A 4/5 - Approximation Algorithm for the Maximum Traveling Salesman Problem

Szymon Dudycz, Jan Marcinkowski, Katarzyna Paluch(✉), and Bartosz Rybicki

Institute of Computer Science, University of Wrocław, Wrocław, Poland
szymon.dudycz@gmail.com, jasiekmarc@cs.uni.wroc.pl, abraka@cs.uni.wroc.pl, rybicki.bartek@gmail.com

Abstract. In the maximum traveling salesman problem (Max TSP) we are given a complete undirected graph with nonnegative weights on the edges and we wish to compute a traveling salesman tour of maximum weight. We present a fast combinatorial $\frac{4}{5}$ – approximation algorithm for Max TSP. The previous best approximation for this problem was $\frac{7}{9}$. The new algorithm is based on a technique of eliminating difficult subgraphs via gadgets with *half-edges*, a new method of edge coloring and a technique of exchanging edges.

1 Introduction

The Maximum Traveling Salesman Problem (Max TSP) is a classical variant of the famous Traveling Salesman Problem. In the problem we are given a complete undirected graph $G = (V, E)$ with nonnegative weights on the edges and we aim to compute a traveling salesman tour of maximum weight. Max TSP, also informally known as the "taxicab ripoff problem", is both of theoretical and practical interest.

Previous approximations of Max TSP have found applications in combinatorics and computational biology: the problem is useful in understanding RNA interactions [27] and providing algorithms for compressing the results of DNA sequencing [26]. It has also been applied to the problem of finding a maximum weight triangle cover of the graph [14] and to a combinatorial problem called *bandpass-2* [7], where we are supposed to find the best permutation of rows in a boolean-valued matrix, so that the weighted sum of structures called *bandpasses* is maximised.

Previous Results. The first approximation algorithms for Max TSP were devised by Fisher et al. [10]. They showed several algorithms having approximation ratio $\frac{1}{2}$ and one with a guarantee of $\frac{2}{3}$. In [16] Kosaraju, Park and Stein presented an improved algorithm giving a ratio of $\frac{19}{27}$ [4]. This was in turn improved

Partly supported by Polish National Science Center grant UMO-2013/11/B/ST6/01748.
J. Marcinkowski—Partially supported by Polish NSC grant 2015/18/E/ST6/00456.

F. Eisenbrand and J. Koenemann (Eds.): IPCO 2017, LNCS 10328, pp. 173–185, 2017.
DOI: 10.1007/978-3-319-59250-3_15

by Hassin and Rubinstein, who gave a $\frac{5}{7}$- approximation [12]. In the meantime Serdyukov [25] presented (in Russian) a simple and elegant $\frac{3}{4}$-approximation algorithm. The algorithm is deterministic and runs in $O(n^3)$, where n denotes the number of vertices in the graph. Afterwards, Hassin and Rubinstein gave [13] a randomized algorithm with expected approximation ratio of at least $\frac{25(1-\epsilon)}{33-32\epsilon}$ and running in $O(n^2(n + 2^{1/\epsilon}))$, where ϵ is an arbitrarily small constant. The first deterministic approximation algorithm with the ratio better than $\frac{3}{4}$ was given in [6] by Chen et al. It is a $\frac{61}{81}$-approximation through a non-trivial derandomization of the algorithm from [13] that runs in $O(n^3)$. The currently best known approximation given by Paluch et al. [22] achieves the ratio of $\frac{7}{9}$. Its running time is also $O(n^3)$.

Related Work. It is known that Max TSP is max-SNP-hard [3], so a constant $\delta < 1$ exists, which is an upper bound on the approximation ratio of any algorithm for this problem. The geometric version of the problem, where all vertices are in R^d and the weight of each edge is defined as the Euclidean distance of its endpoints, was considered in [2] and shown to be solvable in polynomial time for $d = 2$ and NP-hard for $d > 2$. Other metrics are also considered in that paper.

Regarding the path version of Max TSP – Max TSPP (the Maximum Traveling Salesman Path Problem), the approximation algorithms with ratios correspondingly $\frac{1}{2}$ and $\frac{2}{3}$ have been given in [19]. The first one for the case when both endpoints of the path are specified and the other for the case when only one endpoint is given.

Another related problem is called the maximum scatter TSP (see [1]), where the goal is to find a TSP tour (or a path) maximizing the weight of the lightest edge selected in the solution. The problem is motivated by medical imaging and some manufacturing applications. In general there is no constant approximation for this problem, but if the weights of the edges obey the triangle inequality, it is possible to give a $\frac{1}{2}$-approximation algorithm. That paper also studies a more general version of the maximum scatter TSP – the max-min-m-neighbour TSP. The improved approximation results for the max-min-2-neighbour problem have been given in [8].

The maximum metric symmetric traveling salesman problem, in which the edge weights satisfy the triangle inequality - the best approximation factor is $\frac{7}{8}$ [18]. For the maximum asymmetric traveling salesman problem with triangle inequality the best approximation ratio currently equals $\frac{35}{44}$ [17].

In the Maximum Latency TSP problem we are given a complete undirected graph with vertices $v_0, v_1, \ldots, v_n$. Our task is to find a Hamiltonian path starting at a fixed vertex v_0, which maximizes the total latency of the vertices. If in a given path P the weight of the i-th edge is w_i, then the latency of the j-th vertex is $L_j = \sum_{i=1}^{j} w_i$ and the total latency is defined as $L(P) = \sum_{j=1}^{n} L_j$. A ratio $\frac{1}{2}$-approximation algorithm for the metric version of the problem is presented in [5]. Improved ratios for this and other versions (directed, nonmetric) of the problem are shown in [11].

Our Approach and Results. We begin with computing a maximum weight *cycle cover* C_{max} of G. A cycle cover of a graph G is a collection of cycles such that each vertex belongs to exactly one of them. The weight of a maximum weight cycle cover C_{max} is an upper bound on OPT, where by OPT we denote the weight of a maximum weight traveling salesman tour. By computing a maximum weight perfect matching M we get another, even simpler than C_{max}, upper bound – on $OPT/2$. From C_{max} and M we build a multigraph G_1 which consists of two copies of C_{max} and one copy of M, (for each edge e of G the multigraph G_1 contains between zero and three copies of e). Thus the total weight of the edges of G_1 is at least $\frac{5}{2}$ OPT. Next we would like to *path-3-color* G_1, that is to color the edges of G_1 with three colors, so that each color class contains only vertex-disjoint paths. The paths from the color class with maximum weight can then be patched in an arbitrary manner into a tour of weight at least $\frac{5}{6}$ OPT.

Technique of Eliminating Difficult Subgraphs via Half-edges. Not every multigraph G_1 can, however, be path-3-colored. For example, a subgraph of G_1 obtained from a triangle $\mathcal{T}$ of C_{max} such that M contains one of the edges of $\mathcal{T}$ (such triangle is called a *3-kite* of G_1) cannot be path-3-colored as, clearly, it is impossible to color such seven edges with three colors and not create a monochromatic triangle. Similarly, a subgraph of G_1 obtained from a square $\mathcal{S}$ (i.e., a cycle of length four) of C_{max} such that M contains two edges connecting vertices of $\mathcal{S}$ (such square is called a *4-kite*) is not path-3-colorable. To find a way around this difficulty, we compute another cycle cover C_2 *improving* C_{max} *with respect to* M, which is a cycle cover that does not contain any 3-kite or 4-kite of G_1 and whose weight is also at least OPT. An important feature of C_2 is that it may contain *half-edges*. A half-edge of an edge e is, informally speaking, a half of the edge e that contains exactly one of its endpoints. Half-edges have already been introduced in [21]. Computing C_2 is done via a tailored reduction to a maximum weight perfect matching. It is, to some degree, similar to computing a directed cycle cover without length-two cycles in [21], but for Max TSP we need much more complex gadgets.

From one copy of C_2 and M we build another multigraph G_2 with weight at least $\frac{3}{2}$ OPT. It turns out that G_2 can always be *path-2-colored*. The multigraph G_1 may be non-path-3-colorable – if it contains at least one kite. We notice, however, that if we remove one arbitrary edge from each kite, then G_1 becomes path-3-colorable. The edges removed from G_1 are added to G_2. As a result, the modified G_2 may cease to be path-2-colorable. To remedy this, we in turn remove some edges from G_2 and add them to G_1. In other words, we find two disjoint sets of edges – a set $F_1 \subseteq G_1$ and a set $F_2 \subseteq G_2$, called *exchange sets*, such that the multigraph $G_1' = G_1 \setminus F_1 \cup F_2$ is path-3-colorable and the multigraph $G_2' = G_2 \setminus F_2 \cup F_1$ is path-2-colorable. Since G_1 and G_2 have the total weight at least 4 OPT, by path-3-coloring G_1' and path-2-coloring G_2' we obtain a $\frac{4}{5}$-approximate solution to Max TSP.

Edge Coloring. The presented algorithms for path-3-coloring and path-2-coloring are essentially based on a simple notion of a *safe edge* – an edge colored in such a way that it is guaranteed not to belong to any monochromatic cycle, used in an inductive way. The adopted approach may appear simple and straightforward. For comparison, let us point out that the method of path-3-coloring the multigraph obtained from two directed cycle covers described in [15] is rather convoluted.

Generally, the new techniques are somewhat similar to the ones used for the directed version of the problem – Max ATSP – in [20]. We are convinced that they will prove useful for other problems related with TSP, cycle covers or matchings.

The main result of the paper is

Theorem 1. *There exists a $\frac{4}{5}$-approximation algorithm for Max TSP. Its running time is $O(n^3)$ if the graph has an even number of vertices and $O(n^5)$ otherwise.*

Algorithm 1. A $\frac{4}{5}$-approximation for Max TSP

1: $C_{max} \leftarrow$ a maximum-weight cycle cover of G
2: $M \leftarrow$ a maximum-weight perfect matching in G
3: $G_1 \leftarrow C_{max} \uplus C_{max} \uplus M$
4: path-3-color G_1 with colors of $\mathcal{K}_3 = \{1, 2, 3\}$ leaving kites and edges of M incident to kites uncolored. ▷ Section 2
5: $C_2 \leftarrow$ a maximum-weight *relaxed cycle cover improving C_{max} with respect to M*. ▷ Section 3
6: $G_2 \leftarrow C_2 \uplus M$
7: $F_1 \subset C_{max}, F_2 \subset C_2 \leftarrow$ sets of edges such that the multigraph $G_1' = G_1 \backslash F_1 \cup F_2$ is path-3-colorable and $G_2' = G_2 \backslash F_2 \cup F_1$ is path-2-colorable. ▷ Lemma 5
8: Path-2-color G_2' with colors of $\mathcal{K}_2 = \{4, 5\}$. ▷ Full version of the paper
9: Extend the partial path-3-coloring of G_1 to the complete path-3-coloring of G_1'. ▷ Full version of the paper
10: Choose the heaviest color class $k \in \mathcal{K}_3 \cup \mathcal{K}_2$. Complete the disjoint paths of color k into a traveling salesman tour in an arbitrary way.

All missing proofs are contained in the full version of this paper [9].

2 Path-3-Coloring of G_1

We compute a maximum weight cycle cover C_{max} of a given complete undirected graph $G = (V, E)$ and a maximum weight perfect matching M of G. We are going to call cycles of length i, i.e., consisting of i edges ***i*-cycles**. Also sometimes 3-cycles will be called **triangles** and 4-cycles – **squares**. The multigraph G_1 consists of two copies of C_{max} and one copy of M. We want to color each edge of G_1 with one of three colors of $\mathcal{K}_3 = \{1, 2, 3\}$ so that each color class consists of vertex-disjoint paths. The *graph* G_1 is a subgraph of the *multigraph* G_1 that contains an edge (u, v) iff the multigraph G_1 contains an edge between u and v. The path-3-coloring of G_1 can be equivalently defined as coloring each edge of (the graph) G_1 with the number of colors equal to the number of copies contained in the multigraph G_1. From this time on, unless stated otherwise, G_1 denotes a graph and not a multigraph.

We say that a colored edge e of G_1 is **safe** if no matter how we color the so far uncolored edges of G_1 e is guaranteed not to belong to any monochromatic cycle of G_1. An edge e of M is said to be **external** if its two endpoints belong to two different cycles of C_{max}. Otherwise, e is **internal**. We say that an edge e is incident to a cycle c if it is incident to at least one vertex of c.

We prove the following useful lemma.

Lemma 1. *Consider a partial coloring of G_1. Let c be any cycle of C_{max} such that for each color $k \in \mathcal{K}_3$ there exists an edge of M incident to c that is colored k. Then we can color c so that each edge of c and each edge incident to one of the edges of c is safe.*

Proof. The proposed procedure of coloring c is as follows.

If there exists an edge of c that also belongs to M, we color it with all three colors of $\mathcal{K}_3$. For each uncolored edge of M incident to c, we color it with an arbitrary color of $\mathcal{K}_3$. Next, we orient the edges of c (in any of the two ways) so that c becomes a directed cycle c. Let $e = (u, v)$ be any uncolored edge of c oriented from u to v. Then, there exists an edge e' of M incident to u. If e' is contained in c, then we color e with any two colors of $\mathcal{K}_3$. Otherwise e' is colored with some color k of $\mathcal{K}_3$. Then we color e with the two colors belonging to $\mathcal{K}_3 \setminus k$. First, no vertex of c has three incident edges colored with the same color, as for each vertex its outgoing edge is colored with different colors than an incident matching edge. Second, as for each color $k \in \mathcal{K}_3$ there is a matching edge incident to c colored with k, there exists an edge of c that is not colored k, thus c does not belong to any color class, i.e. there exists no color $k \in \mathcal{K}_3$ such that each edge of c is colored with k. Let us consider now any edge $e = (u, v)$ of M incident to some edge of c and not belonging to c. The edge e is colored with some color k. Suppose also that vertex u belongs to c (v may or may not belong to c.) Let u' be any other vertex of c such that some edge of $M \setminus C_{max}$ colored k is incident to it (u' may be equal to v if e is internal). To show that e is safe, it suffices to show that there exists no path consisting of edges of $c \cup M$ that connects u and u' and whose every edge is colored k. However, by the way we color edges of c we know that the outgoing edges of u and u' are not colored with k because of the way we oriented the cycle, there is no path connecting u and u' contained in c that starts and ends with incoming edge. □

For each cycle c of C_{max} we define its **degree of flexibility** denoted as $flex(c)$ and its **colorfulness**, denoted as $col(c)$. The degree of flexibility of a cycle c is the number of internal edges of M incident to c and the colorfulness of c is the number of colors of $\mathcal{K}_3$ that are used for coloring the external edges of M incident to c.

From Lemma 1 we can easily derive.

Lemma 2. *If a cycle c of C_{max} is such that $flex(c) + col(c) \geq 3$, then we can color c so that each edge of c and each edge incident to one of the edges of c is safe.*

Sometimes, even if a cycle c of C_{max} is such that $flex(c) + col(c) < 3$, we can color the edges of c so that each of them is safe. For example, suppose that c is a square consisting of edges $e_1, \ldots, e_4$ and there are four external edges of M incident to c, all colored 1. Suppose also that each external edge incident to c is already safe. Then we can color e_1 with 1 and 2, e_3 with 1 and 3 and both e_2 and e_4 with 2 and 3. We can notice that e_1 is guaranteed not to belong to a cycle colored 1 because external edges incident to e_1 are colored 1 and are safe. Analogously, we can easily check that each other edge of c is safe. However, for example, a triangle t of C_{max} that has three external edges of M incident to it, all colored with the same color of $\mathcal{K}_3$, cannot be colored in such a way that it does not contain a monochromatic cycle.

Consider a cycle c of C_{max} such that every external edge of M incident to c is colored. We say that c is **nice** if and only if (1) $flex(c) + col(c) \geq 3$ or (2) c contains at least $3 - flex(c) - col(c)$ vertex-disjoint edges, each of which has the property that it has exactly two incident external edges of M and the two external edges of M incident to it are colored with the same color of $\mathcal{K}_3$ or (3) c is a square such that $flex(c) = 1$.

Otherwise we say that c is **blocked**. We can see that a cycle c of C_{max} is blocked if and only if

- c is a triangle and all external edges of M incident to c are colored with the same color of $\mathcal{K}_3$,
- c is a square with two internal edges of M incident to it ($flex(c) = 2$),
- c is a cycle of even length, $flex(c) = 0$ and there exist two colors $k_1, k_2 \in \mathcal{K}_3$ such that external edges of M incident to c are colored alternately with k_1 and k_2.

Among blocked cycles we distinguish kites. We say that a cycle c is a **kite** if it is a triangle such that $flex(c) = 1$ and then we call it a **3-kite** or it is a square, whose two edges belong to M (so $flex(c) = 2$) - called a **4-kite**. We can assume that a square with two diagonals in M will not occur, as diagonals are heavier than any two opposite edges in this square (as they are in M), so they would be included in C_{max}. A cycle of C_{max} which is not a kite is said to be **non-kite**.

Now, we are ready to state the algorithm for path-3-coloring G_1. It is presented as Algorithm 2.

Lemma 3. *Let c be a non-kite cycle of C_{max} that at some step of Algorithm Color G_1 has the fewest uncolored external edges incident to it. Then, it is always*

Algorithm 2. Color G_1

1: **while** $\exists$ an uncolored non-kite cycle of C_{max} **do**
2: $\quad \mathcal{C} \leftarrow$ a non-kite uncolored cycle of C_{max} with the fewest uncolored external edges incident to it.
3: $\quad$ Color uncolored external edges incident to $\mathcal{C}$ so that no other cycle of C_{max} becomes blocked and either $flex(\mathcal{C}) + col(\mathcal{C}) \geq 3$ or its external matching edges are all safe. ▷ Lemma 3
4: $\quad$ Color $\mathcal{C}$ and internal edges incident to it in such a way, that each edge incident to $\mathcal{C}$ is safe. ▷ Lemma 4
5: **end while**

possible to color all uncolored external edges incident to c so that no non-kite cycle of C_{max} *becomes blocked. Moreover, if c has at least two uncolored external edges incident to c then, additionally, it is always possible to do it in such a way that* $flex(c) + col(c) \geq 3$*. If c has exactly one uncolored external edge e of M incident to it, then we can color e so that* $flex(c) + col(c) \geq 3$ *or so that e is safe.*

From the above lemma we get

Corollary 1. *After all external edges are colored, each of them is incident to a cycle c of* C_{max} *such that* $flex(c) + col(c) \geq 3$ *or is safe.*

Lemma 4. *Let c be a nice cycle of* C_{max} *whose all incident external edges of M are already colored and safe. Then it is always possible to color c and internal edges incident to c in such a way that each edge incident to c is safe.*

3 A Cycle Cover Improving C_{max} with Respect to M

Since C_{max} may contain kites, we may not be able to path-3-color G_1. Therefore, our next aim is to compute another cycle cover C_2 of G such that it does not contain any kite of C_{max} and whose weight is an upper bound on OPT. Since computing such C_2 may be hard, we relax the notion of a cycle cover and allow C_2 to contain **half-edges**. A half-edge of the edge e is, informally speaking, a half of the edge e that contains exactly one of the endpoints of e. Let us also point out here that C_2 may contain kites which do not belong to C_{max}.

We say that an edge (u, v) is a **kite-edge** if u and v belong to the same kite (so it can be a side of a kite, but also a diagonal of a 4-kite). Every kite-edge $e = (u, v)$ is split into two half edges (u, x_e) and (x_e, v), each carrying half of the weight of e. The graph $\tilde{G} = (\tilde{V}, \tilde{E})$ will be G with kite-edges replaced with half-edges.

Definition 1. *A* ***relaxed cycle cover improving*** C_{max} ***with respect to*** M *is a subset* $\tilde{C} \subseteq \tilde{E}$ *such that*

(i) each vertex in V *has exactly two incident edges in* $\tilde{C}$*;*

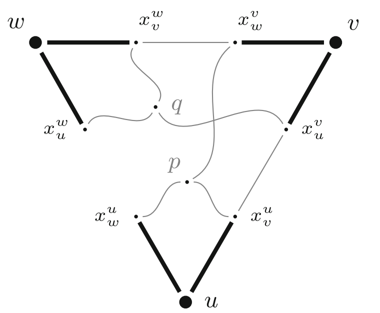
(ii) for each 3-kite $\mathcal{T}$ of C_{max} the number of half-edges of kite-edges of $\mathcal{T}$ contained in $\tilde{C}$ is even and not greater than four;

(iii) for each 4-kite $\mathcal{S}$ of C_{max} the number of half-edges of kite-edges of $\mathcal{S}$ contained in $\tilde{C}$ is even and not greater than six.

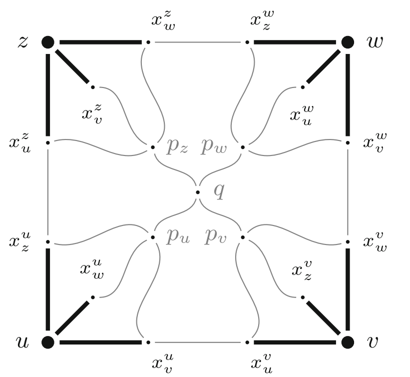
To compute a *relaxed cycle cover C_2 improving C_{max}* with respect to M we construct the following graph $G' = (V', E')$ (by replacing kites with gadgets). The set of vertices V' is a superset of the set of verices $V(G)$. For each kite-edge (u, v) of G we add two vertices x^u_v, x^v_u to V' and edges $(u, x^u_v), (x^v_u, v)$ to E' (these represent the half-edges). For each kite-edge (u, v) which is not a diagonal of a 4-kite or one of the non-matching edges in 3-kite (for each 3-kite we choose arbitrarily one of them) we add also an edge (x^u_v, x^v_u). The edge (x^u_v, x^v_u) has weight 0 in G' and each of the edges $(u, x^u_v), (x^v_u, v)$ has weight equal to $\frac{1}{2}w(u, v)$. Each of the vertices x^u_v, x^v_u is called *a splitting vertex of the edge* (u, v).

For each 3-kite $\mathcal{T}$ on vertices u, v, w we add two vertices $p^{\mathcal{T}}, q^{\mathcal{T}}$ to V'. Let's assume that u is incident to external edge of M and that (x^u_w, x^w_u) was the side not added to G'. The vertex $p^{\mathcal{T}}$ is connected to the splitting vertices of edges of $\mathcal{T}$ that are neighbors of u, i.e. to vertices x^u_v, x^u_w and to vertex x^v_w. The vertex $q^{\mathcal{T}}$ is connected to every other splitting vertex of $\mathcal{T}$, i.e. x^w_u, x^w_v, x^v_u. All edges incident to vertices $p^{\mathcal{T}}, q^{\mathcal{T}}$ have weight 0 in G'.

For each 4-kite $\mathcal{S}$ of C_{max} on vertices u, v, w, z we add five vertices $p^{\mathcal{S}}_u, p^{\mathcal{S}}_v, p^{\mathcal{S}}_w, p^{\mathcal{S}}_z, q^{\mathcal{S}}$ to V'. Vertex $p^{\mathcal{S}}_u$ is connected to the splitting vertices of edges of $\mathcal{S}$ that are neighbors of u, i.e. to vertices x^u_v, x^u_w, x^u_z. Vertices $p^{\mathcal{S}}_v, p^{\mathcal{S}}_w, p^{\mathcal{S}}_z$ are con-



(a) $b(u) = b(v) = b(w) = 2$, $\forall k, l \in \{u, v, w\}$ $b(x^k_l) = 1$, $b(p) = b(q) = 1$



(b) $b(u) = b(v) = b(w) = b(z) = 2$, $\forall k, l \in \{u, v, w, z\}$ $b(x^l_k) = 1$, $b(p_u) = b(p_v) = b(p_w) = b(p_z) = b(q) = 2$

Fig. 1. Gadgets for 3-kites **(a)** and 4-kites **(b)** of G_1 in graph G. Half-edges corresponding to the original edges are thickened, the auxiliary edges are thin. Original vertices (thick dots) are connected with all the other original vertices of graph G. The auxiliary vertices have no connections outside of the gadget. The figures are subtitled with the specifications of $b(v)$ values for different vertices. For a vertex t with $b(t) = i$, the resulting b-matching will contain exactly i edges ending in t.

nected analogously. Vertex $q^{\mathcal{S}}$ is connected to vertices $p_u^{\mathcal{S}}, p_v^{\mathcal{S}}, p_w^{\mathcal{S}}, p_z^{\mathcal{S}}$. All edges incident to vertices $p_u^{\mathcal{S}}, p_v^{\mathcal{S}}, p_w^{\mathcal{S}}, p_z^{\mathcal{S}}, q^{\mathcal{S}}$ have weight 0.

For each edge (u, v) of G that is not a kite-edge we add it to E' with weight $w(u, v)$.

We reduce the problem of computing a relaxed cycle cover improving C_{max} with respect to M, to the problem of computing a perfect b-matching in the graph G'. We define the function $b : V' \rightarrow \mathbb{N}$ in the following way. For each vertex $v \in V$ we set $b(v) = 2$. For each splitting vertex v' of some problematic edge we set $b(v') = 1$. For all vertices $p^{\mathcal{T}}$ and $q^{\mathcal{T}}$, where $\mathcal{T}$ denotes a 3-kite of C_{max} we have $b(p^{\mathcal{T}}) = b(q^{\mathcal{T}}) = 1$. For all vertices $p_u^{\mathcal{S}}$ and $q^{\mathcal{S}}$, where $\mathcal{S}$ denotes a 4-kite of C_{max} and u one of its vertices we have $b(p_u^{\mathcal{S}}) = b(q^{\mathcal{S}}) = 2$ (Fig. 1).

Theorem 2. *Any perfect b-matching of G' yields a relaxed cycle cover C_2 improving C_{max} with respect to M. A maximum weight perfect b-matching of G' yields a relaxed cycle cover C_2 improving C_{max} with respect to M such that $w(C_2) \geq OPT$.*

4 Exchange Sets F_1, F_2 and Path-2-Coloring of G_2'

The multigraph G_2 is constructed from one copy of the relaxed cycle cover C_2 and one copy of the maximum weight perfect matching M. Since C_2 may contain half-edges and we want G_2 to contain only edges of G, for each half-edge of edge (u, v) contained in C_2, we will either include the whole edge (u, v) in G_2 or not include it at all. While doing so we have to ensure that the total weight of the constructed multigraph G_2 is at least $\frac{3}{2}OPT$.

The main idea behind deciding which half-edges are extended to full edges and included in G_2 is that we construct two sets Z_1 and Z_2 – for each kite in G_1 we distribute its edges corresponding to the half-edges so that half of them go into the set Z_1 and the other half to Z_2. (Note that by Definition 1 each kite in G_1 contains an even number of half-edges in C_2.) Let $I(C_2)$ denote the set consisting of whole edges of G contained in C_2. This way $w(C_2) = w(I(C_2)) + \frac{1}{2}(w(Z_1) + w(Z_2))$. Next, let Z denote the one of the sets Z_1 and Z_2 with larger weight. Then G_2 is defined as a multiset consisting of edges of M, edges of $I(C_2)$ and edges of Z. We reach the following

Fact 1. *The total weight of the constructed multigraph G_2 is at least $\frac{3}{2}OPT$.*

Proof. The weight of M is at least $\frac{1}{2}OPT$. The weight of $w(C_2) = w(I(C_2)) + \frac{1}{2}(w(Z_1) + w(Z_2))$ is at least OPT. Since $w(Z) = \max\{w(Z_1), w(Z_2)\}$, we conclude that $w(I(C_2)) + w(Z) \geq w(C_2)$. □

If C_{max} contains at least one kite, G_1 is non-path-3-colorable. We can however notice, that if we remove one edge from each kite in the multigraph G_1, then the obtained multigraph is path-3-colorable.

If we manage to construct a set F_1 containing one edge from each kite, such that additionally the multigraph $G_2 \cup F_1$ is path-2-colorable, then we have a

$\frac{4}{5}$-approximation of Max TSP immediately. Since computing such F_1 may be difficult, we allow, in turn, certain edges of C_2 to be removed from G_2 and added to G_1. Thus, roughly, our goal is to compute such disjoint sets F_1, F_2 that:

1. $F_1 \subset C_{max}$ contains at least one edge of each kite;
2. $F_2 \subset I(C_2)$ contains one edge per each kite $\mathcal{C} \in C_{max}$;
3. the multigraph $G_1' = G_1 \setminus F_1 \cup F_2$ is path-3-colorable;
4. the multigraph $G_2' = G_2 \setminus F_2 \cup F_1$ is path-2-colorable.

Let F_1 and F_2 be two sets of edges that satisfy properties 1. and 2. of the above. Then the set of edges $C_2' = (I(C_2) \cup Z \cup F_1) \setminus F_2$ can be partitioned into *cycles and paths of* G_2', where G_2' denotes the resulting multigraph $G_2 \setminus F_2 \cup F_1$. The partition of C_2' into cycles and paths is carried out in such a way that two incident edges of C_2' belonging to a common path or cycle of C_2, belong also to a common path or cycle of C_2' (and G_2'). Also, the partition is maximal, i.e., we can't add any edge e of C_2' to any path $\mathcal{P}$ of G_2' so that $\mathcal{P} \cup \{e\}$ is also a path or cycle of G_2'.

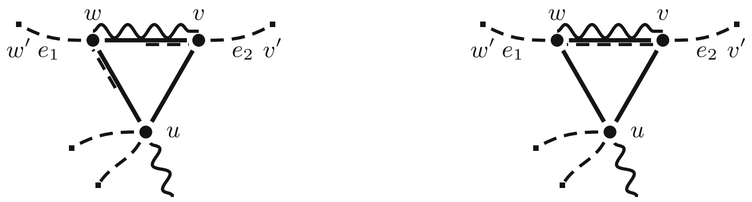
We say that e is a **double edge** of G_2' if the multigraph G_2' contains two copies of e. In any path-2-coloring of G_2' every double edge must have both colors of $\mathcal{K}_2$ assigned to it.

We observe that in order for G_2' to be path-2-colorable, we have to guarantee that there does not exist a cycle $\mathcal{C}$ of G_2' of odd length l that has l incident double edges. When every two consecutive edges of $\mathcal{C}$ are incident to some double edge, they must be assigned different colors of $\mathcal{K}_2$ and if the length of $\mathcal{C}$ is odd, this is clearly impossible. The way to avoid this is to choose one edge of each such potential cycle and add it to F_2.

We say that a path $\mathcal{P}$ of G_2' beginning at w and ending at v is **amenable** if

(i) neither v nor w has degree 4 in G_2', or
(ii) v has degree 4, w has degree smaller than 4 and $\mathcal{P}$ ends with a double edge, the last-but-one edge of $\mathcal{P}$ is a double edge or the last-but-one and the last-but-three vertices in $\mathcal{P}$ are matched in M.

It turns out that G_2' that does not contain odd cycles described above and whose every path is amenable is path-2-colorable — we show it in the full version of the paper. To facilitate the construction of G_2', whose every path is amenable and to ensure that F_1 and F_2 have certain other useful properties we create two opposite orientations of $I(C_2)$: D_1 and D_2. In each of these orientations $I(C_2)$ contains directed cycles and paths and each kite has the same number of incoming and outgoing edges. This can be achieved by pairing the endpoints of paths ending at the same kite and combining them. For example, let us consider a 3-kite in Fig. 2. C_2 contains half-edges $h_1 = (w, x_{\{u,w\}})$ and $h_2 = (v, x_{\{v,w\}})$ of a certain 3-kite $\mathcal{T}$, so for the purpose of orientation we replace h_1 and h_2 with an edge (v, w). Then, if for example C_2 contains edges $e_1 = (w', w), e_2 = (v', v)$ in the orientation in which e_1 is directed from w' to w, the edge e_2 is directed from v to v' and vice versa.



(a) Edges of C_2 incident to some 3-kite (b) Graph used for purpose of orienting paths and cycles.

Fig. 2. Example of creating orientations D_1 and D_2

Apart from the whole edges C_2 also contains the half-edges. Let $H(C_2)$ denote the set of the edges of G such that C_2 contains exactly one half-edge of each of these edges. We would like to partition $H(C_2)$ into two sets Z_1, Z_2 so that for each kite c half of the edges of $H(C_2)$ is contained in Z_1 and the other half in Z_2. We associate Z_1 with orientation D_1 and Z_2 with orientation D_2. Thus, we assume that D_1 contains Z_1, with the edges of Z_1 being oriented in a consistent way with the edges of $I(C_2)$ under orientation D_1, and D_2 contains Z_2, with its edges being oriented accordingly. Depending on which of the sets Z_1, Z_2 has bigger weight, we either choose the orientation D_1 or D_2. Hence, from now on, we assume that the edges of $I(C_2) \cup Z$ are directed.

For example, for the triangle $\mathcal{T}$ described above (and presented in Fig. 2), the partition may be as follows. If e_1 is oriented from w to w' in D_1, then we assume that h_1 is in Z_1 and h_2 is in Z_2. Therefore, we can guarantee, that if h_1 is in Z, e_1 is oriented from v to v'.

The exact details of the construction of Z_1 and Z_2 are given in the proof of Lemma 5.

Lemma 5. *It is possible to compute the sets F_1, F_2 such that they, and the resulting G'_2 satisfy:*

1. $F_1 \subset C_{max} \setminus ((Z \cup I(C_2)) \cap M)$;
2. $F_2 \subseteq I(C_2) \cup Z$;
3. *for each kite $\mathcal{C}$, (i) the set F_1 contains exactly one edge of $\mathcal{C}$ and the set F_2 contains zero edges of $\mathcal{C}$ or (ii) (it can happen only for 4-kites) the set F_1 contains exactly two edges of $\mathcal{C}$ and the set F_2 contains one edge of $\mathcal{C} \setminus M$;*
4. *for each kite $\mathcal{C}$ the set F_2 contains exactly one outgoing edge of $\mathcal{C}$;*
5. *for each kite $\mathcal{C}$ and each vertex v of $\mathcal{C}$ the number of edges of F_2 incident to v is at most greater by one than the number of edges of F_1 incident to v;*
6. *there exists no cycle of G'_2 of odd length l that has l double edges incident to it;*
7. *each path of G'_2 is amenable.*

The property 1 of this lemma guarantees that G'_2 does not contain more than two copies of any edge. It is shown in the full version of the paper that properties

6. and 7. are essentially sufficient for the multigraph G_2' to be path-2-colorable. Properties 4 and 5 will be helpful in finding a path-3-coloring of G_1'. Property 5 ensures that no vertex v has six incident edges in G_1'.

5 Summary

After the construction and path-2-coloring of G_2' we are presented with the task of extending the partial path-3-coloring of G_1 to the complete path-3-coloring of G_1'. In particular, we have to color the edges of kites and edges of F_2 that have been added during the construction of G_2'. This part of the algorithm is described in the full version of the paper.

The presented algorithm works for graphs with an even number of vertices. If the number of vertices of a given graph is odd, we proceed as follows. We select a vertex $v \in V$ arbitrarily. Then we guess its predecessor u and successor t in the optimal solution ($O(n^2)$ guesses). For each guess we replace the vertex v with two new vertices v_1, v_2 (so we have an even number of vertices). The edge (u, v_1) has weight $w(u, v)$, the edge (t, v_2) has weight $w(t, v)$ and all remaining edges incident to v_1 or v_2 have weight equal to 0. Then we run our Algorithm 1 on these instances. The approximation ratio of $\frac{4}{5}$ holds, because the computed solution can be always transformed into a tour in the original graph of at least the same weight, and the optimal tour is certainly present among the guesses.

References

1. Arkin, E.M., Chiang, Y., Mitchell, J.S.B., Skiena, S., Yang, T.: On the maximum scatter TSP (extended abstract). In: Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 211–220 (1997)
2. Barvinok, A.I., Fekete, S.P., Johnson, D.S., Tamir, A., Woeginger, G.J., Woodroofe, R.: The geometric maximum traveling salesman problem. J. ACM **50**(5), 641–664 (2003)
3. Barvinok, A., Johnson, D.S., Woeginger, G.J., Woodroofe, R.: The maximum traveling salesman problem under polyhedral norms. In: Bixby, R.E., Boyd, E.A., Ríos-Mercado, R.Z. (eds.) IPCO 1998. LNCS, vol. 1412, pp. 195–201. Springer, Heidelberg (1998). doi:10.1007/3-540-69346-7_15
4. Bhatia, R.: Private communication
5. Chalasani, P., Motwani, R.: Approximating capacitated routing and delivery problems. SIAM J. Comput. **28**(6), 2133–2149 (1999)
6. Chen, Z.Z., Okamoto, Y., Wang, L.: Improved deterministic approximation algorithms for max TSP. Inf. Process. Lett. **95**(2), 333–342 (2005)
7. Chen, Z.-Z., Wang, L.: An improved approximation algorithm for the bandpass-2 problem. In: Lin, G. (ed.) COCOA 2012. LNCS, vol. 7402, pp. 188–199. Springer, Heidelberg (2012). doi:10.1007/978-3-642-31770-5_17
8. Chiang, Y.J.: New approximation results for the maximum scatter tsp. Algorithmica **41**(4), 309–341 (2005)
9. Dudycz, S., Marcinkowski, J., Paluch, K.E., Rybicki, B.: A 4/5 - approximation algorithm for the maximum traveling salesman problem. CoRR abs/1512.09236 (2015). http://arxiv.org/abs/1512.09236

10. Fisher, M.L., Nemhauser, G.L., Wolsey, L.A.: An analysis of approximations for finding a maximum weight hamiltonian circuit. Oper. Res. **27**(4), 799–809 (1979)
11. Hassin, R., Levin, A., Rubinstein, S.: Approximation algorithms for maximum latency and partial cycle cover. Discrete Optim. **6**(2), 197–205 (2009)
12. Hassin, R., Rubinstein, S.: An approximation algorithm for the maximum traveling salesman problem. Inf. Process. Lett. **67**(3), 125–130 (1998)
13. Hassin, R., Rubinstein, S.: Better approximations for max TSP. Inf. Process. Lett. **75**(4), 181–186 (2000)
14. Hassin, R., Rubinstein, S.: An approximation algorithm for maximum triangle packing. In: Albers, S., Radzik, T. (eds.) ESA 2004. LNCS, vol. 3221, pp. 403–413. Springer, Heidelberg (2004). doi:10.1007/978-3-540-30140-0_37
15. Kaplan, H., Lewenstein, M., Shafrir, N., Sviridenko, M.: Approximation algorithms for asymmetric tsp by decomposing directed regular multigraphs. In: 44th Symposium on Foundations of Computer Science (FOCS 2003) (2003)
16. Kosaraju, S.R., Park, J.K., Stein, C.: Long tours and short superstrings. In: 35th Annual IEEE Symposium on Foundations of Computer Science (FOCS) (1994)
17. Kowalik, Ł., Mucha, M.: 35/44-approximation for asymmetric maximum TSP with triangle inequality. In: Dehne, F., Sack, J.-R., Zeh, N. (eds.) WADS 2007. LNCS, vol. 4619, pp. 589–600. Springer, Heidelberg (2007). doi:10.1007/978-3-540-73951-7_51
18. Kowalik, Ł., Mucha, M.: Deterministic 7/8-approximation for the metric maximum TSP. In: Goel, A., Jansen, K., Rolim, J.D.P., Rubinfeld, R. (eds.) APPROX/RANDOM -2008. LNCS, vol. 5171, pp. 132–145. Springer, Heidelberg (2008). doi:10.1007/978-3-540-85363-3_11
19. Monnot, J.: Approximation algorithms for the maximum hamiltonian path problem with specified endpoint(s). Eur. J. Oper. Res. **161**(3), 721–735 (2005)
20. Paluch, K.E.: Better approximation algorithms for maximum asymmetric traveling salesman and shortest superstring. CoRR (2014)
21. Paluch, K.E., Elbassioni, K.M., van Zuylen, A.: Simpler approximation of the maximum asymmetric traveling salesman problem. In: 29th International Symposium on Theoretical Aspects of Computer Science, STACS (2012)
22. Paluch, K., Mucha, M., Mądry, A.: A 7/9 - approximation algorithm for the maximum traveling salesman problem. In: Dinur, I., Jansen, K., Naor, J., Rolim, J. (eds.) APPROX/RANDOM -2009. LNCS, vol. 5687, pp. 298–311. Springer, Heidelberg (2009). doi:10.1007/978-3-642-03685-9_23
23. Papadimitriou, C.H., Yannakakis, M.: The traveling salesman problem with distances one and two. Math. Oper. Res. **18**(1), 1–11 (1993)
24. Schrijver, A.: Nonbipartite matching and covering. In: Combinatorial Optimization, vol. A, pp. 520–561. Springer (2003)
25. Serdyukov, A.I.: An algorithm with an estimate for the traveling salesman problem of maximum. Upravlyaemye Sistemy **25**, 80–86 (1984) (in Russian)
26. Sichen, Z., Zhao, L., Liang, Y., Zamani, M., Patro, R., Chowdhury, R., Arkin, E.M., Mitchell, J.S.B., Skiena, S.: Optimizing read reversals for sequence compression. In: Pop, M., Touzet, H. (eds.) WABI 2015. LNCS, vol. 9289, pp. 189–202. Springer, Heidelberg (2015). doi:10.1007/978-3-662-48221-6_14
27. Tong, W., Goebel, R., Liu, T., Lin, G.: Approximation algorithms for the maximum multiple RNA interaction problem. In: Widmayer, P., Xu, Y., Zhu, B. (eds.) COCOA 2013. LNCS, vol. 8287, pp. 49–59. Springer, Cham (2013). doi:10.1007/978-3-319-03780-6_5