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 present a randomized $O(\log n/ \log\log n)$-approximation algorithm for the asymmetric traveling salesman problem (ATSP). This provides the first asymptotic improvement over the long-standing $\Theta(\log n)$ approximation bound stemming from the work of Frieze et al. [17].

The key ingredient of our approach is a new connection between the approximability of the ATSP and the notion of so-called thin trees. To exploit this connection, we employ maximum entropy rounding – a novel method of randomized rounding of LP relaxations of optimization problems. We believe that this method might be of independent interest.

1 Introduction

Traveling salesman problem is one of the most celebrated and intensively studied problems in combinatorial optimization [30, 2, 13]. It has found applications in logistics, planning, manufacturing and testing microchips [31], as well as DNA sequencing [36]. The roots of this problem go back as far as the first half of the 19th century, to the works of Hamilton [13] on chess knight movement. However, its most popular formulation is in the context of traveling through a collection of cities: given a list of cities and their pairwise distances, the task is to find a shortest possible tour that visits each city exactly once.

The asymmetric (or general) traveling salesman problem (ATSP) concerns a situation when the distances between the cities are asymmetric. Formally, we are given a set V of n points and an (asymmetric) cost function $c : V \times V \to \mathbb{R}^+$. The goal is to find a minimum cost tour that visits every vertex exactly once. As is standard in the literature, throughout the paper, we make an assumption that the costs form a metric, i.e. they satisfy the triangle inequality $c_{ij} + c_{jk} \leq c_{ik}$ for all vertices i, j, and k. One should observe that if we are allowed to visit each vertex more than once, then by substituting the cost of each arc with the cost of the shortest path from its tail to its head we automatically ensure that all the triangle inequalities hold.

In the very important special case of symmetric costs, i.e., when for every $u, v \in V$ we have $c(u, v) = c(v, u)$, there is a celebrated factor 3/2 approximation algorithm due to Christofides [11]. This algorithm first computes 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 are concerned with the general, asymmetric version and give an $O(\log n/ \log\log n)$-approximation algorithm for it. This finally breaks the thirty-year-old $\Theta(\log n)$ barrier stemming from the work of Frieze et al. [17] and subsequent improvements to $0.999n, 0.842n$, and $0.666n$ [4, 22, 16]. Our approach

*NYU Stern School of Business, Department of Informations, Operations, and Management Science. aasadpou@stern.nyu.edu.
†MIT, Department of Mathematics. Supported by NSF contract CCF-0829878 and by ONR grant N00014-05-1-0148.goemans@math.mit.edu.
‡EPFL, School of Computer and Communication Sciences. This work was done while the author was at MIT and was supported in part by NSF grant CCF-0829878 and by ONR grant N00014-05-1-0148. aleksander.madry@epfl.ch.
§Computer Science Division, U.C. Berkeley. oveisgharan@berkeley.edu. Supported by a miller fellowship.
¶Stanford University, Department of Management Science and Engineering. saberi@stanford.edu.

bears some similarities to Christofides' algorithm. We first construct a spanning tree[1] with certain special properties. (Although these special properties are much harder to satisfy than the ones needed in Christofides' algorithm.) Then, we find a minimum cost Eulerian augmentation of this tree, and finally, shortcut the resulting Eulerian walk. (Recall that for undirected graphs, being Eulerian means being connected and having even degrees, while for directed graphs it means being (strongly) connected and having the in-degree of every vertex equal to its out-degree.)

The key element of our algorithm is the initial step of choosing the spanning tree. To make this choice, we first obtain an optimum solution $\boldsymbol{x}^*$ to the Held-Karp relaxation of the asymmetric TSP [24]. Then, our goal is to find a spanning tree that, on one hand, has relatively small cost compared to the cost $\mathsf{OPT}_{\mathsf{HK}}$ of the solution $\boldsymbol{x}^*$, and, on the other hand, has the crucial property of being "thin" with respect to $\boldsymbol{x}^*$. Roughly speaking, a spanning tree T is "thin" with respect to $\boldsymbol{x}^*$ if, for each cut $\emptyset \neq U \subset V$, the number of arcs of T with exactly one endpoint in U is within a small multiplicative factor ("thinness") of the corresponding value of $\boldsymbol{x}^*$ on all the arcs that have exactly one endpoint in U. (See Section 4 for a precise definition.) The central fact motivating our interest in such a tree is that we show that if a spanning tree is "thin" then the cost of its Eulerian augmentation is within a factor of its "thinness" of the cost $\mathsf{OPT}_{\mathsf{HK}}$ of $\boldsymbol{x}^*$ (and thus, within the same factor of the optimum).

In the light of the above, the technical core of our paper comprises a way of finding such a thin tree with thinness of $O(\log n/\log\log n)$ and cost being within a constant of $\mathsf{OPT}_{\mathsf{HK}}$. We achieve this by, first, symmetrizing (and scaling by $(1 - 1/n)$) our solution $\boldsymbol{x}^*$ to the Held-Karp relaxation, to obtain a vector $\boldsymbol{z}^*$ and, then, sampling a tree from a certain, carefully chosen, distribution over the spanning trees of the corresponding symmetrized graph. This distribution can be seen as a one that is maximizing entropy among all the distributions that (approximately) preserve the edge marginals imposed by $\boldsymbol{z}^*$. The crucial property of such entropy-maximizing distribution is that the events corresponding to edges being present in the sampled tree are negatively correlated. This means that the well-known Chernoff upper-bound for the independent setting still holds (see Panconesi and Srinivasan [35]) and thus, by using this tail bound together with the union-bounding technique of Karger [27], we are able to establish the desired $O(\log n/\log\log n)$-thinness of the sampled tree (with high probability). We believe that this general, maximum entropy–based approach to rounding is also of independent interest. In particular, after the initial publication of this work, it was used to obtain an improved approximation algorithm for the graphic variant of the symmetric TSP [34].

The high level description of our algorithm can be found in Algorithm 1. Also, the proof of our main theorem (Theorem 6.3) gives a more formal overview of the algorithm. In the rest of the introduction we provide an overview of our technical contributions. In Subsection 1.1 we motivate the definition of thin spanning trees and in Subsection 1.2 we describe our maximum entropy-based approach for finding a thin spanning tree.

1.1 Thin Spanning Trees

Suppose we are given an oriented tree $\vec{T}$ spanning over the set V and we want to turn it into a Asymmetric TSP tour (in the next section we describe our ideas for choosing $\vec{T}$). The minimum cost Eulerian augmentation of $\vec{T}$ is a minimum cost set of arcs that can be added to $\vec{T}$ such that the resulting graph is Eulerian, i.e., the in-degree of every vertex equals its out-degree. This problem can be formulated as a minimum cost flow problem and its solution can be computed efficiently. The cost of the resulting ATSP tour is the sum of the cost of $\vec{T}$ and the cost of the Eulerian augmentation. So, to upper-bound the approximation factor of our algorithm, we have to analytically upper-bound the cost of the Eulerian augmentation. We show that if $\vec{T}$ is *thin* with respect to an optimal solution of Held-Karp relaxation, then the cost of the Eulerian augmentation is not significantly larger than the optimum.

By the integrality of the flow polytope, the cost of any *fractional* Eulerian augmentation for $\vec{T}$ upper-bounds the cost of the minimum Eulerian augmentation. We say a capacity vector $u(.)$ on the arcs of the graph is feasible if there is a fractional Eulerian augmentation for $\vec{T}$ where the flow of each arc a is at most $u(a)$. It follows by generalizations of the Max-flow min-cut theorem (see Theorem 4.3) that $u(.)$ is feasible

[1] By a spanning tree of a directed graph we understand a tree that is spanning once the directions of arcs are disregarded.

if for any cut $\emptyset \neq U \subset V$, the number of arcs of $\vec{T}$ with their heads in U is at most $u(\delta^-(U))$, sum of the capacity of arcs with their tails in U. Consequently, if $u(.)$ is feasible, then the cost of the minimum Eulerian augmentation of $\vec{T}$ is at most $\sum_a u(a)c(a)$.

Now, we are ready to motivate the thin spanning tree definition. Let $\boldsymbol{x}^*$ be an optimal solution of Held-Karp relaxation of Asymmetric TSP. We say $\vec{T}$ is α-thin with respect to $\boldsymbol{x}^*$ if for any cut $\emptyset \neq U \subset V$ the number of arcs of $\vec{T}$ with exactly one endpoint in U is within α multiplicative factor of the sum of the factional values $\boldsymbol{x}^*_a$ of arcs with exactly one endpoint in U (we note that this is slightly different from our formal definition 4.1). It is easy to see that if $\vec{T}$ is α-thin with respect to $\boldsymbol{x}^*$ then the vector $u(a) := \alpha \boldsymbol{x}^*_a$ satisfies the feasibility condition that we described in the last paragraph. Putting things together, this upper-bounds the cost of the minimum Eulerian augmentation of $\vec{T}$ by $\sum_a u(a)c(a) = \alpha \cdot c(\boldsymbol{x}^*)$ (see Theorem 4.2 for the details). We say $\vec{T}$ is (α, s) thin with respect to $\boldsymbol{x}^*$, if it is α-thin and $c(\vec{T}) \leq s \cdot c(\boldsymbol{x}^*)$. Therefore, if $\vec{T}$ is (α, s)-thin, then the approximation factor of our algorithm is $\alpha + s$.

The importance of thin tree definition is that it disregards the orientation of the arcs. This helps us to eliminate the inherent asymmetry of the Asymmetric TSP. Consequently, when we are designing an algorithm to select $\vec{T}$ we can simply drop the direction of the arcs (see next section for more details). We conjecture that (α, s)-thin trees exist for small values of α, s and can be found efficiently.

Conjecture 1.1 *There exists constant values of $\alpha, s > 0$ such that for any feasible solution of the Held-Karp relaxation, an (α, s)-thin tree exist and can be found efficiently.*

The above conjecture implies a constant factor approximation algorithm for Asymmetric TSP (see also [33, Corollary 5.1] for a slightly more general variant that does not depend on the cost of the arcs). In addition, even an existential proof implies that the integrality gap of the Held-Karp relaxation for ATSP is a constant. Conversely, one can approach a lower-bound on the integrality gap of the Held-Karp relaxation by studying families of feasible solutions that do not admit thin trees. Also, Goddyn [19] showed a direct relationship between thin trees and the Jaeger's conjecture [25] on the existence of nowhere-zero flows (Jaeger's conjecture is very recently proved by Thomassen [40]). After several years, the above conjecture is only proved for planar and bounded genus graphs [33] (see also [15, 23]).

1.2 Rounding By Sampling From Maximum Entropy Distribution

As our sampling procedure is at the heart of our approach, we provide here some intuition on it. We encourage the reader to review this part after reading Section 5.

At a high level, one can view our sampling procedure as a randomized rounding approach. Namely, as we will show in Section 3 (see Lemma 3.1), if $\boldsymbol{z}^*$ is the symmetrized and slightly scaled down version of our solution $\boldsymbol{x}^*$ to the Held-Karp relaxation, then $\boldsymbol{z}^*$ is a point in the (relative interior of) spanning tree polytope of the support of $\boldsymbol{x}^*$. In other words, $\boldsymbol{z}^*$ can be seen as a "fractional spanning tree".

In the light of this observation, our goal of getting a sufficiently thin and low-cost spanning tree can be phrased as a task of rounding this fractional spanning tree represented by $\boldsymbol{z}^*$ to an integral spanning tree (i.e. a corner point of the spanning tree polytope) while roughly preserving some of its quantitative properties, namely, the number of edges in every cut in this spanning tree (i.e. ensuring it is "sufficiently thin") and also its cost.

Now, one well-known and widely used technique for rounding a fractional vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ with $0 \leq x_i \leq 1$, for each i, is the independent randomized rounding of Raghavan and Thompson [37]. In this rounding technique, each x_i is set to 1 with probability x_i, and to 0 with the remaining probability, independently for each i. This method has two very convenient properties:

(1) The resulting distribution preserves margins , i.e., the expected value of rounding of each variable x_i (its marginal) is close to its (fractional) value. Hence, every quantity that is linear in x_is (such as the total cost of the tree, or, the number of edges crossing the cut that we are interested in) remains the same in expectation.

(2) All the variables are rounded independently, which allows us to use strong concentration bounds such as Chernoff bounds.

The problem with this approach is, however, that the independent randomized rounding ignores any underlying combinatorial structures of the solution and thus might inadvertently destroy it. For example, in our case, it is not hard to see that independent rounding of variables associated with each edge of our underlying graph will not only most likely fail to deliver a tree, but – even more critically – the resulting graph will be almost surely disconnected.[2]

Therefore, our rounding procedure needs to be more careful. In particular, to ensure connectivity, we will restrict ourselves only to sampling from distributions over spanning trees.

Note that the fact that $\boldsymbol{z}^*$ is in the spanning tree polytope implies that it can be expressed as a convex combination of spanning trees. Namely, we have that

$$\boldsymbol{z}^* = c_1 T_1 + c_2 T_2 + \cdots + c_k T_k,$$

for some $c_i \geq 0$, $\sum_i c_i = 1$, and each T_i being an 0-1 vector describing some spanning tree. In fact, one can see that there can be many ways to express $\boldsymbol{z}^*$ in such a manner and each one of the resulting convex combinations defines a distribution over spanning trees that preserves the edge marginals imposed by $\boldsymbol{z}^*$.

So, each of these choices automatically ensures that the first of the above-mentioned properties of the independent randomized rounding still holds, i.e., that the resulting distribution is margin preserving. This implies, in particular, that the *expected* cost of the sampled tree as well as the *expected* numbers of its edges in each of the cuts are exactly as intended.

Unfortunately, the second property, i.e., independence, is much harder – and, in fact, impossible – to satisfy, as the underlying structure of spanning trees imposes inevitable dependencies between the edge variables. This is very problematic, as without the ability to resort to some strong concentration phenomena we cannot hope that all the (exponentially many) quantities that we are interested in preserving are indeed simultaneously close to their desired expected value.

Fortunately, there is a way to get around this difficulty. Namely, the crucial observation is that if we consider all the events corresponding to edges being a part of the sampled tree, then we do not really need these events to be fully independent. It actually suffices that they are only negatively correlated. As was observed first by Panconesi and Srinivasan [35], negative correlation is already enough for the concentration described by the upper tail of the Chernoff bound to emerge. As we then show, this slightly weaker concentration combined with union-bounding technique of Karger [27] will be sufficient to obtain the desired $O(\log n/\log\log n)$ bound on the worst case deviation from the marginals and thus, to get the corresponding approximation bound. (See Sections 5.4 and 6 for details.)

Now, in the light of the above discussion, it remains to devise a way of obtaining (and efficiently sampling from) a margin-preserving distribution over spanning trees that has such a negative correlation property. Our idea is to employ maximum entropy rounding. That is, we look at the one among all the margin-preserving distributions over spanning trees that maximizes entropy. Intuitively, this distribution maximizes the "randomness" while preserving the structural properties of the fractional solution $\boldsymbol{z}^*$. So, one can expect to see small correlation, or even negative correlation between the edges.

To complete the proof first we need to devise an algorithm to find and sample from the margin preserving maximum entropy distribution, second we need to prove the negative correlation property. We note that finding the maximum entropy distribution and sampling from it may seem intractable at first, because this distribution is supported over all (possibly exponentially many) spanning trees of our graph. The key to answer both of these questions is the close connection between maximum entropy distributions and *exponential distributions.* Exponential distributions can be viewed as a generalization of uniform distributions. In general, any set of weights γ_e assigned to edges of a graph defines an exponential distribution over spanning trees where the probability of each tree T is proportional to $\exp\left(\sum_{e\in T}\gamma_e\right)$. It was known before our work

[2]One can show that oversampling the edges by a factor of $\Omega(\log n)$ would ensure connectivity, but then the expected cost of the sampled graph would increase by the same – prohibitively large – factor.

that maximum entropy distributions are exponential distributions and any exponential distribution is a maximum entropy distribution for its own marginals (see e.g., [6, Section 5.2.4]). This has several implications. First, the maximum entropy distribution can be described concisely just by writing down γ_e for all edges, second sampling from maximum entropy distribution reduces to the problem of sampling uniform spanning trees which has been studied for many years [21, 29, 12], third the maximum entropy distribution satisfies negative correlation because any product distribution over spanning tree satisfies negative correlation [32, Chapter 4].

It remains to find the maximum entropy distribution over spanning trees that preserve the margins of $\boldsymbol{z}^*$. This problem can be cast as a simple convex programming optimization problem with exponentially many variables corresponding to the probability of each spanning tree of our graph. We then give two polynomial time algorithms that find product distributions that approximately (with a very good precision) preserve the margins of $\boldsymbol{z}^*$. The first one is a simple combinatorial algorithm based on the idea of multiplicative weight updates. The second one uses the ellipsoid method to find a near optimal to the dual of the maximum entropy convex program (See Sections 7 and 8).

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

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 to the asymmetric traveling salesman problem instance described by V and c.

1. Solve the Held-Karp LP relaxation of the ATSP instance to get an optimum extreme point solution $\boldsymbol{x}^*$. Define $\boldsymbol{z}^*$ as in (5), making it a symmetrized and scaled down version of $\boldsymbol{x}^*$. Vector $\boldsymbol{z}^*$ can be viewed as a point in the spanning tree polytope of the undirected graph on the support of $\boldsymbol{x}^*$ that one obtains after disregarding the directions of arcs. (See Section 3.)
2. Let E be the support graph of z^* when the direction of the arcs are disregarded. Find weights $\{\tilde{\gamma}\}_{e\in E}$ such that the exponential distribution on spanning trees, $\tilde{p}(T) \propto \exp\left(\sum_{e\in T}\tilde{\gamma}_e\right)$ (approximately) preserves the marginals imposed by $\boldsymbol{z}^*$, i.e., for any edge $e \in E$,
$$\sum_{T\in\mathcal{T}:T\ni e}\tilde{p}(T) \leq (1+\epsilon)z^*_e,$$
for a small enough value of ϵ. (In this paper, we show that $\epsilon = 0.2$ suffices for our purpose. See Sections 7 and 8 for a description of how to compute such a distribution.)
3. Sample $2\lceil\log n\rceil$ spanning trees $T_1,\ldots,T_{2\lceil\log n\rceil}$ from $\tilde{p}(.)$. For each of these trees, orient all its edges so as to minimize its cost with respect to our (asymmetric) cost function c. Let T^* be the tree whose resulting cost is minimal among all the sampled trees.
4. Find a minimum cost integral circulation that contains the oriented tree $\vec{T}^*$. Shortcut this circulation to a tour and output it. (See Section 4.)

The rest of the paper is organized as follows. In Section 2 we define some notations, and in Section 3 we recall the Held-Karp linear programming relaxation for ATSP. Our main proof starts afterwards. In Section 4 we formally define thin trees and we reduce our main problem to the problem of finding a thin tree. In Section 5 we formally define the maximum entropy sampling method and the maximum entropy convex program that preserves the marginals of $\boldsymbol{z}^*$. We also prove that the optimizers of this program are the exponential distributions of spanning trees. In Section 6 we prove our main theorem. Finally, in the last two sections provide two different algorithms for finding an exponential distribution that (approximately) preserve the marginals of $\boldsymbol{z}^*$; namely in Section 7 we provide a combinatorial algorithm and in Section 8 we use the ellipsoid method to solve the dual of the maximum entropy convex program.

2 Notation

Throughout this paper, we use $a = (u, v)$ to denote the arc (directed edge) from u to v and $e = \{u, v\}$ to denote an undirected edge. We will use A (resp. E) to denote the set of arcs (resp. edges) of a directed (resp. undirected) graph we are working with. (This graph will be always clear from the context.)

Now, given a function $f : A \to \mathbb{R}$ on the arcs of a graph, we define the cost of f to be

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

and, for a subset $S \subseteq A$ of arcs, we denote by

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

the sum of the values of f on this subset. We use an analogous notation for a function defined on the edge set E of an undirected graph or a vector whose entries are corresponding to the elements of A or E.

For a given subset of vertices $U \subseteq V$, we also define

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

to be the set of arcs that are, respectively, leaving, entering, and contained in U. Also, with a slightly abuse of the notation, we define $\delta^+(v) := \delta^+(\{v\})$ and $\delta^-(v) := \delta^-(\{v\})$, for each single vertex v. Similarly, for an undirected graph, $\delta(U)$ denotes the set of edges with exactly one endpoint in U, and $E(U)$ denotes the edges entirely within U.

Finally, in all that follows, log denotes the natural logarithm.

3 The Held-Karp Relaxation

Our point of start is the Held-Karp relaxation [24] of the asymmetric traveling salesman problem. In this relaxation, given an instance of ATSP with cost function $c : V \times V \to \mathbb{R}^+$, we consider the following linear program defined on the complete bidirected graph over the vertex set V:

$$\min \sum_a c(a) x_a \tag{1}$$

$$\text{s.t. } \boldsymbol{x}(\delta^+(U)) \geq 1 \qquad \forall U \subset V \text{ and } U \neq \emptyset, \tag{2}$$

$$\boldsymbol{x}(\delta^+(v)) = \boldsymbol{x}(\delta^-(v)) = 1 \qquad \forall v \in V, \tag{3}$$

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

It is well-known that an optimal solution $\boldsymbol{x}^*$ to the above relaxation can be computed in polynomial-time (either by employing the ellipsoid algorithm or by reformulating it as an LP with polynomially-bounded size). Furthermore, we can assume that $\boldsymbol{x}^*$ is an extreme point of the corresponding polytope.

Clearly, the cost $\mathsf{OPT}_{\mathsf{HK}} := c(\boldsymbol{x}^*)$ of this optimal solution $\boldsymbol{x}^*$ is a lower bound on the cost OPT of the optimal solution to the input instance of ATSP.

Now, observe that (3) implies that any feasible solution $\boldsymbol{x}$ to the Held-Karp relaxation satisfies

$$\boldsymbol{x}(\delta^+(U)) = \boldsymbol{x}(\delta^-(U)), \tag{4}$$

for any $U \subseteq V$. In other words, for any subset $U \subseteq V$ of vertices, the (fractional) number of arcs leaving U in $\boldsymbol{x}$ is equal to the (fractional) number of arcs entering it.

Our particular interest will be in a symmetrized and slightly scaled down version of $\boldsymbol{x}^*$. Namely, let us define

$$\boldsymbol{z}^*_{\{u,v\}} := \frac{n-1}{n}(x^*_{uv} + x^*_{vu}). \tag{5}$$

Let us also denote by A the support of $\boldsymbol{x}^*$, i.e., $A = \{(u,v) : x^*_{uv} > 0\}$, and by E the support of $\boldsymbol{z}^*$. For every edge $e = \{u,v\}$ of E, we define its cost as $\min\{c(a) : a \in \{(u,v),(v,u)\} \cap A\}$, which corresponds to choosing the cheaper of possible orientations of that edge. With the risk of overloading the notation, we denote this new cost of this edge e by $c(e)$. This implies, in particular, that $c(\boldsymbol{z}^*) < c(\boldsymbol{x}^*)$.

The main purpose of the scaling factor in (5) is to make $\boldsymbol{z}^*$ belong to the *spanning tree polytope* P of the graph (V,E), i.e. to ensure that $\boldsymbol{z}^*$ can be viewed as a convex combination of incidence vectors of spanning trees. In fact, as we prove below, this makes $\boldsymbol{z}^*$ belong to the relative interior of P.

Lemma 3.1 *The vector $\boldsymbol{z}^*$ defined by (5) belongs to* $\mathrm{relint}(P)$*, the relative interior of the spanning tree polytope P.*

Proof: From Edmonds' characterization of the base polytope of a matroid [14], it follows that the spanning tree polytope P is defined by the following inequalities (see [38, Corollary 50.7c]):

$$\begin{aligned} P = \{\boldsymbol{z} \in \mathbb{R}^E :\ & \boldsymbol{z}(E) = |V| - 1, && && (6)\\ & \boldsymbol{z}(E(U)) \le |U| - 1 && \forall U \subset V \text{ and } U \neq \emptyset, && (7)\\ & z_e \ge 0 && \forall e \in E.\} && (8) \end{aligned}$$

The relative interior of P corresponds to those $\boldsymbol{z} \in P$ satisfying all inequalities (7) and (8) *strictly*.

Clearly, $\boldsymbol{z}^*$ satisfies (6) since:

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

Consider any non-empty set $U \subset V$. We have

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

Since $\boldsymbol{x}^*$ satisfies (2) and (3), we have

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

showing that $\boldsymbol{z}^*$ satisfies (7) strictly. Since E is the support of $\boldsymbol{z}^*$, (8) is also satisfied strictly by $\boldsymbol{z}^*$. This shows that $\boldsymbol{z}^*$ is in the relative interior of P. □

Finally, we note that as $\boldsymbol{x}^*$ is an extremal solution, it is known that its support A has at most $3n-4$ arcs (see Theorem 15 in [20]). In addition, as $\boldsymbol{x}^*$ can be expressed as the unique solution of an invertible system with only $0-1$ coefficients, 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 Thin Trees and the Asymmetric Traveling Salesman Problem

The key element of our approach is a connection between the approximability of the asymmetric traveling salesman problem and the notion of thin trees. To describe this connection, let us first formally define thin trees.

Definition 4.1 *Given a point $\boldsymbol{z}$ in a spanning tree polytope and an $\alpha \geq 1$, we say that a tree T is α-thin with respect to $\boldsymbol{z}$ iff, for each subset $U \subset V$ of vertices,*

$$|T \cap \delta(U)| \leq \alpha \cdot \boldsymbol{z}(\delta(U)).$$

Also, we say that T is (α, s)-thin with respect to $\boldsymbol{z}$ iff it is α-thin and moreover

$$c(T) \leq s \cdot \mathsf{OPT}_{\mathsf{HK}},$$

i.e., the cost of T (after directing each of its edges according to the orientation that yields smaller cost) is at most s times the Held-Karp lowerbound $\mathsf{OPT}_{\mathsf{HK}}$ on the value of the optimal solution.

Now, let us consider $\boldsymbol{x}^*$ to be an optimal solution to the Held-Karp relaxation, as described in Section 3, and $\boldsymbol{z}^*$ to be the symmetrized and scaled down version of $\boldsymbol{x}^*$ defined in (5). By Lemma 3.1, we know that $\boldsymbol{z}^*$ belongs to the (relative interior of) the corresponding spanning tree polytope. The crucial observation that we make is that the ability to find an (α, s)-thin tree with respect to $\boldsymbol{z}^*$, for some α and s, translates directly into an ability to obtain an $(2\alpha + s)$-approximation to the asymmetric traveling salesman problem.

Theorem 4.2 *Let $\boldsymbol{x}^*$ be an optimal solution to the Held-Karp relaxation and $\boldsymbol{z}^*$ be the corresponding point in the spanning tree polytope defined in (5). If T^* is an (α, s)-thin spanning tree with respect to $\boldsymbol{z}^*$ for some α and s, then we can find in polynomial-time a Hamiltonian cycle whose cost is at most $(2\alpha + s)c(\boldsymbol{x}^*) = (2\alpha + s)\mathsf{OPT}_{\mathsf{HK}} \leq (2\alpha + s)\mathsf{OPT}$.*

Our proof of the above theorem relies on certain classical result on flow circulations. To state this result, let us recall that a *circulation* is any function $f : A \to \mathbb{R}$ such that $f(\delta^+(v)) = f(\delta^-(v))$ for each vertex $v \in V$. The following theorem (see, e.g., [38, Theorem 11.2] for a proof) gives a necessary and sufficient condition for the existence of a circulation subject to the lower and upper capacities on arcs.

Theorem 4.3 (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 $a \in A$ iff*

1. *$l(a) \leq u(a)$ for all $a \in A$ and*
2. *for all subsets $U \subseteq V$, 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 too.

We proceed now to the proof of Theorem 4.2.

Proof: [**Theorem 4.2**] Let us first orient each edge $\{u, v\}$ of T^* according 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 $c(\vec{T}^*) = c(T^*)$.

Let us now consider a minimum cost augmentation of $\vec{T}^*$ into an Eulerian directed graph. Finding such an augmentation can be formulated as a minimum cost circulation problem with integral lower capacities and infinite upper capacities. One just needs to set

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

and consider the minimum cost circulation problem

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

It is well-known (see, e.g., [38, Corollary 12.2a]) that an optimum solutions f^* to the above problem is polynomial-time computable and can be assumed to be integral (as l is such). This integral circulation f^*

can be viewed as a directed (multi)graph H that contains $\vec{T}^*$ and is Eulerian, i.e., every vertex an in-degree equal to its out-degree in H.

Hence, as H is also weakly connected (due to containing $\vec{T}^*$ as its subgraph), we can take an Eulerian walk of H and shortcut it to obtain a Hamiltonian cycle of cost at most $c(f^*)$. (We are using here the fact that the costs satisfy the triangle inequality.)

To complete the proof, it remains to bound the cost of f^*. That is, our goal is to show that $c(f^*) \leq (2\alpha + s)c(\boldsymbol{x}^*)$. To this end, let us 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 $a \in A$. Note that this claim would imply that

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

and thus establish the desired bound on the cost of f^*.

In the light of this, it only remains to establish that outstanding claim. To this end, observe that the α-thinness of T^* implies that, for any subset $U \subseteq V$ of vertices, the number of arcs of $\vec{T}^*$ in $\delta^-(U)$ is at most $\alpha \boldsymbol{z}^*(\delta(U))$, irrespectively of the orientation of T^* into $\vec{T}^*$. As a consequence,

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

where we used (4) and (5).

On the other hand, we have that

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

where we used the fact that $\boldsymbol{x}^*$ itself is a circulation (see (4)).

Therefore, we can conclude that

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

for any $U \subseteq V$ and thus, by Theorem 4.3, the circulation g indeed exists. This concludes the proof of the theorem. □

5 Maximum Entropy Sampling and Concentration Bounds

In the light of the connection established in the previous section (see Theorem 4.2), our goal is to develop a way of producing a tree that is sufficiently thin with respect to the point $\boldsymbol{z}^*$ (see (5)). We will achieve that by devising an efficient procedure for sampling from an appropriate probability distribution over spanning trees.

To this end, recall that by Lemma 3.1 we know that $\boldsymbol{z}^*$ belongs to the *relative interior* of the spanning tree polytope P that corresponds to the optimal solution $\boldsymbol{x}^*$ to our LP relaxation from Section 3. This means that not only $\boldsymbol{z}^*$ can be expressed as a convex combination of spanning trees on the support E of $\boldsymbol{x}^*$, but also that *each* coefficient in this convex combination has to be positive. In fact, there can be many ways to express $\boldsymbol{z}^*$ in this manner and *each* one of the resulting convex combinations will satisfy this property.

Furthermore, observe that each such convex combination naturally defines a distribution over spanning trees that preserves the marginal probabilities imposed by $\boldsymbol{z}^*$, i.e., it is the case that $\Pr_T[e \in T] = z_e^*$, for every edge $e \in E$. It is, therefore, tempting to use such a distribution for our thin tree sampling procedure. (After all, one can interpret the thinness condition – cf. Definition 4.1 – as a statement about approximate preservation of the respective cost- and cut-based marginals.)

However, as we already mentioned, there are many possible ways of representing $\boldsymbol{z}^*$ as a probability distribution. Which one should we choose? As it turns out, a good choice is to take the distribution that maximizes the *entropy* among all such marginal-preserving distributions. We formalize this notion as well as derive some of its crucial properties below.

5.1 Maximum Entropy Distribution

Let $\mathcal{T}$ be the collection of all the spanning trees of $G = (V, E)$ and let $\boldsymbol{z}$ be an arbitrary point in the corresponding spanning tree polytope P of G. The maximum entropy distribution $p^*(\cdot)$ with respect to the marginal probabilities imposed by $\boldsymbol{z}$ is the optimum solution of the following convex program.

$$\begin{aligned} \inf \quad & \sum_{T \in \mathcal{T}} p(T) \log p(T) \\ \text{s.t.} \quad & \sum_{T \ni e} p(T) = z_e \qquad \forall e \in E, \\ & p(T) \geq 0 \qquad\quad \forall T \in \mathcal{T}. \end{aligned} \tag{9}$$

It is not hard to see that this convex program is feasible since $\boldsymbol{z}$ belongs to the spanning tree polytope P. As the objective function is bounded and the feasible region is compact (closed and bounded), the infimum 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 $\mathsf{OPT}_{\mathsf{Ent}}$ denote the optimum value of this convex program CP (9).

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 $p(\cdot)$, 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 $\boldsymbol{z}$ 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. (Observe that our vector $\boldsymbol{z}^*$ indeed satisfies this assumption.)

For this purpose, we write the Lagrange dual to the convex program CP (9) (see, e.g., [3] for the relevant background). For every $e \in E$, we associate a Lagrange multiplier δ_e to the constraint corresponding to the marginal z_e, and define the Lagrange function as

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

We can then rewrite it 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 (9) is now

$$\sup_{\delta} \inf_{p \geq 0} L(p, \delta). \tag{10}$$

The inner infimum in this dual is easy to solve. Namely, as the contributions of each $p(T)$ 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 partial derivatives with respect to $p(T)$, we derive that

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

or

$$p(T) = e^{\delta(T) - 1}. \tag{11}$$

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 (10) can therefore be rewritten as

$$\sup_{\gamma} \left[1 + \sum_{e\in E} z_e\gamma_e - \sum_{T\in\mathcal{T}} e^{\gamma(T)} \right]. \tag{12}$$

Now, our assumption that the vector $\boldsymbol{z}$ is in the relative interior of the spanning tree polytope translates to satisfying the Slater condition and, together with convexity, implies that the sup in (12) is attained by some vector γ^*, and the Lagrange dual value equals the optimum value $\mathsf{OPT}_{\mathsf{Ent}}$ of our convex program. Furthermore, we have that the (unique) primal optimum solution p^* and any dual optimum solution γ^* must satisfy

$$L(p,\gamma^*) \geq L(p^*,\gamma^*) \geq L(p^*,\gamma), \tag{13}$$

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 (11), we have that

$$p^*(T) = e^{\gamma^*(T)}. \tag{14}$$

In summary, the following theorem holds.

Theorem 5.1 *Given a vector $\boldsymbol{z}$ in the relative interior of the spanning tree polytope P of $G = (V,E)$, 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 $\boldsymbol{z}$ is in the relative interior of the spanning tree polytope (as opposed to being just in this polytope) is crucial. (This has been already observed before, see Exercise 4.19 in [32]). To see that, consider G being a triangle and $\boldsymbol{z}$ being the vector $(\frac{1}{2}, \frac{1}{2}, 1)$. In this case, one can verify that $\boldsymbol{z}$ 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 Sections 7 and 8 we show how to efficiently find $\tilde{\gamma}$s that approximately satisfy the conditions of Theorem 5.1. The method proposed in Section 7 is combinatorial, as opposed to the Ellipsoid-based method of Section 8. However, as we will see, the running time of the former grows polynomially by the inverse of the desired error, while for the latter the running time grows polylogarithmically.

More formally, we prove the following theorem whose result we use in the rest of the paper. For our application to the asymmetric traveling salesman problem, we set $z_{\min}$ to be $2^{-O(n\log n)}$ and ε to be $\frac{1}{5}$.

Theorem 5.2 *Given $\boldsymbol{z}$ 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*

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

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

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

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 $1/\varepsilon$. (For the same theorem with running time polynomial in $n = |V|$, $-\log z_{\min}$ and $\log(1/\varepsilon)$, see Section 8.)

5.2 Maximum Entropy Distribution and λ-Random Trees

Interestingly, the distributions over trees considered in Theorem 5.2 turn out to be 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 [32]) – note that in case of all λ_e's 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.

Furthermore, observe that a tree T sampled from an exponential family distribution $p(\cdot)$ as given in Theorem 5.2 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 5.3, and to derive sharp concentration bounds for the distribution $p(\cdot)$, see Section 6.

5.3 Sampling a λ-Random Tree

There is a host of results (see [21, 29, 12, 1, 7, 41, 28] 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_es 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 in that case. Therefore, we resort to an iterative approach similar to [29] that remains polynomial-time for general λ_es.

The basic idea of this approach 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 the 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:

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

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, to obtain each p_j we just need to be able to compute efficiently for a given (multi)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'. It is well-known how to perform such computation. 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 [5]). 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 to computing $p_{G'}[\lambda, f]$ is to use the fact (see e.g. Ch. 4 of [32]) 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 [18] for details).

5.4 Negative Correlation and Concentration Bounds

We now derive a concentration bound that will be instrumental in establishing the thinness of our sampled tree.

Theorem 5.3 *For each edge e, let X_e be an indicator random variable associated with the event $[e \in T]$, where T is a sampled 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 since our distribution is in product form, 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., Chapter 4 of [32].[3]

Lemma 5.4 *The random variables $\{X_e\}_E$ are negatively correlated.*

Once we have established negative correlation between the X_e's, Theorem 5.3 follows directly from the result of Panconesi and Srinivasan [35] that the upper tail part of the Chernoff bound requires only negative correlation (or even a weaker notion, see [35]) and not the full independence of the random variables.

Finally, it is worth point out that, since the initial publication of this work, another way of producing *negatively correlated* marginal-preserving probability distributions on trees (or, more generally, on matroid bases) has been proposed [10]. This approach can also be used in the framework developed in this paper.

6 Establishing Thinness of the Sampled Tree

In this section, we focus on the exponential family distribution $\tilde{p}(\cdot)$ that we obtain by applying the algorithm of Theorem 5.2 to $\boldsymbol{z}^*$. We show that the tree sampled from the distribution $\tilde{p}(\cdot)$ is almost surely "thin".

We first prove that if we focus on a particular cut then the corresponding marginal is α-approximately preserved with overwhelming probability, where α has the desired value of $O(\frac{\log n}{\log \log n})$.

Lemma 6.1 *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 $U \subset V$,*

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

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 $\boldsymbol{z}^*$ by $\tilde{z}$ as in Theorem 5.2. Hence,

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

[3] Lyons and Peres prove this fact only in the case of T being a uniform spanning tree i.e. when all λ_es are equal, but Section 4.1 of [32] contains a justification why this proof implies this property also in the case of arbitrary λ_es. As mentioned before, for rational λ_es, the main idea is to replace each edge e with $C\lambda_e$ edges (for an appropriate choice of C, e.g. the lowest command denominator of all λ_es) and consider a uniform spanning tree in the corresponding multigraph. The irrational case follows from a straightforward limit argument where $C \to \infty$.

Applying Theorem 5.3 with

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

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

$$\begin{aligned}
&\Pr[|T \cap \delta(U)| > (1+\delta)\mathrm{E}[|T \cap \delta(U)|]] \\
\leq\;& \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\tilde{z}\,(\delta(U))} \\
\leq\;& \left(\frac{e}{1+\delta}\right)^{(1+\delta)\tilde{z}\,(\delta(U))} \\
=\;& \left(\frac{e}{1+\delta}\right)^{\beta\boldsymbol{z}^*(\delta(U))} \\
\leq\;& \left[\left(\frac{e(1+\varepsilon)}{\beta}\right)^{\beta}\right]^{\boldsymbol{z}^*(\delta(U))} \\
\leq\;& n^{-2.5\boldsymbol{z}^*(\delta(U))}.
\end{aligned}$$

Note that 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 $n \geq 3$). □

Now, we are ready to combine the above concentration result with union-bounding technique of Karger [27] to establish the desired thinness of our sampled tree.

Theorem 6.2 *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 5.2. Let T^* be the tree among these samples that minimizes the cost $c(T_j)$. Then, with high probability, T^* is $(4\log n/\log\log n, 2)$-thin with respect to $\boldsymbol{z}^*$.*

Here, high probability means probability at least $1-\frac{2}{n-1}$. However, one can make this probability be $1-1/n^k$, for any k, by increasing the value of β by a factor of k.

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 6.1 we know that the probability of some particular cut $\delta(U)$ violating the β-thinness of T_j is at most $n^{-2.5\boldsymbol{z}^*(\delta(U))}$. Now, we use a result of Karger [27] 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 $\boldsymbol{z}^*$, we know that $\boldsymbol{z}^*(\delta(U)) \geq 2(1-1/n)$, it means there is at most n^l cuts $\delta(U)$ with $\boldsymbol{z}^*(\delta(U)) \leq l(1-1/n)$ 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 $|T_j \cap \delta(U)| > \beta\boldsymbol{z}^*(\delta(U))$ 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 $[(i-1)(1-1/n), i(1-1/n)]$. For $n \geq 5$, 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, the probability that there exists a cut for which the β-thinness property is violated by T^* is at most $1 - \frac{1}{n-1}$.

Now, the expected cost of T_j is

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

So, by Markov inequality we have that for any j, the probability that $c(T_j) > 2\mathsf{OPT}_{\mathsf{HK}}$ 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 $c(T^*) > 2\mathsf{OPT}_{\mathsf{HK}}$.

Now, taking a union bound for the events of the violation of β-thinness in some cut and the event of the cost being more than $2\mathsf{OPT}_{\mathsf{HK}}$, we conclude that the probability that T^* is not $(\beta, 2)$-thin is at most $1 - \frac{2}{n-1}$. This completes the proof of the theorem. □

After proving the above thinness result, we can put it together with the developments of Section 4 to establish the main result of the paper.

Theorem 6.3 *Algorithm 1 finds a $(2 + 8\log n/\log\log n)$-approximate solution to the Asymmetric Traveling Salesman Problem with high probability and in time that is polynomial in the size of the input.*

Proof: The algorithm starts by finding an optimal extreme-point solution $\boldsymbol{x}^*$ to the Held-Karp LP relaxation of ATSP of value $\mathsf{OPT}_{\mathsf{HK}}$. Next, using the algorithm of Theorem 5.2 on $\boldsymbol{z}^*$ (which is defined by (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 5.2 indeed runs in polynomial time.

Next, we use the polynomial time sampling procedure described in Subsection 5.3 to sample $2\lceil \log n \rceil$ 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 6.2, we know that T^* is $(4\log n/\log\log n, 2)$-thin with high probability.

Now, we use Theorem 4.2 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 [9]. Closing this gap is a challenging open question, and this possibly could be answered using thinner spanning trees.

Corollary 6.4 *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 constant.*

7 Solving the Maximum Entropy Convex Program: A Combinatorial Approach

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 5.2. As an alternative, in Section 8 we show that the maximum entropy convex program can also be solved via ellipsoid method. The advantage of the latter approach is that the resulting running-time dependence on ϵ is only *polylogarithmic* instead of being polynomial as in the case of the combinatorial approach described below.

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 5.2. 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 δ which is at least $\varepsilon/4$ (refer to the discussion after the statement of Lemma 7.1). 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

$$\gamma(T_{\max}) > \gamma(T) + \frac{\varepsilon\tau}{4m^2}, \tag{15}$$

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

$$\sum_{e\in E\setminus Q}\hat{q}_e = n-r-1. \tag{16}$$

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, (16) 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

$$\begin{aligned}
q_e &= \frac{\sum_{T\in\mathcal{T}:e\in T} e^{\gamma(T)}}{\sum_{T\in\mathcal{T}} e^{\gamma(T)}} \\
&= \frac{\sum_{T\in\mathcal{T}_=:e\in T} e^{\gamma(T)} + \sum_{T\in\mathcal{T}_<:e\in T} e^{\gamma(T)}}{\sum_{T\in\mathcal{T}} e^{\gamma(T)}} \\
&\leq \frac{\sum_{T\in\mathcal{T}_=:e\in T} e^{\gamma(T)}}{\sum_{T\in\mathcal{T}_=} e^{\gamma(T)}} + \frac{\sum_{T\in\mathcal{T}_<:e\in T} e^{\gamma(T)}}{\sum_{T\in\mathcal{T}} e^{\gamma(T)}} \\
&\leq \frac{\sum_{T\in\mathcal{T}_=:e\in T} e^{\gamma(T)}}{\sum_{T\in\mathcal{T}_=} e^{\gamma(T)}} + \sum_{T\in\mathcal{T}_<:e\in T} \frac{e^{\gamma(T)}}{e^{\gamma(T_{\max})}},
\end{aligned} \tag{17}$$

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 (i.e. $T_{\max}$) in the second denominator. Using (15) and the fact that the number of spanning trees is at most n^{n-2}, the second term is bounded by:

$$\begin{aligned} \sum_{T \in \mathcal{T}_< : e \in T} \frac{e^{\gamma(T)}}{e^{\gamma(T_{\max})}} &\leq n^{n-2} e^{-\varepsilon\tau/4m^2} \\ &< \frac{1}{2} n^n e^{-\varepsilon\tau/4m^2} \\ &= \frac{\varepsilon z_{\min}}{2}, \end{aligned} \tag{18}$$

by definition of τ. To handle the first term of (17), 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 (17) reduces to:

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

Together with (17) and (18), 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 [5]), as discussed in Section 5.3 (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.

8 Solving the Maximum Entropy Convex Program: Ellipsoid Algorithm

In this section we design an algorithm to find a λ-random spanning tree distribution that preserves the marginal probability of all the edges within multiplicative error of $1 + \epsilon$ using the Ellipsoid method as an alternative to the combinatorial approach provided in Section 7. The advantage of the Ellipsoid-based method is that it runs in time polynomial in $n, -\log z_{\min}$, and $\log(1/\epsilon)$ (as opposed to $1/\epsilon$ for the combinatorial approach discussed in the previous section) where $z_{\min} = \min_{e \in E} z_e$ is the smallest non-zero value assigned to the edges. Note that, as we discussed earlier, if $\boldsymbol{z} = (1 - 1/n)\boldsymbol{x}$ for $\boldsymbol{x}$ being an extreme point solution of Eq. (1) then $z_{\min} \geq 2^{-n\log(n)}$. For the sake of completeness, we briefly overview a property of convex programs that will be useful for us.

The maximum entropy convex programs that we study here has an exponential size. To efficiently find a feasible or an extreme point solution we need to provide a *separating hyperplane oracle* and use the *ellipsoid algorithm*. Let $P \subset \mathbb{R}^n$ be an arbitrary bounded polytope. Let $R > 0$ such that for a point $\mathbf{y}_0 \in \mathbb{R}^n$,

$$P \subseteq \{\mathbf{y} : \|\mathbf{y} - \mathbf{y}_0\| \leq R\}.$$

Also, let $r > 0$ such that for a point $\boldsymbol{x}_0 \in P$, $\{\boldsymbol{x} \in P : \|\boldsymbol{x} - \boldsymbol{x}_0\| \leq r\} \subseteq P$. A separating hyperplane oracle is a deterministic algorithm that for any given point $\mathbf{y} \in \mathbb{R}^n$ either decides $\mathbf{y} \in P$, or finds a vector $\mathbf{a} \in \mathbb{R}^n$ such that for all $\boldsymbol{x} \in P$,

$$\langle \mathbf{a}, \mathbf{y} \rangle < \langle \mathbf{a}, \boldsymbol{x} \rangle.$$

The following theorem follows from Khachiyan's ellipsoid algorithm.

Theorem 8.1 *If the separating hyperplane oracle runs in time polynomial with respect to n and $\log(R/r)$, then the ellipsoid algorithm finds a feasible solution of P in time polynomial with respect to n and $\log(R/r)$.*

Note that the running time is independent of the number of constraints, or the number of faces of P. The following is the main theorem of this section.

Theorem 8.2 *Given $\boldsymbol{z}$ in the relative interior of the spanning tree polytope of $G = (V, E)$. For any $e^{-n^2/2} < \epsilon \leq 1/4$, values $\tilde{\gamma}_e$ for all $e \in E$ can be found, so that if we let $\tilde{\lambda}_e = \exp(\tilde{\gamma}_e)$ for all $e \in E$, then the corresponding $\tilde{\lambda}$-random spanning tree distribution, $\tilde{\mu}$, satisfies*

$$\sum_{T \in \mathcal{T} : T \ni e} \Pr_{\tilde{\mu}}[T] \leq (1 + \varepsilon) z_e, \qquad \forall e \in E,$$

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

Very recently, Singh and Vishnoi [39] generalized and improved the above theorem; they show that for any family of discrete objects, $\mathcal{M}$, and any given marginal probability vector in the interior of the convex hull of $\mathcal{M}$, one can efficiently compute the approximate weight of the ground elements in the maximum entropy distribution if and only if there is an efficient algorithm that approximates the weighted sum of all the objects for any given weights, i.e., an efficient algorithm that approximates $\sum_{M \in \mathcal{M}} \exp(\gamma(M))$ for any vector γ. For example, since there is an efficient algorithm that approximates the weighted sum of all perfect matchings of a bipartite graph with respect to given weights γ, [26], one can approximately compute the maximum entropy distribution of the perfect matchings of any bipartite graph with respect to any given marginals in the relative interior of the perfect matching polytope (see [39] for more details).

In the rest of this section we prove the above theorem. We will use the Ellipsoid method, i.e. Theorem 8.1. Thus, we just need to provide a separating hyperplane oracle, a bound polynomial in n on the radius of a ball that contains our polytope, and a bound inversely polynomial in n on the radius of a ball in the interior of our polytope.

First, we show that the optimum value of the following convex program is the same as the optimum value of the original dual program (12).

$$\begin{aligned} \sup_{\gamma} \quad & \sum_{e} z_e \gamma_e, \\ \text{s.t.} \quad & \sum_{T \ni e} e^{\gamma(T)} \leq z_e \quad \forall e \in E \end{aligned} \tag{19}$$

This is because on one hand for any vector γ that is a feasible solution of above program,

$$1 + \sum_{e \in E} z_e \gamma_e - \sum_{T \in \mathcal{T}} e^{\gamma(T)} = 1 + \sum_{e \in E} z_e \gamma_e - \frac{1}{n-1} \sum_{e \in E} \sum_{T \ni e} e^{\gamma(T)} \geq 1 + \sum_{e \in E} z_e \gamma_e - \frac{1}{n-1} \sum_{e \in E} z_e = \sum_{e \in E} z_e \gamma_e,$$

where the last equation holds since $\boldsymbol{z}$ is a fractional spanning tree. So the optimum of CP (19) is at most the optimum of CP (12). On the other hand, since $\boldsymbol{z}$ is in the interior of spanning tree polytope, there is a unique optimum γ^* to CP (12) that satisfies Eq. (13), so for all $e \in E$, $\sum_{T \ni e} \exp(\gamma^*(T)) = \sum_{T \ni e} p^*(T) = z_e$, and γ^* is a feasible solution of CP (19). Furthermore,

$$1 + \sum_{e \in E} z_e \gamma_e^* - \sum_{T \in \mathcal{T}} \exp(\gamma^*(T)) = 1 + \sum_{e \in E} z_e \gamma_e^* - \sum_{T \in \mathcal{T}} p^*(T) = \sum_{e \in E} \gamma_e^* z_e.$$

Therefore, the optimum of CP (12) is at most the optimum of CP (19). Hence, they are equal, and the optimum of CP (19) is $\mathsf{OPT}_{\mathsf{Ent}}$.

Next, we use the ellipsoid method, Theorem 8.1, to find a near optimal solution of CP (19). The main difficulty is that the coordinates of the optimizers of CP (19) are not necessarily bounded by a function of n. First, we simply turn the optimization problem into a feasibility problem by doing a binary search on the value of the optimum. Suppose we guess the optimum is at least t. Now, instead of proving that every feasible solution of CP (19) that satisfies $\sum_e z_e \gamma_e \geq t$ falls in a ball of radius that is a polynomial function of n, we *restrict* the set of feasible solutions of CP (19) to the vectors whose coordinates are bounded by a polynomial function of n. Furthermore, to ensure that the new polytope has a non-empty interior, we relax the RHS of the constraint $\sum_{T \ni e} \exp(\gamma(T)) \leq z_e$. More precisely, for any $\alpha > 0, M > 0$ and $t \in \mathbb{R}$, let $\mathcal{F}(\alpha, t, M)$ be the following feasibility convex program

$$\begin{aligned} & \sum_e z_e \gamma_e \geq t, \\ & \sum_{T \ni e} e^{\gamma(T)} \leq (1+\alpha) z_e \qquad \forall e \in E, \\ & -M \leq \gamma_e \leq M \qquad \forall e \in E. \end{aligned} \tag{20}$$

The following lemma relates the above convex program to CP (19).

Lemma 8.3 *For any $t \leq \mathsf{OPT}_{\mathsf{Ent}}$, $\mathcal{F}(e^{-n^2/2}, t, n^4 - n^2 \log z_{min})$ is non-empty.*

Proof: We say that a vector $\gamma : E \to \mathbb{R}$ has a gap at an edge $f \in E$ if for any $e \in E$, either $\gamma_e \leq \gamma_f$ or $\gamma_e > \gamma_f + \mathsf{gap}$ where $\mathsf{gap} := n^2 - \log z_{\min}$. Observe that for any $\gamma : E \to \mathbb{R}$, the number of gaps of γ is at most $|E| \leq \binom{n}{2}$.

In the following claim we show that if γ has a gap at an edge e then we can construct another vector $\tilde{\gamma}$ with fewer number of gaps while losing a small amount in the objective function. The proof is very similar in nature to the proof of Lemma 7.3, but for the sake of completeness, we provide it here.

Claim 8.4 *Let $\gamma : E \to \mathbb{R}$ that has at least one gap. Let $T_{\max}$ be a maximum spanning tree of G with respect to weights γ, i.e., $T_{\max} = \mathrm{argmax}_T \gamma(T)$. There exists $\tilde{\gamma} : E \to \mathbb{R}$ with at least one fewer gap such that for any $e \in E$,*

$$\sum_{T \ni e} e^{\tilde{\gamma}(T)} \leq \sum_{T \ni e} e^{\gamma(T)} + n^{n-2} e^{-\gamma(T_{\max}) - \mathsf{gap}}. \tag{21}$$

and

$$\sum_e z_e \tilde{\gamma}_e \geq \sum_e z_e \gamma_e. \tag{22}$$

Proof: Suppose that γ has a gap at an edge $e^* \in E$. Let $F := \{e \in E : \gamma_e > \gamma_{e^*}\}$. Let $k = \mathrm{rank}(F)$ be the size of the maximum spanning forest of F. Recall that by definition any spanning tree of G has at most k edges from F, so $\boldsymbol{z}(F) \leq k$. We reduce the γ_e for all $e \in F$ and increase it for the the rest of the edges. In particular,

$$\tilde{\gamma}_e = \begin{cases} \gamma_e + \frac{k\Delta}{n-1} & \text{if } e \notin F, \\ \gamma_e - \Delta + \frac{k\Delta}{n-1} & \text{if } e \in F, \end{cases}$$

where $\Delta = \min_{e\in F}\gamma_e - \gamma_{e^*} - \mathsf{gap}$. Note that by the assumption of the claim $\Delta > 0$. By above definition, $\tilde{\gamma}$ does not have a gap at e^*, and for any edge $e \neq e^*$, $\tilde{\gamma}$ has a gap at e if γ has a gap at e.

First, observe that,

$$\sum_e z_e\tilde{\gamma}_e = \sum_e z_e\gamma_e + \frac{k\Delta}{n-1}\sum_e z_e - \boldsymbol{z}(F)\Delta \geq \sum_e z_e\gamma_e + k\Delta - k\Delta = \sum_e z_e\gamma_e.$$

where we used $\boldsymbol{z}(F) \leq k$. This proves (22).

It remains to prove (21). If a spanning tree T has exactly k edges from F, then $\tilde{\gamma}(T) = \gamma(T)$, and $\exp(\tilde{\gamma}(T)) = \exp(\gamma(T))$. By Lemma 7.2 any maximum weight spanning tree of (V, E, γ) or $(V, E, \tilde{\gamma})$ has exactly k edges of F. Since $\tilde{\gamma}(T) = \gamma(T)$ for any tree where $|T \cap F| = k$, the maximum spanning trees of (V, E, γ) are the same as the maximum spanning trees of $(V, E, \tilde{\gamma})$. So, $T_{\max}$ is also a maximum weight spanning tree of $(V, E, \tilde{\gamma})$.

Now, suppose a spanning tree T has less than k edges in F. Since $|T \cap F| < k$, there exists an edge $f \in (T_{\max} \cap F) \setminus T$ such that $(T \cap F) \cup \{f\}$ is a forest of G. Therefore, the unique circuit in $T \cup \{f\}$ contains an edge $e \notin F$. Thus $T' = T \cup \{f\} \setminus \{e\}$ is a spanning tree. By the definition of $\tilde{\gamma}$,

$$\tilde{\gamma}(T_{\max}) \geq \tilde{\gamma}(T') = \tilde{\gamma}(T) - \tilde{\gamma}_e + \tilde{\gamma}_f > \tilde{\gamma}(T) + \mathsf{gap}, \tag{23}$$

which yields the desired inequality. Therefore, for any tree T,

$$e^{\tilde{\gamma}(T)} \leq e^{\gamma(T)} + e^{\tilde{\gamma}(T_{\max})-\mathsf{gap}} = e^{\gamma(T)} + e^{\gamma(T_{\max})-\mathsf{gap}},$$

where the last equality follows by the fact that $|T_{\max} \cap F| = k$. Now, (21) follows by the fact that any graph at most n^{n-2} spanning trees [8]. □

Let γ^* be an optimum solution of CP (19). If γ^* does not have any gap we let $\tilde{\gamma} = \gamma^*$. Otherwise, we repeatedly apply the above claim and remove all of the gaps and find a vector $\tilde{\gamma}$ such that $\sum_e z_e\tilde{\gamma}_e \geq \sum_e z_e\gamma^*_e$, and for any edge $e \in E$,

$$\sum_{T\ni e} e^{\tilde{\gamma}(T)} \leq \sum_{T\ni e} e^{\gamma^*(T)} + |E|n^{n-2}e^{\gamma^*(T_{\max})-\mathsf{gap}} \leq z_e + n^n e^{-n^2} z_{\min} \leq (1 + n^{-n^2/2})z_e. \tag{24}$$

where the first inequality follows by the fact that γ^* has at most $|E|$ gaps, the second inequality follows by the feasibility of γ^* in CP (19) and that $e^{\gamma^*(T_{\max})} \leq \max_e z_e \leq 1$.

Since $\tilde{\gamma}$ does not have any gap

$$\max_e \tilde{\gamma}_e - \min_e \tilde{\gamma}_e \leq |E| \cdot \mathsf{gap}.$$

So, it is sufficient to lower-bound $\max_e \tilde{\gamma}_e$ and upper-bound $\min_e \tilde{\gamma}_e$. Let $f = \mathrm{argmax}_e \tilde{\gamma}_e$. Since the number of the spanning trees of G is at most n^{n-2}, [8],

$$\mathsf{OPT}_{\mathsf{Ent}} \geq \log(1/|\mathcal{T}|) \geq -\log(n^{n-2}) \geq -n\log n.$$

Therefore, we have

$$-n\log n \leq \mathsf{OPT}_{\mathsf{Ent}} = \sum_e z_e\gamma^*_e \leq \sum_e z_e\tilde{\gamma}_e \leq n \cdot \max_e \tilde{\gamma}_e.$$

On the other hand, by (24), $e^{\tilde{\gamma}(T)} \leq 2$ for any tree T, so $\min_e \tilde{\gamma}_e \leq 1$. Therefore,

$$\begin{aligned}\max_e \tilde{\gamma}_e &\leq \min_e \tilde{\gamma}_e + |E| \cdot \mathsf{gap} \leq 1 + |E| \cdot \mathsf{gap} \leq n^4 - n^2 \log z_{\min},\\ \min_e \tilde{\gamma}_e &\geq \max_e \tilde{\gamma}_e - |E| \cdot \mathsf{gap} \geq -\log(n) - |E| \cdot \mathsf{gap} \geq -n^4 + n^2 \log z_{\min}.\end{aligned}$$

This completes the proof of Lemma 8.3. □

In Algorithm Algorithm 2 we provide a separating hyperplane oracle for CP (20). Note that all the

Algorithm 2 Separating hyperplane oracle for CP (20)

Input: $\gamma \in \mathbb{R}^{|E|}$

if γ violates any of the linear constraints **then**
  return the violated inequality as a separating hyperplane.
else
  Compute $q_e(\gamma) = \sum_{T \ni e} e^{\gamma(T)}$ for every e.
  if $q_e(\gamma) \le (1+\alpha)z_e$ for all edges e **then**
    report $\gamma \in \mathcal{F}(\alpha, t, M)$.
  else
    Let $\hat{e}$ be an edge for which the constraint is violated. Compute the gradient of $q_{\hat{e}}(\gamma)$.
    return the hyperplane $\{(\gamma' - \gamma).\nabla q_{\hat{e}}(\gamma) > 0, \gamma' \in \mathbb{R}^{|E|}\}$ as a violated constraint.
  end if
end if

steps of the algorithm can be done in polynomial time. The only one which may need some explanation is computing $q_{\hat{e}}(\gamma)$ for some edge e and its gradient.

$$q_e(\gamma) = e^{\gamma_e} \sum_{T \ni e} e^{\gamma(T \setminus \{e\})} \quad \text{and} \quad \frac{\partial q_e(\gamma)}{\partial \gamma_{e'}} = e^{\gamma_e + \gamma_{e'}} \sum_{T \ni e, e'} e^{\gamma(T \setminus \{e, e'\})}.$$

Both of the above expressions can be computed efficiently by the Kirchhoff's matrix tree theorem (see [5]).

Now, we are ready to prove Theorem 8.2.

Proof: [**Theorem 8.2**] Let $\alpha = \epsilon/6$. By Lemma 8.3, $\mathcal{F}(\alpha, \mathsf{OPT}_{\mathsf{Ent}}, M)$ where $M = n^4 - n^2 \log z_{\min}$ is non-empty. Let γ^* be a point in $\mathcal{F}(\alpha, \mathsf{OPT}_{\mathsf{Ent}}, M)$ and let $B = \{\gamma : \|\gamma - \gamma^*\|_\infty \le \beta\}$, where $\beta = \epsilon/4n$. For any $\gamma \in B$,

$$\sum_e z_e \gamma_e \ge \sum_e z_e(\gamma_e^* - \beta) \ge \mathsf{OPT}_{\mathsf{Ent}} - n\beta = \mathsf{OPT}_{\mathsf{Ent}} - \epsilon/4.$$

Also, for any edge $e \in E$,

$$\sum_{T \ni e} e^{\gamma(T)} \le \sum_{T \ni e} e^{\gamma^*(T) + n\beta} \le e^{n\beta}(1+\alpha)z_e \le (1+\epsilon/2)z_e.$$

where the last inequality follows by the assumption $\epsilon < 1/4$.

So $B \subseteq \mathcal{F}(\epsilon/2, \mathsf{OPT}_{\mathsf{Ent}} - \epsilon/4, M + \beta)$. Therefore, $\mathcal{F}(\epsilon/2, \mathsf{OPT}_{\mathsf{Ent}} - \epsilon/4, M+1)$ is non-empty and contains a ball of radius $\beta = \epsilon/4n$ and is contained in a ball of radius $|E| \cdot n$ which is a polynomial function of n, $-\log z_{min}$ and $1/\epsilon$. Using binary search on t and the ellipsoid method, we can find a point γ in $\mathcal{F}(\epsilon/2, \mathsf{OPT}_{\mathsf{Ent}} - \epsilon/4, M+1)$ in time polynomial in n, $-\log z_{\min}$, and $\log 1/\epsilon$.

Since γ is a feasible solution of the CP (12), $1 + \sum_e z_e \gamma_e - \sum_T e^{\gamma(T)} \le \mathsf{OPT}_{\mathsf{Ent}}$. On the other hand, since $\gamma \in \mathcal{F}(\epsilon/2, \mathsf{OPT}_{\mathsf{Ent}} - \epsilon/4, M+1)$,

$$\sum_e z_e \gamma_e \ge \mathsf{OPT}_{\mathsf{Ent}} - \epsilon/4.$$

These two imply that

$$\sum_T e^{\gamma(T)} \ge 1 - \epsilon/4.$$

The last step is to normalize γ_e's. Define $\tilde{\gamma}(e) = \gamma(e) - \frac{\log \sum_T e^{\gamma(T)}}{n-1}$. By definition, $\sum_T e^{\tilde{\gamma}(T)} = 1$. So, for any edge $e \in E$,

$$\sum_{T \ni e} e^{\tilde{\gamma}(T)} = \frac{\sum_{T \ni e} e^{\gamma(T)}}{\sum_T e^{\gamma(T)}} \le \frac{(1+\epsilon/2)z_e}{1 - \mathsf{OPT}_{\mathsf{Ent}} + \sum_e z_e \gamma_e} \le \frac{(1+\epsilon/2)z_e}{1-\epsilon/4} \le (1+\epsilon)z_e.$$

where the first inequality follows by the fact that the optimum of (19) is $\mathsf{OPT}_{\mathsf{Ent}}$ and γ is a feasible point of that program. □

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. 12

[2] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2007. 1

[3] A. Ben-Tal and A. S. Nemirovski. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001. 10

[4] M. Bläser. A new approximation algorithm for the asymmetric TSP with triangle inequality. In *SODA*, pages 638–645, 2002. 1

[5] B. Bollobas. *Modern Graph Theory*. Springer-Verlag, 2002. 12, 19, 23

[6] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2006. 5

[7] A. Broder. Generating random spanning trees. In *FOCS*, pages 442–447, 1989. 12

[8] A. Cayley. A theorem on trees. *Quarterly Journal of Mathematics*, 23:376–378, 1889. 22

[9] M. Charikar, M. Goemans, and H. Karloff. On the integrality ratio for the asymmetric traveling salesman problem. *MOR*, 31:245–252, 2006. 15

[10] C. Chekuri, J. Vondrak, and R. Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *FOCS'*, pages 575–584, 2010. 13

[11] 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. 1

[12] C. J. Colbourn, W. J. Myrvold, and E. Neufeld. Two algorithms for unranking arborescences. *Journal of Algorithms*, 20(2):268–281, 1996. 5, 12

[13] W. J. Cook. *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation*. Princeton University Press, 2012. 1

[14] J. Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, pages 127–136, 1971. 7

[15] J. Erickson and A. Sidiropoulos. A near-optimal approximation algorithm for asymmetric tsp on embedded graphs. *CoRR*, abs/1304.1810, 2013. 3

[16] U. Feige and M. Singh. Improved approximation ratios for traveling salesman tours and paths in directed graphs. In *APPROX*, pages 104–118, 2007. 1

[17] 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. 1

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

[19] L. A. Goddyn. Some open problems i like. available at `http://www.math.sfu.ca/~goddyn/Problems/problems.html`. 3

[20] M. X. Goemans. Minimum bounded degree spanning trees. In *FOCS*, pages 273–282, 2006. 7

[21] A. Guenoche. Random spanning tree. *Journal of Algorithms*, 4:214–220, 1983. 5, 12

[22] N. S. H. Kaplan, M. Lewenstein and M. Sviridenko. Approximation algorithms for asymmetric TSP by decomposing directed regular multigraphs. *JACM*, pages 602–626, 2005. 1

[23] N. J. A. Harvey and N. Olver. Pipage rounding, pessimistic estimators and matrix concentration. In *SODA*, 2014. 3

[24] M. Held and R. Karp. The traveling salesman problem and minimum spanning trees. *OR*, 18:1138–1162, 1970. 2, 6

[25] F. Jaeger. On circular flows in graphs, finite and infinite sets. *Colloquia Mathematica Societatis Janos Bolyai 37*, pages 391–402, 1984. 3

[26] M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *JACM*, 51(4):671–697, 2004. 20

[27] D. R. Karger. Global min-cuts in RNC, and other ramifications of a simple min-cut algorithm. In *SODA*, pages 21–30, Philadelphia, PA, USA, 1993. Society for Industrial and Applied Mathematics. 2, 4, 14

[28] J. Kelner and A. Mądry. Faster generation of random spanning trees. In *FOCS*, pages 13–21, 2009. 12

[29] V. G. Kulkarni. Generating random combinatorial objects. *Journal of Algorithms*, 11:185–207, 1990. 5, 12

[30] E. Lawler, D. Shmoys, A. Kan, and J. Lenstra. *The Traveling Salesman Problem*. John Wiley & Sons, 1985. 1

[31] J. K. Lenstra and A. H. G. R. Kan. Some simple applications of the Traveling Salesman Problem. *Operations Research Quarterly*, 26:717–733, 1974. 1

[32] Y. Lyons, R. with Peres. *Probability on Trees and Networks*. Cambridge University Press, 2014. In preparation. Current version available at `http://mypage.iu.edu/~rdlyons/`. 5, 11, 12, 13

[33] S. Oveis Gharan and A. Saberi. The asymmetric traveling salesman problem on graphs with bounded genus. In *SODA*, pages 967–975, 2011. 3

[34] S. Oveis Gharan, A. Saberi, and M. Singh. A randomized rounding approach to the traveling salesman problem. In *FOCS'*, pages 267–276, 2011. 2

[35] A. Panconesi and A. Srinivasan. Randomized distributed edge coloring via an extension of the Chernoff-Hoeffding bounds. *SIAM Journal on Computing*, 26:350–368, 1997. 2, 4, 13

[36] P. A. Pevzner, H. Tang, , and M. S. Waterman. An eulerian path approach to dna fragment assembly. In *National Academy of Science 98(17)*, pages 9748–9753, August 2001. 1

[37] P. Raghavan and C. D. Thompson. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987. 3

[38] A. Schrijver. *Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 2003. 7, 8

[39] M. Singh and N. K. Vishnoi. Entropy, optimization and counting. *CoRR*, abs/1304.8108, 2013. 20

[40] C. Thomassen. The weak 3-flow conjecture and the weak circular flow conjecture. *J. Comb. Theory, Ser. B*, 102(2):521–529, 2012. 3

[41] D. B. Wilson. Generating random spanning trees more quickly than the cover time. In *STOC*, pages 296–303. ACM, 1996. 12