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

Jeff Erickson†
Department of Computer Science
University of Illinois, Urbana-Champaign

Anastasios Sidiropoulos‡
Department of Computer Science & Engineering
and Department of Mathematics
The Ohio State University

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. Given any algorithm that achieves an approximation ratio of $f(n)$ on arbitrary n-vertex graphs as a black box, our algorithm achieves an approximation factor of $O(f(g))$ on graphs with genus g. 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. Moreover, recent results on approximating the genus of graphs imply that our $O(\log g/\log\log g)$-approximation algorithm can be applied to bounded-degree graphs even if no genus-g embedding of the graph is given. Our result improves and generalizes the $O(\sqrt{g}\log g)$-approximation algorithm of Oveis Gharan and Saberi [SODA 2011], which applies only to graphs with *orientable* genus g and requires a genus-g embedding as part of the input, even for bounded-degree graphs. Finally, our techniques yield a $O(1)$-approximation algorithm for ATSP on graphs of genus g with running time $2^{O(g)} \cdot n^{O(1)}$.

Categories and Subject Descriptors: F.2.2 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems—*Computations on discrete structures*; G.2.2 [Discrete Mathematics]: Graph Theory—*Graph algorithms, Path and circuit problems*

Keywords: approximation algorithms, topological graph algorithms, traveling salesman

1. INTRODUCTION

The Asymmetric Traveling Salesman Problem (ATSP) is one of the most fundamental and well studied problems in combinatorial optimization. An instance of ATSP consists of a a directed graph $\vec{G} = (V, A)$ and a (not necessarily symmetric) cost function $c : A \to \mathbb{R}^+$; we can assume without loss of generality that the cost of any arc is equal to the shortest-path distance between its endpoints. The goal is to find a spanning closed walk of $\vec{G}$ with minimum total cost.

The exact problem is well-known to be NP-hard. In the early 1980s, Frieze *et al.* [7] described a polynomial-time algorithm that achieved an approximation ratio of $\log_2 n$; subsequent papers improved this approximation ratio by constant factors [3,6,11]. A recent breakthrough result of Asadpour *et al.* [1] improved the approximation ratio to $O(\log n/\log\log n)$, marking the first asymptotic improvement in almost 30 years. Building on this breakthrough, Oveis Gharan and Saberi [14] described a polynomial-time $O(\sqrt{g}\log g)$-approximation algorithm when the input includes an embedding of the input graph into an orientable surface of genus g. We refer the interested reader to the two previous papers [1, 14] and the references therein for a more detailed overview of the rich history of the problem.

In this paper, we present a polynomial-time approximation algorithm for the asymmetric traveling salesman problem in graphs embedded on surfaces of bounded genus, whose approximation ratio is optimal up to constant factors.

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

Combining our main result with the approximation algorithm of Asadpour *et al.* [1] immediately implies the following corollary.

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

Our algorithm has several advantages over the result of Oveis Gharan and Saberi [14], aside from improving the approximation ratio from $O(\sqrt{g}\log g)$ to $O(\log g/\log\log g)$. First, ours is the first constant-factor approximation algorithm for graphs of bounded *non-orientable* genus; Oveis Gharan and Saberi's algorithm requires an embedding onto an *orientable* surface .

Second, recent results of Chekuri and Sidiropoulos [4] imply that for bounded-degree graphs, our algorithm does not require an embedding to be given as part of the input. Given a graph G with bounded maximum degree, Chekuri and Sidiropoulos [4] describe an algorithm that outputs an embedding of G into a

*See http://www.cs.uiuc.edu/~jeffe/pubs/atsp.htmlfor the most recent version of this paper.

†Supported in part by NSF grant CCF-0915519.

‡Work on this paper was done while this author was at the University of Illinois, supported in part by a David and Lucille Packard Fellowship and by NSF grants CCF-0915984 and CCF-0915519.

SoCG'14, June 8–11, 2014, Kyoto, Japan.

surface with orientable (resp. non-orientable) genus $g^{O(1)}$, where g is the orientable (resp. non-orientable) genus of G, in polynomial time (independent of g). Combining their result with Theorem 1.1 immediately implies us the following corollary.

Corollary 1.3. *There is a polynomial-time $O(\log g/\log\log g)$-approximation algorithm for ATSP for bounded-degree graphs with either orientable or non-orientable genus g; this algorithm does not require an embedding of the graph as part of the input.*

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

Finally, a minor modification of our algorithm improves the approximation guarantee to a *constant*, in time exponential in the genus of the input graph. In other words, obtaining a constant-factor approximation for ATSP is fixed-parameter tractable, when parametrized by the genus of the input graph. In particular, we obtain the first polynomial-time constant-factor approximation for graphs of genus $O(\log n)$.

Theorem 1.4. *There is an $O(1)$-approximation algorithm for ATSP for graphs with either orientable or non-orientable genus g that runs in time $2^{O(g)}n^{O(1)}$; this algorithm does not require an embedding of the graph as part of the input.*

Corollary 1.5. *There is a polynomial-time $O(1)$-approximation algorithm for ATSP for graphs of either orientable or non-orientable genus $O(\log n)$, which does not require an embedding of the graph as part of the input.*

1.1 Algorithm Overview

Previous work on rounding the Held-Karp LP. Our algorithm is inspired by and relies heavily on recent results of Asadpour *et al.* for general graphs [1] and Oveis Gharan and Saberi for surface-embedded graphs [14]. Both of these algorithms use the classical linear programming relaxation of ATSP introduced by Held and Karp [9]. 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 weight 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 [14] 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 $W_1, W_2, \ldots, W_g$ that collectively visit all vertices in the graph and that have total cost $O(\mathsf{OPT})$. 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. Our algorithm maintains a decomposition of the input graph into *ribbons*, where each ribbon is 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* a 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 subgraph 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 our notation and terminology. For a more detailed background in topological graph theory, we refer the reader to Mohar and Thomassen [13].

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 (possibly with parallel edges and self-loops) into 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 $\mathcal{S}$, and 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 embeddings for which 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 [5] 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 $n - m + f = \chi(\mathcal{S})$, 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 $\mathcal{S}$ to be $2 - \chi(\mathcal{S})$. The orientable genus of a graph G is the minimum genus of an orientable surface that supports an embedding of G; the non-orientable genus and the Euler genus of G are defined similarly.

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 few 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

We now recall the definition of Held and Karp's linear programming relaxation of ATSP [9]. Fix a directed graph $\vec{G} = (V, A)$ and a cost function $c : A \to \mathbb{R}^+$. For any subset $U \subseteq V$, we define

$$\delta^+_{\vec{G}}(U) := \{u{\to}v \in A \mid u \in U \text{ and } v \notin U\}$$
$$\text{and} \quad \delta^-_{\vec{G}}(U) := \delta^+_{\vec{G}}(V \setminus U).$$

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 $x : A \to \mathbb{R}$ 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.

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

The Held-Karp LP is traditionally defined with variables for each arc in a *complete* directed graph, with the additional constraint $x(\delta^+(v)) = 1$ for every vertex $v \in V$. However, the weaker formulation given above is both more convenient for our algorithm, since any feasible solution is supported on the arcs of the input graph $\vec{G}$, and sufficient for our analysis.

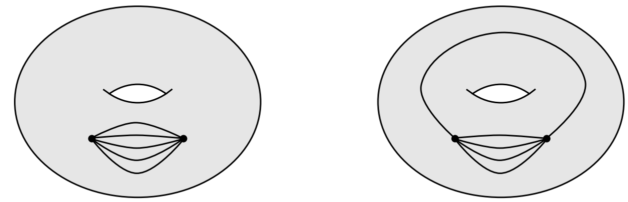

Figure 1. Left: A ribbon. Right: Parallel edges *not* forming a ribbon.

Let $x : A \to \mathbb{R}$ be a feasible solution for the Held-Karp LP. We define the *symmetrization* of x as the function $z : E \to \mathbb{R}$ where

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

for every edge $uv \in E$. For any subset $W \subseteq E$ of edges, we write $z(W) := \sum_{e \in W} z(e)$.

3. RIBBON DECOMPOSITIONS

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.) By definition, every non-loop edge in G belongs to exactly one ribbon. We call the set of all ribbons in any embedded graph G the *ribbon decomposition* of G. See Figure 1.

Lemma 3.1. *Let $G = (V, E)$ be a graph embedded into a surface $\mathcal{S}$, and let $\mathcal{R}$ be its ribbon decomposition. We have* $|\mathcal{R}| \leq 3|V| - 3\chi(\mathcal{S})$.

Proof: Let H be a subgraph of G containing exactly one edge from each ribbon in $\mathcal{R}$. Every pair of parallel edges in H defines a non-contractible cycle, and there are no self-loops in H. Thus, every face of H has at least 3 distinct edges, which implies that H has at most $2|\mathcal{R}|/3$ faces. The inequality $|\mathcal{R}| \leq 3|V| - 3\chi(\mathcal{S})$ now follows immediately from Euler's formula. □

Every ribbon R is contained in a closed disk in the surface that does not intersect the interior of any edge outside R. The edges in R are naturally ordered inside this disk, so that every consecutive pair of edges bounds a bigon. We say that an edge in R is *central* if it is a weighted median (with respect to z) in this ordering. Every ribbon has either one or two central edges.

4. COMPUTING A THIN FOREST

For the rest of the paper, fix a directed graph $\vec{G} = (V, A)$, embedded on a surface with Euler genus g, and a positive cost function c on the arcs of G. Let x be any feasible solution to the Held-Karp linear program for $\vec{G}$ and c, and let z be the symmetrization of x. For any subset $W \subseteq E$, we refer to $c(W)$ as its *cost* and $z(W)$ as its *weight*.

The notion of a *thin* set, introduced by Asadpour *et al.* [1], captures a key idea for rounding a solution of the Held-Karp linear program. Fix two positive real parameters α and s. A subset $W \subseteq E$ is said to be (α, s)*-thin* (with respect to x) if

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

for every subset $U \subseteq V$, and

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

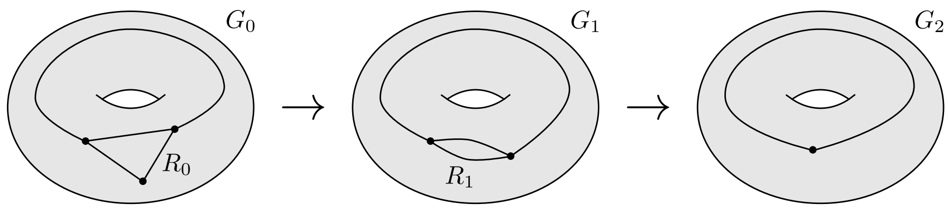

Figure 2. A sequence of ribbon contractions.

That is, the number of edges in W that cross any cut is at most α times the total weight of all edges crossing that cut, and the total cost of W is at most s times the Held-Karp objective value.

In this section we show how to compute a $(O(1), O(1))$-thin forest with at most g connected components. We begin by describing how to compute a forest T with at most g components that satisfies a slightly weaker notion of thinness.

To that end, we compute a sequence of graphs $G_0, G_1, \ldots, G_t$, with $G_0 = G$, as follows. Fix an index $i \geq 0$, and suppose we have already computed G_i. If G_i has at most g vertices, we set $t = i$ and terminate the sequence. Otherwise, let R_i denote a maximum-weight ribbon in G_i; more formally,

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

where $\mathcal{R}_i$ be the ribbon decomposition of G_i. Let G_{i+1} be the graph obtained from G_i by contracting all the edges in R_i. See Figure 2 for an example. Finally, let T be a subgraph containing one central edge e_i from each ribbon R_i. Because ribbons do not contain self-loops and we only contract ribbons, T must be a forest with exactly one component for each vertex of G_t, and therefore at most g components in total.

Lemma 4.1. *For each index i between 0 and $t-1$, we have $z(R_i) \geq 2/5$.*

Proof: Fix an index i between 0 and $t-1$, and consider the graph $G_i = (V_i, E_i)$. By construction we have $|V_i| \geq g$, so Lemma 3.1 implies that

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

Therefore, G_i has at least one vertex v_i that is incident to at most 5 ribbons.

For any vertex subset $U_i \subseteq V_i$, there is a corresponding subset $U \subseteq V$ such that $\delta_{G_i}(U_i) = \delta_G(U)$, because G_i is obtained from G via a sequence of edge contractions; it follows that

$$z(\delta_{G_i}(U_i)) \;=\; z(\delta_G(U')) \;=\; x(\delta^+_{\vec{G}}(U)) + x(\delta^-_{\vec{G}}(U)) \;\geq\; 2.$$

for every subset $U_i \subseteq V_i$. In particular, we have $z(\delta(v_i)) \geq 2$. Thus, the maximum-weight ribbon incident to v_i has weight at least $2/5$. The lemma follows immediately. □

We can now show that T satisfies the first condition in the definition of thinness.

Lemma 4.2. *For any subset $U \subseteq V$, we have $|T \cap \delta(U)| \leq O(1) \cdot z(\delta(U))$.*

Proof: For any index i between 0 and t, let $G_i^* = (V_i^*, E_i^*)$ denote the dual of G_i. Similarly, for any index i between 0 and $t-1$, let $R_i^* \subseteq E_i^*$ denote the set of the duals of all edges in the maximum-weight ribbon R_i, and let e_i^* denote the dual of the central edge e_i.

Recall that the edges of R_i are linearly ordered so that any successive pair bounds a bigon; with the same linear ordering, any two successive edges in R_i^* share a vertex of degree 2. Thus, the edges in R_i^* define either a path in G_i^* whose endpoints have degree at least three, or a cycle in G_i^* with a single vertex of degree at least three. Each graph G_{i+1} is obtained from its predecessor G_i by contracting the ribbon R_i. Because the edges in R_i are (by definition) not self-loops, the dual graph G_{i+1}^* can be obtained from G_i^* by deleting all edges in R_i^* and their shared degree-2 vertices; see Figure 3.

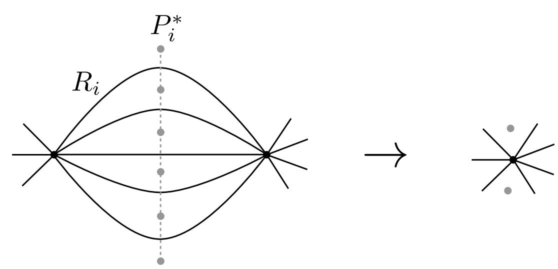

Figure 3. Contracting a ribbon deletes its dual path.

Now consider an arbitrary subset $U \subseteq V$. Let $X = \delta(U)$, and let X^* be the set of the duals of all edges in X. There is a set $\mathcal{K}^*$ of pairwise edge-disjoint cycles in G^* that exactly cover X^*:

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

Fix one such dual cycle $K^* \in \mathcal{K}^*$. There are two cases to consider.

- First, if the primal subgraph K contains exactly one edge $e_i \in T$, then by construction $R_i^* \subseteq K^*$, so Lemma 4.1 implies that $z(K) \geq z(R_i) \geq 2/5$.
- Otherwise, suppose that K contains $k \geq 2$ edges of T. Let e_i^* and e_j^* be the duals of two such edges that are consecutive around K^*. Let Q^* be the subpath of K^* between (and including) e_i^* and e_j^*. Assume without loss of generality that $i < j$; then Q^* is a subgraph of G_i^*. Thus, the definition of central edge and Lemma 4.1 imply that $z(Q) \geq z(R_i)/2 \geq 1/5$. Since the cycle K^* contains $\lfloor k/2 \rfloor \geq k/3$ such subpaths Q^* that are pairwise disjoint, we conclude that $z(K) \geq k/15$.

In both cases, we have shown that $|T \cap K| \leq 15 \cdot z(K)$. Summing over all cycles in $\mathcal{K}^*$ gives us

$$|T \cap \delta(U)| = \sum_{K^* \in \mathcal{K}^*} |T \cap K| \leq \sum_{K^* \in \mathcal{K}^*} 15 \cdot z(K) = 15 \cdot z(\delta(U)),$$

concluding the proof. □

Although T satisfies the first condition in the definition of thinness, it is not necessarily thin, because its cost (with respect to c) could be large. We now describe how to transform T into a $(O(1), O(1))$-thin spanning forest, using an argument of Oveis Gharan and Saberi [14], who described a similar transformation for spanning *trees*. In fact, their argument applies to spanning forests with a fixed number of components with only trivial modifications. We include their proof here for completeness.

Lemma 4.3. *A* $(O(1), O(1))$*-thin spanning forest of* G *with at most* g *components can be computed in polynomial time.*

Proof: First we observe that for some fixed constant α, the algorithm of Lemma 4.2 can return a forest T such that

$$|T \cap \delta(U)| \leq \alpha z(\delta(U)) \text{ for every subset } U \subseteq V$$

even if z is *not* the symmetrization of a feasible solution to the Held-Karp LP. In fact, the success of the algorithm requires only that $z(e) \geq 0$ for every edge e and $z(\delta(U)) \geq 2$ for every subset $U \subseteq V$. We call a function $z \colon E \to \mathbb{R}$ *suitable* if it satisfies these two requirements.

Let $N = n^2/\alpha$; we assume without loss of generality that $N \geq 2$. We computes a sequence $z_0, z_1, \dots, z_N$ of suitable functions and a sequence $T_1, \dots, T_N$ of spanning forests, each with at most g components, and then return the forest T_i with minimum total cost. Our construction begins by setting $z_0(e) := 8\lfloor n^2 z(e)\rfloor/n^2$ for every edge $e \in E$; it is immediate that z_0 is suitable.

For each index $i \geq 1$, we compute the function z_i and spanning forest T_i as follows. First, using Lemma 4.2, we find a spanning forest T_i with at most g components; because z_{i-1} is suitable, by the inductive hypothesis, we have

$$|T_i \cap \delta(U)| \leq \alpha z_{i-1}(\delta(U)) \text{ for every subset } U \subseteq V. \quad (1)$$

Then for each edge $e \in E$, we define

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

We prove that the function z_i is suitable as follows. All values in z_0 are integer multiples of $1/n^2$, and every value in $z_i - z_{i-1}$ is either 0 or $1/n^2$. It follows inductively that all values in z_i are integer multiples of $1/n^2$. An edge e can appear in the forest T_i only if $z_{i-1}(e) > 0$. We conclude that $z_i(e) \geq 0$ for every edge e.

Now fix an arbitrary nonempty subset $U \subsetneq V$. Equation (1) and the definitions of z_i and N imply inductively that

$$\begin{aligned} z_i(\delta(U)) &= z_{i-1}(\delta(U)) - |T_i \cap \delta(U)|/n^2 \\ &\geq z_{i-1}(\delta(U))(1-\alpha/n^2) \\ &\geq z_0(\delta(U))(1-\alpha/n^2)^i \\ &\geq z_0(\delta(U))(1-1/N)^N \\ &\geq z_0(\delta(U))/4. \end{aligned}$$

The definition of z_0 implies that

$$z_0(\delta(U)) > 8(z(\delta(U)) - |\delta(U)|/n^2) > 8(z(\delta(U)) - 1),$$

since at most n^2 edges cross any cut in G. Finally, because z is the symmetrization of a feasible solution to the Held-Karp LP, we have $z(\delta(U)) \geq 2$, which implies $z_0(\delta(U)) > 8$ and therefore $z_i(\delta(U)) > 2$. We conclude that z_i is suitable, as claimed.

Finally, choose an index i such that the cost $c(T_i)$ of forest T_i is minimized; our algorithm returns the forest T_i. Equation (1) implies that

$$|T_i \cap \delta(U)| \leq \alpha z_{i-1}(\delta(U)) \leq \alpha z_0(\delta(U)) \leq 3\alpha z(\delta(U))$$

for every subset $U \subseteq V$. Each edge $e \in E$ appears in at most $n^2 z_0(e) < 3n^2 z(e)$ forests T_j, so

$$\begin{aligned} c(T_i) &\leq \frac{1}{N}\sum_{j=1}^{N} c(T_j) \\ &\leq \frac{3n^2}{N}\sum_{e\in E} z(e)c(e) \\ &= 3\alpha \sum_{e\in E} z(e)c(e) \\ &\leq 3\alpha \sum_{a\in A} x(a)c(a). \end{aligned}$$

(The last inequality follows from the fact that $z(uv)c(uv) \leq x(u{\to}v)c(u{\to}v) + x(v{\to}u)c(v{\to}u)$ for every edge $uv \in E$.) We conclude that T_i is $(3\alpha, 3\alpha)$-thin, as required. □

5. WALKING IN THE FOREST

The last missing ingredient of our main result is an algorithm to transform a thin forest with few components into a small number of closed walks with small total cost that span the input graph. We follow an argument of Asadpour *et al.* [1], with only minor modifications to deal with thin forests instead of thin trees; we include their complete argument here to keep our presentation self-contained. The argument relies on the following classical result of Hoffman [10]; for a more modern presentation, see Schrijver [15, Chapter 11].

Hoffman's Circulation Theorem [10]. *Fix a directed graph* $\vec{G} = (V, A)$ *with upper and lower bounds* $l, u : A \to \mathbb{R}$*. There is a circulation* $f : A \to \mathbb{R}$ *such that* $l(a) \leq f(a) \leq u(a)$ *for every arc* $a \in A$ *if and only if the following two conditions are satisfied:*

(I) $l(a) \leq u(a)$ *for every arc* $a \in A$.

(II) $l(\delta^-(U)) \leq u(\delta^+(U))$ *for every subset* $U \subseteq V$.

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

Lemma 5.1. *Fix positive integer parameters* α *and* s *and a feasible solution* x *to the Held-Karp LP. Given a spanning forest* T *of* G *with at most* k *connected components that is* (α, s)*-thin with respect to* x*, we can compute in polynomial time a sequence* $C_1, \dots, C_{k'}$ *of closed walks in* $\vec{G}$*, for some* $k' \leq k$*, that visit every vertex 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: First we direct the edges of T to obtain a directed subgraph $\vec{T} \subseteq A$. Specifically, for every edge $uv \in T$, we have $u{\to}v \in \vec{T}$ if $c(u{\to}v) \leq c(v{\to}u)$ and $v{\to}u \in \vec{T}$ otherwise. Then for each arc $a \in A$, we define

$$l(a) := \begin{cases} 1 & \text{if } a \in \vec{T} \\ 0 & \text{if } a \notin \vec{T} \end{cases} \quad \text{and} \quad u(a) := l(a) + \lceil 2\alpha\, x(a) \rceil.$$

We verify that these two functions satisfy the conditions of Hoffman's circulation theorem as follows. Condition (I) follows trivially from the fact that $x(a) \geq 0$ for every arc $a \in A$. To verify condition (II), fix a subset $U \subseteq V$. Recall that z is the symmetrization of x. We have a sequence of equations and inequalities

$$
\begin{aligned}
l(\delta^-(U)) &= |\vec{T} \cap \delta^-(U)| && (2)\\
&\le |T \cap \delta(U)| && (3)\\
&\le \alpha z(\delta(U)) && (4)\\
&= \alpha(x(\delta^+(U)) + x(\delta^-(U))) && (5)\\
&= 2\alpha\, x(\delta^+(U)) && (6)\\
&\le u(\delta^+(U)), && (7)
\end{aligned}
$$

where (2) follows from the definition of l, (3) from the definition of $\vec{T}$, (4) from the thinness of T, (5) from the definition of z, (6) from the second constraint in the Held-Karp LP, and (7) from the definition of u.

Hoffman's circulation theorem now implies that there is a feasible circulation f in $\vec{G}$; moreover, because the functions l and u are integer-valued, f can be chosen to be integer-valued. Let $\vec{H}$ be the directed multigraph obtained from $\vec{G}$ by splitting each arc $v{\to}w$ into $f(v{\to}w)$ parallel copies, or removing $v{\to}w$ if $f(v{\to}w) = 0$. Flow conservation implies that $\vec{H}$ is Eulerian. Moreover, $\vec{H}$ contains $\vec{T}$ as a subgraph. Because T has k components, $\vec{H}$ has at most k strongly-connected components, which implies that $\vec{H}$ can be decomposed into closed walks $C_1, C_2, \dots, C_{k'}$ for some $k' \le k$. Every vertex in $\vec{G}$ lies in one of these closed walks. Finally, the total cost of these walks is

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

which completes the proof. □

6. OUR APPROXIMATION ALGORITHM

We are now finally ready to describe our polynomial-time approximation algorithm. Suppose we are given a directed graph $\vec{G} = (V, A)$ embedded on a surface of Euler genus g and a non-negative cost function $c\colon A \to \mathbb{R}$. Let $\mathsf{OPT} \ge 0$ denote the minimum total cost of a closed spanning walk of $\vec{G}$.

We begin by computing an optimal solution x to the Held-Karp LP relaxation for this instance of ATSP. Using Lemma 4.3, we compute a spanning forest T of G with at most g connected components that is $(O(1), O(1))$-thin with respect to x. Then using Lemma 5.1, we then compute a sequence $C_1, \dots, C_k$ of closed walks, for some integer $k \le g$, that visit every vertex of G and have total cost at most $O(\mathsf{OPT})$.

Next, we choose a vertex v_i of each closed walk C_i and define a new instance of ATSP over those $O(g)$ vertices. Specifically, we construct a complete directed graph $\vec{G}' = (V', A')$ with $V' = \{v_1, \dots, v_k\}$, and for each pair of vertices v_i and v_j, we define $c'(v_i{\to}v_j)$ to be the shortest-path distance from v_i to v_j in $\vec{G}$ with respect to the original cost function c.

This new instance of ATSP clearly has a solution of cost $\mathsf{OPT}' \le \mathsf{OPT}$. Using any polynomial-time $f(n)$-approximation algorithm for ATSP in general graphs, we can compute a closed spanning walk C' in $\vec{G}'$ such that $c'(C') \le f(k)\cdot \mathsf{OPT}' \le f(g)\cdot \mathsf{OPT}$, in time polynomial in g. By composing C' with the earlier closed walks $C_1, \dots, C_k$, we obtain a single closed spanning walk C of $\vec{G}$ with total cost $c(C) \le O(\mathsf{OPT}) + f(g)\cdot \mathsf{OPT} = O(f(g))\cdot \mathsf{OPT}$. The overall running time of the algorithm is clearly polynomial.

This completes the proof of Theorem 1.1.

To prove Theorem 1.4, we simply replace the black-box $f(n)$-approximation algorithm with the classical dynamic programming algorithm of Bellman [2] and Held and Karp [8], which solves any n-vertex instance of ATSP *exactly* in $O(2^n n^2)$ time. If no embedding is given as part of the input, we can compute a minimum-genus embedding in $2^{O(g)} n$ time using an algorithm of Kawarabayashi, Mohar, and Reed [12]. The rest of the algorithm and its analysis is identical to Theorem 1.1.

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. *SODA*, 379–389, 2010.

[2] Richard Bellman. Dynamic programming treatment of the traveling salesman problem. *J. ACM* 9(1):61–63, 1962.

[3] Markus Bläser. A new approximation algorithm for the asymmetric tsp with triangle inequality. *Proc., 14th Ann. ACM-SIAM Symp. Discrete Algorithms*, 638–645, 2003.

[4] Chandra Chekuri and Anastasios Sidiropoulos. Approximation algorithms for Euler genus, and related problems. *FOCS*, 2013.

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

[6] Uriel Feige and Mohit Singh. Improved approximation ratios for traveling salesperson tours and paths in directed graphs. *Proc. 10th Ann. APPROX Workshop*, 104–118, 2007. Lect. Notes Comp. Sci. 4627, Springer.

[7] 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.

[8] Michael Held and Richard Karp. A dynamic programming approach to sequencing problems. *J. SIAM* 10(1):196–210, 1962.

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

[10] Alan J. Hoffman. Some recent applications of the theory of linear inequalities to extremal combinatorial analysis. *Proc. 10th Symp. Applied Math.*, 113–128, 1960. Amer. Math. Soc.

[11] Haim Kaplan, Moshe Lewenstein, Nira Shafrir, and Maxim Sviridenko. Approximation algorithms for asymmetric tsp by decomposing directed regular multigraphs. *J. ACM* 52(4):602–626, 2005.

[12] Ken-ichi Kawarabayashi, Bojan Mohar, and Bruce A. Reed. A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of graphs of bounded tree-width. *FOCS*, 771–780, 2008.

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

[14] Shayan Oveis Gharan and Amin Saberi. The asymmetric traveling salesman problem on graphs with bounded genus. *SODA*, 967–975, 2011. SIAM.

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