustrated in Fig. 22. Then we have

1. \(|C^*| > 4\) and \(\mathcal{P}(T_1 \cap T_2)\) contains a \((v_0, v_4)\)-path.

2. \(|C^*| > 4\) and \(\mathcal{P}(T_1 \cap T_2)\) contains no \((v_0, v_4)\)-path.

3. \(|C^*| = 4\).

**Case 1:** Let \(R_0\) denote a \((v_0, v_4)\)-path in \(\mathcal{P}(T_1 \cap T_2)\), and let \(R_1 = \{(v_1, v_2), (v_2, v_3)\}\). By definition \(R_1\) is a \((v_1, v_3)\)-path in \(\mathcal{P}(T_1 \cap T_2)\). Then we note that

\[
\begin{align*}
\mathcal{P}(T_1) &= \{O_1, \ldots, O_d\} \cup \{R_0 \cup \{(v_3, v_4)\} \cup R_1\} \\
\mathcal{P}(T_2) &= \{O_1, \ldots, O_d\} \cup \{R_0 \cup \{(v_0, v_1)\} \cup R_1\},
\end{align*}
\]

where \(R_0 \cup \{(v_3, v_4)\} \cup R_1\) and \(R_0 \cup \{(v_0, v_1)\} \cup R_1\) are \((v_0, v_1)\)- and \((v_3, v_4)\)-paths, respectively. Define \(B_1\) and \(B_2\) by

\[
\begin{align*}
B_1 &= \{(v_1, z_1)\} \cup \{(w_i, z_{i+1}) \mid i = 1, \ldots, d - 1\} \cup \{(w_d, v_0)\} \\
B_2 &= \{(v_3, w_1)\} \cup \{(z_i, w_{i+1}) \mid i = 1, \ldots, d - 1\} \cup \{(z_d, v_4)\},
\end{align*}
\]

as illustrated in Fig. 22. Then we have

\[
\begin{align*}
T_i \cup B_i &\text{ is a tour of } G \text{ for } i = 1, 2, \\
B_1 \cap B_2 &= \emptyset \text{ and } V(B_i) = V_i(T_i) \text{ for } i = 1, 2, \text{ and} \\
B_1 \cup B_2 &\text{ consist of vertex-disjoint } (v_0, v_1)\text{- and } (v_3, v_4)\text{-paths if } d \text{ is even,} \\
&\text{and vertex-disjoint } (v_0, v_3)\text{- and } (v_1, v_4)\text{-paths if } d \text{ is odd.}
\end{align*}
\]

**Case 2:** Let \(R_1 = \{(v_1, v_2), (v_2, v_3)\}\) (i.e., let \(R_1\) be a \((v_1, v_3)\)-path in \(\mathcal{P}(T_1 \cap T_2)\)). Let \(R_0\) and \(R_4\) respectively denote \((v_0, r_0)\)- and \((v_4, r_4)\)-paths in \(\mathcal{P}(T_1 \cap T_2)\). Then, we have

\[
\begin{align*}
\mathcal{P}(T_1) &= \{O_1, \ldots, O_d\} \cup \{R_0, R_1 \cup \{(v_3, v_4)\} \cup R_4\} \\
\mathcal{P}(T_2) &= \{O_1, \ldots, O_d\} \cup \{R_4, R_0 \cup \{(v_0, v_1)\} \cup R_1\},
\end{align*}
\]
Figure 21: Three cases for path covers $T_1$ and $T_2$ returned by Procedure FourPathCovers($S, T$).
where $R_1 \cup \{(v_3, v_4)\} \cup R_4$ and $R_0 \cup \{(v_0, v_1)\} \cup R_1$ are $(v_1, r_4)$- and $(v_3, r_0)$-paths, respectively. Define $B_1$ and $B_2$ by

$$
B_1 = \{(v_1, z_1)\} \cup \{(w_i, z_{i+1}) \mid i = 1, \ldots, d - 1\} \cup \{(w_d, r_0), (v_0, r_4)\}
$$

$$
B_2 = \{(v_3, w_1)\} \cup \{(z_i, w_{i+1}) \mid i = 1, \ldots, d - 1\} \cup \{(z_d, r_4), (v_4, r_0)\},
$$

as illustrated in Fig. 23. Similarly to Case 1, we have (31) and (32). Furthermore, $B_1 \cup B_2$ consists of $(v_0, v_3)$- and $(v_1, v_4)$-paths if $d$ is even, and vertex-disjoint $(v_0, v_1)$- and $(v_3, v_4)$-paths if $d$ is odd.

**Case 3:** In this case, we have $v_0 = v_4$. Let $R_1 = \{(v_1, v_2), (v_2, v_3)\}$ (i.e., let be a $(v_1, v_3)$-path in $\mathcal{P}(T_1 \cap T_2)$), let $R_4$ denote $(v_4, r_4)$-path in $\mathcal{P}(T_1 \cap T_2)$. Then we have

$$
\mathcal{P}(T_1) = \{O_1, \ldots, O_d\} \cup R_1 \cup \{(v_3, v_4)\} \cup R_4
$$

$$
\mathcal{P}(T_2) = \{O_1, \ldots, O_d\} \cup R_1 \cup \{(v_1, v_4)\} \cup R_4,
$$

where $R_1 \cup \{(v_3, v_4)\} \cup R_4$ and $R_1 \cup \{(v_1, v_4)\} \cup R_4$ are $(v_1, r_4)$- and $(v_3, r_4)$-paths, respectively. Define $B_1$ and $B_2$ by

$$
B_1 = \{(v_1, z_1)\} \cup \{(w_i, z_{i+1}) \mid i = 1, \ldots, d - 1\} \cup \{(w_d, r_4)\}
$$

$$
B_2 = \{(v_3, w_1)\} \cup \{(z_i, w_{i+1}) \mid i = 1, \ldots, d - 1\} \cup \{(z_d, r_4)\},
$$

as illustrated in Fig. 24. Similarly to the previous cases, we have (31) and (32). Furthermore, we have $B_1 \cup B_2$ is a $(v_1, v_3)$-path.

In summary, we have the following lemma.

**Lemma 13.** Let $B_1$ and $B_2$ be edge sets defined as above. Then they satisfy (31) and (32), and $B_1 \cup B_2$ consists of either (i) vertex-disjoint $(v_0, v_3)$- and $(v_1, v_4)$-paths or (ii) vertex-disjoint $(v_0, v_1)$- and $(v_3, v_4)$-paths if $|C^*| > 4$, and (iii) a $(v_1, v_3)$-path if $|C^*| = 4$.

Similarly, $B'_1$ and $B'_2$ can be obtained from $T_1'$ and $T_2'$ as follows. Let $O'_i (i = 1, \ldots, d)$ denote vertex-disjoint $(z'_i, w'_i)$-paths such that $\{O'_1, \ldots, O'_d\} = \mathcal{P}(T'_1) \cap \mathcal{P}(T'_2)$, where $z'_i$ and $w'_i$ satisfy

$$
\ell(v_4, z'_1) + \ell(v_2, w'_1) \leq \ell(v_4, w'_1) + \ell(v_2, z'_1).
$$

Recall that $d \geq 1$ (i.e., $\mathcal{P}(T'_1) \cap \mathcal{P}(T'_2) \neq \emptyset$) holds if $n \geq 17$. We construct $B'_1$ and $B'_2$ by considering the following three cases.
Case 2

Figure 23: Two edge sets $B_1$ and $B_2$ for Case 2 (as illustrated in Fig. 21).

Case 3

Figure 24: Two edge sets $B_1$ and $B_2$ for Case 3 (as illustrated in Fig. 21).
Similarly to the previous cases, we have (38) and (39). Furthermore, we have (40) and (41). Let $R'_1$ denote a $(v_1, v_5)$-path in $P(T'_1 \cap T'_2)$, and let $R'_2 = \{(v_2, v_3), (v_3, v_4)\}$. By definition $R'_2$ is a $(v_2, v_4)$-path in $P(T'_1 \cap T'_2)$. Then we note that
\[
P(T'_1) = \{O_1', \ldots, O_d'\} \cup \{R'_1 \cup \{(v_1, v_2)\} \cup R'_2\}
\]
\[
P(T'_2) = \{O_1', \ldots, O_d'\} \cup \{R'_1 \cup \{(v_4, v_5)\} \cup R'_2\},
\]
where $R'_1 \cup \{(v_1, v_2)\} \cup R'_2$ and $R'_1 \cup \{(v_4, v_5)\} \cup R'_2$ are $(v_3, v_4)$- and $(v_1, v_2)$-paths, respectively. Define $B'_1$ and $B'_2$ by
\[
B'_1 = \{(v_4, z'_1)\} \cup \{(w'_i, z'_{i+1}) \mid i = 1, \ldots, d - 1\} \cup \{(w'_d, v_5)\}
\]
\[
B'_2 = \{(v_2, w'_1)\} \cup \{(z'_i, w'_{i+1}) \mid i = 1, \ldots, d - 1\} \cup \{(z'_d, v_1)\}.
\]
Then we have
\[\text{If } d \text{ is even, and vertex-disjoint } (v_1, v_4) \text{- and } (v_2, v_5) \text{-paths if } d \text{ is odd. (40)}\]

Case 2: Let $R'_2 = \{(v_2, v_3), (v_3, v_4)\}$ (i.e., let be a $(v_2, v_4)$-path in $P(T'_1 \cap T'_2)$). Let $R'_1$ and $R'_5$ respectively denote $(v_1, r'_1)$- and $(v_5, r'_5)$-paths in $P(T'_1 \cap T'_2)$. Then, we have
\[
P(T'_1) = \{O_1', \ldots, O_d'\} \cup \{R'_1 \cup \{(v_1, v_2)\} \cup R'_2\}
\]
\[
P(T'_2) = \{O_1', \ldots, O_d'\} \cup \{R'_1 \cup \{(v_5, v_6)\} \cup R'_2\}
\]
where $R'_1 \cup \{(v_1, v_2)\} \cup R'_2$ and $R'_1 \cup \{(v_5, v_6)\} \cup R'_2$ are $(v_3, v_4)$- and $(v_1, v_2)$-paths, respectively. Define $B'_1$ and $B'_2$ by
\[
B'_1 = \{(v_4, z'_1)\} \cup \{(w'_i, z'_{i+1}) \mid i = 1, \ldots, d - 1\} \cup \{(w'_d, v_5), (v_5, r'_1)\}
\]
\[
B'_2 = \{(v_2, w'_1)\} \cup \{(z'_i, w'_{i+1}) \mid i = 1, \ldots, d - 1\} \cup \{(z'_d, r'_1), (v_5, r'_5)\}.
\]
Similarly to Case 1, we have (38) and (39). Furthermore, $B'_1 \cup B'_2$ consists of $(v_1, v_4)$- and $(v_5, v_5)$-paths if $d$ is even, and vertex-disjoint $(v_1, v_2)$- and $(v_4, v_5)$-paths if $d$ is odd.

Case 3: In this case, we have $v_5 = v_1$. Let $R'_2 = \{(v_2, v_3), (v_3, v_4)\}$ (i.e., let $R_1$ be a $(v_2, v_4)$-path in $P(T'_1 \cap T'_2)$), let $R'_1$ denote $(v_1, r'_1)$-path in $P(T'_1 \cap T'_2)$. Then we have
\[
P(T'_1) = \{O_1', \ldots, O_d'\} \cup \{R'_1 \cup \{(v_1, v_2)\} \cup R'_2\}
\]
\[
P(T'_2) = \{O_1', \ldots, O_d'\} \cup \{R'_1 \cup \{(v_1, v_4)\} \cup R'_2\}
\]
where $R'_1 \cup \{(v_1, v_2)\} \cup R'_2$ and $R'_1 \cup \{(v_1, v_4)\} \cup R'_2$ are $(v_3, v_4)$- and $(v_1, v_2)$-paths, respectively. Define $B'_1$ and $B'_2$ by
\[
B'_1 = \{(v_4, z'_1)\} \cup \{(w'_i, z'_{i+1}) \mid i = 1, \ldots, d - 1\} \cup \{(w'_d, r'_1)\}
\]
\[
B'_2 = \{(v_2, w'_1)\} \cup \{(z'_i, w'_{i+1}) \mid i = 1, \ldots, d - 1\} \cup \{(z'_d, r'_1)\}.
\]
Similarly to the previous cases, we have (38) and (39). Furthermore, we have $B'_1 \cup B'_2$ is a $(v_2, v_4)$-path. In summary, we have the following lemma.
Figure 25: Two edge sets $B_1$ and $B_2$ for path covers $T_1$ and $T_2$ in Fig. 15.

Figure 26: Two edge sets $B'_1$ and $B'_2$ for path covers $T'_1$ and $T'_2$ in Fig. 16.
Lemma 14. Let $B'_1$ and $B'_2$ be edge sets defined as above. Then they satisfy (38) and (39), and $B'_1 \cup B'_2$ consists of either (i) vertex-disjoint $(v_1, v_4)$- and $(v_2, v_5)$-paths or (ii) vertex-disjoint $(v_1, v_2)$- and $(v_4, v_5)$-paths if $|C^*| > 4$, and (iii) a $(v_2, v_4)$-path if $|C^*| = 4$.

Figures 25 and 26 show an example of edge sets $B_1, B_2, B'_1$, and $B'_2$ for path covers $T_1, T_2, T'_1$, and $T'_2$ in Figs. 15 and 16. Furthermore, $A^{(i)}$ and $B^{(i)} (i = 1, 2)$ satisfy the following properties.

Lemma 15. Let $A_1, A_2, B_1$, and $B_2$ be defined as above. Then they are all pairwise disjoint, and $C = A_1 \cup A_2 \cup B_1 \cup B_2$ consists of either one or two cycles such that $V(C) = V \setminus \{v_2\}$. Furthermore, there exists a cycle $D$ such that $V(D) = V \setminus \{v_2\}$, $\ell(D) \geq \ell(C)$ and $(q, v_3) \in D$.

Proof. It is not difficult to see that $A_1, A_2, B_1$, and $B_2$ are pairwise disjoint. Lemmas 9, 11, and 13 imply that $C = A_1 \cup A_2 \cup B_1 \cup B_2$ consists of either one or two cycles such that $V(C) = V \setminus \{v_2\}$. By $x_1 = q$ and (27), we have $(q, v_3) \in C$. Thus if $C$ is a single cycle, the latter statement in the lemma holds. Assume that $C$ consists of two cycles. In this case, we can see that two edges $(v_1, z_1)$ and $(v_3, w_1)$ belong to different cycles by (30), (34), and (35). Let $D = (C \setminus \{(v_1, z_1), (v_3, w_1)\}) \cup \{(v_1, w_1), (v_3, z_1)\}$. Then $D$ is a cycle such that $V(D) = V \setminus \{v_2\}$. By assumption (29), we have $\ell(D) \geq \ell(C)$. Since $C$ contains $(q, v_3)$, so does $D$, which completes the proof.

Lemma 16. Let $A'_1, A'_2, B'_1$, and $B'_2$ be defined as above. Then they are all pairwise disjoint, and $C' = A'_1 \cup A'_2 \cup B'_1 \cup B'_2$ consists of either one or two cycles such that $V(C') = V \setminus \{v_3\}$. Furthermore, there exists a cycle $D'$ such that $V(D') = V \setminus \{v_3\}$, $\ell(D') \geq \ell(C')$ and $(q, v_2) \in D'$.

Proof. It is not difficult to see that $A'_1, A'_2, B'_1$, and $B'_2$ are pairwise disjoint. Lemmas 10, 12, and 14 imply that $C' = A'_1 \cup A'_2 \cup B'_1 \cup B'_2$ consists of either one or two cycles such that $V(C') = V \setminus \{v_3\}$. By $x'_1 = q$ and (28), we have $(q, v_2) \in C'$. Thus if $C'$ is a single cycle, the latter statement in the lemma holds. Assume that $C'$ consists of two cycles. In this case, we can see that two edges $(v_4, z'_1)$ and $(v_2, w'_1)$ belong to different cycles by (37), (41), and (42). Let $D' = (C' \setminus \{(v_4, z'_1), (v_2, w'_1)\}) \cup \{(v_4, w'_1), (v_2, z'_1)\}$. Then $D'$ is a cycle such that $V(D') = V \setminus \{v_3\}$. By assumption (36), we have $\ell(D') \geq \ell(C')$. Since $C'$ contains $(q, v_2)$, so does $D'$, which completes the proof.

Lemma 17. Let $A^{(i)}_1$ and $B^{(i)}_1$ for $i = 1, 2$ be defined as above. Then there exist two tours $H$ and $H'$ in $G$ such that $\ell(H) + \ell(H') \geq \ell(A_1) + \ell(A_2) + \ell(B_1) + \ell(B_2) + \ell(A'_1) + \ell(A'_2) + \ell(B'_1) + \ell(B'_2) + 2\ell(v_2, v_3)$.

Proof. Let $D$ and $D'$ be a cycles in Lemmas 15 and 16, respectively. Then we have $V(D) = V \setminus \{v_2\}$, $V(D') = V \setminus \{v_3\}$, $(q, v_3) \in D$, and $(q, v_2) \in D'$. Define $H$ and $H'$ by

$$
H = (D \setminus \{(q, v_3)\}) \cup \{(q, v_2), (v_2, v_3)\}
$$

$$
H' = (D' \setminus \{(q, v_2)\}) \cup \{(q, v_3), (v_3, v_2)\}.
$$

Then $H$ and $H'$ are tours. Furthermore, we have

$$
\ell(H) + \ell(H') = \ell(D) + \ell(D') + 2\ell(v_2, v_3)
\geq \ell(A_1 \cup A_2 \cup B_1 \cup B_2) + \ell(A'_1 \cup A'_2 \cup B'_1 \cup B'_2) + 2\ell(v_2, v_3),
$$

which completes the proof.

We are now ready to describe our approximation algorithm, called TourOdd. Before analysis of $T_{\text{approx}}$, let us evaluate $\ell(S), \ell(T)$ and $\ell(T')$.

Lemma 18. For a path $P = \{(v_1, v_2), (v_2, v_3), (v_3, v_4)\}$, let $S, T$ and $T'$ be defined as above. If there exists an optimal tour that contains $P$, then

$$
2\ell(S) + \ell(T) + \ell(T') \leq 3 \text{opt}(G, \ell) + \ell(v_2, v_3).
$$

(43)
Algorithm TourOdd

Input: A complete graph $G = (V, E)$ with odd $|V|$, and an edge length function $\ell : E \to \mathbb{R}_+$.  
Output: A tour $T_{\text{apx}}$ in $G$.  

if $n < 17$ then  
    Compute an optimal tour $T_{\text{opt}}$ of $(G, \ell)$ by exhaustive search.  
    Output $T_{\text{opt}}$ and halt.  
else  
    $T := \emptyset$.  
    for $v_1, v_2, v_3$ and $v_4$ in the 4-permutations of $V$ do  
        Compute a minimum weighted 2-factor $S$ among those containing $\{(v_1, v_2), (v_2, v_3), (v_3, v_4)\}$.  
        Compute a minimum weighted path cover $T$ among those satisfying $(v_1, v_2), (v_2, v_3) \in T$ and $V_1(T) = V \setminus \{v_2\}$.  
        Compute a minimum weighted path cover $T'$ among those satisfying $(v_2, v_3), (v_3, v_4) \in T'$ and $V_1(T') = V \setminus \{v_3\}$.  
        if $S$ is a tour then  
            $T := T \cup \{S\}$.  
        else  
            $S_1, T_1, S_2, T_2 := \text{FourPathCovers}(S, T)$.  
            Compute edge sets $A_1, A_2, B_1, B_2$ defined in (27), (30), (34), and (35).  
            $\mathcal{T} := \mathcal{T} \cup \{S_1 \cup A_1, S_2 \cup A_2, T_1 \cup B_1, T_2 \cup B_2\}$.  
            $S_1', T_1', S_2', T_2' := \text{FourPathCovers}(S', T')$.  
            Compute edge sets $A_1', A_2', B_1', B_2'$ defined in (28), (37), (41), and (42).  
            $\mathcal{T} := \mathcal{T} \cup \{S_1' \cup A_1', S_2' \cup A_2', T_1' \cup B_1', T_2' \cup B_2'\}$.  
        end if  
    end for  
    $T_{\text{apx}} := \arg\min_{T \in \mathcal{T}} \ell(T)$.  
    Output $T_{\text{apx}}$ and halt.  
end if

Proof. Obviously $\ell(S) \leq \text{opt}(G, \ell)$ by definition. Let $T_{\text{opt}} = \{(v_i, v_{i+1}) | i = 1, \ldots, n\}$ be an optimal tour of $(G, \ell)$, where $v_{n+1} = v_1$, and let $U$ and $U'$ be two path covers defined by  

$U = \{(v_i, v_{i+1}) | i = 2, 3, \ldots, n - 1\} \cup \{(v_1, v_2)\}$  
$U' = \{(v_i, v_{i+1}) | i = 3, 4, \ldots, n\} \cup \{(v_2, v_3)\}$.

Then by the definition of $T$ and $T'$, $\ell(T) \leq \ell(U)$ and $\ell(T') \leq \ell(U')$. Therefore, we have  

$\ell(T) + \ell(T') \leq \ell(U) + \ell(U') = \text{opt}(G, \ell) + \ell(v_2, v_3)$.

Theorem 19. For a complete graph $G = (V, E)$ with an odd number of vertices and an edge length function $\ell : E \to \mathbb{R}_+$, Algorithm TourOdd computes a 3/4-differential approximate tour of $(G, \ell)$ in polynomial time.

Proof. If $n < 17$, Algorithm TourOdd clearly outputs an optimal tour in constant time. Otherwise (i.e., $n \geq 17$), let $T_{\text{opt}}$ be an optimal tour of $(G, \ell)$ and let $P = \{(v_1, v_2), (v_2, v_3), (v_3, v_4)\}$ be a path contained in $T_{\text{opt}}$. For this $P$, let $S, T,$ and $T'$ be defined as above. If $S$ is a tour, then $S$ is an optimal tour of $(G, \ell)$ and is output by the algorithm, which guarantees the statement of the theorem. On the other
hand, if $S$ is not a tour, then we have

$$8\ell(T_{\text{apx}}) \leq \ell(S_1 \cup A_1) + \ell(S_2 \cup A_2) + \ell(T_1 \cup B_1) + \ell(T_2 \cup B_2)
+ \ell(S'_1 \cup A'_1) + \ell(S'_2 \cup A'_2) + \ell(T'_1 \cup B'_1) + \ell(T'_2 \cup B'_2)
= 2(2\ell(S) + \ell(T) + \ell(T')) + \ell(A_1 \cup A_2 \cup B_1 \cup B_2) + \ell(A'_1 \cup A'_2 \cup B'_1 \cup B'_2)
\leq 6\text{opt}(G, \ell) + 2\text{wor}(G, \ell),$$

where the first equality follows from Lemmas 11, 12, 13, and 14, and the last inequality follows from Lemmas 17 and 18. Thus $T_{\text{apx}}$ is a 3/4-differential approximate tour of $(G, \ell)$. Note that $S$, $T$, and $T'$ can be computed in polynomial time, since minimum weighted 1- and 2-factors can be computed in polynomial time. Furthermore, $A_i^{(l)}$ and $B_i^{(l)}$ for $i = 1, 2$ can be computed in polynomial time. Thus Algorithm TourOdd is polynomial, which completes the proof.

Acknowledgement

This work was partially supported by the joint project of Kyoto University and Toyota Motor Corporation, titled ”Advanced Mathematical Science for Mobility Society” and by KAKENHI.

References

[1] Sanjeev Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998.
[2] Giorgio Ausiello, Cristina Bazgan, Marc Demange, and Vangelis Th Paschos. Completeness in differential approximation classes. *International Journal of Foundations of Computer Science*, 16(06):1267–1295, 2005.
[3] Richard Bellman. Dynamic programming treatment of the travelling salesman problem. *Journal of the ACM (JACM)*, 9(1):61–63, 1962.
[4] Robert G Bland and David F Shallcross. Large travelling salesman problems arising from experiments in x-ray crystallography: a preliminary report on computation. *Operations Research Letters*, 8(3):125–128, 1989.
[5] Miroslav Chlebík and Janka Chlebíková. Approximation hardness of travelling salesman via weighted amplifiers. In *International Computing and Combinatorics Conference*, pages 115–127. Springer, 2019.
[6] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group, 1976.
[7] William J Cook. *In pursuit of the traveling salesman: mathematics at the limits of computation*. Princeton University Press, 2011.
[8] Marc Demange and Vangelis Th Paschos. On an approximation measure founded on the links between optimization and polynomial approximation theory. *Theoretical Computer Science*, 158(1-2):117–141, 1996.
[9] Lars Engebretsen and Jonas Holmerin. Clique is hard to approximate within $n^{1-O(1)}$. In *International Colloquium on Automata, Languages, and Programming*, pages 2–12. Springer, 2000.
[10] Bruno Escoffier and Jérôme Monnot. A better differential approximation ratio for symmetric TSP. *Theoretical Computer Science*, 396(1-3):63–70, 2008.
[11] Martin Grötschel, Michael Jünger, and Gerhard Reinelt. Optimal control of plotting and drilling machines: a case study. *Zeitschrift für Operations Research*, 35(1):61–84, 1991.

[12] Refael Hassin and Samir Khuller. $z$-approximations. *Journal of Algorithms*, 41(2):429–442, 2001.

[13] Michael Held and Richard M Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied mathematics*, 10(1):196–210, 1962.

[14] Shen Lin and Brian W Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, 21(2):498–516, 1973.

[15] John DC Little, Katta G Murty, Dura W Sweeney, and Caroline Karel. An algorithm for the traveling salesman problem. *Operations research*, 11(6):972–989, 1963.

[16] Jérôme Monnot. Differential approximation results for the traveling salesman and related problems. *Information Processing Letters*, 82(5):229–235, 2002.

[17] Jérôme Monnot, Vangelis Th Paschos, and Sophie Toulouse. Approximation algorithms for the traveling salesman problem. *Mathematical methods of operations research*, 56(3):387–405, 2003.

[18] Jérôme Monnot, Vangelis Th Paschos, and Sophie Toulouse. Differential approximation results for the traveling salesman problem with distances 1 and 2. *European Journal of Operational Research*, 145(3):557–568, 2003.

[19] Jérôme Monnot and Sophie Toulouse. The traveling salesman problem and its variations. *Paradigms of Combinatorial Optimization: Problems and New Approaches*, pages 173–214, 2014.

[20] Christos H Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998.

[21] Abraham P Punnen. The traveling salesman problem: Applications, formulations and variations. In *The traveling salesman problem and its variations*, pages 1–28. Springer, 2007.

[22] DB Shmoys, JK Lenstra, AHG Rinnooy Kan, and EL Lawler. *The traveling salesman problem*, volume 12. Wiley, 1985.