Geometric TSP on Sets

Henk Alkema
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Mark de Berg
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Abstract

In One-of-a-Set TSP, also known as the Generalised TSP, the input is a collection $\mathcal{P} := \{P_1, ..., P_r\}$ of sets in a metric space and the goal is to compute a minimum-length tour that visits one element from each set.

In the Euclidean variant of this problem, each P_i is a set of points in $\mathbb{R}^d$ that is contained in a given hypercube H_i. We investigate how the complexity of Euclidean One-of-a-Set TSP depends on λ, the ply of the set $\mathcal{H} := \{H_1, ..., H_r\}$ of hypercubes (The ply is the smallest λ such that every point in $\mathbb{R}^d$ is in at most λ of the hypercubes). Furthermore, we show that the problem can be solved in $2^{O(\lambda^{1/d} n^{1-1/d})}$ time, where $n := \sum_{i=1}^{r} |P_i|$ is the total number of points. Finally, we show that the problem cannot be solved in $2^{o(n)}$ time when $\lambda = \Theta(n)$, unless the Exponential Time Hypothesis (ETH) fails.

In Rectilinear One-of-a-Cube TSP, the input is a set $\mathcal{H}$ of hypercubes in $\mathbb{R}^d$ and the goal is to compute a minimum-length rectilinear tour that visits every hypercube. We show that the problem can be solved in $2^{O(\lambda^{1/d} n^{1-1/d} \log n)}$ time, where n is the number of hypercubes.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Theory of computation → Computational geometry

Keywords and phrases Euclidean TSP, TSP on Sets, Rectilinear TSP, TSP on Neighbourhoods

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2023.6

Funding The work in this paper is supported by the Dutch Research Council (NWO) through Gravitation-grant NETWORKS-024.002.003.

1 Introduction

In the Traveling Salesman Problem we are given an edge-weighted complete graph and the goal is to compute a tour, i.e., a simple cycle visiting all nodes, of minimum total weight. The Traveling Salesman Problem is among the most famous problems in computer science and combinatorial optimization. One variation is the Euclidean TSP. In Euclidean TSP the input is a set P of n points in $\mathbb{R}^d$, and the goal is to compute a minimum-length tour visiting each point. This problem was proven to be NP-hard in the 1970s [6, 15]. However, unlike the general (metric) version, Euclidean TSP in the plane can be solved in *subexponential* time, i.e., in time $2^{o(n)}$. Both Kann [10] and Hwang et al. [8] have given algorithms with $n^{O(\sqrt{n})}$ running time. Smith and Wormald [16] gave a subexponential algorithm that works in any (fixed) dimension d, taking $n^{O(n^{1-1/d})}$ time. Recently De Berg et al. [3] improved this to $2^{O(n^{1-1/d})}$, which is tight up to constant factors in the exponent, under the Exponential-Time Hypothesis (ETH) [9].

Meanwhile, generalised versions of the Traveling Salesman Problem have also been studied. One popular example is the One-of-a-Set TSP, also known as Generalised TSP or Group TSP. Here, the n nodes of the graph are partitioned into sets V_i and the goal is to compute a tour of minimal weight visiting at least (or exactly) one node of every set. The One-of-a-Set TSP has been studied extensively, see for example the survey by Gutin and Punnen [7]. In 2008, Dror and Orlin showed that even if the vertices and their distances correspond to locations in $\mathbb{R}^d$ and their Euclidean distances, and every set V_i contains only two vertices, the problem is still APX-hard, i.e., there exists no PTAS unless P = NP [5].

34th International Symposium on Algorithms and Computation (ISAAC 2023).
Editors: Satoru Iwata and Naonori Kakimura; Article No. 6; pp. 6:1–6:19
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

One generalisation of Euclidean TSP is TSP with Neighbourhoods. Here, we are given a set of neighbourhoods in $\mathbb{R}^d$ – the shape of these neighbourhoods depends on the variant studied – and the goal is to find a shortest Euclidean tour visiting all neighbourhoods. As it is a generalised version of an APX-hard problem it is APX-hard itself [4]. Since then, many variants which place restrictions on the shape of the neighbourhoods have been shown to have a PTAS. This includes disjoint fat convex regions [4], pairwise-disjoint connected regions of any shape [14], arbitrary disjoint fat regions [13], and weakly disjoint neighbourhoods [2].

In this paper, we focus on two geometric variations of One-of-a-Set TSP. For the first variation, Euclidean One-of-a-Set TSP, most research has been focused on the so-called *grid cluster* variant, introduced by Bhattacharya et al. [1]. In this variant, a partition is specified by the cells of the integer 1×1 grid (on the Euclidean plane); from every non-empty cell, exactly one point needs to be visited. Khachay and Neznakhina showed that a PTAS exists if there are many ($O(\log n)$) or few ($n - O(\log n)$) non-empty cells [11], and that if the grid has fixed height or width, a solution can be found in polynomial time [12].

For the second variation, Rectilinear One-of-a-Cube TSP, instead of discrete sets V_i we are given a set of (hyper)cubes H_i. The goal is to find the shortest rectilinear tour which visits at least one point of every H_i. Thus this is a variant of TSP with Neighbourhoods where the neighborhoods are hypercubes.

Our contribution. We investigate the complexity of Euclidean One-of-a-Set TSP and Rectilinear One-of-a-Cube TSP. For Euclidean One-of-a-Set TSP, let $\mathcal{H}$ be a set of hypercubes $H_1, ..., H_r$. Let $\mathcal{P}$ be a family of sets of points $P_1, ..., P_r$ with $P_i \subset H_i$ and $|P_i| \leq k$ for all i. We will use P to denote $\cup_i P_i$. Our objective is to find a shortest tour $T = (p_1, ..., p_n)$ such that for every P_i there exists a $p \in P_i$ such that $p \in T$. Let λ be the *ply* of the given hypercubes, i.e., the smallest number such that every point in $\mathbb{R}^d$ is in at most λ of the hypercubes.

Intuitively, one would expect that the complexity of Euclidean One-of-a-Set TSP depends on how well separated the sets P_i are. To formalize this intuition, we investigate the dependency of its complexity on the ply of $\mathcal{H}$. We present an algorithm that runs in $2^{O(\lambda^{1/d} n^{1-1/d})}$ time, which is based on a recent algorithm by De Berg et al. [3]. Note that for $\lambda = 1$ this matches the ETH-tight running time for Euclidean TSP. For $\lambda = n$, however, the running time becomes $2^{O(n)}$, so it is no longer sub-exponential. We show this is unavoidable (assuming ETH) by proving that Euclidean One-of-a-Set TSP in $\mathbb{R}^2$ cannot be solved in $2^{o(n)}$ for $\lambda = n$. Finally, we show that we instead of using the ply of a set of hypercubes covering the P_i, one can use the ply of a set of more generic objects covering the P_i.

For Rectilinear One-of-a-Cube TSP, let $\mathcal{H}$ be a set $\{H_1, ..., H_n\}$ of hypercubes in $\mathbb{R}^d$. Let λ be the *ply* of these hypercubes. Our objective is to find a shortest rectilinear tour $T = (p_1, ..., p_n)$ such that for every H_i there exists a $p \in H_i$ such that $p \in T$. For this case we present an algorithm running in $2^{O(\lambda^{1/d} n^{1-1/d}) \log n}$ time.

2 A subexponential algorithm for Euclidean One-of-a-Set TSP

We start by giving an overview of our algorithm. The problem it solves is the more generic Euclidean One-of-a-Set Path Cover problem. In this problem, we are given a collection of point sets P_i and a set of *boundary points* B, where $|B| \geq 2$ is even. The goal is for every possible matching on B to find the shortest collection of paths that (i) have the so-called *Packing Property*, (ii) adhere to the matching and (iii) visit one point of every P_i. The Packing Property, which is known to hold for the set of edges of an optimal TSP tour, intuitively states that an optimal tour cannot contain many long edges close together; the precise definition is not important for this paper. Note that this property also holds for an

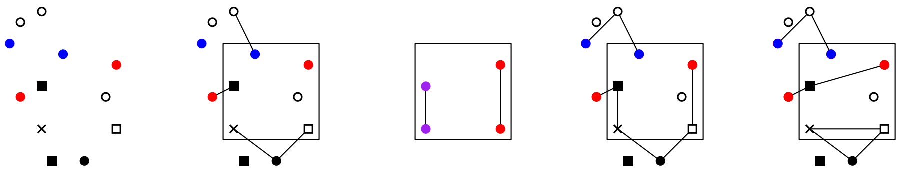

Figure 1 An example of the Euclidean One-of-a-Set TSP algorithm. (a) The given problem. In red and blue, the four boundary points, and the corresponding matching we will be using. In black, the five given point sets (circles, disks, squares, filled squares, and crosses). (b) We find a good separator σ, and guess how it is crossed. (c) The resulting subproblem for the points inside σ, and one possible matching. Note that the circle inside σ has been removed, as we guessed we visit a circle outside σ. (d) The answer after combining two answers of the subproblems. (e) Note that not every combination of matchings leads to a valid answer.

optimal Euclidean One-of-a-Set TSP tour T_{opt}, as it is obviously also an optimal tour for the Euclidean TSP instance obtained by taking only a single point from each P_i into account, namely the point visited by T_{opt}.

Our algorithm broadly works in the following way; see Figure 1. We define a *separator* to be the boundary of an axis-aligned hypercube. First, we find a separator σ that is crossed only a few times by the optimal collection of paths, and splits only few of the point sets P_i. Then, we "guess" how the solution crosses σ, by iterating over all possibilities. By doing so, we create two subproblems: one for the points inside σ, and one for the points outside σ. However, these two subproblems are not completely independent yet: first, for every point set P_i which contains at least one point inside σ and one outside σ, we need to "guess" on which side of σ we visit a point of P_i. After solving all versions of both subproblems recursively, the so-called rank-based approach is used to efficiently find the correct combination of matchings on both sides resulting in the shortest overall valid answer.

A good separator. Our algorithm will need a so-called distance-based separator, similar to the distance-based separator introduced by De Berg et al. [3]. The properties they proved for their separator are not quite sufficient for us, though, so below we present a stronger version of their separator result. Before we can state our result, we need to introduce some terminology and notation from their paper. We denote the region of all points in $\mathbb{R}^d$ inside or on a separator σ by σ_{in}, and the region of all points in $\mathbb{R}^d$ strictly outside σ by σ_{out}. The *size* of a separator σ, denoted by $\mathit{size}(\sigma)$, is defined to be its edge length. For a separator σ and a scaling factor $t > 0$, we define $t\sigma$ to be the separator obtained by scaling σ by a factor t with respect to its center. In other words, $t\sigma$ is the separator whose center is the same as the center of σ and with $\mathit{size}(t\sigma) = t \cdot \mathit{size}(\sigma)$. Note that a separator σ induces a partition of the given point set P into two subsets, namely $P \cap \sigma_{\text{in}}$ and $P \cap \sigma_{\text{out}}$. A separator is *balanced* with respect to a set $Q \subseteq P$ if $\max(|Q \cap \sigma_{\text{in}}|, |Q \cap \sigma_{\text{out}}|) \leq \frac{4^d}{4^d+1} n$. If a separator is balanced with respect to P itself, we call it a balanced separator.

Our separator is chosen such that there are only few points close to it. To quantify this, let the *relative distance* from a point p to σ, denoted by $\mathit{rdist}(p, \sigma)$, be defined as follows:

$$\mathit{rdist}(p, \sigma) := d_\infty(p, \sigma)/\mathit{size}(\sigma),$$

where $d_\infty(p, \sigma)$ denotes the shortest ℓ_∞-distance between p and any point on σ. Recall that σ is the boundary of a hypercube; hence, all points in the interior of σ_{in} have a nonzero relative distance to σ. Note that if t is the scaling factor such that $p \in t\sigma$, then $\mathit{rdist}(p, \sigma) = |1 - t|/2$. For integers i define

$$P(i,\sigma) := \{p \in P : \mathit{rdist}(p,\sigma) \le 2^i/n^{1/d}\}.$$

Note that the smaller i is, the closer to σ the points in $P(i,\sigma)$ are required to be. De Berg et al. choose σ such that the size of the sets $P(i,\sigma)$ decrease rapidly as i decreases. Our generalised theorem also allows for some control over the number of P_i which are *split* by σ, i.e., the P_i of which at least one point of P_i is in σ_{in} and at least one point of P_i is in σ_{out}.

▶ **Theorem 1.** *Let $\mathcal{P} = \{P_1, ..., P_r\}$ be a collection of point sets in $\mathbb{R}^d$, and let $Q \subseteq P$. Let n be the total number of points in P. Then there is a separator σ that is balanced with respect to Q and such that*

$$|P(i,\sigma)| = \begin{cases} O((3/2)^i n^{1-1/d}) & \text{for all } i < 0 \\ O(4^i n^{1-1/d}) & \text{for all } i \ge 0. \end{cases}$$

Furthermore, σ splits at most $\lambda^{1/d} n^{1-1/d}$ of the point sets P_i. Moreover, such a separator can be found in $O(n^{d+1})$ time.

Proof. The proof is analogous to the proof of Theorem 1 and Corollary 3 in the paper by De Berg et al. [3], with one major difference. Instead of the weight function $w_p(t)$, we use the weight function

$$w_p^*(t) := \frac{\mathbf{1}\{t\sigma^* \text{ intersects } H_{i(p)}\}}{\mathit{size}(H_{i(p)})} + w_p(t),$$

where σ^* is the smallest balanced separator, which we assume w.l.o.g. has size 1 (as in the original proof), $i(p)$ denotes the i such that $p \in P_i$, and $\mathbf{1}\{\mathit{bool}\}$ denotes the indicator function, which is 1 if *bool* is true, and 0 otherwise. Since $\int_0^3 w_p(t) = O(1)$, we have $\int_0^3 w_p^*(t) = O(1)$ as well, because $\int_0^3 \mathbf{1}\{t\sigma^* \text{ intersects } H_{i(p)}\} \le \mathit{size}(H_{i(p)})$. Hence, we can find a t^* such that $\sum_p w_p^*(t^*) = O(n)$. Therefore, we can use this t^* to prove the bounds on $|P(i,\sigma)|$ analogously to the original proofs.

It remains to prove the bound on the number of P_i split by $t^*\sigma^*$. We get

$$\sum_i \frac{\mathbf{1}\{t^*\sigma^* \text{ intersects } H_i\}}{\mathit{size}(H_i)} \le \sum_p \frac{\mathbf{1}\{t^*\sigma^* \text{ intersects } H_{i(p)}\}}{\mathit{size}(H_{i(p)})} < \sum_p w_p^*(t) = O(n).$$

Therefore, for any L, at most $O(nL)$ different H_i of size at most L intersect $t\sigma^*$. Furthermore, an H_i of size at least L intersecting σ covers at least L^{d-1} of the $(d-1)$-dimensional volume of σ, which is $2d = O(1)$. (Recall that σ is defined as the boundary of a hypercube, which is a $(d-1)$-dimensional object, and that we consider d to be fixed.) See Figure 2. Since the H_i have a ply of λ, there can be at most λ/L^{d-1} of these. Hence, for any L, at most $O(nL + \lambda/L^{d-1})$ hypercubes intersect σ. Specifically, by taking $L = \lambda^{1/d} n^{-1/d}$, we conclude that σ intersects $O(\lambda^{1/d} n^{1-1/d})$ of the H_i. Finally, we note that the number of P_i split by σ is bounded by the number of H_i intersected by σ, finishing our proof. ◀

We will now use this distance-based separator theorem to present an efficient algorithm for EUCLIDEAN ONE-OF-A-SET TSP.

Let $S(P) := \{pq : (p,q) \in P \times P\}$ be the set of all line segments defined by P. Since we wish to guess how σ is crossed by the optimal answer, and we know that the edges of the answer have the packing property, we are interested in the following set:

$$\mathcal{C}(\sigma, P) := \{S \subseteq S(P) : S \text{ has the packing property and all segments in } S \text{ cross } \sigma\}.$$

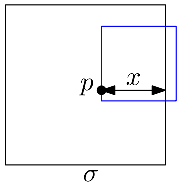

Figure 2 Example for the proof of Theorem 1. The point p is at distance x from σ. Hence, any square H (in blue) that contains both p and a point outside σ covers at least x of the total length of the edges of σ. For arbitrary d, any d-dimensional hypercube H that contains both p and a point outside σ covers at least x^{d-1} of the $(d-1)$-dimensional volume of σ. (Recall that σ is defined to be the $(d-1)$-dimensional boundary of a hypercube, so σ does not include the region enclosed by it.)

Our main separator theorem, presented next, states that we can find a separator σ that is balanced, splits few P_i, and is such that the sets in $\mathcal{C}(\sigma, P)$, as well as the collection $\mathcal{C}(\sigma, \mathcal{P})$ itself, are small. Since the packing property is hard to test, we will not enumerate $\mathcal{C}(\sigma, P)$ but a slightly larger collection of candidate sets, which we denote by $\mathcal{C}'(\sigma, P)$.

▶ **Theorem 2.** *Let $\mathcal{P} = \{P_1, ..., P_r\}$ be a family of point sets in $\mathbb{R}^d$, and let $\mathcal{H} = \{H_1, ..., H_r\}$ be a set of hypercubes such that $P_i \subset H_i$ for every i. Let $Q \subseteq P$, where $P = P_1 \cup \cdots \cup P_r$. Then there exists a separator σ with the following properties:*

1. *σ is balanced with respect to Q.*
2. *Each candidate set $S \in \mathcal{C}'(\sigma, P)$ contains $O(n^{1-1/d})$ segments.*
3. *$\mathcal{C}(\sigma, P) \subseteq \mathcal{C}'(\sigma, P)$ and $|\mathcal{C}'(\sigma, P)| = 2^{O(n^{1-1/d})}$.*
4. *σ splits $O(\lambda^{\frac{1}{d}} n^{1-1/d})$ of the sets P_i, where λ is the ply of $\mathcal{H}$.*

Moreover, σ and the collection $\mathcal{C}'(\sigma, P)$ can be computed in $2^{O(n^{1-1/d})}$ time.

Proof. The separator chosen is the one found by applying Theorem 1. Hence, the proof of properties 1-3 is analogous to that of the original paper by De Berg et al. Property 4 is directly implied by Theorem 1, as well. ◀

The algorithm. Our adapted algorithm contains four changes compared to the original:

- We choose our separator σ using Theorem 2 instead of the equivalent from the original paper. Note that this does not impact the running time.
- Candidate sets of which the endpoints of the edges contain more than one point of any set P_i can be ignored. Note that this can be easily checked.
- When "guessing" the correct candidate set, for every point set P_i split by σ we also guess whether a point of P_i in σ_{in} or a point in σ_{out} is used. If we guess that we will visit a point in $P_i \cap \sigma_{\text{in}}$, then we can ignore the points in $P_i \cap \sigma_{\text{out}}$ for the recursive call outside σ, and vice versa. Note that for every P_i that contains a boundary point this choice (if applicable) is implied by the location of the boundary point. Furthermore, once we have chosen a point from a set P_i as one of our boundary points, then we can remove all other points from P_i from further consideration. This way, the subproblems generated remain independent, while ensuring that exactly one point of every P_i is visited.
- In the initial call, the original algorithm turns the problem into a Euclidean Path Cover problem by duplicating point p_1 and taking the boundary set $B = (p_1, p_1')$. In other words, it simply searches for a path from p_1 to p_1 through all other points. In our case, we guess which p in P_1 is used in the optimal tour. Then, we remove all other points in P_1 and duplicate p as in the original algorithm.

This brings us to our main theorem for this section.

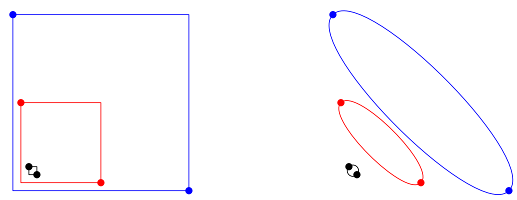

Figure 3 On the left, an example of a point set whose hypercubes generate a high ply. The pattern can be repeated to get a ply of $\Theta(n)$. However, the separator we find will not split $\Theta(n)$ point sets. On the right, the same point set covered by α-fat objects. Here, we have a ply of only 1.

▶ **Theorem 3.** *Let $\mathcal{P} = \{P_1, .., P_r\}$ be a collection of point set in $\mathbb{R}^d$ with n points in total. Let $H_1, ..., H_r$ be hypercubes with ply λ such that $P_i \subset H_i$ for all i. Then* Euclidean One-of-a-Set TSP *on $\mathcal{P}$ can be solved in $2^{O(\lambda^{\frac{1}{d}} n^{1-1/d})}$ time.*

For the full proof, see Appendix A. It follows the proof of the original algorithm [3] almost verbatim; the main difference is that we need to take the dependency on λ into account.

An improved analysis of the running time. So far, we have used the ply of hypercubes covering the point sets to bound the running time of our algorithm. (Note that the algorithm itself does not use the hypercubes, we only used their ply in the analysis to quantify how separated the sets are.) However, in some cases, these hypercubes can have a high ply, even though they are still fairly well separable. Thus the analysis may be overly pessimistic. We show that this is indeed the case by replacing the hypercubes by so-called *α-fat objects*, whose ply can be much smaller than the ply of the hypercubes. let $0 < \alpha < 1$ be arbitrary but fixed. We say a connected closed set of points $O \subseteq \mathbb{R}^d$ is an α-fat object if and only if for every ball $B \subset \mathbb{R}^d$ whose center lies in O we have that (i) O fully lies in B or (ii) at least a fraction α of the volume of B is covered by O. See Figure 3 for an example showing how α-fat objects can have significantly lower ply than hypercubes.

It remains to show that with this new λ our running time of $2^{O(\lambda^{1/d} n^{1-1/d})}$ is still accurate. Note that we use the fact that the objects containing the sets P_i are hypercubes only once, namely during the proof of Theorem 1, where we bound the number of H_i split by σ to $2^{O(\lambda^{1/d} n^{1-1/d})}$. We will now prove that a similar bound holds for arbitrary α-fat objects.

Before we define the size of an α-fat object, we need the following observation.

▶ **Observation 4.** *Let $\alpha > 0$ be arbitrary but fixed. Let O be an α-fat object. Let $\mathcal{B}$ be the bounding box of O, i.e., the smallest box such that O lies fully in $\mathcal{B}$. Then the dimensions of $\mathcal{B}$ are within a constant factor of each other.*

Proof. Let α, O and $\mathcal{B}$ be as defined above. Let s be the largest dimension of $\mathcal{B}$. W.l.o.g., $s = 1$. Let p be a point in O. W.l.o.g., p is the origin. Note that O fully lies in $[-1, 1]^d$. Let B be the ball of radius $1/3$ centered at the origin. Note that B does not fully cover O (in fact, any radius strictly smaller than $1/2$ suffices). Hence, since O is an α-fat object, at least $\alpha c_d/3^d$ of B (and therefore $[-1, 1]^d$) is covered by O, where c_d is the volume of a d-dimensional ball with radius 1. Since for any dimension x of $\mathcal{B}$, the volume of O inside $[-1, 1]^d$ can be bounded by x, we get that $x \geq \alpha c_d/3^d = O(1)$, as required. ◀

We now define the size of an α-fat object O to be the largest dimension of its bounding box.

We will now show that these objects have all required properties.

▶ **Lemma 5.** *Let $\mathcal{P} = \{P_1, ..., P_r\}$ be a collection of point sets in $\mathbb{R}^d$, and let $Q \subseteq P$. Let n be the total number of points in $\mathcal{P}$. Let σ be the separator found when applying Theorem 1 with α-fat objects instead of hypercubes. Then σ splits $O(\lambda^{1/d} n^{1-1/d})$ point sets P_i.*

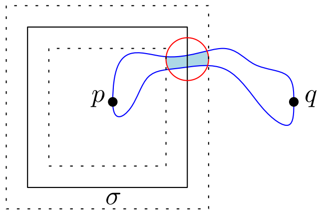

Figure 4 The dotted lines denote the square annulus A of all points with an *rdist* of at most $2^j/n^{1/d}$ to σ. In blue, the object O_i crossing σ. It has at least one of (i) a point p in σ_{in} but not in A, or (ii) a point q in σ_{out} but not in A. Therefore, if we draw a ball (in red) the width of A centered at a point where O_i coincides with σ, this circle does not fully contain O_i. Hence, the volume of O_i inside the ball (light blue), and therefore inside A, is at least α times the volume of the ball.

Proof. First, we note that the logic showing that $w_p^*(t) = O(1)$ still holds for α-fat objects. Furthermore, analogously to the original proof of Theorem 1, there are $O(\lambda^{1/d}n^{1-1/d})$ objects of size at most $\lambda^{1/d}n^{-1/d}$ intersected by σ. It remains to prove that σ splits $O(\lambda^{1/d}n^{1-1/d})$ point sets whose objects have size larger than $\lambda^{1/d}n^{-1/d}$. Note that it is sufficient to prove that $O(\lambda^{1/d}n^{-1/d})$ objects of size larger than $O(\lambda^{1/d}n^{-1/d})$ intersect σ.

Let O_i be an object of size strictly larger than $\lambda^{1/d}n^{-1/d}$ intersected by σ. Then O_i must contain a point at a distance more than $c_{\alpha,d}\lambda^{1/d}n^{-1/d}$ from σ for some constant $c_{\alpha,d}$: this distance is clearly nonzero, as O_i has a positive volume, and since α and d are fixed, the problem scales linearly.

For brevity, we write $x := c_{\alpha,d}\lambda^{1/d}n^{-1/d}$. Let B be the ball with radius x and centered at an intersection point of O_i and σ. Now, B does not fully cover O. Therefore, O covers at least a fraction α of B. Let A be the square annulus defined by $\{p \in \mathbb{R}^d : \mathit{rdist}(p, \sigma) \leq x\}$. Note that B fully in A. See Figure 4 for an example. Since B is d-dimensional, every O_i covers $\Theta(x^d)$ of the volume of A. Now, the total volume of A is smaller than $2d \cdot (1+2x)^{d-1}2x$, since each of the $2d$ facets of σ contributes $(1+2x)^{d-1}2x$ to the annulus. (This is a conservative estimate since we ignore overlap between the contributions of the facets.) Since $x < 1$, this can be further bounded by $3^{d+2}dx$. Furthermore, the volume of a d-dimensional ball with radius x is $c_d x^d$ for some constant c_d. Therefore, the maximum number of O_i intersected by σ of size strictly larger than $\lambda^{1/d}n^{-1/d}$ is bounded by

$$\lambda\frac{3^{d+2}dx}{\alpha c_d x^d} = O(\lambda x^{1-d}) = O(\lambda \cdot \lambda^{(1-d)/d}n^{-(1-d)/d}) = O(\lambda^{1/d}n^{1-1/d}),$$

as we wanted to prove. ◀

We thus obtain the following theorem.

▶ **Theorem 6.** *Let $\alpha > 0$ be arbitrary but fixed. Let $\mathcal{P} = \{P_1, .., P_r\}$ be a collection of point set in $\mathbb{R}^d$ with n points in total. Suppose there exist $O_1, ..., O_r$ be α-fat objects with ply λ such that $P_i \subset O_i$ for all i. Then* Euclidean One-of-a-Set TSP *on $\mathcal{P}$ can be solved in $2^{O(\lambda^{\frac{1}{d}}n^{1-1/d})}$ time.*

A lower bound on the running time when $\lambda = \Theta(n)$. In this section we show that for $\lambda = \Theta(n)$, the problem cannot be solved in subexponential time.

▶ **Theorem 7.** Euclidean One-of-a-Set TSP *in $\mathbb{R}^2$ cannot be solved in $2^{o(n)}$ time, unless ETH fails.*

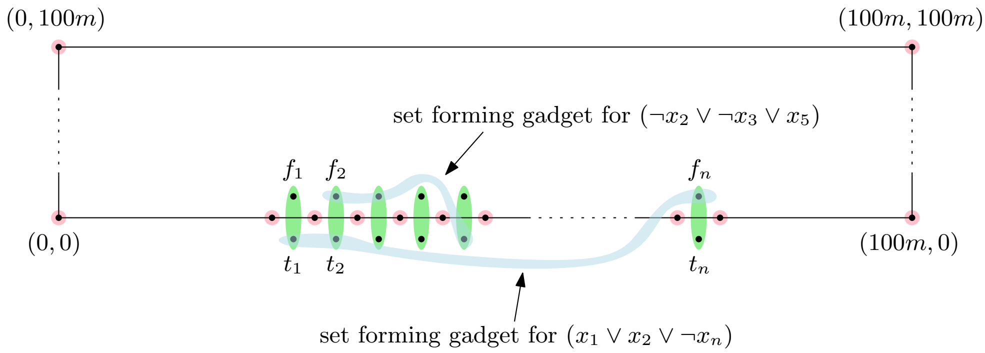

Figure 5 An example for the proof of Theorem 7. Not to scale. The points inside the small pink disks (which are the points p_i and q_i as defined in the text) must all be visited. The sets indicated by the green ellipses ensure that for each i at least one of t_i and f_i is visited. The sets indicated by the blue regions correspond to the clauses. The pink and green sets imply a lower bound on the length of the shortest tour. If this bound is tight, the shortest tour visits exactly one of each pair (t_i, f_i). Note that each such tour maps directly to an assignment of True and False to the variables. If an assignment of True and False to the variables satisfying all clauses exists, then the corresponding tour indeed visits all sets at least once.

Proof. The ETH states that 3-SAT cannot be solved in $2^{o(n)}$ time [9]. We will prove Theorem 7 by showing that if Euclidean One-of-a-Set TSP can be solved in $2^{o(n)}$ with $d = 2$ and $\lambda = \Theta(n)$, then 3-SAT can be solved in $2^{o(n)}$ time as well.

Let F be a 3-SAT formula containing clauses $C_1, \ldots, C_n$ over variables $x_1, \ldots, x_m$. Note that $m = O(n)$. We define $p_1 = (0, 0)$, $p_2 = (0, 100m)$, $p_3 = (100m, 100m)$, and $p_4 = (100m, 0)$. Furthermore, let $q_i = (50m - 2i, 0)$ for all $i = 0, \ldots, m$, let $t_i = (50m - 2i + 1, -1)$ and let $f_i = (50m - 2i + 1, 1)$ for all $i = 1, \ldots, m$. See Figure 5 for an example. Now, let $\mathcal{P}$ be the family containing the following point sets:

- For all $0 \le i \le 3$, one point set containing only p_i.
- For all $1 \le i \le m$, one point set containing t_i and f_i. This is the gadget representing the variable x_i.
- For all $1 \le i \le n$, one point set representing the clause C_i. Specifically, this point set should contain t_j iff C_i contains the literal x_j, and should contain f_j iff C_j contains the literal $\neg x_j$. Note that each of these point sets contains three elements, and that each point in a clause gadget coincides with a point from a variable gadget.

Now we claim that F is satisfiable if and only if $\mathcal{P}$ admits a shortest tour of length exactly $L := (398 + 2\sqrt{2})m$. Note that if this claim indeed holds, we are done; if we can solve Euclidean One-of-a-Set TSP in $2^{o(n)}$ time, then the shortest tour on $\mathcal{P}$ can be found in $2^{o(n+m)} = 2^{o(n)}$ time, and therefore we answer whether F is satisfiable in $2^{o(n)}$ time.

Note that if a shortest tour of length L exists, F is satisfiable. To satisfy F, simply set x_i to True if t_i is visited in the shortest tour, and set it to False otherwise. Since the shortest tour visits every point set corresponding to a clause, all clauses are indeed satisfied.

Next, we check that if F is satisfiable, a shortest tour of length L exists. We simply do the reverse: let $x_1, \ldots, x_m$ satisfy F. Then note that the tour passing through t_i if x_i is True and through f_i otherwise indeed has the required length. Finally, we still have to show the shortest tour can never be shorter than L. It is easy to see that the shortest tour must visit $q_m, p_1, p_2, p_3, p_4, q_0$ consecutively in that order, giving a length of $398m$. Connecting q_0 to q_m while passing through at least one of every pair (t_i, f_i) and through every q_i inbetween takes a total length of at least $2\sqrt{2}m$. Therefore, the shortest tour has length at least $(398 + 2\sqrt{2})m$.

In conclusion, F is satisfiable if and only if the shortest tour on $\mathcal{P}$ has length L. Therefore, if Euclidean One-of-a-Set TSP can be solved in subexponential time when $\lambda = \Theta(n)$, then so can 3-SAT. Hence, unless ETH fails, Euclidean One-of-a-Set TSP cannot be solved in subexponential time. ◀

3 Rectilinear One-of-a-Cube TSP

We continue with Rectilinear One-of-a-Cube TSP. Recall that for this setting, $\mathcal{H} := \{H_1, ..., H_n\}$ is a set of hypercubes, and λ is the ply of $\mathcal{H}$, i.e., the smallest number such that every point in $\mathbb{R}^d$ is in at most λ of the hypercubes. We now want to find a minimum-length rectilinear tour visiting all of the hypercubes H_i.

The algorithm works using the same divide-and-conquer approach as the Euclidean One-of-a-Set TSP algorithm; see the beginning of Section 2 for a more detailed description.

Properties of an optimal tour. We start by limiting the set of points and edges we need to consider. We show that there is an optimal tour using only edges from a specific set, and that these edges have the packing property.

First, we introduce some terminology. An *edge* is defined as a rectilinear line segment. A *link* between two points is any shortest path formed by at most d edges of different orientations (which always exists). We define, with slight abuse of notation, $|pq|$ to be the L_1-distance between points p and q. Note that this is also the length of a link between p and q. A *tour* is a sequence of links, where the endpoint of each link in the sequence is the starting point of the next one, and the endpoint of the last link is the startpoint of the first link. Note that the fact that we see the tour as a sequence (and not as a cycle) implies that tours have a starting point and a direction – this is solely for the purpose of the analysis.

Let q^i denote the i'th coordinate of a point q. Define C to be the set of $2^d n$ corners of the cubes in the input set $\mathcal{H}$. Let G be the *generalised Hanan grid* induced by the set C which is defined as the grid formed by drawing all axis-aligned lines through every point in the point set $C^* := \{p \in \mathbb{R}^d : \forall 1 \leq i \leq d : \exists c \in C : p^i = c^i\}$. In other words, the lines of the grid G are the intersections of $d-1$ differently oriented hyperplanes each containing the facet of one of the hypercubes.

▶ **Lemma 8.** *There exists a shortest tour on $\mathcal{H}$ which lies fully on G.*

For the full proof, see Appendix B. Intuitively, this can be done by taking any shortest tour T and "shifting" it onto the grid, bit by bit.

Given a tour T, we can reorder the hypercubes in $\mathcal{H}$ such that H_i is the i'th hypercube visited by T; ties can be broken in any way. Define p_i to be the first point where H_i is visited by T. Define $P(T) := \{p_1, ..., p_n\}$; we call the points in $P(T)$ the *entry points* of T. Note that the length of a shortest tour T that visits the points p_i in the given order equals $\sum_{1 \leq i \leq n} |p_i p_{i+1}|$, where we define $p_{n+1} := p_1$. (Recall that $|pq|$ denotes the L_1-distance from p to q.) We say a tour T is a *canonical tour* on $\mathcal{H}$ if it has the following properties:

1. T is a shortest tour on $\mathcal{H}$
2. T lies fully on G
3. Each pair of consecutive entry points in $P(T)$ is connected by a link, that is, the portion of T connecting consecutive entry points consist of at most d edges of different orientations.

▶ **Observation 9.** *For every $\mathcal{H}$, a canonical tour on $\mathcal{H}$ exists.*

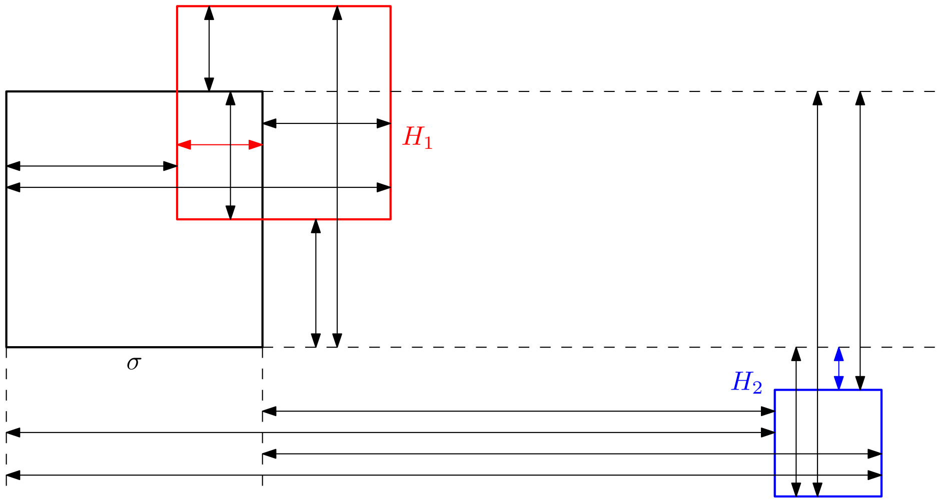

Figure 6 An example of *odrdist*. Given an H_i, in every direction e_j the four distances between one of the sides of H_i perpendicular to e_j and one of the sides of σ perpendicular to e_j are measured. The shortest of all the measured distances, scaled such that $\text{size}(\sigma) = 1$, defines the *odrdist* between the two. Note how H_1, in red, intersects σ and H_2, in blue, seems far away from σ. Yet, the *odrdist* of H_1, denoted by the red arrow, is larger than that of H_2, denoted by the blue arrow.

Proof. By Lemma 8 there exists a shortest tour T which lies fully on G, satisfying the first two properties. Now, we can create a new tour T' by creating a link between every two consecutive p_i in $P(T)$. Since T' is a shortest tour on a set of points on T, it must be a shortest tour itself as well. Furthermore, since T' visits all p_i in $P(T)$, it visits all H_i. Finally, since T lies on the generalised Hanan grid G, so do the points p_i. Furthermore, note that a link between two points on G lies on G itself. We conclude that T' has the required properties. ◀

▶ **Lemma 10.** *The edges of a canonical tour have the Packing Property.*

For the full proof, see Appendix C. Intuitively, suppose we have two long edges in the same direction (e.g. left to right) and close to each other. We can then replace these edges by two new edges – one connecting the two starting points of the removed edges and one connecting the end points – whose total length is shorter.

A good separator. As mentioned, our algorithm will be a divide-and-conquer algorithm, based on separators. Thus we need a good separator for tours on hypercubes. Our separator will again be based in the distance-based separator from [3]. It will not be sufficient to work with a distance-based separator on the corners of the hypercubes. Instead, we want to have only a few H_i with a facet close to one of the parallel facets of σ, measured in the dimension perpendicular to these facets. To be precise, for a hypercube H, let $\text{center}(H)$ be its center and $\text{size}(H)$ its edge length. Let the *one-dimensional distance* from a hypercube H to a separator σ, denoted by $\mathit{oddist}(H, \sigma)$ be defined as the minimum distance between any pair of parallel hyperplanes h, h' such that h contains a facet of H and h' contains a facet of σ. The *one-dimensional relative distance* from H to σ, denoted by $\mathit{odrdist}(H, \sigma)$ is now defined as $\mathit{odrdist}(H, \sigma)/\mathit{size}(\sigma)$. See Figure 6 for an example. For integers j define

$$P_j(\sigma) := \{H \in \mathcal{H} : 0 < \mathit{odrdist}(H, \sigma) \leq 2^j/n^{1/d}\}.$$

We can now prove the following theorem.

▶ **Theorem 11.** *Let $\mathcal{H}$ be a set of n hypercubes in $\mathbb{R}^d$ and let $\mathcal{I} \subseteq \mathcal{H}$. Then there is a separator σ that is balanced with respect to the corner points of $\mathcal{I}$, and such that*

$$|P_j(\sigma)| = \begin{cases} O((3/2)^j n^{1-1/d}) & \text{for all } j < 0 \\ O(4^j n^{1-1/d}) & \text{for all } 0 \le j < \infty. \end{cases}$$

Furthermore, at most $O(\lambda^{1/d} n^{1-1/d})$ elements of $\mathcal{H}$ intersect σ. Moreover, such a separator can be found in $O(n^{d+1})$ time.

Proof. Let C be the set of corner points of the hypercubes in $\mathcal{I}$. Let σ^* be a smallest separator such that $|\sigma^*_{\text{in}} \cap C| \ge (4n)/(4^d + 1)$. As in the original proof, one can argue that for all $1 \le t \le 3$, the separator $t\sigma^*$ is balanced w.r.t. C. Assume w.l.o.g. that $\mathit{size}(\sigma^*) = 1$. Define $j_H(t)$ to be the integer such that

$$2^{j_H(t)-1}/n^{1/d} < \mathit{odrdist}(H, t\sigma^*) \le 2^{j_H(t)}/n^{1/d},$$

where $j_H(t) = \infty$ if $\mathit{odrdist}(H, t\sigma^*) = 0$. We define the weight function as

$$w_H(t) := \frac{\mathbf{1}\{H \text{ intersects } t\sigma^*\}}{\mathit{size}(H)} + \begin{cases} \frac{n^{1/d}}{(3/2)^{j^i_H(t)}} & \text{if } j^i_H(t) < 0 \\ \frac{n^{1/d}}{4^{j^i_H(t)}} & \text{otherwise } 0 \le j^i_H(t) < \infty \\ \text{undefined} & \text{otherwise.} \end{cases}$$

(Recall that $\mathbf{1}\{\mathit{bool}\}$ denotes the indicator function, which is 1 if *bool* is true, and 0 otherwise.) Now, for every $H \in \mathcal{H}$ we have $\int_1^3 w_H(t)dt = O(1)$, since the second part of $w_H(t)$ can be expressed as the maximum of $2d$ different versions of the weight function $w_p(t)$ of the original proof. Since H can obviously intersect $t\sigma$ only during an interval of t of size size(H), we get that $\int_1^3 w_H(t)dt = O(1)$.

Therefore, we can find a t^* such that $\sum_H w_H(t^*) = O(n)$. We claim that $t^*\sigma^*$ has the desired properties. We have already shown that it is balanced w.r.t. the corner points of $\mathcal{I}$.

Let $1 \le i \le d$, and let $j < 0$. Then each element in $P^i_j(\sigma)$ contributes at least $w^i_H(t^*) \le \frac{n^{1/d}}{(3/2)^j}$ to the total weight. Therefore, there are at most $O(n/\frac{n^{1/d}}{(3/2)^j}) = O(n^{1-1/d}(3/2)^j)$ such elements, as required. The case for $0 \le j < \infty$ can be proven analogously.

Finally, we note that at most $O(2d\lambda/(\lambda^{1/d}n^{-1/d})^{d-1}) = O(\lambda^{1/d}n^{1-1/d})$ of the H_i of size at least $\lambda^{1/d}n^{-1/d}$ can intersect σ (otherwise, somewhere, $\lambda+1$ would overlap). Furthermore, there are $O(\frac{n}{1/(\lambda^{1/d}n^{-1/d})}) = O(\lambda^{1/d}n^{1-1/d})$ of the H_i of size at most $\lambda^{1/d}n^{-1/d}$ that intersect σ, as they all contribute weight at least $1/(\lambda^{1/d}n^{-1/d})$. Therefore, there are $O(\lambda^{1/d}n^{1-1/d})$ hypercubes intersecting σ.

As in the original proof, we can argue that we can compute $t^*\sigma^*$ quickly by truncating $w^i_H(t)$. ◀

This brings us to the candidate sets. Instead of guessing how we cross the separators precisely, guessing *where* we cross the separators will suffice. For simplicity, we consider the boundary points created this way to be infinitely small hypercubes.

▶ **Theorem 12.** *Let $\mathcal{H} = \{H_1, ..., H_n\}$ be a set of n hypercubes in $\mathbb{R}^d$ and let $\mathcal{I} \subseteq \mathcal{H}$. Then there is a separator σ and a collection $C'(\sigma, \mathcal{H})$ of candidate point sets such that*

1. *σ is balanced with respect to the corner points of $\mathcal{I}$*
2. *Each candidate set $X \in C'(\sigma, H)$ contains $O(n^{1-1/d})$ points.*
3. *There exists a shortest tour T and an $X \in C'(\sigma, \mathcal{H})$ such that X is the set of locations where T crosses σ, and $|C'(\sigma, \mathcal{H})| \le 2^{O(n^{1-1/d} \log n)}$*
4. *σ splits $O(\lambda^{1/d}n^{1-1/d})$ of the H_i, where λ is the ply of $\mathcal{H}$.*

Moreover, σ and $C'(\sigma, \mathcal{H})$ can be calculated in $2^{O(n^{1-1/d}\log n)}$ time.

Proof. Let σ be the separator given by Theorem 11. Then σ has properties 1 and 4. W.l.o.g., we assume that size$(\sigma) = 1$ and σ is centered at the origin. From now on, we will only consider edges that cross σ once and lie on the Hanan grid G. Any set $S \in C'(\sigma, \mathcal{H})$ we return can be divided into two subsets:

- $S_{short} := \{s \in S : length(s) \leq 1/n^{1/d}\}$
- $S_{long} := \{s \in S : length(s) > 1/n^{1/d}\}$

(The original proof uses three subsets: S_{short}, S_{mid} and S_{long}. However, since we have an extra factor $\log n$ in the exponent, we can merge S_{short} with part of S_{mid}, and merge the rest of S_{mid} with S_{long}.) We start with S_{short}. Let us take a look at a single facet of σ. We will now show that we can cross this facet in only a limited number of ways. First, we note that for the corresponding i, we have $|P_0^i(\sigma)| = O(n^{1-1/d})$. Let e be an arbitrary edge of our tour crossing σ through this face. Now, by property (3) of a canonical tour, the p_j and p_{j+1} that are connected by e must both have a distance at most $1/n^{1/d}$ to σ in the i'th coordinate. Therefore, the same holds of the odrdistances of the corresponding H_j. Furthermore, note that if we charge every edge e crossing σ through our facet to the two hypercubes of the corresponding p_j and p_{j+1}, no hypercube is charged more than twice. Hence, the number of short edges crossing σ through this facet is bounded by the number of hypercubes with odrdistance at most $1/n^{1/d}$ to σ, of which there are $O(n^{1-1/d})$. As there are $O(n^{d-1})$ possible locations for these edges to cross σ, there are $(n^{d-1})^{O(n^{1-1/d})} = 2^{O(n^{1-1/d}\log n)}$ possible combinations for every face. Finally, since there are $O(d)$ facets, the total amount of possible combinations is $2^{O(n^{1-1/d}\log n)}$.

We continue to S_{long}, the set of edges longer than $1/n^{1/d}$. Let us take a look at those edges which cross some arbitrary facet of σ. Using the same logic as in the original paper, by using Lemma 10, we can see that there are $(2n^{1/d})^{d-1} = O(n^{1-1/d})$ of these edges at most. Analogously, there are at most $O(n^{1-1/d})$ edges of which at least $1/(2n^{1/d})$ is inside σ. Since there are $O(n^{d-1})$ options for every edge, there are $(n^{d-1})^{O(n^{1-1/d})} = 2^{O(n^{1-1/d}\log n)}$ options for every face, and just as many for every σ. ◀

The algorithm. Our algorithm contains the following changes compared to the original:

- We choose our separator σ using Theorem 11 instead of the equivalent from the original paper, and use Theorem 12 to obtain the candidate sets of crossing points.
- Instead of choosing already existing points as boundary points, we create new boundary points as explained above. To ensure that the recursion ends, we bruteforce the solution if n is smaller than some arbitrarily large but fixed N, instead of recurring until $n = 1$.
- All H_i visited by one of the new boundary points are removed from both subproblems. For every H_i split by σ but not visited by one of the new boundary points, we need to "guess" whether it is visited inside or outside σ.
- For the initial call, we guess p_1 and p_n of the final tour, and connect them with a link - there are $O(n^{2d})$ viable combinations of points on the generalised Hanan grid. We remove all H_i we already visit by doing so, and then run the algorithm on the remaining H_i and the boundary set $B = \{p_1, p_n\}$.

Since there are $O(\lambda^{1/d})$ hypercubes H_i split by σ, there are $2^{O(\lambda^{1/d}n^{1-1/d}\log n)} \cdot 2^{O(\lambda^{1/d})} = 2^{O(\lambda^{1/d}n^{1-1/d}\log n)}$ subproblems generated in total, leading to the following theorem. (For the full proof, see Appendix D, where we show how to analyze the dependency on λ, and deal with the larger amount of candidate sets and the creation of extra boundary points.)

▶ **Theorem 13.** *Then Rectilinear One-of-a-Cube TSP on hypercubes with ply λ can be solved in $2^{O(\lambda^{\frac{1}{d}}n^{1-1/d}\log n)}$ time.*

References

1 Binay Bhattacharya, Ante Ćustić, Akbar Rafiey, Arash Rafiey, and Vladyslav Sokol. Approximation algorithms for generalized mst and TSP in grid clusters. In Zaixin Lu, Donghyun Kim, Weili Wu, Wei Li, and Ding-Zhu Du, editors, *Combinatorial Optimization and Applications*, pages 110–125, Cham, 2015. Springer International Publishing.

2 T.-H. Hubert Chan and Shaofeng H.-C. Jiang. Reducing curse of dimensionality: Improved PTAS for TSP (with neighborhoods) in doubling metrics. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016*, pages 754–765. SIAM, 2016. doi:10.1137/1.9781611974331.ch54.

3 Mark de Berg, Hans L. Bodlaender, Sándor Kisfaludi-Bak, and Sudeshna Kolay. An ETH-tight exact algorithm for euclidean TSP. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 450–461, 2018. doi:10.1109/FOCS.2018.00050.

4 Mark de Berg, Joachim Gudmundsson, Matthew J. Katz, Christos Levcopoulos, Mark H. Overmars, and A. Frank van der Stappen. TSP with neighborhoods of varying size. *Journal of Algorithms*, 57(1):22–36, 2005. doi:10.1016/j.jalgor.2005.01.010.

5 Moshe Dror and James B. Orlin. Combinatorial optimization with explicit delineation of the ground set by a collection of subsets. *SIAM Journal on Discrete Mathematics*, 21(4):1019–1034, 2008. doi:10.1137/050636589.

6 M. R. Garey, Ronald L. Graham, and David S. Johnson. Some NP-complete geometric problems. In *STOC*, pages 10–22. ACM, 1976.

7 Gregory Gutin and Abraham P. Punnen. *The Traveling Salesman Problem and Its Variations*. Springer, 2006.

8 R. Z. Hwang, R. C. Chang, and Richard C. T. Lee. The searching over separators strategy to solve some NP-hard problems in subexponential time. *Algorithmica*, 9(4):398–423, 1993. doi:10.1007/BF01228511.

9 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

10 Viggo Kann. *On the approximability of NP-complete optimization problems*. PhD thesis, Royal Institute of Technology Stockholm, 1992.

11 Michael Khachay and Katherine Neznakhina. Approximation algorithms for generalized TSP in grid clusters. *CEUR Workshop Proceedings*, 1623:39–48, 2016.

12 Michael Khachay and Katherine Neznakhina. Complexity and approximability of the euclidean generalized traveling salesman problem in grid clusters. *Annals of Mathematics and Artificial Intelligence*, 88(1):53–69, 2020. doi:10.1007/s10472-019-09626-w.

13 Joseph S. B. Mitchell. A PTAS for TSP with neighborhoods among fat regions in the plane. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007*, pages 11–18, 2007.

14 Joseph S.B. Mitchell. A constant-factor approximation algorithm for TSP with pairwise-disjoint connected neighborhoods in the plane. In *Proceedings of the Twenty-Sixth Annual Symposium on Computational Geometry*, pages 183–191, 2010. doi:10.1145/1810959.1810992.

15 Christos H. Papadimitriou. The Euclidean traveling salesman problem is NP-complete. *Theor. Comput. Sci.*, 4(3):237–244, 1977.

16 Warren D. Smith and Nicholas C. Wormald. Geometric separator theorems & applications. In *FOCS*, pages 232–243. IEEE Computer Society, 1998. doi:10.1109/SFCS.1998.743449.

A Running time of the One-of-a-Set TSP algorithm

To prove the running time of the One-of-a-Set TSP algorithm, we can follow the proof of the running time of the the original algorithm [3] almost verbatim. We only need to take the dependency on λ into account at the right places. Define $T(n, b)$ to be the running time of the algorithm when run on an input containing n points of which b are boundary points. Let $n_{S,in}$, $n_{S,out}$, $b_{S,in}$ and $b_{S,out}$ denote the numbers of points and boundary points in the subproblems generated.

As far as the candidate sets are concerned, we can restrict our attention to candidate sets $S \in \mathcal{C}'(\sigma, P)$ that contain at most one edge incident to any given point in B, and at most two edges incident to any given point in $P \setminus B$. Define $\mathcal{C}''(\sigma, \mathcal{P})$ to be the family of candidate sets gained from restricting $\mathcal{C}'(\sigma, P)$ this way. Furthermore, given a candidate set and a boundary set, for every point set split by σ we need to choose whether to use a point in σ_{in} or σ_{out}. Let $I(\sigma, \mathcal{P})$ be the set of 2^m possible combinations of choices, where m is the number of point sets split by σ. Clearly, if one of the points of a P_i is in B, no choice needs to be made. Define $\mathcal{C}^*(\sigma, \mathcal{P}) \subseteq \mathcal{C}''(\sigma, P) \times I(\sigma, \mathcal{P})$ to be the set of combinations of candidate sets and splitting choices restricted this way.

We get:

$$T(n,b) \leq \begin{cases} c_0 & \text{if } n \leq 1 \\ \sum_{S \in \mathcal{C}^*(\sigma_\mathcal{P}, \mathcal{P})} e^{c_3(n^{1-1/d}+b)} + T(n_{S,in}, b_{S,in}) + T(n_{S,out}, b_{S,out}) & \text{if } b \leq \gamma n^{1-1/d} \\ \sum_{S \in \mathcal{C}^*(\sigma_B, \mathcal{P})} e^{c_3(n^{1-1/d}+b)} + T(n_{S,in}, b_{S,in}) + T(n_{S,out}, b_{S,out}) & \text{if } b > \gamma n^{1-1/d}. \end{cases}$$

We prove by induction that $T(n,b) \leq e^{\lambda^{\frac{1}{d}}(d_1 n^{1-1/d} + d_2 b)}$ for some constants d_1 and d_2 and for all $1 \leq b \leq n$. This clearly holds for $b, n \leq 1$, so by induction, for each S we have

$$\begin{aligned} & e^{c_3(n^{1-1/d}+b)} + T(n_{S,in}, b_{S,in}) + T(n_{S,out}, b_{S,out}) \\ \leq\ & e^{c_3(n^{1-1/d}+b)} + e^{\lambda^{\frac{1}{d}}(d_1 n_{S,in}^{1-1/d} + d_2 b_{S,in})} + e^{\lambda^{\frac{1}{d}}(d_1 n_{S,out}^{1-1/d} + d_2 b_{S,out})}. \end{aligned}$$

Let c_2, c_4 and c_5 be the constants from part (2), (3) and (4) of Theorem 2. For any $S \in \mathcal{C}^*(\sigma_B, \mathcal{P})$, we have $b_{S,in} \leq \delta b + c_2 n^{1-1/d}$ and $b_{S,out} \leq \delta b + c_2 n^{1-1/d}$; similarly, for any $S \in \mathcal{C}^*(\sigma_\mathcal{P}, \mathcal{P})$, we have $n_{S,in} \leq \delta n$ and $n_{S,out} \leq \delta n$. In the remaining cases, we can just use the trivial bounds $b_{S,\cdot} \leq b + c_2 n^{1-1/d}$ and $n_{S,\cdot} \leq n$. Since $|\mathcal{C}^*(\sigma, \mathcal{P})| \leq e^{c_4 \lambda^{\frac{1}{d}} n^{1-1/d}}$, we get the following:

$$T(n,b) \leq \begin{cases} c_0 & \text{if } n \leq 1 \\ \sum_{S \in \mathcal{C}^*(\sigma_\mathcal{P}, \mathcal{P})} e^{c_3(n^{1-1/d}+b)} + T(n_{S,in}, b_{S,in}) + T(n_{S,out}, b_{S,out}) & \text{if } b \leq \gamma n^{1-1/d} \\ \sum_{S \in \mathcal{C}^*(\sigma_B, \mathcal{P})} e^{c_3(n^{1-1/d}+b)} + T(n_{S,in}, b_{S,in}) + T(n_{S,out}, b_{S,out}) & \text{if } b > \gamma n^{1-1/d}. \end{cases}$$

$$T(n,b) \leq \begin{cases} c_0 & \text{if } n \leq 1 \\ e^{c_4 \lambda^{\frac{1}{d}} n^{1-1/d}} \left(e^{c_3(n^{1-1/d}+b)} + 2e^{\lambda^{\frac{1}{d}}(d_1(\delta n)^{1-1/d} + d_2(b + c_2 n^{1-1/d}))} \right) & \text{if } b \leq \gamma n^{1-1/d} \\ e^{c_4 \lambda^{\frac{1}{d}} n^{1-1/d}} \left(e^{c_3(n^{1-1/d}+b)} + 2e^{\lambda^{\frac{1}{d}}(d_1 n^{1-1/d} + d_2(\delta b + c_2 n^{1-1/d}))} \right) & \text{if } b > \gamma n^{1-1/d}. \end{cases}$$

For simplicity, let $\lambda' := \lambda^{\frac{1}{d}}$, and Let $c := \max\{c_0, c_2, c_3, c_4\}$. We get

$$T(n,b) \leq \begin{cases} c & \text{if } n \leq 1 \\ e^{c\lambda' n^{1-1/d} + c(n^{1-1/d}+b) + 1 + \lambda'(d_1(\delta n)^{1-1/d} + d_2(b + c n^{1-1/d}))} & \text{if } b \leq \gamma n^{1-1/d} \\ e^{c\lambda' n^{1-1/d} + c(n^{1-1/d}+b) + 1 + \lambda'(d_1 n^{1-1/d} + d_2(\delta b + c n^{1-1/d}))} & \text{if } b > \gamma n^{1-1/d}. \end{cases}$$

$$T(n,b) \leq \begin{cases} c & \text{if } n \leq 1 \\ e^{\lambda'\left((2c + cd_2 + d_1 \delta^{1-1/d}) n^{1-1/d} + (d_2 + 2c) b\right)} & \text{if } b \leq \gamma n^{1-1/d} \\ e^{\lambda'\left((2c + cd_2 + d_1) n^{1-1/d} + (d_2 \delta + 2c) b\right)} & \text{if } b > \gamma n^{1-1/d}. \end{cases}$$

Now, for the case $b > \gamma n^{1-1/d}$, we have

$$e^{\lambda'\left((2c + cd_2 + d_1) n^{1-1/d} + (d_2 \delta + 2c) b\right)} \leq e^{\lambda'\left(d_1 n^{1-1/d} + d_2 b\right)}$$

if and only if

$$(2c + cd_2)n^{1-1/d} + (d_2(\delta - 1) + 2c)b \leq (2c + d_2(c - \gamma(1 - \delta)) + 2c\gamma)n^{1-1/d} \leq 0.$$

We choose $\gamma = \frac{2c}{1-\delta}$ and $d_2 = 2 + 2\gamma$, satisfying the equation.

For the case $b \leq \gamma n^{1-1/d}$, we have

$$e^{\lambda'\left((2c+cd_2+d_1\delta^{1-1/d})n^{1-1/d}+(d_2+2c)b\right)} \leq e^{\lambda'(d_1 n^{1-1/d}+d_2 b)}$$

if and only if

$$(2c + cd_2 + d_1(\delta^{1-1/d} - 1))n^{1-1/d} + 2cb \leq (2c + cd_2 + d_1(\delta^{1-1/d} - 1) + 2c\gamma)n^{1-1/d} \leq 0.$$

We choose $d_1 = \frac{4c(1+\gamma)}{1-\delta^{1-1/d}}$, satisfying the equation.

Finally, we note that d_1 and d_2 are indeed (nonnegative) constants, as they only depend on c, γ and δ, which in turn only depend on d.

B Proof of Lemma 8

Let T be any shortest tour with a minimal number of edges on the given hypercubes $H_1, ..., H_n$. Note that because T has a minimal number of edges, no edges have length 0. Recall that p^i denotes the i'th coordinate of p. For simplicity, let us call the first coordinate the *x-coordinate*, and let us call those edges whose endpoints have different x-coordinates *horizontal*. For every H_i, let x_{i1} and x_{i2} denote the x-coordinates of the corner points of H_i. We will now show that we can change T into a shortest tour of which all x-coordinates of the endpoints of the edges used are in the set $X_H := \{x_{ij} | i \in \{1, ..., n\}, j \in \{1, 2\}\}$. Furthermore, we do so without changing the sets used for the second to d'th coordinate. Then, by applying this method repeatedly, we obtain a shortest tour of which all coordinates match those of the corner points of the H_i, i.e., a shortest tour which lies on the generalised Hanan grid.

Let $X_T := \{x_1 < ... < x_r\}$ be the set of x-coordinates used by T. For every x_i, let E_i be the set of horizontal edges of with an endpoint with x-coordinate x_i.

Let x_i be an x-coordinate not in X_H. Then let e_1 and e_2 be two consecutive edges in E_1 (consecutive as in there are no edges in E_1 in between e_1 and e_2 in T). Let E denote the set of edges between e_1 and e_2 in T. Note that all endpoints of these edges have x-coordinate x_i. W.l.o.g., let at least one of endpoints of e_1 and e_2 lie to the right of x_i. Let x' denote the smallest x-coordinate in $X_H \cup X_T$ strictly larger than x_i.

Now, let us change the x-coordinate of all edges in E to x'. Furthermore, we move the endpoints of e_1 and e_2 with x-coordinate x_i to the x-coordinate x'. See Figure 7 for an example. Note that the resulting tour T' is indeed still a tour. Furthermore, T' visits all H_i: Let p be an arbitrary point in H_i visited by T but not by T'. Note that this is only possible if $x_i \leq p^0 x'$. However, in that case, the point $p' = (x', p^1, ..., p^n)$ is in H_i as well, and p' is visited by T'.

Now, if both other endpoints of e_1 and e_2 are on the same side of the hyperplane defined by x-coordinate x_i, the resulting tour T' is strictly shorter than T. Since T is a shortest tour, we conclude that the other endpoints of e_1 and e_2 are on different sides of the hyperplane defined by x-coordinate x_i. Furthermore, note that the edge that has been shortened still has a positive length, otherwise the assumption that T has a minimal number of edges fails. Finally, note that this change does not change the set of all other coordinates used except the x-coordinates.

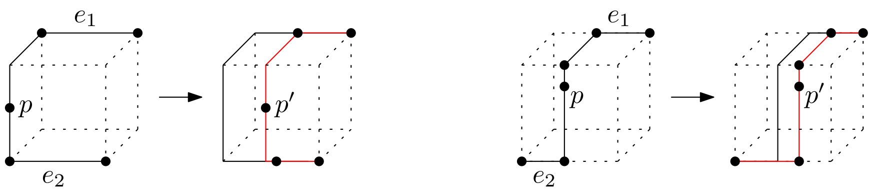

Figure 7 An example for the proof of Lemma 8. In the left case, we can shorten the tour T by shortening the edges e_1 and e_2 and moving the connected edges correspondingly. As long as the x-coordinate of these points was not a coordinate in X_H, if a point p is in a hypercube, then so is p'. In the right case, we are free to move the edges between e_1 and e_2 to the smallest x-coordinate bigger than x_i, as long as their x-coordinate is not a coordinate in X_H.

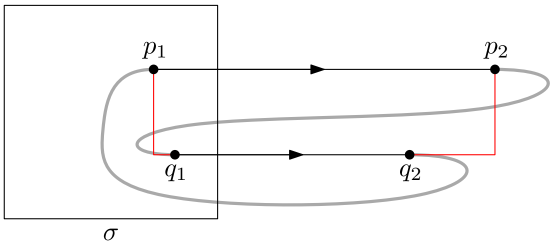

Figure 8 An example for the proof of Lemma 10. If there are enough (directed) edges of length at least $size(\sigma)$ crossing σ, there must be two edges (p_1, p_2) and (q_1, q_2) going in the same direction, crossing the same face, both with at least a length of $size(\sigma)/2$ on the same side of this face, and with $|p_1q_1| < size(\sigma)/2$. We can then create a strictly shorter tour by removing both edges and connecting p_1 to q_1 and q_2 to p_2 (in red). The resulting set of edges is indeed a tour, if we flip the direction of the edges between p_2 and q_1.

Now, we can apply the above change exhaustively: in every step, we increase the sum of all x-coordinates of all endpoints of all edges in T by at least some amount dependent only on $X_H \cup X_T$, and the total sum is bounded by $2n$ times the maximum x-coordinate in $X_H \cup X_T$. After applying this change exhaustively, no more x-coordinates not in X_H are used.

Since this procedure does not change the set of other coordinates used, we can apply this procedure once for every coordinate, obtaining a T' which lies on the generalised Hanan grid.

C Proof of Lemma 10

Let T be a shortest (directed) rectilinear tour on the hypercubes $H_1, ..., H_n$. Let σ be a separator. We will first show that T contains $O(1)$ edges crossing σ of length at least $size(\sigma)$. W.l.o.g., assume $size(\sigma) = 1$. Now, suppose T contains at least $2d \cdot 4 \cdot 4^d$ edges crossing σ of length at least $size(\sigma)$. Then there exists a face f of σ such that at least $4 \cdot 4^d$ edges cross f. W.l.o.g., at least $2 \cdot 4^d$ edges cross f from σ_{in} to σ_{out}. W.l.o.g., at least 4^d of these edges have length at least $1/2$ outside σ. Therefore, there must be two of these edges (p_1, p_2) and (q_1, q_2) with $|p_1, q_1| < 1/2$. Recall that $|pq|$ denotes the rectilinear distance between p and q. See Figure 8 for an example. Let us remove these two edges, and connect p_1 to q_1 and q_2 to q_2. Next, we flip the direction of the edges from p_2 to q_1. We claim that the resulting tour T' is a strictly shorter rectilinear tour visiting all H_i. Since this directly contradicts our assumption, we can then conclude that T contains $O(1)$ edges crossing σ of length at least $size(\sigma)$.

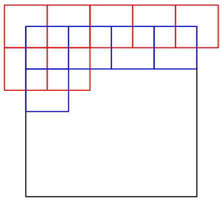

Figure 9 An example for the proof of Lemma 10. In black, the separator σ. In red and blue, some of the smaller hypercubes covering σ_{in}. Any edge of length at least $size(\sigma)/4$ crosses at least one of the smaller hypercubes. Since there are $O(1)$ smaller hypercubes, each being crossed $O(1)$ time,s there are $O(1)$ edges of length at least $size(\sigma)/4$ of T in σ_{in}.

First, we note that T' is indeed a tour: see Figure 8 for an example. Furthermore, T' indeed visits all H_i: since T is a simple tour, any tour visiting all endpoints of the edges of T (and hence, the points $p_1, ..., p_n$) visits all H_i. Finally, T' is strictly shorter than T: clearly,

$$||T|| - ||T'|| = |p_1p_2| + |q_1q_2| - |p_1q_1| - |p_2q_2|.$$

W.l.o.g., let $|p_1p_2| \geq |q_1q_2|$. We know that $|p_1p_2 \geq |q_1q_2| \geq 1$. Furthermore, we know that $|p_1q_1| < 1/2$. Since p_1p_2 and q_1q_2 both go in the same direction, we get

$$|p_2q_2| \leq |p_1q_1| + |p_1p_2| - |q_1q_2|.$$

Combining these, we get

$$\begin{aligned} ||T|| - ||T'|| &= |p_1p_2| + |q_1q_2| - |p_1q_1| - |p_2q_2| \\ &\geq |p_1p_2| + |q_1q_2| - |p_1q_1| - (|p_1q_1| + |p_1p_2| - |q_1q_2|) \\ &\geq 2|q_1q_2| - 2|p_1q_1| \\ &\geq 2 \cdot 1 - 2 \cdot 1/2 > 0, \end{aligned}$$

as we wanted to prove.

Next, we show that T contains $O(1)$ edges fully in σ_{in} with length at least $size(\sigma)/4$. W.l.o.g., let σ be the hypercube of size 1 with center $c = (1/2, ..., 1/2)$. For $0 \leq i_1, ..., i_d \leq 8$, let $\sigma_{i_1,...,i_d}$ be the hypercube of size $1/4$ with center $(i_1/8, ..., i_2/8)$. Then, every edge of T fully in σ_{in} of length at least $1/4$ crosses at least one of these hypercubes; see Figure 9 for an example. On the other hand, by using the first part of this proof we conclude that every one of these smaller hypercubes is crossed $O(1)$ times. Since the number of smaller hypercubes is $9^d = O(1)$, we conclude that there are $O(1)$ edges of T of length at least $size(\sigma)/4$ fully in σ_{in}. This concludes the proof of the second part of the Packing Property, and hence, the proof of the Packing Property for edges of a simple tour.

D Running time of the Rectilinear One-of-a-Cube TSP algorithm

There are three differences between the algorithms that impact the running time. First, as mentioned, there are n^{2d} initial calls made to the algorithm, one for every pair of points. However, since we will prove that the running time is $2^{O(\lambda^{1/d} n^{1-1/d} \log n)}$, this factor is irrelevant. Second is the fact that there are more candidate sets. Specifically, $2^{O(\lambda^{1/d} n^{1-1/d} \log n)}$ subproblems are generated. Finally, because we only guess where the separator is crossed, $O(n^{1-1/d})$ new boundary points are generated instead of selected from the already existing points. We will now compute the impact of the last two differences on the running time of the algorithm.

Define $T(n,b)$ to be the running time of the algorithm when run on an input containing n points of which b are boundary points. Let $n_{S,in}$, $n_{S,out}$, $b_{S,in}$ and $b_{S,out}$ denote the numbers of points and boundary points in the subproblems generated.

Let $I(\sigma,\mathcal{H})$ be the set of 2^m possible combinations of choices, where m is the number of hypercubes split by σ. Clearly, if one of the points of an H_i is in B, no choice needs to be made. Define $\mathcal{C}^*(\sigma,\mathcal{H}) \subseteq \mathcal{C}'(\sigma,H) \times I(\sigma,\mathcal{H})$ to be the set of combinations of candidate sets and splitting choices restricted this way.

Let N be arbitrarily large but fixed. We get:

$$T(n,b) \le \begin{cases} c_0 & \text{if } n \le N \\ \sum_{S\in\mathcal{C}^*(\sigma_{\mathcal{H}},\mathcal{H})} e^{c_3(n^{1-1/d}+b)\log n} + T(n_{S,in},b_{S,in}) + T(n_{S,out},b_{S,out}) & \text{if } b \le \gamma n^{1-1/d} \\ \sum_{S\in\mathcal{C}^*(\sigma_B,\mathcal{H})} e^{c_3(n^{1-1/d}+b)\log n} + T(n_{S,in},b_{S,in}) + T(n_{S,out},b_{S,out}) & \text{if } b > \gamma n^{1-1/d}. \end{cases}$$

We prove by induction that $T(n,b) \le e^{\lambda^{\frac{1}{d}}(d_1 n^{1-1/d}+d_2 b)\log n}$ for some constants d_1 and d_2 and for all $1 \le b \le n$. This clearly holds for $b,n \le N$, so by induction, for each S we have

$$\begin{aligned} & e^{c_3(n^{1-1/d}+b)\log n} + T(n_{S,in},b_{S,in}) + T(n_{S,out},b_{S,out}) \\ \le\ & e^{c_3(n^{1-1/d}+b)\log n} + e^{\lambda^{\frac{1}{d}}(d_1 n_{S,in}^{1-1/d}+d_2 b_{S,in})\log n} + e^{\lambda^{\frac{1}{d}}(d_1 n_{S,out}^{1-1/d}+d_2 b_{S,out})\log n}. \end{aligned}$$

Let c_2, c_4 and c_5 be the constants from part (2), (3) and (4) of Theorem 12. For any $S \in \mathcal{C}^*(\sigma_B,\mathcal{H})$, we have $b_{S,in} \le \delta b + c_2 n^{1-1/d}$ and $b_{S,out} \le \delta b + c_2 n^{1-1/d}$; similarly, for any $S \in \mathcal{C}^*(\sigma_{\mathcal{H}},\mathcal{H})$, we have $n_{S,in} \le \delta n + c_2 n^{1-1/d}$ and $n_{S,out} \le \delta n + c_2 n^{1-1/d}$. In the remaining cases, for $b_{S,\cdot}$ we can use the trivial bound $b_{S,\cdot} \le b + c_2 n^{1-1/d}$. For $n_{S,\cdot}$, we can use $n_{S,\cdot} \le n$; despite the possibility of new points being created, there will never be more points created then there are points on either side of σ. Since $|\mathcal{C}^*(\sigma,\mathcal{H})| \le e^{c_4\lambda^{\frac{1}{d}} n^{1-1/d}\log n}$, we get the following:

$$T(n,b) \le \begin{cases} c_0 & \text{if } n \le N \\ \sum_{S\in\mathcal{C}^*(\sigma_{\mathcal{H}},\mathcal{H})} e^{c_3(n^{1-1/d}+b)\log n} + T(n_{S,in},b_{S,in}) + T(n_{S,out},b_{S,out}) & \text{if } b \le \gamma n^{1-1/d} \\ \sum_{S\in\mathcal{C}^*(\sigma_B,\mathcal{H})} e^{c_3(n^{1-1/d}+b)\log n} + T(n_{S,in},b_{S,in}) + T(n_{S,out},b_{S,out}) & \text{if } b > \gamma n^{1-1/d}. \end{cases}$$

$$\le \begin{cases} c_0 & \text{if } n \le N \\ e^{c_4\lambda^{\frac{1}{d}} n^{1-1/d}\log n}\left(e^{c_3(n^{1-1/d}+b)\log n} + 2e^{\lambda^{\frac{1}{d}}(d_1(\delta n+c_2 n^{1-1/d})^{1-1/d}+d_2(b+c_2 n^{1-1/d}))\log n}\right) & \text{if } b \le \gamma n^{1-1/d} \\ e^{c_4\lambda^{\frac{1}{d}} n^{1-1/d}\log n}\left(e^{c_3(n^{1-1/d}+b)\log n} + 2e^{\lambda^{\frac{1}{d}}(d_1 n^{1-1/d}+d_2(\delta b+c_2 n^{1-1/d}))\log n}\right) & \text{if } b > \gamma n^{1-1/d}. \end{cases}$$

For simplicity, let $\lambda' := \lambda^{\frac{1}{d}}$, let $n' := n^{1-1/d}$, and Let $c := \max\{c_0,c_2,c_3,c_4\}$. We get

$$T(n,b) \le \begin{cases} c & \text{if } n \le N \\ e^{c\lambda' n'\log n + c(n'+b)\log n + 1 + \lambda'(d_1(\delta n+cn')^{1-1/d}+d_2(b+cn'))\log n} & \text{if } b \le \gamma n' \\ e^{c\lambda' n'\log n + c(n'+b)\log n + 1 + \lambda'(d_1 n'+d_2(\delta b+cn'))\log n} & \text{if } b > \gamma n'. \end{cases}$$

Now, if n is large enough (dependent only on d), then $\delta n + cn' \le \zeta n$, where $\zeta = \frac{1+\delta}{2}$. Since we know that $n > N$ and N is arbitrarily large, we get

$$T(n,b) \le \begin{cases} c & \text{if } n \le N \\ e^{c\lambda' n'\log n + c(n'+b)\log n + 1 + \lambda'(d_1\zeta^{1-1/d}n'+d_2(b+cn'))\log n} & \text{if } b \le \gamma n' \\ e^{c\lambda' n'\log n + c(n'+b)\log n + 1 + \lambda'(d_1 n'+d_2(\delta b+cn'))\log n} & \text{if } b > \gamma n'. \end{cases}$$

$$T(n,b) \leq \begin{cases} c & \text{if } n \leq N \\ e^{(c\lambda'+c+\lambda' d_1\zeta^{1-1/d}+\lambda' d_2 c)n' \log n + (cb+\lambda' d_2 b)\log n + 1} & \text{if } b \leq \gamma n' \\ e^{(c\lambda'+c+\lambda' d_1+\lambda' d_2 c)n' \log n + (cb+\lambda' d_2 \delta b)\log n + 1} & \text{if } b > \gamma n'. \end{cases}$$

$$T(n,b) \leq \begin{cases} c & \text{if } n \leq N \\ e^{\lambda' \log n((\zeta^{1-1/d}d_1+3cd_2)n'+(2c+d_2)b)} & \text{if } b \leq \gamma n' \\ e^{\lambda' \log n((d_1+3cd_2)n'+(2c+\delta d_2)b)} & \text{if } b > \gamma n'. \end{cases}$$

Now, for the case $b > \gamma n'$, we have

$$e^{\lambda' \log n((d_1+3cd_2)n'+(2c+\delta d_2)b)} \leq e^{\lambda'\left(d_1 n' + d_2 b\right)\log n}$$

if and only if

$$(d_1 + 3cd_2)n' + (2c + \delta d_2)b \leq d_1 n' + d_2 b.$$

We choose $\gamma = \frac{4c}{1-\delta}$ and $d_2 = \frac{8c}{1-\delta}$, satisfying the equation.

For the case $b \leq \gamma n^{1-1/d}$, we have

$$e^{\lambda' \log n((\zeta^{1-1/d}d_1+3cd_2)n'+(2c+d_2)b)} \leq e^{\lambda'(d_1 n'+d_2 b)\log n}$$

if and only if

$$(\zeta^{1-1/d}d_1 + 3cd_2)n' + (2c + d_2)b \leq d_1 n' + d_2 b.$$

We choose $d_1 = \frac{32c^2}{(1-\zeta^{1-1/d})(1-\delta)}$, satisfying the equation.

Finally, we note that γ, d_1 and d_2 are indeed (nonnegative) constants, as they only depend on c, δ and ζ, which in turn only depend on d.