A near-optimal approximation algorithm for Asymmetric TSP on embedded graphs

Jeff Erickson[*] Anastasios Sidiropoulos[†]

Abstract

We present a near-optimal polynomial-time approximation algorithm for the asymmetric traveling salesman problem for graphs of bounded orientable or non-orientable genus. Our algorithm achieves an approximation factor of $O(f(g))$ on graphs with genus g, where $f(n)$ is the best approximation factor achievable in polynomial time on arbitrary n-vertex graphs. In particular, the $O(\log n/\log\log n)$-approximation algorithm for general graphs by Asadpour *et al.* [SODA 2010] immediately implies an $O(\log g/\log\log g)$-approximation algorithm for genus-g graphs. Our result improves the $O(\sqrt{g}\log g)$-approximation algorithm of Oveis Gharan and Saberi [SODA 2011], which applies only to graphs with *orientable* genus g; ours is the first approximation algorithm for graphs with bounded no-orientable genus. Moreover, using recent progress on approximating the genus of a graph, our $O(\log g/\log\log g)$-approximation can be implemented even without an embedding when the input graph has bounded degree. In contrast, the $O(\sqrt{g}\log g)$-approximation algorithm of Oveis Gharan and Saberi requires a genus-g embedding as part of the input.

[*]Department of Computer Science, University of Illinois at Urbana-Champaign; http://www.cs.uiuc.edu/~jeffe; jeffe@illinois.edu. Supported in part by NSF grant CCF-0915519.

[†]Department of Computer Science, University of Illinois at Urbana-Champaign; http://www.sidiropoulos.org/; sidiropo@illinois.edu. Supported in part by David and Lucille Packard Fellowship and by NSF grants CCF-0915984 and CCF-0915519.

1 Introduction

The asymmetric traveling salesman problem is one of the most fundamental and well studied problems in combinatorial optimization. An instance of ATSP consists of a pair directed graph $\vec{G} = (V, A)$ and a (not necessarily symmetric) cost function $c \colon A \to \mathbb{R}^+$ that satisfies the triangle inequality $c(x\to z) \leq c(x\to y) + c(y\to z)$ for all vertices x, y, and z. The goal is to find a closed walk of minimum cost that visits every vertex at least once.

Recently, Asadpour *et al.* [1] gave an $O(\log n/\log\log n)$-approximation algorithm for ATSP, improving the nearly 30 year old $O(\log n)$-approximating of Frieze *et al.* [4]. Building on this breakthrough, Oveis Gharan and Saberi [8] gave a $O(\sqrt{g}\log g)$-approximation for graphs of orientable genus g. We refer the reader to these previous papers [1, 8] and the references therein for a more detailed overview of the rich history of the ATSP problem.

We obtain an improved approximation algorithm for graph of genus g, by generalizing the upper bound of Asadpour *et al.* [1] for general graphs. Intuitively, we describe a general reduction from instances of ATSP in arbitrarily large genus-g graphs to instances of ATSP with only g vertices, which introduces only constant approximation error. The following theorem summarizes our main result.

Theorem 1.1. *If there is a polynomial-time $f(n)$-approximation for ATSP on n-vertex graphs, then there is polynomial-time $O(f(g))$-approximation for ATSP on graphs embedded into a surface of either orientable or non-orientable genus g.*

The $O(\log n/\log\log n)$-approximation algorithm of Asadpour *et al.* [1] for arbitrary n-vertex graphs immediately implies the following corollary.

Corollary 1.2. *There is a polynomial-time $O(\log g/\log\log g)$-approximation algorithm for ATSP, on graphs embedded into a surface of either orientable, or non-orientable genus g.*

We remark that Oveis Gharan and Saberi's algorithm [8] requires the input graph to be embedded on an orientable surface, while our algorithm works for both orientable and non-orientable surfaces. Our algorithm is the first to achieve a constant-factor approximation for graphs of bounded non-orientable genus.

Recently, Chekuri and Sidiropoulos [2] obtained a polynomial-time algorithm to approximate the genus of bounded-degree graphs. Specifically, given a graph G with bounded maximum degree, their algorithm outputs an embedding of G into a surface of orientable (resp. non-orientable) genus g^α, where g is the orientable (resp. non-orientable) genus of G and α is a constant. Combining this result with Theorem 1.1, we immediately obtain an $O(\log g/\log\log g)$-approximation algorithm for ATSP on bounded-degree graphs of genus g, even when a embedding of the input graph is not given as part of the input.

Corollary 1.3. *There is a polynomial-time $O(\log g/\log\log g)$-approximation algorithm for ATSP, on graphs of either orientable or non-orientable genus g, even when an embedding of the graph is not given as part of the input.*

In contrast, if we apply the same approach to the $O(\sqrt{g}\log g)$-approximation algorithm of Oveis Gharan and Saberi, the resulting approximation guarantee deteriorates to $O(g^\beta)$, for some constant $\beta > 6$.

1.1 A high-level description of the algorithm

Previous work on rounding the Held-Karp LP. Our algorithm is inspired by and uses many fundamental ideas from Asadpour *et al.* [1] and Oveis Gharan and Saberi [8]. We begin our description by recalling some key steps from these algorithms.

Both of these algorithms use the classical linear programming relaxation of ATSP introduced by Held and Karp [5]. Intuitively, a feasible solution to the Held-Karp LP assigns a weight to every arc, so that the total weight crossing every cut (in either direction) is at least 1, and the total weight of the edges entering each vertex is 1; see Section 2 for a formal definition. The main tool introduced by Asadpour *et al.* [1] for rounding such a fractional solution is the notion of a *thin spanning tree*. Roughly speaking, an undirected spanning tree T is (α, s)-thin if it satisfies two conditions:

(i) For every cut, the total number of edges in T crossing the cut is at most α times the total fractional cost of the cut.

(ii) The total cost of the tree is at most s times the total cost of the fractional solution.

Given such an (α, s)-thin spanning tree, one can obtain a tour of total cost $O((\alpha + s) \cdot \mathsf{OPT})$, where OPT is the cost of the minimum TSP tour, via a careful application of Hoffman's circulation theorem. Asadpour *et al.* [1] describe a randomized algorithm that constructs a $(O(\log\log n), 2)$-thin spanning tree with high probability for any graph; Oveis Gharan and Saberi [8] show how to compute a $(O(\sqrt{g}\log g), O(\sqrt{g}\log g))$-thin spanning tree for any graph embedded on an orientable surface of genus g.

The forest for the trees. We depart slightly from the previous paradigm by using *thin spanning forests* instead of thin spanning trees. Given a graph of genus g, we show how to construct a $(O(1), O(1))$-thin spanning forest with at most g connected components. Using a modified application of Hoffman's circulation theorem, we can transform this thin forest into a collection of g closed walks $W_1, \ldots, W_g$ that collectively visit all vertices in the graph and that have total cost $O(\mathsf{OPT})$. We then apply an idea from Frieze *et al.* [4] to merge these g walks into a single closed walk. We choose an arbitrary vertex from each walk w_i and form a smaller ATSP instance on a graph with only these g vertices. We solve this smaller instance using the algorithm for general graphs, obtaining a tour C. Finally, we merge the walks $C, W_1, \ldots, W_g$ and shortcut the resulting closed walk to obtain the approximate solution to the original ATSP instance.

How can we find a thin forest? The main technical part of our algorithm is the computation of a thin spanning forest with few connected components. This is done by introducing a certain structure that we call a *ribbon*, which is intuitively a maximal set of parallel edges that are contained inside a disk in the surface. We show that unless the graph has at most g vertices, we can find such a ribbon having large total fractional cost. Our algorithm repeatedly *contracts* the ribbon with largest fractional cost, until we arrive at a graph with at most g vertices. Every vertex in the contracted graph corresponds to a connected component of ribbons in the original graph. The fact that every ribbon has large fractional cost allows us to find a spanning tree in each such component, so that the resulting spanning forest is thin.

1.2 Preliminaries

We give a brief overview of some of the notation that we will use. For a more detailed background in topological graph theory, we refer the reader to Mohar and Thomassen [7].

A *surface* is a compact 2-dimensional manifold without boundary. A surface is *orientable* if it can be embedded in $\mathbb{R}^3$, and *non-orientable* otherwise. An *embedding* of an undirected graph G in a surface $\mathcal{S}$ is a continuous injective function from the graph (as a topological space) to $\mathcal{S}$. Vertices of G are mapped to distinct points in $\mathcal{S}$; edges are mapped to simple, interior-disjoint paths. A face of an embedding is a component of the complement of the image of the embedding. Without loss of generality, we consider only cellular embeddings, meaning every face is homeomorphic to an open disk. To avoid excessive notation, we do not distinguish between vertices, edges, and subgraphs of G and their images under the embedding. A *bigon* is a face bounded by exactly two edges.

A cycle is *contractible* if it can be continuously deformed to a point; classical results of Epstein [3] imply that a simple cycle in $\mathcal{S}$ is contractible if and only if it is the boundary of a disk in $\mathcal{S}$. Two paths with matching endpoints are *homotopic* if they form a contractible cycle. Thus, two edges in an embedded graph are homotopic if and only if they bound a bigon.

The *dual* G^* of an embedded graph G is defined as follows. For each face f of G, the dual graph has a corresponding vertex f^*. For each edge e of G, the dual graph has an edge e^* connecting the vertices dual to the two faces on either side of e. Intuitively, one can think of f^* as an arbitrary point in the interior of f and e^* has a path that crosses e at its midpoint and does not intersect any other edge of G. Each face of G^* corresponds to a vertex of G; thus, the dual of G^* is isomorphic to the original embedded graph G. Trivially, a face f of G is a bigon if and only if its dual vertex f^* has degree 2.

The *genus* of a surface $\mathcal{S}$ is the maximum number of cycles in $\mathcal{S}$ whose complement is connected. Let G be an embedded graph with n vertices, m edges, and f faces. *Euler's formula* states that $n - m + f = 2 - \chi(\mathcal{S})$, where $\chi(\mathcal{S})$ is a topological invariant of the surface called its *Euler characteristic*. For a surface of genus g, the Euler characteristic is $2 - 2g$ if the surface is orientable and $2 - g$ if the surface is non-orientable. To simplify our notation, we define the *Euler genus* of $\mathcal{S}$ as $\mathsf{eg}(\mathcal{S}) := 2 - \chi(\mathcal{S})$.

1.3 Organization

The rest of the paper is organized as follows. In Section 2 we recall the Held-Karp LP relaxation of ATSP. In Section 3 we introduce the notion of a ribbon, and prove some basic properties of decompositions of the edges of a graph into a collection of ribbons. Using these ribbon decompositions, we show in Section 4 how to construct a thin forest, with a small number of connected components. In Section 5 we how how to turn a thin forest with a small number connected components into a small collection of closed walks visiting all the vertices, and with small total cost. Finally, in Section 6 we show how to combine the above technical ingredients to obtain the algorithm for ATSP.

2 The Held-Karp LP relaxation

We now recall the definition of Held and Karp's linear programming relaxation of ATSP [5]. Fix a directed graph $\vec{G} = (V, A)$ and a cost function $c\colon A \to \mathbb{R}^+$. For any subset $U \subseteq V$, we define

$$\begin{aligned} \delta^+_{\vec{G}}(U) &= \{u{\to}v \in A \mid u \in U \text{ and } v \notin U\}, \\ \delta^-_{\vec{G}}(U) &= \delta^+_{\vec{G}}(V \setminus U). \end{aligned}$$

We omit the subscript $\vec{G}$ when the underlying graph is clear from context. To simplify notation, we write $\delta^+(v) = \delta^+(\{v\})$ and $\delta^-(v) = \delta^-(\{v\})$ for any single vertex v.

Let $G = (V, E)$ be the undirected graph such that $uv \in E$ if and only if $u{\to}v \in A$ or $v{\to}u \in A$. For any $U \subseteq V$, we define

$$\delta_G(U) = \{uv \in E \mid u \in U \text{ and } v \notin U\}.$$

Again, we omit the subscript G when the underlying graph is clear from context. We also extend the cost function c to undirected edges by defining

$$c(uv) = \min\{c(u{\to}v), c(v{\to}u)\}.$$

For any function $x\colon A \to \mathbb{R}$ and any subset $W \subseteq A$ of arcs, we write $x(W) = \sum_{a \in W} x(a)$. With this notation, the Held-Karp LP relaxation is defined as follows.

$$\begin{array}{rrl} \text{minimize} & \displaystyle\sum_{a \in A} c(a) \cdot x(a) & \\ \text{subject to} & x(\delta^+(U)) \geq 1 & \text{for all } U \subseteq V \\ & x(\delta^+(v)) = 1 & \text{for all } v \in V \\ & x(\delta^-(v)) = 1 & \text{for all } v \in V \\ & x(a) \geq 0 & \text{for all } a \in A \end{array}$$

Let $x\colon A \to \mathbb{R}$ be a feasible solution for the Held-Karp LP. We define the *symmetrization* of x to be a function $z\colon E \to \mathbb{R}$, such that for any $uv \in E$, we have

$$z(uv) = x(u{\to}v) + x(v{\to}u).$$

3 Ribbon decompositions

In this section we introduce some of the machinery that we will use in our algorithm for computing a thin forest.

Let G be a graph embedded into a surface. A *ribbon* in G is a maximal set R of parallel non-self-loop edges in G, such that for any $e \neq e' \in R$, the cycle $e \cup e'$ is contractible (see Figure 1 for an example). Note that for every ribbon R, if $|R| > 1$, then there exists a disk in the surface that obtains R. The edges in R are naturally ordered inside this disk, so that every two consecutive edges in R bound a face. We say that an edge in R is *central* if it is a weighted (w.r.t. z) median in this ordering. Note that every ribbon has either one, or two central edges.

Lemma 3.1. *Let G be a graph embedded into some surface. Then, the set of non-self-loop edges of G can be uniquely decomposed into a collection of pairwise disjoint ribbons.*

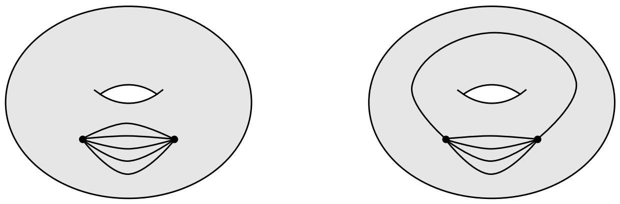


Figure 1. Left: A set of parallel edges forming a ribbon. Right: A set of parallel edges *not* forming a ribbon.

Proof: It follows immediately from the definition of a ribbon. $\square$

Lemma 3.2. *Let $G = (V, E)$ be a multi-graph embedded into a surface $\mathcal{S}$. Let $\mathcal{R}$ be a decomposition of the non-self-loop edges of G into ribbons. Then, $|\mathcal{R}| \leq 3|V| - 3\chi(\mathcal{S})$.*

Proof: Let $Y \subseteq E$ be a set obtained by choosing an arbitrary edge from every ribbon $R \in \mathcal{R}$. Let $G' = (V, Y)$ be the subgraph of G induced by Y. Since for every pair of edges $e, e' \in Y$ with common endpoints, we have that the loop $\varphi(e) \cup \varphi(e')$ is noncontractible, and there are no self-loops in Y, it follows that every face of G' contains at least 3 distinct edges. Let F denote the set of faces of G'. Because no face of G' is a bigon, we have $3|F| \leq 2|Y|$. Euler's formula $|V| - |Y| + |F| = \chi(\mathcal{S})$ now immediately implies that $|V| - |Y|/3 \geq \chi(\mathcal{S})$. We conclude that $|\mathcal{R}| = |Y| \leq 3|V| - 3\chi(\mathcal{S})$, as required. $\square$

4 Computing a thin forest

As before, let $(\vec{G}, c)$ be an instance of ATSP, with $\mathsf{eg}(\vec{G}) = g$, let x be a feasible solution to the Held-Karp relaxation of $(\vec{G}, c)$, and let z be the symmetrization of x. The notion of a *thin* set captures a key idea for rounding a solution of the Held-Karp LP [1, 8]. We begin by recalling the definition of a thin set.

Definition 4.1 (Thinness). *Let $W \subseteq E$ and $\alpha, s > 0$. We say that W is (α, s)-*thin *(w.r.t. x) if for any $U \subseteq V$,*

$$|W \cap \delta(U)| \leq \alpha \cdot z(\delta(U)),$$

and

$$c(W) \leq s \cdot \sum_{a \in A} c(a) \cdot x(a).$$

In this section we show how to compute a $(O(1), O(1))$-thin forest, with at most g connected components.

We begin by first showing how to compute a forest T with at most g components, satisfying a slightly weaker notion of thinness. To that end, we compute a sequence of graphs $G_0, \ldots, G_t$, with $G_0 = G$, as follows. Let $i \geq 0$, and suppose we have computed G_i. If G_i has at most g vertices, then we terminate the sequence at G_i, and we set $t = i$. Otherwise, let $\mathcal{R}_i$ be the decomposition of the non-self-loop edges in G_i into ribbons (by Lemma 3.1 the decomposition $\mathcal{R}_i$ is unique). We set

$$R_i = \arg\max_{R' \in \mathcal{R}_i} z(R').$$

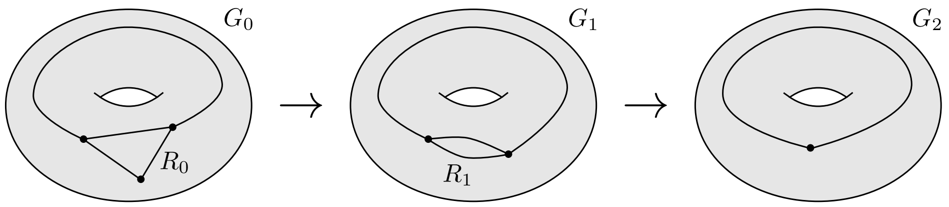


Figure 2. An example of a sequence of ribbon contractions preformed by our algorithm.

We let G_{i+1} be the graph obtained from G_i by contracting all the edges in R_i. This concludes the definition of the sequence of graphs $G_0, \ldots, G_t$ (see Figure 2 for an example). For every $i \in \{0, \ldots, t-1\}$, let e_i be a central edge in R_i. We define T to be the graph induced by $\{e_0, \ldots, e_{t-1}\}$. Since ribbons do not contain self-loop edges, and we only contract ribbons, it immediately follows that T is a forest. It is also immediate that T contains one connected component for every vertex of G_t, and therefore it contains at most g connected components.

It remains to show that T satisfies a weak thinness condition. We first derive the following auxiliary fact.

Lemma 4.2. *For any $i \in \{0, \ldots, t-1\}$, we have $z(R_i) \geq 2/5$.*

Proof: Let $i \in \{0, \ldots, t-1\}$, $G_i = (V_i, E_i)$, and $U \subseteq V_i$. Since G_i is obtained from G via a sequence of edge contractions, it follows that there exists $U' \subseteq V$, such that $\delta_{G_i}(U) = \delta_G(U')$. Thus,

$$\begin{aligned} z(\delta_{G_i}(U)) &= z(\delta_G(U')) \\ &= x(\delta^+_{\vec{G}}(U')) + x(\delta^-_{\vec{G}}(U')) \\ &\geq 2. \end{aligned} \tag{1}$$

By construction we have that for any $i \in \{0, \ldots, t-1\}$, $|V_i| \geq g$. By lemma 3.2 we have that for any $i \in \{0, \ldots, t-1\}$,

$$\begin{aligned} |\mathcal{R}_i| &\leq 3|V_i| - 3\chi(\mathcal{S}) \\ &= 3|V_i| - 6 + 3g \\ &< 6|V_i| - 6. \end{aligned}$$

Therefore, there exists $v_i \in V_i$, such that all the non-self-loop edges incident to v_i are contained in at most 5 ribbons. By (1), we have that

$$\sum_{e \in \delta(\{v_i\})} z(e) \geq 2.$$

This implies that there exists a ribbon $R'_i \in \mathcal{R}_i$, such that $z(R'_i) \geq 2/5$. Therefore,

$$z(R_i) \geq z(R'_i) \geq 2/5,$$

which concludes the proof. □

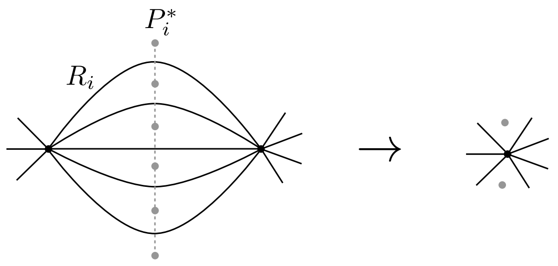
We can now show that T satisfies a weak notion of thinness.

Lemma 4.3. *For any $U \subseteq V$, we have $|T \cap \delta(U)| \leq O(1) \cdot z(\delta(U))$.*

Proof: For any $i \in \{0, \ldots, t\}$, let $G_i^* = (V_i^*, E_i^*)$ be the dual of G_i. For any $i \in \{0, \ldots, t-1\}$, let $R_i^* \subseteq E_i^*$ be the set of the duals of all edges in R_i, and let e_i^* be the dual of e_i.

Note that for any $i \in \{0, \ldots, t-1\}$, all self-loop edges in G_i are noncontractible. This is because self-loops can only be created when contracting the edges in a ribbon R. For any such self-loop e, since $e \notin R$, it follows that there exists $e' \in R$, such that the loop $e \cup e'$ is noncontractible, which implies that e becomes a noncontractible loop after contracting all edges in R. This implies that for any $i \in \{0, \ldots, t-1\}$, there exists a path P_i^* in G_i^*, such that $E(P_i^*) = R_i^*$ and all internal vertices in P_i^* have degree 2.

For any $i \in \{0, \ldots, t-1\}$, G_{i+1} is obtained from G_i by contracting all edges in R_i. Since the edges in R_i are non-self-loops, this means that G_{i+1}^* is obtained from G_i^* by deleting all edges in R_i^* and all internal vertices in P_i^*.



Consider some set $U \subseteq V$. Let $X = \delta(U)$, and let X^* be the corresponding set of dual edges $\{e^* \in E^* \mid e \in X\}$. There exists a collection $\mathcal{K}^*$ of pairwise edge-disjoint cycles in G^*, such that

$$X^* = \bigcup_{K^* \in \mathcal{K}^*} E(K^*).$$

Let $K^* \in \mathcal{K}^*$. Suppose first that K, contains exactly one edge of T, and let e_i be that edge. It follows by construction that $P_i^* \subseteq K^*$, which implies $z(K) \geq z(R_i) \geq 2/5$. Otherwise, suppose that K, contains at least two edges of T. Let e_i^*, e_j^* be two such edges that are consecutive in K^*. Assume w.l.o.g. that $i < j$. By construction we have that the subpath Q^* of K^* between (and including) e_i^*, and e_j^* satisfies $z(Q) \geq z(R_i)/2 \geq 1/10$. To summarize, we have that for any $K^* \in \mathcal{K}^*$

$$z(K) \geq \frac{1}{20}|T \cap K|.$$

Summing up over all cycles in $\mathcal{K}^*$, we obtain

$$\begin{aligned} |T \cap \delta(U)| &= \sum_{K^* \in \mathcal{K}^*} |T \cap K| \\ &\leq \sum_{K^* \in \mathcal{K}^*} 20 \cdot z(K) \\ &= 20 \cdot z(\delta(U)), \end{aligned}$$

concluding the proof. $\square$

We are now ready to prove the main result of this section. Lemma 4.3 gives a spanning forest that satisfies a weak notion of thinness. More specifically, T might have very large cost (w.r.t. c). We show how to transform this forest into a $(O(1), O(1))$-thin spanning forest. Our proof uses the argument from Oveis Gharan and Saberi [8], who obtained this transformation for the case of spanning trees. We observe that their proof works in the more general setting of spanning forests with a fixed number of connected components. We give the proof for completeness.

Lemma 4.4 (Computing a thin forest). *There exists a polynomial time algorithm which computes a $(O(1), O(1))$-thin (w.r.t. x) spanning forest T in G, with at most g connected components.*

Proof: We first remark that the algorithm of Lemma 4.3 returns a forest T satisfying for all $U \subseteq V$, $|T \cap \delta(U)| \leq \alpha z(\delta(U))$, for some $\alpha = O(1)$, even if z is not the symmetrization of some feasible solution to the Held-Karp LP. The only requirements for the algorithm of Lemma 4.3 are that z is non-negative, and for all $U' \subseteq V$, we have $z(\delta_G(U')) \geq 2$.

Let $N = n^2/\alpha$. We compute a sequence of non-negative functions $z_0, \ldots, z_N$, and a sequence of spanning forests $T_1, \ldots, T_N$, each having at most g connected components, as follows. We set $z_0 = 3\lfloor zn^2 \rfloor / n^2$, and inductively maintain the invariant that for any $i \in \{1, \ldots, N\}$:

$$\text{for all } U' \subseteq V, \quad z(\delta_G(U')) \geq 2 \tag{2}$$

It is immediate that z_0 is non-negative, and satisfies (2). Given z_i, for some $i \in \{0, \ldots, N-1\}$, we compute z_{i+1}, T_{i+1} as follows. Using Lemma 4.3 we find a spanning forest T_i with at most g connected components, such that

$$\text{for all } U \subseteq V, \quad |T_{i+1} \cap \delta_G(U)| \leq \alpha z_i(\delta_G(U)) \tag{3}$$

For any $e \in E$, we set

$$z_{i+1}(e) = \begin{cases} z_i(e) - 1/n^2 & \text{if } e \in T_{i+1} \\ z_i(e) & \text{if } e \notin T_{i+1} \end{cases}$$

We need to verify that (2) holds for z_{i+1}. We have that for any $U' \subseteq V$,

$$\begin{aligned} z_{i+1}(d_G(U')) &\geq z_i(\delta_G(U')) - \alpha/n^2 && (4)\\ &\geq z_0(\delta_G(U')) - N\alpha/n^2 \\ &> 3(z(\delta_G(U')) - 1) - N\alpha/n^2 && (5)\\ &\geq 2, && (6) \end{aligned}$$

where (4) follows by (3), (5) by the fact that $z_0 = 3\lfloor zn^2 \rfloor / n^2$, and every cut contains less than n^2 edges, and (6) by the fact that z is the symmetrization of a feasible solution to the Held-Karp LP. Therefore, the inductive invariant (2) is maintained. Lastly, we need to verify that z_{i+1} is non-negative. Note that z_0 takes values that are multiples of $1/n^2$, and we only decrease the values of z_i by $1/n^2$, which implies that for all i, all values in z_i are multiples of $1/n^2$. An edge e can appear in the forest T_{i+1} only if $z_i(e) > 0$. Therefore, when setting $z_{i+1}(e) = z_i(e)$, the value $z_{i+1}(e)$ remains non-negative.

We now set the desired forest T to be

$$T = \operatorname*{arg\,min}_{T' \in \{T_1, \ldots, T_N\}} c(T').$$

It remains to verify that T is $(O(1), O(1))$-thin. Suppose that $T = T_i$, for some $i \in \{1, \ldots, N\}$. By (3), we have that for all $U \subseteq V$,

$$\begin{aligned} |T_i \cap \delta_G(U)| &\leq \alpha z_{i-1}(\delta_G(U)) \\ &\leq \alpha z_0(\delta_G(U)) \\ &\leq 3\alpha z(\delta_G(U)) \end{aligned} \tag{7}$$

For every $uv \in E$, we have $\sum_{uv \in T_i} 1/n^2 \leq z_0(uv) \leq 3z(uv)$, and therefore $c(uv) \sum_{uv \in T_i} 1/n^2 \leq 3z(uv)c(uv) \leq 3x(u \to v)c(u \to v) + x(v \to u)c(v \to u)$. Summing over all edges $uv \in E$, we obtain

$$\begin{aligned} c(T) &\leq \frac{1}{N} \sum_{i=1}^{N} c(T_i) \\ &\leq 3\frac{1}{N} n^2 \sum_{a \in A} x(a)c(a) \\ &= 3\alpha \sum_{a \in A} x(a)c(a) \end{aligned} \tag{8}$$

Combining (7) & (8) we obtain that T is $(3\alpha, 3\alpha)$-thin, as required. □

5 From thin forests to walks

In this section we present the last missing ingredient of our algorithm. More specifically, we show how given a thin forest with a few connected components, we can construct a small number of closed walks that span the input graph and have small total cost. This step is similar to the argument in [1]. However, here we need a slightly modified version since we are dealing with a thin forest, instead of a thin tree, so we include the proof for completeness. We will use the following result due to Hoffman [6]. (See also Schrijver [9, Theorem 11.2].)

Theorem 5.1 (Hoffman's circulation theorem [6]). *Let $\vec{G} = (V, A)$ be a directed graph, and let $l, u \colon A \to \mathbb{R}$. Then, there exists a circulation f with $l(a) \leq f(a) \leq u(a)$ for all $a \in A$, if and only if the following two conditions are satisfied:*

(1) $l(a) \leq u(a)$ for all $a \in A$.

(2) For all $U \subseteq V$, we have $l(\delta^-(U)) \leq u(\delta^+(U))$.

Furthermore, if l and u are integer-valued, then f can be chosen to be integer-valued.

We are now ready to prove the main result of this section.

Lemma 5.2 (From thin forests to Eulerian walks). *Let $\alpha, s > 0$, and let T be a (α, s)-thin (w.r.t. x) spanning forest in G with at most k connected components. Then, there exists a polynomial-time algorithm which computes a collection of closed closed walks $C_1, \ldots, C_{k'}$ in $\vec{G}$, for some $k' \leq k$, that visit all the vertices in $\vec{G}$, and such that $\sum_{i=1}^{k} c(C_i) \leq (2\alpha + s) \sum_{a \in A} c(a) \cdot x(a)$.*

Proof: Define a set $\vec{T} \subset A$ as follows. For every $\{u, v\} \in T$, if $c((u, v)) \leq c((v, u))$, then we add (u, v) to $\vec{T}$, and otherwise we add (v, u) to $\vec{T}$.

Define functions $l, u \colon A \to \mathbb{R}^+$, such that for any $a \in A$,

$$l(a) = \begin{cases} 1 & a \in \vec{T} \\ 0 & a \notin \vec{T} \end{cases},$$

and

$$u(a) = \begin{cases} 1 + 2\alpha x(a) & a \in \vec{T} \\ 2\alpha x(a) & a \notin \vec{T} \end{cases}.$$

We argue that there exists a circulation $f \colon A \to \mathbb{R}^+$, such that for any $a \in A$, $l(a) \leq f(a) \leq u(a)$. To that end, it suffices to verify conditions (1), and (2) of theorem 5.1. For any $a \in A$ we have $l(a) = u(a) - 2\alpha x(a) \leq u(a)$, and therefore condition (1) is satisfied. To verify condition (2), consider a set $U \subseteq V$. Let z be the symmetrization of x. We have

$$\begin{aligned} l(\delta^-(U)) &= |\vec{T} \cap \delta^-(U)| & (9) \\ &\leq |T \cap \delta(U)| & (10) \\ &\leq \alpha z(\delta(U)) & (11) \\ &= \alpha(x(\delta^+(U)) + x(\delta^-(U))) & (12) \\ &= 2\alpha x(\delta^+(U)) & (13) \\ &\leq 2\alpha u(\delta^+(U)), & (14) \end{aligned}$$

where (9) follows by the definition of l, (10) by the definition of $\vec{T}$, (11) by the thinness of T, (12) by the definition of z, (13) by flow conservation, and (14) by the definition of u. This shows that condition (2) of theorem 5.1 is satisfied, and therefore there exists a circulation f as required. Moreover, since l, and u are integer-valued, f can be chosen to be integer-valued.

We define the directed multigraph $\vec{H}$ containing an arc (v, w) for every unit of flow on (v, w) in f. By flow conservation, $\vec{H}$ is Eulerian. Moreover, $\vec{H}$ contains $\vec{T}$ as a subgraph. Since $\vec{T}$ has k weakly-connected components, it follows that $\vec{H}$ contains $k' \leq k$ strongly-connected components. Therefore, $\vec{H}$ can be decomposed into k' closed walks $C_1, \ldots, C_{k'}$ spanning $\vec{G}$, as required.

It remains to bound the total cost of these walks. We have

$$\begin{aligned} \sum_{i=1}^{k'} c(C_i) &= \sum_{a \in A} c(a) \cdot f(a) \\ &\leq \sum_{a \in A} c(a) \cdot u(a) \\ &= c(T) + 2\alpha \sum_{a \in A} c(a) \cdot x(a) \\ &= (2\alpha + s) \sum_{a \in A} c(a) \cdot x(a), \end{aligned}$$

concluding the proof. □

6 The algorithm

We are now ready to prove our main result. The first ingredient in our algorithm is a procedure which computes a $(O(1), O(1))$-thin forest T, having at most g connected components. This is done in Lemma 4.4. The second ingredient is a procedure which given this forest T, outputs a collection of at most g walks that visit all vertices, and with total cost at most a constant factor times the cost of the fractional solution x, i.e. at most $O(1) \cdot \sum_{a \in A} c(a) \cdot x(a)$. This is done in Lemma 5.2. Using these Lemmas as subroutines, we can obtain an approximation algorithm for ATSP.

Proof (Proof of Theorem 1.1): Let $\mathsf{OPT} \geq 0$ be the minimum cost of a solution to $(\vec{G}, c)$. We first compute an optimal solution x to the Held-Karp LP relaxation of $(\vec{G}, c)$. Using lemma 4.4, we compute in polynomial time a $(O(1), O(1))$-thin (w.r.t. x) spanning forest T in G, such that T has at most g connected components. Using lemma 5.2, we compute in polynomial time a collection of k closed walks $C_1, \ldots, C_k$, with $k \leq g$, such that every vertex in G is visited by some walk, and the total cost of all walks is at most $O(\mathsf{OPT})$. For every $i \in \{1, \ldots, k\}$, pick a vertex v_i visited by C_i. We create a new instance $(\vec{G}', c')$ of ATSP induced on $\{v_1, \ldots, v_k\}$. More precisely, we define the directed graph $\vec{G}' = (V', A')$, with $V' = \{v_1, \ldots, v_k\}$, and for any $u, v \in V'$, we have an arc $u \to v$ in A', with $c'(u \to v) = \min_P c(P)$, where P ranges over all directed paths form u to v in $\vec{G}$. Clearly, the instance $(\vec{G}', c')$ has a solution of cost $\mathsf{OPT}' \leq \mathsf{OPT}$. Using the $f(n)$-approximation algorithm for n-vertex graphs, we find a closed tour C in $\vec{G}'$, of cost at most $\mathsf{OPT} \cdot f(k)$. By composing C with the closed walks $C_1, \ldots, C_k$, and shortcutting appropriately (as in [4]), we obtain a solution for $(\vec{G}, c)$, of total cost $O(\mathsf{OPT}) + \mathsf{OPT} \cdot f(k) = O(\mathsf{OPT} \cdot f(k))$, as required. $\square$

References

[1] Arash Asadpour, Michel X. Goemans, Aleksander Madry, Shayan Oveis Gharan, and Amin Saberi. An $O(\log n / \log\log n)$-approximation algorithm for the asymmetric traveling salesman problem. *Proc. 21st Ann. ACM-SIAM Symp. Discrete Algorithms*, 379–389, 2010.

[2] Chandra Chekuri and Anastasios Sidiropoulos. Approximation algorithms for Euler genus, and related problems. Manuscript, 2013.

[3] David B. A. Epstein. Curves on 2-manifolds and isotopies. *Acta Mathematica* 115:83–107, 1966.

[4] Alan M. Frieze, Giulia Galbiati, and Francesco Maffioli. On the worst-case performance of some algorithms for the asymmetric traveling salesman problem. *Networks* 12(1):23–39, 1982.

[5] Martin Held and Richard Karp. The traveling salesman problem and minimum spanning trees. *Operations Research* 18:1138–1162, 1970.

[6] A. J. Hoffman. Some recent applications of the theory of linear inequalities to extremal combinatorial analysis. *Combinatorial Analysis*, 113–127, 1960. Symp. Appl. Math. 10.

[7] Bojan Mohar and Carsten Thomassen. *Graphs on surfaces.* Johns Hopkins Univ. Press, 2001.

[8] Shayan Oveis Gharan and Amin Saberi. The asymmetric traveling salesman problem on graphs with bounded genus. *Proc. 22nd Ann. ACM-SIAM Symp. Discrete Algorithms*, 967–975, 2011.

[9] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency.* Algorithms and Combinatorics 24. Springer, 2003.