A Lower Bound for the Max Entropy Algorithm for TSP

Billy Jin[1], Nathan Klein[2], and David P. Williamson[1](✉)

[1] Cornell University, Ithaca, NY, USA
{bzj3,davidpwilliamson}@cornell.edu
[2] Institute for Advanced Study, Princeton, NJ, USA
nklein@ias.edu

Abstract. One of the most famous conjectures in combinatorial optimization is the four-thirds conjecture, which states that the integrality gap of the subtour LP relaxation of the TSP is equal to $\frac{4}{3}$. For 40 years, the best known upper bound was 1.5, due to Wolsey [Wol80]. Recently, Karlin, Klein, and Oveis Gharan [KKO22] showed that the max entropy algorithm for the TSP gives an improved bound of $1.5 - 10^{-36}$. In this paper, we show that the approximation ratio of the max entropy algorithm is at least 1.375, even for graphic TSP. Thus the max entropy algorithm does not appear to be the algorithm that will ultimately resolve the four-thirds conjecture in the affirmative, should that be possible.

1 Introduction

In the traveling salesman problem (TSP), we are given a set of n cities and the costs c_{ij} of traveling from city i to city j for all i, j. The goal of the problem is to find the cheapest tour that visits each city exactly once and returns to its starting point. An instance of the TSP is called *symmetric* if $c_{ij} = c_{ji}$ for all i, j; it is *asymmetric* otherwise. Costs obey the *triangle inequality* (or are *metric*) if $c_{ij} \leq c_{ik} + c_{kj}$ for all i, j, k. All instances we consider will be symmetric and obey the triangle inequality. We treat the problem input as a complete graph $G = (V, E)$, where V is the set of cities, and $c_e = c_{ij}$ for edge $e = \{i, j\}$.

In the mid-1970s, Christofides [Chr76] and Serdyukov [Ser78] each gave a $\frac{3}{2}$-approximation algorithm for the symmetric TSP with triangle inequality. The algorithm computes a minimum-cost spanning tree, and then finds a minimum-cost perfect matching on the odd degree vertices of the tree to compute a connected Eulerian subgraph. Because the edge costs satisfy the triangle inequality, any Eulerian tour of this Eulerian subgraph can be "shortcut" to a tour of no greater cost. Until very recently, this was the best approximation factor known for the symmetric TSP with triangle inequality, although over the last decade substantial progress was made for many special cases and variants of the problem. For example, in *graphic TSP*, the input to the problem is an unweighted connected graph, and the cost of traveling between any two nodes is the number of edges in the shortest path between the two nodes. A sequence of papers led to a 1.4-approximation algorithm for this problem due to Sebő and Vygen [SV14].

J. Vygen and J. Byrka (Eds.): IPCO 2024, LNCS 14679, pp. 238–251, 2024.
https://doi.org/10.1007/978-3-031-59835-7_18

In the past decade, a variation on the Christofides-Serdyukov algorithm has been considered. Its starting point is a well-known linear programming relaxation of the TSP introduced by Dantzig, Fulkerson, and Johnson [DFJ54], sometimes called the *Subtour LP* or the *Held-Karp bound* [HK71]. It is not difficult to show that for any optimal solution x^* of this LP relaxation, $\frac{n-1}{n}x^*$ is a feasible point in the spanning tree polytope. The spanning tree polytope is known to have integer extreme points, and so $\frac{n-1}{n}x^*$ can be decomposed into a convex combination of spanning trees, and the cost of this convex combination is a lower bound on the cost of an optimal tour. In particular, the convex combination can be viewed a distribution over spanning trees such that the expected cost of a spanning tree sampled from this distribution is a lower bound on the cost of an optimal tour. The variation of Christofides-Serdyukov algorithm considered is one that samples a random spanning tree from a distribution on spanning trees given by the convex combination, and then finds a minimum-cost perfect matching on the odd vertices of the tree. This idea was introduced in work of Asadpour et al. [Asa+17] (in the context of the asymmetric TSP) and Oveis Gharan, Saberi, and Singh [OSS11] (for symmetric TSP).

Asadpour et al. [Asa+17] and Oveis Gharan, Saberi, and Singh [OSS11] consider a particular distribution of spanning trees known as the *maximum entropy distribution.* We will call the algorithm that samples from the maximum entropy distribution and then finds a minimum-cost perfect matching on the odd degree vertices of the tree the *maximum entropy algorithm* for the TSP. In a breakthrough result, Karlin, Klein, and Oveis Gharan [KKO21] show that this approximation algorithm has performance ratio better than 3/2, although the amount by which the bound was improved is quite small (approximately 10^{-36}). The achievement of the paper is to show that choosing a random spanning tree from the maximum entropy distribution gives a distribution of odd degree nodes in the spanning tree such that the expected cost of the perfect matching is cheaper (if marginally so) than in the Christofides-Serdyukov analysis. Note that [KKO21] actually choose a tree plus an edge, thus working with x^* instead of $\frac{n-1}{n}x^*$. Since it is cleaner to analyze we will work with this version of the algorithm.

It has long been conjectured that there should be a 4/3-approximation algorithm for the TSP based on rounding the Subtour LP, given other conjectures about the integrality gap of the Subtour LP. The subtour LP is as follows:

$$\begin{aligned} \min \quad & \sum_{e\in E} c_e x_e \\ \text{s.t.} \quad & x(\delta(v)) = 2, && \forall v \in V, \\ & x(\delta(S)) \geq 2, && \forall S \subset V, S \neq \emptyset, \\ & 0 \leq x_e \leq 1, && \forall e \in E, \end{aligned} \tag{1}$$

where $\delta(S)$ is the set of all edges with exactly one endpoint in S and we use the shorthand that $x(F) = \sum_{e\in F} x_e$. The *integrality gap* of an LP relaxation is the worst-case ratio of an optimal integer solution to the linear program to the optimal linear programming solution. Wolsey [Wol80] (and later Shmoys and Williamson [SW90]) showed that the analysis of the Christofides-Seryukov algorithm could be used to show that the integrality gap of the Subtour LP is at most 3/2, and Karlin, Klein, and Oveis Gharan [KKO22] have shown that the integrality gap is at most $\frac{3}{2} - 10^{-36}$. It is well-known that the integrality gap of the Subtour LP is at least 4/3, and it has long been conjectured that the integrality gap is exactly 4/3. However, until the work of Karlin et al., there had been no progress on that conjecture since the work of Wolsey in 1980.

A reasonable question is whether the maximum entropy algorithm is itself a 4/3-approximation algorithm for the TSP; there is no reason to believe that the Karlin et al. [KKO21] analysis is tight. Experimental work by Genova and Williamson [GW17] has shown that the max entropy algorithm produces solutions which are very good in practice, much better than those of the Christofides-Serdyukov algorithm. It does extremely well on instances of graphic TSP, routinely producing solutions within 1% of the value of the optimal solution.

In this paper, we show that the maximum entropy algorithm can produce tours of expected cost strictly greater than 4/3 times the value of the optimal tour (and thus the subtour LP), even for instances of graphic TSP. In particular, we show:

Theorem 1. *There is an infinite family of instances of graphic TSP for which the max entropy algorithm outputs a tour of expected cost at least* $1.375 - o(1)$ *times the cost of the optimum solution.*

The instances are a variation on a family of TSP instances recently introduced in the literature by Boyd and Sebő [BS21] known as *k-donuts* (see Fig. 1). k-donuts have $n = 4k$ vertices, and are known to have an integrality gap of 4/3 under a particular metric. In contrast, we consider k-donuts under the graphic metric, in which case the optimal tour is a Hamiltonian cycle, which has cost n. The objective value of the Subtour LP for graphic k-donuts is also n; thus, these instances have an integrality gap of 1. We show that as the instance size grows, the expected length of the connected Eulerian spanning subgraph found by the max entropy algorithm (using the graphic metric) converges to $1.375n$ from below and thus the ratio of this cost to the value of the LP (and the optimal tour) converges to 1.375. We can further show that there is a bad Eulerian tour of the Eulerian subgraph such that shortcutting the Eulerian tour results in a tour that is still at least 1.375 times the cost of the optimal tour.

It thus appears that the maximum entropy algorithm is not the algorithm that will ultimately resolve the 4/3 conjecture in the affirmative, should that be possible. While this statement depends on the fact that the algorithm uses a particular Eulerian tour, all work in this area of which we are aware considers the ratio of the cost of the connected Eulerian subgraph to the LP value, rather than the ratio of the shortcut tour to the LP value. We also do not know of work which shows that there is always a way to shortcut the subgraph to a tour of significantly cheaper cost. Indeed, it is known that finding the best shortcutting is an NP-hard problem in itself [PV84].

To our knowledge, our result implies that there is currently no candidate 4/3-approximation algorithm for the TSP. Interestingly, earlier work of the authors

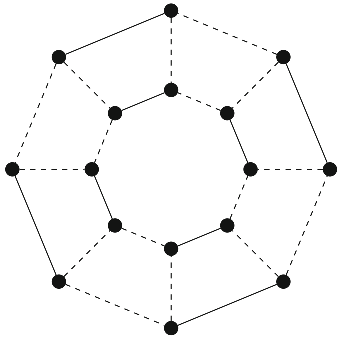

Fig. 1. Our variant on the k-donut for $k = 4$, where k indicates the number of squares of dashed edges. There are $n = 4k$ vertices. The dashed edges have $x_e = \frac{1}{2}$ and the solid edges have $x_e = 1$ in the LP solution. All edges have cost 1, as this is a graphic instance. We will refer to the outer cycle as the *outer ring*, and the inner cycle as the *inner ring*.

[JKW23] gave a 4/3-approximation algorithm for a set of instances of the TSP which includes these k-donuts.

Our work continues a thread of papers showing lower bounds on TSP approximation algorithms. Rosenkrantz, Stearns, and Lewis [RSL77] show that the nearest neighbor algorithm can give a tour of cost $\Omega(\log n)$ times the optimal, and that the nearest and cheapest insertion algorithms can give a tour of cost $2 - \frac{1}{n}$ times the optimal. Cornuéjols and Nemhauser [CN78] show that the Christofides-Serdyukov approximation factor of $\frac{3}{2}$ is essentially tight.

A full version of this paper can be found at https://arxiv.org/abs/2311.01950. Some details are omitted from this version for space reasons.

2 *k*-Donuts

We first formally describe the construction of a graphic k-donut instance, which will consist of $4k$ vertices. The cost function $c_{\{u,v\}}$ is given by the shortest path distance in the following graph.

Definition 1 (k-Donut Graph). *For $k \in \mathbb{Z}_+$, $k \geq 3$, the k-donut is a 3-regular graph consisting of $2k$ "outer" vertices $u_0, \ldots, u_{2k-1}$ and $2k$ "inner" vertices $v_0, \ldots, v_{2k-1}$. For each $0 \leq i \leq 2k-1$, the graph has edges $\{u_i, u_{i+1 \pmod{2k}}\}$, $\{v_i, v_{i+1 \pmod{2k}}\}$, and $\{u_i, v_i\}$. See Fig. 1. We call the cycle of u vertices the* outer ring *and the cycle of v vertices the* inner ring.

For clarity of notation, in the rest of the paper we will omit the "mod $2k$" when indexing the vertices of the k-donut. Thus whenever we write u_j or v_j, it should be taken to mean $u_{j \pmod{2k}}$ or $v_{j \pmod{2k}}$, respectively.

As noted by Boyd and Sebő [BS21], there is a half-integral extreme point solution x of value $4k$ as follows, which we will work with throughout this note.

Let $x_{\{u_i,v_i\}} = 1/2$ for all $0 \le i \le 2k-1$, $x_{\{u_i,u_{i+1}\}} = x_{\{v_i,v_{i+1}\}} = 1/2$ for all even i and $x_{\{u_i,u_{i+1}\}} = x_{\{v_i,v_{i+1}\}} = 1$ for all odd i.[1] In the rest of the paper, we will say a set $S \subseteq V$ is *tight* if $x(\delta(S)) = 2$, and S is *proper* if $2 \le |S| \le |V| - 2$. For a set of edges M, we'll use $c(M) = \sum_{e \in M} c_e$, and for an LP solution x, we let $c(x)$ denote the value of the LP objective function $\sum_{e \in E} c_e x_e$. For $S \subseteq V$, we use $E(S)$ to denote the subset of edges with both endpoints in S.

2.1 The Max Entropy Algorithm on the k-Donut

We now describe the max entropy algorithm, and in particular discuss what it does when specialized to the k-donut. We will work with a description of the max entropy algorithm which is very similar to the one used for half-integral TSP in [KKO20]. In [KKO20], the authors show that without loss of generality there exists an edge e^+ with $x_{e+} = 1$. To sample a 1-tree[2] T, their algorithm iteratively chooses a minimal proper tight set S not containing e^+ which is not crossed by any other tight set, picks a tree from the max entropy distribution on the induced graph $G[S]$, adds its edges to T, and contracts S. [KKO20] shows that if no such set remains, the graph is a cycle, possibly with multiple edges between (contracted) vertices. The algorithm then randomly samples a cycle and adds its edges to T. Finally the algorithm picks a minimum-cost perfect matching M on the odd vertices of T, computes an Eulerian tour on $M \uplus T$, and shortcuts it to a Hamiltonian cycle.[3] We also remark that this algorithm from [KKO20] is equivalent to the one used in [KKO21] as one lets the error measuring the difference between the marginals of the max entropy distribution and the subtour LP solution x go to 0 (see [KKO20,KKO21] for more details).

For ease of exposition, we work with the variant in which we do not use an edge e^+ and instead contract any minimal proper tight set which is not crossed. The two distributions over trees are essentially identical, perhaps with the exception of the edges adjacent to the vertices adjacent to e^+. The performance of the two algorithms on graphic k-donuts can easily be seen to be the same as $k \to \infty$ since one can adjust the matching M with an additional cost of $O(1)$ to simulate any discrepancy between the two tree distributions. We show the essential equivalence of these two versions of the max entropy algorithm in the full version of the paper.

The reason we use this description of the algorithm is that when specialized to k-donuts, Algorithm 1 is very simple and its behavior can be easily understood without using any non-trivial properties of the max entropy algorithm. It first adds the edges with $x_e = 1$ to the 1-tree. Then, it contracts the vertices $\{u_i, u_{i+1}\}$

[1] By slightly perturbing the metric, one could ensure that x is the *only* optimal solution to the LP and thus the solution the max entropy algorithm works with. (Of course then the instance is no longer strictly graphic.).

[2] A spanning tree plus an edge.

[3] Given an Eulerian tour $(t_0, \ldots, t_\ell, t_0)$, we shortcut it to a Hamiltonian cycle by keeping only the first occurrence of every vertex except t_0 (for which we keep the first and last occurrences). Due to the triangle inequality, the resulting Hamiltonian cycle has cost no greater than that of the Eulerian tour.

Algorithm 1. Max Entropy Algorithm (Slight Variant of [KKO20])

Solve for an optimal solution x of the Subtour LP (1).
Let G be the support graph of x.
Set $T = \emptyset$. ▷ T will be a 1-tree
while there exists a proper tight set of G that is not crossed by a proper tight set **do**
  Let S be a minimal such set.
  Compute the maximum entropy distribution μ of $E(S)$ with marginals $x_{|E(S)}$.
  Sample a tree from μ and add its edges to T.
  Set $G = G/S$.
end while
▷ At this point G consists of a single cycle of length at least three, or two vertices with a set of edges F between them with $\sum_{e \in F} x_e = 2$.
if G consists of two vertices **then**
  Randomly sample two edges with replacement, choosing each edge each time with probability $x_e/2$.
else
  Independently sample one edge between each adjacent pair, choosing each edge with probability x_e.
end if
Compute the minimum-cost perfect matching M on the odd degree vertices of T. Compute an Eulerian tour of $T \uplus M$ and shortcut it to return a Hamiltonian cycle.

to a single vertex for all odd i, and does the same for $\{v_i, v_{i+1}\}$ (in other words, it contracts the 1-edges). After that, the minimal proper tight sets consist of pairs of newly contracted vertices $\{u_i, u_{i+1}\}, \{v_i, v_{i+1}\}$ for odd i. Since each of these pairs have two edges set to $1/2$ between them, the algorithm will simply choose one at random for each independently. After contracting these pairs the graph is a cycle. It follows that:

Proposition 1. *On the k-donut, the max entropy algorithm will independently put exactly one edge among every pair $\{\{u_i, v_i\}, \{u_{i+1}, v_{i+1}\}\}$ in T for every odd i and exactly one edge among every pair $\{\{u_i, u_{i+1}\}, \{v_i, v_{i+1}\}\}$ in T for every even i.*

We visualize these pairs in Fig. 2. The following claim is the only property we need in the remainder of the proof:

Proposition 2. *For every pair of vertices (u_i, v_i), $0 \le i \le 2k-1$, exactly one of u_i or v_i will have odd degree in T, each with probability $\frac{1}{2}$. Let O_i indicate if u_i and u_{i+1} have the same parity. Then if $i \neq j$ and i, j have the same parity, then O_i and O_j are independent.*

Proof. We prove the first part of the claim when i is odd; the case where i is even is similar. Since i is odd, the edges $\{u_i, u_{i+1}\}$ and $\{v_i, v_{i+1}\}$ are in T. Then, one of the two edges $\{u_i, v_i\}$ and $\{u_{i+1}, v_{i+1}\}$ is added to T, and regardless of the choice, u_i and v_i so far have the same parity. Finally, one edge in $\{\{u_{i-1}, u_i\}, \{v_{i-1}, v_i\}\}$ is added uniformly at random, which flips the parity of exactly one of u_i, v_i.

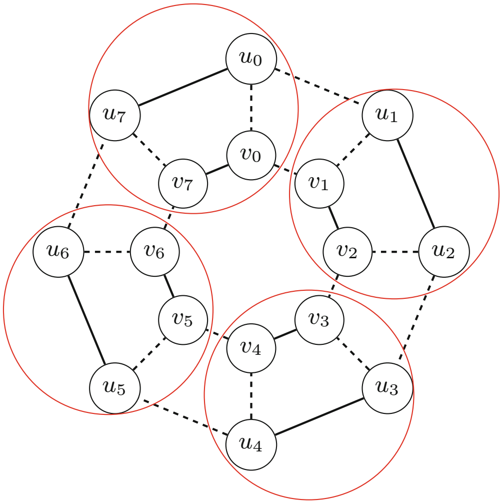

Fig. 2. One edge among the pair of dotted edges inside each red cut will be chosen independently. Then one edge among each pair of dotted edges in the cycle resulting from contracting the red sets will be chosen independently. (Color figure online)

To prove the second part of the claim, we will only do the case that both i and j are odd, as the other case is similar. To slightly simplify the notation we assume $i = 1$ perhaps after a cyclic shift of the indices. Here the event O_1 depends only on the choice of the edges among the pairs $\{\{u_0, u_1\}, \{v_0, v_1\}\}$ and $\{\{u_2, u_3\}, \{v_2, v_3\}\}$; recall the 1-tree picks one edge from each pair, independently and uniformly at random. Similarly, O_j only depends on the independent choices among $\{\{u_{j-1}, u_j\}, \{v_{j-1}, v_j\}\}$ and $\{\{u_{j+1}, u_{j+2}\}, \{v_{j+1}, v_{j+2}\}\}$. The first choice for O_1 is independent of O_j if $j \neq 2k-1$, and the second is independent of O_j if $j \neq 3$. Since $k \geq 3$ by definition of the k-donut, at most one of the independent choices is shared among the two events O_1, O_j. The proof follows by noticing that even after fixing one of the pairs, O_1 remains equally likely to be 0 or 1.

3 Analyzing the Performance of Max Entropy

We now analyze the max entropy algorithm on graphic k-donuts. We first characterize the structure of the min-cost perfect matching on the odd vertices of T. We then use this structure to show that in the limit as $k \to \infty$, the approximation ratio of the max entropy algorithm approaches 1.375 from below.

Proposition 3. *Let T be any 1-tree with the property that for every pair of vertices (u_i, v_i) for $0 \leq i \leq 2k-1$, exactly one of u_i or v_i has odd degree in T. (This is Proposition 2).*

Let $o_0, \ldots, o_{2k-1}$ indicate the odd vertices in T where o_i is the odd vertex in the pair (u_i, v_i). Let M be a minimum-cost perfect matching on the odd vertices of T. Define:

$$M_1 = \{(o_0, o_1), (o_2, o_3), \ldots, (o_{2k-2}, o_{2k-1})\}$$
$$M_2 = \{(o_{2k-1}, o_0), (o_1, o_2), \ldots, (o_{2k-3}, o_{2k-2})\}$$

Then,

$$c(M) = \min\{c(M_1), c(M_2)\}.$$

Proof. We will show a transformation from M to a matching in which every odd vertex o_i is either matched to $o_{i-1 \pmod{2k}}$ or $o_{i+1 \pmod{2k}}$. This completes the proof, since then after fixing (o_0, o_1) or (o_{2k-1}, o_0) the rest of the matching is uniquely determined as M_1 or M_2. During the process, we will ensure the cost of the matching never increases, and to ensure it terminates we will argue that the (non-negative) potential function $\sum_{e=(o_i,o_j)\in M} \min\{|i-j|, 2k-|i-j|\}$ decreases at every step. Note that this potential function is invariant under any reindexing corresponding to a cyclic shift of the indices.

So, suppose M is not yet equal to M_1 or M_2. Then there is some edge $(o_i, o_j) \in M$ such that $j \notin \{i-1, i+1 \pmod{2k}\}$. Without loss of generality (by switching the role of i and j if necessary), suppose $j \in \{i+2, i+3, \ldots, i+k \pmod{2k}\}$. Possibly after a cyclic shift of the indices, we can further assume $i = 0$ and $2 \le j \le k$. Let o_l be the vertex that o_1 is matched to. We consider two cases depending on if $l \le k+1$ or $l > k+1$.

Case 1: $l \le k+1$. In this case, replace the edges $\{\{o_0, o_j\}, \{o_1, o_l\}\}$ with $\{\{o_0, o_1\}, \{o_j, o_l\}\}$. This decreases the potential function, as the edges previously contributed $j + l - 1$ and now contribute $1 + |j - l|$, which is a smaller quantity since $j, l \ge 2$. Moreover this does not increase the cost of the matching: We have $c_{\{o_0,o_1\}} \le 2$ and $c_{\{o_l,o_j\}} \le |j-l|+1$, so the two new edges cost at most $|j-l|+3$. On the other hand, the two old edges cost at least $c_{\{o_0,o_j\}} + c_{\{o_1,o_l\}} \ge j + l - 1$, which is at least $|j - l| + 3$ since $j, l \ge 2$.

Case 2: $l > k+1$. In this case, we replace the edges $\{\{o_0, o_j\}, \{o_1, o_l\}\}$ with $\{\{o_0, o_l\}, \{o_1, o_j\}\}$. This decreases the potential function, as the edges previously contributed $j + (2k - l + 1)$ and now they contribute $(2k - l) + (j - 1)$. Also, the edges previously cost at least $j+(2k-l+1)$, and now cost at most $(2k-l+1)+j$. Thus the cost of the matching did not increase. □

We now analyze the approximation ratio of the max entropy algorithm without shortcutting.

Lemma 1. *If $A = T \uplus M$ is the connected Eulerian subgraph computed by the max entropy algorithm on the k-donut, then*

$$\lim_{k\to\infty} \frac{\mathbb{E}[c(A)]}{c(\mathsf{OPT})} = \lim_{k\to\infty} \frac{\mathbb{E}[c(A)]}{c(x)} = 1.375,$$

where $c(x)$ is the cost of the solution x to the subtour LP.

Proof. We know that the LP value is $4k$. Since the k-donut is Hamiltonian, we also have that the optimal tour has length $4k$. On the other hand, $c(A) = c(T) +$

$c(M)$, where T is the 1-tree and M is the matching. Note that the cost of the 1-tree is always $4k$. On the other hand, we know that $c(M) = \min\{c(M_1), c(M_2)\}$ from the previous claim. Thus, it suffices to reason about the cost of M_1 and M_2. We know that for every i, $c_{\{o_i,o_{i+1}\}} = 2$ with probability $1/2$ and 1 otherwise, using Proposition 2. Thus, the expected cost of each edge in M_1 and M_2 is 1.5. Since each matching consists of k edges, by linearity of expectation, $\mathbb{E}\left[c(M_1)\right] = \mathbb{E}\left[c(M_2)\right] = 1.5k$. This implies $\mathbb{E}\left[c(M)\right] \le 1.5k$. This immediately gives an upper bound on the approximation ratio of $\frac{4k+1.5k}{4k} = 1.375$. In the remainder we prove the lower bound.

For each i, construct a random variable X_i indicating if $c_{\{o_i,o_{i+1}\}} = 2$. By Proposition 2, the variables $\{X_0, X_2, X_4 \dots\}$ are pairwise independent and the variables $\{X_1, X_3, X_5 \dots\}$ are pairwise independent. Thus, for M_1, we have $\mathrm{Var}(\sum_{i=0}^{k-1} X_{2i}) = \sum_{i=0}^{k-1} \mathrm{Var}(X_{2i}) = k/4$, so the standard deviation is $\sigma(\sum_{i=0}^{k-1} X_{2i}) = \sqrt{k}/2$. We define $\mu = \mathbb{E}\left[\sum_{i=0}^{k-1} X_{2i}\right] = k/2$ to be the expected value of $\sum_{i=0}^{k-1} X_{2i}$.

Then, applying Chebyshev's inequality for M_1,

$$\mathbb{P}\left[c(M_1) \ge \left(\frac{3}{2} - \epsilon\right) k\right] = \mathbb{P}\left[\sum_{i=0}^{k-1} X_{2i} \ge \left(\frac{1}{2} - \epsilon\right) k\right]$$
$$\ge 1 - \mathbb{P}\left[\left|\sum_{i=0}^{k-1} X_{2i} - \mu\right| \ge \epsilon k\right] \ge 1 - \frac{1}{4\epsilon^2 k}.$$

Choosing $\epsilon = k^{-1/4}$ and applying a union bound (the same bound applies to M_2), we obtain the chance that both matchings cost at least $\frac{3}{2}k - k^{3/4}$ occurs with probability at least $1 - \frac{1}{2\sqrt{k}}$. Even if the matching has cost 0 on the remaining instances, the expected cost of the matching is therefore at least $(1 - \frac{1}{2\sqrt{k}})(\frac{3}{2}k - k^{3/4}) \ge \frac{3}{2}k - 2k^{3/4}$. Since the cost of the 1-tree is always $4k$, we obtain an expected cost of $\frac{11}{2}k - 2k^{3/4}$ with a ratio of

$$\mathbb{E}\left[c(T \uplus M)\right] = \frac{\frac{11}{2}k - 2k^{3/4}}{\mathsf{OPT}} = \frac{\frac{11}{2}k - 2k^{3/4}}{4k} \ge \frac{11}{8} - k^{-1/4},$$

which goes to $\frac{11}{8}$ as $k \to \infty$. $\square$

4 Shortcutting

So far, we have shown that the expected cost of the connected Eulerian subgraph returned by the max entropy algorithm is 1.375 times that of the optimal tour. However, after shortcutting the Eulerian subgraph to a Hamiltonian cycle, its cost may decrease. Ideally, we would like a lower bound on the cost of the tour after shortcutting. One challenge with this is that the same Eulerian subgraph can be shortcut to different Hamiltonian cycles with different costs, depending on which Eulerian tour is used for the shortcutting. What we will show in this section is that there is always *some* bad Eulerian tour of the connected Eulerian subgraph, whose cost does not go down after shortcutting. We highlight two important aspects of this analysis:

1. In the analysis in Sect. 3, the matching was not required to be either M_1 or M_2. Thus, we lower bounded the cost of the Eulerian subgraph for any procedure that obtains a minimum-cost matching. Here we will require that the matching algorithm always selects M_1 or M_2. From Proposition 3, we know one of these matchings is a candidate for the minimum-cost matching. However, there may be others. Therefore, we only lower bound the shortcutting for a specific choice of the minimum-cost matching.
2. Similar to above, we only lower bound the shortcutting for a specific Eulerian tour of the Eulerian subgraph. Indeed, our lower bound only holds for a small fraction of Eulerian tours.

We remark that the max entropy algorithm as described in e.g. [OSS11,KKO21] does not specify how a minimum-cost matching or an Eulerian tour is generated. Therefore, our lower bound does hold for the general description of the algorithm. Thus, despite the caveats, this section successfully demonstrates our main result: The max entropy algorithm is not a 4/3-approximation algorithm.

In the rest of this section, for space reasons, we will only sketch the argument for why shortcutting does not decrease the cost of the tour. The interested reader is invited to see https://arxiv.org/abs/2311.01950 for the full details. At a high level, we will consider Eulerian graphs resulting from adding M_1 and construct Eulerian tours whose costs do not decrease after shortcutting. We then do the same for the graphs resulting from adding M_2. These two statements together complete the proof.

4.1 Bad Tours on M_1

Recall $M_1 = \{(o_0, o_1), (o_2, o_3), \ldots, (o_{2k-2}, o_{2k-1})\}$, where o_i is the vertex in $\{u_i, v_i\}$ with odd degree in the 1-tree T. This creates graphs of the type seen in Fig. 3. By doing case analysis on which edges are chosen in the tree, it can be shown that the resulting Eulerian subgraph $T \uplus M_1$ consists of a collection of circuits of length 2, 5, or 8 arranged in a circle. Note that doubled edges (circuits of length 2) are never adjacent to one another on this circle, since they are only created between vertices (o_i, o_{i+1}) for i even.

We now describe the problematic tours on such graphs, i.e. Eulerian tours such that the resulting Hamiltonian cycle after shortcutting is no cheaper. For intuition, a reader may want to first consider the bad tour in Fig. 3. We begin at an arbitrary vertex t_0 of degree 2 and pick an edge in the clockwise direction. We now describe the procedure for picking the next vertex t_{k+1} given our current vertex t_k. If t_k has degree 2 (or has degree 4 but is being visited for the second time), there is no choice to make, as only one edge remains. So it is sufficient to describe decisions on vertices t_k of degree 4 visited for the first time. Note that since t_k has degree 4, it is at the intersection of two adjacent circuits. Let

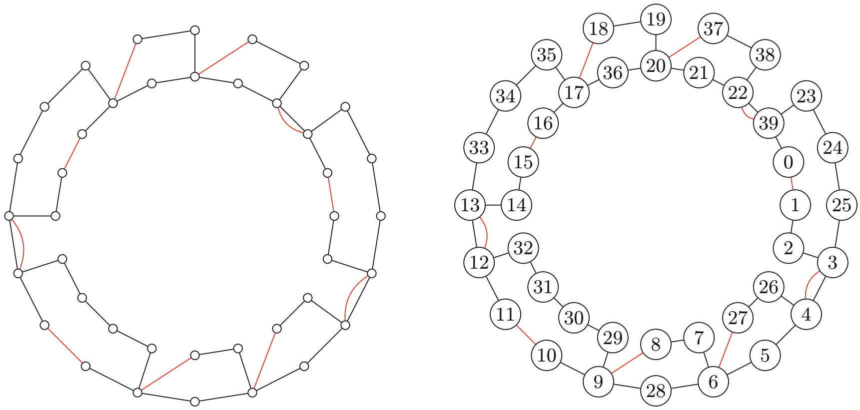

Fig. 3. On the left is an example Eulerian graph when M_1 is added. The graph consists of circuits of length 5 or 8 joined by doubled edges. On the right is the Hamiltonian cycle resulting from shortcutting the adversarial Eulerian tour we construct here, in which we always alternate the side of the circuit we traverse. One can check that every shortcutting operation does not decrease the cost.

C denote the circuit in the clockwise direction adjacent to t_k, and let C' denote the circuit in the counterclockwise direction adjacent to t_k. The next edge to pick is determined by the following two rules:

1. **Never traverse an edge in the counterclockwise direction.** Therefore, if C is a circuit of length 2, we immediately traverse one of its edges.
2. **Alternate the visited side of adjacent circuits.** Otherwise, C has length 5 or 8. For simplicity, suppose t_k is on the outer ring; the case where t_k is on the inner ring is symmetric. Let $e_{\text{outer}} = \{t_k, u\}$ and $e_{\text{inner}} = \{t_k, v\}$ be the two edges in C adjacent to t_k, where u is on the outer ring and v is on the inner ring. Let $e = \{t_j, t_{j+1}\}$ be the previous edge in the tour that was not part of a circuit of length 2. Thus, $j = k-1$, $e = \{t_{k-1}, t_k\}$ if C' is of length 5 or 8, and $j = k-2$, $e = \{t_{k-2}, t_{k-1}\}$ if C' is of length 2. Now, if t_j is on the outer ring, we take e_{inner}. Otherwise, take e_{outer}. The intuition here is that if we visited the inner ring while traversing the last circuit of length greater than 2, we now wish to visit the outer ring, and vice versa.

We call the resulting Eulerian tours B-tours because they are bad for the objective function, even after shortcutting. Indeed, we can prove the following lemma, which together with Lemma 1 demonstrates that the asymptotic cost of the tour after shortcutting is still $11/8$ times that of the optimal solution. We omit the proof of Lemma 2 for space reasons, but we hope the intuition behind the result can be seen by tracing through the example tour in Fig. 3.

Lemma 2. *Shortcutting a B-tour on a graph $T \cup M_1$ to a Hamiltonian cycle does not reduce the cost by more than 3.*

4.2 Bad Tours on M_2

Recall that $M_2 = \{(o_{2k-1}, o_0), (o_1, o_2), \ldots, (o_{2k-3}, o_{2k-2})\}$. It can be shown that in this case, the structure of $T \uplus M_2$ is one large cycle with circuits of length 2 or 3 hanging off to create some vertices of degree 4 on the large cycle; see Fig. 4.

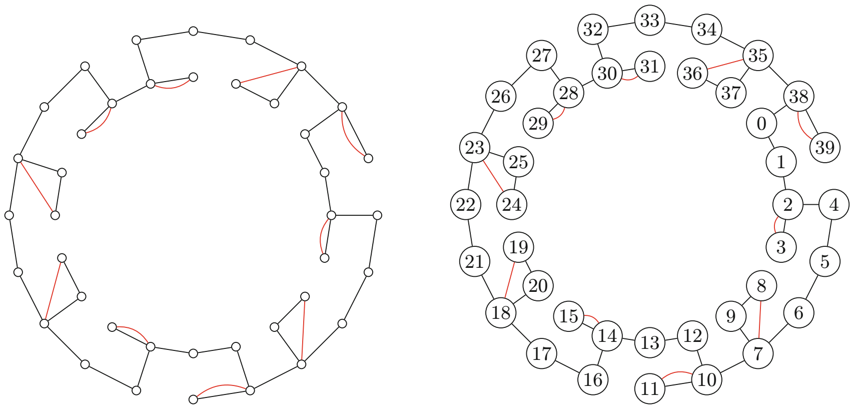

Fig. 4. An example Eulerian graph when M_2 is added. The graph consists of a single long cycle, onto which cycles of length two and three are grafted. As in the case of M_1, one can see that the length of this Hamiltonian cycle is equal to the length of the Eulerian tour that generated it.

We now describe B-tours in this instance. We will start at an arbitrary vertex t_0 on of degree 2 on the large cycle, and traverse an edge in clockwise direction. As before, it suffices to dictate the rules for degree 4 vertices visited for the first time. In this case, the only rule to produce a B-tour is: **traverse the adjacent edge in** M_2. Once again, we omit the proof of Lemma 3 for space reasons, and we hope the intuition behind the result can be seen by tracing through the example tour in Fig. 4.

Lemma 3. *The cost of the Hamiltonian cycle resulting from shortcutting a B-tour on $T \cup M_2$ is equal to the cost of $T \cup M_2$.*

5 Conclusion

We demonstrated that the max entropy algorithm as stated in e.g. [OSS11, KKO21] is not a candidate for a 4/3-approximation algorithm for the TSP. This raises the question: what might be a candidate algorithm? The algorithm in [JKW23] is a 4/3-approximation for half-integral cycle cut instances of the TSP, which include k-donuts as a special case. However, it is not clear if the algorithm can be extended to general TSP instances. One interesting direction is to find a modification of the max entropy algorithm which obtains a 4/3 or better approximation on k-donuts.

It would also be interesting to know whether one can obtain a lower bound for the max entropy algorithm which is larger than 11/8. While we have some intuition based on [KKO20, KKO21] for why the k-donuts are particularly problematic for the max entropy algorithm[4], it would be interesting to know if there are worse examples.

Acknowledgments. The first and third authors were supported in part by NSF grant CCF-2007009. The first author was also supported by NSERC fellowship PGSD3-532673-2019. The second author was supported in part by NSF grants DMS-1926686, DGE-1762114, and CCF-1813135.

References

[Asa+17] Asadpour, A., Goemans, M.X., Mądry, A., Gharan, S.O., Saberi, A.: An $O(\log n/\log\log n)$-approximation algorithm for the asymmetric traveling salesman problem. Oper. Res. **65**, 1043–1061 (2017)

[BS21] Boyd, S., Sebő, A.: The salesman's improved tours for fundamental classes. Math. Program. **186**, 289–307 (2021)

[Chr76] Christofides, N.: Worst case analysis of a new heuristic for the traveling salesman problem. Report 388. Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh (1976)

[CN78] Cornuejols, G., Nemhauser, G.L.: Tight bounds for Christofides' traveling salesman heuristic. Math. Program. **14**, 116–121 (1978)

[DFJ54] Dantzig, G., Fulkerson, R., Johnson, S.: Solution of a large-scale traveling-salesman problem. J. Oper. Res. Soc. Am. **2**(4), 393–410 (1954)

[GW17] Genova, K., Williamson, D.P.: An experimental evaluation of the best-of-many Christofides' algorithm for the traveling salesman problem. Algorithmica **78**, 1109–1130 (2017)

[HK71] Held, M., Karp, R.M.: The traveling-salesman problem and minimum spanning trees. Oper. Res. **18**, 1138–1162 (1971)

[JKW23] Jin, B., Klein, N., Williamson, D.P.: A 4/3-approximation algorithm for half-integral cycle cut instances of the TSP. In: Del Pia, A., Kaibel, V. (eds.) IPCO 2023. LNCS, vol. 13904, pp. 217–230. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-32726-1_16

[KKO20] Karlin, A.R., Klein, N., Gharan, S.O.: An improved approximation algorithm for TSP in the half integral case. In: Proceedings of the 52nd Annual ACM Symposium on the Theory of Computing, pp. 28–39 (2020)

[KKO21] Karlin, A.R., Klein, N., Gharan, S.O.: A (slightly) improved approximation algorithm for metric TSP. In: Proceedings of the 53rd Annual ACM Symposium on the Theory of Computing, pp. 32–45 (2021)

[4] The intuition is as follows. Say an edge $e = (u, v)$ is "good" if the probability that u and v have even degree in the sampled tree is at least a (small) constant and "bad" otherwise. One can show that the ratio (in terms of x weight) of bad edges to good edges in the k-donut is as large as possible. For more details see [KKO21].

[KKO22] Karlin, A., Klein, N., Gharan, S.O.: A (slightly) improved bound on the integrality gap of the subtour LP for TSP. In: 2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS), Los Alamitos, CA, USA, pp. 832–843. IEEE Computer Society (2022)

[OSS11] Gharan, S.O., Saberi, A., Singh, M.: A randomized rounding approach to the traveling salesman problem. In: Proceedings of the 52nd Annual IEEE Symposium on the Foundations of Computer Science, pp. 550–559 (2011)

[PV84] Papadimitriou, C.H., Vazirani, U.V.: On two geometric problems related to the travelling salesman problem. J. Algorithms **5**, 231–246 (1984)

[RSL77] Rosenkrantz, D.J., Stearns, R.E., Lewis, P.M., II.: An analysis of several heuristics for the traveling salesman problem. SIAM J. Comput. **6**, 563–581 (1977)

[Ser78] Serdyukov, A.: On some extremal walks in graphs. Upravlyaemye Sistemy **17**, 76–79 (1978)

[SV14] Sebő, A., Vygen, J.: Shorter tours by nicer ears: 7/5- approximation for the graph-TSP, 3/2 for the path version, and 4/3 for two-edge-connected subgraphs. Combinatorica **34**, 597–629 (2014)

[SW90] Shmoys, D.B., Williamson, D.P.: Analyzing the Held-Karp TSP bound: a monotonicity property with application. Inf. Process. Lett. **35**, 281–285 (1990)

[Wol80] Wolsey, L.A.: Heuristic analysis, linear programming and branch and bound. Math. Program. Study **13**, 121–134 (1980)