An $O(\log n/\log\log n)$-approximation Algorithm for the Asymmetric Traveling Salesman Problem

Arash Asadpour* Michel X. Goemans† Aleksander Mądry‡ Shayan Oveis Gharan§
Amin Saberi¶

Abstract

We consider the Asymmetric Traveling Salesman problem for costs satisfying the triangle inequality. We derive a randomized algorithm which delivers a solution within a factor $O(\log n/\log\log n)$ of the optimum with high probability.

1 Introduction

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 a 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.

When the costs are symmetric, i.e. when for every $u, v \in V$, $c(u, v) = c(v, u)$, there is a factor $3/2$ approximation algorithm due to Christofides [8]. This algorithm first finds a minimum cost spanning tree T on V, then finds the minimum cost Eulerian augmentation of that tree, and finally shortcuts the corresponding Eulerian walk into a tour.

In this paper, we give an $O(\log n/\log\log n)$ approximation algorithm for the general asymmetric version. This factor finally breaks the $\Theta(\log n)$ barrier from Frieze et al. [12] and subsequent improvements [3, 16, 11]. Our approach for ATSP has similarities with Christofides' algorithm; we first construct a spanning tree with special properties. Then we find a minimum cost Eulerian augmentation of this tree, and finally, shortcut the resulting Eulerian walk. For undirected graphs, being Eulerian means being connected and having even degrees, while for directed graphs it means being (strongly) connected and having the indegree of every vertex equal to its outdegree.

A simple flow argument using Hoffman's circulation theorem [24] shows that if the tree chosen in the first step is "thin" then the cost of the Eulerian augmentation is within a factor of the "thinness" of the (asymmetric) Held-Karp linear programming (LP) relaxation value ($\mathrm{OPT}_{\mathrm{HK}}$) [17]. This flow argument works irrespectively of the actual directions of the (directed) arcs correponding to the (undirected) edges of the tree. Roughly speaking, a *thin* tree with respect to the optimum solution $\boldsymbol{x}^*$ of the Held-Karp relaxation is a spanning tree that, for every cut, contains a small multiple (the *thinness*) of the corresponding value of $\boldsymbol{x}^*$ in this cut when the direction of the arcs are disregarded.

A key step of our algorithm is to find a thin tree of small cost compared to the LP relaxation value $\mathrm{OPT}_{\mathrm{HK}}$. For this purpose, we consider the distribution with maximum entropy among all those with marginal probabilities obtained from the symmetrized LP solution (scaled by $1 - 1/n$). From the optimality conditions of a convex programming formulation, we derive that this maximum entropy distribution corresponds to sampling a tree T with probability proportional to $\prod_{e\in T} \lambda_e$ for appropriately defined λ_e's for $e \in E$. We develop a simple iterative algorithm for approximately computing these λ_e's efficiently. An important property of this scheme is that the events correponding to edges being present in the sampled tree are negatively correlated. This means that the well-known Chernoff bound for the independent setting still holds, see Panconesi and Srinivasan [23].The proof of the $O(\log n/\log\log n)$ thinness of the sampled tree is based on this tail bound.

The high level description of our algorithm can be found in Figure 1. The proof of our main Theorem 6.3 also gives a more formal overview of the algorithm.

2 Notation

Before describing our approximation algorithm for ATSP in details, we need to introduce some notation. Throughout this paper, we use $a = (u, v)$ to denote the arc (directed edge) from u to v and $e = \{u, v\}$ for an undirected edge. Also we use A (resp. E) for the set of arcs (resp. edges) in a directed (resp. undirected) graph.

For a given function $f : A \to \mathbb{R}$, the cost of f is defined

*Stanford University, Department of Management Science and Engineering. asadpour@stanford.edu.

†MIT, Department of Mathematics. Supported by NSF contract CCF-0829878 and by ONR grant N00014-05-1-0148.goemans@math.mit.edu.

‡MIT, Computer Science and Artificial Intelligence Laboratory. Supported by Fulbright Science and Technology Award, by NSF contract CCF-0829878, and by ONR grant N00014-05-1-0148. madry@mit.edu.

§Stanford University, Department of Management Science and Engineering. shayan@stanford.edu.

¶Stanford University, Department of Management Science and Engineering. saberi@stanford.edu.

INPUT: A set V consisting of n points and a cost function $c : V \times V \to \mathbb{R}^+$ satisfying the triangle inequality.
OUTPUT: $O(\frac{\log n}{\log \log n})$-approximation for the Asymmetric Traveling Salesman Problem on V.

ALGORITHM:

1. Solve the Held-Karp LP relaxation of the ATSP instance to get an optimum extreme point solution $\boldsymbol{x}^*$. [See LP (3.1).] Define $\boldsymbol{z}^*$ by (3.5); $\boldsymbol{z}^*$ can be interpreted as the marginal probabilities on the edges of a probability distribution on spanning trees.
2. Sample $\Theta(\log n)$ spanning trees T_j's from a distribution $\tilde{p}(\cdot)$ that approximates the maximum entropy distribution among all the distributions that approximately preserve the marginal probabilities imposed by $\boldsymbol{z}^*$. Let T^* be the tree with minimum (undirected) cost among all the sampled trees. [See Sections 4 and 5.]
3. Orient each edge of T^* so as to minimize its cost. Find a minimum cost integral circulation that contains the oriented tree $\vec{T}^*$. Shortcut this multigraph and output the resulting Hamiltonian cycle.

Figure 1: An $O(\log n/\log\log n)$-approximation algorithm for the ATSP.

as follows:

$$c(f) := \sum_{a \in A} c(a) f(a).$$

For a set $S \subseteq A$, we define

$$f(S) := \sum_{a \in S} f(a).$$

We use the same notation for a function defined on the edge set E of an undirected graph. For $U \subseteq V$, we also define the following sets of arcs:

$$\begin{aligned} \delta^+(U) &:= \{a = (u,v) \in A : u \in U, v \notin U\}, \\ \delta^-(U) &:= \delta^+(V \backslash U) \\ A(U) &:= \{a = (u,v) \in A : u \in U, v \in U\}. \end{aligned}$$

Similarly, for an undirected graph $G = (V, E)$, $\delta(U)$ denotes the set of edges with exactly one endpoint in U, and $E(U)$ denotes the edges entirely within U, i.e. $E(U) = \{\{u, v\} \in E : u \in U, v \in U\}$.

Throughout the paper, log denotes the natural logarithm.

3 The Held-Karp Relaxation

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:

$$\min \sum_a c(a) x_a \tag{3.1}$$

$$\text{s.t. } \boldsymbol{x}(\delta^+(U)) \geq 1 \qquad \forall U \subset V, \tag{3.2}$$

$$\boldsymbol{x}(\delta^+(v)) = \boldsymbol{x}(\delta^-(v)) = 1 \qquad \forall v \in V, \tag{3.3}$$

$$x_a \geq 0 \qquad \forall a.$$

This relaxation is known as the Held-Karp relaxation [17] and its optimum value, which we denote by $\mathrm{OPT}_{\mathrm{HK}}$, can be computed in polynomial-time (either by the ellipsoid algorithm or by reformulating it as an LP with polynomially-bounded size). Observe that (3.3) implies that any feasible solution $\boldsymbol{x}$ to the Held-Karp relaxation satisfies

$$\boldsymbol{x}(\delta^+(U)) = \boldsymbol{x}(\delta^-(U)), \tag{3.4}$$

for any $U \subset V$.

Let $\boldsymbol{x}^*$ denote an optimum solution to this LP (3.1); thus $c(\boldsymbol{x}^*) = \mathrm{OPT}_{\mathrm{HK}}$. We can assume that $\boldsymbol{x}^*$ is an extreme point of the corresponding polytope. We first make this solution symmetric and slightly scale it down by setting

$$\boldsymbol{z}^*_{\{u,v\}} := \frac{n-1}{n}(x^*_{uv} + x^*_{vu}). \tag{3.5}$$

Let A denote the support of $\boldsymbol{x}^*$, i.e. $A = \{(u, v) : x^*_{uv} > 0\}$, and E the support of $\boldsymbol{z}^*$. For every edge $e = \{u, v\}$ of E, we can define its cost as $\min\{c(a) : a \in \{(u, v), (v, u)\} \cap A\}$; with the risk of overloading the notation, we denote this new cost of this edge e by $c(e)$. This implies that $c(\boldsymbol{z}^*) < c(\boldsymbol{x}^*)$.

The main purpose of the scaling factor in the definition (3.5) is to obtain a vector $\boldsymbol{z}^*$ which belongs to the *spanning tree polytope* P of the graph (V, E), i.e. $\boldsymbol{z}^*$ can be viewed as a convex combination of incidence vectors of spanning trees, see Lemma 3.1. In fact, $\boldsymbol{z}^*$ even belongs to the relative interior of P.

Lemma 3.1. *The vector $\boldsymbol{z}^*$ defined by (3.5) belongs to* relint(P)*, the relative interior of the spanning tree polytope P.*

Proof. From Edmonds' characterization of the base polytope of a matroid [10], it follows that the spanning tree

polytope P is defined by the following inequalities (see [24, Corollary 50.7c]):

$$P = \{z \in \mathbb{R}^E : z(E) = |V| - 1 \tag{3.6}$$

$$z(E(U)) \le |U| - 1, \qquad \forall U \subset V \tag{3.7}$$

$$z_e \ge 0 \qquad \forall e \in E.\} \tag{3.8}$$

The relative interior of P corresponds to those $z \in P$ satisfying all inequalities (3.7) and (3.8) *strictly*.

Clearly, z^* satisfies (3.6) since:

$$\begin{aligned} \forall v \in V,\ x^*(\delta^+(v)) = 1 \quad &\Rightarrow \quad x^*(A) = n = |V| \\ &\Rightarrow \quad z^*(E) = n - 1 = |V| - 1. \end{aligned}$$

Consider any set $U \subset V$. We have

$$\begin{aligned} \sum_{v \in U} x^*(\delta^+(v)) \quad &= \quad |U| = x^*(A(U)) + x^*(\delta^+(U)) \\ &\ge \quad x^*(A(U)) + 1. \end{aligned}$$

Since x^* satisfies (3.2) and (3.3), we have

$$z^*(E(U)) = \frac{n-1}{n} x^*(A(U)) < x^*(A(U)) \le |U| - 1,$$

showing that z^* satisfies (3.7) strictly. Since E is the support of z^*, (3.8) is also satisfied strictly by z^*. This shows that z^* is in the relative interior of P.

One implication of being in the relative interior of P is that z^* can be expressed as a convex combination of spanning trees such that the coefficient corresponding to *any* spanning tree is positive.

Regarding the size of the extreme point x^*, it is known that its support A has at most $3n - 4$ arcs (see [14, Theorem 15]). In addition, we know that it can be expressed as the unique solution of an invertible system with only $0 - 1$ coefficients, and therefore, we have that every entry x^*_a is rational with integral numerator and denominator bounded by $2^{O(n \log n)}$. In particular, $z^*_{\min} = \min_{e \in E} z^*_e > 2^{-O(n \log n)}$.

4 Maximum Entropy Sampling and Concentration Bounds

Our aim in this section is to round z^*, as a point in the relative interior of the spanning tree polytope, to a spanning tree. Suppose that we are looking for a distribution over spanning trees of G that preserves the marginal probabilities imposed by z^*, i.e. $\Pr_T[e \in T] = z^*_e$ for every edge $e \in E$. There are plenty such distributions. The approach we use is based on sampling from the distribution that maximizes the *entropy* among all marginal preserving distributions. Such a *maximum entropy rounding* scheme has been used in [2] for sampling a random matching in a bipartite graph with given marginal probabilities.

In Section 4.1, through a convex program, we formally define the maximum entropy distribution over the spanning trees with respect to the marginal probabilities given by z (an *arbitrary* point in the relative interior of the spanning tree polytope). From the optimality conditions for this convex program and its dual, we show that this distribution generates a λ-random spanning tree for some vector $\lambda \in \mathbb{R}^{|E|}$, where any tree T is output with probability proportional to $\prod_{e \in T} \lambda_e$.

Section 4.3 explains the main implication of such distributions. The events corresponding to the edges of G being included in a sampled λ-random tree are negatively correlated. This enables us to use Chernoff bounds on such events. We use these tail bounds to establish the *thinness* of a sampled tree. (Roughly speaking, a tree is said to be thin if the number of its edges in each cut is not much higher than its expected value; see Section 5 for a formal definition of thinness.)

It is possible to approximately find the γ_e's efficiently. In fact, we have a rather simple and iterative algorithm that, after a polynomial number of iterations, finds approximate $\tilde{\lambda}_e$'s with new marginal probabilities $\tilde{z}_e$, where for all edges e, $\tilde{z}_e \le (1 + \varepsilon) z_e$. We postpone the description of this algorithm and its analysis to Section 7. Instead, in Section 4.2 we show how to efficiently sample a tree from such a distribution given any vector λ.

4.1 Maximum Entropy Distribution.

Let $\mathcal{T}$ be the collection of all the spanning trees of $G = (V, E)$. The maximum entropy distribution $p^*(\cdot)$ with respect to given marginal probabilities z is the optimum solution of the following convex program CP:

$$\begin{aligned} \inf \quad & \sum_{T \in \mathcal{T}} p(T) \log p(T) \\ \text{s.t.} \quad & \sum_{T \ni e} p(T) = z_e \quad \forall e \in E, \\ & p(T) \ge 0 \qquad \forall T \in \mathcal{T}. \end{aligned} \tag{4.9}$$

This convex program is feasible whenever z belongs to the spanning tree polytope P defined on $G = (V, E)$. As the objective function is bounded and the feasible region is compact (closed and bounded), the inf is attained and there exists an optimum solution $p^*(\cdot)$. Furthermore, since the objective function is strictly convex, this maximum entropy distribution $p^*(\cdot)$ is unique. Let OPT_{CP} denote the optimum value of this convex program CP.

The value $p^*(T)$ determines the probability of sampling any tree T in the maximum entropy rounding scheme. Note that it is implicit in the constraints of this convex program that, for any feasible solution $p(.)$, we have $\sum_T p(T) = 1$ since

$$n - 1 = \sum_{e \in E} z_e = \sum_{e \in E} \sum_{T \ni e} p(T) = (n-1) \sum_T p(T).$$

We now want to show that, if we assume that the vector z is in the *relative interior* of the spanning tree polytope of (V, E) then $p^*(T) > 0$ for every $T \in \mathcal{T}$ and $p^*(T)$ admits a simple exponential formula. Note that the vector z^* obtained from the LP relaxation of the ATSP indeed satisfies this assumption.

For this purpose, we write the Lagrange dual to the convex program CP, see for example [22]. For every $e \in E$, we associate a Lagrange multiplier δ_e to the constraint corresponding to the marginal probability z_e, and define the Lagrange function by

$$L(p, \delta) = \sum_{T \in \mathcal{T}} p(T) \log p(T) - \sum_{e \in E} \delta_e \left(\sum_{T \ni e} p(T) - z_e \right).$$

This can also be written as:

$$L(p, \delta) = \sum_{e \in E} \delta_e z_e + \sum_{T \in \mathcal{T}} \left(p(T) \log p(T) - p(T) \sum_{e \in T} \delta_e \right).$$

The Lagrange dual to CP is now

$$\sup_{\delta} \inf_{p \geq 0} L(p, \delta). \tag{4.10}$$

The inner infimum in this dual is easy to solve. As the contributions of the $p(T)$'s are separable, we have that, for every $T \in \mathcal{T}$, $p(T)$ must minimize the convex function

$$p(T) \log p(T) - p(T)\delta(T),$$

where, as usual, $\delta(T) = \sum_{e \in T} \delta_e$. Taking derivatives, we derive that

$$0 = 1 + \log p(T) - \delta(T),$$

or

$$p(T) = e^{\delta(T) - 1}. \tag{4.11}$$

Thus,

$$\inf_{p \geq 0} L(p, \delta) = \sum_{e \in E} \delta_e z_e - \sum_{T \in \mathcal{T}} e^{\delta(T) - 1}.$$

Using the change of variables $\gamma_e = \delta_e - \frac{1}{n-1}$ for $e \in E$, the Lagrange dual (4.10) can therefore be rewritten as

$$\sup_{\gamma} \left[1 + \sum_{e \in E} z_e \gamma_e - \sum_{T \in \mathcal{T}} e^{\gamma(T)} \right]. \tag{4.12}$$

Our assumption that the vector z is in the relative interior of the spanning tree polytope is a Slater condition and, together with convexity, implies that the sup in (4.12) is attained by some vector γ^*, and the Lagrange dual value equals the optimum value OPT_{CP} of our convex program. Furthermore, we have that the (unique) primal optimum solution p^* and any dual optimum solution γ^* must satisfy

$$L(p, \gamma^*) \geq L(p^*, \gamma^*) \geq L(p^*, \gamma),$$

for any $p \geq 0$ and any γ, where we have implicitly redefined L due to our change of variables from δ to γ. Therefore, p^* is the unique minimizer of $L(p, \gamma^*)$ and from (4.11), we have that

$$p^*(T) = e^{\gamma^*(T)}. \tag{4.13}$$

Summarizing, the following theorem holds.

Theorem 4.1. *Given a vector z in the relative interior of the spanning tree polytope P on $G = (V, E)$, there exist γ_e^* for all $e \in E$ such that if we sample a spanning tree T of G according to $p^*(T) := e^{\gamma^*(T)}$ then $\Pr[e \in T] = z_e$ for every $e \in E$.*

It is worth noting that the requirement that z is in the relative interior of the spanning tree polytope (as opposed to being just in this polytope) is necessary (the fact that being in the spanning tree polytope was not sufficient had been observed before, see Exercise 4.19 in [21]). Let G be a triangle and z be the vector $(\frac{1}{2}, \frac{1}{2}, 1)$. In this case, z is in the polytope (but not in its relative interior) and there are no γ_e^*'s that would satisfy the statement of the theorem (however, one can get arbitrarily close to z_e for all $e \in E$).

In Section 7 we show how to efficiently find $\tilde{\gamma}$'s that approximately satisfy the conditions of Theorem 4.1. More formally, we prove the following theorem whose result we use in the rest of the paper. For our ATSP application, $z_{\min}$ is $2^{-O(n \log n)}$ and ε can be chosen to be 0.2.

Theorem 4.2. *Given z in the spanning tree polytope of $G = (V, E)$ and some $\varepsilon > 0$, values $\tilde{\gamma}_e$ for all $e \in E$ can be found, so that if we define the exponential family distribution*

$$\tilde{p}(T) := \frac{1}{P} \exp\left(\sum_{e \in T} \tilde{\gamma}_e\right)$$

for all $T \in \mathcal{T}$ where

$$P := \sum_{T \in \mathcal{T}} \exp\left(\sum_{e \in T} \tilde{\gamma}_e\right)$$

then, for every edge $e \in E$,

$$\tilde{z}_e := \sum_{T \in \mathcal{T} : T \ni e} \tilde{p}(T) \leq (1 + \varepsilon) z_e,$$

i.e. the marginals are approximately preserved. Furthermore, the running time is polynomial in $n = |V|$, $-\log z_{\min}$ and $1/\varepsilon$.

The distributions over trees considered in Theorem 4.2 are closely related to the notion of λ-random (spanning) trees. Given $\lambda_e \geq 0$ for $e \in E$, a *λ-random tree* T of G is a tree T chosen from the set of all spanning trees of G with probability proportional to $\prod_{e \in T} \lambda_e$. The notion of λ-random trees has been extensively studied (see e.g. Ch.4 of

[21]) - note that in case of all λ_e being equal, a λ-random tree is just a uniform spanning tree of G. Many of the results for uniform spanning trees carry over to λ-random spanning trees in a graph G since, for rational λ_e's, a λ-random spanning tree in G corresponds to a uniform spanning tree in a multigraph obtained from G by letting the multiplicity of edge e be proportional to λ_e.

Observe that a tree T sampled from an exponential family distribution $p(\cdot)$ as given in Theorem 4.2 is λ-random for $\lambda_e := e^{\gamma_e}$ for all $e \in E$. As a result, we can use the tools developed for λ-random trees to obtain an efficient sampling procedure, see Section 4.2, and to derive sharp concentration bounds for the distribution $p(\cdot)$, see Section 4.3.

4.2 Sampling a λ-Random Tree .

There is a host of results (see [15, 20, 9, 1, 5, 25, 19] and references therein) on obtaining polynomial-time algorithms for generating a uniform spanning tree, i.e. a λ-random tree for the case of all λ_e's being equal. Almost all of them can be easily modified to allow arbitrary λ; however, not all of them still guarantee a polynomial running time for general λ_e's. We use for example an iterative approach similar to [20].

The idea is to order the edges $e_1, \ldots, e_m$ of G arbitrarily and process them one by one, deciding probabilistically whether to add a given edge to the final tree or to discard it. More precisely, when we process the j-th edge e_j, we decide to add it to a final spanning tree T with probability p_j being the probability that e_j is in a λ-random tree *conditioned* on the decisions that were made for edges $e_1, \ldots, e_{j-1}$ in earlier iterations. Clearly, this procedure generates a λ-random tree, and its running time is polynomial as long as the computation of the probabilities p_j can be done in polynomial time.

To compute these probabilities efficiently we note that, by definition, $p_1 = z_{e_1}$. Now, if we choose to include e_1 in the tree then:

$$p_2 = \Pr[e_2 \in T | e_1 \in T] = \frac{\sum_{T' \ni e_1, e_2} \prod_{e \in T'} \lambda_e}{\sum_{T' \ni e_1} \prod_{e \in T'} \lambda_e} = \frac{\sum_{T' \ni e_1, e_2} \prod_{e \in T' \setminus e_1} \lambda_e}{\sum_{T' \ni e_1} \prod_{e \in T' \setminus e_1} \lambda_e}.$$

As one can see, the probability that $e_2 \in T$ conditioned on the event that $e_1 \in T$ is equal to the probability that e_2 is in a λ-random tree of a graph obtained from G by contracting the edge e_1. Similarly, if we choose to discard e_1, the probability p_2 is equal to the probability that e_2 is in a λ-random tree of a graph obtained from G by removing e_1. In general, p_j is equal to the probability that e_j is included in a λ-random tree of a graph obtained from G by contracting all edges that we have already decided to add to the tree, and deleting all edges that we have already decided to discard.

Therefore, the only thing we need in order to compute the p_j's is to be able to compute efficiently for a given graph G' and values of λ_e's, the probability $p_{G'}[\lambda, f]$ that some edge f is in a λ-random tree of G'. This is well-known. For this purpose, one can evaluate $\sum_{T \in \mathcal{T}} \prod_{e \in T} \lambda_e$ for both G' and $G'/\{f\}$ (in which edge f is contracted) using Kirchhoff's matrix tree theorem (see [4]). The matrix tree theorem states that $\sum_{T \in \mathcal{T}} \prod_{e \in T} \lambda_e$ for any graph G is equal to the absolute value of any cofactor of the weighted Laplacian L where

$$L_{i,j} = \begin{cases} -\lambda_e & e = (i,j) \in E \\ \sum_{e \in \delta(\{i\})} \lambda_e & i = j \\ 0 & \text{otherwise.} \end{cases}$$

An alternative approach for computing $p_{G'}[\lambda, f]$ is to use the fact (see e.g. Ch. 4 of [21]) that $p_{G'}[\lambda, f]$ is equal to λ_f times the effective resistance of f in G' treated as an electrical circuit with conductances of edges given by λ. The effective resistance can be expressed by an explicit linear-algebraic formula whose computation boils down to inverting a certain matrix that can be easily derived from the Laplacian of G' (see e.g. section 2.4 of [13] for details).

4.3 Negative Correlation and a Concentration Bound.

We derive now the following concentration bound. As discussed in the next section, this bound is instrumental in establishing the thinness of a sampled tree.

THEOREM 4.3. *For each edge e, let X_e be an indicator random variable associated with the event $[e \in T]$, where T is a λ-random tree. Also, for any subset C of the edges of G, define $X(C) = \sum_{e \in C} X_e$. Then we have*

$$\Pr[X(C) \geq (1+\delta)E[X(C)]] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}} \right)^{E[X(C)]}.$$

Usually, when we want to obtain such concentration bounds, we prove that the variables $\{X_e\}_e$ are independent and use the Chernoff bound. Unfortunately, in our case, the variables $\{X_e\}_e$ are not independent. However, it is well-known that they are *negatively correlated*, i.e. for any subset $F \subseteq E$, $\Pr[\forall_{e \in F} X_e = 1] \leq \prod_{e \in F} \Pr[X_e = 1]$, see e.g. ch. 4 of [21][1].

LEMMA 4.1. *The random variables $\{X_e\}_e$ are negatively correlated.*

Once we have established negative correlation between the X_e's, Theorem 4.3 follows directly from the result of Panconesi and Srinivasan [23] that the upper tail part of the

[1] Lyons and Peres prove this fact only in the case of T being a uniform spanning tree i.e. when all λ_e's are equal, but Section 4.1 of [21] contains a justification why this proof implies this property also in the case of arbitrary λ_e's. As mentioned before, for rational λ_e's, the main idea is to replace each edge e with $C\lambda_e$ edges (for an appropriate choice of C) and consider a uniform spanning tree in the corresponding multigraph. The irrational case follows from a limit argument.

Chernoff bound requires only negative correlation (or even a weaker notion, see [23]) and not the full independence of the random variables.

Since an earlier version of this paper, other ways of producing *negatively correlated* probability distributions on trees (or, more generally, matroid bases) satisfying some given marginals z have been proposed [7, 26]. Chekuri and Vondrak [7] use a randomized variant of pipage rounding while Zenklusen uses a randomized selection based on the basis exchange property of matroid bases. Both approaches can also be used in the framework developed in this paper.

5 The Thinness Property

In this section, we focus on the exponential family distribution $\tilde{p}(\cdot)$ that we obtain by applying the algorithm of Theorem 4.2 to z^*. We show that the tree sampled from the distribution $\tilde{p}(\cdot)$ is almost surely "thin". First we define the thinness property.

Definition 5.1. *We say that a tree T is α-thin if for each set $U \subset V$,*

$$|T \cap \delta(U)| \leq \alpha z^*(\delta(U)).$$

Also we say that T is (α, s)-thin if it is α-thin and moreover,

$$c(T) \leq sOPT_{HK}.$$

We first prove that if we focus on a particular cut then the "α-thinness" property holds for it with overwhelming probability where $\alpha \sim \frac{\log n}{\log\log n}$.

Lemma 5.1. *If T is a spanning tree sampled from distribution $\tilde{p}(\cdot)$ for $\varepsilon = 0.2$ in a graph G with $n \geq 5$ vertices then for any set $U \subset V$,*

$$\Pr[|T \cap \delta(U)| > \beta z^*(\delta(U))] \leq n^{-2.5 z^*(\delta(U))},$$

where $\beta = 4\log n/\log\log n$.

Proof. Note that by definition, for all edges $e \in E$, $\tilde{z}_e \leq (1+\varepsilon)z^*_e$, where $\varepsilon = 0.2$ is the desired accuracy of approximation of z^* by $\tilde{z}$ as in Theorem 4.2. Hence,

$$\mathrm{E}[|T \cap \delta(U)|] = \tilde{z}(\delta(U)) \leq (1+\varepsilon) z^*(\delta(U)).$$

Applying Theorem 4.3 with

$$1 + \delta = \beta \frac{z^*(\delta(U))}{\tilde{z}(\delta(U))} \geq \frac{\beta}{1+\varepsilon},$$

we derive that $\Pr[|T \cap \delta(U)| > \beta z^*(\delta(U))]$ can be bounded from above by

$$\begin{aligned}
&\Pr[|T \cap \delta(U)| > (1+\delta)\mathrm{E}[|T \cap \delta(U)|]] \\
&\leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^{\tilde{z}(\delta(U))} \\
&\leq \left(\frac{e}{1+\delta}\right)^{(1+\delta)\tilde{z}(\delta(U))} \\
&= \left(\frac{e}{1+\delta}\right)^{\beta z^*(\delta(U))} \\
&\leq \left[\left(\frac{e(1+\varepsilon)}{\beta}\right)^{\beta}\right]^{z^*(\delta(U))} \\
&\leq n^{-4(1-1/e) z^*(\delta(U))},
\end{aligned}$$

where, in the last inequality, we have used that

$$\begin{aligned}
\log\left[\left(\frac{e(1+\varepsilon)}{\beta}\right)^{\beta}\right] &= 4\frac{\log n}{\log\log n}[1 + \log(1+\varepsilon) - \log(4) \\
&\quad - \log\log n + \log\log\log n] \\
&\leq -4\log n\left(1 - \frac{\log\log\log n}{\log\log n}\right) \\
&\leq -4\left(1 - \frac{1}{e}\right)\log n \leq -2.5\log n,
\end{aligned}$$

since $e(1+\varepsilon) < 4$ and $\frac{\log\log\log n}{\log\log n} \leq \frac{1}{e}$ for all $n \geq 5$ (even for $n \geq 3$).

We are ready to prove the main theorem of this section.

Theorem 5.1. *Let $n \geq 5$ and $\varepsilon = 0.2$. Let $T_1, \ldots, T_{\lceil 2\log n\rceil}$ be $\lceil 2\log n\rceil$ independent samples from a distribution $\tilde{p}(\cdot)$ as given in Theorem 4.2. Let T^* be the tree among these samples that minimizes $c(T_j)$. Then, T^* is $(4\log n/\log\log n, 2)$-thin with high probability.*

Here, high probability means probability at least $1 - 1/n$, but probability $1 - 1/n^k$ can be achieved by sampling $2k\log n$ trees.

Proof. We start by showing that for any $1 \leq j \leq \lceil 2\log n\rceil$, T_j is β-thin with high probability for $\beta = 4\log n/\log\log n$. From Lemma 5.1 we know that the probability of some particular cut $\delta(U)$ violating the β-thinness of T_j is at most $n^{-2.5 z^*(\delta(U))}$. Now, we use a result of Karger [18] that shows that there are at most n^{2l} cuts of size at most l times the minimum cut value for any half-integer $l \geq 1$. Since, by the definitions of the Held-Karp relaxation and of z^*, we know that $z^*(\delta(U)) \geq 2(1-1/n)$, it means there is at most n^l cuts $\delta(U)$ with $z^*(\delta(U)) \leq l(1-1/n)$ for any integer $l \geq 2$. Therefore, by applying the union bound (and $n \geq 5$), we derive that the probability that there exists some cut $\delta(U)$ with $|T_j \cap \delta(U)| > \beta z^*(\delta(U))$ is at most

$$\sum_{i=3}^{\infty} n^i n^{-2.5(i-1)(1-1/n)},$$

where each term is an upper bound on the probability that there exists a violating cut of size within $[(i-1)(1-1/n), i(1-1/n)]$. Fo $n \geq 5$, this simplifies to:

$$\sum_{i=3}^{\infty} n^i n^{-2.5(i-1)(1-1/n)} \leq \sum_{i=3}^{\infty} n^{-i+2} = \frac{1}{n-1},$$

Thus, indeed, T_j is a β-thin spanning tree with high probability.

Now, the expected cost of T_j is

$$\mathrm{E}[c(T_j)] \leq \sum_{e \in E} \tilde{z}_e \leq (1+\varepsilon)\frac{n-1}{n} \sum_{a \in A} x^*_a \leq (1+\varepsilon)\mathrm{OPT}_{\mathrm{HK}}.$$

So, by Markov inequality we have that for any j, the probability that $c(T_j) > 2\mathrm{OPT}_{\mathrm{HK}}$ is at most $(1+\varepsilon)/2$. Thus, with probability at most $(\frac{1+\varepsilon}{2})^{2\log n} < \frac{1}{n}$ for $\varepsilon = 0.2$, we have $c(T^*) > 2\mathrm{OPT}_{\mathrm{HK}}$. This concludes the proof of the theorem.

6 Transforming a Thin Spanning Tree into an Eulerian Walk

As the final step of the algorithm, we show how one can find an Eulerian walk with small cost using a thin tree. After finding such a walk, one can use the metric property to convert this walk into an Hamiltonian cycle of no greater cost (by shortcutting). In particular, the following theorem justifies the definition of thin spanning trees.

Theorem 6.1. *Assume that we are given an (α, s)-thin spanning tree T^* with respect to the LP relaxation $\boldsymbol{x}^*$. Then we can find a Hamiltonian cycle of cost no more than $(2\alpha + s)c(\boldsymbol{x}^*) = (2\alpha + s)OPT_{HK}$ in polynomial time.*

Before proceeding to the proof of Theorem 6.1, we recall some basic network flow results related to circulations. A function $f : A \to \mathbb{R}$ is called a *circulation* if $f(\delta^+(v)) = f(\delta^-(v))$ for each vertex $v \in V$. Hoffman's circulation theorem [24, Theorem 11.2] gives a necessary and sufficient condition for the existence of a circulation subject to lower and upper capacities on arcs.

Theorem 6.2. (Hoffman's circulation theorem) *Given lower and upper capacities $l, u : A \to \mathbb{R}$, there exists a circulation f satisfying $l(a) \leq f(a) \leq u(a)$ for all $a \in A$ if and only if*

1. *$l(a) \leq u(a)$ for all $a \in A$ and*
2. *for all subsets $U \subset V$, we have $l(\delta^-(U)) \leq u(\delta^+(U))$.*

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

Proof. **[Theorem 6.1]** We first orient each edge $\{u, v\}$ of T^* to $\arg\min\{c(a) : a \in \{(u,v),(v,u)\} \cap A\}$, and denote the resulting directed tree by $\vec{T}^*$. Observe that by definition of our undirected cost function, we have $c(\vec{T}^*) = c(T^*)$. We then find a minimum cost augmentation of $\vec{T}^*$ into an Eulerian directed graph; this can be formulated as a minimum cost circulation problem with integral lower capacities (and no or infinite upper capacities). Indeed, set

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

and consider the minimum cost circulation problem

$$\min\{c(f) : f \text{ is a circulation and } f(a) \geq l(a)\ \forall a \in A\}.$$

An optimum circulation f^* can be computed in polynomial time and can be assumed to be integral, see e.g. [24, Corollary 12.2a]. This integral circulation f^* corresponds to a directed (multi)graph H which contains $\vec{T}^*$. Every vertex in H has an indegree equal to its outdegree. Therefore, every cut has the same number of arcs in both directions. As H is weakly connected (as it contains $\vec{T}^*$), it is strongly connected and thus, H is an Eulerian directed multigraph. We can extract an Eulerian walk of H and shortcut it to obtain our Hamiltonian cycle of cost at most $c(f^*)$ since the costs satisfy the triangle inequality.

To complete the proof of Theorem 6.1, it remains to show that $c(f^*) \leq (2\alpha + s)c(\boldsymbol{x}^*)$. For this purpose, we define

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

We claim that there exists a circulation g satisfying $l(a) \leq g(a) \leq u(a)$ for every $a \in A$. To prove this claim, we use Hoffman's circulation theorem 6.2. Indeed, by construction, $l(a) \leq u(a)$ for every $a \in A$; furthermore, Lemma 6.1 below shows that, for every $U \subset V$, we have $l(\delta^-(U)) \leq u(\delta^+(U))$. Thus the existence of the circulation g is established. Furthermore,

$$c(f^*) \leq c(g) \leq c(u) = c(\vec{T}^*) + 2\alpha c(\boldsymbol{x}^*) \leq (2\alpha + s)c(\boldsymbol{x}^*),$$

establishing the bound on the cost of f^*. This completes the proof of Theorem 6.1.

Lemma 6.1. *For the capacities l and u as constructed in the proof of Theorem 6.1, the following holds for any subset $U \subset V$:*

$$l(\delta^-(U)) \leq u(\delta^+(U)).$$

Proof. Irrespective of the orientation of T^* into $\vec{T}^*$, the number of arcs of $\vec{T}^*$ in $\delta^-(U)$ is at most $\alpha \boldsymbol{z}^*(\delta(U))$ by definition of α-thinness. Thus

$$l(\delta^-(U)) \leq \alpha \boldsymbol{z}^*(\delta(U)) < 2\alpha \boldsymbol{x}^*(\delta^-(U)),$$

due to (3.4) and (3.5). On the other hand, we have

$$u(\delta^+(U)) \geq 2\alpha \boldsymbol{x}^*(\delta^+(U)) = 2\alpha \boldsymbol{x}^*(\delta^-(U)) \geq l(\delta^-(U)),$$

where we have used the fact that $\boldsymbol{x}^*$ itself is a circulation (see (3.4)). The lemma follows.

THEOREM 6.3. *For a suitable choice of parameters, the algorithm given in Figure 1 finds a* $(2 + 8 \log n / \log \log n)$*-approximate solution to the Asymmetric Traveling Salesman Problem with high probability and in time polynomial in the size of the input.*

Proof. The algorithm start by finding an optimal extreme-point solution $\boldsymbol{x}^*$ to the Held-Karp LP relaxation of ATSP of value $\mathrm{OPT}_{\mathrm{HK}}$. Next, using the algorithm of Theorem 4.2 on $\boldsymbol{z}^*$ (which is defined by (3.5)) with $\varepsilon = 0.2$, we obtain $\tilde{\gamma}_e$'s that define the exponential family distribution $\tilde{p}(T) := e^{\sum_{e \in T} \tilde{\gamma}_e}$. Since $\boldsymbol{x}^*$ was an extreme point, we know that $z^*_{\min} \geq e^{-O(n \log n)}$; thus, the algorithm of Theorem 4.2 indeed runs in polynomial time.

Next, we use the polynomial time sampling procedure described in section 4.2 to sample $\Theta(\log n)$ trees T_j from the distribution $\tilde{p}(\cdot)$, and take T^* to be the one among them that minimizes the cost $c(T_j)$. By Theorem 5.1, we know that T^* is $(\beta, 2)$-thin with high probability.

Now, we use Theorem 6.1 to obtain, in polynomial time, a $(2+8 \log n / \log \log n)$-approximation of our ATSP instance.

The proof also shows that the integrality gap of the Held-Karp relaxation for the Asymmetric TSP is bounded above by $2 + 8 \log n / \log \log n$. The best known lower bound on the integrality gap is only 2, as shown in [6]. Closing this gap is a challenging open question, and this possibly could be answered using thinner spanning trees.

COROLLARY 6.1. *If there always exists a* (C_1, C_2)*-thin spanning tree where* C_1 *and* C_2 *are constants, the integrality gap of the ATSP Held-Karp linear programming relaxation is a constant.*

7 A Combinatorial Algorithm for Approximately Solving CP

In this section, we provide a combinatorial algorithm to efficiently find $\tilde{\gamma}_e$'s that approximately preserve the marginal probabilities given by $\boldsymbol{z}$ and therefore prove Theorem 4.2.

Given a vector γ, for each edge e, define $q_e(\gamma) := \frac{\sum_{T \ni e} \exp(\gamma(T))}{\sum_T \exp(\gamma(T))}$, where $\gamma(T) = \sum_{f \in T} \gamma_f$. For notational convenience, we have dropped the fact that $T \in \mathcal{T}$ in these summations; this shouldn't lead to any confusion. Restated, $q_e(\gamma)$ is the probability that edge e will be included in a spanning tree T that is chosen with probability proportional to $\exp(\gamma(T))$.

We compute $\tilde{\gamma}$ using the following simple algorithm. Start with all γ_e equal, and as long as the marginal $q_e(\gamma)$ for some edge e is more than $(1+\varepsilon)z_e$, we decrease appropriately γ_e in order to decrease $q_e(\gamma)$ to $(1 + \varepsilon/2)z_e$. More formally, here is a description of the algorithm.

1. Set $\gamma = \vec{0}$.
2. While there exists an edge e with $q_e(\gamma) > (1 + \varepsilon)z_e$:
   - Compute δ such that if we define γ' as $\gamma'_e = \gamma_e - \delta$, and $\gamma'_f = \gamma_f$ for all $f \in E \setminus \{e\}$, then $q_e(\gamma') = (1 + \varepsilon/2)z_e$
   - Set $\gamma \leftarrow \gamma'$
3. Output $\tilde{\gamma} := \gamma$.

Clearly, *if* the above procedure terminates then the resulting $\tilde{\gamma}$ satisfies the requirement of Theorem 4.2. Therefore, what we need to show is that this algorithm terminates in time polynomial in n, $-\log z_{\min}$ and $1/\varepsilon$, and that each iteration can be implemented in polynomial time.

We start by bounding the number of iterations - we will show that it is $O(\frac{1}{\varepsilon}|E|^2[|V| \log(|V|) - \log(\varepsilon z_{\min})])$. In the next lemma, we derive an equation for δ, and prove that for $f \neq e$ the probabilities $q_f(\cdot)$ do not decrease as a result of decreasing γ_e.

LEMMA 7.1. *If for some* $\delta \geq 0$ *and an edge* e*, we define* γ' *by* $\gamma'_e = \gamma_e - \delta$ *and* $\gamma'_f = \gamma_f$ *for all* $f \neq e$*, then*

1. *for all* $f \in E \setminus \{e\}$, $q_f(\gamma') \geq q_f(\gamma)$,
2. $q_e(\gamma')$ *satisfies* $\frac{1}{q_e(\gamma')} - 1 = e^{\delta}\left(\frac{1}{q_e(\gamma)} - 1\right)$.

In particular, in the main loop of the algorithm, since $q_e(\gamma) > (1 + \varepsilon)z_e$ and we want $q_e(\gamma') = (1 + \varepsilon/2)z_e$, we get $\delta = \log \frac{q_e(\gamma)(1-(1+\varepsilon/2)z_e)}{(1-q_e(\gamma))(1+\varepsilon/2)z_e} > \log \frac{(1+\varepsilon)}{(1+\varepsilon/2)} > \frac{\varepsilon}{4}$ for $\varepsilon \leq 1$ (for larger values of ε, we can simply decrease ε to 1).

Proof. Let us consider some $f \in E \setminus \{e\}$. We have

$$\begin{aligned} q_f(\gamma') &= \frac{\sum_{T \in \mathcal{T}: f \in T} \exp(\gamma'(T))}{\sum_{T \in \mathcal{T}} \exp(\gamma'(T))} \\ &= \frac{\sum_{T: e \in T, f \in T} e^{\gamma'(T)} + \sum_{T: e \notin T, f \in T} e^{\gamma'(T)}}{\sum_{T \ni e} e^{\gamma'(T)} + \sum_{T: e \notin T} e^{\gamma'(T)}} \\ &= \frac{e^{-\delta} \sum_{T: e \in T, f \in T} e^{\gamma(T)} + \sum_{T: e \notin T, f \in T} e^{\gamma(T)}}{e^{-\delta} \sum_{T \ni e} e^{\gamma(T)} + \sum_{T: e \notin T} e^{\gamma(T)}} \\ &= \frac{e^{-\delta} a + b}{e^{-\delta} c + d} \end{aligned}$$

with a, b, c, d appropriately defined. The same expression holds for $q_f(\gamma)$ with the $e^{-\delta}$ factors removed. But, for general $a, b, c, d \geq 0$, if $\frac{a}{c} \leq \frac{a+b}{c+d}$ then $\frac{xa+b}{xc+d} \geq \frac{a+b}{c+d}$ for $x \leq 1$. Since

$$\frac{a}{c} = \frac{\sum_{T \in \mathcal{T}: e \in T, f \in T} e^{\gamma(T)}}{\sum_{T \in \mathcal{T}: e \in T} e^{\gamma(T)}} \leq q_f(\gamma) = \frac{a+b}{c+d}$$

by negative correlation (since a/c represents the conditional probability that f is present given that e is present), we get that $q_f(\gamma') \geq q_f(\gamma)$ for $\delta \geq 0$.

Now, we proceed to deriving the equation for δ. By definition of $q_e(\gamma)$, we have

$$\frac{1}{q_e(\gamma)} - 1 = \frac{\sum_{T: e \notin T} e^{\gamma(T)}}{\sum_{T \ni e} e^{\gamma(T)}}.$$

Hence,

$$\begin{aligned}\frac{1}{q_e(\gamma')} - 1 &= \frac{\sum_{T: e\notin T} e^{\gamma'(T)}}{\sum_{T\ni e} e^{\gamma'(T)}} \\ &= \frac{\sum_{T: e\notin T} e^{\gamma(T)}}{e^{-\delta}\sum_{T\ni e} e^{\gamma(T)}} \\ &= e^{\delta}\left(\frac{1}{q_e(\gamma)} - 1\right).\end{aligned}$$

Before bounding the number of iterations, we collect some basic results regarding spanning trees which we need for the proof of the number of iterations.

LEMMA 7.2. *Let $G = (V, E)$ be a graph with weights γ_e for $e \in E$. Let $Q \subset E$ be such that for all $f \in Q$, $e \in E \setminus Q$, we have $\gamma_f > \gamma_e + \Delta$ for some $\Delta \geq 0$. Let r be the size of a maximum spanning forest of Q. Then*

1. *For any $T \in \mathcal{T}$, we have $|T \cap Q| \leq r$.*

   Define $\mathcal{T}_= := \{T \in \mathcal{T} : |T \cap Q| = r\}$ and $\mathcal{T}_< := \{T \in \mathcal{T} : |T \cap Q| < r\}$.

2. *Any spanning tree $T \in \mathcal{T}_=$ can be generated by taking the union of any spanning forest F (of cardinality r) of the graph (V, Q) and a spanning tree (of cardinality $n-r-1$) of the graph G/Q in which the edges of Q have been contracted.*

3. *Let $T_{\max}$ be a maximum spanning tree of G with respect to the weights $\gamma(\cdot)$, i.e. $T_{\max} = \arg\max_{T\in\mathcal{T}} \gamma(T)$. Then, for any $T \in \mathcal{T}_<$, we have $\gamma(T) < \gamma(T_{\max}) - \Delta$.*

Proof. These properties easily follow from the matroidal properties of spanning trees. To prove 3., consider any $T \in \mathcal{T}_<$. Since $|T \cap Q| < r$, there exists an edge $f \in (T_{\max} \cap Q) \setminus T$ such that $(T \cap Q) \cup \{f\}$ is a forest of G. Therefore, the unique circuit in $T \cup \{f\}$ contains an edge $e \notin Q$. Thus $T' = T \cup \{f\} \setminus \{e\}$ is a spanning tree. Our assumption on Q implies that

$$\gamma(T_{\max}) \geq \gamma(T') = \gamma(T) - \gamma_e + \gamma_f > \gamma(T) + \Delta,$$

which yields the desired inequality.

We proceed to bounding the number of iterations.

LEMMA 7.3. *The algorithm executes at most $O(\frac{1}{\varepsilon}|E|^2[|V|\log(|V|) - \log(\varepsilon z_{\min})])$ iterations of the main loop.*

Proof. Let $n = |V|$ and $m = |E|$. Assume for the sake of contradiction that the algorithm executes more than

$$\tau := \frac{4}{\varepsilon}m^2[n\log n - \log(\varepsilon z_{\min})]$$

iterations. Let γ be the vector of γ_e's computed at such an iteration. For brevity, let us define $q_e := q_e(\gamma)$ for all edges e.

We prove first that there exists some $e^* \in E$ such that $\gamma_{e^*} < -\frac{\varepsilon\tau}{4m}$. Indeed, there are m edges, and by Lemma 7.1 we know that in each iteration we decrease γ_e of one of these edges by at least $\varepsilon/4$. Thus, we know that, after more than τ iterations, there exists e^* for which γ_{e^*} is as desired.

Note that we never decrease γ_e for edges e with $q_e(\cdot)$ smaller than $(1+\varepsilon)z_e$, and Lemma 7.1 shows that reducing γ_f of edge $f \neq e$ can only increase $q_e(\cdot)$. Therefore, we know that all the edges with γ_e being negative must satisfy $q_e \geq (1+\varepsilon/2)z_e$. In other words, all edges e such that $q_e < (1+\varepsilon/2)z_e$ satisfy $\gamma_e = 0$. Finally, by a simple averaging argument, we know that $\sum_e q_e = n - 1 < (1+\varepsilon/2)(n-1) = (1+\varepsilon/2)\sum_e z_e$. Hence, there exists at least one edge f^* with $q_{f^*} < (1+\varepsilon/2)z_{f^*}$ and thus having $\gamma_{f^*} = 0$.

We proceed now to exhibiting a set Q such that:

(*I*)**:** $\emptyset \neq Q \subset E$, and

(*II*)**:** for all $e \in E \setminus Q$ and $f \in Q$, $\gamma_e + \frac{\varepsilon\tau}{4m^2} < \gamma_f$.

We construct Q as follows. We set threshold values $\Gamma_i = -\frac{\varepsilon\tau i}{4m^2}$, for $i \geq 0$, and define $Q_i = \{e \in E \,|\gamma_e \geq \Gamma_i\}$. Let $Q = Q_j$ where j is the first index such that $Q_j = Q_{j+1}$. Clearly, by construction of Q, property (*II*) is satisfied. Also, Q is non-empty since $f^* \in Q_0 \subseteq Q_j = Q$. Finally, by the pigeonhole principle, since we have m different edges, we know that $j < m$. Thus, for each $e \in Q$ we have $\gamma_e > \Gamma_m = -\frac{\varepsilon\tau}{4m}$. This means that $e^* \notin Q$ and thus Q has property (*I*).

Observe that Q satisfies the hypothesis of Lemma 7.2 with $\Delta = \frac{\varepsilon\tau}{4m^2}$. Thus, for any $T \in \mathcal{T}_<$, we have

$$\gamma(T_{\max}) > \gamma(T) + \frac{\varepsilon\tau}{4m^2}, \tag{7.14}$$

where $T_{\max}$ and r are as defined in Lemma 7.2.

Let $\widehat{G}$ be the graph G/Q obtained by contracting all the edges in Q. So, $\widehat{G}$ consists only of edges not in Q (some of them can be self-loops). Let $\hat{\mathcal{T}}$ be the set of all spanning trees of $\widehat{G}$, and for any given edge $e \notin Q$, let $\hat{q}_e := \frac{\sum_{\hat{T}\in\hat{\mathcal{T}}, \hat{T}\ni e}\exp(\gamma(\hat{T}))}{\sum_{\hat{T}\in\hat{\mathcal{T}}}\exp(\gamma(\hat{T}))}$ be the probability that edge e is included in a random spanning tree $\hat{T}$ of $\widehat{G}$, where each tree $\hat{T}$ is chosen with probability proportional to $e^{\gamma(\hat{T})}$. Since spanning trees of $\widehat{G}$ have $n-r-1$ edges, we have

$$\sum_{e\in E\setminus Q} \hat{q}_e = n - r - 1. \tag{7.15}$$

On the other hand, since z satisfies $z(E) = n-1$ and $z(Q) \leq r$ (by definition of r, see Lemma 7.2, part 1.), we have that $z(E \setminus Q) \geq n - r - 1$. Therefore, (7.15) implies that there must exist $\hat{e} \notin Q$ such that $\hat{q}_{\hat{e}} \leq z_{\hat{e}}$.

Our final step is to show that for any $e \notin Q$, $q_e < \hat{q}_e + \frac{\varepsilon z_{\min}}{2}$. Note that once we establish this, we know that

$q_{\hat{e}} < \hat{q}_{\hat{e}} + \frac{\varepsilon z_{\min}}{2} \leq (1 + \frac{\varepsilon}{2}) z_{\hat{e}}$, and thus it must be the case that $\gamma_{\hat{e}} = 0$. But this contradicts the fact that $\hat{e} \notin Q$, as by construction all e with $\gamma_e = 0$ must be in Q. Thus, we obtain a contradiction that concludes the proof of the Lemma.

It remains to prove that for any $e \notin Q$, $q_e < \hat{q}_e + \frac{\varepsilon z_{\min}}{2}$. We have that

$$\begin{aligned} q_e &= \frac{\sum_{T \in \mathcal{T}: e \in T} e^{\gamma(T)}}{\sum_{T \in \mathcal{T}} e^{\gamma(T)}} \\ &= \frac{\sum_{T \in \mathcal{T}_=: e \in T} e^{\gamma(T)} + \sum_{T \in \mathcal{T}_<: e \in T} e^{\gamma(T)}}{\sum_{T \in \mathcal{T}} e^{\gamma(T)}} \\ &\leq \frac{\sum_{T \in \mathcal{T}_=: e \in T} e^{\gamma(T)}}{\sum_{T \in \mathcal{T}_=} e^{\gamma(T)}} + \frac{\sum_{T \in \mathcal{T}_<: e \in T} e^{\gamma(T)}}{\sum_{T \in \mathcal{T}} e^{\gamma(T)}} \\ &\leq \frac{\sum_{T \in \mathcal{T}_=: e \in T} e^{\gamma(T)}}{\sum_{T \in \mathcal{T}_=} e^{\gamma(T)}} + \sum_{T \in \mathcal{T}_<: e \in T} \frac{e^{\gamma(T)}}{e^{\gamma(T_{\max})}}, \end{aligned} \tag{7.16}$$

the first inequality following from replacing $\mathcal{T}$ with $\mathcal{T}_=$ in the first denominator, and the second inequality following from considering only one term in the second denominator. Using (7.14) and the fact that the number of spanning trees is at most n^{n-2}, the second term is bounded by:

$$\begin{aligned} \sum_{T \in \mathcal{T}_<: e \in T} \frac{e^{\gamma(T)}}{e^{\gamma(T_{\max})}} &\leq n^{n-2} e^{-\varepsilon\tau/4m^2} \\ &< \frac{1}{2} n^n e^{-\varepsilon\tau/4m^2} \\ &= \frac{\varepsilon z_{\min}}{2}, \end{aligned} \tag{7.17}$$

by definition of τ. To handle the first term of (7.16), we can use part 2. of Lemma 7.2 and factorize:

$$\sum_{T \in \mathcal{T}_=} e^{\gamma(T)} = \left(\sum_{\hat{T} \in \hat{\mathcal{T}}} e^{\gamma(\hat{T})} \right) \left(\sum_{T' \in \mathcal{F}} e^{\gamma(T')} \right),$$

where $\mathcal{F}$ is the set of all spanning forests of (V, Q). Similarly, we can write

$$\sum_{T \in \mathcal{T}_=, T \ni e} e^{\gamma(T)} = \left(\sum_{\hat{T} \in \hat{\mathcal{T}}, \hat{T} \ni e} e^{\gamma(\hat{T})} \right) \left(\sum_{T' \in \mathcal{F}} e^{\gamma(T')} \right).$$

As a result, we have that the first term of (7.16) reduces to:

$$\frac{\left(\sum_{\hat{T} \in \hat{\mathcal{T}}} e^{\gamma(\hat{T})} \right) \left(\sum_{T' \in \mathcal{F}} e^{\gamma(T')} \right)}{\left(\sum_{\hat{T} \in \hat{\mathcal{T}}, \hat{T} \ni e} e^{\gamma(\hat{T})} \right) \left(\sum_{T' \in \mathcal{F}} e^{\gamma(T')} \right)} = \frac{\sum_{\hat{T} \in \hat{\mathcal{T}}} e^{\gamma(\hat{T})}}{\sum_{\hat{T} \in \hat{\mathcal{T}}, \hat{T} \ni e} e^{\gamma(\hat{T})}} = \hat{q}_e.$$

Together with (7.16) and (7.17), this gives

$$q_e \leq \hat{q}_e + \frac{\varepsilon z_{\min}}{2},$$

which completes the proof.

To complete the analysis of the algorithm, we need to argue that each iteration can be implemented in polynomial time. First, for any given vector γ, we can compute efficiently the sums $\sum_T \exp(\gamma(T))$ and $\sum_{T \ni e} \exp(\gamma(T))$ for any edge e - this will enable us to compute all $q_e(\gamma)$'s. This can be done using Kirchhoff's matrix tree theorem (see [4]), as discussed in Section 4.2 (with $\lambda_e = e^{\gamma_e}$). Observe that we can bound all entries of the weighted Laplacian matrix in terms of the input size since the proof of Lemma 7.3 actually shows that $-\frac{\varepsilon\tau}{4|E|} \leq \gamma_e \leq 0$ for all $e \in E$ and any iteration of the algorithm. Therefore, we can compute these cofactors efficiently, in time polynomial in n, $-\log z_{\min}$ and $1/\varepsilon$. Finally, δ can be computed efficiently from Lemma 7.1.

Acknowledgments. We would like to thank the reviewers for many insightful comments.

References

[1] D. J. Aldous. A random walk construction of uniform spanning trees and uniform labelled trees. *SIAM Journal on Discrete Mathematics*, 3(4):450–465, 1990.

[2] A. Asadpour and A. Saberi. Maximum entropy selection: a randomized rounding method. *submitted*.

[3] M. Bläser. A new approximation algorithm for the asymmetric TSP with traingle inequality. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 638–645, 2002.

[4] B. Bollobas. *Modern Graph Theory*. Springer, 2002.

[5] A. Broder. Generating random spanning trees. In *30th Annual Symposium on Foundations of Computer Science*, pages 442–447, 1989.

[6] M. Charikar, M. Goemans, and H. Karloff. On the integrality ratio for the asymmetric traveling salesman problem. *Mathematics of Operations Research*, 31:245–252, 2006.

[7] C. Chekuri and J. Vondrak. Randomized pipage rounding for matroid polytopes and applications, 2009. http://arxiv.org/abs/0909.4348v1.

[8] N. Christofides. Worst case analysis of a new heuristic for the traveling salesman problem. Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA, 1976.

[9] C. J. Colbourn, W. J. Myrvold, and E. Neufeld. Two algorithms for unranking arborescences. *Journal of Algorithms*, 20(2):268–281, 1996.

[10] J. Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, pages 127–136, 1971.

[11] U. Feige and M. Singh. Improved approximation ratios for traveling salesman tours and paths in directed graphs. In *10th International Workshop, APPROX*, pages 104–118, 2007.

[12] A. M. Frieze, G. Galbiati, and F. Maffioli. On the worst-case performance of some algorithms for the asymmetric traveling salesman problem. *Networks*, 12:23–39, 1982.

[13] A. Ghosh, S. Boyd, and A. Saberi. Minimizing effective resistance of a graph. *SIAM Rev.*, 50(1):37–66, 2008.

[14] M. X. Goemans. Minimum bounded degree spanning trees.

In *47th Annual Symposium on Foundations of Computer Science*, pages 273–282, 2006.

[15] A. Guenoche. Random spanning tree. *Journal of Algorithms*, 4:214–220, 1983.

[16] N. S. H. Kaplan, M. Lewenstein and M. Sviridenko. Approximation algorithms for asymmetric TSP by decomposing directed regular multigraphs. *J. ACM*, pages 602–626, 2005.

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

[18] D. R. Karger. Global min-cuts in RNC, and other ramifications of a simple min-cut algorithm. In *4th annual ACM-SIAM Symposium on Discrete algorithms*, pages 21–30, Philadelphia, PA, USA, 1993. Society for Industrial and Applied Mathematics.

[19] J. Kelner and A. Madry. Faster generation of random spanning trees. In *50th Annual Symposium on Foundations of Computer Science*, 2009.

[20] V. G. Kulkarni. Generating random combinatorial objects. *Journal of Algorithms*, 11:185–207, 1990.

[21] R. Lyons and Y. Peres. *Probability on Trees and Networks*. 2009. In preparation. Current version available at http://mypage.iu.edu/~rdlyons/.

[22] A. Nemirovski. Lectures on modern convex optimization, 2005.

[23] A. Panconesi and A. Srinivasan. Randomized distributed edge coloring via an extension of the Chernoff-Hoeffding bounds. *SIAM J. Comput.*, 26:350–368, 1997.

[24] A. Schrijver. *Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 2003.

[25] D. B. Wilson. Generating random spanning trees more quickly than the cover time. In *28th Annual ACM Symposium on the Theory of Computing*, pages 296–303. ACM, 1996.

[26] R. Zenklusen, July 2009. Personal communication.