Improved Approximation Algorithms for Metric Max TSP

Zhi-Zhong Chen *   Takayuki Nagoya †

Abstract

We present two polynomial-time approximation algorithms for the metric case of the maximum traveling salesman problem. One of them is for directed graphs and its approximation ratio is $\frac{27}{35}$. The other is for undirected graphs and its approximation ratio is $\frac{7}{8} - o(1)$. Both algorithms improve on the previous bests.

1 Introduction

The *maximum traveling salesman problem* (MaxTSP) is to compute a maximum-weight Hamiltonian circuit (called a *tour*) in a given complete edge-weighted (undirected or directed) graph. Usually, MaxTSP is divided into the *symmetric* and the *asymmetric* cases. In the symmetric case, the input graph is undirected; we denote this case by SymMaxTSP. In the asymmetric case, the input graph is directed; we denote this case by AsymMaxTSP. Note that SymMaxTSP can be trivially reduced to AsymMaxTSP.

A natural constraint one can put on AsymMaxTSP and SymMaxTSP is the *triangle inequality* which requires that for every set of three vertices u_1, u_2, and u_3 in the input graph G, $w(u_1, u_2) \leq w(u_1, u_3) + w(u_3, u_2)$, where $w(u_i, u_j)$ is the weight of the edge from u_i to u_j in G. If we put this constraint on AsymMaxTSP, we obtain a problem called *metric* AsymMaxTSP. Similary, if we put this constraint on SymMaxTSP, we obtain a problem called *metric* SymMaxTSP.

Both metric SymMaxTSP and metric AsymMaxTSP are Max-SNP-hard [1] and there have been a number of approximation algorithms known for them [7, 4, 5]. In 1985, Kostochka and Serdyukov [7] gave an $O(n^3)$-time approximation algorithm for metric SymMaxTSP that achieves an approximation ratio of $\frac{5}{6}$. Their algorithm is very simple and elegant. Tempted by improving the ratio $\frac{5}{6}$, Hassin and Rubinstein [4] gave a randomized $O(n^3)$-time approximation algorithm for metric SymMaxTSP whose *expected* approximation ratio is $\frac{7}{8} - o(1)$. This randomized algorithm was recently (partially) derandomized by Chen *et al.* [3]; their result is a (deterministic) $O(n^3)$-time approximation algorithm for metric SymMaxTSP whose approximation ratio is $\frac{17}{20} - o(1)$. In this paper, we completely derandomize the randomized algorithm, i.e., we obtain a (deterministic) $O(n^3)$-time approximation algorithm for metric SymMaxTSP whose approximation ratio is $\frac{7}{8} - o(1)$. Our algorithm also has the advantage of being easy to parallelize. Our derandomization is based on the idea of Chen *et al.* [3] and newly discovered properties of a folklore partition of the edges of a $2n$-vertex complete undirected graph into $2n - 1$ perfect matchings. These properties may be useful elsewhere. In particular, one of the properties says that if $G = (V, E)$ is a $2n$-vertex complete undirected graph and M is a perfect matching of G, then we can partition $E - M$ into $2n - 2$ perfect matchings $M_1, \ldots, M_{2n-2}$ among which there are at most $k^2 - k$ perfect matchings M_i such that the graph $(V, M \cup M_i)$ has a cycle of length at most $2k$ for every natural number k. This property is interesting because Hassin and Rubinstein [4] prove that if G and M are as before and M' is a random perfect matching of G, then with probability $1 - o(1)$ the multigraph $(V, M \cup M')$ has no cycle of length at most $\sqrt{n}$. Our result shows that instead

*Supported in part by the Grant-in-Aid for Scientific Research of the Ministry of Education, Science, Sports and Culture of Japan, under Grant No. 14580390. Department of Mathematical Sciences, Tokyo Denki University, Hatoyama, Saitama 350-0394, Japan. Email: chen@r.dendai.ac.jp.

†Department of Mathematical Sciences, Tokyo Denki University, Hatoyama, Saitama 350-0394, Japan. Email: nagoya@r.dendai.ac.jp.

of sampling from the set of all perfect matchings of G, it suffices to sample from $M_1, \ldots, M_{2n-2}$. This enables us to completely derandomize their algorithm.

As for metric AsymMaxTSP, Kostochka and Serdyukov [7] gave an $O(n^3)$-time approximation algorithm that achieves an approximation ratio of $\frac{3}{4}$. Their result remained the best in two decades until Kaplan *et al.* [5] gave a polynomial-time approximation algorithm whose approximation ratio is $\frac{10}{13}$. The key in their algorithm is a polynomial-time algorithm for computing two cycle covers $\mathcal{C}_1$ and $\mathcal{C}_2$ in the input graph G such that $\mathcal{C}_1$ and $\mathcal{C}_2$ do not share a 2-cycle and the sum of their weights is at least twice the optimal weight of a tour of G. They then observe that the multigraph formed by the edges in 2-cycles in $\mathcal{C}_1$ and $\mathcal{C}_2$ can be split into two subtours of G. In this paper, we show that the multigraph formed by the edges in 2-cycles in $\mathcal{C}_1$ and $\mathcal{C}_2$ *together with* a constant fraction of the edges in non-2-cycles in $\mathcal{C}_1$ and $\mathcal{C}_2$ can be split into two subtours of G. This enables us to improve Kaplan *et al.*'s algorithm to a polynomial-time approximation algorithm whose approximation ratio is $\frac{27}{35}$.

2 Basic Definitions

Throughout this paper, a *graph* means a simple undirected or directed graph (i.e., it has neither multiple edges nor self-loops), while a multigraph may have multiple edges but no self-loops.

Let G be a multigraph. We denote the vertex set of G by $V(G)$, and denote the edge set of G by $E(G)$. For a subset F of $E(G)$, $G - F$ denotes the graph obtained from G by deleting the edges in F. Two edges of G are *adjacent* if they share an endpoint.

Suppose G is undirected. The *degree* of a vertex v in G is the number of edges incident to v in G. A *cycle* in G is a connected subgraph of G in which each vertex is of degree 2. A *cycle cover* of G is a subgraph H of G with $V(H) = V(G)$ in which each vertex is of degree 2. A *matching* of G is a (possibly empty) set of pairwise nonadjacent edges of G. A *perfect matching* of G is a matching M of G such that each vertex of G is an endpoint of an edge in M.

Suppose G is directed. The *indegree* of a vertex v in G is the number of edges entering v in G, and the *outdegree* of v in G is the number of edges leaving v in G. A *cycle* in G is a connected subgraph of G in which each vertex has indegree 1 and outdegree 1. A *cycle cover* of G is a subgraph H of G with $V(H) = V(G)$ in which each vertex has indegree 1 and outdegree 1. A *2-path-coloring* of G is a partition of $E(G)$ into two subsets E_1 and E_2 such that both graphs $(V(G), E_1)$ and $(V(G), E_2)$ are collections of vertex-disjoint paths. G is *2-path-colorable* if it has a 2-path-coloring.

Suppose G is undirected or directed. A *path* in G is either a single vertex of G or a subgraph of G that can be transformed to a cycle by adding a single (new) edge. The *length* of a cycle or path C is the number of edges in C. A *k-cycle* is a cycle of length k. A *3^+-cycle* is a cycle of length at least 3. A *tour* (also called a *Hamiltonian cycle*) of G is a cycle C of G with $V(C) = V(G)$. A *subtour* of G is a subgraph H of G which is a collection of vertex-disjoint paths.

A *closed chain* is a directed graph that can be obtained from an undirected k-cycle C with $k \geq 3$ by replacing each edge $\{u, v\}$ of C with the two directed edges (u, v) and (v, u). Similarly, an *open chain* is a directed graph that can be obtained from an undirected path P by replacing each edge $\{u, v\}$ of P with the two directed edges (u, v) and (v, u). An open chain is *trivial* if it is a single vertex. A *chain* is a closed or open chain. A *partial chain* is a subgraph of a chain.

For a graph G and a weighting function w mapping each edge e of G to a nonnegative real number $w(e)$, the *weight* of a subset F of $E(G)$ is $w(F) = \sum_{e \in F} w(e)$, and the *weight* of a subgraph H of G is $w(H) = w(E(H))$.

3 New Algorithm for Metric AsymMaxTSP

Throughout this section, fix an instance (G, w) of metric AsymMaxTSP, where G is a complete directed graph and w is a function mapping each edge e of G to a nonnegative real number $w(e)$.

Let OPT be the weight of a maximum-weight tour in G. Our goal is to compute a tour in G whose weight is large compared to OPT. We first review Kaplan *et al.*'s algorithm and define several notations on the way.

3.1 Kaplan et al.'s Algorithm

The key in their algorithm is the following:

Theorem 3.1 [5] *We can compute two cycle covers $\mathcal{C}_1$, $\mathcal{C}_2$ in G in polynomial time that satisfy the following two conditions:*

1. *$\mathcal{C}_1$ and $\mathcal{C}_2$ do not share a 2-cycle. In other words, if C is a 2-cycle in $\mathcal{C}_1$ (respectively, $\mathcal{C}_2$), then $\mathcal{C}_2$ (respectively, $\mathcal{C}_1$) does not contain at least one edge of C.*
2. *$w(\mathcal{C}_1) + w(\mathcal{C}_2) \geq 2 \cdot OPT$.*

Let G_2 be the subgraph of G such that $V(G_2) = V(G)$ and $E(G_2)$ consists of all edges in 2-cycles in $\mathcal{C}_1$ and/or $\mathcal{C}_2$. Then, G_2 is a collection of vertex-disjoint chains. For each closed chain C in G_2, we can compute two edge-disjoint tours T_1 and T_2 (each of which is of length at least 3), modify $\mathcal{C}_1$ by substituting T_1 for the 2-cycles shared by C and $\mathcal{C}_1$, modify $\mathcal{C}_2$ by substituting T_2 for the 2-cycles shared by C and $\mathcal{C}_2$, and further delete C from G_2. After this modification of $\mathcal{C}_1$ and $\mathcal{C}_2$, the two conditions in Theorem 3.1 still hold. So, we can assume that there is no closed chain in G_2.

For each $i \in \{1, 2\}$, let $W_{i,2}$ denote the total weight of 2-cycles in $\mathcal{C}_i$, and let $W_{i,3} = w(\mathcal{C}_i) - W_{i,2}$. For convenience, let $W_2 = \frac{1}{2}(W_{1,2} + W_{2,2})$ and $W_3 = \frac{1}{2}(W_{1,3} + W_{2,3})$. Then, by Condition 2 in Theorem 3.1, we have $W_2 + W_3 \geq OPT$. Moreover, using an idea in [7], Kaplan *et al.* observed the following:

Lemma 3.2 [5] *We can use $\mathcal{C}_1$ and $\mathcal{C}_2$ to compute a tour T of G with $w(T) \geq \frac{3}{4}W_2 + \frac{5}{6}W_3$ in polynomial time.*

Since each nontrivial open chain has a 2-path-coloring, we can use G_2 to compute a tour T' of G with $w(T') \geq W_2$ in polynomial time. Combining this observation, Lemma 3.2, and the fact that $W_2 + W_3 \geq OPT$, the heavier one between T and T' is of weight at least $\frac{10}{13}OPT$.

3.2 Details of the New Algorithm

The idea behind our new algorithm is to improve the second tour T' in Kaplan *et al.*'s algorithm so that it has weight at least $W_2 + \frac{1}{9}W_3$. The tactics is to add some edges of 3^+-cycles in $\mathcal{C}_i$ with $W_{i,3} = \max\{W_{1,3}, W_{2,3}\}$ to G_2 so that G_2 remains 2-path-colorable. Without loss of generality, we may assume that $W_{1,3} \geq W_{2,3}$. Then, our goal is to add some edges of 3^+-cycles in $\mathcal{C}_1$ to G_2 so that G_2 remains 2-path-colorable.

We say that an open chain P in G_2 *spoils* an edge (u, v) of a 3^+-cycle in $\mathcal{C}_1$ if u and v are the two endpoints of P. Obviously, adding a spoiled edge to G_2 destroys the 2-path-colorability of G_2. Fortunately, there is no 3^+-cycle in $\mathcal{C}_1$ in which two consecutive edges are both spoiled. So, let C_1, $\ldots$, C_ℓ be the 3^+-cycles in $\mathcal{C}_1$; we modify each C_j $(1 \leq j \leq \ell)$ as follows (see Figure 1):

- For every two consecutive edges (u, v) and (v, x) of C_j such that (u, v) is spoiled, replace (u, v) by the two edges (u, x) and (x, v). (*Comment*: We call (u, x) a *bypass edge* of C_j, call the 2-cycle between v and x a *dangling 2-cycle* of C_j, and call v the *articulation vertex* of the dangling 2-cycle. We also say that the bypass edge (u, x) and the dangling 2-cycle between v and x *correspond* to each other.)

We call the above modification of C_j the *bypass operation* on C_j. Note that applying the bypass operation on C_j does not decrease the weight of C_j because of the triangle inequality. Moreover, the edges of C_j not contained in dangling 2-cycles of C_j form a cycle. We call it the *primary cycle* of C_j. Note that C_j may have neither bypass edges nor dangling 2-cycles (this happens when C_j has no spoiled edges).

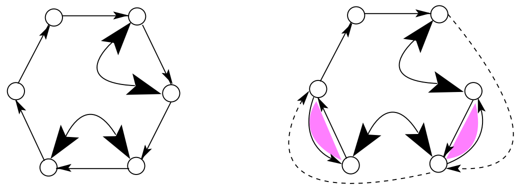

Figure 1: (1) A 3^+-cycle C_j (formed by the one-way edges) in $\mathcal{C}_1$ and the open chains (each shown by a two-way edge) each of which has a parallel edge in C_j. (2) The modified C_j (formed by the one-way edges), where bypass edges are dashed and dangling 2-cycles are painted.

Let H be the union of the modified $C_1, \ldots, C_\ell$, i.e., let H be the directed graph with $V(H) = \bigcup_{1 \le j \le \ell} V(C_j)$ and $E(H) = \bigcup_{1 \le j \le \ell} E(C_j)$. We next show that $E(H)$ can be partitioned into three subsets each of which can be added to G_2 without destroying its 2-path-colorability. Before proceeding to the details of the partitioning, we need several definitions and lemmas.

Two edges (u_1, u_2) and (v_1, v_2) of H form a *critical pair* if u_1 and v_2 are the endpoints of some open chain in G_2 and u_2 and v_1 are the endpoints of another open chain in G_2 (see Figure 2). Note that adding both (u_1, u_2) and (v_1, v_2) to G_2 destroys its 2-path-colorability. An edge of H is *critical* if it together with another edge of H forms a critical pair. Note that for each critical edge e of H, there is a unique edge e' in H such that e and e' form a critical pair. We call e' the *rival* of e. An edge of H is *safe* if it is not critical. A *bypass edge* of H is a bypass edge of a C_j with $1 \le j \le \ell$. Similarly, a *dangling 2-cycle* of H is a dangling 2-cycle of a C_j with $1 \le j \le \ell$. A *dangling edge* of H is an edge in a dangling 2-cycle of H.

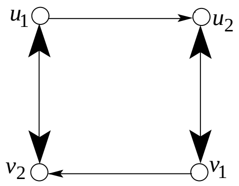

Figure 2: A critical pair formed by edges (u_1, u_2) and (v_1, v_2).

Lemma 3.3 *No bypass edge of H is critical.*

PROOF. Suppose that $e = (u_1, u_2)$ is a bypass edge of a C_j with $1 \le j \le \ell$. Then, u_2 is the articulation vertex of a dangling 2-cycle C of C_j. Let u_3 be the vertex of C other than u_2. Then, there is an open chain P in G_2 whose endpoints are u_1 and u_3. Since e leaves u_1 and $e' = (u_2, u_3)$ is the unique edge in C_j entering u_3, e' has to be the rival of e whenever e is critical. However, by the definition of criticalness, each critical edge and its rival should not be adjacent. So, e cannot be critical. □

Lemma 3.4 *Fix a j with $1 \le j \le \ell$. Suppose that an edge e of C_j is a critical dangling edge of H. Let C be the dangling 2-cycle of C_j containing e. Let e' be the rival of e. Then, the following statements hold:*

1. *e' is also an edge of C_j.*
2. *If e' is also a dangling edge of H, then the primary cycle of C_j consists of the two bypass edges corresponding to C and C', where C' is the dangling 2-cycle of C_j containing e'.*
3. *If e' is not a dangling edge of H, then e' is the edge in the primary cycle of C_j whose head is the tail of the bypass edge corresponding to C.*

PROOF. Let u_1 be the articulation vertex of C, and let u_2 be the other vertex of C. Then, there is an open chain P one of whose endpoints is u_2. Let u_3 be the other endpoint of P. We now prove the statements separately as follows.

Statement 1. Note that u_3 must be a vertex of C_j (indeed, (u_3, u_1) is a bypass edge of C_j). By the definition of criticalness, the rival of e is an edge incident to u_3. However, every edge of H incident to u_3 is in C_j. Thus, the rival of e must be in C_j whenever e is critical.

Statement 2. Suppose that e' is also a dangling edge of H. Then, since e' is incident to u_3 (as observed in the proof of Statement 1) and u_3 appears in the primary cycle of C_j, u_3 must be the articulation vertex of the dangling 2-cycle C' containing e'. Let u_4 be the vertex of C' other than u_3. Then, by the definition of criticalness, there is an open chain in G_2 whose endpoints are u_4 and u_1. Now, (u_1, u_3) has to be the bypass edge corresponding to C'. Recall that (u_3, u_1) is the bypass edge corresponding to C. This completes the proof of Statement 2.

Statement 3. Suppose that e' is not a dangling edge of H. Recall that e' is incident to u_3 and (u_3, u_1) is a bypass edge of C_j. By Lemma 3.3, e' cannot be (u_3, u_1). So, e' has to be the edge in the primary cycle of C_j entering u_3. □

Lemma 3.5 *Fix a j with $1 \le j \le \ell$ such that the primary cycle C of C_j contains no bypass edge. Let $u_1, \ldots, u_k$ be a cyclic ordering of the vertices in C. Then, the following hold:*

1. *Suppose that there is a chain P in G_2 whose endpoints appear in C but not consecutively (i.e., its endpoints are not connected by an edge of C). Then, at least one edge of C is safe.*
2. *Suppose that every edge of C is critical. Then, there is a unique $C_{j'}$ with $j' \in \{1, \ldots, \ell\} - \{j\}$ such that (1) the primary cycle C' of $C_{j'}$ has exactly k vertices and (2) the vertices of C' have a cyclic ordering $v_1, \ldots, v_k$ such that for every $1 \le i \le k$, u_i and v_{k-i+1} are the endpoints of some chain in G_2.* (See Figure 4.)

PROOF. We prove the two statements separately as follows.

Statement 1. By the existence of P, we can find two vertices u_i and u_h in C with $i < h$ such that (1) neither (u_i, u_h) nor (u_h, u_i) is an edge of C, (2) there is a chain in G_2 whose endpoints are u_i and u_h, and (3) there is no chain in G_2 whose endpoints both are in the set $\{u_{i+1}, u_{i+2}, \ldots, u_{h-1}\}$. Obviously, (u_i, u_{i+1}) is safe.

Statement 2. Each vertex u_i of C is an endpoint of a chain P_i in G_2 or else the two edges incident to u_i would be safe. Moreover, $P_1 \ne P_2$, $P_2 \ne P_3$, ..., $P_{k-1} \ne P_k$, and $P_k \ne P_1$ because we have applied the bypass operation on C_j. Furthermore, by Statement 1, there do not exist i and h with $1 \le i \ne h \le k$ with $P_i = P_h$. Therefore, for every $i \in \{1, \ldots, k\}$, the endpoint of P_i other than u_i is not in C.

For each $i \in \{1, \ldots, k\}$, let v_{k-i+1} be the endpoint of P_i other than u_i. Obviously, for each $i \in \{1, \ldots, k-1\}$, (v_{k-i}, v_{k-i+1}) has to be an edge of H because (u_i, u_{i+1}) is a critical edge. Similarly, (v_k, v_1) has to be an edge of H because (u_k, u_1) is a critical edge. So, $v_1, \ldots, v_k$ is a cyclic ordering of the vertices of some cycle C' in H. Let j' be the integer in $\{1, \ldots, \ell\}$ such that C' is a cycle in $C_{j'}$.

It remains to show that C' is not a dangling 2-cycle of $C_{j'}$. For a contradiction, assume that C' is a dangling 2-cycle of $C_{j'}$. Then, by Statement 1 in Lemma 3.4, $j = j'$ and C has to be the primary cycle of $C_{j'}$. Moreover, since C' is a 2-cycle, C is a 2-cycle, too. But then, $\{u_1, u_2\} \cap \{v_1, v_2\} \ne \emptyset$, because the articulation vertex of C' has to be a vertex of C. This contradicts the fact that for each $i \in \{1, \ldots, k\}$, the endpoint of P_i other than u_i is not in C (as observed above). □

Now we are ready to describe how to partition $E(H)$ into three subsets each of which can be added to G_2 without destroying its 2-path-colorability. We use the three colors 0, 1, and 2 to represent the three subsets, and want to assign each edge of $E(H)$ a color in $\{0, 1, 2\}$ so that the following conditions are satisfied:

(C1) For every critical edge e of H, e and its rival receive different colors.

(C2) For every dangling 2-cycle C of H, the two edges in C receive the same color.

(C3) If two adjacent edges of H receive the same color, then they form a 2-cycle of H.

To compute a coloring of the edges of H satisfying the above three conditions, we process $C_1, \ldots,$ C_ℓ in an arbitrary order. While processing C_j ($1 \le j \le \ell$), we color the edges of C_j by distinguishing four cases as follows (where C denotes the primary cycle of C_j):

Case 1: C is a 2-cycle. Then, C contains either one or two bypass edges. In the former (respectively, latter) case, we color the edges of C_j as shown in Figure 3(2) (respectively, Figure 3(1)). Note that the colored edges satisfy Conditions (C1) through (C3) above.

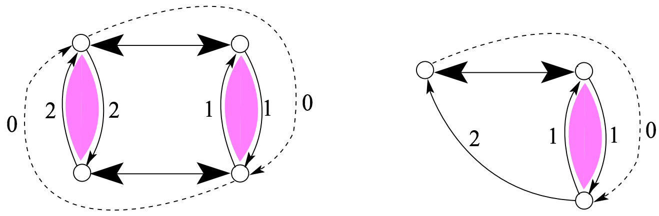

Figure 3: Coloring C_j when its primary cycle is a 2-cycle.

Case 2: Every edge of C is critical. Then, by Lemma 3.3, C contains no bypass edge. Let j' be the integer in $\{1, \ldots, \ell\} - \{j\}$ such that $C_{j'}$ satisfies the two conditions (1) and (2) in Statement 2 in Lemma 3.5. Then, by Lemma 3.4 and Statement 2 in Lemma 3.5, neither C_j nor $C_{j'}$ has a bypass edge or a dangling 2-cycle. So, the primary cycle of C_j (respectively, $C_{j'}$) is C_j (respectively, $C_{j'}$) itself. We color the edges of C_j and $C_{j'}$ simultaneously as follows (see Figure 4). First, we choose one edge e of C_j, color e with 2, and color the rival of e with 0. Note that the uncolored edges of C_j form a path Q. Starting at one end of Q, we then color the edges of Q alternatingly with colors 0 and 1. Finally, for each uncolored edge e' of $C_{j'}$, we color it with the color $h \in \{1, 2\}$ such that the rival of e' has been colored with $h - 1$. Note that the colored edges satisfy Conditions (C1) through (C3) above.

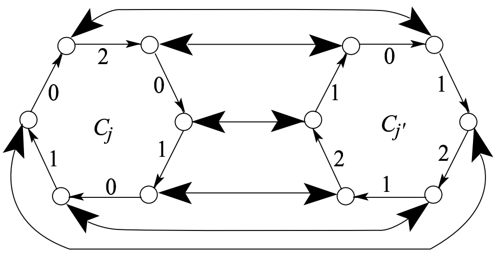

Figure 4: Coloring C_j and $C_{j'}$ when all their edges are critical.

Case 3: Neither Case 1 nor Case 2 occurs and no edge of C_j is a critical dangling edge of H. Then, by Lemma 3.3 and Statement 1 in Lemma 3.5, C contains at least one safe edge. Let $e_1, \ldots, e_k$ be the edges of C, and assume that they appear in C cyclically in this order. Without loss of generality, we may assume that e_1 is a safe edge. We color e_1 with 0, and then color the edges $e_2, \ldots, e_k$ in this order as follows. Suppose that we have just colored e_i with a color $h_i \in \{0, 1, 2\}$ and we want to color e_{i+1} next, where $1 \le i \le k - 1$. If e_{i+1} is a critical edge and its rival has been colored with $(h_i + 1) \bmod 3$, then we color e_{i+1} with $(h_i + 2) \bmod 3$; otherwise, we color e_{i+1} with $(h_i + 1) \bmod 3$. If e_k is colored 0 at the end, then we change the color of e_1 from 0 to the color in $\{1, 2\}$ that is not the color of e_2. Now, we can further color each dangling 2-cycle C' of C_j with the color in $\{0, 1, 2\}$ that has not been used to color the two edges of C incident to the articulation vertex of C'. Note that the colored edges satisfy Conditions (C1) through (C3) above.

Case 4: Neither Case 1 nor Case 2 occurs and some edge of C_j is a critical dangling edge of H. For each dangling edge e of H with $e \in E(C_j)$, we define the *partner* of e to be the edge e' of C leaving the articulation vertex u of the dangling 2-cycle containing e, and define the *mate* of e to be the bypass edge e'' of C_j entering u (see Figure 6). We say that an edge e of C_j is *bad* if e is a critical dangling edge of H and its partner is the rival of another critical dangling edge of H. If C_j has a bad edge e, then Statement 3 in Lemma 3.4 ensures that C_j is as shown in Figure 5 and can be colored as shown there without violating Conditions (C1) through (C3) above.

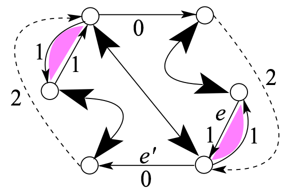

Figure 5: C_j (formed by the one-way edges) and its coloring when it has a bad edge e.

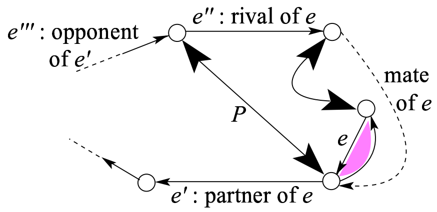

Figure 6: The rival, the mate, and the partner of a critical dangling edge e of H together with the opponent of the partner of e.

So, suppose that C_j has no bad edge. We need one more definition (see Figure 6). Consider a critical dangling edge e of H with $e \in E(C_j)$. Let e' and e'' be the partner and the rival of e, respectively. Let e''' be the edge of C entering the tail of e''. Let P be the open chain in G_2 whose endpoints are the tails of e' and e''. We call e''' the *opponent* of e'. Note that $e' \neq e'''$ because the endpoints of P are the tail of e' and the head of e'''. Moreover, if e' is a critical edge of H, then the rival of e' has to be e''' because e is not bad and P exists. In other words, whenever an edge of C has both its rival and its opponent, they must be the same. Similarly, if e''' is a critical edge of H, then its rival has to be e'. Obviously, neither e' nor e''' can be the rival or the mate of a critical dangling edge of H (because C_j has no bad edge).

Now, let $e_1, \ldots, e_q$ be the edges of C none of which is the rival or the mate of a critical dangling edge of C_j. We may assume that $e_1, \ldots, e_q$ appear in C cyclically in this order. Without loss of generality, we may further assume that e_1 is the partner of a critical dangling edge of H. Then, we color e_1 with 0, and further color $e_2, \ldots, e_q$ in this order as follows. Suppose that we have just colored e_i with a color $h_i \in \{0, 1, 2\}$ and we want to color e_{i+1} next, where $1 \leq i \leq q-1$. If e_{i+1} is a critical edge of H and its rival or opponent has been colored with $(h_i + 1) \bmod 3$, then we color e_{i+1} with $(h_i + 2) \bmod 3$; otherwise, we color e_{i+1} with $(h_i + 1) \bmod 3$. Note that the colored edges satisfy Conditions (C1) through (C3) above, because the head of e_q is not the tail of e_1.

We next show how to color the rival and the mate of each critical dangling edge of C_j. For each critical dangling edge e of C_j, since its partner e' and the opponent of e' have been colored, we can color the rival of e with the color of e' and color the mate of e with a color in $\{0, 1, 2\}$ that is not the color of e'. Note that the colored edges satisfy Conditions (C1) through (C3) above, because e' and its opponent have different colors.

Finally, for each dangling 2-cycle D of C_j, we color the two edges of D with the color in $\{0, 1, 2\}$ that has not been used to color an edge incident to the articulation vertex of D. Note that the colored edges satisfy Conditions (C1) through (C3) above, because the rival of each critical dangling edge e of H has the same color as the partner of e does. This completes the coloring of C_j (and hence H).

We next want to show how to use the coloring to find a large-weight tour in G. For each $i \in \{0, 1, 2\}$, let E_i be the edges of H with color i. Without loss of generality, we may assume that $w(E_0) \geq \max\{w(E_1), w(E_2)\}$. Then, $w(E_0) \geq \frac{1}{3} W_{1,3}$ (see the beginning of this subsection for $W_{1,3}$). Consider the undirected graph $U = (V(G), F_1 \cup F_2)$, where F_1 consists of all edges $\{v_1, v_2\}$ such that (v_1, v_2) or (v_2, v_1) is an edge in E_0, and F_2 consists of all edges $\{v_3, v_4\}$ such that v_3 and v_4 are the endpoints of an open chain in G_2. We further assign a weight to each edge of F_1 as follows. We first initialize the weight of each edge of F_1 to be 0. For each edge $(v_1, v_2) \in E_0$, we then add the weight of edge (v_1, v_2) to the weight of edge $\{v_1, v_2\}$. Note that for each $i \in \{1, 2\}$, each connected component of the undirected graph $(V(G), F_i)$ is a single vertex or a single edge because of Condition (C3) above. So, each connected component of U is a path or a cycle. Moreover, each cycle of U contains at least three edges of F_1 because of Condition (C1) above. For eacy cycle D of U, we mark exactly one edge $\{v_1, v_2\} \in F_1$ in D whose weight is the smallest among all edges $\{v_1, v_2\} \in F_1$ in D. Let E_3 be the set of all edges $(v_1, v_2) \in E_0$ such that $\{v_1, v_2\}$ is marked. Then, $w(E_3) \leq \frac{1}{3}w(E_0)$. Consider the directed graph G'_2 obtained from G_2 by adding the edges of $E_0 - E_3$. Obviously, $w(G'_2) \geq (W_{1,2} + W_{2,2}) + \frac{1}{9}W_{1,3}$. Moreover, G'_2 is a collection of partial chains and hence is 2-path-colorable. So, we can partition the edges of G'_2 into two subsets E'_1 and E'_2 such that both graphs $(V(G), E'_1)$ and $(V(G), E'_2)$ are subtours of G. The heavier one among the two subtours can be completed to a tour of G of weight at least $\frac{1}{2}(W_{1,2} + W_{2,2}) + \frac{1}{18}W_{1,3} \geq W_2 + \frac{1}{9}W_3$. Combining this with Lemma 3.2, we now have:

Theorem 3.6 *There is a polynomial-time approximation algorithm for* AsymMaxTSP *achieving an approximation ratio of* $\frac{27}{35}$.

4 New Algorithm for Metric SymMaxTSP

Throughout this section, fix an instance (G, w) of metric SymMaxTSP, where G is a complete undirected graph with n vertices and w is a function mapping each edge e of G to a nonnegative real number $w(e)$. Because of the triangle inequality, the following fact holds (see [3] for a proof):

Fact 4.1 *Suppose that $P_1, \ldots, P_t$ are vertex-disjoint paths in G each containing at least one edge. For each $1 \leq i \leq t$, let u_i and v_i be the endpoints of P_i. Then, we can use some edges of G to connect $P_1, \ldots, P_t$ into a single cycle C in linear time such that $w(C) \geq \sum_{i=1}^{t} w(P_i) + \frac{1}{2}\sum_{i=1}^{t} w(\{u_i, v_i\})$.*

Like Hassin and Rubinstein's algorithm (H&R2-algorithm) for the problem, our algorithm computes two tours T_1 and T_2 of G and outputs the one with the larger weight. The first two steps of our algorithm are the same as those of H&R2-algorithm:

1. Compute a maximum-weight cycle cover $\mathcal{C}$. Let $C_1, \ldots, C_r$ be the cycles in G.
2. Compute a maximum-weight matching M in G.

Lemma 4.2 [3] *In linear time, we can compute two disjoint subsets A_1 and A_2 of $\bigcup_{1 \leq i \leq r} E(C_i) - M$ satisfying the following conditions:*

(a) For each $j \in \{1, 2\}$, each connected component of the graph $(V(G), M \cup A_j)$ is a path of length at least 1.

(b) For each $j \in \{1, 2\}$ and each $i \in \{1, \ldots, r\}$, $|A_j \cap E(C_i)| = 1$.

For a technical reason, we will allow our algorithm to use only 1 random bit (so we can easily derandomize it, although we omit the details). The third through the seventh steps of our algorithm are as follows:

3. Compute two disjoint subsets A_1 and A_2 of $\bigcup_{1 \leq i \leq r} E(C_i) - M$ satisfying the two conditions in Lemma 4.2.
4. Choose A from A_1 and A_2 uniformly at random.
5. Obtain a collection of vertex-disjoint paths each of length at least 1 by deleting the edges in A from $\mathcal{C}$; and then connect these paths into a single (Hamiltonian) cycle T_1 as described in Fact 4.1.
6. Let $S = \{v \in V(G) \mid$ the degree of v in the graph $(V, M \cup A)$ is 1$\}$ and $F = \{\{u, v\} \in E(G) \mid \{u, v\} \subseteq S\}$. Let H be the complete graph (S, F). Let $\ell = \frac{1}{2}|S|$. (Comment: $|S|$ is even, because of Condition (a) in Lemma 4.2.)
7. Let M' be the set of all edges $\{u, v\} \in F$ such that some connected component of the graph $(V, M \cup A)$ contains both u and v. (Comment: M' is a perfect matching of H because of Condition (a) in Lemma 4.2.)

Lemma 4.3 [3] *Let $\alpha = w(A_1 \cup A_2)/w(\mathcal{C})$. For a random variable X, let $\mathcal{E}[X]$ denote its expected value. Then, $\mathcal{E}[w(F)] \geq \frac{1}{4}(1 - \alpha)(2\ell - 1)w(\mathcal{C})$.*

The next lemma shows that there cannot exist matchings of large weight in an edge-weighted graph where the weights satisfy the triangle inequality:

Lemma 4.4 *For every perfect matching N of H, $w(N) \leq w(F)/\ell$.*

Proof. Let the edges of N be $\{u_1, u_2\}$, $\{u_3, u_4\}$, ..., $\{u_{2\ell-1}, u_{2\ell}\}$.

Case 1: ℓ is odd. For each odd number i with $1 \leq i \leq \ell$, we assign the vertices u_{i+2}, u_{i+3}, ..., $u_{\ell+i}$ of H to the edge $\{u_i, u_{i+1}\}$ of N. For each even number j with $1 \leq j \leq \ell$, we assign the vertices u_1, u_2, ..., u_j, $u_{\ell+j+2}$, $u_{\ell+j+3}$, ..., $u_{2\ell}$ of H to the edge $\{u_{\ell+j}, u_{\ell+j+1}\}$ of N. Note that each edge in N is assigned exactly $\ell - 1$ vertices of H. For each edge $e_i = \{u_i, u_{i+1}\} \in N$ and each vertex u_h assigned to e_i, we then assign the two edges $\{u_i, u_h\}$ and $\{u_{i+1}, u_h\}$ of H to e_i. Since $w(\{u_i, u_h\}) + w(\{u_{i+1}, u_h\}) \geq w(e_i)$ by the triangle inequality, the total weight of edges assigned to each edge $e_i \in N$ is at least $(\ell - 1)w(e_i)$. Obviously, no edge of N is assigned to itself or another edge of N. Moreover, a simple but crucial observation is that no edge of H is assigned to two or more edges of N. Thus, $w(F - N) \geq (\ell - 1)w(N)$. Hence, $w(N) \leq w(F)/\ell$.

Case 2: ℓ is even. Let $N_1 = \{\{u_1, u_2\}, \{u_3, u_4\}, \ldots, \{u_{n-1}, u_n\}\}$ and $N_2 = N - N_1$. We assume that $w(N_1) \geq w(N_2)$; the other case is similar. For each odd number i with $1 \leq i \leq \ell - 1$, we assign the vertices u_{i+2}, u_{i+3}, ..., $u_{\ell+i+1}$ of H to the edge $\{u_i, u_{i+1}\}$ of N, and assign the vertices u_1, u_2, ..., u_{i-1}, $u_{\ell+i+2}$, $u_{\ell+i+3}$, ..., $u_{2\ell}$ of H to the edge $\{u_{\ell+i}, u_{\ell+i+1}\}$ of N. Note that each edge in N_1 (respectively, N_2) is assigned exactly ℓ (respectively, $\ell - 2$) vertices of H. For each edge $e_i = \{u_i, u_{i+1}\} \in N$ and each vertex u_h assigned to e_i, we then assign the two edges $\{u_i, u_h\}$ and $\{u_{i+1}, u_h\}$ of H to e_i. Since $w(\{u_i, u_h\}) + w(\{u_{i+1}, u_h\}) \geq w(e_i)$ by the triangle inequality, the total weight of edges assigned to each edge $e_i \in N_1$ (respectively, $e_i \in N_2$) is at least $\ell w(e_i)$ (respectively, $(\ell - 2)w(e_i)$). Obviously, no edge of N is assigned to itself or another edge of N. Moreover, a simple but crucial observation is that no edge of H is assigned to two or more edges of N. Thus, $w(F - N) \geq \ell w(N_1) + (\ell - 2)w(N_2) \geq (\ell - 1)w(N)$. Hence, $w(N) \leq w(F)/\ell$. □

The following is our main lemma and will be proved in Section 4.1:

Lemma 4.5 *We can partition $F - M'$ into $2\ell - 2$ perfect matchings M_1, ..., $M_{2\ell-2}$ of H in linear time satisfying the following condition:*

- *For every natural number q, there are at most $q^2 - q$ matchings M_i with $1 \leq i \leq 2\ell - 2$ such that the graph $(S, M' \cup M_i)$ has a cycle of length at most $2q$.*

Now, the eighth through the thirteenth steps of our algorithm are as follows:

8. Partition $F - M'$ into $2\ell - 2$ perfect matchings M_1, ..., $M_{2\ell-2}$ of H in linear time satisfying the condition in Lemma 4.5.
9. Let $q = \lceil \sqrt[3]{\ell} \rceil$. Find a matching M_i with $1 \leq i \leq 2\ell - 2$ satisfying the following two conditions:
   (a) The graph $(S, M' \cup M_i)$ has no cycle of length at most $2q$.
   (b) $w(M_i) \geq w(M_j)$ for all matchings M_j with $1 \leq j \leq 2\ell - 2$ such that the graph $(S, M' \cup M_j)$ has no cycle of length at most $2q$.
10. Construct the graph $G'_i = (V(G), M \cup A \cup M_i)$. (Comment: $M_i \cap (M \cup A) = \emptyset$ and each connected component of G'_i is either a path, or a cycle of length $2q + 1$ or more.)
11. For each cycle D in G'_i, mark exactly one edge $e \in M_i \cap E(D)$ such that $w(e) \leq w(e')$ for all $e' \in M_i \cap E(D)$.
12. Obtain a collection of vertex-disjoint paths each of length at least 1 by deleting the marked edges from G'_i; and then connect these paths into a single (Hamiltonian) cycle T_2 as described in Fact 4.1.
13. If $w(T_1) \geq w(T_2)$, output T_1; otherwise, output T_2.

Theorem 4.6 *There is an $O(n^3)$-time approximation algorithm for metric* SymMaxTSP *achieving an approximation ratio of $\frac{7}{8} - O(1/\sqrt[3]{n})$.*

PROOF. Let OPT be the maximum weight of a tour in G. It suffices to prove that $\max\{\mathcal{E}[w(T_1)], \mathcal{E}[w(T_2)]\} \geq (\frac{7}{8} - O(1/\sqrt[3]{n}))OPT$. By Fact 4.1, $\mathcal{E}[w(T_1)] \geq (1 - \frac{1}{2}\alpha + \frac{1}{4}\alpha)w(\mathcal{C}) \geq (1 - \frac{1}{4}\alpha)OPT$.

We claim that $|S| \geq \frac{1}{3}n$. To see this, consider the graphs $G_M = (V(G), M)$ and $G_A = (V(G), M \cup A)$. Because the length of each cycle in $\mathcal{C}$ is at least 3, $|A| \leq \frac{1}{3}n$ by Condition (b) in Lemma 4.2. Moreover, since M is a matching of G, the degree of each vertex in G_M is 0 or 1. Furthermore, G_A is obtained by adding the edges of A to G_M. Since adding one edge of A to G_M increases the degrees of at most two vertices, there exist at least $n - 2|A| \geq \frac{1}{3}n$ vertices of degree 0 or 1 in G_A. So, by Condition (a) in Lemma 4.2, there are at least $\frac{1}{3}n$ vertices of degree 1 in G_A. This establishes that $|S| \geq \frac{1}{3}n$. Hence, $\ell \geq \frac{1}{6}n$.

Now, let x be the number of matchings M_j with $1 \leq j \leq 2\ell - 2$ such that the graph $(S, M' \cup M_i)$ has a cycle of length at most $2q$. Then, by Lemmas 4.4 and 4.5, the weight of the matching M_i found in Step 9 is at least $(1 - \frac{x+1}{\ell}) \cdot w(F) \cdot \frac{1}{2\ell-2-x}$. So, $w(M_i) \geq \frac{1}{\ell} \cdot (1 - \frac{\ell-1}{2\ell-2-q^2+q}) \cdot w(F)$ because $x \leq q^2 - q$. Let N_i be the set of edges of M_i marked in Step 11. Then, $w(M_i - N_i) \geq \frac{q}{q+1} \cdot \frac{\ell-q^2+q-1}{\ell(2\ell-2-q^2+q)} \cdot w(F)$. Hence, by Lemma 4.3 and the inequality $\ell \geq \frac{1}{6}n$, we have $\mathcal{E}[w(M_i - N_i)] \geq \frac{1}{4}(1-\alpha)(1-O(1/\sqrt[3]{n}))w(\mathcal{C})$.

Obviously, $\mathcal{E}[w(T_2)] \geq \mathcal{E}[w(M \cup A)] + \mathcal{E}[w(M_i - N_i)] \geq (\frac{1}{2} - \frac{1}{2n})OPT + \frac{1}{2}\alpha w(\mathcal{C}) + \mathcal{E}[w(M_i - N_i)]$. Hence, by the last inequality in the previous paragraph, $\mathcal{E}[w(T_2)] \geq (\frac{3}{4} + \frac{1}{4}\alpha - O(1/\sqrt[3]{n}))OPT$. Combining this with the inequality $\mathcal{E}[w(T_1)] \geq (1 - \frac{1}{4}\alpha)OPT$, we finally have $\mathcal{E}[\max\{w(T_1), w(T_2)\}] \geq (\frac{7}{8} - O(1/\sqrt[3]{n}))OPT$.

The running time of the algorithm is dominated by the $O(n^3)$ time needed for computing a maximum-weight cycle cover and a maximum-weight matching. $\square$

As observed in [3], the subsets A_1 and A_2 in Lemma 4.2 can be computed in $O(\log^3 n)$ time using a linear number of processors. So, our algorithm for metric Max TSP is parallelizable because maximum-weight cycle covers and maximum-weight matchings can be computed by fast parallel algorithms [6, 8]. We omit the details here.

4.1 Partitioning into Perfect Matchings

Let the vertices of H be ∞, 0, 1, …, $2\ell - 2$, and let the edges of M' be

$$\{\infty, 0\}, \{1, 2\ell - 2\}, \{2, 2\ell - 3\}, \ldots, \{\ell - 1, \ell\}.$$

Then, a folklore partitioning of $F - M'$ into $2\ell - 2$ perfect matchings M_1, …, $M_{2\ell-2}$ of H is as follows:

$$M_1: \ \{\infty, 1\}, \{2, 0\}, \{3, 2\ell - 2\}, \ldots, \{\ell, \ell + 1\}$$

$$M_2: \ \{\infty, 2\}, \{3, 1\}, \{4, 0\}, \ldots, \{\ell + 1, \ell + 2\}$$

$$\vdots$$

$$M_{2\ell-2}: \ \{\infty, 2\ell - 2\}, \{0, 2\ell - 3\}, \{1, 2\ell - 4\}, \ldots, \{\ell - 2, \ell - 1\}.$$

For each integer $j \notin \{0, 1, \ldots, 2\ell - 2\}$, we identify j with the vertex h of H such that $h \equiv j \pmod{2\ell - 1}$. Then, for each integer $i \in \{0, 1, \ldots, 2\ell - 2\}$, M_i consists of edge $\{\infty, i\}$ and all edges $\{j, -j + 2i\}$ with $j \in \{0, 1, \ldots, 2\ell - 2\} - \{i\}$. Obviously, for each $i \in \{1, \ldots, 2\ell - 2\}$, the graph $H_i = (S, M_i \cup M')$ is a collection of vertex-disjoint cycles; we call the cycle containing vertex ∞ the *main cycle* of H_i and denote it by D_i. For two natural numbers x and y, let $\gcd(x, y)$ denote the greatest common divisor of x and y, and let $\mathrm{lcm}(x, y)$ denote the least common multiple of x and y.

Lemma 4.7 *For each $i \in \{1, \ldots, 2\ell - 2\}$, the length of D_i is $\left(\frac{2\ell-1}{\gcd(2\ell-1, i)} + 1\right)$.*

PROOF. Recall that for each integer $i \in \{0, 1, \ldots, 2\ell - 2\}$, M_i consists of edge $\{\infty, i\}$ and all edges $\{j, -j + 2i\}$ with $j \in \{0, 1, \ldots, 2\ell - 2\} - \{i\}$. Fix an $i \in \{1, \ldots, 2\ell - 2\}$. Let $2h$ be the length of D_i.

Suppose that we traverse D_i by starting at vertex ∞, then visiting i, and proceeding along the cycle until reaching vertex 0. This traversal should give the following ordering of the vertices of D_i:

$$\infty, i, -i, 3i, -3i, 5i, \cdots, -(2h-3)i, (2h-1)i$$

where $(2h-1)i \equiv 0 \pmod{2\ell-1}$ because vertex 0 is the last one in the traversal. Note that for every odd $x \in \{1, 2, \ldots, 2h-1\}$, xi is a vertex of D_i.

Since $(2h-1)i \equiv 0 \pmod{2\ell-1}$, $(2h-1)i$ is a common multiple of integers $2\ell-1$ and i, and hence there exists an integer $\alpha \geq 1$ such that

$$(2h-1)i = \alpha \operatorname{lcm}(2\ell-1, i) = \left(\alpha \cdot \frac{2\ell-1}{\gcd(2\ell-1, i)}\right) i. \tag{4.1}$$

The last equality follows from the fact that $(2\ell-1)\, i = \gcd(2\ell-1, i) \operatorname{lcm}(2\ell-1, i)$. By Equation 4.1, $2h-1 = \alpha \cdot \frac{2\ell-1}{\gcd(2\ell-1,i)}$. Therefore, α is an odd integer because $\frac{2\ell-1}{\gcd(2\ell-1,i)}$ is an integer and $2h-1$ is odd.

We claim that $\alpha = 1$. For a contradiction, assume that α is an odd integer greater than 1. Then, by Equation 4.1, $(2h-1)i - (\alpha-1)\operatorname{lcm}(2\ell-1, i) = \operatorname{lcm}(2\ell-1, i)$ and hence

$$2h - 1 - (\alpha - 1) \cdot \frac{2\ell-1}{\gcd(2\ell-1, i)} = \frac{\operatorname{lcm}(2\ell-1, i)}{i}. \tag{4.2}$$

Since $\alpha - 1$ is a possitive even integer, the left side of Equation 4.2 is an odd integer less than $2h-1$. Moreover, recall that $2h-1 = \alpha \cdot \frac{2\ell-1}{\gcd(2\ell-1,i)}$. So, the left side of Equation 4.2 is a positive odd integer less than $2h-1$. Hence, $(2h-1-(\alpha-1)\cdot\frac{2\ell-1}{\gcd(2\ell-1,i)})\, i$ is an integer in the subsequence i, $3i$, $5i$, $\ldots$, $(2h-3)i$, and is a multiple of $2\ell-1$ by Equation 4.2. However, this implies that vertex 0 of D_i is in the subsequence i, $3i$, $5i$, $\ldots$, $(2h-3)i$, a contradiction. Thus, the claim holds.

By the claim, $2h-1 = \frac{2\ell-1}{\gcd(2\ell-1,i)}$ and so the length of D_i is $2h = \frac{2\ell-1}{\gcd(2\ell-1,i)} + 1$. □

Corollary 4.8 *If* $\gcd(2\ell-1, i) = 1$, *then* D_i *is a tour of* H_i.

We next show that if D_i is not a tour of H_i, then D_i is the shortest cycle in H_i.

Lemma 4.9 *Fix an* i *such that* $1 \leq i \leq 2\ell-2$ *and* $\gcd(2\ell-1, i) \neq 1$. *Then, each cycle of* H_i *other than* D_i *is of length* $\frac{2(2\ell-1)}{\gcd(2\ell-1,i)}$.

Proof. Fix a cycle D of H_i other than D_i. Let $2h$ be the length of D. Consider an arbitrary vertex j of D. As in the proof of Lemma 4.7, a traversal of D started at vertex j and ended at vertex $-j$ produces the following ordering of the vertices of D:

$$j, -j+2i, j-2i, -j+4i, j-4i, -j+6i, \ldots, j-2(h-1)i, -j+2hi$$

where $-j + 2hi \equiv -j \pmod{2\ell-1}$. Note that for every even $x \in \{2, 3, \ldots, 2h\}$, $-j + xi$ is a vertex of D.

Since $2hi \equiv 0 \pmod{2\ell-1}$, $2hi$ is a common multiple of integers $2\ell-1$ and i, hence there exists an integer $\alpha \geq 1$ such that

$$2hi = \alpha \operatorname{lcm}(2\ell-1, i) = \left(\alpha \cdot \frac{2\ell-1}{\gcd(2\ell-1, i)}\right) i. \tag{4.3}$$

By Equation 4.3, $2h = \alpha \cdot \frac{2\ell-1}{\gcd(2\ell-1,i)}$. Therefore, α is an even integer.

We claim that $\alpha = 2$. For a contradiction, assume that α is an even number greater than 2. Then, by Equation 4.3, $2hi - (\alpha-2)\operatorname{lcm}(2\ell-1, i) = 2\operatorname{lcm}(2\ell-1, i)$ and hence

$$2h - (\alpha - 2) \cdot \frac{2\ell-1}{\gcd(2\ell-1, i)} = \frac{2\operatorname{lcm}(2\ell-1, i)}{i}. \tag{4.4}$$

Since $\alpha - 2$ is a possitive even integer, the left side of Equation 4.4 is an even integer less than $2h$. Moreover, recall that $2h = \alpha \cdot \frac{2\ell-1}{\gcd(2\ell-1,i)}$. So, the left side of Equation 4.4 is a positive even integer less than $2h$. Hence, $-j + (2h - (\alpha - 2) \cdot \frac{2\ell-1}{\gcd(2\ell-1,i)})\, i$ is an integer in the subsequence $-j + 2i$, $-j + 4i$, $\ldots$, $-j + 2(h-1)i$, and is congruent to $-j$ modulo $2\ell - 1$ by Equation 4.4. However, this implies that vertex $-j$ of D_i is in the subsequence $-j + 2i$, $-j + 4i$, $\ldots$, $-j + 2(h-1)i$, a contradiction. Thus, the claim holds.

By the claim, $2h = \frac{2(2\ell-1)}{\gcd(2\ell-1,i)}$ and so the length of D_i is $2h = \frac{2(2\ell-1)}{\gcd(2\ell-1,i)}$. □

Corollary 4.10 *For every $i \in \{1, 2, \ldots, 2\ell - 2\}$, D_i is the shortest cycle in H_i.*

Proof. Fix an $i \in \{1, 2, \ldots, 2\ell - 2\}$. If $\gcd(2\ell - 1, i) = 1$, then D_i is the unique cycle (and hence the shortest cycle) in H_i by Corollary 4.8. Otherwise, by Lemmas 4.7 and 4.9, D_i is shorter than the other cycles in H_i. □

Now, we are ready to prove Lemma 4.5:

Proof of Lemma 4.5: Fix a natural number q. By Corollary 4.10, it suffices to show that there are at most $q^2 - q$ integers $i \in \{1, 2, \ldots, 2\ell - 2\}$ such that D_i is of length at most $2q$.

Consider a natural number $p \le q$. For each $i \in \{1, 2, \ldots, 2\ell - 2\}$, if the length of D_i is exactly $2p$, then by Lemma 4.7, $\frac{2\ell-1}{\gcd(2\ell-1,i)} + 1 = 2p$ and so

$$\gcd(2\ell - 1, i) = \frac{2\ell - 1}{2p - 1}.$$

Since each integer i satisfying the above equality has to be a multiple of $\frac{2\ell-1}{2p-1}$, there can be at most $2p - 2$ such integers in $\{1, 2, \ldots, 2\ell - 2\}$.

Hence, there can be at most $\sum_{p=1}^{q}(2p - 2) = q^2 - q$ integers $i \in \{1, 2, \ldots, 2\ell - 2\}$ such that H_i has a cycle of length at most $2q$. □

References

[1] A. I. Barvinok, D. S. Johnson, G. J. Woeginger, and R. Woodroofe. Finding Maximum Length Tours under Polyhedral Norms. *Proceedings of the Sixth International Conference on Integer Programming and Combinatorial Optimization* (IPCO), Lecture Notes in Computer Science, **1412** (1998) 195–201.

[2] Z.-Z. Chen and L. Wang. An Improved Randomized Approximation Algorithm for Max TSP. *Submitted.*

[3] Z.-Z. Chen, Y. Okamoto, and L. Wang. Improved Deterministic Approximation Algorithms for Max TSP. To appear in *Information Processing Letters.*

[4] R. Hassin and S. Rubinstein. A 7/8-Approximation Approximations for Metric Max TSP. *Information Processing Letters*, **81** (2002) 247–251.

[5] H. Kaplan, M. Lewenstein, N. Shafrir, and M. Sviridenko. Approximation Algorithms for Asymmetric TSP by Decomposing Directed Regular Multigraphs. *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pp. 56–75, 2003.

[6] R. M. Karp, E. Upfal, and A. Wigderson. Constructing a Perfect Matching is in random NC. *Combinatorica*, **6** (1986) 35–48.

[7] A. V. Kostochka and A. I. Serdyukov. Polynomial Algorithms with the Estimates $\frac{3}{4}$ and $\frac{5}{6}$ for the Traveling Salesman Problem of Maximum (in Russian). *Upravlyaemye Sistemy*, **26** (1985) 55–59.

[8] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, **7** (1987) 105–113.