

SCALEFREE HARDNESS OF AVERAGE-CASE EUCLIDEAN TSP APPROXIMATION

ALAN FRIEZE AND WESLEY PEGDEN

ABSTRACT. We show that if $P \neq NP$, then a wide class of TSP heuristics fail to approximate the length of the TSP to asymptotic optimality, even for random Euclidean instances. Previously, this result was not even known for *any* heuristics (*greedy, etc.*) used in practice. As an application, we show that when using a heuristic from this class, a natural class of branch-and-bound algorithms takes exponential time to find an optimal tour (again, even on a random point-set), regardless of the particular branching strategy or lower-bound algorithm used.

1. INTRODUCTION

In this manuscript, we prove that if $P \neq NP$, then *scalefree* heuristics cannot find asymptotically optimal approximate solutions even to random instances of the Euclidean TSP. Roughly speaking, *scalefree* heuristics are those which do not work especially hard at small scales. This has two important consequences.

First, it shows rigorously for the first time that several simple heuristics used for the TSP (Nearest Neighbor, Nearest-Insertion, *etc.*) cannot approximate the TSP to asymptotic optimality (even in average-case analysis), since these heuristics are all scalefree. In particular, our result can be seen as a defense of the intricacy of the celebrated polynomial-time approximation schemes of Arora [1] and Mitchell [15] for the Euclidean TSP, as we can show that the simpler algorithms cannot match their performance on sufficiently large random instances.

The second consequence is a new view on the complexity of the Euclidean TSP versus other “actually easy” problems on Euclidean point-sets. Recall that for problems such as determining whether a Minimum Spanning Tree of cost ≤ 1 exists, or finding the shortest path between 2 points from a restricted set of allowable edges, the complexity status in the Euclidean case is unknown, as no algorithm is known to efficiently compare sums of radicals. In particular, it is conceivable that these problems are NP-hard (even if $P \neq NP$) as is the Euclidean TSP [16, 8]. Just as there are (sophisticated) efficient approximation algorithms for the Euclidean TSP, there are (trivial) efficient approximation algorithms for, say, the MST also: simply carry out Kruskal’s algorithm and calculate the length of the spanning tree to some

Date: July 31, 2019.

Research supported in part by NSF grant DMS-1362785.

Research supported in part by NSF grant DMS-1363136.

suitable precision, to obtain a good approximation. Our result allows a rigorous distinction between these types of approximations: Kruskal's algorithm is scalefree in the sense of our paper, and we show that no algorithm with this property can well approximate the Euclidean TSP.

In particular, a surprising message from our result is that it is possible to connect Turing machine complexity to the practical difficulty of a problem for Euclidean point-sets, in spite of the unresolved state of the difficulty of comparing sums of radicals. In particular our result connects traditional worst-case deterministic analysis to *average-case, approximate* analysis of the Euclidean TSP, for a certain class of algorithms.

To motivate our definition of scalefree, we recall a simplification of the dissection algorithm of Karp, which succeeds at efficiently approximating the length of the shortest TSP tour in a random point set to asymptotic optimality.

- (1) Let $s(n) = \lfloor \frac{n^{1/d}}{\log^{1/d}(n)} \rfloor^d$ and divide the hypercube $[0, 1]^d$ into $s(n)$ congruent subcubes $Q_1, \dots, Q_{s(n)}$, letting $X_i = X \cap Q_i$ for each $i = 1, 2, \dots, s(n)$.
- (2) Using the dynamic programming algorithm of Held and Karp [9], find an optimal tour T_i for each set X_i .
- (3) Patch the tours T_i into a tour through all of X .

This algorithm runs in expected time $O(n^2 \log n)$ and finds an asymptotically optimal tour; i.e., the length of T is $(1 + o(1))$ times the optimal length, w.h.p. In some sense, its defining feature is that it works hard (running in exponential time in small sets of vertices) on a small scale, and is more careless on a large scale. Our definition of scalefree is intended to capture algorithms which do not exhibit this kind of behavior.

We will define a *heuristic* H to be a function which takes as input the distance matrix for a point-set X , and outputs a structure (in this paper, either a TSP of X or a list of paths through X) which depends only on comparisons of sums of distances in X .

Formally, given the matrix of distances $d_1, \dots, d_{\binom{n}{2}}$ among points in X , for each choice of coefficients $\xi_i \in \{0, 1, -1\}$ ($1 \leq i \leq \binom{n}{2}$), the sum $\sum \xi_i d_i$ is either negative, zero, or positive. If all $3^{\binom{n}{2}} - 1$ nontrivial sums are nonzero, we say the points of X are in general position. Now, H is a heuristic if for any points in general position, these $3^{\binom{n}{2}} - 1$ sums determine the output of H . We call the heuristic H polynomial-time if it is always possible to determine the output of the function with only polynomially-many comparisons.

Our definition of scalefree requires that the small-scale behavior of a heuristic can be efficiently simulated (usually, by simply running the heuristic on the local data). Care must be taken when making this precise, and our precise definitions of scale-free, together with proofs that commonly used heuristics satisfy the definition, are given in Section 2. Our main theorem is as follows:

Theorem 1.1. *If $H(X)$ is a scalefree polynomial-time TSP heuristic and $P \neq NP$, then there is an $\varepsilon_H > 0$ so that w.h.p., $H(\bar{\mathcal{X}}_n)$ has length greater than $(1 + \varepsilon_H)$ times optimal, where $\bar{\mathcal{X}}_n$ is a discretization of the random set $\mathcal{X}_n \subseteq [0, 1]^d$.*

Note that it is possible that a polynomial-time heuristic as we have defined it may not actually be implementable in polynomial time on a Turing machine, since the calls to the comparison blackbox may not be efficiently implementable, depending on the hardness of comparing sums of radicals. But our theorem applies even just if the number of comparisons is polynomially bounded.

We will also consider the effectiveness of a scalefree heuristic in the context of a Breadth-First Branch and Bound algorithm. Defined precisely in Section 5, this is a branch-and-bound algorithm which branches as a binary tree, which is explored in a breadth-first manner. Branch-and-bound algorithms used in practice do typically satisfy this definition, as they make use of an LP-based lower bound on the TSP, and branch on the binary $\{0, 1\}$ -possibilities for fractional variables.

Theorem 1.2. *Any Breadth-First Branch and Bound algorithm that employs a scalefree heuristic to generate upper bounds w.h.p. requires $e^{\Omega(n)}$ time to complete.*

We close this section by noting that our proof of Theorem 1.1 provides a recipe to attempt to eliminate the $P \neq NP$ assumption for any specific, fixed scalefree heuristic H as follows. Let \mathcal{L} be any decision problem in NP.

Theorem 1.3. *For any scalefree polynomial-time TSP heuristic $H(X)$, there is a polynomial-time algorithm A_H such that for any $L \in \mathcal{L}$, we have:*

- (a) *If $L \notin \mathcal{L}$, then $A_H(L)$ returns false.*
- (b) *If $L \in \mathcal{L}$, then $A_H(L)$ returns either true, or exhibits an $\varepsilon_H > 0$ and a proof that $H(\bar{\mathcal{X}}_n)$ has length greater than $(1 + \varepsilon_H)$ times optimal w.h.p.*

2. SCALEFREENESS OF HEURISTICS

In this section, we will in fact give two definitions of scalefree, the first simpler but more restrictive, and the second more general, but significantly more complicated.

For simplicity of notation, we will let $t = n^{\frac{1}{d}}$ and work with $\mathcal{Y}_n \subseteq [0, t]^d$, defined as the rescaling $t \cdot \mathcal{X}_n$. Observe that with this choice of rescaling, a typical vertex in \mathcal{Y}_n is at distance $\Theta(1)$ away from its nearest neighbor.

Our notion of scalefreeness captures a common property of many simple TSP heuristics: namely, the small-scale behavior of the algorithm can be efficiently simulated (because it is essentially governed by the same rules as the large-scale behavior). To give a precise definition of scalefree, we will use the notion of a polynomial-time path-finding heuristic A_H for the Heuristic H :

Definition 2.1. A polynomial-time path-finding heuristic is an algorithm which has access to a blackbox for making comparisons of sums of distances among the input points, and outputs a list of Hamilton paths through a set of points in general

position in time polynomial in the number of points, given the distance matrix through the point set.

Now we are ready for the simpler of our two definitions of scalefree:

Definition 2.2. Call H *scalefree* if there exists a constant R and a polynomial-time path-finding Heuristic A_H , such that given an input set X for H , the following implication holds for all sets $S_p = B(p, 1) \cap X$:

If:

- (a) The annulus $B(p, R) \setminus B(p, 1)$ around S_p contains no points of X ,
- (b) The points of S_p are in general position,

Then: H traverses S in a path, which belongs to the list produced by A_H when a congruent copy of S is used as the input to A_H .

(Here, $B(p, r)$ is the Euclidean ball of radius r centered at p .)

Perhaps the best motivation for this definition is simply the proofs that it is satisfied on some important examples. We begin with the GREEDY heuristic, which produces a tour by adding, at each step, the shortest edge which would not create a non-Hamilton cycle or a vertex of degree 3. (Note that if there is a tie, the points are not in general position, and we allow GREEDY to have undefined behavior).

Fact 2.3. *The Greedy Heuristic $\text{GREEDY}(X)$ Heuristic is scalefree.*

Proof. For $\text{GREEDY}(X)$, we let $R = 3$, say. A_{GREEDY} is defined simply to be the Greedy heuristic for choosing a Hamilton path, which adds at each stage the shortest edge which would not create a cycle or a vertex of degree 3; as usual, when the input points are not in general position, the behavior of the A_{GREEDY} can be arbitrary.

Given a set $S_p = B(p, 1) \cap X$, the distance between any two points in S_p is smaller than the distance across the annulus $B(p, 3) \setminus B(p, 1)$, and so at some stage of GREEDY on X , S_p will be covered by a path, while no edges cross the annulus. Assuming the points of S_p are in general position, this is the same as the path that will be returned by A_H . \square

This case was particularly simple. For example, A_{GREEDY} actually always outputs just a single path, instead of a list. For the Nearest Neighbor algorithm, things are just slightly more complicated:

Fact 2.4. *The Nearest Neighbor heuristic $\text{NN}(X)$ is scalefree.*

Proof. Recall that the Nearest Neighbor heuristic $\text{NN}(X)$ begins from a distinguished point $x_0 \in X$, and then grows a path by choosing, at each step, the nearest vertex to the current one not already on the path. (In the last step, the endpoints of the Hamilton path are joined to create a tour.) To show that $\text{NN}(X)$ is scalefree,

we again let $R = 3$; we then consider a set $S_p = B(p, 1) \cap X$ such that the annulus $B(p, 3) \setminus B(p, 1)$ contains no points of X .

Observe that as $\text{NN}(X)$ progresses, there will be some first step when the heuristic chooses a point from S_p . Thereafter, since all points in S_p are closer to each other than to points in $X \setminus S_p$, it will exhaust the points in S_p before revisiting $X \setminus S_p$.

Thus, we let $A_{\text{NN}}(S)$ be the algorithm which uses the Nearest Neighbor heuristic to choose a path through S for each choice of the initial vertex x in S . In particular, $A_{\text{NN}}(S)$ will output up to $|S|$ distinct paths through S . (Again, if any ties would be encountered, the points of S are not in general position and so $A_{\text{NN}}(S)$ is allowed to have undefined behavior.) With this choice of $A_{\text{NN}}(S)$, the path taken by $\text{NN}(X)$ through S will be among the list produced by $A_{\text{NN}}(S)$, assuming the points of S are in general position. \square

These examples show the essential character of the notion of a scalefree heuristic. But if we restrict ourselves to Definition 2.2, it would seem we cannot hope to show that some other common heuristics are scalefree. Consider, for example, the Nearest-Insertion heuristic $\text{NEAR-INSERT}(X)$, which begins with a triangle on vertices $x_1, x_2, x_3 \in X$, and then, at each step, grows the existing cycle by finding the vertex nearest to the vertex set of the existing cycle, and inserts the vertex into the existing cycle at minimum cost. If we consider an isolated collection of vertices S_p , predicting the local behavior of $\text{NEAR-INSERT}(X)$ at S_p seems difficult. Consider for example the first step at which the insertion of a vertex $s \in S_p$ occurs, joined to vertices x and y already on the cycle. The next vertex in S_p to be joined to the cycle depends not only on S_p and the vertex s but also potentially on the positions of the vertices x and y , which are outside the “local configuration”.

But it will turn out that this is not really a problem. The theorems we prove essentially will work with any definition of scalefreeness where the implication Definition 2.2 is only required to hold for some reasonable fraction of sets S_p . Since X is itself a random set in our Theorems, this means that we can impose any number of reasonable restrictions on the set S_p in the implication, and still have the resulting notion of scalefree be strong enough for our techniques to give Theorem 1.1.

One way we will do this is by restricting the implication to sets S_p surrounded by special configurations of points. To this end, given $X \subseteq [0, t]^d$ and $Y = \{y_1, y_2, \dots, y_K\} \subseteq [0, 2R]^d$, we say X is (R, ε) -protected by Y at p if

$$X \cap (B(p, R) \setminus B(p, 1)) = Y' \approx_\varepsilon (Y + p) \quad \text{and} \quad Y' \subseteq B(p, \sqrt{R}).$$

Here, $A \approx_\varepsilon B$ means that there is a bijection f from A to B such that for all $a \in A$, $\text{dist}(a, f(a)) < \varepsilon$.

Thus, roughly speaking, a set is (R, ε) -protected by Y at p if it is surrounded by a nearly-congruent copy of Y in an annulus containing no other points of X . Notice that requirement (a) on S_p in Definition 2.2 is simply the requirement that S_p is (R, ε) -protected by $Y = \emptyset$. One way we will generalize Definition 2.2 is simply by allowing the restriction of the implication to subsets S_p of X which are (R, ε) -protected by a fixed set Y_H other than \emptyset .

In particular, the following generalization of scalefreeness adds several restrictions to the sets S_p required to satisfy the implication (thus generalizing the definition) in a way which is tailor-made to be sufficient to easily include common insertion heuristics.

Definition 2.5. H is *scalefree* if there exists R, ε , some *gadget*

$$Y = \{y_1, y_2, \dots, y_s\} \subseteq B(0, \sqrt{R}) \setminus B(0, 1),$$

and a polynomial path-finding heuristic A_H , such that given an input set X for H , there is a bounded-size exceptional set X_0 such that the following implication holds

If: We have:

- (a) X is (R, ε) -protected by Y at p .
- (b) The points of $B(p, R) \cap X$ are in general position.
- (c) $B(p, R) \cap X$ contains no points of the exceptional set X_0 .
- (d) The tour T_H found by H traverses $S_p \cup Y$ in a single path P_H , whose length is within ε of the length of the shortest path through S_p with the same endpoints.
- (e) The vertices $x, y \in X \setminus P_H$ adjacent in T_H to the endpoints of P_H satisfy $\angle xpq, \angle ypq < \varepsilon$, where $q = p + (1, 0)$.

Then: The path P_H by which H traverses S_p belongs to the list produced by A_H when a congruent copy of S_p is used as the input to A_H .

Again we motivate the applicability of this definition by example. Recall that the *Nearest Insertion* heuristic NEAR-INSERT(X) begins, say, with T as the triangle on distinguished vertices $x_1, x_2, x_3 \in X$. (For Definition 2.5, we will choose this triple of vertices as the exceptional set X_0 .) At each step of the algorithm until T is a tour, the Nearest Insertion algorithm finds the vertex z in $X \setminus T$ which is closest to $V(T)$, finds the edge $\{x, y\} \in T$ for which $C = d(x, z) + d(y, z) - d(x, y)$ is minimized, and patches the vertex z in between x and y in the tour, at cost C .

Fact 2.6. NEAR-INSERT(X) is *scalefree*.

Proof. We let X_0 be the set of distinguished vertices $x_1, x_2, x_3 \in X$ which are the vertex-set of the initial tour for NEAR-INSERT(X). We will assume R is a large constant and ε is a small constant, without determining the weakest possible requirements on their magnitudes.

To prove scalefreeness, we first observe that given an S_p which is (R, ε) -protected by the gadget Y , we need only show the implication of Definition 2.5 holds for S_p assuming that there is only ever one insertion of a vertex from $Y \cup S$ at an edge whose endpoints both lie outside of $Y \cup S$; if more than one such insertion occurs in the running of NEAR-INSERT, then the final tour chosen by the Heuristic will not intersect $Y \cup S$ at a path, violating condition (d) in the definition.

We use the gadget Y shown in Figure 1. This consists of, say, 18 equally spaced points at angles $(2k-1)\pi/18$ to the horizontal ($k = 1, \dots, 18$), on the circle of radius \sqrt{R} with center $p + (0, \frac{\sqrt{R}}{2})$.

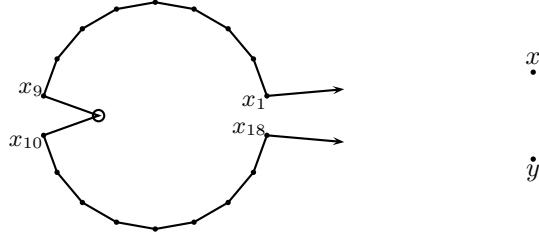


FIGURE 1. (Proving that the Nearest Insertion heuristic is scale-free.) The gadget Y (consisting of the 18 points x_i) prevents the optimal tour from entering directly to a vertex in $S \subseteq B(p, 1)$. The boundary of $B(p, 1)$, which contains all points in the set S_p , is drawn as the tiny circle.

Hypothesis (e) from the implication in Definition 2.5 ensures that the two vertices in T_H adjacent to vertices in $B(R, p)$ are significantly closer to the points x_1, x_{18} than to any other points of $B(R, p)$. Now the path drawn in Figure 1 transits $S \cup Y$ optimally given that it uses these endpoints (assuming the route through S is optimal), and in particular, it follows that for a sufficiently small choice of ε , if the hypothesis (d) and (e) are satisfied in Definition 2.5, then the tour $T_{\text{NEAR-INSERT}}$ traverses $X \cap B(p, R)$ as shown in Figure 1 (and transits S within ε of optimally).

Assuming this is the case, we aim to predict the precise path taken by the heuristic (in particular, in S). Our choice of Y ensures that the subtour constructed by $\text{NEAR-INSERT}(X)$ will contain all of Y before it contains any vertex from S ; thus, we are guaranteed that the two edges leaving $B(p, R)$ are never used for insertions of points in S_p (as there is always a cheaper insertion using closer edges). Since no edges with both endpoints outside of $B(p, R)$ are used for insertions after the first insertion, all insertions of points in S have both endpoints in $B(p, R) \cap X$. In particular, we can use the nearest insertion algorithm locally to determine the resulting path: our path finding Heuristic $A_{\text{NEAR-INSERT}}$ for this case simply begins with a path from x_1 to x_{18} through Y , and uses Nearest Insertion to extend this to a path through all of $B(p, R)$. (Recall that in the case of any ties, the input set is not in general position and so we require nothing of the behavior of $A_{\text{NEAR-INSERT}}$.)

We note that with small modifications, it is not hard to extend the scalefreeness proof to the *Farthest Insertion* heuristic, which inserts at minimum cost the farthest vertex from the vertex-set of the current subtour. The main problem is just that the heuristic will visit S before exhausting Y . However, it will only visit one vertex of S before exhausting Y , which means that we can (at polynomial cost) simply guess the first vertex of S visited by the heuristic, and so A_H will output a polynomially-long list of candidate paths. \square

3. PROOF OUTLINE

We begin by giving a broad outline of the proof of Theorem 1.1, so that the reader can have a preview of the overall structure of the argument. We begin by discussing the task of proving that a Heuristic satisfies the simpler and more powerful Definition 2.2 definition of scalefree.

The proof begins by leveraging Papadimitriou’s reduction [16] of the NP-complete Set Partition problem to the Hamilton path problem. Given any instance of the Set Partition problem, Papadimitriou constructs a set of points and a threshold such that a tour shorter than the threshold exists if and only if the Set-Partition problem is feasible.

Next, we surround this point set with a suitable arrangement of two points to create a configuration \mathcal{Q} , which has the property that *if* the optimal tour through a set $X \supset \mathcal{Q}$ transits \mathcal{Q} in a single pass, then in doing so, it transits the Papadimitriou set optimally (thus solving the corresponding Set Partition problem).

Next we use the path-finding Heuristic A_H guaranteed to exist by Definition 2.2 to define an algorithm to solve the Set Partition problem as follows:

- Generate the set \mathcal{Q} as above corresponding to the given Set Partition instance through Papadimitriou’s reduction,
- Use A_H to produce a list of paths through \mathcal{Q} ,
- If any path is below the threshold given by Papadimitriou’s reduction, return TRUE; otherwise, return FALSE.

We will show the this algorithm can be suitably adapted to the Turing machine setting with polynomial running time, despite the the obvious questions about how to deal with precision issues.

Now, if $P \neq NP$, it must be the case that on some instance of the Set Partition problem, this algorithm gives the wrong answer. Note that when it answers incorrectly, it necessarily answers FALSE. (It can fail to find a short tour, but not incorrectly report the existence of a short tour.)

But for this instance of the Set Partition problem, the corresponding set \mathcal{Q} has the property that any time $\mathcal{Q} \subset X$, and the Heuristic H finds a tour through X , it will transit \mathcal{Q} suboptimally.

Finally, we will apply a Lemma we proved in [7], which shows that if X consists of n independent uniformly random points in the unit square, than X contains linearly many approximate copies of the set \mathcal{Q} . These approximate copies will be close enough to \mathcal{Q} that the tour will still transit them suboptimally, gaining some excess length for each copy, and linearly many such copies are sufficient to ensure a multiplicative $(1 + \varepsilon)$ error in the final tour length.

The proofs below are written for the weaker, more complicated Definition 2.5. The only important modifications to the above outline are surrounding \mathcal{Q} with more configurations of points, so that:

- the set Y required by Definition 2.5 is present,
- it is necessarily the case that \mathcal{Q} is transited in a single pass,
- the angle of entry/exit from Y is controlled, as required by Definition 2.5.

4. ASYMPTOTIC LENGTH OF SCALEFREE HEURISTICS

Our proof involves a multi-layered geometric construction of a configuration, which, when it exists, will ensure that it contains a special set S satisfying hypotheses (a), (b), (c), (d), (e) of the implication in definition of scalefreeness. We consider the layers one at a time.

4.1. Papadimitrou’s set \mathcal{P} . Our proof begins with Papadimitriou’s reduction [16] to the TSP path problem from the NP-complete Set Partition problem. This will form the basis for the sets S we wish to apply the definition of scalefreeness to.

Recall that an instance of the Set Partition problem is a family of subsets $A_i, i = 1, 2, \dots, M$ of $[N] = \{1, 2, \dots, N\}$; the decision problem is to determine whether there is a subfamily which covers $[N]$ and consists of pairwise disjoint sets.

In particular, Papadimitriou shows that for any instance (\mathcal{A}, N) of the set cover problem, there is a k (polynomial in the size of the Set Partition problem), and a set of k points $\mathcal{P} = \mathcal{P}(\mathcal{A}, N)$ in $[0, \sqrt{k}]^2$ with distinguished vertices p and q which can be produced in polynomial time, such that for some absolute constant ε_0 we have that for any approximator $f_{\varepsilon_0} : \mathcal{P} \rightarrow [0, \sqrt{k}]^2$ with $\text{dist}(f(x), x) < \varepsilon_0$ for all $x \in \mathcal{P}$ that

- P1 The shortest TSP path on \mathcal{P} begins at $f_{\varepsilon_0}(p)$ and ends at $f_{\varepsilon_0}(q)$.
- P2 There is a real number L such that the length of the shortest TSP path on $f_{\varepsilon_0}(\mathcal{P})$ is either less than L or greater than $L + \varepsilon_0$, according to whether the Set Partition instance problem should be answered Yes or No, respectively.

Papadimitriou’s discussion does not reference an approximating function like f_{ε_0} ; the role of this function here is to capture the imprecision which can be tolerated by the construction, which is discussed on page 241 of his paper (where one finds, for example, that we can take, e.g., $\varepsilon_0 = \frac{\sqrt{a^2+1}-a}{100(4a^2+2a)}$ for $a = 20$).

4.2. The set \mathcal{Q} . Papadimitriou’s construction in [16] does not have the following property, but it is easy to ensure by simple modification of his construction (using “1-chains” to relocate the original p and q to suitable locations):

- P3 There is a rhombus R with vertices p, q so that all points in $f_{\varepsilon_0}(\mathcal{P} \setminus \{p, q\})$ lie inside R and at least ε_0 from the boundary of R .

FIGURE 2. \mathcal{Q} forces a path through a Papadimitriou path.

When scaled to $[0, \sqrt{k}]^2$ as we have done here, the minimum TSP path length through Papadimitriou's set will always be less than $C_0 k$ for some absolute constant $C_0 > 1$ (indeed, this is true even for a worst-case placement of k points in $[0, \sqrt{k}]^2$ [6]). Thus, given the configuration $\mathcal{P} = \mathcal{P}(\mathcal{A}, N)$ with $k = k(\mathcal{A}, N)$ points and a small $\lambda > 0$, we rescale \mathcal{P} by a factor of say, $\frac{1}{\lambda} C_0 k$, to produce a corresponding set of points $\bar{\mathcal{P}} \subseteq [0, \frac{\lambda}{C_0 \sqrt{k}}]^2$ which necessarily admits a TSP path of at most $\leq \lambda$; note that $\bar{\mathcal{P}}$ satisfies the same properties P1-P3 above, with ε_0 rescaled to $\frac{\lambda \varepsilon_0}{C_0 k}$.

Finally, we modify this configuration (as indicated in Figure 2) by adding two points x, y to the set. With $\bar{\mathcal{P}}$ centered at the origin $(0, 0)$, we take $x = (-1, -\beta), y = (1, -\beta)$, where $\beta \gg \sqrt{\lambda}$ is chosen sufficiently small so that p, q are the closest points on the rhombus R to x and y , respectively. Thus x and y are $\frac{\lambda \varepsilon_0}{C_0 k}$ closer to p and q than to any other point in $\bar{\mathcal{P}}$. We call the resulting set $\mathcal{Q}(\mathcal{A}, N) \subseteq [-1, 1] \times [0, 1]$.

Essentially, the point set \mathcal{Q} ensures that any optimal path passing through it will transit the Papadimitriou set optimally, by ensuring the optimal paths will only enter/exit \mathcal{Q} where we expect.

Lemma 4.1. *Let (\mathcal{A}, N) be an instance of the Set Partition problem, and $\mathcal{Q} = \mathcal{Q}(\mathcal{A}, N)$ with $|\mathcal{Q}| = k$. There is a sufficiently large D_0 , such that If we have that*

- (i) $\mathcal{Q} \approx_\delta \mathcal{Q} \subseteq [0, t]^d$ for $\delta = \frac{\lambda \varepsilon_0}{10 C_0 k}$
- (ii) $w, z \in [0, t]^d$ with $\text{dist}(\{w, z\}, \mathcal{Q}) \geq D_0$,

then the shortest TSP path W from w to z in $\mathcal{Q} \cup \{w, z\}$ has the property that W transits the approximate Papadimitriou set $P \approx_\delta \bar{\mathcal{P}}$ in \mathcal{Q} optimally, from p to q .

Sets $Q \approx_\delta Q$ will serve as the sets S to which we apply the definition of scalefreeness. Very roughly speaking, we will eventually be aiming to contradict $P \neq NP$, since a polynomial-time algorithm to predict optimal paths through Q 's would seem to solve the Set Partition problem in polynomial time.

Proof. Recall that \mathcal{Q} is constructed by adding two points x, y to the set $\bar{\mathcal{P}}$. By construction, the shortest path covering \mathcal{Q} has endpoints x, y , and is of length $< 2 + \lambda + 2\beta$. Moreover, it is apparent that any path covering \mathcal{Q} which does not have the endpoint pair $\{x, y\}$ has length at least 3. Finally, our choice of δ ensures that the accumulated error in path-lengths when comparing paths in Q vs \mathcal{Q} is less than $2(k+1)\delta < \lambda$. Now we suppose that in the shortest path W , w is adjacent to a and z is adjacent to b , where $\{a, b\} \neq \{x', y'\} \subseteq Q$, where x', y' correspond to $x, y \in \mathcal{Q}$. Since W is shortest, we must have that

$$\text{dist}(w, a) + \text{dist}(z, b) + 3 \leq \text{dist}(w, x') + \text{dist}(z, y') + 2 + 2\lambda + 2\beta,$$

and so

$$(1) \quad \text{dist}(w, a) + \text{dist}(z, b) \leq \text{dist}(w, x') + \text{dist}(z, y') - 1 + 2\lambda + 2\beta.$$

Similarly, we have

$$(2) \quad \text{dist}(w, a) + \text{dist}(z, b) \leq \text{dist}(w, y') + \text{dist}(z, x') - 1 + 2\lambda + 2\beta.$$

So we suppose now that (1) and (2) hold simultaneously. Moreover, let us assume without loss of generality that we have one of the following three cases:

Case 1: $\text{dist}(w, x') < \text{dist}(w, y')$ and $\text{dist}(z, y') < \text{dist}(z, x')$, or

Case 2: $\text{dist}(w, x') < \text{dist}(w, y')$ and $\min\{\angle wx'y', \angle wy'x'\}$ is at least $\gamma > 0$, for some γ depending on β , or

Case 3: wx' and zp is the shortest pair of independent edges joining $\{w, z\}$ to Q .

Before finishing the proof for each case, let's verify that for D_0 large, these cases do indeed cover all cases. If either $\min\{\angle wx'y', \angle wy'x'\}$ or $\min\{\angle zx'y', \angle zy'x'\}$ are at least γ , then we are already in Case 2, by appropriate choices of the labels w, z, x', y' from the available pairs. If on the other hand both $\min\{\angle wx'y', \angle wy'x'\}$ and $\min\{\angle zx'y', \angle zy'x'\}$ are at most γ , then either the angle from w to the center of Q to z is in $(\pi - \gamma, \pi + \gamma)$, and we are in Case 1 with the correct choice of which endpoints of Q are called x', y' , or the angle is less than γ , and we are in Case 3 with a suitable choice of labels.

Case 1: In this case, making D_0 large ensures that $\text{dist}(w, x') - \text{dist}(w, a)$ and $\text{dist}(z, y') - \text{dist}(z, b)$ are bounded by a number arbitrarily close to 0, violating (1) (or (2), if we had flipped the roles of w and z). So this case cannot occur in simultaneously with (1) and (2).

Case 2: The second condition of this case implies that for any $\alpha \in Q$, we have that $\text{dist}(w, x') < \text{dist}(w, a) + \lambda$ (for sufficiently large D_0), which allows us to modify (1) to the inequality

$$(3) \quad \text{dist}(z, y') - \text{dist}(z, b) \geq 1 - 3\lambda - 2\beta.$$

But z is at at least some fixed positive angle from the line through x, y (to which all points in Q are arbitrarily close). Thus, making D_0 large ensures that $\text{dist}(z, a)$ varies by an arbitrarily small amount as we vary $\alpha \in Q$, contradicting (3).

Case 3: Recall that the shortest path on Q goes from x to p , then through \bar{P} optimally to q , and then ends at y . In particular, this path has endpoint pair $\{x, y\}$.

Let us consider the lengths of shortest paths through Q for choices of endpoints other than $\{x, y\}$. In particular, we claim that the next-best pair of endpoints is either $\{x, p\}$, and $\{q, y\}$. For the pair x, p , the short path goes from x to y to q , through \bar{P} to p . The path for the pair y, q goes from y to x to p , through \bar{P} to q (which of these choices gives rise to a shorter path depends on the precise rounding $Q \approx \mathcal{Q}$). These paths both have length $3 + \beta^2/2 + \Theta(\lambda + \beta^3)$ (here we use that $\sqrt{1 + \beta^2} \approx 1 + \beta^2/2$). And now we will show that any path with a pair of endpoints other than $\{x, y\}$, $\{x, p\}$, or $\{q, y\}$ will be longer than any of these choices.

We prove this as follows. First let us consider the case where both endpoints lie in $\bar{\mathcal{P}}$. In this case, both x and y are internal vertices of the path through Q , and thus it has length at least 4. Thus we may assume that one endpoint is x , while the other is some vertex $v \in \bar{\mathcal{P}}$. We consider two cases: suppose from v the path transits all of $\bar{\mathcal{P}}$ before leaving $\bar{\mathcal{P}}$; in this case, the optimal choice (over all choices for $v \in \bar{\mathcal{P}}$) is clearly to begin from p , transit $\bar{\mathcal{P}}$ optimally ending in q , proceed to y , and then return to x , as before. Suppose instead that from v , the path visits some of $\bar{\mathcal{P}}$, then visits y , then returns to visit the rest of $\bar{\mathcal{P}}$, before exiting to $\bar{\mathcal{P}}$ to x . In this case, the total length used is roughly $3 + 2\beta^2/2 + \Theta(\lambda)$.

Thus, since wx and zp is the shortest pair of independent edges joining $\{w, z\}$ to Q , we must have either that the shortest path from w to z through Q uses these edges and takes the optimum path in Q from x to p , or else that it takes a path in Q which is shorter than this optimum path from x to p . But from above, we see that each such choice transits $\bar{\mathcal{P}}$ optimally. \square

4.3. The set \mathcal{M}_H . We use \mathcal{Q} to construct a larger set \mathcal{M}_H . \mathcal{M}_H consists of:

- (1) a copy of \mathcal{Q} , rescaled by a factor α to lie in $B(0, 1)$, and
- (2) the set $Y = Y_H \subseteq B(0, \sqrt{R})$ from the definition of scalefreeness.

The set \mathcal{M}_H ensures that hypothesis (a) of Definition 2.5 is satisfied at points centering approximations of \mathcal{M}_H .

4.4. Perturbing input \mathcal{M}_H for simulation of A_H by a Turing machine. The heart of our proof will consist of using the path-finding heuristic A_H to attempt to find a good path through the set \mathcal{M}_H , which would require solving the Papadimitrou set, and contradicting the assumption that $P \neq NP$. We will show that the failure of A_H to transit \mathcal{M}_H optimally leads to a significant excess length in the tour T_H found by the heuristic H , since a significant number of suitable approximate copies of \mathcal{M}_H occur throughout the random point set, just by chance.

For the behavior of A_H on \mathcal{M}_H to reliably predict the behavior of the approximate copies throughout the tour T_H , however, we will need to know that the behavior of A_H is stable to small perturbations in the positions of the points in \mathcal{M}_H . Indeed, this may not be the case in general, but at least, we want to know that we can efficiently, with a deterministic algorithm, perturb the points of \mathcal{M}_H to produce a set \mathcal{M}_H^g on which the behavior of A_H is stable with respect to (even smaller) perturbations.

This leads to the following question, which one might hope would be easier than resolving the computational status of comparing sums of radicals:

Question 4.2. Given K , is there an L and a polynomial-time deterministic algorithm which takes as input a set X of N points (at polynomial precision), and a parameter $\varepsilon = 2^{-N^K}$, and outputs a perturbed set of points $X' \approx_\varepsilon X$, such that for all nontrivial choices of $\xi_i \in \{-1, 0, 1\}$, we have that $|\sum \xi_i x_i| \geq 2^{-N^L}$?

We do not need to answer to this question here, however. Instead, we take advantage of the fact that A_H is required to terminate after only polynomially many comparisons, to prove the following Lemma, which offers an easier way out:

Lemma 4.3. *For a polynomial path-finding heuristic A_H , a set Z of N points, and a constant K , we can deterministically, in polynomial time, find a set $Z' \approx_\varepsilon Z$ for $\varepsilon = 2^{-N^K}$ such that each of the polynomially many sums $\sum_{i=1}^{\binom{N}{2}} \xi_i d_i$ evaluated by A_H on input Z' will be separated from 0 by at least 2^{-N^L} , for a constant L depending only on K and A_H .*

Proof. Recall that the path-finding heuristic is an algorithm which runs in polynomial time in the number of points in its input, with access to a blackbox for making comparisons among sums of distances between input points. The Lemma we are proving claims the existence of a deterministic Turing machine to round the points of the input set Z . To carry out this procedure, we begin by simply running A_H on the input Z . At each time $t = 1, 2, \dots, \text{poly}(N)$ when a comparison $C = \sum \xi_i^{(t)} d_i$ is requested of the blackbox, we do the following:

- (1) Compute $\sum \xi_i^{(t)} d_i$ to precision $2^{-(N^K+10t+8)}$.
- (2) IF the result lies within $2^{-(N^K+10t+4)}$ of 0, THEN:

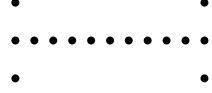
Step A: Change the position of some points by at most (in total) distance $2^{-(N^K+10t+2)}$ so that after the change, the same sum differs by at least $2^{-(N^K+10t+4)}$ from 0.
- (3) Accept the resulting status (positive/negative) of the sum as if it had been returned by the blackbox, and continue running A_H .

(The implementation of Step A is discussed below.) In this way, as A_H runs on Z , we perturb the input points repeatedly so that all comparisons requested of the blackbox can be distinguished from 0 by a Turing machine operating at polynomial precision.

Consider some comparison step t after which the t th sum $\sum \xi_i^{(t)} d_i$ was guaranteed to differ by at least $2^{-(N^K+10t+4)}$ from 0. In each subsequent step, this t th sum will change slightly, as points of the input are perturbed further. However, by the triangle inequality, the total impact on this t th sum of later perturbations is bounded by

$$\sum_{\ell=t+1}^{\infty} 2^{-(N^K+10\ell+2)} = 2^{-(N^K+10t+2)} \sum_{\ell'=1}^{\infty} 2^{-10\ell'} \leq 2^{-(N^K+10t+11)},$$

which is smaller than the difference $2^{-(N^K+10t+4)}$ guaranteed at step t by the step t perturbation. As a result, at the end of the this procedure, (after polynomially many perturbations have been carried out, for each of the polynomially many steps of the run of A_H), each of the comparisons which A_H requested can be reliably computed on the perturbed point set by a Turing machine simulating A_H at polynomial precision 2^{-N^L} for a suitable constant L , which depends just on K and the polynomial running time of A_H . \square

FIGURE 3. The set $\Pi(10)$ (rotated 90 degrees).

Given \mathcal{M}_H , we let \mathcal{M}_H^g denote the the set resulting from the perturbation procedure of Lemma 4.3.

Implementing Step A. We first choose two points p, q such that there exists i such that $\xi_i^{(t)}$ is not zero. Let $d_i = d = |p - q|$. WLOG we let $p = (0, 0)$ and $q = (d, 0)$. Let $\rho = 2^{-(N^K+10t+3)}$ and define the points $p' = p - (\rho, 0)$, $q' = q + (\rho, 0)$.

Next let $p_j, j = 1, 2, \dots, K$ be the points, other than q , such that for some i we have $d_i = |p - p_j|$ and $\xi_i^{(t)} \neq 0$. Let $d'_j = |p_j - p'|$ for $i = 1, \dots, K$ and then re-defining $d_j = |p - p_j|$ we let

$$\Delta(p) = \left| \sum_{j=1}^K \xi_j^{(t)} (d_j - d'_j) \right|.$$

We observe that replacing p by p' results in a new comparison C'_p where

$$(4) \quad C'_p - C = \Delta(p) + \zeta_p \text{ where } \zeta_p = |p' - q| - |p - q|.$$

Similarly, if we replace q by q' then we obtain a new comparison C'_q where

$$(5) \quad C'_q - C = \Delta(q) + \zeta_q \text{ where } \zeta_q = |p - q'| - |p - q|.$$

We now consider two cases:

Case 1: $\max \{|C'_p|, |C'_q|\} \geq \rho/2$.

In this case the move $p \rightarrow p'$ or the move $q \rightarrow q'$ implements Step A.

Case 2: $\max \{|C'_p|, |C'_q|\} < \rho/2$.

Suppose now we move $p \rightarrow p'$ and $q \rightarrow q'$ to obtain a comparison $C'_{p,q}$. Then we have

$$(6) \quad C'_{p,q} = \Delta(p) + \Delta(q) = C'_p + C'_q - (\zeta_p + \zeta_q).$$

But,

$$\zeta_p = \zeta_q = \rho,$$

So, from (6), we see that

$$|C'_{p,q}| \geq 2\rho - \rho = \rho,$$

and so moving $p \rightarrow p'$ and $q \rightarrow q'$ implements Step A.

4.5. The set Π_H . We now construct a certain set $\Pi = \Pi(k)$ and show that it constrains the optimal tour in a useful way. In particular, we let $\Pi(k)$ consist of the four points $\pi_1 = (0, 5)$, $\pi_2 = (0, 0)$, $\pi_3 = (1, 0)$, $\pi_4 = (1, 5)$ together with all the points $(\frac{1}{2}, \frac{5j}{k})$ for $0 \leq j \leq k$ (see Figure 3).

Lemma 4.4. *If $\text{dist}(\{x, y\}, \Pi(k))$ is sufficiently large and P is a shortest Hamilton path from x to y in $\Pi(k) \cup \{x, y\}$, then for at least one $i \in \{1, 2, 3, 4\}$ we have that neither neighbor v_i^1, v_i^2 of π_i on P is in $\{x, y\}$, and moreover that $\text{dist}(\pi_i, v_i^1)$ and $\text{dist}(\pi_i, v_i^2)$ become arbitrarily close to $\frac{1}{2}$ as k increases; in particular, the neighbors of π_i are nearly horizontal translates of π_i , lying on the line $x = \frac{1}{2}$. \square*

We now define Π_H as follows. We take four copies of the set \mathcal{M}_H^g , each rescaled to lie in small balls of radius $\varepsilon_\Pi > 0$. In $\Pi(k)$ (for $k = k(H)$ sufficiently large), we replace the four points π_i with these copies; those corresponding to π_3 and π_4 are reflected horizontally. (In particular, the resulting set Π_H still has horizontal reflection symmetry.)

Suppose we take this ε_Π suitably small, that $U \approx_{\delta_H} \Pi_H$ for sufficiently small $\delta_H > 0$, and that the optimal tour on $X \supseteq U$ transits U in a single path. Then each approximate copy of \mathcal{M}_H^g in U is transited in single path by the optimal tour (corresponding to hypothesis (d) from Definition 2.5) and moreover, Lemma 4.4 implies that for at least one of the four copies M_1, M_2, M_3, M_4 of \mathcal{M}_H^g , H either satisfies hypothesis (e) or else pays an additive error.

4.6. The set Π_H^3 . We let Π_H^3 denote 3 copies of Π_H centered at the vertices of an equilateral triangle of sidelength $2D_1$, say. This triple configuration ensures that the optimum tour will transit at least one of the copies of \mathcal{M}_H^g in a single pass. Indeed, Observations 2.9 and 2.10 from [7] now give the following:

Lemma 4.5. *Suppose that D_2 is a sufficiently large absolute constant, (\mathcal{A}, N) is an instance of the Set Partition problem, and $\Pi_H^3 = \Pi_H^3(\mathcal{A}, N)$. If $\Pi_H^3 \approx_{\delta_0} Z \subseteq X \subseteq [0, t]^d$ for $\delta_0 = \frac{\alpha \varepsilon_0}{10^4 C_0 D_0 k}$ and $\text{dist}(Z, X \setminus Z) \geq D_2$, then any TSP tour T on X can either be shortened in Π_H^3 by some additive constant or otherwise has the property that at least one of the (approximate) copies of Π_H in Z is traversed (optimally) by a path by T . \square*

We emphasize that the constants $\alpha \leq 1 \ll D_0 \ll D_1 \ll D_2$ are absolute, independent of (\mathcal{A}, N) .

4.7. Using a TSP Heuristic to solve the Set Partition problem. At this point, in preparation for the proof of Theorem 1.1, we use the scalefree heuristic H to define the following polynomial time algorithm to solve a Set Partition instance (\mathcal{A}, N) , which will be correct unless Theorem 1.1 holds for H . (Thus, $P \neq NP$ will imply Theorem 1.1.)

- (1) Compute $\mathcal{M}_H(\mathcal{A}, N)$ to precision $\frac{\delta_0}{4}$. Using Lemma 4.3, perturb the points as necessary to produce a set M , for which the path-finding heuristic A_H can be simulated by a Turing machine in polynomial-time.
- (2) Produce a list of paths through M using the algorithm A_H . Because of the rounding M produced by Lemma 4.3, comparisons of sums of distances can be done with Turing machine operations.

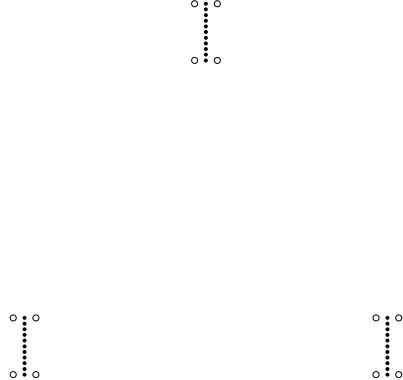


FIGURE 4. Π_H^3 , consisting of three copies of Π_H , forces an optimum tour to transit one of 12 copies of \mathcal{M}_H^g (indicated here as the small circles) in a single pass, from a narrow prescribed angle.

- (3) Let L' be the minimum length of a path covering one of the three Papadimitriou sets which is a subpath of one of the paths enumerated in step 2.
- (4) Return TRUE/FALSE according to whether L' lies within or above the threshold given by P2, respectively.

Proof of Theorem 1.1. We suppose that H is scalefree. $P \neq NP$ implies that there is some instance (\mathcal{A}_0, N_0) of the Set Partition problem for which the algorithm above returns an incorrect answer. Observe first that it cannot happen that the algorithm returns TRUE when the correct answer to the Set Partition instance is FALSE: when (\mathcal{A}_0, N_0) is FALSE, property P2 implies that there can be no path through the Papadimitriou sets shorter than the threshold below which the algorithm above would return TRUE.

Thus we are to consider the case that the algorithm above returns FALSE even though the correct answer to (\mathcal{A}_0, N_0) is TRUE. In this case, no path enumerated by A transits M in such a way that a Papadimitriou set is traversed optimally.

We prove Theorem 1.1 by showing that this implies there exists an $\varepsilon_H > 0$ so that the length of the tour found by H through the random set $\mathcal{Y}_n \subseteq [0, t]^d$ is w.h.p at least $(1 + \varepsilon_H)$ times the length of the optimal tour T .

An (ε, R) -copy of Z in $\mathcal{Y}_n \subseteq [0, t]^d$ is a set $Z' \subseteq \mathcal{Y}_n$ such that $Z' \approx_\varepsilon Z$ and such that $\text{dist}(Z', \mathcal{Y}_n \setminus Z') \geq R$. The following Lemma shows that we find a linear number of (ε, R) copies of any fixed finite set in the random set \mathcal{Y}_n (see Observation 3.1 from [7]):

Lemma 4.6. *Given any finite point set S , any $\varepsilon, \delta > 0$, and any R , we have that the number ζ_S of (ε, R) -copies S' of S in a random set $X = \mathcal{Y}_n \subseteq [0, t]^d$, such that the points of S' are δ -distance separated, satisfies*

$$(7) \quad \zeta_S \geq C_{S, R, \varepsilon} n \quad \text{w.h.p.}$$

for some constant $C_{S,R,\varepsilon} > 0$. □

Now we take ε_2 to be the minimum of $\frac{\delta_0}{8}$ and the parameter ε from the definition of scalefreeness for the heuristic H , take $R = D_2$, take $S = \Pi_H^3 = \Pi_H^3(\mathcal{A}_0, N_0)$, and use Lemma 4.6 to find a linear number of $(\frac{\varepsilon_2}{(Kd+1)^{12}}, R + \varepsilon_2)$ copies Z of Π_H^3 which are δ -distance separated.

We say that such a copy Z of Π_H^3 has the property Λ_X if the tour T_H can be shortened within Z by δ_1 for some sufficiently small but fixed $\delta_1 > 0$, and we let ν_Λ denote the number of copies Z of Π_H^3 with property Λ_X .

Claim: There exists $C > 0$ so that if H is scalefree, then $\nu_\Lambda \geq Cn$ w.h.p.

Note that the claim immediately implies the theorem: in the rescaled torus $[0, t]^d$, the heuristic pays a total error of $\delta_1 \cdot \nu_\Lambda$, and rescaling by $t = n^{1/d}$, this gives Theorem 1.1.

Proof of the Claim. Each δ -separated copy Z itself consists of three different copies $Z_1, Z_2, Z_3 \approx_\varepsilon \Pi_H(\mathcal{A}_0, N_0)$, and when Z fails to have property Λ_X , Lemma 4.5 implies that at least one of the copies Z_j is transited by T_H in a single path. Fixing a choice of such a copy Z_j , Lemma 4.4 gives that at least one of the four copies M_i of \mathcal{M}_H^g in Z_j (and so one of the twelve copies of \mathcal{M}_H^g in Z) satisfies hypotheses (d) and (e) of Definition 2.5. Of course, by construction of \mathcal{M}_H , hypothesis (a) is satisfied at the center of the copy. Moreover, with at most finitely many exceptions, we may assume that \mathcal{M}_H satisfies hypothesis (c). Moreover, by construction of the perturbation \mathcal{M}_H^g of \mathcal{M}_H using Lemma 4.3, \mathcal{M}_H^g can be rounded to polynomial precision and still satisfy (b). Thus all the hypotheses of the implication in Definition 2.5 are satisfied for this copy M_i of \mathcal{M}_H^g . In particular, we conclude that the tour T_H traverses this copy M_i with one of the paths output by A_H on input \mathcal{M}_H^g ; by hypothesis (since our proposed Set Partition algorithm output FALSE on the instance), the path through M_i can be shortened by δ_1 . □

5. BRANCH AND BOUND

In our paper [7] we considered branch and bound algorithms for solving the Euclidean TSP. Branch-and-bound is a pruning process, which can be used to search for an optimum TSP tour. Branch-and-bound as we consider here depends on three choices:

- (1) A choice of heuristic to find (not always optimal) TSP tours;
- (2) A choice of lower bound for the TSP;
- (3) A branching strategy (giving a *branch-and-bound* tree).

For us a branch-and-bound tree is a rooted tree $T_{B\&B}$ where each vertex v is labelled by a 4-tuple $(b_v, \Omega_v, I_v, O_v)$. Here I_v, O_v are disjoint sets of edges and Ω_v is the set of tours T such that $T \supseteq I_v$ and $T \cap O_v = \emptyset$. The value b_v is some lower bound estimate of the minimum length of a tour in Ω_v e.g. the optimal value of

the Held-Karp linear bound relaxation [5], [10], [11]. In addition there is an upper bound B , which is the length of the shortest currently known tour, found by some associated heuristic. This is updated from time to time as we discover better and better tours. If the root of the tree is denoted by x then we have $I_x = O_x = \emptyset$.

In [7] we allowed essentially any branching strategy. Given \mathcal{X}_n , we allowed any method to produce a tree satisfying the following:

- (a) When v is a child of u , $I_v \supseteq I_u$ and $O_v \supseteq O_u$.
- (b) If the children of u are v_1, \dots, v_k , then we have $\Omega_u = \bigcup_{i=1}^k \Omega_{v_i}$.
- (c) The leaves of the (unpruned) branch-and-bound tree satisfy $|\Omega_v| = 1$.

This process terminates when the set L of leaves of the pruned branch-and-bound tree satisfies $v \in L \implies b_v \geq B$; such a tree corresponds to a certificate that the best TSP tour found so far by our heuristic is indeed optimum. It is clear that if $v \in L$ then Ω_v does not contain any tours better than one we already know.

In [7] we concentrated on showing that even if we had access to the exact optimum i.e. letting $B = \lambda$, the minimum length of a tour, none of a selected set of natural lower bounds would be strong enough to make the branch and bound tree polynomial size. Note that this result does not depend on the branching process itself being efficient.

The aim of this section is to show that even if $b_v = \lambda$ then a certain branching strategy will fail. Unlike in [7], we cannot allow any branching strategy for our present result, as we might (though extreme computation in the branching process) find that we directly branch to a vertex w where I_w is exactly the set of edges of the shortest tour, giving $B = \lambda$, causing the algorithm to terminate, given that $b_v = B$ for all leaves of the tree.

It turns that to prove our result, we will need only a mild restriction on the branching strategies allowed.

- (1) A vertex v of out tree has two children w_+, w_- . Here $I_{w_+} = I_v \cup \{e\}, O_{w_+} = O_v$ and $I_{w_-} = I_v, O_{w_-} = O_v \cup \{e\}$ for some edge e .
- (2) The branch and bound tree is explored in a breadth first manner i.e. if the root is at level 0, we do not produce vertices of level $k + 1$ until all vertices at level k have been pruned or branched on.

Note that this captures most branching strategies used in practice, which typically are using an LP-based lower bound on the length of the tour, and branching on the binary values possible for fractionally-valued variables in the linear program.

Theorem 1.2 will follow easily from the following claim: let H denote some scale-free heuristic and for a vertex v of $T_{B\&B}$ let $H(v)$ denote the length of the tour constructed by H , when it accounts for I_v, O_v . Let $\lambda(v)$ denote the length of the shortest tour in Ω_v .

Lemma 5.1. *There exist constants $\varepsilon_1, \varepsilon_2$ such that w.h.p. if vertex v is at depth at most $k_1 = \varepsilon_1 n$ then $H(v) \geq \lambda(v) + \varepsilon_2 n^{1/2}$.*

Proof. Let $\alpha_0 n$ be the minimum number of copies of Z with property Λ_X promised by our analysis above, and let $\alpha_1 n^{1/2}$ be a lower bound on the penalty paid by our heuristic for each copy of Z . Then if $\varepsilon_1 = \alpha_0/2$ we have the lemma for $\varepsilon_2 = \alpha_0 \alpha_1/2$. \square

Here we have used the fact that at depth at most k_1 , w.h.p. there will be a linear number of copies of Z that are unaffected by I_v, O_v . These copies provide the necessary increases over the optimum.

It follows from Lemma 5.1 that for v at depth at most k_1 we have

$$H(v) \geq \lambda(v) + \varepsilon_2 n^{1/2} \geq B + \varepsilon_2 n^{1/2}.$$

This means that v is not a leaf. It follows that w.h.p. there will be at least $2^{k_1} = e^{\Omega(n)}$ leaves and Theorem 1.2 follows. \square

6. FURTHER WORK

From among the heuristics used in practice, the major omission from the present manuscript are the k -opt improvement heuristics, and their relatives (such as the Lin-Kernighan heuristic). Are they scalefree in our sense (or a related sense for which Theorem 1.1 holds)?

Theoretically, a natural question is whether our definition of a scalefree heuristic can be significantly simplified, while still allowing Theorem 1.1.

REFERENCES

- [1] S. Arora, Polynomial time approximation schemes for Euclidean Traveling Salesman and other geometric problems, *Journal of the Association for Computing Machinery* 45 (1998) 753-782.
- [2] D. Applegate, R. Bixby, V. Chvátal and W. Cook, *The Traveling Salesman Problem: A Computational Study*, Princeton University Press, 2007.
- [3] J. Beardwood, J. H. Halton and J. M. Hammersley, The shortest path through many points, *Mathematical Proceedings of the Cambridge Philosophical Society* 55 (1959) 299-327.
- [4] N. Christofides, The Traveling Salesman Problem, in *Combinatorial Optimization*, N. Christofides, P. Toth, A. Mingozzi and C. Sandi, Eds., John Wiley and Sons, New York, 1971.
- [5] G.B. Dantzig, D.R. Fulkerson and S.M. Johnson, Solution of a large scale traveling salesman problem, *Operations Research* 2 (1954) 393-410.
- [6] L. Few, The shortest path and the shortest road through n points, *Mathematika* 2 (1955) 141-144.
- [7] A. Frieze and W. Pegden, Separating subadditive Euclidean functionals, *Random Structures and Algorithms* 51 (2017) 375-403.
- [8] M. R. Garey, R. L. Graham, and D. S. Johnson, Some NP-complete geometric problems, in *Proceedings of the eighth annual ACM symposium on Theory of computing (STOC '76)*, ACM, New York, NY, USA, 10-22 (1976).

- [9] M. Held and R.M. Karp, A dynamic programming approach to sequencing problems, *Journal of the Society for Industrial and Applied Mathematics* 10 (1962) 196-210
- [10] M. Held and R.M. Karp, The Traveling Salesman Problem and Minimum Spanning Trees, *Operations Research* 18 (1970) 1138-1162.
- [11] M. Held and R.M. Karp, The Traveling Salesman Problem and Minimum Spanning Trees Part II, *Mathematical Programming* 1 (1971) 6-25.
- [12] D. Johnson, L. Mcgeoch and E. Rothberg, Asymptotic Experimental Analysis of the Held-Karp Traveling Salesman Bound, *Proceedings of the 7th Annual ACM Symposium on Discrete Algorithms* (1991) 341-350.
- [13] C. McDiarmid, On the Method of Bounded Differences, *Surveys in Combinatorics* 141 (1989) 148188.
- [14] D. Miller and J. Pekny, Exact Solution of Large Asymmetric Traveling Salesman Problems, in *Science* 251 754-761.
- [15] J. 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 (1999) 1298-1309.
- [16] C. H. Papadimitriou, The Euclidean travelling salesman problem is NP-complete, in *Theoretical Computer Science* 4 (1977) 237-244
- [17] J. Michael Steele, Subadditive Euclidean functionals and nonlinear growth in geometric probability, *The Annals of Probability* 9 (1981) 365-376.
- [18] J. Michael Steele, Probability Theory and Combinatorial Optimization, SIAM CBMS series, 1996.
- [19] S. Steinerberger, New Bounds for the Traveling Salesman Constant, *Advances in Applied Probability* 47 (2013)
- [20] L. Tóth, Über einen geometrischen Satz, *Mathematische Zeitschrift* 46 (in German) (1940) 8385.
- [21] T. Volgenant and R. Jonker, A Branch and Bound Algorithm for the Symmetric Traveling Salesman Problem based on the 1-tree Relaxation, *European Journal of Operations Research* 9 (1982) 83-89.

DEPARTMENT OF MATHEMATICAL SCIENCES, CARNEGIE MELLON UNIVERSITY, PITTSBURGH, PA 15213, U.S.A.

Email address, Alan Frieze: alan@random.math.cmu.edu

Email address, Wesley Pegden: wes@math.cmu.edu