The Asymmetric Traveling Salesman Problem on Graphs with Bounded Genus

Shayan Oveis Gharan* Amin Saberi*

Abstract

We give a constant factor approximation algorithm for the asymmetric traveling salesman problem when the support graph of the solution of the Held-Karp linear programming relaxation has bounded orientable genus.

1 Introduction

We present the first constant-factor approximation algorithm for the Asymmetric Traveling Salesman Problem (ATSP) for metrics defined by a weighted directed graph with a bounded orientable genus. This is a very natural special case: consider a metric obtained by shortest path distances in a city with one way streets and a constant number of bridges and underpasses.

The result is in fact more general: we can obtain constant factor approximation algorithms when the underlying graph of the fractional solution of the Held-Karp linear programming relaxation has bounded orientable genus. It is easy to see that this is a less strict condition. In fact, it is known that the corner points of the Held-Karp relaxation polytope define very sparse graphs [9] and in practice they often turn out to be planar.

Even though the symmetric version of this problem (STSP) has been studied extensively on Euclidean [1], planar [10, 2, 13] or low-genus metrics [6], to the best of our knowledge, this is the first result of this type for Asymmetric TSP. Our algorithm rounds the solution of the Held-Karp linear programming relaxation. Therefore, it also gives a constant upper bound on the integrality gap. It is worth noting that the best-known constructions that lower bound the integrality gap [5] are also planar.

Our result builds on a central lemma in Asadpour et al. [3] that shows for finding a constant-factor approximation algorithm for ATSP, it is sufficient to find a "thin" tree in the fractional solution. Roughly speaking, a tree is ϵ-thin with respect to a graph G, if it does not contain more than an ϵ-fraction of the edges of G across any cut.

On the other hand, our approach for finding a tree and establishing its thinness is quite different from Asadpour et al. [3]: the embedding of the graph and its geometric dual will be crucial in finding a tree and proving its thinness. In particular, we take advantage of the correspondence between the cutsets of the graph G and cycles of the dual graph G^*. If G^* does not have any short cycles and all the edges of T are far apart in G^*, then T can not contain too many edges from any cut of G and therefore it is thin.

*Department of Management Science and Engineering, Stanford University, Stanford, CA 94305. Email:`{shayan, saberi}@stanford.edu`.

Thin trees were first defined in the graph embedding literature in an attempt to prove a notoriously difficult conjecture by Jaeger on the existence of certain nowhere-zero flows [7]. Our result on the existence of thin trees in graphs with bounded genus also proves a weaker version of Jaeger's conjecture and essentially implies the main result of [18]. Furthermore, our algorithm and its proof is much simpler than the vertex splitting argument of [18] that needs extensive case analysis.

In the rest of this section, let us sketch the main steps of the algorithm and its analysis. The main result of the paper is to find a polynomial-time algorithm for finding an $f(\gamma)/k$-thin tree in a k-edge connected graph of genus γ. In order to that, we first show how to find such a tree if the dual of the graph has high girth. This is sufficient for planar graphs. The dual of every cutset in G is a cycle in G^*. Therefore if G is k-edge connected, G^* has girth at least k. In graphs with genus even slightly bigger than one, high connectivity does not imply high dual girth. In fact, the dual of a graph with high edge connectivity can have several short cycles. In section 4, we show how to remove short cycles from G^* without creating too many connected components in G. For doing this, we have to use "surgical" operations like cutting handles and adding topological disks to the surface . The reader interested in the algorithm for planar graphs can skip this section and reads sections 2,3, and 5.

In the last section, we make the connection between ATSP and thin trees concrete. We show that an algorithm that finds an $O(1)/k$-thin spanning tree in a k-edge connected graph gives a constant factor approximation algorithm for ATSP.

2 Preliminaries

In the Asymmetric Traveling Salesman problem (ATSP), we are given a set V of n points and a cost function $c : V \times V \to \mathbb{R}^+$. The goal is to find the minimum cost tour that visits every vertex at least once. Since we can replace every arc (u, v) in the tour with the shortest path from u to v, we can assume c satisfies the triangle inequality.

In this paper, we refer to multi-graphs (graphs with loops and parallel edges) simply as graphs. Let $G(V, E)$ be a weighted undirected graph with cost function $c(e)$, and $F \subseteq E$ be a collection of edges. We define $c(F) := \sum_{e \in F} c(e)$.

Definition 2.1 *A subset $F \subseteq E$ is α-thin with respect to G, if for each set $U \subset V$,*

$$|F(U, \overline{U})| \leq \alpha |E(U, \overline{U})|,$$

where $F(U, \overline{U})$ and $E(U, \overline{U})$ are respectively the sets of edges of F and E that are in the cut $(U, \overline{U})$. We say F is (α, σ)-thin with respect to G, if it is α-thin and $c(F) \leq \sigma c(E)$.

Given an instance of ATSP corresponding to the cost function $c : V \times V \to \mathbb{R}^+$, we can obtain a lower bound on the optimum value by considering the following linear programming relaxation defined on the complete bidirected graph with vertex set V:

$$\begin{aligned} \min \quad & \sum_a c(a) x_a && (1) \\ \text{s.t.} \quad & \boldsymbol{x}(\delta^+(S)) \geq 1 \quad && \forall S \subset V, \quad (2) \\ & \boldsymbol{x}(\delta^+(v)) = \boldsymbol{x}(\delta^-(v)) = 1 \quad && \forall v \in V, \\ & x_a \geq 0 && \forall a. \end{aligned}$$

In the above linear program $\delta^+(S)$ ($\delta^-(S)$) denotes the set of directed edges leaving (entering) S in the bidirectional complete graph on V. Also define $\delta(S) = \delta^+(S) \cup \delta^-(S)$.

This linear programming relaxation is known as the Held-Karp relaxation [11], and its optimum value, which we denote by $\mathrm{OPT}_{\mathrm{HK}}$, can be computed in polynomial time.

Let x be a feasible solution of Held-Karp relaxation (1). Our approximation algorithm will be based on rounding x to an integral solution. For this rounding, it turns out that it is sufficient to find a spanning tree that is thin with respect to x. The following definition is a natural extension from undirected graph that we incorporate here from [3] with a slight modification.

Definition 2.2 *We say that a spanning tree T is α-thin with respect to x, if for each set $U \subset V$,*

$$|T(U, \overline{U})| \leq \alpha x(\delta(U)),$$

where $T(U, \overline{U})$ is the set of the edges of T that are in the cut $(U, \overline{U})$. Also we say that T is (α, σ)-thin with respect to x, if it is α-thin and moreover it is possible to orient the edges of T into T^* such that

$$c(T^*) \leq \sigma c(x).$$

2.1 Surfaces and graph embedding

We also need to recall some of the concepts in topological graph theory. By a *surface*, we mean a compact connected 2-manifold without boundary. It is well known that all surfaces are classified into the sphere with γ handles (denoted by S_γ) or the crosscaps (denoted by N_γ) and are called orientable and non-orientable surfaces respectively. Throughout this paper by a surface we mean an orientable surface, and all the theorems have been proved for orientable surfaces.

In the above definition, γ represents the genus of the surface. An equivalent definition for the genus of an orientable surface is the maximum number of disjoint simple closed curve which can be cut from the orientable surface without disconnecting it.

An embedding of a graph G into a surface Σ, is a homeomorphism $i : G \rightarrow \Sigma$ of G into Σ. The *orientable genus* of a graph G is the minimum γ such that G has an embedding in S_γ. For example, planar graphs have genus zero.

Let $G(V, E)$ be a graph embedded on a surface Σ. A set $S \subseteq E$ on Σ is *separating* if $\Sigma - S$ is disconnected; otherwise S is called *non-separating*. For instance, the definition of orientable genus implies that any set of $\gamma(\Sigma) + 1$ disjoint cycles of G is separating.

Suppose that we have embedded G on a surface Σ. The *geometric dual* of G on Σ, G^*, is defined similar to the planar graphs. Particularly, The vertices of G^* correspond to the faces of G. The edges of G^* are in bijective correspondence $e \rightarrow e^*$ with the edges of G, and the edge e^* joins the vertices corresponding to the faces containing e in G. For a more extensive discussion of embeddings of graphs in surfaces, see [15].

3 Constructing a thin-tree

Let $G(V, E)$ be a connected graph embedded on an orientable surface, and G^* be its geometric dual. The dual-girth of G, denoted by $g^*(G)$ is the length of the shortest cycle in G^*. The main result of this section is the following lemma.

Lemma 3.1 *A connected graph embedded on an orientable surface with genus* γ *and dual-girth* $g*$ *has a spanning tree with thinness* $\frac{2\alpha}{g^*}$, *where* $\alpha = 4 + \lfloor 2\log_2(\gamma + \frac{3}{2})\rfloor$. *Furthermore, such a tree can be found in polynomial time.*

We will prove this lemma in the rest of this section. First note that if $g^* = 1$, the lemma holds for trivial reasons. Therefore, without loss of generality assume that $g^* > 1$. That implies that no face of G can have two copies of an edge. In particular, G does not have any cut edge.

Define the distance of two edges in a graph to be the closest distance between their endpoints. Our most basic tool for establishing the thinness of a tree T in G is to relate it to the pairwise distance of its corresponding edges T^* in G^*. If G^* does not have any short cycles and all the edges of T^* are far apart in G^*, then the tree can not contain too many edges from any cut. We will establish that for any subset of edges:

Lemma 3.2 *Let* F *be a set of edges in* G *and* F^* *be the corresponding edges in the dual. If for some* $m \leq g^*(G)$, *the distance between each pair of edges in* F^* *is at least* m, *then* F *is* $\frac{1}{m}$*-thin in* G.

Proof: Consider a cut $S = (U, \overline{U})$ in G. Let us start by showing that S^* is a collection of edge-disjoint cycles $C_1, C_2, \ldots, C_l$ in G^*. This is because the number of edges from S^* incident to a vertex in G^* is equal to the intersection of S with its corresponding face in G and that is an even number. Otherwise, either that face contains two copies of an edge of S, or one could find a path P in that face such that $P \cap S = \emptyset$, while the endpoints of P are in different sides of the cut, which are both impossible.

Because the distance of each pair of edges in F^* is at least m, F^* can not have more than $\max(1, \lfloor length(C_i)/m\rfloor)$ edges in C_i, for $1 \leq i \leq l$. Therefore,

$$|F^*| \leq \sum_{i=1}^{l} \max(1, \lfloor \frac{length(C_i)}{m} \rfloor) = \sum_{i=1}^{l} \lfloor \frac{length(C_i)}{m} \rfloor \leq \frac{|S^*|}{m}.$$

Note that the equality holds by the assumption $length(C_i) \geq g^* \geq m$. Thus the number of edges of F in the cut $(U, \overline{U})$ is no more than $\lfloor |(U, \overline{U})|/m \rfloor$ and F is a $1/m$-thin. □

Considering the above Lemma, our goal will be to find a set of edges in G^* that are sufficiently far. We will do this by finding long threads iteratively and selecting one edge from each thread.

A *thread* in a graph G is a maximal subgraph of G which is

- a path whose internal vertices all have degree 2 in G and its endpoints have degree at least 2, or
- a cycle in which all vertices except possibly one have degree 2.

Let us start by showing the existence of long threads. That is a straightforward application of the result of Goddyn et al. [8].

Lemma 3.3 *A graph with minimum degree* 2 *and girth* g, *embedded on a surface with genus* γ *has a thread of length at least* g/α.

Algorithm 1 Finds a thin tree in a graph with large dual-girth

Input: A connected graph G embedded on an orientable surface with genus γ, and its dual G^* with girth g^*.

Output: A spanning tree T with thinness at least $g^*/2\alpha$.

```
F* ← ∅
while there exists an edge in G* do
    Find a thread P of length at least g*/α in G*.
    Add the middle edge of P to F* and remove it from G*. If P is a cycle, define its middle edge
      to be the one with the maximum distance from the high-degree vertex.
    Iteratively delete all the degree one vertices with their incident edges.
end while
return A spanning tree T ⊆ F, where F is the set of edges corresponding to F* in G.
```

Proof: Let H be a graph satisfying the conditions of the theorem and H' be the graph obtained by iteratively replacing the vertices of degree 2 in H with an edge. In other words, let H' be the graph obtained by replacing every thread in H by an edge. By Goddyn et al. [8, Theorem 3.2], every graph with minimum degree 3 embedded on an orientable surface with genus γ has a cycle of length at most α. Therefore H' has a cycle of length at most α. Now it is easy to see that at least one of the edges of that cycle is obtained from a thread of length at least $\frac{g}{\alpha}$ in H. □

Because of the above lemma, Algorithm 1 terminates in polynomial time. The algorithm has an equivalent description in terms of the original graph G. Roughly speaking, in each iteration, we find a collection of consecutive parallel edges, add the middle edge from that collection to F and contract the end points. The embedding is crucial for the execution of this procedure because it provides a notion of a middle edge, and the notion of consecutive parallel edges (parallel edges that form a face).

It is also worth noting that $|F|$ may end up being bigger than $|V(G)| - 1$ in an execution of Algorithm 1. This is because a thread in G^* may be equivalent to a collection of parallel *loops*. The next lemma immediately proves Lemma 3.1.

Lemma 3.4 *The set F computed in Algorithm 1 is connected and spanning in G. Furthermore, the pairwise distance of the edges of F^* in G^* is at most $g^*/2\alpha$.*

Proof: For the proof of the first statement, consider a non-empty cut $S = (U, \overline{U})$ in G, and let S^* be its dual. As we argued in the proof of Lemma 3.2, S^* is a collection of cycles. It is also easy to see that Algorithm 1 selects at least one edge from each cycle.

For the second statement, first observe that after adding an edge e to F^*, the algorithm immediately removes all the edges that are of distance less than $g^*/2\alpha$ from e. This is because all these edges are a part of the thread and therefore they are deleted sequentially.

Furthermore, although each iteration of the while loop may increase the distance of some pairs of edges, it never increases the distance of two edges that are closer than g^*/α. Therefore, the distance of any pairs of edges that are closer than g^*/α remains the same until one of them is deleted. □

4 Increasing the girth of dual graph

Let $G(V, E)$ be a planar graph and G^* be its geometric dual. In the previous section we showed that the girth of G^* plays an important role in finding a thin spanning tree in G. By Whitney's theorem [17], $S \subseteq E$ is a cutset (minimal cut) in a *planar* graph G if and only if S^* is a cycle in G^*. Therefore, if G is planar and k-edge connected, the girth of G^* will be at least k. Unfortunately, this relation does not hold for non-planar graphs as their dual may have very small cycles.

In the rest of this section we show that we can get rid of these small cycles by deleting their edges. Deleting these edges may result in making the graph disconnected. We will try to increase the girth as much as possible while creating only a small number of connected components.

Later, we will find a thin tree in every connected component and merge them into a spanning tree using an arbitrary set of edges from the original graph. Since the number of connected components is small, this is possible with only a small loss in the thinness of the final spanning tree. In fact, in the statement of Theorem 5.1, the $O(\sqrt{\gamma})$ dependence of the thinness on the genus of the surface comes from the balance between the number of connected components at the end of the procedure with the girth of the final graph in Lemma 4.1.

Lemma 4.1 *Let G be a k-edge connected graph embedded on an orientable surface with genus $\gamma > 0$, and G^* its geometric dual. There is a polynomial time algorithm that deletes some of the edges of G, and obtains a new graph H with the dual H^* such that H has at most $2\sqrt{\gamma}$ connected components while $girth(H^*) \geq \frac{k}{3\sqrt{\gamma}}$.*

The algorithm considers each small cycle in G^*, and simply deletes its corresponding edges from G, and updates G^* accordingly.

Algorithm 2 Constructing a high girth dual

Input: A k-edge connected graph G embedded on a surface with genus $\gamma > 0$ and its dual G^*.
Output: A graph H and its dual H^* where $\kappa(H) \leq 2\sqrt{\gamma}$ and $girth(H^*) \geq \frac{k}{3\sqrt{\gamma}}$.

```
while girth(G*) < k/(3√γ) do
    Find a cycle C* of length less than k/(3√γ) in G*.
    Delete its corresponding edges from G, and update G* accordingly.
end while
return G and G*
```

To see the effect of this cycle deletion process we use the following lemma.

Lemma 4.2 *Let G be a non-planar graph embedded on an orientable surface Σ with genus γ, and G^* its geometric dual. If C is the set of corresponding edges of a cycle C^* in G^*, then either $G - C$ can be embedded on a surface with smaller genus, or $\kappa(G - C) > \kappa(G)$, where $\kappa(G)$ is the number of connected components of G.*

Proof: We define a surgery operation in which the surface Σ is cut along the simple curve defined by C^*, and then a topological disk is attached to each side of the cut. We will show how $G - C$ can be embedded on the resulting surface (or surfaces).

If C^* is a non-separating cycle in G, then cutting along C^* and adding the two topological disks removes one of the handles of Σ, thus giving rise to a unique connected surface with smaller genus.

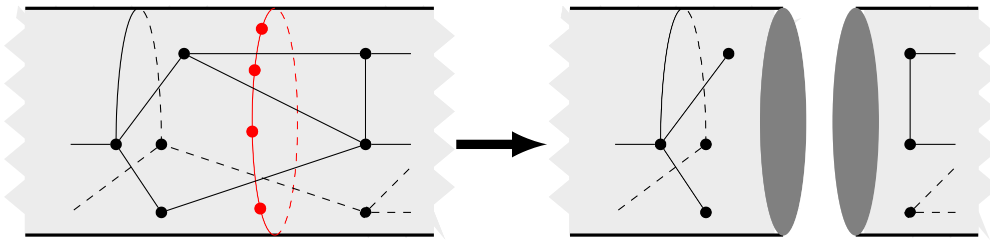

Figure 1: Cutting a surface along a non-separating cycle. The red cycle in the left diagram is a cycle in the dual. We cut the surface along the cycle, remove the edges that are cut from the original graph and attach two topological disks where the cut is made (right diagram).

The edges of C crossing the curve C^* are removed from G, therefore $G - C$ is embeddable on the new surface. Figure 1, shows the details of this operation.

If C^* is a separating cycle, then cutting Σ along C^*, creates two surfaces Σ_1 and Σ_2, where each one contains a connected component of $G - C$. The sum of genera of Σ_1 and Σ_2 is γ. Therefore, in this case only $\kappa(G - C) > \kappa(G)$.

In both cases the dual embedding is obtained by removing the edges and vertices of the cycle along which the surface is cut and adding two vertices c_l and c_r to the disks attached to left and right side of the cut. The edges of the left side of C^* are attached to c_l and the rest of them to c_r. □

Roughly speaking, the above lemma says that by removing the edges corresponding to a cycle in G^* from G, we will either decrease its genus or increase its number of connected components. As we will show next, the number of connected components in the final graph is bounded and therefore the procedure has to stop after deleting a bounded number of cycles.

Proof: [**Lemma 4.1**] We show that the algorithm 2 works correctly. First of all, note that the algorithm eventually terminates, even if it has to delete all of the edges of G. Hence, it runs in polynomial time.

Let H be the output of the algorithm and H^* be its dual. When the algorithm terminates, the girth of H^* is at most $\frac{k}{3\sqrt{\gamma}}$. Therefore, the only thing we need to prove is that $\kappa(H) \leq 2\sqrt{\gamma}$.

Suppose that the while loop is finished after m iterations. Consider the total number of edges deleted during the execution of the algorithm. Since the number of deleted edges in each iteration of the while loop is no more than $\frac{k}{3\sqrt{\gamma}}$, at most $\frac{mk}{3\sqrt{\gamma}}$ edges have been deleted totally. On the other hand, all the edges between different connected components of H have been deleted in the loop. Because G was k-edge connected, there was originally $\frac{\kappa(H)k}{2}$ edges between these components, where all of them have been deleted. Therefore,

$$\frac{\kappa(H)k}{2} \leq \frac{mk}{3\sqrt{\gamma}}. \tag{3}$$

In order to bound $\kappa(H)$, we need an upper bound on m. By Lemma 4.2, we have

$$m \leq (\gamma(G) - \gamma(H)) + (\kappa(H) - \kappa(G)) \leq \gamma + \kappa(H).$$

By combining this with inequality (3) we get

$$\frac{\kappa(H)}{2} \leq \frac{\gamma + \kappa(H)}{3\sqrt{\gamma}} \quad \Rightarrow \quad \frac{\kappa(H)}{6} \leq \frac{\sqrt{\gamma}}{3}.$$

This implies that $\kappa(H) \leq 2\sqrt{\gamma}$. □

5 Thin trees, Goddyn's conjecture and ATSP

The algorithms presented in sections 3 and 4 and their analysis imply the following result:

Theorem 5.1 *A k-edge connected graph embeddable on an orientable surface with genus γ has a spanning tree with thinness $\frac{f(\gamma)}{k}$ for some function $f(\gamma) = O(\sqrt{\gamma}\log(\gamma))$. Such a spanning tree can be found in polynomial time.*

Proof: If $\gamma = 0$, i.e., G is planar then by [17], g^*, the girth of G^* will be at least k. By Lemma 3.1, Algorithm 1 can find a spanning tree in G with thinness $10/k$.

If $\gamma > 0$, then run Algorithm 2 to obtain a subgraph H of G which has by Lemma 4.1 at most $2\sqrt{\gamma}$ connected components while $girth(H^*) \geq \frac{k}{3\sqrt{\gamma}}$. Again, use Algorithm 1 to find a spanning tree in each connected component of H with thinness $\frac{6\sqrt{\gamma}\alpha}{k}$ for $\alpha = 4 + \lfloor 2\log_2(\gamma + \frac{3}{2})\rfloor$. By the matroid property of spanning trees one can extend this collection of spanning trees to a spanning tree of G by adding a set $F \subset E(G)$ of size at most $2\sqrt{\gamma}$. Since G is k-edge connected the thinness increases by at most $2\sqrt{\gamma}/k$. Therefore since $\alpha \geq 5$, the resulting tree is $\frac{7\sqrt{\gamma}\alpha}{k} = \frac{O(\sqrt{\gamma}\log\gamma)}{k}$ thin. □

An equivalent way to state above theorem is that for every orientable surface Σ, there exists a function f_Σ such that, for any $\epsilon > 0$, every $f_\Sigma(\epsilon)$-edge connected graph G embedded in S has an ϵ-thin spanning tree. This can be considered as a partial result for the following conjecture of Goddyn [7].

Conjecture 5.2 (Goddyn [7]) *There exists a function $f(\epsilon)$ such that, for any $0 < \epsilon < 1$, every $f(\epsilon)$-edge connected graph has an ϵ-thin spanning tree.*

Goddyn's conjecture is intimately related to the asymmetric traveling salesman problem and the integrality gap of Held-Karp relaxation. This is established by Asadpour et al. [3] through the following theorem:

Theorem 5.3 (Asadpour et al. [3]) *Assume that we are given an (α, s)-thin spanning tree T with respect to the LP solution $\boldsymbol{x}$ of cost $c(\boldsymbol{x}) \leq C \times OPT_{HK}$. Then we can find a Hamiltonian cycle of cost no more than $(2\alpha + s)c(\boldsymbol{x}) = C(2\alpha + s)OPT_{HK}$ in polynomial time.*

The above theorem relies on a stronger notion of thinness which takes into consideration the costs of edges. Proposition 5.4 makes this connection more concrete.

Proposition 5.4 *Suppose there exists a non-decreasing function $g(k)$ such that every k-edge connected graph contains a $\frac{g(k)}{k}$-thin spanning tree. Then any* weighted *k-edge connected graph contains a $(\frac{2g(k)}{k}, \frac{2g(k)}{k})$-thin spanning tree.*

Proof: Let G_0 be a weighed k-edge connected graph with cost function $c(e)$ for each edge e. Select a $\frac{g(k)}{k}$-thin spanning tree T_0, in G_0, and remove its edges. Call this new graph G_1. Note that each cut $(U, \overline{U})$ of G_0 will lose at most $\frac{g(k)}{k}|G_0(U, \overline{U})|$ of its edges. As the size of the minimum cut in G_0 is k, G_1 will be $(k - g(k))$-edge connected.

Similarly, find a

$$\frac{g(k - g(k))}{k - g(k)} \leq \frac{g(k)}{k - g(k)}$$

thin spanning tree T_1 in G_1. The inequality holds by the monotonicity assumption on $g(k)$. Remove the edges of T_1 to obtain a $(k - 2g(k))$-edge connected graph G_2.. Repeat this algorithm on G_2 to obtain $k/2g(k)$ spanning trees $T_0, \ldots, T_{k/2g(k)-1}$, where for each i, T_i is a $\frac{g(k)}{k-ig(k)}$-thin spanning tree of the $(k - ig(k))$-edge connected graph G_i.

Because G_i is a spanning subgraph of G_0, any spanning thin tree of G_i will be spanning and thin in G_0. Moreover, since $0 \leq i \leq k/2g(k) - 1$ and

$$\frac{g(k)}{k - ig(k)} \leq \frac{2g(k)}{k},$$

each T_i is a $\frac{2g(k)}{k}$-thin spanning tree in G_0. Among the selected trees find the one with the smallest cost. Let T_j be that tree. We have

$$\frac{k}{2g(k)} c(T_j) \leq \sum_{i=0}^{k/2g(k)-1} c(T_i) \leq c(G_0).$$

Thus T_j is a $(\frac{2g(k)}{k}, \frac{2g(k)}{k})$-thin spanning tree of G_0.

□

In the proof of Theorem 5.1 we give a polynomial-time algorithm that finds an $O(\sqrt{\gamma}\log\gamma)/k$-thin spanning tree in a k-edge connected graph embedded on an orientable surface with genus γ. This result plus the above proposition gives a constant factor approximation algorithm for ATSP when γ is constant. The next theorem establishes this claim.

Theorem 5.5 *Given a feasible solution x of the Held-Karp linear program (1), embedded on an orientable surface with genus γ, there is a polynomial-time algorithm that finds a hamiltonian cycle with a cost that is within an $O(\sqrt{\gamma}\log\gamma)$ of the cost of x. In particular, the approximation factor of the algorithm is at most $22.5(1 + \frac{1}{n})$ when the underlying graph is planar.*

Proof: Let x be a feasible solution of LP (1) that can be embedded on a surface with genus γ. Construct an undirected version of x by defining $y_{\{i,j\}} = x_{ij} + x_{ji}$. Define a new cost function $c'(\{u, v\}) = \min\{c(u, v), c(v, u)\}$.

Round down the fractions in y to the nearest multiple of $1/n^3$. Construct the integral weighted graph H by adding $n^3 y_{\{i,j\}}$ parallel edges between every pair i and j. Since the size of the support of y is less than n^2, we may loose at most $\frac{1}{n^3}n^2 = \frac{1}{n}$ fractions while we are rounding down the edge fractions and therefore H is $n^3(2 - \frac{1}{n})$-edge connected.

Theorem 5.1 finds a $\frac{\beta}{n^3(2-\frac{1}{n})}$-thin spanning tree in H, for $\beta = 7\sqrt{\gamma}\alpha$ if $\gamma > 0$ and $\beta = 10$ if H is planar. Use Proposition 5.4 to compute a $(\frac{2\beta}{n^3(2-\frac{1}{n})}, \frac{2\beta}{n^3(2-\frac{1}{n})})$-thin tree T in H with respect to cost function c'. It is easy to see that it is possible to orient the edges of T into T^* such that $c'(T) = c(T^*)$. Since the size of each cut of H is at most n^3 times of that of x and $c(H) \leq n^3 c(y) \leq n^3 c(x)$, T^* is $(\frac{2\beta}{(2-\frac{1}{n})}, \frac{2\beta}{(2-\frac{1}{n})})$-thin with respect to x. Therefore, using Theorem 5.3, we can find a Hamiltonian cycle of cost no more than

$$\left(2\frac{2\beta}{2-\frac{1}{n}} + \frac{2\beta}{2-\frac{1}{n}}\right)c(x) \leq 3\beta(1+\frac{1}{n})c(x)$$

in polynomial time.

Since $\beta = O(\sqrt{\gamma}\log\gamma)$ is only a function of γ, we have a constant factor approximation for ATSP when the genus of the graph obtained by x is bounded by a constant. In particular if $\gamma = 0$, the above calculation shows that we have a $30(1+\frac{1}{n})$-approximation algorithm. A slightly better optimization of parameters and a minor change of Algorithm 1 leads to a $22.5(1+\frac{1}{n})$.

We should also add that Mohar [14] proves for any constant γ, there is a linear time algorithm that finds an embedding of a given graph on the orientable surface with genus γ, if such an embedding exists. Therefore, the embeddability condition in the above Theorem can be checked in polynomial time for any constant γ. □

Remark 5.6 *It is worth noting that the genus of an extreme point solution of Held-Karp relaxation instance with n vertices can be as large as $\Omega(n)$. In fact, for any odd r, it is possible to construct an extreme point on r^2 vertices that has K_r as a minor. Such an extreme point can be obtained by the same construction as Carr and Vempala [4, Theorem 3.5] applied to K_r.*

An argument similar to the proof of Theorem 5.5 shows that Goddyn's conjecture implies constant integrality gap of the Held-Karp relaxation of ATSP. Furthermore, an algorithmic proof of Goddyn's conjecture implies a constant factor approximation algorithm for ATSP.

Corollary 5.7 *If Goddyn's conjecture is true for some function $f(\epsilon) = O(1/\epsilon)$, then the integrality gap of Held-Karp relaxation is bounded from above by a constant.*

5.1 Nowhere-zero flows and Jaeger's conjecture

Goddyn's conjecture was inspired by the study of nowhere-zero flows and in particular in attempting Jaeger's conjecture [12]. Here, we just state the Jaeger's conjecture and refer the reader to Seymour [16] for more information.

Conjecture 5.8 (Jaeger [12]) *If G is $4k$-edge connected, then for some orientation of G, every subset $S \subset V(G)$ satisfies*

$$(k-1)|\delta^-(S)| \leq k|\delta^+(S)| \leq (k+1)|\delta^-(S)|.$$

Jaeger's conjecture has not been proved for any positive integer k yet. For $k = 1$, this is Tutte's 3-flow conjecture and is proved only for planar graphs. Goddyn's conjecture implies a weaker version of Jaeger's conjecture in which $4k$ is replaced by an arbitrary function of k. Even this version is still open [8].

Previously, Zhang [18] proved Jaeger's conjecture on graphs with bounded genus. The dependence of his result on the genus of the graph is pretty much similar to ours: he proves Jaeger's conjecture for graphs with connectivity $O(\sqrt{\gamma}k)$ embedded on a surface with genus γ. Since Goddyn's conjecture implies Jaeger's conjecture with the same parameters, our result can be seen as a strengthening of [18]. On the other hand, the result of [18] is based on a vertex splitting argument that needs extensive case analysis. It seems that our approach for proving the stronger result has also lead to a much simpler proof.

As a final note, it is easy to extend the polynomial-time dynamic programming algorithm for solving TSP on graphs with bounded treewidth to ATSP. Nevertheless, we still do not know if every k-edge connected graph with bounded treewidth has an $O(1)/k$-thin tree. The answer to this question will be helpful in finding thin trees in families of graphs with excluded minors.

Acknowledgements. The authors would like to thank Arash Asadpour, Luis Goddyn, Michel Goemans, Sergey Norin, Dana Randall, David Shmoys, Prasad Tetali, and Jan Vondrak for discussions and comments on an earlier version of this paper.

References

[1] S. Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. In *Journal of the ACM*, pages 2–11, 1996.

[2] S. Arora, M. Grigni, D. Karger, P. Klein, and A. Woloszyn. A polynomial-time approximation scheme for weighted planar graph tsp. In *SODA '98: Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*, pages 33–41, Philadelphia, PA, USA, 1998. Society for Industrial and Applied Mathematics.

[3] A. Asadpour, M. X. Goemans, A. Madry, S. Oveis Gharan, and A. Saberi. O(logn/loglogn) approximation to the asymmetric traveling salesman problem. *to appear in SODA 2010*, 2010.

[4] R. D. Carr and S. Vempala. Towards a 4/3 approximation for the asymmetric traveling salesman problem. In *SODA '00: Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, pages 116–125, Philadelphia, PA, USA, 2000. Society for Industrial and Applied Mathematics.

[5] M. Charikar, M. X. Goemans, and H. Karloff. On the integrality ratio for the asymmetric traveling salesman problem. *Math. Oper. Res.*, 31(2):245–252, 2006.

[6] E. D. Demaine, M. Hajiaghayi, and B. Mohar. Approximation algorithms via contraction decomposition. In *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 278–287, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[7] L. A. Goddyn. Some open problems i like. available at `http://www.math.sfu.ca/~goddyn/Problems/problems.html`.

[8] L. A. Goddyn, P. Hell, A. Galluccio, and A. Galluccio. High girth graphs avoiding a minor are nearly bipartite. *J. Comb. Theory (B)*, 83:1–14, 1999.

[9] M. X. Goemans. Minimum bounded degree spanning trees. In *FOCS*, pages 273–282, 2006.

[10] M. Grigni, E. Koutsoupias, and C. Papadimitriou. An approximation scheme for planar graph tsp. In *FOCS '95: Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, page 640, Washington, DC, USA, 1995. IEEE Computer Society.

[11] M. Held and R. Karp. The traveling salesman problem and minimum spanning trees. *Operations Research*, 18:1138–1162, 1970.

[12] F. Jaeger. On circular flows in graphs, finite and infinite sets. *Colloquia Mathematica Societatis Janos Bolyai 37*, pages 391–402, 1984.

[13] P. N. Klein. A linear-time approximation scheme for planar weighted tsp. In *FOCS '05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 647–657, Washington, DC, USA, 2005. IEEE Computer Society.

[14] B. Mohar. Embedding graphs in an arbitrary surface in linear time. In *STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 392–397, New York, NY, USA, 1996. ACM.

[15] B. Mohar and C. Thomassen. *Graphs on Surfaces.* Johns Hopkins University Press, Baltimore, 2001.

[16] P. D. Seymour. Nowhere zero flows. In R. L. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of combinatorics*, volume 1. MIT Press, Cambridge, MA, USA, 1995.

[17] H. Whitney. Non-separable and planar graphs. *Trans. Amer. Math. Soc*, 34(2):339–362, 1932.

[18] C. Q. Zhang. Circular flows of nearly eulerian graphs and vertex-splitting. *J. Graph Theory*, 40(3):147–161, 2002.