

An Improved Approximation Algorithm for TSP in the Half Integral Case

Anna R. Karlin
karlin@cs.washington.edu
University of Washington
USA

Nathan Klein
nwklein@cs.washington.edu
University of Washington
USA

Shayan Oveis Gharan
shayan@cs.washington.edu
University of Washington
USA

ABSTRACT

We design a 1.49993-approximation algorithm for the metric traveling salesperson problem (TSP) for instances in which an optimal solution to the subtour linear programming relaxation is half-integral. These instances received significant attention over the last decade due to a conjecture of Schalekamp, Williamson and van Zuylen stating that half-integral LP solutions have the largest integrality gap over all fractional solutions. So, if the conjecture of Schalekamp et al. holds true, our result shows that the integrality gap of the subtour polytope is bounded away from $3/2$.

CCS CONCEPTS

• Theory of computation → Routing and network design problems.

KEYWORDS

Traveling Salesman Problem, Approximation Algorithms, Randomized Rounding, Max Entropy, Strongly Rayleigh, Cactus Representation

ACM Reference Format:

Anna R. Karlin, Nathan Klein, and Shayan Oveis Gharan. 2020. An Improved Approximation Algorithm for TSP in the Half Integral Case. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC '20)*, June 22–26, 2020, Chicago, IL, USA. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3357713.3384273>

1 INTRODUCTION

In an instance of the traveling salesperson problem (TSP) we are given a set of n cities along with their pairwise symmetric distances. The goal is to find a Hamiltonian cycle of minimum cost. In the metric TSP problem, which we study here, the distances satisfy the triangle inequality. Therefore the problem is equivalent to finding a closed Eulerian connected walk of minimum cost. It is NP-hard to approximate TSP with a factor better than $\frac{123}{122}$ [KLS13]. A classical algorithm of Christofides [Chr76] from 1976 gives a $\frac{3}{2}$ -approximation

algorithm for TSP and remains the best known approximation algorithm for the general version of the problem despite significant work [Wol80, SW90, BP91, Goe95, CV00, GLS05, BEM10, BC11, SWvZ12].

Polynomial-time approximation schemes (PTAS) have been found for Euclidean [Aro96, Mit99], planar [GKP95, AGK⁺98, Kle05], and low-genus metric [DHM07] instances. The case of graph metrics has received significant attention. In 2011, the third author, Saberi, and Singh [OSS11] found a $\frac{3}{2} - \epsilon_0$ approximation for this case. Mömke and Svensson [MS11] then obtained a combinatorial algorithm for graphic TSP with an approximation ratio of 1.461. This approximation ratio was later improved by Mucha [Muc12] to $\frac{13}{9} \approx 1.444$, and by Sebő and Vygen [SV12] to 1.4.

In this paper we study metric TSP for instances in which an optimal solution to the subtour linear programming relaxation is half-integral, i.e., when the optimal solution x satisfies $x_e \in \{0, 1/2, 1\}$ for all edges e . These instances are conjectured to be “the hardest” instances of TSP by Schalekamp, Williamson and van Zuylen.

Conjecture 1.1 ([SWvZ13]). *The integrality gap for the subtour LP is attained on half-integral vertices of the polytope.*

The above conjecture is motivated by the fact that the worst known integrality gap examples of TSP (and TSP-path) are half-integral. Furthermore, as shown in [SWvZ13] the worst case ratio of 2-matchings to optimal solutions of the subtour-LP is attained by half-integral instances. Very little progress has been made on half integral instances even though they have been a subject of study for decades, [CR98, BC11, BS17, HNR17, HN19].

Our main result is the following theorem:

Theorem 1.2. *There is a randomized polynomial time algorithm which when given any half-integral fractional solution x of the subtour LP produces a tour with expected cost at most 1.49993 times the cost of x .*

So, if Conjecture 1.1 holds affirmatively, the above theorem implies that the integrality gap of the subtour-LP is at most 1.49993. Our result also has a direct consequence to the minimum cost 2-edge connected subgraph problem. About 20 years ago, Carr and Ravi [CR98] showed that the integrality gap of the half-integral LP solutions of the min cost 2-edge connected subgraph problem is at most $4/3$. But, to the best of our knowledge, no polynomial time algorithm with an approximate factor better than $3/2$ is known. Our theorem also implies a 1.49993 approximation algorithm for half-integral

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

STOC '20, June 22–26, 2020, Chicago, IL, USA

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-6979-4/20/06...\$15.00

<https://doi.org/10.1145/3357713.3384273>

LP solutions of the minimum cost 2-edge connected subgraph problem.

There are two well known lines of attack to metric TSP: (i) Start from an optimal Eulerian subgraph and make it connected by adding new edges while preserving the parity of the degrees, or (ii) Start with an optimal connected subgraph, then correct the parities of vertex degrees by adding the minimum cost Eulerian augmentation.

Here, we take the second approach. It turns out that in approach (ii) the minimum cost Eulerian augmentation of any connected graph is simply the min cost matching on odd degree vertices which can be computed in polynomial time. So, the main question is how to choose a spanning tree of cost at most OPT such that the cost of the minimum Eulerian augmentation is bounded away from $\text{OPT}/2$. Here we follow the approach initiated in [OSS11]. We sample a random spanning tree that does not cost more than the optimum in expectation. More precisely, we sample from the maximum entropy distribution of spanning trees with marginals equal to the given LP solution, x , and then add the minimum cost matching on odd degree vertices.

It was conjectured in [OSS11] that this algorithm beats Christofides for general metric TSP, but the authors could only justify a variant of this conjecture for graph metrics. To bound the approximation factor for graph metrics, [OSS11] showed that this random spanning tree “locally” looks like a Hamiltonian cycle, e.g., each vertex has degree 2 with constant probability, and, except in some special cases, each pair of vertices have degree 2 simultaneously with constant probability. Roughly speaking, the analysis of [OSS11] is “local” in the sense that it shows that there is a set F of edges with $x(F) \geq \Omega(n)$, such that each $e \in F$ is only contained in a constant number of “local” (near) min cuts and all these (near) min cuts have even number of edges in the random spanning tree with constant probability.

Such a method provably fails for the problem on general metrics since most of the cost of the LP may be concentrated on edges which show up in many (near) min cuts. So, one needs a more “global” analysis technique. In this paper we take the first step towards a global *amortized* analysis. The hard instances of TSP are those where the cost of the LP is dominated by the edges which show up in many (near) min cuts; in this case, there is no hope to show that all such cuts are even simultaneously in a random spanning tree with constant probability. Our high level framework is to build a hierarchy over edges. Roughly speaking, a more “global” edge shows up higher in the hierarchy. When an edge e is even in its highest cuts in the hierarchy, we gain from e at the expense of using descendants of e in the hierarchy to pay for lower cuts of e which may be odd. We then show that the amount e gains when it reduces exceeds by a constant factor the amount it may have to pay to fix cuts containing edges going higher in the hierarchy.

Putting this together, we show that a variant of the max entropy sampling algorithm beats Christofides if the underlying LP solution is half-integral. We expect that many of our techniques can be generalized to apply to LP solutions

that are not half-integral, however the most difficult barrier to overcome seems to be that the structure of near minimum cuts (all cuts in the LP of value within 2 and $2 + \epsilon$ for a fixed $\epsilon > 0$) is more complex than the structure of minimum cuts.

2 OUR ALGORITHM

Before we discuss our algorithm, we need a few definitions and tools. Where V is the set of vertices, let $c : V \times V \rightarrow \mathbb{R}_+$ denote the cost of going from u to v for any $u, v \in V$. For a graph $G = (V, E)$ and a set $S \subseteq V$, we write

$$E(S) := \{\{u, v\} \in E : u, v \in S\},$$

$$\delta(S) := \{\{u, v\} \in E : u \in S, v \notin S\}.$$

For a vector $x : E \rightarrow \mathbb{R}$, and a set $F \subseteq E$, we write $x(F) = \sum_{e \in F} x_e$. For a graph $G = (V, E)$ and $S \subseteq V$, we write $G \setminus S$ to be the graph where S and all edges incident to S are removed, and we write G/S to denote the graph in which S is contracted. For two sets S, T , we write $S \setminus T$ to denote the set difference.

2.1 Held-Karp Relaxation

The following linear program was first formulated by Dantzig, Fulkerson and Johnson [DFJ54] and is known as the subtour elimination polytope or the Held-Karp LP relaxation (see also [HK70]).

$$\begin{aligned} \min_{x} \quad & \sum_{u,v} c(u,v)x_{\{u,v\}} \\ \text{s.t.} \quad & \sum_{u \in S, v \in \bar{S}} x_{\{u,v\}} \geq 2 & \forall S \subsetneq V \\ & \sum_{v \in V} x_{\{u,v\}} = 2 & \forall u \in V \\ & x_{\{u,v\}} \geq 0 & \forall u, v \in V. \end{aligned} \tag{1}$$

Assumption 2.1. *Throughout the paper, we assume that we are given a feasible half-integral solution of the Held-Karp LP, that is, for each $\{u, v\}$, $x_{\{u,v\}} \in \{0, 0.5, 1\}$.*

Remark 2.2. We will often talk about the support graph $G = (V, E)$ of x , replacing any edge of value 1 with two parallel edges. Therefore the number of edges crossing any minimum cut is 4 (corresponding to fractional value 2), and the graph is Eulerian. **Henceforth, any reference to the graph G refers to this support graph.**

In our algorithm we will repeatedly consider subsets S of vertices such that $\delta(S)$ is a min-cut of x .

For any graph $G = (V, E)$, Edmonds [Edm70] gave the following description for the convex hull of spanning tree of a graph G , known as the *spanning tree polytope*.

$$\begin{aligned} z(E) &= n - 1 \\ z(E(S)) &\leq |S| - 1 & \forall S \subseteq V \\ z_e &\geq 0 & \forall e \in E. \end{aligned} \tag{2}$$

Edmonds [Edm70] proved that the extreme point solutions of this polytope are the characteristic vectors of the spanning trees of G .

We formally define tight sets in Section 2.4 but for now assume S is tight if $x(\delta(S)) = 2$. In the half-integral case this corresponds to $|\delta(S)| = 4$.

Fact 2.3. *If in the fractional solution x of the Held-Karp LP, the set S is a tight set, then the restriction of x to edges in $E(S)$, that is, the fractional solution $\{x_e\}_{e \in E(S)}$, is in the spanning tree polytope on $(S, E(S))$.*

PROOF. Since every min-cut has fractional value 2, and every vertex has fractional degree 2 in any Held-Karp solution, we have

$$\sum_{e \in E(S)} x_e = \frac{2|S| - 2}{2} = |S| - 1.$$

For the same reason, the constraint (2) of the spanning tree polytope holds for each $S' \subset S$. $\square$

It follows immediately that:

Fact 2.4. *If S is a tight set w.r.t. x , then a random spanning tree on S selected from a distribution whose marginals are x_e for each $e \in E(S)$ has expected cost $\sum_{e \in E(S)} x_e c_e$.*

2.2 Maximum Entropy Distribution

We say a distribution μ over spanning trees is λ -uniform or maximum entropy if there are nonnegative weights $\lambda : E \rightarrow \mathbb{R}_+$ such that for any tree T ,

$$\mathbb{P}[T] \propto \prod_{e \in T} \lambda_e.$$

Given a point z in the spanning tree polytope, for a connected graph $G = (V, E)$, Asadpour et al. [AGM⁺17] show that there is an efficient algorithm that finds non-negative λ_e 's in a such a way that for every edge $e \in E$ and tree T sampled from μ , $\mathbb{P}[e \in T]$ is (approximately) equal to z_e .

To sample from a distribution on spanning trees, we follow [AGM⁺17, OSS11] and sample spanning trees using a distribution

Theorem 2.5 ([AGM⁺17]). *Let z be a point in the spanning tree polytope of the graph $\tilde{G} = (\tilde{V}, \tilde{E})$. For any $0 < \epsilon$, values λ_e for all $e \in \tilde{E}$ can be found such that the corresponding λ -uniform spanning tree distribution, μ , satisfies*

$$\mathbb{P}_\mu[T] \leq (1 + \epsilon)z_e, \quad \forall e \in E,$$

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

We can now briefly explain the main algorithm of [OSS11]: Given a feasible solution x of subtour-LP, define $z = (1 - 1/n)x$. Then, sample T from a λ -uniform distribution with marginals z and add a min-cost matching on the odd degree vertices of T .

2.3 Min-cost Eulerian Augmentation

Once we have sampled a tree (or, as we shall see later, a tree plus an edge), we will be finding the minimum cost Eulerian augmentation. For this purpose, we use the following characterization of the O -join polytope due to Edmonds and Johnson [EJ73].

Proposition 2.6. *For any graph $G = (V, E)$, cost function $c : E \rightarrow \mathbb{R}_+$, and a set $O \subseteq V$ with an even number of vertices, the minimum weight of an O -join equals the optimum value of the following integral linear program.*

$$\begin{aligned} \min \quad & c(y) \\ \text{s.t.} \quad & y(\delta(S)) \geq 1 \quad \forall S \subseteq V, |S \cap O| \text{ odd} \\ & y_e \geq 0 \quad \forall e \in E \end{aligned} \quad (3)$$

2.4 Description of Algorithm

Our algorithm is a slight modification of the one studied in [OSS11]. Given a feasible solution x of the subtour-LP, without loss of generality, we assume that there exists an edge e^+ such that $x_{e^+} = 1$. If such an edge does not exist, we split a node v into two nodes v_1, v_2 ; connect 2 of the edges out of v to v_1 and the other two to v_2 . Then, we connect v_1 to v_2 with edge e^+ of cost $c(e^+) = 0$ and $x_{e^+} = 1$.

Given such a solution x our algorithm is as follows: Define a fractional spanning tree z where $z_{e^+} = 0$ and $z_e = x_e$ for any $e \neq e^+$. Then, we sample T from the λ -uniform spanning tree distribution μ with marginals z for some $\epsilon = 2^{-n}$ using Theorem 2.5. Define $T = T \cup \{e^+\}$; this gives a **1-tree**. A 1-tree is a union of a spanning tree and an edge. Finally, we add a minimum cost Eulerian augmentation on the odd degree vertices of T . Throughout we let $T^- = T \setminus e^+$. T^- is an actual spanning tree.

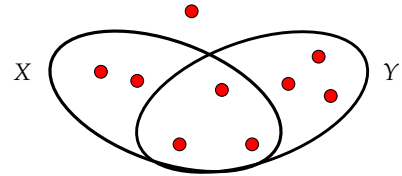


Figure 1: An example of two crossing sets.

There is an equivalent description of the above algorithm. Before discussing this, we need to define three more concepts.

Definition 2.7. *Consider a graph $G = (V, E)$ with min-cuts of value k .*

- Any set $S \subseteq V$ such that $|\delta(S)| = k$ (i.e., its boundary is a min-cut) is called a **tight set**.
- A cut $(S, \bar{S})$ is **proper** if $|S| \geq 2$ and $|\bar{S}| \geq 2$.
- Two sets S and S' **cross** if all of $S \setminus S'$, $S' \setminus S$, $S \cap S'$ and $V \setminus (S \cup S')$ are non-empty.

See Algorithm 1 for an equivalent description of our algorithm, which we will work with throughout the paper. As the equivalence is not fundamental to our proof, we omit the (simple) proof here.

Fact 2.8. *In step 5 of the algorithm, we have $e^+ \notin \delta(S)$.*

PROOF. Say $e^+ \in \delta(S)$, and let $e^+ = \{u, v\}$. Then, since $x_{e^+} = 1$, $\{u, v\}$ is a tight set. It also crosses S (as S is a proper set). That is a contradiction. $\square$

Algorithm 1 Algorithm for Half-Integral TSP

-
- 1: Given a half-integral solution x of the subtour LP, with an edge e^+ with $x_{e^+} = 1$.
 - 2: Let G be the support graph of x .
 - 3: Set $T = \emptyset$ $\triangleright T$ will be a 1-tree
 - 4: **while** there exists a proper tight set of G that is not crossed (by a tight set) **do**
 - 5: Let S be a minimal such set such that $e^+ \notin E(S)$ $\triangleright$
 Note such a set always exists, as $S, \bar{S}$ are both proper tight sets, so one does not have e^+ . In [Fact 2.8](#) we show that $e^+ \notin \delta(S)$.
 - 6: Compute the maximum entropy distribution μ of $E(S)$
 - 7: Sample a tree from μ and add its edges to T
 - 8: Set $G = G/S$ $\triangleright$ Note we never contract e^+ .
 - 9: **end while**
 - 10: Randomly sample a cycle from G (including e^+) and add it to T $\triangleright$ In [Fact 3.3](#) we show G itself is a cycle
 - 11: Compute the minimum O-Join on the odd nodes of T .
 Shortcut it and output the resulting Hamiltonian cycle.
-

A few remarks are in order. By [Fact 2.4](#), $\mathbb{E}[c(T)] = c(x)$ (up to an error of 2^{-n}). Therefore, to prove [Theorem 1.2](#) all we need to do is to bound the expected cost of the O-join by $c(x)(1/2 - \epsilon_0)$ for some $\epsilon_0 > 0$. Also, crucial to our analysis are the independence properties we get from our algorithm, see the following and [Fact 3.9](#).

Fact 2.9. *Any tree chosen from a max-entropy distribution corresponding to a proper tight set S which is not crossed is independent of all other edges of T that we choose in different iterations of the while loop in our algorithm.*

3 TOOLS

In this section we state two main tools that we use in our analysis, namely the cactus representation of minimum cuts and strongly Rayleigh probability distributions. We will conclude this section by giving an overview of our proof.

3.1 Min-Cuts and the Cactus

To understand our algorithm and analysis it is useful to recall the cactus representation [\[FF09\]](#) of the min-cuts of a graph. We briefly recall some basic definitions and the recursive construction of the cactus. We will rely on a number of basic facts about min-cuts. For proofs, see [\[FF09\]](#). Suppose G is a k -edge connected graph.

Fact 3.1. *If two tight sets S and S' cross, then each of $S \setminus S'$, $S' \setminus S$, $S \cap S'$ and $\bar{S} \cup \bar{S}'$ are tight. Moreover, there are no edges from $S \setminus S'$ to $S' \setminus S$, and there are no edges from $S \cap S'$ to $\bar{S} \cup \bar{S}'$.*

Therefore, if two distinct tight sets S and S' cross each other, then $\delta(S) \cap \delta(S') = \emptyset$.

The following fact is especially useful to us, since the support graph of x is 4-edge-connected.

Fact 3.2. *Suppose that every proper mincut is crossed by some other proper mincut. Then k is even and G is a cycle, with $k/2$ parallel edges between each adjacent pair of vertices.*

Fact 3.3. *In step 10 of the algorithm the remaining graph G is a cycle of length at least 3 such that there are exactly two parallel edges between each pair of consecutive vertices.*

PROOF. Let G be the graph which remains after the while loop in the algorithm terminates. By the algorithm, $e^+ = \{u, v\}$ is not contracted yet. G has at least 3 vertices, as otherwise in the last of the while we contracted a set S where $e^+ \in \delta(S)$ which contradicts [Fact 2.8](#). If G has 3 vertices then it must be a cycle. Otherwise, $\{u, v\}$ is a proper tight set in G , and it must be crossed. In this case by [Fact 3.2](#) G is a cycle of length at least 4. $\square$

Definition 3.4 (Cactus Graph). *A loopless and 2-edge connected graph $C = (U, F)$ is a cactus if each edge belongs to exactly one cycle.*

Theorem 3.5 (Dinits, Karzanov, Lomonosov). *Let $G = (V, E)$ be a loopless graph with min-cut size $k \geq 1$. There is a cactus $C = (U, F)$ and a mapping $\phi : V \rightarrow U$ such that the 2-element cuts of C are in one to one correspondence with the min-cuts of G . Equivalently, S is a tight set of G if and only if $\phi(S)$ is a tight set of C .*

Our algorithm can be viewed as essentially constructing a cactus representation of the min-cuts. More precisely, the critical cuts of our algorithm (defined below) are in one to one correspondence with the cycles of the cactus.

In the rest of this section we will more fully explore the interaction between our algorithm and the structure of the cactus representation of minimum cuts. From now on, assume G is the 4-regular support graph of the half-integral LP solution x and that we have executed our algorithm on G .

Critical sets and cuts: A tight set S selected in step 5 of the algorithm is called a **critical set** and the corresponding cut $\delta(S) := E(S, \bar{S})$ is called a **critical cut**. Vertices of G are degenerate critical sets.

There is a natural *hierarchy of critical sets* associated with the execution of the algorithm. The leaves of the hierarchy are vertices of the original graph. If S and S' are critical sets such that S or a contracted version of S is a vertex in S' , then S is a child of S' (respectively S' is the parent of S). If $\tilde{S}$ is an ancestor of S in the hierarchy of critical sets, then we say that $\tilde{S}$ is a **higher** critical set than S (resp. S is a **lower** critical set than $\tilde{S}$). For example, in [Fig. 3](#), critical set F is the parent of A and is higher in the hierarchy than critical sets A , B and C .

The root of the hierarchy is the graph G once we get to step 10 of the algorithm.

Definition 3.6 (Going higher). *An edge e in $\delta(S)$ goes higher if the lowest critical set S' such that $S \subsetneq S'$ satisfies $e \in \delta(S')$.*

Note that by [Fact 2.9](#) any edge going higher is independent of all edges which do not.

Structure of critical cuts: Consider a critical set S chosen in step 5 in the algorithm. We will abuse notation and, at any time during the execution of the algorithm, refer to G with vertex set V as the graph remaining at that time, after contraction of all trees that have been sampled before S is

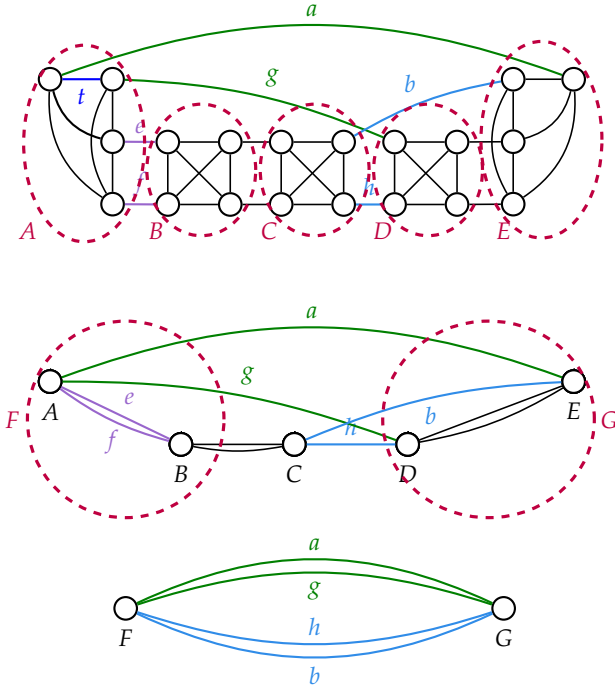


Figure 2: Example execution on a half integral graph. In the first figure, we visualize five tree operations in parallel, which we may do since all these tight sets have size 4 and are not crossed by other tight sets. Similarly in the second figure we do two operations in parallel. In the final step, a cycle is chosen by picking two edges at random. A, B, C, D, E are all “degree cuts” whereas F and G are both “cycle cuts.” t is an example top edge (as are all edges picked in the first graph). a, g and b, h are cycle partners with respect to the cut F . e, f are companions.

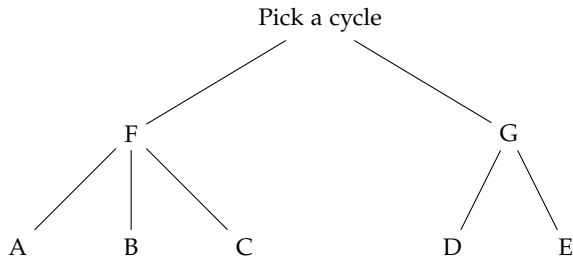


Figure 3: An execution tree on this graph.

considered. Consider the graph $G' := G/V \setminus S$ and let w be the contracted vertex representing $V \setminus S$. There are two possibilities for the structure of G' :

- Case 1: There are no proper min-cuts inside S . In this case, we call $\delta(S)$ a **degree cut**. In Fig. 2, A, B, C, D, E are all degree cuts.¹

¹ These cuts correspond to cycles of length two in the cactus.

- Case 2: There is a proper min-cut $(S_0, \bar{S}_0)$ such that $S_0 \subsetneq S$. In this case, it (and every other proper min-cut inside S) is crossed by some other min-cut (or would be more minimal than S).

It follows that in G' , every proper mincut is crossed by some other proper mincut and therefore, by Fact 3.2, the graph is a cycle with two edges between each pair of adjacent vertices in the cycle. In this case, we call $\delta(S)$ a **cycle cut**. For example, F and G in Fig. 2 are cycle cuts.

We divide the 4 edges from w into two pairs, such that each pair share an endpoint inside S . We call each such pair **cycle partners** with respect to $\delta(S)$. Every other pair of edges between two adjacent vertices in the cycle are called **companions**.

For example, in Fig. 2, $\delta(F)$ is a cycle cut and a, g and b, h are cycle partners with respect to $\delta(F)$. e and f are companions.

Cycle cuts correspond to cycles of length 3 or more in the cactus.

Note that every edge has at most one companion but possibly many partners depending on the underlying cactus.

Definition 3.7 (Highest critical cuts). For a vertex u and an edge $e = \{u, v\}$, let $S_{u,e}$ be the highest critical set S such that $u \in S$ and $v \notin S$, and let S_e be the lowest critical set such that both $S_{u,e}$ and $S_{v,e}$ are (contracted) nodes in S_e . Then $\delta(S_{u,e})$ and $\delta(S_{v,e})$ are the highest critical cuts containing e . If the edge e is clear from context, we may drop e in the notation $S_{u,e}$.

Definition 3.8 (Bottom Edge and Top Edges). For an edge e , if S_e is a cycle cut, we say that e is a bottom edge and otherwise it is a top edge.

For example, in Fig. 2, e, f, a, g, b, h are bottom edges (among the labeled edges) and t is a top edge. The following fact is immediate:

Fact 3.9. Companion bottom edges e, f are in or out of T independently of every other edge of T .

Min-cuts containing a particular edge: The set of min-cuts an edge $e = (u, v)$ is on are the following:

- all critical degree cuts $\delta(S)$ such that $e \in \delta(S)$. (This includes the cuts $(u, V \setminus u)$ and $(v, V \setminus v)$.)
- For any set S such that $\delta(S)$ is a critical cycle cut, and e is either in S or on $\delta(S)$, every cut of the cycle that includes the edge e is a min-cut e is on.

It is easy to see that each of the above is a min-cut. To see that there are no others, it suffices to observe by induction that whenever a set S is contracted, we have accounted for all min-cuts in which nodes inside S are partitioned between the two sides of the cut.

Other facts: We end this part by recording the following basic facts about structure of min cuts, and we will use them throughout our proofs.

Fact 3.10. Suppose that S is a critical set. If some (contracted) vertex $v \in S$ has two edges to $w := V \setminus S$, then S is a cycle cut.

PROOF. This is immediate if $|S| = 2$, so suppose that $|S| > 2$. Then v has two edges to w , which has two edges to $S \setminus v$ which has two edges to w . Since $S \setminus v$ is therefore a proper min-cut but was not selected in step 5, it must be crossed by some other set, which, by the earlier discussion of the structure of critical cuts, means that $\delta(S)$ is a cycle cut. $\square$

Fact 3.11. Suppose that S and S' are two distinct tight sets. Then $|\delta(S) \cap \delta(S')| \leq 2$.

PROOF. By contradiction. Suppose that S and S' are both proper min-cuts and have $\delta(S) \cap \delta(S') \geq 3$. Then by Fact 3.1, they do not cross. Therefore it must be that, say, $S \subset S'$. But in this case, if $\delta(S) \cap \delta(S') \geq 3$, since $\delta(S)$ and $\delta(S')$ are both min-cuts, there can only be one edge from S to $S' \setminus S$ and at most one edge from $S' \setminus S$ to $V \setminus (S \cup S')$ which contradicts $\delta(S' \setminus S) \geq 4$. $\square$

Fact 3.12. Suppose that S and S' are two critical sets such that $S \subset S'$. Then if $\delta(S) \cap \delta(S') = 2$, then S' is a cycle cut.

PROOF. Once S is contracted, it has two edges to $V \setminus S'$, and therefore by Fact 3.10 is a cycle cut. $\square$

Fact 3.13. Suppose that $S \subset S'$ are two critical cycle cuts. Then any two edges are cycle partners on at most one of these (cycle) cuts.

PROOF. Suppose not. Then there is a pair of edges e and f that are cycle partners on both. Suppose that g and h are the other pair of cycle partners on $\delta(S)$ and that their endpoint inside S is node u . Then $(S' \setminus S) \cup u$ is a min-cut that crosses S , which is a contradiction to the selection of S . [Essentially this means that in fact there is a larger cycle here.] $\square$

Fact 3.14. Say S is a critical set and exactly two edges of $\delta(S)$ are bottom edges that do not go higher. Then the other two edges of $\delta(S)$ must go higher.

PROOF. Say $\delta(S) = \{a, b, c, d\}$ and suppose a, b are bottom edges that do not go higher. Say S' is the parent of S in the hierarchy of critical cuts. This implies that $\delta(S')$ is a cycle cut. So, a, b are companions in this cycle. This implies that either c, d are also companions or they are cycle partners in $\delta(S')$. $\square$

3.2 Strongly Rayleigh Distributions

Let $\mathcal{B}_E$ be the set of all probability measures on the Boolean algebra 2^E . Let $\mu \in \mathcal{B}_E$. The generating polynomial g_μ of μ is defined as follows:

$$g_\mu(y) := \sum_{S \subseteq E} \mu(S) \prod_{e \in S} y_e.$$

We say μ is a strongly Rayleigh distribution if $g_\mu \neq 0$ over all $\{y_e\}_{e \in E} \in \mathbb{C}^E$ where $\text{Im}(y_e) > 0$ for all $e \in E$. Strongly Rayleigh (SR) distributions were defined in [BBL09] where it was shown any λ -uniform spanning tree distribution is strongly Rayleigh. In this subsection we recall several properties of SR distributions proved in [BBL09, OSS11] which will be useful to us.

Closure Operations. Strongly Rayleigh distributions are closed under the following operations.

- **Projection.** For any $\mu \in \mathcal{B}_E$, and any $F \subseteq E$, the projection of μ onto F is the measure μ_F where for any $A \subseteq F$,

$$\mu|_F(A) = \sum_{S: S \cap F = A} \mu(S).$$

- **Conditioning.** For any $e \in E$, $\{\mu|e \text{ out}\}$ and $\{\mu|e \text{ in}\}$.

Negative Dependence Properties. An increasing function $f : 2^E \rightarrow \mathbb{R}$, is a function where for any $A \subseteq B \subseteq E$, we have $f(A) \leq f(B)$. For example, if E is the set of edges of a graph G , then the existence of a Hamiltonian cycle is an increasing function, and the 3-colorability of G is a decreasing function.

Definition 3.15 (Negative Association). A measure $\mu \in \mathcal{B}_E$ is negatively associated or NA if for any increasing functions $f, g : 2^E \rightarrow \mathbb{R}$, that depend on disjoint sets of edges,

$$\mathbb{E}_\mu[f] \cdot \mathbb{E}_\mu[g] \geq \mathbb{E}_\mu[f \cdot g]$$

It is shown in [BBL09, FM92] that strongly Rayleigh measures are negatively associated.

Let μ be a strongly Rayleigh measure on edges of G . For a set A , let

$$X_A = |A \cap S|$$

be the random variable indicating the number of edges in A chosen in a random sample S . The following facts immediately follow from the negative association property and the fact that any tree has exactly $n - 1$ edges, see [OSS11] for more details.

Fact 3.16. If μ is a λ -uniform spanning tree distribution on $G = (V, E)$, then for any $S \subseteq E$, $p \in \mathbb{R}$

- (1) If $e \notin S$, then $\mathbb{E}_\mu[X_e | X_S \geq p] \leq \mathbb{E}_\mu[X_e]$ and $\mathbb{E}_\mu[X_e | X_S \leq p] \geq \mathbb{E}_\mu[X_e]$
- (2) If $e \in S$, then $\mathbb{E}_\mu[X_e | X_S \geq p] \geq \mathbb{E}_\mu[X_e]$ and $\mathbb{E}_\mu[X_e | X_S \leq p] \leq \mathbb{E}_\mu[X_e]$.

Fact 3.17. For any set of edges S and $e \notin S$,

$$\mathbb{E}_\mu[X_S] \leq \mathbb{E}_\mu[X_S | X_e = 0] \leq \mathbb{E}_\mu[X_S] + x_e \quad (4)$$

and

$$\mathbb{E}_\mu[X_S] - x_e \leq \mathbb{E}_\mu[X_S | X_e = 1] \leq \mathbb{E}_\mu[X_S]. \quad (5)$$

Lemma 3.18. Let $S = \{e_1, \dots, e_k\}$ be a set of k edges and suppose that $e \notin S$. Then there are $k - 1$ edges in S such that for each edge e_i among these $k - 1$

$$0.25 \leq \mathbb{P}[e_i \in T | e \in T] \leq 0.5.$$

PROOF. By Eq. (5), $\mathbb{P}[e_i | e \in T] \leq 0.5$ for all e_i . Suppose that (after renaming) $\mathbb{P}[e_1 | e \in T]$ and $\mathbb{P}[e_2 | e \in T]$ are the smallest among all $1 \leq i \leq k$. Observe that by Eq. (5) $\mathbb{E}[X_{e_1} + X_{e_2} | e \in T] \geq 0.5$. Therefore, the bigger one is at least 0.25. $\square$

Rank Sequence. The rank sequence of μ is the sequence

$$\mathbb{P}[X_E = 0], \mathbb{P}[X_E = 1], \dots, \mathbb{P}[X_E = m].$$

Let $g_\mu(y)$ be the generating polynomial of μ . The *diagonal specialization* of μ , $\bar{g}(\cdot)$ is a univariate polynomial obtained by treating $g(\cdot)$ as a univariate polynomial (i.e., considering $g(y, y, \dots, y)$). Observe that $\bar{g}(\cdot)$ is the generating polynomial of the rank sequence of μ . If $\bar{g}(c) = 0$ for $c \in \mathbb{C}$, then $g(c, c, \dots, c) = 0$. So, if $g(\cdot)$ is a real stable polynomial then so is $\bar{g}$. Since a univariate polynomial with real coefficients is stable if and only if all of its roots are real, $\bar{g}(\cdot)$ is a polynomial with real roots.

Generating polynomials of probability distributions with real roots are very well studied in the literature (see [Pit97] and references therein). If $\bar{g}(\cdot)$ is a real rooted univariate polynomial of degree m with nonnegative coefficients, then coefficients of $\bar{g}(\cdot)$ correspond to the probability density function of the convolution of a set of m independent Bernoulli random variables (up to a normalization). In other words, there are m independent Bernoulli random variables $B_1, \dots, B_m$ with success probabilities $p_1, \dots, p_m \in [0, 1]$ such that the probability that exactly k variables succeed is the coefficient of y^k in $\bar{g}(\cdot)$.

Fact 3.19 ([BBL09, Pit97]). *The rank sequence of a strongly Rayleigh measure is the probability distribution of the number of successes in m independent trials for some sequence of success probabilities $p_1, \dots, p_m \in [0, 1]$.*

Given this, we can apply the following theorem by Hoeffding [Hoe56], following the approach of [OSS11].

Theorem 3.20 ([Hoe56, Corollary 2.1]). *Let $g : \{0, 1, \dots, m\} \rightarrow \mathbb{R}$ and $0 \leq p \leq m$ for some integer $m \geq 0$. Let $B_1, \dots, B_m$ be m independent Bernoulli random variables with success probabilities $p_1, \dots, p_m$ that minimizes (or maximizes)*

$$\mathbb{E}[g(B_1 + \dots + B_m)]$$

over all distributions in $\mathcal{B}_m(p)$. Then, $p_1, \dots, p_m \in \{0, x, 1\}$ for some $0 < x < 1$.

Lemma 3.21. *Let $S \subseteq E$ with $|S| = 3$. Furthermore, assume that $\mathbb{P}[|S \cap T| \geq 1] = 1$. Then, $\mathbb{P}[|S \cap T| = 1] \geq \frac{1}{2}$ and $\mathbb{P}[|S \cap T| = 2] \geq \frac{3}{8}$.*

PROOF. By Fact 3.19, we can write the rank sequence of $|S \cap T|$ as a sum of 3 independent Bernoullis B_1, B_2, B_3 , and since $\mathbb{P}[|S \cap T| \geq 1] = 1$ we know that for one Bernoulli $p = 1$. Without loss of generality let $p_1 = 1$. Then by Theorem 3.20 we know that $\mathbb{P}[|S \cap T| = 1]$ and $\mathbb{P}[|S \cap T| = 2]$ are minimized when $p_2 = p_3 = \frac{1}{4}$ or $p_2 = \frac{1}{2}$ and $p_3 = 0$. Therefore:

$$\begin{aligned} \mathbb{P}[|S \cap T| = 1] &\geq \min \left\{ \left(\frac{3}{4} \right)^2, \frac{1}{2} \right\} = \frac{1}{2} \\ \mathbb{P}[|S \cap T| = 2] &\geq \min \left\{ 2 \left(\frac{1}{4} \right) \left(\frac{3}{4} \right), \frac{1}{2} \right\} = \frac{3}{8} \end{aligned}$$

□

The following two lemmas are proved using a similar analysis.

Lemma 3.22. *Let $S \subseteq E$ with $|S| = 2$. Let $\frac{1}{2} \leq \mathbb{E}[|S \cap T|] \leq \frac{3}{2}$. Then $\mathbb{P}[|S \cap T| = 1] \geq \frac{3}{8}$.*

Lemma 3.23. *For a min-cut C , $\mathbb{P}[|T \cap C| \text{ even}] \geq 13/27$.*

Lemma 3.24. *Let $S_1, S_2 \subseteq E$ with $|S_1 \cap S_2| = \emptyset$. Let $|S_1| = |S_2| = 2$, or equivalently $\mathbb{E}[|S_1 \cap T|] = \mathbb{E}[|S_2 \cap T|] = 1$. Then $\mathbb{P}[|S_1 \cap T| = 1 \wedge |S_2 \cap T| = 1] \geq \frac{3}{16}$.*

PROOF. Let $S_1 = \{e, f\}$. Then condition on $e \in T$: this occurs with probability $\frac{1}{2}$. By Fact 3.16 we have

$$\mathbb{E}[|f \cap T| \mid e \in T] \leq \frac{1}{2}$$

Then condition on $f \notin T$. Given the above, this happens with probability at least $\frac{1}{2}$. Similarly consider the event $e \notin T$ and $f \in T$. One of these occurs with probability $\frac{1}{2}$. Therefore, in either event we have:

$$\frac{1}{2} \leq \mathbb{E}[|S_2 \cap T|] \leq \frac{3}{2}$$

And now by Lemma 3.22 both events occur simultaneously with probability at least $\frac{3}{16}$. □

4 OVERVIEW OF ANALYSIS

As already mentioned, our algorithm consists of two steps: sample a 1-tree T , and then construct an optimal O -join for the odd degree vertices in the 1-tree.

Given a feasible LP solution x , the choice $y_e = x_e/2$ for each edge $e \in E$ (which gives $y_e := 1/4$ in the half integral case), yields an O -join solution of total cost at most $OPT/2$. However, this is essentially Christofides' algorithm and guarantees only a $3/2$ approximation.

The key to improving on this is the observation that constraint (3) in the O -join LP is not binding if the intersection of the cut $\delta(S)$ with the tree is even.

Definition 4.1 (Even cuts). *A cut $\delta(S)$ is **even** in T (or simply "**even**" when T is understood) if $|T \cap \delta(S)|$ is even.*

Thus, for every edge e with the property that every min-cut that e is on is even, we can reduce y_e to $1/6$, since every non-min-cut has at least 6 edges, and therefore this guarantees that constraint (3) remains satisfied everywhere. This is the gist of the approach taken in [OSS11].

Suppose that there are multiple "good" edges e with the property that every min-cut they are on is even, say with probability at least p (over the randomness in the selection of T). Then for those outcomes T in which e has this property, we could set $y_e := 1/6$ and satisfy the O -join constraints. This would save us $\frac{1}{12}c(e)$ on every such edge e (the reduction from $1/4$ to $1/6$) and thereby guarantee a reduction in the cost of the O -join solution of e "good" $\frac{p}{12}c(e)$.

Unfortunately, in general, it is not possible to argue that every min-cut an edge is on is even simultaneously in T with constant probability. So, we will use a careful charging scheme.

Definition 4.2 (Last Cuts). *For an edge e , the last cuts of e are the only (two) min-cuts containing e and edges going higher in the graph right before contracting S_e .*

Observe that the last cuts of a top edge are critical cuts, but the last cuts of bottom edges are not critical.

Definition 4.3 (Even at Last). *For an edge e we say e is even at last if the two last cuts of e are even.*

Equivalently, if e is a bottom edge, we say e is even at last if all the min cuts containing e on the cycle defined by the graph consisting of S_e with $V \setminus S_e$ contracted are even. Otherwise, if $e = \{u, v\}$ is a top edge, then it is even at last if the critical cuts S_u, S_v are even simultaneously.

Fact 4.4. *If a bottom edge e is even at last, then all (bottom) edges f where $S_f = S_e$ are even at last.*

PROOF. Since $\delta(S_e)$ is a cycle cut, the edges inside S_e form a path, and thus, exactly one edge between each pair of (possibly contracted) vertices inside S_e is selected as part of the tree on S_e chosen in step 6 of the algorithm. If, e is even at last, we must have exactly one of each pair of cycle partners on $\delta(S)$ is in T ; therefore, every pair of adjacent nodes in the cycle have one edge connecting them in the tree. So, all cuts on the cycle have exactly two edges in T . This implies every bottom edge of this cycle is even at last. $\square$

Remark 4.5. By [Fact 3.2](#), the companion of every bottom edge e has exactly the same pair of last cuts as e .

Definition 4.6 (Good Edges). *An edge $e = (u, v)$ is **good** if it is even at last with probability at least p for some constant $p > 0$.*

Instead of proving that all min-cuts that a single edge is on are even, we will instead prove that every minimum cut contains at least one good edge. Each good edge e will then be responsible for its last two cuts. This will allow edges to be reduced when they are even at last, as all cuts lower in the hierarchy are handled by other edges.

Theorem 4.7. *There is a universal constant $p \geq 1/27$ such that every every min-cut has at least one good edge.*

The proof of the above theorem together with some strengthened statements will be in [Section 5](#). The proof mainly exploits properties of strongly Rayleigh distributions.

As we hinted at, we will reduce the value of y_e to $1/6$ whenever an edge e is even at last. However, since e may also be on many other lower min-cuts, if we reduce y_e , the solution may not be feasible ((3) may be violated) as the lower min-cuts may be odd. To handle any lower min-cut C that e is on, we show that, conditioned on e being even at last, the probability that C is also even is at least q for some $q \geq \Omega(1)$. Therefore, we only need to worry about the lower cuts with probability $1 - q$ each. In the bad event that a lower cut C is odd, we will need to fix the solution to guarantee that (3) still holds: our approach is to split the deficit introduced in the O -Join constraint for C among the good edges that do not go higher (see [Definition 3.6](#)). We then simply show that in expectation each edge gains. This part of the proof heavily exploits the properties of cactus representation of the min-cuts that we discussed above.

We note that [Theorem 4.7](#) on its own is not enough to run our charging argument; so, we need a slightly stronger

version. In particular, in some cuts we may need to have two or three good edges.

5 PROBABILISTIC LEMMAS

In this section, we present three probabilistic lemmas which show that in every min-cut there is at least one good edge, (and in some there are even more). This immediately proves [Theorem 4.7](#).

Note the last cycle that we choose in step 10 of the algorithm has all edges even at last so we don't need to address it in this section. Furthermore, by [Fact 2.8](#), e^+ does not belong to any critical cut.

Lemma 5.1 (Bottom Edge Lemma). *Suppose that $e = (u, v)$ is a bottom edge. Then e is good (where $p \geq 3/16$).*

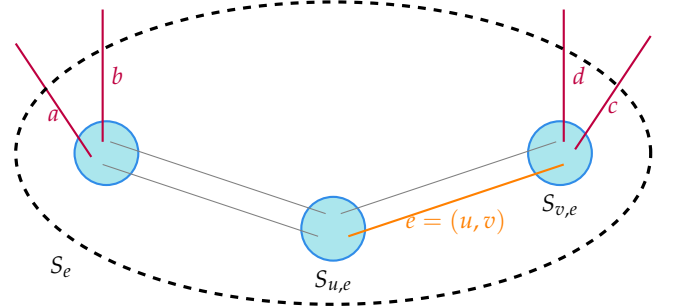


Figure 4: Illustration of edges in Bottom Edge Lemma.

PROOF. If e is a bottom edge then $S_e (= S_{(u,v)})$ is a cycle cut. By construction, when a tree on S_e is selected in step 8 of the algorithm, exactly one edge is chosen between every pair of adjacent nodes in $E(S_e)$. So it suffices to consider the edges in $\delta(S_e)$. These divide up into two pairs of cycle partners connecting S_e to $V \setminus S_e$, say $\{a, b\}$ and $\{c, d\}$. (See [Fig. 4](#).) Then by [Lemma 3.24](#), setting $S_1 := \{a, b\}$ and $S_2 := \{c, d\}$, we have $\mathbb{P}[|S_1 \cap T| = 1 \text{ and } |S_2 \cap T| = 1] \geq 3/16$. $\square$

Lemma 5.2 (Top Edge Lemma). *In a critical cut $\delta(S)$ with one edge that goes higher, of the remaining three edges in the cut, at least two are good with $p \geq \frac{1}{16}$.*

PROOF. First, suppose that e is the edge that goes higher from $\delta(S)$, and f, g and h are the other edges in $\delta(S)$.

If the other endpoint of one of these three edges, say S_h , has no edge that goes higher then h is good. (See left side of [Fig. 5](#).) To see this, observe that we can condition on $\delta(S_h)$ being even which by [Lemma 3.23](#) has probability at least $13/27$. Given this event, $|\{f, g, h\} \cap T|$ is either even or odd. In either case, the event $e \in T$ is an independent event that occurs with probability $1/2$. Therefore,

$$\begin{aligned} & \mathbb{P}[|\delta(S_h) \cap T| \text{ even}] \cdot \mathbb{P}[e \text{ makes } \delta(S) \text{ even} \mid |\delta(S_h) \cap T| \text{ even}] \\ & \geq \frac{13}{27} \cdot \frac{1}{2}. \end{aligned}$$

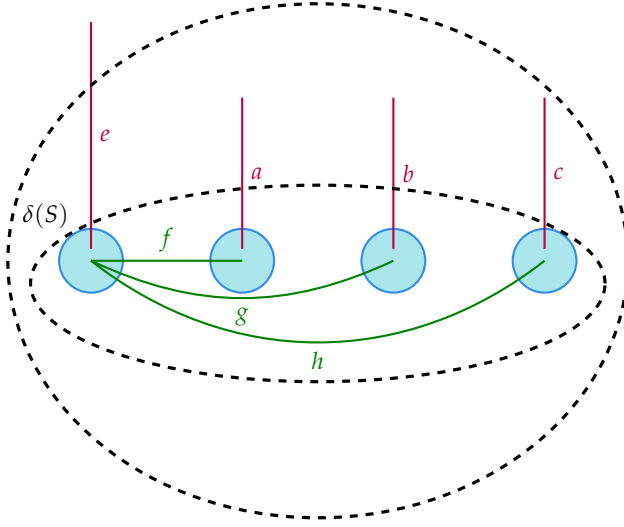


Figure 5: Illustration of top lemma. The figure shows the case where all three of S_f , S_g and S_h have an edge that goes higher (a, b, c respectively). It may be that some number of these nodes do not have an edge going higher.

Therefore, at least two of f, g and h are good if at least two of S_f , S_g and S_h do not have an edge that goes higher.

Consider next the case that, say, S_f and S_g both have an edge that goes higher, but S_h doesn't. As before, h is good. We claim that one of f and g is also good. To see this, since $e \in T$ is independent of $f, g, h \in T$, we observe using Lemma 3.21 that

$$\mathbb{P}[e \in T \text{ and } |\{f, g, h\} \cap T| = 1] \geq \frac{1}{2} \cdot \frac{1}{2}.$$

Next, apply Lemma 3.18 to a and b to conclude that for one of a and b , say a ,

$$\mathbb{P}[a \in T | e \in T] \geq \frac{1}{4}.$$

Therefore, as before, regardless of the even/odd status of $X \cup f$ after conditioning e in and f, g, h to 1, the cut can be fixed by a , meaning we have $p \geq \frac{1}{16}$.

Finally, if all three of S_f , S_g and S_h have an edge that goes higher, then again, condition e in and $\{f, g, h\} \cap T = 1$. Then apply Lemma 3.18 to a, b and c to conclude that for two of them, say a and b , their probability of being in T given that $e \in T$ is at least $1/4$. Therefore, each can fix their corresponding cut ($\delta(S_f)$ for a and $\delta(S_g)$ for b) and both f and g are good. $\square$

Lemma 5.3. *For every critical cut S , there is an edge $e \in \delta(S)$ such that $\mathbb{P}[e \text{ even at last}] \geq 1/27$.*

PROOF. First, if there is a critical set S' ($S' \neq S$) such that $|\delta(S) \cap \delta(S')| = 2$, then by Fact 3.14, there exist at least two bottom edges in $\delta(S)$ and by Lemma 5.1 they are good and we are done. Furthermore, if there is an edge in $\delta(S)$ that goes higher then we are done by Lemma 5.2.

Otherwise, assume S has at most one edge to any other critical set and no edge goes higher. Recall that $T^- = T - e^+$ is always a spanning tree. Let μ' be the conditional measure where $|\delta(S) \cap T^-| = 1$. Observe that in μ' , first we sample a tree² in $G \setminus S$, and then we *independently* add an edge in $\delta(S)$. Therefore, by Fact 3.16, $\mathbb{P}_{\mu'}[e \in T] \leq 1/2$. Thus, for at least one edge $e \in \delta(S)$, we have $1/4 \leq \mathbb{P}_{\mu'}[e \in T] \leq 1/2$. In the special case that $\mathbb{P}[|\delta(S) \cap T| = 1] = 0$, we just let e be an arbitrary edge in $\delta(S)$. Say, $\{e\} = \delta(S) \cap \delta(S')$. It remains to prove that $\mathbb{P}_{\mu}[\delta(S), \delta(S') \text{ even}] > 1/27$ since S, S' are the last cuts of e . Therefore

$$\begin{aligned} & \mathbb{P}[\delta(S), \delta(S') \text{ even}] \\ &= 1 - \mathbb{P}[\delta(S) \text{ or } \delta(S') \text{ odd}] \\ &= 1 - \mathbb{P}[\delta(S) \text{ odd}] - \mathbb{P}[\delta(S') \text{ odd}] + \mathbb{P}[\delta(S), \delta(S') \text{ odd}] \\ &\geq 13/27 - \mathbb{P}[\delta(S) \text{ odd}] + \mathbb{P}[|T \cap \delta(S)| = 1 \wedge \delta(S') \text{ odd}], \end{aligned}$$

by Lemma 3.23. First, note if $\mathbb{P}[|T \cap \delta(S)| = 1] = 0$, then we get that $\mathbb{P}[|T \cap \delta(S)| = 2] = 1$ (since $\mathbb{P}[|\delta(S) \cap T| \geq 1] = 1$, and $\mathbb{E}[|\delta(S) \cap T|] = 2$). So, the RHS is $13/27$ and we are done.

Otherwise,

$$\begin{aligned} & \mathbb{P}[|T \cap \delta(S)| = 1 \wedge \delta(S') \text{ odd}] \\ &= \mathbb{P}[\delta(S') \text{ odd} | |T \cap \delta(S)| = 1] \cdot \mathbb{P}[|T \cap \delta(S)| = 1] \\ &\geq \frac{1}{4} \cdot \mathbb{P}[|T \cap \delta(S)| = 1]. \end{aligned}$$

The inequality is because edge e can make $\delta(S')$ odd by being in/out of the tree and that has probability at least $1/4$. Therefore,

$$\begin{aligned} & \mathbb{P}[\delta(S), \delta(S') \text{ even}] \\ &\geq 13/27 - \mathbb{P}[\delta(S) \text{ odd}] + \frac{1}{4} \mathbb{P}[|T \cap \delta(S)| = 1] \\ &= 13/27 - \frac{3}{4} \mathbb{P}[|T \cap \delta(S)| = 1] - \mathbb{P}[|T \cap \delta(S)| = 3]. \end{aligned}$$

Finally, by Eq. (5), the RHS attains its minimum value when $|T \cap \delta(S)|$ is sum of 4 Bernoullis with success probabilities $1, 1/3, 1/3, 1/3$. $\square$

6 PROOF OF MAIN THEOREM

Recall that every good edge is even on top with probability at least p . The following statement is the main technical result of this section.

Lemma 6.1. *There is a (random) feasible O-join solution such that for every good edge e ,*

$$\mathbb{E}[y_e] \leq 1/4 - p/240,$$

and for every bad edge $y_e = 1/4$ with probability 1.

Before, proving the above statement we use it to prove Theorem 1.2.

²Note T is not a spanning tree in $G \setminus S$, that is why we need to look at T^- .

PROOF OF **THEOREM 1.2**. Consider the trivial O -join solution y' where $y'_e = 1/2$ if e is good and $y'_e = 1/6$ otherwise. Note that this is a valid O -join by **Theorem 4.7**. Now, define $z = \alpha y + (1 - \alpha)y'$ for some α that we choose later. It follows that for any good edge e ,

$$\mathbb{E}[z_e] \leq \alpha \left(\frac{1}{4} - \frac{p}{240} \right) + (1 - \alpha) \frac{1}{2},$$

and for a bad edge f :

$$\mathbb{E}[z_f] = \alpha \frac{1}{4} + (1 - \alpha) \frac{1}{6}$$

So, for $p = \frac{1}{27}$ and $\alpha = \frac{2160}{2161}$ we obtain $\mathbb{E}[z_e] \leq 0.249962$. Since any edge e is chosen in T with probability $1/2$ (up to a 2^{-n} error), we pay at most $1/2 + \mathbb{E}[z_e]$ for any edge e whereas x pays $1/2$. Therefore, we get a $0.749962/0.5$ approximation algorithm. $\square$

So, in the rest of this section we prove **Lemma 6.1**.

O-join construction for good edges: For each good edge e , define B_e to be an independent Bernoulli random variable which is 1 with probability p/p_e , where p is the lower bound on the probability that any good edge is even on top, and p_e is the actual probability that e is even on top. If e, f are bottom edge companions, then we let $B_f = B_e$ (with probability 1). Note that this still makes selection of e, f independent of B_e and any other edge of the graph.

We then construct an O -join solution for each 1-tree T using the following three step process:

- (1) Initialize $y_e := 1/4$ for each edge $e \in E$.
- (2) Next, if e is even at last in T and $B_e = 1$, reduce y_e by $r_e(T)$ where:

$$r_e(T) := \begin{cases} \beta & \text{if } e \text{ is a bottom edge.} \\ \tau_2 & \text{if } e = \{u, v\} \text{ is a good top edge and there are} \\ & \text{exactly 2 good top edges in both } \delta(S_u) \text{ and} \\ & \delta(S_v) \text{ that do not go higher.} \\ \tau_3 & \text{if } e \text{ is a good top edge that does not meet} \\ & \text{the previous criteria.} \end{cases}$$

β, τ_2, τ_3 are parameters we will set later. For now, we just assume $\tau_3 \leq \tau_2 \leq \beta \leq 1/12$. When $r_e(T) > 0$, we say that e is **reduced**.

- (3) On each cut C that is odd, let $\Delta(C) := \sum_{e \in C} r_e(T)$ be the amount by which edges on that cut were reduced in step 2, and let G_C be the set of good edges on C such that C is one of their last cuts. Now, for an edge $e \in G_C$, let C' (and C) be the last cuts of e . Then increase y_e by $\max \left\{ \frac{\Delta(C)}{|G_C|}, \frac{\Delta(C')}{|G_{C'}|} \right\}$. Notice that in this case, since C is one of e 's top cuts, e is not even on top in T and therefore is not reduced in step 2.

By construction, this is a valid O -join solution since on every min-cut we began with at least 4 edges crossing every cut with $y_e = 1/4$ and then guaranteed that every reduction on an odd cut in step 2 was compensated for by a matching increase on that cut in step 3. (The ultimate gain of course will come from the fact that many cuts will be even and hence,

there will not need to be an increase.) All non-min-cuts have at least 6 edges on them, each of value at least $1/6$ (after reduction) and are therefore satisfied in (3).

In the rest of the proof, it is enough to show that for any good edge e , $\mathbb{E}[y_e] \leq 1/4 - p/240$. We complete the proof using the following two lemmas.

Lemma 6.2. *If $\beta \geq 5\tau_2/4$, then for any good top edge $e = \{u, v\}$,*

$$\mathbb{E}[y_e] \leq \frac{1}{4} - p \min\{\tau_2 - \beta/2, \tau_3 - 5\beta/12\}.$$

PROOF. Let $C := \delta(S_u)$ and $C' := \delta(S_v)$. Recall that, since e is a top edge, C and C' are critical cuts. Let H_C (resp. $H_{C'}$) be the good edges in C (resp. C') that go higher. Note that $|H_C|$ and $|H_{C'}|$ is either 0 or 1 by **Fact 3.12**. We consider 3 cases:

Case (i): $|G_C| = |G_{C'}| = 2$. In the worst case, there is an edge, say $f \in H_C$ and an edge $g \in H_{C'}$. If f is a bottom edge, then

$$\mathbb{E}[r_f(T)] = \beta \cdot p \quad \text{and} \quad \mathbb{P}[C \text{ is odd} | f \text{ reduced}] = 1/2$$

since

$$\mathbb{P}[\text{parity of } |f \cap T| = \text{parity of } |(\delta(S_u) \setminus f) \cap T|] = 1/2,$$

by **Fact 3.9**. Therefore, the expected reduction in step 2 on C is at most

$$\mathbb{E}[r_f(T)] \cdot \frac{1}{|G_C|} \cdot \mathbb{P}[C \text{ is odd} | f \text{ reduced}] = \beta p \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{\beta \cdot p}{4}.$$

On the other hand, if f is a top edge, then the expected reduction in step 2 on C is at most

$$\mathbb{E}[r_f(T)] \cdot \frac{1}{|G_C|} \cdot \mathbb{P}[C \text{ is odd} | f \text{ reduced}] \leq \tau_2 p \cdot \frac{1}{2} \cdot \frac{5}{8} \leq \frac{\beta p}{4},$$

where we use **Lemma 3.21** and the fact that edge f is independent of the rest of edges in C to infer that $\mathbb{P}[C \text{ is odd} | f \text{ reduced}] \leq 5/8$. The same reasoning applies to g , so we get

$$\mathbb{E}[y_e] \leq \frac{1}{4} - \tau_2 p + 2 \cdot \frac{\beta p}{4}.$$

Case (ii): $|G_C| \geq 3$ or $|G_{C'}| \geq 3$ (or both). Again, in the worst case, there is an edge, say $f \in H_C$ and an edge $g \in H_{C'}$. By the same reasoning as above, $\mathbb{E}[y_e]$ is largest if f and g are bottom edges. In this case, the same calculation as above gives,

$$\mathbb{E}[r_f(T)] \cdot \frac{1}{|G_C|} \cdot \mathbb{P}[C \text{ is odd} | f \text{ reduced}] \leq \beta p \cdot \frac{1}{|G_C|} \cdot \frac{1}{2},$$

and similarly for g , so

$$\mathbb{E}[y_e] \leq \frac{1}{4} - \tau_3 p + \frac{\beta p}{2} \left(\frac{1}{2} + \frac{1}{3} \right).$$

Case (iii): $|H_C| + |H_{C'}| \leq 1$. In the worst case, $|H_C| = 1$ and $|H_{C'}| = 0$ and $f \in H_C$ is a bottom edge. Note that in this case we may have $G_{C'} = \{e\}$. But the advantage is that we never increase y_e to fix C' since $H_{C'} = \emptyset$. Then,

$$\mathbb{E}[y_e] = \frac{1}{4} - \tau_3 p + \beta p \cdot \frac{1}{|G_C|} \cdot \frac{1}{2} \leq 1/4 - \tau_3 p + \frac{\beta p}{4}.$$

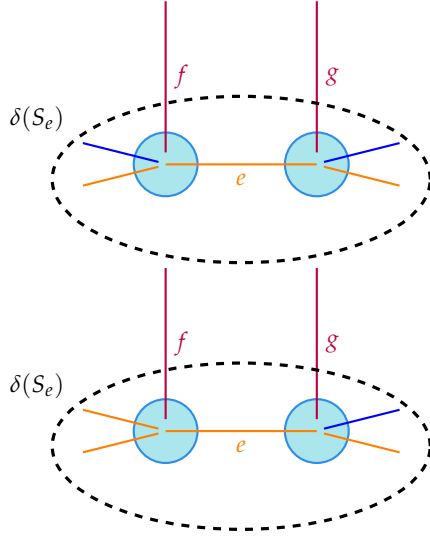


Figure 6: Cases (i) is on the left and case (ii) is on the right. Orange edges are good. On the left, y_e is reduced by τ_2 with probability p and pays half the burden on both sides; on the right y_e is reduced by τ_3 with probability p and pays half the burden on one side and one third of the burden on the other.

Note that if case (iii) does not happen, then by Lemma 5.2 we have $|G_C|, |G_{C'}| \geq 2$. So, either (i) or (ii) will happen. $\square$

Lemma 6.3. *If $3\tau_3 \leq 2\tau_2$, then for any (good) bottom edge e ,*

$$\mathbb{E}[y_e] \leq 1/4 - p \min\{\beta/4, 3\beta/4 - \tau_2, \beta - 4\tau_2/3\}.$$

PROOF. Say f is the companion of e . Let $S = S_e$ and S' be the parent of S in the hierarchy of critical cuts. Say the last cuts of e (and f) are $C = \{e, f, a, b\}$ and $C' = \{e, f, g, h\}$. In other words a, b are partners and g, h are partners. Note that $|G_C| = |G_{C'}| = 2$ because all edges $\{a, b, g, h\}$ go to the higher critical cut S_e .

Case (i): a and g go higher than S . We have $a, g \in \delta(S')$. So, by Fact 3.12, S' is also a cycle cut. This means that b and h are companions and a and g are cycle partner pairs on $\delta(S')$. (See Fig. 7). Edge e has to increase to fix the cuts C, C' whenever a, b, g or h are decreased and the corresponding cut is odd. The expected increase in y_e due to reductions on a, b, g, h divides into two types. We start with b, h : By Fact 3.9 and the fact that $B_b = B_h$, we know that b and h are always reduced at the same time. Furthermore, conditioned on b (and h) being reduced, we have

$$\text{parity of } |T \cap C| = \text{parity of } |T \cap C'|.$$

This is simply because b (and h) are reduced only when they are even at last which implies $|T \cap \{a, g\}| = 1$. So, we can fix the reduction of b, h simultaneously when we increase e (or f). In other words, it is enough to only take into account the expected increase of y_e due to b ,

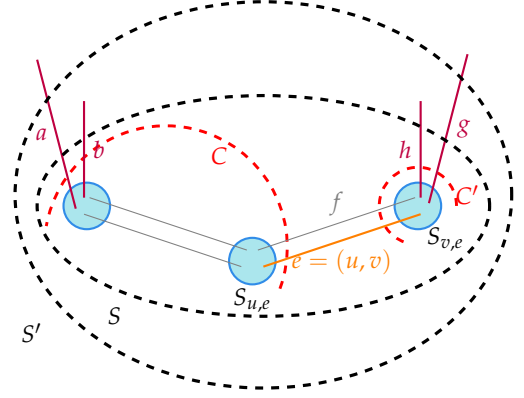


Figure 7: Case (i) of Lemma 6.3.

i.e.,

$$\frac{\mathbb{E}[r_b(T)]}{|G_C|} \cdot \mathbb{P}[|C \cap T| \text{ odd} | b \text{ reduced}] = \beta p \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{\beta p}{4}.$$

Now, we calculate the expected increase due to a, g : We compute the charge due to a and the same will hold for g . Again, by Fact 3.9, b and h are chosen independently of a, g , i.e., $\mathbb{P}[|T \cap C| \text{ odd} | a \text{ reduced}] \leq 1/2$. Therefore, the expected increase due to a is

$$\frac{\mathbb{E}[r_a(T)]}{|G_C|} \cdot \mathbb{P}[|C \cap T| \text{ odd} | a \text{ reduced}] \leq \beta p \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{\beta p}{4}$$

using $\tau_3, \tau_2 \leq \beta$. Therefore, altogether,

$$\mathbb{E}[y_e] \leq \frac{1}{4} - \beta p + 3 \cdot \frac{\beta p}{4}.$$

Case (ii): Only one edge, say a , goes higher than S . In this case, by the same reasoning as above,

$$\frac{\mathbb{E}[r_a(T)]}{|G_C|} \cdot \mathbb{P}[|C \cap T| \text{ is odd} | a \text{ reduced}] \leq \beta p \cdot \frac{1}{2} \cdot \frac{1}{2},$$

since b is independent of a and it can be chosen to correct the parity.

For the remaining three edges $\{b, g, h\}$, by Lemma 5.2 either two or three of b, g and h are good. If two are good, then each has an expected reduction of at most $\tau_2 p$ or three are good and each has an expected reduction of at most $\tau_3 p$.

Therefore, altogether,

$$\begin{aligned} \mathbb{E}[y_e] &\leq \frac{1}{4} - \beta p + \frac{\beta p}{4} + \frac{p}{2} \cdot \max\{2\tau_2, 3\tau_3\} \\ &\leq \frac{1}{4} - \frac{3\beta p}{4} + p\tau_2 \end{aligned}$$

by the assumption of the lemma.

Case (iii): a, g and b, h are companions. This case follows the same analysis as case (i) but gains because in this case a, g are also reduced simultaneously.

Case (iv): No edge goes above S , $\{a, b, g, h\}$ are top edges. Some number of these edges are good; if more than two are good we pay $4\tau_3$ (at most) with probability p

and otherwise we pay $2\tau_2$ (at most) with probability p . Then:

$$\mathbb{E}[y_e] \leq \frac{1}{4} - \beta p + \frac{p}{|G_C|} \max\{2\tau_2, 4\tau_3\} \leq \frac{1}{4} - \beta p + \frac{4\tau_2 p}{3}.$$

To finish the proof we just need to argue that we exhausted all cases. By [Fact 3.11](#), among $\{a, b, g, h\}$ at most two go higher. By [Fact 3.13](#), from each pair of cycle partners, i.e., $\{a, b\}$ or $\{g, h\}$, at most one goes higher. Therefore, if case (i) does not happen, we have at most one that goes higher. If (i), (ii) do not happen, then no edge goes higher. So, by [Fact 3.14](#) either all four edges in $\{a, b, g, h\}$ are bottom edges, i.e., case (iii), or none are bottom edges, i.e., case (iv). $\square$

To finish the proof of [Lemma 6.1](#), let $\beta = 1/12$, $\tau_2 = 7/120$ and $\tau_3 = 7/180$ chosen to satisfy $\tau_3 \leq \tau_2 \leq \beta$, $\beta \geq 5\tau_2/4$ and $3\tau_3 \leq 2\tau_2$. Plugging in these numbers into [Lemma 6.2](#) and [Lemma 6.3](#) we obtain that $\mathbb{E}[y_e] \leq 1/4 - p/240$ for any good edge e as desired.

ACKNOWLEDGEMENTS

The research of the first two authors was supported in part by NSF grant CCF-1813135. The research of the last two authors was supported in part by NSF grants CCF-1552097 and CCF-1907845, and ONR-YIP grant N00014-17-1-2429.

REFERENCES

- [AGK⁺98] Sanjeev Arora, Michelangelo Grigni, David Karger, Philip Klein, and Andrzej Woloszyn. A polynomial-time approximation scheme for weighted planar graph tsp. In *SODA*, pages 33–41, 1998.
- [AGM⁺17] Arash Asadpour, Michel X. Goemans, Aleksander Madry, Shayan Oveis Gharan, and Amin Saberi. An $O(\log n/\log \log n)$ -approximation algorithm for the asymmetric traveling salesman problem. *Operations Research*, 65(4):1043–1061, 2017.
- [Aro96] Sanjeev Arora. Polynomial time approximation schemes for euclidean tsp and other geometric problems. In *FOCS*, pages 2–11, 1996.
- [BBL09] Julius Borcea, Petter Branden, and Thomas M. Liggett. Negative dependence and the geometry of polynomials. *Journal of American Mathematical Society*, 22:521–567, 2009.
- [BC11] Sylvia Boyd and Robert Carr. Finding low cost tsp and 2-matching solutions using certain half-integer subtour vertices. *Discrete Optimization*, 8(4):525–539, 2011.
- [BEM10] S. Boyd and P. Elliott-Magwood. Structure of the extreme points of the subtour elimination polytope of the stsp. In *Combinatorial Optimization and Discrete Algorithms*, volume B23, pages 33–47, Research Institute for Mathematical Sciences, Kyoto University, Kyoto Japan, 2010.
- [BP91] S. C. Boyd and William R. Pulleyblank. Optimizing over the subtour polytope of the travelling salesman problem. *Math. Program.*, 49:163–187, 1991.
- [BS17] Sylvia C. Boyd and András Sebő. The salesman’s improved tours for fundamental classes. In *IPCO*, pages 111–122, 2017.
- [Chr76] Nicos 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.
- [CR98] Robert Carr and R. Ravi. A new bound for the 2-edge connected subgraph problem. In *IPCO*, pages 112–125, 1998.
- [CV00] Robert D. Carr and Santosh Vempala. Towards a $4/3$ approximation for the asymmetric traveling salesman problem. In *SODA*, pages 116–125, 2000.
- [DF54] G Dantzig, R Fulkerson, and S Johnson. Solution of a large-scale traveling-salesman problem. *Operations Research*, 2:393–410, 1954.
- [DHM07] Erik D. Demaine, MohammadTaghi Hajiaghayi, and Bojan Mohar. Approximation algorithms via contraction decomposition. In *SODA*, pages 278–287, 2007.
- [Edm70] Jack Edmonds. Submodular functions, matroids and certain polyhedra. In *Combinatorial Structures and Their Applications*, pages 69–87, New York, NY, USA, 1970. Gordon and Breach.
- [EJ73] Jack Edmonds and Ellis L. Johnson. Matching, euler tours and the chinese postman. *Mathematical Programming*, 5(1):88–124, 1973.
- [FF09] Tamás Fleiner and András Frank. A quick proof for the cactus representation of mincuts. Technical Report QP-2009-03, Egerváry Research Group, Budapest, 2009. www.cs.elte.hu/egres.
- [FM92] Tomás Feder and Milena Mihail. Balanced matroids. In *Proceedings of the twenty-fourth annual ACM symposium on Theory of Computing*, pages 26–38, New York, NY, USA, 1992. ACM.
- [GKP95] M. Grigni, E. Koutsoupias, and C. Papadimitriou. An approximation scheme for planar graph tsp. In *FOCS '95: Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, page 640, Washington, DC, USA, 1995. IEEE Computer Society.
- [GLS05] David Gamarnik, Moshe Lewenstein, and Maxim Sviridenko. An improved upper bound for the tsp in cubic 3-edge-connected graphs. *Oper. Res. Lett.*, 33(5):467–474, September 2005.
- [Goe95] Michel X. Goemans. Worst-case comparison of valid inequalities for the tsp. *MATH. PROG.*, 69:335–349, 1995.
- [HK70] M. Held and R.M. Karp. The traveling salesman problem and minimum spanning trees. *Operations Research*, 18:1138–1162, 1970.
- [HN19] Arash Haddadan and Alantha Newman. Towards improving christofides algorithm for half-integer TSP. [abs/1907.02120](https://arxiv.org/abs/1907.02120), 2019.
- [HNR17] Arash Haddadan, Alantha Newman, and R. Ravi. Cover and conquer: Augmenting decompositions for connectivity problems. *CoRR*, [abs/1707.05387](https://arxiv.org/abs/1707.05387), 2017.
- [Hoe56] W. Hoeffding. On the distribution of the number of successes in independent trials. *Ann. Math. Statist.*, 27:713–721, 1956.
- [Kle05] Philip N. Klein. A linear-time approximation scheme for planar weighted tsp. In *FOCS*, pages 647–657, 2005.
- [KLS13] Marek Karpinski, Michael Lampis, and Richard Schmied. New inapproximability bounds for tsp. In *Algorithms and Computation*, pages 568–578, 2013.
- [Mit99] Joseph SB Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric tsp, k-mst, and related problems. *SIAM Journal on Computing*, 28(4):1298–1309, 1999.
- [MS11] Tobias Moemke and Ola Svensson. Approximating graphic tsp by matchings. In *FOCS*, pages 560–569, 2011.
- [Muc12] M Mucha. $\frac{13}{10}$ -approximation for graphic tsp. In *STACS*, pages 30–41, 2012.
- [OSS11] Shayan Oveis Gharan, Amin Saberi, and Mohit Singh. A randomized rounding approach to the traveling salesman problem. In *FOCS*, pages 550–559, 2011.
- [Pit97] Jim Pitman. Probabilistic bounds on the coefficients of polynomials with only real zeros. *J. Comb. Theory Ser. A*, 77:279–303, February 1997.
- [SV12] András Sebő and Jens Vygen. Shorter tours by nicer ears. *CoRR* [abs/1201.1870](https://arxiv.org/abs/1201.1870), 2012.
- [SW90] D. B. Shmoys and D. P. Williamson. Analyzing the held-karp tsp bound: a monotonicity property with application. *Inf. Process. Lett.*, 35(6):281–285, September 1990.
- [SWvZ12] Frans Schalekamp, David P. Williamson, and Anke van Zuylen. A proof of the boyd-carr conjecture. In *SODA*, pages 1477–1486, 2012.
- [SWvZ13] Frans Schalekamp, David P. Williamson, and Anke van Zuylen. 2-matchings, the traveling salesman problem, and the subtour lp: A proof of the boyd-carr conjecture. *Mathematics of Operations Research*, 39(2):403–417, 2013.
- [Wol80] Laurence A. Wolsey. Heuristic analysis, linear programming and branch and bound. In *Combinatorial Optimization II*, volume 13 of *Mathematical Programming Studies*, pages 121–134. Springer Berlin Heidelberg, 1980.