A Summary of the Current State of the Art Algorithms for the Asymmetric Traveling Salesman Problem

Alex Wong (asw2181) - asw2181@columbia.edu

May 1, 2017

Abstract

We look at the state of the art algorithm for the general case of the Asymmetric Traveling Salesman problem and its current known inapproximability bounds. The currently best algorithm, by Asadpour et al.[2], details an $O(\log\ n\ /\ \log\ \log\ n)$-approximation algorithm for the general case of ATSP for costs satisfying the triangle inequality. A proof by Karpinski et al.[12] gives us a hardness of approximation bounds of 75 / 74 for ATSP. We also briefly touch on some recent work on approximation factors of special cases of ATSP, including an approach by Svensson et al.[20] that details a constant factor approximation algorithm for a special case of ATSP that contains only two different edge weights of arbitrary weight where the costs satisfy the triangle inequality.

1 Introduction

The traveling salesman problem is one of the most well-known and studied problems in theoretical computer science and operations research. Many real world problems, from delivery routing to latency networks to DNA sequencing, can be framed as some variation of the traveling salesman problem. In this paper, we look at the current state of the art algorithms for the asymmetric traveling salesman problem (ATSP) where the costs satisfy the triangle inequality (costs from some point u to point w is at most the cost from point u to point v + cost from point v to point w) but the cost from some point u to point v does not necessarily equal the cost from point v to point u. From this description, we can frame the symmetric traveling salesman problem (the cost from some point u to point v does equal the cost from point v to point u) as a special case of ATSP.

In Section 2, we provide an overview of an $O(\log\ n\ /\ \log\ \log\ n)$-approximation algorithm for the general case of ATSP for costs satisfying the triangle inequality which comes from the work of Asadpour et al. [2]. The $O(\log\ n\ /\ \log\ \log\ n)$-approximation algorithm introduces the concept of the "thinness" of a spanning tree where the thinness correlates with the approximation bound of ATSP. This is an improvement over the decades-long standing $\Theta(\log n)$-approximation bound given by the work of Frieze et al. [5].

In Section 3, we give an overview of the tools and methods that Karpinski et al. [12] used to derive the current best 75/74 inapproximability bounds for ATSP. They introduce a tool called the "bi-wheel amplifier" that improves the number of satisfied constraints for a certain type of CSP. This improvement triumphs the 117/116 inapproximability bound given by Papadimitriou and Vempala [16].

In Section 4, we detail some of the recent work for special cases of ATSP. Gharan and Saberi showed that there is a constant factor approximation on ATSP of planar graphs and graphs with bounded genus [6]. Anari and Gharan showed that the integrality gap of ATSP is $O(\text{poly}\ \log\ \log\ n)$ [1], but there is no current method known to construct a solution of $O(\text{poly}\ \log\ \log\ n)$-approximation. The recent work of Svensson et al. showcased a new heuristic for tackling approximating ATSP [19], with a constant factor approximation on a special case of ATSP that contains only two different edge weights [20].

In Section 5, we will wrap up the paper with opinions on the next steps towards a constant factor approximation algorithm for the general case of ATSP.

2 An $O(\log n\ /\ \log\log n)$-approximation algorithm for ATSP

In this section, we dive deeply into the works of Asadpour et al. [2] and look through all the methods and theorems to see how the $O(\log n\ /\ \log\log n)$-approximation algorithm has resulted. Algorithm 1 details the $O(\log n\ /\ \log\log n)$-approximation algorithm for ATSP (from [2] section 1).

Algorithm 1 An $O(\log n\ /\ \log\log n)$-approximation algorithm for ATSP [2]

Input: A set V consisting of n points and a cost function $c : V \times V \to \mathbb{R}^+$ where the costs satisfy the triangle inequality.
Output: An $O(\frac{logn}{loglogn})$-approximation to the ATSP instance described by the inputs V and c.

1. Solve the Held-Karp LP relaxation of the ATSP instance to get an optimum extreme point solution $\mathbf{x}^*$. Create a symmetrized and scaled down version of $\mathbf{x}^*$ from (5) and define it as $\mathbf{z}^*$. $\mathbf{z}^*$ is a vector which can be interpretted as a point in the relative interior (not touching any boundaries) of the spanning tree polytope P created from an undirected graph supported by $\mathbf{x}^*$ where we disregard the directions of arcs. Another way to think of $\mathbf{z}^*$ is the marginal probabilities on the edges z^*_e of an exponential distribution on spanning trees $\tilde{p}(.)$.

2. Let E be the support graph of $\mathbf{z}^*$ where we disregard the directions of arcs. We then find weights $\{\tilde{\gamma}\}_{e\in E}$ such that $\tilde{p}(T)$ is approximately proportional to $\exp(\sum_{e\in T} \tilde{\gamma}_e)$ where, 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 value ϵ.

3. Sample $2\lceil logn \rceil$ spanning trees T_i from $\tilde{p}(.)$. Orient all the edges of the sampled spanning trees to minimize its cost since the spanning tree only has to be weakly connected. Let $\overrightarrow{T}$ represent an oriented tree. Let T^* be the spanning tree that has minimum total cost amongst all the sample spanning trees.

4. Find a minimum cost integral circulation that contains the oriented tree $\overrightarrow{T}^*$. This will give us a multigraph. Shortcut the multigraph to get a Eulerian tour and output the result.

Throughout the rest of this section, we will divide each step of the algorithm into subsection(s) to describe it in more detail and/or prove its correctness.

2.1 Preliminaries and Notation

In this subsection, we describe much of the notation that will be used throughout the rest of this section. We will also define the notion of ”thinness” of a spanning tree.

Let $a = (u, v)$ be the arc (directed edge) from u to v and let $e = (u, v)$ be an undirected edge. Then, A (resp. E) is the set of arcs (resp. edges) in a directed (resp. undirected) graph G.

For a given function $f : A \to \mathbb{R}$, the cost of f is defined as $c(f) := \sum_{a\in A} c(a)f(a)$. For some set of arcs $S \subseteq A$, we define $f(S) := \sum_{a\in S} f(a)$. The same notation is used for the edge set E of an undirected graph.

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

$$\delta^+(U) := \{a = (u, v) \in A : u \in U, v \notin U\},$$

$$\delta^-(U) := \{a = (u, v) \in A : u \notin U, v \in U\},$$

$$A(U) := \{a = (u, v) \in A : u \in U, v \in U\}$$
$$\delta(U) := \delta^+(U) \cup \delta^-(U),$$

to be the set of arcs that are leaving, entering, contained in U, and total of incoming and outgoing arcs, respectively. We also define $\delta^+(v) := \delta^+(\{v\})$ and $\delta^-(v) := \delta^-(\{v\})$ for each single vertex v. For an undirected graph, $\delta(U)$ represents the set of edges with just one endpoint in U, and $E(U)$ represents the edges that are contained within U. Lastly, all *log* in equations represents the natural logarithm.

We say that a spanning tree T is α-thin with respect to $\mathbf{z}$ if and only if:

$$|T \cap \delta(U)| \leq \alpha \cdot z(\delta(U)) \qquad \forall U \subset V$$

which essentially states that for all the subsets of vertices U of the spanning tree T, the number of edges in the cut between U and $V \backslash U$ must be $\leq$ the thinness multiplied by the cost of the edges in the cut.

We also say that T is (α, s)-thin with respect to $\mathbf{z}$ if and only if it is α-thin and $c(T) \leq s \cdot OPT_{HK}$, which essentially states that the cost of T is at most s times the cost of the optimal Held-Karp solution.

By showing that a spanning tree is "thin", we would be able to get a Eulerian augmentation of the spanning tree where the total cost is within a factor of α of the cost OPT_{HK}, which implies within the same factor of the optimum solution.

2.2 The Held-Karp Relaxation

We detail and prove step 1 of the algorithm. Given an ATSP instance with a cost function $c : V \times V \rightarrow \mathbb{R}^+$, we can get a lower bound on the optimal value of ATSP by the following linear programming (LP) relaxation defined on the complete bidirected graph over the vertex set V:

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

$$s.t. \;\; \mathbf{x}(\delta^+(U)) \geq 1 \qquad \forall U \subset V, \tag{2}$$

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

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

This relaxation is accurate since (2) makes sure that all the vertices are strongly connected and (3) makes sure that the indegree = outdegree for all vertices of V which makes this a Eulerian graph. This relaxation is known as the Held-Karp relaxation [10] and it's well-known that an optimal solution $\mathbf{x}^*$ to the relaxation can be computed in polynomial time (using the ellipsoid method). Thus, we can say that $c(\mathbf{x}^*) = OPT_{HK}$. Also, notice that (3) implies that any feasible solution $\mathbf{x}$ satisfies

$$\mathbf{x}(\delta^+(U)) = \mathbf{x}(\delta^-(U)) \qquad \forall U \subset V. \tag{4}$$

2.3 Proving z* $\in$ Relative Interior of P

As mentioned in the algorithm, $\mathbf{z}^*$ is the symmetreized and scaled down version of $\mathbf{x}^*$. So, we formally define

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

As mentioned in step 2 of the algorithm , we let E be the support of $\mathbf{z}^*$. We also let A be the support of $\mathbf{x}^*$ where $A = \{(u, v) : x^*_{uv} > 0\}$. We also define the costs of all edges $e \in E$ where $c(e) = min\{c(a) : a \in \{(u, v), (v, u)\} \cap A\}$, $\forall e \in E$. This shows that $c(\mathbf{z}*) < c(\mathbf{x}*)$. Now we can prove the rest of step 1 of the algorithm and define a lemma:

Lemma 2.1 ([2], section 3): The vector $\mathbf{z}^*$ belongs to the relative interior of the spanning tree polytope P.

Proof: From the characterization of the base polytope of a matroid given by Edmonds[4], it follows that the spanning tree polytope P is defined by the following inequalities (see Corollary 50.7c of [18]):

$$P = \{z \in \mathbb{R}^E : z(E) = |V| - 1, \tag{6}$$

$$z(E(U)) \leq |U| - 1 \quad \forall U \subset V \ where \ U \neq \emptyset, \tag{7}$$

$$z_e \geq 0 \quad \forall e \in E.\} \tag{8}$$

Thus, for some vector z to belong in the relative interior of P, it must satisfy both inequalities (7) and (8) strictly ($z(E(U)) < |U| - 1$ and $z_e > 0$).
From the above equations, we see that $\mathbf{z}^*$ satisfies constraint (6) because:

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

Now, consider any set $U \subset V$ where $U \neq \emptyset$. We get that

$$\sum_{v \in U} \mathbf{x}^*(\delta^+(v)) \ = \ |U| \ = \ \mathbf{x}^*(A(U)) + \mathbf{x}^*(\delta^+(U)) \ \geq \ \mathbf{x}^*(A(U)) + 1$$

$$\Rightarrow \ \mathbf{x}^*(A(U)) \ \leq \ |U| - 1$$

We see that $\mathbf{x}^*$ satisfies constraints (2) and (3), so we can use equation (5) to show

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

$$\Rightarrow \ \mathbf{z}^*(E(U)) < |U| - 1$$

where we see that $\mathbf{z}^*$ satisfies inequality (7) strictly. Also, since E is the support of $\mathbf{z}^*$, that means $\mathbf{z}^*$ also satisfies inequality (8) strictly. Thus, we can conclude that $\mathbf{z}^*$ is in the relative interior of P. $\square$

We also make a note that since $\mathbf{x}^*$ is an optimum extreme point solution, it's known that its support A has at most $3n - 4$ arcs (see [9], Theorem 15). Also, $\mathbf{x}^*$ can be expressed as the unique solution (due to global minima of a convex program) of an invertible system with only 0 - 1 coefficients, then every entry $\mathbf{x}^*_a$ is rational with an integral numerator and denominator bounded by $2^{O(nlogn)}$, thus we can say that $\mathbf{z}^*_{min} > 2^{-O(nlogn)}$ (see [2] section 3).

2.4 Thin Trees, Flow Circulation, and Their Relation to ATSP

A keyf element to the algorithm of approximating ATSP is the notion of thin trees. In section 2.1, we briefly described what "thinness" means and what properties a thin tree has. We will now detail it further to showcase why thin trees are crucial to this approximation algorithm and prove its correctness.

We jump to step 4 of the algorithm to prove it. First, though, a theorem that is required for the proof of correctness of thin trees is Hoffman's circulation theorem [18]. A circulation is essentially any function $f : A \to \mathbb{R}$ such that $f(\delta^+(v)) = f(\delta^-(v))$ for each vertex $v \in V$. Hoffman's circulation theorem gives a necessary and sufficient condition for the existence of circulation on the arcs subject to the arcs' lower and upper capacities. The following is Hoffman's circulation theorem (we won't prove Hoffman's circulation theorem; see [18], Theorem 11.2 for the proof):

Theorem 2.2 (Hoffman's Circulation Theorem [18]): Given lower and upper capacities $l, u : A \to \mathbb{R}$, there exists a circulation f satisfying $l(a) \leq f(a) \leq u(a)$ for all $a \in A$ if and only if:

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

Given this theorem on circulation, we now provide a theorem that proves that the ability to find an (α, s)-thin spanning tree with respect to $\mathbf{z}^*$, for some α and s, translates directly into an ability to obtain an $(2\alpha + s)$-approximation to ATSP:

Theorem 2.3 ([2], section 4): Let $\mathbf{x}^*$ be an optimal solution to the Held-Karp relaxation and $\mathbf{z}^*$ to be the symmetrized and scaled down version of $\mathbf{x}^*$ defined in (5). if T^* is an (α, s)-thin spanning tree with respect to $\mathbf{z}^*$ for some α and s, then we can find, in polynomial-time, a Hamiltonian cycle whose cost is at most $(2\alpha + s)c(\mathbf{x}^*) = (2\alpha + s)OPT_{HK} \leq (2\alpha + s)OPT$.

Proof: We first orient the edges (u, v) of T^* to minimize the cost of T^* where each arc $a \in T^* = min\{c(a) : a \in \{(u, v), (v, u)\} \cap A\}$, and we denote the oriented tree as $\overrightarrow{T}^*$. Notice that by the definition of the undirected cost function that we defined back in section 2.3, we have $c(\overrightarrow{T}^*) = c(T^*)$.

Lets now consider an augmentation of minimum cost to make $\overrightarrow{T}^*$ an Eulerian directed graph. To get the augmentation, we can translate it as a minimum cost circulation problem with integral lower capacities and a bound on the upper capacities bsed on what we want to prove here in Theorem 2.3. So, we first set the lower capacities of the arcs by

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

and the minimum circulation problem is $min\{c(f) : f \text{ is a circulation and } f(a) \geq l(a),\ \forall a \in A\}$. This minimum circulation problem is well known so we do not prove it here (see [18], Corollary 12.2a for the proof). The minimum cost circulation problem's proof will give us an optimal integral circulation solution f^* in polynomial time. When we augment this solution with $\overrightarrow{T}^*$, we end up with a directed multigraph that is Eulerian (indegree = outdegree $\forall v \in V$), which we will define as H. Thus, H is strongly connected so we can take a Eulerian tour and shortcut it to get a Hamiltonian cycle of cost $\leq c(f^*)$.

The last part of this proof is showing an upper bound on $c(f^*)$. As we stated in our theorem, we are looking for cost $\leq (2\alpha + s)c(\mathbf{x}^*)$, meaning that $c(f^*) \leq (2\alpha + s)c(\mathbf{x}^*)$. So, we can define the upper capacities of the arc by

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

Now, we claim that there $\exists$ a circulation g where $l(a) \leq g(a) \leq u(a)\ \forall a \in A$. This implies that

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

which gives us the bound we want for $c(f^*)$. To prove this claim, we see that the α-thinness of T^* implies that for any $U \subseteq V$, the number of arcs of $\overrightarrow{T}^* \in \delta^-(U) \leq \alpha \mathbf{z}^*(\delta(U))$, regardless of the orientation where T^* oriented to $\overrightarrow{T}^*$. Thus, we can say that

$$l(\delta^-(U)) \leq \alpha \mathbf{z}^*(\delta(U))$$

and by using a combination of (4) and (5), we can say that

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

We also know that

$$2\alpha \mathbf{x}^*(\delta^+(U)) \leq u(\delta^+(U)).$$

By equation (4), we can say that

$$2\alpha\mathbf{x}^*(\delta^+(U)) = 2\alpha\mathbf{x}^*(\delta^-(U)).$$

So, we can finally conclude

$$l(\delta^-(U)) \leq 2\alpha\mathbf{x}^*(\delta^-(U)) = 2\alpha\mathbf{x}^*(\delta^+(U)) \leq u(\delta^+(U))$$
$$\Rightarrow l(\delta^-(U)) \leq u(\delta^+(U))$$

for any $U \subseteq V$, thus showing that the circulation g does exist, which concludes the proof of this theorem. □

2.5 Maximum Entropy Distribution

In this subsection, we detail the approach on how to find an entropy distribution $p(.)$. We let $\mathcal{T}$ represent the collection of all spanning trees of $G = (V, E)$ and let $\mathbf{z}$ represent some arbitrary point in the relative interior of the spanning tree polytope P of G. We now let $p^*(.)$ be the maximum entropy distribution with respect to the marginal probabilites imposed by $\mathbf{z}$. This is considered the optimum solution of the following convex program:

$$inf \qquad \sum_{T\in\mathcal{T}} p(T)logp(T) \tag{9}$$
$$s.t. \qquad \sum_{T\ni e} p(T) = z_e \qquad \forall e \in E,$$
$$p(T) \geq 0 \qquad \forall T \in \mathcal{T}.$$

This convex program is feasible since $\mathbf{z}$ is in the relative interior of the spanning tree polytope P and as the objective function is bounded and the feasible region is compact, the infinum (greatest lower bound) is found, which shows the existence of an optimal solution $p^*(.)$. Also, since the objective function is stricly convex (any local minima is essentially a global minima), this implies that the maximum entropy distribution $p^*(.)$ is unique. OPT_{CP} will represent the optimum value of the convex program (9).

We define $p(T)$ as the probability of sampling a spanning tree T in $p(.)$. Also, for any feasible solution $p(.)$, we have $\sum\limits_{T} p(T) = 1$ because

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

based on the constraints given by convex program (9). We now define a theorem:

Theorem 2.4 (see [2], section 5): Given a vector $\mathbf{z}$ in the relative interior of the spanning tree polytope P of $G = (V, E)$, $\exists\ \gamma_e^*\ \forall e \in E$, such that if we sample a spanning tree T of G according to $p^*(T) := e^{\gamma^*(T)}$, then $\Pr[e \in T] = z_e \ \ \forall e \in E$.

Proof: Now, assuming that $\mathbf{z}$ is in the relative interior of the spanning tree polytope P, we want to now show that show that $p^*(T) > 0 \ \ \forall T \in \mathcal{T}$ and that $p^*(T)$ gives a simple exponential formula. To show this, we find the Lagrange to the convex program (9). Thus, $\forall e \in E$, we have a Lagrange multiplier δ_e (not to be confused with δ^- and δ^+ which represents incoming and outgoing edges, respectively) to the constraint corresponding to the marginal probability z_e, and we define the Lagrange function as

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

which can be rewritten as

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

thus making the Lagrange dual to the convex program (9)

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

We solve the inner infimum. Since each of the contributions of $p(T)$ is seperable, then $\forall T \in \mathcal{T}$, $p(T)$ must minimize the convex function $p(T)logp(T) - p(T)\delta(T)$ where $\delta(T) = \sum_{e \in T} \delta_e$. Now, taking partial derivatives of the convex function with respect to $p(T)$, we get

$$1 + logp(T) - \delta(T) = 0$$

$$logp(T) = \delta(T) - 1$$

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

So, the inner infimum

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

and we use a change of variables to define $\gamma_e = \delta_e - \frac{1}{n-1}$ for $e \in E$, the dual (10) can be rewritten as

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

This tells us that vector $\mathbf{z}$ being in the relative interior of P satisfies Slater's condition and being convex implies that the supremum of (12) is given by γ^*, and thus the Lagrange dual value = the optimum value OPT_{CP}. Also, given that we have the primal optimum solution p^*, 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. Thus, p^* is the unique minimizer of $L(p,\gamma^*)$ and from equation (11),

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

thus proving the theorem. □

To find the weights $\tilde{\gamma}_e, \forall e \in E$, requires lengthy non-trivial calculations which will not be detailed within this paper. There are two methods to find $\tilde{\gamma}_e$'s: either the combinatorial approach (see [2] section 7) where the basic idea is to iteratively tweak the weights of $\tilde{\gamma}_e$ until their marginals $= (1+\epsilon/2)z_e$ or the ellipsoid method (see [2] section 8) where the basic idea is to find a near optimal solution of the dual of the convex program. We will define a theorem that utilizes those weights $\tilde{\gamma}_e$ since it is crucial for subsequent theorems and proofs:

Theorem 2.5 (see [2], section 5): Given $\mathbf{z}$ in the spanning tree polytope of $G = (V,E)$ and some $\epsilon > 0$, values $\tilde{\gamma}_e \ \ \forall e \in E$ 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 $e \in E$,

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

This theorem essentially states that the marginal probabilities of $\mathbf{z}$ are approximately preserved. Also, we set $\epsilon = 0.2$ and $z_{min} = 2^{-O(nlogn)}$ to be used for calculations in subsequent subsections.

We now define the notion of λ-random trees, which essentially correlate with spanning trees. We say that a λ-random tree T of G is a spanning tree T chosen from the set of all spanning trees of G with probability proportional to $\prod_{e \in T} \lambda_e$ where $\lambda_e \geq 0$ for $e \in E$. For the case where all of the λ_e's are equal, the result is a uniform spanning tree of G. For rational λ_e's, a λ-random spanning tree in G correlates to a uniform spanning tree in a multigraph obtained from G by allowing the multiplicity of edge e be proportional to λ_e. From Theorem 2.5, we see that if we sample a tree T from an exponential distribution $p(.)$, the tree T is then λ-random (spanning) for $\lambda_e := e^{\gamma_e} \quad \forall e \in E$.

2.6 λ-Random (Spanning) Trees

To sample a λ-tree, we will utilize an iterative approach similiar to [13] so that our approach will have a polynomial running time for general λ_e's. The basic idea is to order the edges $e_1, ..., e_m$ of G arbitrarily and then iteratively process the edges by deciding probabilistically whether or not to add the edge to the final tree or discard the edge. Essentially, when processing the i-th edge e_i, we decide whether or not to add it to the final spanning tree T based on the probability p_i, which is based off the probability that e_i is in a λ-random tree conditioned on the past decisions that were made for edges $e_1, ..., e_{i-1}$ in past iterations. To compute these probabilities efficiently, first note that $p_1 = z_{e1}$. If we choose to include e_1 in the tree, then

$$p_2 = Pr[e_2 \in T | e_1 \in T] = \frac{\sum\limits_{T' \ni e_1, e_2} \prod\limits_{e \in T'} \lambda_e}{\sum\limits_{T' \ni e_1} \prod\limits_{e \in T'} \lambda_e}$$

$$= \frac{\sum\limits_{T' \ni e_1, e_2} \prod\limits_{e \in T' e_1} \lambda_e}{\sum\limits_{T' \ni e_1} \prod\limits_{e \in T' e_1} \lambda_e}.$$

This tells us that the probability of $e_2 \in T$ on the condition that $e_1 \in T$ is equal to the probability that e_2 is in a λ-random tree of a graph given by G by contracting edge e_1. If we discard e_1, then the probability p_2 is equal to the probability that e_2 resides in a λ-random tree of a graph given by G by removing e_1. Thus, for the general case, we can say that p_i is equal to the probability that edge e_i resides in a λ-random tree of a graph given by G by contracting all edges that we decided to add to the tree and deleting all the edges that we decided to discard. Thus, to get each p_i, we compute the probability $p_{G'}[\lambda, f]$ that some edge f is in a λ-random tree of a given multigraph $G^{'}$ and values of λ_e's by noting that $p_{G'}[\lambda, f]$ is equal to λ_f times the effective resistence of f in G' treated as an electrical circuit with conductances of edges given by λ ([14], Ch. 4). Intuitively, $p_{G'}[\lambda, f]$ will be negatively correlated with resistance. The effective resistance is computed by the method given by [8], Section 2.4, where, in simplistic terms, involves inverting some matrix that is derived from the Laplacian of G'. The Laplacian L is where

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

To put it in words, the Laplacian entry $L_{i,j}$ is equal to the negative λ weight of edge e assuming there exists an edge $e = (i,j) \in E$, or equal to the sum of all the λ weights of all incoming and outgoing edges of the vertice $i \in V$, or it is 0 otherwise.

A concentration bound is key to establishing the thinness of a sampled tree. Thus, we derive the concentration bounds with the following theorem:

Theorem 2.6 (see [2], section 5): 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, we 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, in order to obtain concentration bounds, we would prove that the variables $\{X_e\}_E$ are independent and use Chernoff bounds. But, our variables $\{X_e\}_E$ are not independent. Rather, they are negatively correlated since our probability distribution is in product form ([14], Ch. 4). Thus, we get the following lemma:

Lemma 2.7 ([2], Section 5): The indicator random variables $\{X_e\}_E$ are negatively correlated.

Now, since we have determined that the random variables are negatively correlated, we can use the fact that the upper tail part of the Chernoff bound only requires negative correlation [15].

2.7 O(log n / log log n) Thinness of the Sampled Tree

We first define $\tilde{p}(.)$ as the exponential distribution that we get from applying Theorem 2.5 to $\mathbf{z}^*$. We prove that a sampled tree from $\tilde{p}(.)$ is almost guaranteed to be "thin". But first, we prove that if we look at a particular cut, the corresponding sampled tree will have the α-thinness property with high probability where $\alpha = O(\frac{logn}{loglogn})$.

Lemma 2.6 ([2], Section 6): If T is a spanning tree sample from distribution $\tilde{p}(.)$ for $\epsilon = 0.2$ in a graph G with $n \geq 5$ vertices then, for any subset $U \subset V$,

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

where $\beta = 4logn/loglogn$.

Proof ([2], Section 6): $\forall e \in E, \tilde{z}_e \leq (1+\epsilon)z^*_e$, where $\epsilon = 0.2$ is our desired accuracy of approximation of $\mathbf{z}^*$ by $\tilde{z}$ as given in section 2.5, Theorem 2.5. Thus

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

Applying Theorem 2.6 with

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

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

$$Pr\Big[|T \cap \delta(U)| > (1+\delta)E[|T \cap \delta(U)|\Big] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\tilde{z}(\delta(U))} \leq \left(\frac{e}{1+\delta}\right)^{(1+\delta)\tilde{z}(\delta(U))}$$

$$= \left(\frac{e}{1+\delta}\right)^{\beta z^*(\delta(U))} \leq \left[\left(\frac{e(1+\epsilon)}{\beta}\right)^{\beta}\right]^{z^*(\delta(U))} \leq n^{-2.5z^*(\delta(U))}$$

Where in the last inequality, we did

$$log\left[\left(\frac{e(1+\epsilon)}{\beta}\right)^{\beta}\right] = 4\frac{logn}{loglogn}[1 + log(1+\epsilon) - log(4) - loglogn + logloglogn]$$

$$\leq -4logn\left(1 - \frac{logloglogn}{loglogn}\right) \leq -4\left(1 - \frac{1}{e}\right)logn \leq -2.5logn,$$

because $e(1+\epsilon) < 4$ and $\frac{logloglogn}{loglogn} \leq \frac{1}{e}$ for all $n \geq 5$. □

Now we can combine the concentration results along with the union-bounding technique of Karger [11] to get the desired thinness of a sampled tree.

Theorem 2.7 ([2], Section 6): Let $n \geq 5$ and $\epsilon = 0.2$. Let $T_1, ..., .T_{\lceil 2logn \rceil}$ be $\lceil 2logn \rceil$ independent samples from a distribution $\tilde{p}(.)$ as given in Theorem 2.5. Let T^* be the tree among these samples that minimizes the cost $c(T_j)$. Then, with high probability, $T*$ is $(4logn/loglogn, 2)$-thin with respect to $\mathbf{z}^*$.

We say high probability is a probability that is at at least $1 - \frac{1}{n-1}$.

Proof ([2], Section 6): First, by Lemma 2.6, the probability that some cut $\delta(U)$ violates the β-thinness of T_j is at most $n^-2.5z^*(\delta(U))$. This shows that for any tree T_j where $1 \leq j \leq \lceil 2logn \rceil$, T_j is β-thin with high probability for $\beta = 4logn/loglogn$.

Now we show that there are at most n^{2l} cuts of size at most l times the minimum cost value for any half-integer $l \geq 1$ [11]. By the Held-Karp relaxation and $\mathbf{z}^*$, $\mathbf{z}^*(\delta(U)) \geq 2(1 - 1/n)$, which shows there is at most n^l cuts $\delta(U)$ with $\mathbf{z}^*(\delta(U)) \leq l(1 - 1/n)$ for any integer $l \geq 2$. Thus, applying union bound [11] along with $n \geq 5$, we find that the probability that there exists a cut $\delta(U)$ with $|T_j \cap \delta(U)| > \beta\mathbf{z}^*(\delta(U))$ is at most

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

showing that there exists a cut where the β-thinness property is violated by T^* is at most $1 - \frac{1}{n-1}$. The excepted cost of T_j is

$$E[c(T_j)] \leq \sum_{e \in E} \tilde{z}_e \leq (1+\epsilon)\frac{n-1}{n}\sum_{a \in A} x_a^* \leq (1+\epsilon)OPT_{HK}$$

From here, we can use Markov's inequality ($Pr[X \geq a] \leq E(X)/a$) to say that for any T_j, the probability that $c(T_j) > 2OPT_{HK}$ is at most $(1+\epsilon)/2$, which shows that with a probability at most $(\frac{1+\epsilon}{2})^{2logn} < 1/n$ for $\epsilon = 0.2$, we have $c(T^*) > 2OPT_{HK}$, thus proving the theorem. □

Now we can prove that there is $O(logn/loglogn)$-approximation factor solution to ATSP.

Theorem 2.8 ([2], Section 6): Algorithm 1 finds a $(2 + 8logn/loglogn)$-approximate solution to the Asymmetric Traveling Salesman Problem with high probability and in time that is polynomial in the size of the input.

Proof: We start with step 1 of the algorithm to find an optimal extreme-point solution $\mathbf{x}^*$ to the Held-Karp LP relaxation of ATSP. In section 2.2, we detailed that the value of $\mathbf{x}^* = OPT_{HK}$. We then define $\mathbf{z}^*$ by (5) in section 2.3.

Next, in step 2 of the algorithm, we use Theorem 2.5 on $\mathbf{z}^*$, along with $\epsilon = 0.2$, to get the weights $\{\tilde{\gamma}\}_{e \in E}$. This runs in polynomial time because since $\mathbf{x}^*$ was an extreme point, then $z_{min}^* \geq e^{-O(nlogn)}$.

For step 3 of the algorithm, we use the sampling method detailed in section 2.6 to sample $2\lceil logn \rceil$

spanning trees in polynomial time. And from Theorem 2.7, we know that T^* is $(4logn/loglogn, 2)$-thin with high probability.

Lastly, in step 4 of the algorithm, we use Theorem 2.3 to get a $((8logn/loglogn)+2)$-approximation of the ATSP instance. This concludes the proof of this theorem. □

3 Inapproximability Bounds of ATSP

4 Special Cases of ATSP

4.1 ATSP on Graphs with Bounded Genus

4.2 Polyloglog$n$ Integrality Gap of ATSP

4.3 Local-Connectivity ATSP

5 Concluding Remarks

References

[1] Anari, N., and Gharan, S. O. Effective-resistance-reducing flows, spectrally thin trees, and asymmetric TSP. *In: Proceedings of FOCS* (2015).

[2] Asadpour, A., Goemans, M. X., Madry, A., Gharan, S. O., and Saberi, A. An $\mathcal{O}$(logn / log logn)-approximation algorithm for the asymmetric traveling salesman problem. *In: Proceedings of SODA* (2010), pages 379–389.

[3] Christofides, N. Worst-case analysis of a new heuristic for the traveling salesman problem. *Technical Report, DTIC Document* (1976).

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

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

[6] Gharan, S. O., and Saberi, A. The asymmetric traveling salesman problem on graphs with bounded genus. *In: Proceedings of SODA* (2011), pages 967–975.

[7] Gharan, S. O., Saberi, A., and Singh, M. A randomized rounding approach to the traveling salesman problem. *In: Proceedings of FOCS* (2011).

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

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

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

[11] Karger, D. R. Global min-cuts in rnc, and other ramifications of a simple min-cut algorithm. *In SODA* (1993), pages 21–30.

[12] Karpinski, M., Lampis, M., and Schmied, R. New approximability bounds for TSP. *J. Comput. Syst. Sci.* (2015), 81(8):1665–1667.

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

[14] Lyons, R., and Peres, Y. *Probability on Trees and Networks.* Cambridge University Press, 2014.

[15] Panconesi, A., and Srinivasan, A. Randomized distributed edge coloring via an extension of the chernoff-hoeffding bounds. *SIAM Journal on Computing* (1997), 26:350–368.

[16] Papadimitriou, C. H., and Vempala, S. On the approximability of the traveling salesman problem. *Combinatorica* (2006), 26(1):101–120.

[17] Papadimitriou, C. H., and Yannakakis, M. The traveling salesman problem with distances one and two. *Math. Oper. Res.* (1993), 18(1):1–11.

[18] Schrijver, A. Combinatorial optimization. *vol. 2 of Algorithms and Combinatorics* (2003).

[19] Svensson, O. Approximating ATSP by relaxing connectivity. *In: Proceedings of FOCS* (2015).

[20] Svensson, O., Tarnawski, J., and Végh, L. A. Constant factor approximation for ATSP with two edge weights. *In: Proceedings of IPCO* (2016), pages 226–237.