

Approximation Schemes for Orienteering and Deadline TSP in Doubling Metrics

Kinter Ren ✉

Department of Computer Science, University of Alberta, Edmonton, Canada

Mohammad R. Salavatipour ✉ 

Department of Computer Science, University of Alberta, Edmonton, Canada

Abstract

In this paper we look at various extensions of the classic Traveling Salesman Problem (TSP) on graphs with bounded doubling dimension and bounded treewidth and present approximation schemes for them. Suppose we are given a weighted graph $G = (V, E)$ with a start node $s \in V$, distances on the edges $d : E \rightarrow \mathbb{Q}^+$ and integer k . In k -stroll problem the goal is to find a path from s of minimum length that visits at least k vertices. In k -path we are given an additional end node $t \in V$ and the path is supposed to go from s to t . The dual problem to k -stroll is the rooted orienteering in which instead of k we are given a budget B and the goal is to find a walk of length at most B starting at s that visits as many vertices as possible. In the point-to-point orienteering (P2P orienteering) we are given start and end nodes s, t and the walk is supposed to start at s and end at t . In the deadline TSP (which generalizes P2P orienteering) we are given a deadline $D(v)$ for each $v \in V$ and the goal is to find a walk starting at s that visits as many vertices as possible before their deadline (where the visit time of a node is the distance travelled from s to that node). The best approximation for rooted orienteering (or P2P orienteering) is $(2 + \varepsilon)$ -approximation [13] and $O(\log n)$ -approximation for deadline TSP [4]. For Euclidean metrics of fixed dimension, Chen and Har-Peled present [16] a PTAS for rooted orienteering. There is no known approximation scheme for deadline TSP for any metric (not even trees). Our main result is the first approximation scheme for deadline TSP on metrics with bounded doubling dimension (which includes Euclidean metrics). To do so we first we present a quasi-polynomial time approximation scheme for k -path and P2P orienteering on such metrics. More specifically, if G is a metric with doubling dimension κ and aspect ratio Δ , we present a $(1 + \varepsilon)$ -approximation that runs in time $n^{O((\log \Delta / \varepsilon)^{2\kappa+1})}$. Building upon these, we obtain an approximation scheme for deadline TSP when the distances and deadlines are integer which runs in time $n^{O((\log \Delta / \varepsilon)^{2\kappa+2})}$. The same approach also implies a bicriteria $(1 + \varepsilon, 1 + \varepsilon)$ -approximation for deadline TSP for when distances and deadlines are in $\mathbb{Q}^+$. For graphs with bounded treewidth ω we show how to solve k -path and P2P orienteering exactly in polynomial time and a $(1 + \varepsilon)$ -approximation for deadline TSP in time $n^{O((\omega \log \Delta / \varepsilon)^2)}$.

2012 ACM Subject Classification Theory of computation $\rightarrow$ Routing and network design problems

Keywords and phrases Deadline Traveling Salesman Problem, Orienteering, Doubling Metrics, Approximation algorithm

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2025.1

Category APPROX

Related Version Full Version: <https://arxiv.org/abs/2405.00818> [31]

Funding Kinter Ren: Supported by NSERC DG of the 2nd author.

Mohammad R. Salavatipour: NSERC

1 Introduction

We study a fundamental variant of the *Traveling Salesman Problem* (**TSP**) in which the “salesman” wants to visit as many customers as possible before their required service *deadlines*. Suppose we are given a weighted graph $G = (V, E)$ with a start node $s \in V$, deadlines



© Kinter Ren and Mohammad R. Salavatipour;
licensed under Creative Commons License CC-BY 4.0

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2025).

Editors: Alina Ene and Eshan Chattopadhyay; Article No. 1; pp. 1:1–1:22



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

$D : V \rightarrow \mathbb{Q}^+$, and a distance/length function $d : E \rightarrow \mathbb{Q}^+$. In the *deadline TSP* problem, our objective is to find a path (or a walk) starting at s that visits as many nodes as possible by their deadlines. We say that the path visits node v by its deadline if the length of the path, starting at s until it visits v (for the first time), is at most $D(v)$. A node might be visited multiple times, but it is counted only once. Alternatively, we can assume G is a complete weighted graph where the edge weights satisfy triangle inequality, i.e. the (V, d) is a metric space.

A special case of the deadline TSP is the *rooted Orienteering* problem, where all the nodes have a universal deadline B (think of it as a budget for the length of the path travelled) and the goal is to find a path, starting at s , of length at most B that visits as many vertices as possible. Researchers have also studied another version of the Orienteering, known as *point-to-point Orienteering* problem, denoted by **P2P orienteering**, where we are given a *start node* $s \in V$, an *end node* $t \in V$, and a *budget* B . Here, our objective is to find an s - t path of length at most B that visits as many nodes (other than s, t) as possible. This too can be viewed as a special case of deadline TSP: the deadlines for each node v are set as $B - d(v, t)$, and the deadline TSP path is completed by connecting it directly to the end node t . Notice that P2P orienteering can be seen as the “dual” of the k -path problem [12]. It is almost the same as the classical **k -stroll** problem [9] (also known as the (path) k -TSP problem [16]) except in k -stroll we are given a start node $v \in V$ and integer k and the objective is to find a minimum-length path from s visiting at least k nodes, while in k -path problem we are given an additional end node $t \in V$ and the path is supposed to go from s to t . It is important to note that P2P orienteering and k -path are equivalent in terms of exact algorithms. However, an approximation for one problem does not imply an approximation for the other.

Blum et al. [9] developed the first $O(1)$ -approximation algorithm for the rooted orienteering problem in a general metric space. Their main approach involves a clever reduction to k -TSP, via an intermediate problem. They introduced the concept of *excess* (also referred to as *regret*) for a path and formulated the related *minimum-excess* problem, demonstrating that an approximation for the latter extends to an approximation for the orienteering problem. Given a path P that includes two vertices u, v , let $d_P(u, v)$ denote the length of the sub-path from u to v . The excess of an s, t -path P , denoted by $\mathcal{E}_P$, is the difference between the length of the path and the distance between the endpoints of the path, $\mathcal{E}_P = d_P(s, t) - d(s, t)$ and the minimum-excess problem seeks to find an s - t path of minimum-excess that visits at least k nodes. Moreover, they [9] demonstrated that the min-excess problem can be approximated using algorithms for k -stroll, implying that an approximation algorithm for k -stroll leads to an approximation algorithm (with a constant factor loss in the ratio) for the orienteering problem, via the intermediate min-excess problem. Applying a similar approach and concentrating on enhancing the approximation algorithm for the k -stroll, the factor has been refined to the currently best $(2 + \epsilon)$ -approximation, which also extends to P2P orienteering [13] in a general metric space.

Chen and Har-Peled [16] developed the first Polynomial-Time Approximation Scheme (PTAS) for the rooted orienteering problem in a fixed-dimensional Euclidean space. The prior reduction by [9] increases the approximation ratio by a constant factor and can not provide a PTAS. To achieve a PTAS for the rooted orienteering problem, Chen and Har-Peled [16] introduced the concept of μ -excess for a path (extending the notion of the excess of a path) and defined (ϵ, μ) -approximation for k -TSP. Roughly speaking, the μ -excess of a path is the difference between its length and the length of the best approximation of the path using straight lines between only μ nodes of the path (as opposed to considering the length of the

line connecting the two endpoints of the path, as in the min-excess path). This notion is formally defined soon. An (ϵ, μ) -approximation for rooted k -TSP is any path that starts from the root and visits k points and whose length is at most $\|P^*\| + \epsilon \cdot \mathcal{E}_{P^*, \mu}$, where $\|P^*\|$ denotes the length of the optimal rooted k -TSP path and $\mathcal{E}_{P^*, \mu}$ is the μ -excess for the optimal path. Note that the upper bound provided by (ϵ, μ) -approximation for the k -TSP may be significantly tighter than that by $(1 + \epsilon)$ -approximation (namely, $(1 + \epsilon) \cdot \|P^*\|$), especially when μ is sufficiently large, as the μ -excess of a path can be much smaller than the length of the path. In fact, Chen and Har-Peled [16] showed that the algorithm of Mitchell for classic TSP on Euclidean plane [29] provides a $(\epsilon, \frac{1}{\epsilon})$ -approximation for the rooted k -TSP problem on the that metric and this leads to a PTAS for the rooted orienteering problem.

Deadline TSP seems substantially more difficult to approximate. For deadline TSP Bansal et al. [4] presented an $O(\log n)$ -approximation and extended that to an $O(\log^2 n)$ -approximation for the Time-Window TSP problem (where each node v , other than a deadline $D(v)$, has also “release” time $R(v)$, and it is “counted” if the time it is visited along the output path falls in the interval $[R(v), D(v)]$). They also provided a bicriteria approximation: one that produces a $\log(1/\epsilon)$ -approximate solution that violates the deadlines by $(1 + \epsilon)$. The $O(\log n)$ -approximation remains the best known approximation for deadline TSP on general metrics (in polynomial time). More recently, Friggstad and Swamy [20] presented an $O(1)$ -approximation for deadline TSP which runs in time $O(n^{\log n \Delta})$ where Δ is the diameter of the graph; for this they assume that distances (and so deadlines) are all integers. Approximations for both orienteering and deadline TSP for other metrics (such as directed graphs) have been obtained as well (discussed in Section 1.2). For instance, for deadline TSP on trees Farbstein and Levin [18], present a bicriteria $(1 + \epsilon, 1 + \epsilon)$ -approximation (i.e. $(1 + \epsilon)$ -approximation and violating the deadlines by $(1 + \epsilon)$).

To the best of our knowledge, there is no true approximation scheme for deadline TSP on any metrics (even on Trees). Our goal in this work has been to obtain an approximation scheme for deadline TSP on bounded doubling dimensions (which includes Euclidean metrics as a special case). We should note that the best known approximation for deadline TSP on Euclidean metrics is the same as general metrics: $O(\log n)$ -approximation in polynomial time and $O(1)$ -approximation in time $O(n^{\log n \Delta})$.

1.1 Our Results and Techniques

By scaling we assume all distances are between 1 and Δ . Our main result of the paper is a Quasi-Polynomial Time Approximation Scheme QPTAS (under mild conditions) for deadline TSP on doubling metrics. (see the full version of our paper [31] for all missing proofs)

► **Theorem 1.** *Suppose $G = (V, E)$ is a graph with doubling dimension κ , start nodes $s \in V$ and deadline $D(v)$ for $v \in V$ as an instance of deadline TSP where distances (and deadlines) are integer values. There is a (randomized) $(1 + \epsilon)$ -approximation for deadline TSP that runs in time $n^{O((\log \Delta / \epsilon)^{2\kappa+2})}$.*

This is the first approximation scheme for any non-trivial metrics for deadline TSP. We note that not violating any of the deadlines is the critical. Along the way to obtain this goal we present a number of other results that are used as building blocks in the proof of this main theorem. As a starting point we show that orienteering can be solved exactly in polynomial time on graphs with bounded treewidth and how this can be turned into a QPTAS for deadline TSP on such metrics when the distances are integer and Δ is quasi-polynomial in n .

► **Theorem 2.** *Given a graph with treewidth ω , there is an exact algorithm with running time $O(\text{Poly}(n) \cdot \omega^{\omega^2})$ for solving the k -path or P2P orienteering problem. Furthermore, there is an approximation scheme for deadline TSP on such graphs with integer distances with running time $n^{O((\omega \log \Delta/\varepsilon)^2)}$.*

Proof of this Theorem (see the full version [31]) is simpler and has some of the main ideas used in the subsequent theorems; so is a good starting point. We should note that recently (independently and after the first version of our paper [31] appeared), [10] considered orienteering on graph with bounded treewidth and obtained a $(1 + \varepsilon)$ -approximation by designing a different dynamic programming.

Next we focus on metrics with bounded doubling dimension.

► **Theorem 3.** *Given a graph $G = (V, E)$ with doubling dimension κ , start and end nodes $s, t \in V$ and integer k . There is a (randomized) $(1 + \varepsilon)$ -approximation for k -path that runs in $n^{O((\log \Delta/\varepsilon)^{2\kappa+1})}$ time.*

We actually show a stronger bound on the length of the k -path solution produced by this theorem which bounds the *excess* or *regret* similar to [16] for Euclidean metrics. Although getting a QPTAS for k -path on doubling metrics isn't difficult using the known techniques, one needs the stronger bound obtained here (with respect to excess) as it is crucially used to prove the following:

► **Theorem 4.** *Given a graph $G = (V, E)$ with doubling dimension κ , start and end nodes $s, t \in V$ and length budget B . There is a (randomized) $(1 + \varepsilon)$ -approximation for P2P orienteering that runs in time $n^{O((\log \Delta/\varepsilon)^{2\kappa+1})}$.*

Building upon these theorems we then have the tools needed to prove Theorem 1. We note that the algorithm of this theorem can be adapted to obtain a (randomized) bicriteria $(1 + \varepsilon, 1 + \varepsilon)$ -approximation for deadline TSP (i.e. cost at most $1 + \varepsilon$ the optimum while violating the deadlines by at most $1 + \varepsilon$) that works even when distances are in $\mathbb{Q}^+$ (instead of integer) with the same running time.

Limitations of known techniques. The reason that our running time in Theorems 1, 3, and 4, includes $\log \Delta$ in the exponent is that our algorithm uses the (randomized) hierarchical decomposition of such metrics [32], which produces a decomposition of height $O(\log \Delta)$. A natural question is: why can't we get rid of the dependence on Δ (like [32])? One should note that for most vehicle routing problems on Euclidean or doubling metrics, one can assume that $\Delta = \text{Poly}(n)$ by a standard scaling of distances at a loss of $(1 + \varepsilon)$ in approximation factor (for e.g. this has been the first step in [2, 32, 7] for designing approximation schemes for TSP on Euclidean and doubling metrics). This however does not work for budgeted versions (such as orienteering or deadline TSP) as a feasible solution for the scaled version might violate the budget(s) by a $(1 + \varepsilon)$ factor in the original version. In other words, once we scale the distances (by $1 + \varepsilon$ factor) to reduce to an instance with $\text{Poly}(n)$ distances, a solution in the new instance will not yield a feasible solution in the original as the hard budget constraint (on the total distance) may be violated. Another point worth mentioning is that, although one can get an approximation scheme (with time $n^{O(\text{Polylog}(\Delta+n))}$) for k -path on doubling metrics by embedding them into $\text{Polylog}(\Delta)$ -treewidth metrics [33] and using Theorem 2, this will not lead to an approximation scheme for orienteering or deadline TSP as one needs a stronger guarantee, like we obtain in the proof of Theorem 3. Again, this is for the same reason that obtaining a true (and not bicriteria) PTAS for Orienteering on Euclidean metrics [16] required stronger k -path algorithms. In general, embedding results cannot be used for budgeted versions of vehicle routing (such as orienteering or deadline TSP).

A natural question is: can't one extend the known techniques developed for designing PTAS's for TSP on various metrics (such as planar metrics [28] or doubling metrics [7, 3]) to the setting of orienteering or deadline TSP? It appears the answer is not easy for orienteering and deadline TSP. In general, budgeted versions seem more difficult on any metrics. Extending the ideas of [7] to orienteering or deadline TSP encounters some problems. The very first one (as mentioned above) is the assumption that distances are bounded by $\text{Poly}(n)$ by scaling and losing a $(1 + \varepsilon)$ -factor. This can be done for k -path but not in the budgeted settings such as orienteering or deadline TSP. Even if one bypasses this issue, the algorithm of [7] considers tours that are sparse or dense (we have a similar breakdown in our proof). However, for both situations [7] considers (approximate) portal respecting tours. Such tours might have large *excess* (or regret); so although they will have an approximately optimal overall length, they don't have the stronger bound (with respect to the excess) needed to obtain a P2P orienteering solution (see [13, 16]). Furthermore, our algorithm for deadline TSP has a step (similar to [20]) that needs to guess a set of vertices whose "excess" (which is defined to be the difference of its visit time in optimum vs. the shortest path distance to the root) is growing geometrically. This requires guessing $\Theta(\log \Delta)$ many vertices. So overall, there are several obstacles to overcome to turn these algorithms into a PTAS. Nonetheless we hope that ideas developed here (in the proof of Theorem 1) can be used and extended to obtain a PTAS for deadline TSP in doubling or maybe Euclidean metrics.

1.2 Related Works

The seminal works of Arora [2] and Mitchell [29] presented the first PTAS for Euclidean TSP and some of its variants such as k -TSP. However, extending this to orienteering proved to be challenging. For orienteering on Euclidean plane Arkin et al. [1] presented a $(2 + \varepsilon)$ -approximation. The first PTAS for rooted orienteering on Euclidean metrics was given by Chen and Har-Peled [16]. More recently Gottlieb et al. [23] presented a more efficient PTAS for rooted orienteering on Euclidean metrics.

For orienteering on general metrics there are several $O(1)$ -approximations [9, 13, 19], with the current best being $(2 + \varepsilon)$ -approximation by [13]. For orienteering on directed graphs Chekuri and Pál [15] present a quasi-polynomial time for various problems including deadline TSP, such as an $O(\log \text{opt})$ -approximation for time window in quasi-polynomial time. Nagarajan and Ravi [30] also present poly-logarithmic approximations for directed variants of k -TSP and Orienteering, some of these results were improved in [8]. Chekuri et al. [13] present an $O(\log^2 \text{opt})$ -approximation for the time-window version and an $O(\log \text{opt})$ -approximation for P2P orienteering on directed graphs. There is also a line of research on stochastic orienteering (e.g. see [25, 5, 27]).

As said earlier, the best known result for deadline TSP on general metrics is by [4] which is an $O(\log n)$ -approximation. They also provide a bicriteria $(O(\log \frac{1}{\varepsilon}), 1 + \varepsilon)$. Assuming distances (and deadlines) are integers, this implies an $O(\log D_{\max})$ -approximation where $D_{\max}$ is the maximum deadline. Different extensions of deadline TSP have been studied also. For instance, Bansal et al. [4] present an $O(\log^2 n)$ -approximation for the time-window version and an $O(\log D_{\max})$ -approximation when the time window values are integer. Chekuri et al. [13] prove that any α -approximation for P2P orienteering implies an $O(\alpha \max \{\log \text{opt}, \log \frac{L_{\max}}{L_{\min}}\})$ -approximation for the time-window version where opt is the optimal value and $L_{\max}$ and $L_{\min}$ are the sizes of the largest and smallest windows. For the variant of time-window TSP where there are multiple time-windows given for each node, Garg et al. [22] proved an $\Omega(\frac{\log \log n}{\log \log \log n})$ -hardness of approximation even on trees. There are other variants of orienteering and deadline TSP that have been studied (see [13, 15, 20, 21]).

Some variants of vehicle routing problems have been studied on doubling metric as well. Talwar [32] presented a QPTAS for TSP on such metrics. Bartal et al. [7] building upon the work of Talwar presented a PTAS for TSP on doubling metrics. Several variants of TSP, such as Capacitated Vehicle Routing Problem (CVRP) or TSP with neighborhoods have approximation schemes on doubling metrics [11, 26].

2 Preliminaries

We consider a metric space (V, d) as a (complete) weighted graph $G = (V, E)$. By scaling we assume the minimum pair-wise distance is 1 and diameter is Δ . For $v \in V$ and $r \geq 0$, we let $B(v, r) = \{u \in V \mid d(u, v) \leq r\}$ denote the ball of radius r around v . The *doubling dimension* [24] of a metric space (V, d) is the smallest value κ such that for all $v \in V$, for all $\rho > 0$, every ball $B(v, 2\rho)$ can be covered by the union of at most 2^κ balls of the form $B(z, \rho)$, where $z \in V$. A metric is called *doubling* when its doubling dimension is a constant. For a subset $U \subseteq V$, the *diameter* of U , denoted by Δ_U , is defined as $\max_{u, v \in U} d(u, v)$.

We now formally define the problems. We consider a metric space (V, d) which is induced by an edge-weighted graph $G = (V, E)$. Let $P = \langle v_1, v_2, \dots, v_k \rangle$ be a path in G , starting at $s = v_1$ and ending at $t = v_k$. The *length of the path* is denoted by $\|P\| = \sum_{i=1}^{k-1} d(v_i, v_{i+1})$. We use $|P|$ to denote the number of vertices on the path. If P is a walk then $|P|$ is the number of distinct nodes in P . Let a μ -jump of P be a path $\langle v_1 = v_{i_1}, v_{i_2}, \dots, v_{i_\mu} = v_k \rangle$ of size $\mu \leq k$ obtained from P by bypassing some of its intermediate points. The optimum μ -jump of P , denoted by $J_\mu^*(P)$, is the μ -jump of P with the maximum length. We define the μ -excess of P (see [16]), denoted by $\mathcal{E}_{P, \mu}$, to be the difference between the length of P and the length of $J_\mu^*(P)$, i.e. $\mathcal{E}_{P, \mu} = \|P\| - \|J_\mu^*(P)\|$. Observe that $\mathcal{E}_{P, \mu}$ may be significantly smaller than $\|P\|$, especially when μ is sufficiently large.

In k -path, we are given a *start node* s , an *end node* t , and a *target integer* k . The goal is to find a minimum length path from s to t that visits at least k nodes. An (ε, μ) -approximation for k -path is an algorithm that finds a path P with $|P| = k$ where $\|P\| \leq \|P^*\| + \varepsilon \cdot \mathcal{E}_{P^*, \mu}$ where P^* is the optimum solution. We study the following extension of k -path to *multi-path* in Section 3 where we design our algorithms for P2P orienteering.

► **Definition 5** (multi k -path). *Given a graph $G = (V, E)$, m pairs of nodes (s_i, t_i) , $1 \leq i \leq m$, and a target integer k , the goal of **multi k -path** is to find m paths P_i from s_i to t_i in G such that:*

- *the total number of distinct nodes visited by all paths $P_1, P_2, \dots, P_m$ is at least k . (a node might be visited by multiple paths, but it is counted only once.)*
- *the total length of all m paths $\sum_{i=1}^m \|P_i\|$ is minimized.*

Notice that multi k -path becomes k -path when $m = 1$.

In P2P orienteering, we are given a *start node* s , an *end node* t , and a *budget* B . The objective is to find an s - t path P with $\|P\| \leq B$ that maximizes the count of distinct nodes visited.

For any path P in G and nodes u, v in P , let P_{uv} denote the *subpath* from u to v in P . In deadline TSP, we are given a *start node* $s \in V$ and *deadlines* $D(v)$ for each node in V . A feasible solution is a path starting at s . We say that such a path visits node $v \in P$ by its deadline if $\|P_{sv}\| \leq D(v)$. The objective is to find a path P starting at s to maximize the count of distinct nodes visited within their deadlines.

For graphs of bounded doubling dimension, like earlier works on problems on such metrics, we rely on a hierarchical decomposition of V . Suppose (V, d) is a doubling metric with diameter Δ . We employ the hierarchical decomposition of metric spaces by means of

probabilistic partitioning. The decomposition is essentially the one introduced by Bartal [6], and subsequently used by others. In particular, Talwar [32] used this to design the first approximation scheme (QPTAS) for TSP and other problems in doubling metrics. A *cluster* C in the metric (V, d) is a subset of nodes in V . A *decomposition* of the metric (V, d) is a partitioning of V into clusters. A hierarchical decomposition of V is a sequence of partitions of V , where each partition is a refinement of the previous one. Normally this is represented by a tree T (called the *split-tree*), where each node of T corresponds to a cluster. We use C to both refer to a node in T as well as the cluster (set of nodes in V) it corresponds to. The root node of T corresponds to the single set $\{V\}$ and the leaf nodes correspond to singleton sets $\{\{v\}\}_{v \in V}$. The children of each node $C \in T$ correspond to a partition of C where each part has diameter about half of that of C . The union of all subsets corresponding to the vertices at each level of T constitutes a partition of V .

► **Theorem 6** ([32]). *There is a hierarchical decomposition of V , i.e. a sequence of partitions $\Pi_0, \Pi_1, \dots, \Pi_h$, where Π_{i-1} is a refinement of Π_i , $\Pi_h = \{V\}$, and $\Pi_0 = \{\{v\}\}_{v \in V}$, and satisfies the following:*

1. Π_0 corresponds to the leaves and Π_h corresponds to the root of the split-tree T , and height of T is $h = \delta + 2$, where $\delta = \log \Delta$ and Δ is the diameter of the metric.
2. For each $C \in T$ at level i , cluster $C \in \Pi_i$, has diameter at most 2^{i+1} .
3. The branching factor b of T is at most $2^{O(\kappa)}$.
4. For any $u, v \in V$, the probability that they are in different sets corresponding to nodes in level i of T is at most $O(\kappa) \cdot \frac{d(u,v)}{2^i}$.

3 An Approximation Scheme for k -path and Orienteering in Doubling Metrics

Before we can give the proof of Theorem 1 we prove Theorems 3 and 4 as our algorithm for Theorem 1 builds upon the idea in these proofs. Suppose that $G = (V, E)$ (a graph with bounded doubling dimension κ), $s, t \in V$, and budget $B \in \mathbb{Q}^+$ are given as an instance of P2P orienteering. Let $\delta = \log \Delta_G$. We assume Δ_G is quasi-polynomial in which case δ is polylogarithmic.

Overview. Before we present our approximation scheme for P2P orienteering on G let us review the algorithm of Chen and Har-Peled [16] for P2P orienteering on Euclidean plane. They showed that the algorithm of Mitchell [29] for Euclidean TSP in fact implies a (ϵ, u) -approximation for k -path, i.e. an algorithm that finds a path P with $|P| = k$ where $\|P\| \leq \|P^*\| + \epsilon \cdot \mathcal{E}_{P^*, \mu}$, where P^* is the optimum k -path solution. Using this approximation algorithm to solve the orienteering problem (with the assumption that k is the guessed value of the optimum solution for orienteering), they show that one can break the path obtained by the algorithm into $\mu = O(\frac{1}{\epsilon})$ segments, each containing $\epsilon \cdot k$ vertices; this will be a μ -jump; once we remove one segment that has the longest length the total length of the path is dropped by at least $\epsilon \cdot \mathcal{E}_{P^*, \mu}$, and therefore the length of the resulting path is at most the length of the optimum path $\|P^*\|$ and hence below the given budget and the total number of nodes is at least $(1 - \epsilon)k$. This yields the approximation scheme for orienteering (via k -path). The algorithm for (ϵ, u) -approximation of k -path is essentially the algorithm of Mitchell [29] for Euclidean TSP which breaks the problem into (polynomially many) subinstances, each defined by a window w (a minimum bounding box). Mitchell defines a vertical (or horizontal) line ℓ that cuts a window w a *cut* and for a parameter $m = O(1/\epsilon)$ if the number of edges of the optimum path that cross ℓ is no more than m then this cut is *sparse*; else ℓ is dense.

If ℓ is sparse one can *guess* the edges of the optimum in time $O(n^m)$ and break the problem into independent instances. If there is no sparse cut then the value of the optimum in this window is large. In this case using either Mitchell's [29] scheme (using bridges) or Arora's scheme [2] for TSP (making paths portal respecting paths) one can modify the solution for the problem restricted to w to a near optimum one of length at most $(1 + \frac{1}{m})\|P^*(w)\|$, where $P^*(w)$ is the restriction of the optimum path to window w . We should point out that this idea of partitioning sub-instances into sparse and dense regions has been used in other works (e.g. in [7] to obtain a PTAS for TSP on doubling metrics).

Our QPTAS for orienteering on doubling metrics builds upon this idea of sparse and dense sub-instances, but the difficulty is we do not have a polynomial size set of windows or cuts as in the Euclidean case. We first try to find a good approximation for k -path on G with a similar stronger upper bound on the quality of the solution: a solution with a stronger upper bound of the form $\|P^*\| + \epsilon \cdot \mathcal{E}_{P^*, \mu}$. We show the existence of a near-optimum solution for k -path which has some structural properties. Suppose that T is a hierarchical decomposition of G as in Theorem 6. Consider an arbitrary node $C \in T$ of the hierarchical decomposition at level i with children $C_1, \dots, C_\sigma$. Roughly speaking we would like this near optimum solution to have very few edges crossing between different clusters C_i 's when we focus on C or these edges be portal-respecting if we impose some portals for each C_i as in the algorithm for TSP on doubling metrics [32]. The existence of such a structured near optimum allows us to find it using a DP on T .

For simplicity of presentation, suppose P^* (for k -path) is known (let's call this "Assumption 1"). As said above, consider $C \in T$ with children $C_1, \dots, C_\sigma$. We also consider the restriction of P^* to C , denoted by $P_C^* = P^* \cap C$ and let Δ_C denote the diameter of C . We consider two cases based on whether $\|P_C^*\| \leq \frac{\log n}{\epsilon} \Delta_C$ holds or not. If the inequality holds we call P_C^* *sparse* with respect to C ; else we call it *dense*.

Sparse case. If $\|P_C^*\| \leq \frac{\log n}{\epsilon} \Delta_C$, we show in this case the *expected* number of edges of P_C^* crossing between different subgraphs $C_1, \dots, C_\sigma$ is $O(\frac{\kappa \log n}{\epsilon})$; this uses property 4 of Theorem 6. For simplicity assume this actually holds with high probability instead of "in expectation" (call this "Assumption 2")

Dense case. If $\|P_C^*\| > \frac{\log n}{\epsilon} \Delta_C$ we modify P_C^* to a near optimum one by making P_C^* to be portal respecting when it crosses between $C_1, \dots, C_\sigma$ (as was done for TSP for e.g. in [32]). We consider each of C_i and generate a set of $O(\log^\kappa n)$ portals for it. We make P_C^* portal respecting, that is, we modify it such that it crosses between different subgraphs C_i only through portals to reduce the number of times it crosses between different subgraphs to be poly-logarithmic. We show the *expected* increase of length of P_C^* in C is bounded by $O(\|P_C^*\| \cdot \epsilon / \log n)$, which holds by Theorem 6. Again, assume that this actually holds with high probability instead of "in expectation" every time we do this (call this "Assumption 3")

We modify P^* to a structured near optimum solution P' as follows. Starting from the root of T , and let initially $P' = P^*$. We modify P' as we go down the tree T : for each node $C \in T$, if P_C' (restriction of P' to C) is sparse w.r.t. C we make no modification going down to children of C . However, if P_C' is dense w.r.t. C , we make P_C' portal respecting when crossing between the subgraphs corresponding to children of C . This increases the length of P_C' by at most $O(\|P_C'\| \cdot \epsilon / \log n)$. Given that height of T is $O(\log n)$, the total increase in length of P' over all steps can be bounded to be $O(\epsilon \|P^*\|)$.

We have made a number of assumptions above that we should remove. We don't know P^* (Assumption 1) and whether it is sparse or dense in each step we are going down the tree T . To remove this assumption we *guess* both cases of whether P^* is sparse or dense at

each step and add both into the DP. To remove Assumptions 2 and 3, we repeat the random decomposition of Theorem 6 $\Omega(\log n)$ many times at each step. In other words, imagine we do parallel (or non-deterministic) runs of the decomposition at each step we want to partition a cluster. Then instead of things holding in expectation, one can show that in at least one such decomposition Assumptions 2 and 3 actually hold with high probability. This idea of repeating random decompositions has been used in earlier works, most recently by [17] to present approximation scheme for k -MST on minor free graphs. This non-deterministic decomposition can be described as a tree itself, which we call a γ -split-tree. Recall that in the hierarchical decomposition we would partition each cluster C into $C_1, \dots, C_\sigma$ where each C_i has diameter at most $\frac{\Delta_C}{2}$ and $\sigma \in 2^{O(\kappa)}$. In a γ -split-tree we consider γ many such (independent) partitions of C in each step when we decompose C . This will help us to show that for at least one partition, the properties that are shown to hold in expectation (Assumptions 2 and 3) actually hold with high probability for at least one partition of C . We don't know which of the several parallel decompositions we run at each step will have this property; so we guess and try all of them in our DP.

In the next subsection we formalize this and present our structural theorem that proves the existence of a nearly optimum solution with certain properties. We then show in Subsection 3.2 that the structural theorem in fact shows the existence of a (ε, μ) -approximation for k -path. In Subsection 3.3 we show how we can find that using a suitable DP. Finally in Subsection 3.4 we prove that we show an approximation scheme for P2P orienteering.

3.1 Structural Theorem

Let P be any feasible solution and P^* be an optimal solution to k -path. For a subgraph (or subset of vertices) $U \subseteq G$ we use P_U to denote the restriction of P to U .

► **Definition 7.** P is called η -dense with respect to $U \subseteq G$ if $\|P_U\| > \eta \Delta_U$; otherwise P is η -sparse.

We will set $\eta = \frac{\log n}{\epsilon}$. A set $C \subseteq V$ is a ρ -cover for $U \subseteq V$ if for any $u \in U$, there exists $c \in C$ such that $d(u, c) \leq \rho$. A set S is a ρ -packing if for any two distinct points u and v in S , $d(u, v) \geq 2\rho$ holds. A set N is a ρ -net for $U \subseteq V$ if N is a $\frac{\rho}{2}$ -packing and a ρ -cover for U . We can find a ρ -net for any graph G easily by iteratively removing balls of radius ρ . The center of these balls forms a ρ -net.

Consider a set $C \subseteq G$. A *random partition* of C is a partition $\pi_C = C_1, C_2, \dots, C_\sigma$ of C that is obtained by taking a random $\frac{\Delta_C}{4}$ -net of C (i.e. the centers chosen iteratively are picked randomly); so the balls of the net have diameter $\frac{\Delta_C}{2}$ and we know that $|\pi_C| \leq 2^{O(\kappa)}$. Note that the algorithm for proving Theorem 6 essentially starts from the single cluster $\{V\}$ (as the root of the decomposition) and at each step, when we have a cluster C , it is decomposed into $2^{O(\kappa)}$ parts by taking a random partition of C . We say an edge (u, v) is crossing π_C if u and v are in different parts of π_C i.e. $u \in C_i$ and $v \in C_j$ for $i \neq j$. Such an edge is called a **bridge-edge**. The following lemma (implied by Theorem 6) shows one property of bridge-edges of π_C .

► **Lemma 8** ([32]). *For any edge (u, v) with $u, v \in C$, the probability that (u, v) crosses π_C (and so is a bridge-edge) is at most $\kappa' \frac{d(u, v)}{\Delta_C}$ for some constant $\kappa' = O(\kappa)$.*

Consider $P_C = P \cap C$. We consider the set of edges of P that are bridge-edges, denoted as E_{π_C} , i.e. $E_{\pi_C} = \{(u, v) \in P : (u, v) \text{ crosses } \pi_C\}$. Suppose P_C is η -sparse with respect to C , i.e. $\|P_C\| \leq \eta \Delta_C$. We upper bound the number of bridge-edges of π_C , i.e. the size of E_{π_C} in this case.

► **Lemma 9.** *If P is η -sparse with respect to C , then $\mathbb{E}[|E_{\pi_C}|] \leq \frac{\kappa' \log n}{\epsilon}$ for some constant $\kappa' = O(\kappa)$.*

Now suppose P is η -dense with respect to C . Again consider the (random) partition π_C of C . For each $C_i \in \pi_C$, we further consider a $\beta\Delta_{C_i}$ -net of it where $\beta = \frac{\epsilon}{4\kappa'\delta}$ (recall $\delta = \log \Delta_C$ and κ' is the constant in Lemma 8), denoted as S_i and we call them the portal set for C_i . The following packing property of doubling metrics bounds the size of $|S_i|$:

► **Proposition 10** ([24]). *Let (V, d) be a doubling metric with doubling dimension κ and diameter Δ , and let S be a ρ -packing. Then $|S| \leq \left(\frac{\Delta}{\rho}\right)^\kappa$.*

Using this proposition: $|S_i| \leq \left(\frac{8\kappa'\delta}{\epsilon}\right)^\kappa$. We are going to modify P_C such that when it crosses between different clusters C_i in π_C (i.e. uses bridge-edges) it does so via portals only. This means after the modification it will use **portal-edges**; those bridge-edges whose both end-points are portals. We say a path P' is portal respecting with respect to π_C if for any edge u, v of P' crossing between two parts $C_i, C_j \in \pi_C$, it only cross through portals, i.e. $u \in S_i$ and $v \in S_j$. The following lemma shows that one can modify P_C to be portal respecting with respect to π_C with a small increase of length.

► **Lemma 11.** *P_C can be changed to another path P' that is portal respecting with respect to π_C such that $\mathbb{E}[|P'|] \leq (1 + \frac{\epsilon}{2\delta})|P_C|$.*

Proof. We start with P_C and whenever an edge u, v of P_C is crossing two parts $C_i, C_j \in \pi_C$ that are not portals we replace that edge with a path between u, v via the closest portals in C_i and C_j . Consider any edge u, v in P_C that crosses π_C , say $u \in C_i$ and $v \in C_j$. Let u' be the nearest portal to u in S_i and v' be the nearest portal to v in S_j . Replace edge u, v in P_C by the edges (u, u') , (u', v') , (v', v) . The increased length incurred is $d(u, u') + d(v, v') + d(u', v') - d(u, v)$, which is at most $2d(u, u') + 2d(v, v')$ by triangle inequality. Note that because S_i is a $\beta\Delta_{C_i}$ -net of C_i then $d(u, u') \leq \beta\Delta_{C_i} \leq \beta\frac{\Delta_C}{2}$ and $d(v, v') \leq \beta\Delta_{C_j} \leq \beta\frac{\Delta_C}{2}$. Thus the increased length incurred is at most $2\beta\Delta_C$. Recall that for $u, v \in P_C$, the probability that (u, v) crosses π_C is at most $\delta\kappa\frac{d(u, v)}{\Delta_C}$. Thus in expectation, the increased length of P_C after making it portal respecting with respect to π_C is at most $\sum_{u, v \in P_C} \kappa' \frac{d(u, v)}{\Delta_C} 2\beta\Delta_C = \kappa'\beta|P_C| = \frac{\epsilon}{2\delta}|P_C|$. ◀

Now we formalize the idea of parallel runs of the hierarchical decomposition of Theorem 6. Note that the algorithm for proving Theorem 6 starts from the single cluster $\{V\}$ (as the root of the decomposition) and at each step uses a *random partition* to the current cluster C to decompose it into $2^{O(\kappa)}$ clusters of diameter half the size. This continues until we arrive at singleton node clusters (leaves of the split-tree). We call one random partition of a cluster C a *split* operation. The γ -split-tree hierarchical decomposition of a doubling metric is obtained by considering γ random partitions of C (instead of just one) at each step. A γ -split-tree decomposition has two types of nodes that appear in alternating layers: cluster nodes and split nodes. For each cluster C we have γ *split* nodes in the γ -split-tree (each corresponding to a random partition of C) and these split nodes are all children of C . For each split node s that is a child of C , it will have children $C_1, \dots, C_\sigma$ that are clusters obtained by decomposing C according to the random partition corresponding to s . We continue until leaf nodes are cluster nodes with constant size.

► **Definition 12.** *A γ -split tree for G is a rooted tree Γ with alternating levels of cluster nodes and split nodes. A cluster node C corresponds to a subset of V . The root of Γ , corresponds to the single cluster $\{V\}$ and each leaf node corresponds to a set of size $O(1)$ of vertices.*

Each non-leaf cluster node C has γ split nodes as its children. Each split node corresponds to a (random) partition of C into clusters of diameter $\frac{\Delta_C}{2}$. Each of those clusters becomes cluster children of the split node. So each split node has $2^{O(\kappa)}$ (cluster) children.

Suppose Γ is a γ -split-tree for G and Φ is a partial function from (some) cluster nodes of Γ to one of their split node children with the following properties:

- $\Phi(C_0) = s$ for some child split node of C_0 (where C_0 is the root of Γ).
- if every ancestor split node of C is in the image of Φ then $\Phi(C)$ is defined.

Note that this function induces a “standard” hierarchical decomposition split-tree as in Theorem 6 as follows: start at the root node $C_0 = \{V\}$ of Γ and let root of tree T be C_0 and repeat the following procedure: at each step, being at a cluster node C pick $\Phi(C)$ (a split node child of C) and consider the cluster children of $\Phi(C)$, say $C_1, \dots, C_\sigma$, and create nodes corresponding to these as children of C in T . Recursively repeat the procedure from each of them. This builds a split-tree tree T . We call such a (partial) function Φ a *determining* function for Γ and we say T is induced by Φ on Γ : $T = \Gamma|_\Phi$.

For any cluster C and the split node defined by $\Phi(C)$, we use $|P^* \cap E_{\pi_{\Phi(C)}}|$ to denote the number of bridge-edges of P^* in $\pi_{\Phi(C)}$, i.e. $|P^* \cap R_{\pi_{\Phi(C)}}| = E_{\pi_{\Phi(C)}}$. Let $\gamma = 3 \log n$, we can build a γ -split tree Γ for G in the following way. We start by making $C_0 = \{V\}$ to be the root of Γ and iteratively add levels to Γ . For a cluster node C , we generate its children split nodes as follows: we compute γ independent random partitions of C , denoted as $\{\pi_i\}$, $1 \leq i \leq \gamma_i$. Each $\pi_i = C_{i,1}, \dots, C_{i,\sigma_i}$ (where $\sigma_i \leq 2^{O(\kappa)}$) is obtained by taking a random $\frac{\Delta_C}{4}$ -net for C . Let R_{π_i} be the set of bridge-edges of C , i.e. edges in C crossing π_i : $R_{\pi_i} = \{u, v \in C : u \in C_{i,j}, v \in C_{i,j'}, \text{ for } j \neq j'\}$. For each $C_{i,j}$, we further consider the portal set $S_{i,j}$ which is a $\beta \Delta_{C_{i,j}}$ -net of it. Let R'_{π_i} be the set of portal edges of C : $R'_{\pi_i} = \{u, v \in C : u \in S_{i,j}, v \in S_{i,j'}, \text{ for } j \neq j'\}$. Since G is a doubling metric and S is a net, we find the following bound on the sizes of the portals: $|R'_{\pi_i}| \leq (\sigma_i |S_{i,j}|)^2$ which is bounded by $(\frac{16\kappa'\delta}{\epsilon})^{2\kappa}$. We create a child split node s_i for C for each π_i .

For a split node s , let C be its parent cluster node and let π_i be the partition (i.e. the $\frac{\Delta_C}{4}$ -net for C) corresponding to s . Then for each $C_{i,j} \in \pi_i$ we create a child cluster node C_j for s . The number of the children cluster nodes of s is at most $2^{O(\kappa)}$. We continue this process until each leaf node is a cluster node C with $|C| = O(1)$. From the construction above we know $\Delta_{C_i} \leq \frac{\Delta_C}{2}$ if C_i is a child of a split node s which is a child of C . Thus there are at most $2 \log \Delta_G = 2\delta$ levels in Γ . If we define the height of Γ as the number of levels of cluster nodes then the height of Γ is at most δ . The branching factor of Γ is then the product of the branching factor of a cluster node and a split node, which is at most $\gamma 2^{O(\kappa)}$. Hence the size of Γ is $(3 \log n 2^{O(\kappa)})^\delta$. Now we are ready to prove the following structure theorem for a near optimum solution.

► **Theorem 13** (structure theorem). *Let graph $G = (V, E)$ with doubling dimension κ , start and end nodes $s, t \in V$, and integer k be given as an instance of k -path. Assuming P^* is an optimum solution and $\gamma = 3 \log n$ we can construct a γ -split-tree where with probability at least $1 - \frac{1}{n}$ there exists a determining function Φ and a corresponding split-tree hierarchical decomposition $T = \Gamma|_\Phi$ of G and a nearly optimal solution P' such that P' visits at least k vertices and for any cluster $C \in T$ we have either:*

- 1) $|P' \cap R_{\pi_{\Phi(C)}}| \leq 2\kappa' \frac{\log n}{\epsilon}$, or
- 2) $|P' \cap R'_{\pi_{\Phi(C)}}| \leq (\frac{16\kappa'\delta}{\epsilon})^{2\kappa}$ and $\|P' \cap C\| \leq (1 + \epsilon) \|P^* \cap C\|$.

Proof. Suppose Γ is a γ -split-tree for G . We build P' iteratively based on P^* and at the same time build $\Phi(\cdot)$ and hence T from the top to bottom.

Initially we set P' to be P^* and start from the root cluster node C_0 of Γ . At any point when we are at a cluster node C we consider whether P' is sparse or dense with respect to C :

Sparse case. If $\|P' \cap C\| \leq \eta \cdot \Delta_C$, i.e. P' is η -sparse with respect to C , we don't modify $P' \cap C$ for when going down from C to any split node s of C . Consider any child split node s of C and let π_s be the partition of C according to the split node s . Consider the number of edges of $P' \cap C$ that are bridge-edges based on this partition i.e. $|E_{\pi_s}|$ (where $E_{\pi_s} = P' \cap R_{\pi_s}$). According to Lemma 9, $\mathbb{E}(|P' \cap R_{\pi_s}|) = \mathbb{E}(|E_{\pi_s}|) \leq \kappa' \frac{\log n}{\epsilon}$. Let the event Λ_s be the event that $|P' \cap R_{\pi_s}| \leq 2\kappa' \frac{\log n}{\epsilon}$. By Markov inequality $\Pr[\Lambda_s] \geq \frac{1}{2}$. Recall there are $\gamma = 3 \log n$ many children split nodes of c , i.e. γ independent random partitions of C . Thus the probability that for at least one child split node s of c , event Λ_s holds is at least $1 - (1 - \frac{1}{2})^{3 \log n} \geq 1 - \frac{1}{n^3}$. In this case we select split node s and define $\Phi(C) = s$ and consider each cluster child node of s iteratively (P' has not changed in this step).

Dense case. Suppose P' is η -dense with respect to C . Consider an arbitrary split child s of C . We will modify P' going down the split node s to be portal respecting with respect to π_s as described in Lemma 11. Assume $\pi_s = C_1, \dots, C_\sigma$. For each C_i let S_i be a set of portals. If we go down the split node s we modify P' to be portal-respecting for each C_i . Note that the set of edges P' crossing π_s after making it portal respecting with respect to π_s is a subset of R'_{π_s} . Thus $|P' \cap R'_{\pi_s}|$ is at most $|R'_{\pi_s}|$ which is at most $(\frac{16\kappa'\delta}{\epsilon})^{2\kappa}$ in this case. According to Lemma 11, the expected increase of length of P' after making it portal respecting with respect to π_s is at most $\frac{\epsilon}{2\delta} \|P' \cap C\|$. Let Λ'_s be the event that the increase of length of P' after making it portal respecting with respect to π_s is at most $\frac{\epsilon}{\delta} \|P' \cap C\|$. By Markov inequality, $\Pr[\Lambda'_s] \geq \frac{1}{2}$. Recall there are γ many children split nodes of C . Thus the probability that for at least one child split node s of C we have Λ'_s is at least $1 - (1 - \frac{1}{2})^{3 \log n} \geq 1 - \frac{1}{n^3}$. In this case we set $\Phi(C) = s$ for this particular split node s for which Λ'_s happens.

Therefore, regardless of whether P' is sparse or dense w.r.t. C , such s exists for cluster C with the probability at least $1 - \frac{2}{n^3}$ and we can define $\Phi(C)$. Once we have $\Phi(\cdot)$ defined for clusters at a level of Γ , we have determined the clusters at the same level of $T = \Gamma|_\Phi$. Note there are at most n cluster nodes in one level of T . Thus with probability at least $1 - \frac{2}{n^2}$ such split nodes exist for all cluster nodes in one level. Since height of Γ (and T) is δ , thus with probability at least $(1 - \frac{2}{n^2})^\delta \geq 1 - \frac{1}{n}$ (assuming that δ is polylogarithmic in n) such $\Phi(\cdot)$ is well defined over all levels. Note the increase of length of P' only occurs when P' is η -dense with respect to C and for any of these clusters C in $\Gamma|_\Phi$, the increase of length P' by modifying P' to be portal respecting with respect to $\pi_{\Phi(C)}$ is at most $\frac{\epsilon}{\delta} \|P' \cap C\|$. Since this $(1 + \frac{\epsilon}{\delta})$ -factor increase occurs each time we go down the decomposition from a cluster C to the next cluster level down, and the height of the decomposition is δ , thus inductively, for any cluster C in $\Gamma|_\Phi$: $\|P' \cap C\| \leq (1 + \frac{\epsilon}{\delta})^\delta \|P^* \cap C\| \leq e^\epsilon \|P^* \cap C\| \leq (1 + \epsilon') \|P^* \cap C\|$ for some $\epsilon' > 0$ depending on ϵ . Replacing ϵ with ϵ' we get $\|P' \cap C\| \leq (1 + \epsilon) \|P^* \cap C\|$. ◀

3.2 (ϵ, μ) -approximation for k -path

In this section we show a stronger bound on the length of the near optimum solution P' guaranteed by Theorem 13. Recall that a (ϵ, μ) -approximation for k -path is a path P with $|P| = k$ where $\|P\| \leq \|P^*\| + \epsilon \cdot \mathcal{E}_{P^*, \mu}$ where P^* is the optimum solution. We prove that path P' in Theorem 13 is in fact an (ϵ, μ) -approximation for $\mu = \lceil \frac{1}{\epsilon} \rceil + 1$.

Recall that in the proof of the structure theorem, the increase in length of P' only happens in cases when P' is dense with respect to a cluster C in Γ and we make the path portal respecting. Consider such a dense cluster C in the hierarchical decomposition $T = \Gamma|_\Phi$, i.e. P' is η -dense with respect to C . We show P' has high μ -excess in this case and the increased length of P' in C can be upper bounded by a factor of μ -excess of P^* .

► **Lemma 14.** *Let D be a set of disjoint clusters and Q be a path. Then $\mathcal{E}_{Q,2} \geq \sum_{C \in D} (\|Q \cap C\| - \Delta_C)$.*

Proof. Intuitively, this is saying if Q passes through several (disjoint) clusters in D then the excess of Q is at least as big as sum of excess of Q in those clusters.

Suppose start-end node of Q are u, v and let Q_0 be the path just consisting of u, v . By definition of the excess, $\mathcal{E}_{Q,2} = \|Q\| - \|Q_0\|$. Now consider following path Q' , which starts at u and follows Q but when it encounters a cluster C in D and it visits C for the first time it directly connects the start and end node of the subpath of Q in C . When it encounters a cluster C in D that is visited before, then bypasses C entirely, i.e. directly connects the last vertex in Q before it enters C this time and the first vertex in Q after it visits after it exits C this time. From the construction of Q' , if $C \in D$, then $\|Q' \cap C\| \leq \Delta_C$. Clearly for any cluster $C \notin D$: $\|Q' \cap C\| = \|Q \cap C\|$. Thus $\mathcal{E}_{Q,2} = \|Q\| - \|Q_0\| \geq \|Q\| - \|Q'\| = \sum_{C \in D} (\|Q \cap C\| - \|Q' \cap C\|) \geq \sum_{C \in D} (\|Q \cap C\| - \Delta_C)$. ◀

► **Theorem 15.** *Let graph $G = (V, E)$ with doubling dimension κ , start and end nodes $s, t \in V$, and integer k be given as an instance of k -path. Suppose P' and $\Gamma|_\Phi$ are as guaranteed by Theorem 13. Let $\mu = \lceil \frac{1}{\epsilon} \rceil + 1$, then P' is a (ϵ, μ) -approximation.*

Proof. Note that in the proof of Theorem 13, the increase of length of P' only occurs when P' is η -dense with respect to a cluster C . We generate a set of disjoint clusters D in $T = \Gamma|_\Phi$: we start from C_0 (the root of T) to generate D iteratively. If P' is η -dense with respect to C then add C to D , if P' is η -sparse with respect to C and C is a non-leaf cluster node then iteratively consider all children cluster nodes of C . Let D be the set of cluster nodes returned by this process. From the construction of D , clusters in D are disjoint, and for each cluster $C \in D$, P' is η -dense with respect to C and $\|P'\| - \|P^*\| = \sum_{C \in D} (\|P' \cap C\| - \|P^* \cap C\|)$.

Let $\{v_1, \dots, v_\mu\}$ be the optimal μ -jump of P^* and let $P_1^*, P_2^*, \dots, P_{\mu-1}^*$ be the subpaths divided by the vertices in this μ -jump, i.e. P_i^* is the subpath of P^* whose start and end are v_i and v_{i+1} , respectively. By definition of excess, $\mathcal{E}_{P^*,\mu} = \sum_{i=1}^{\mu-1} \mathcal{E}_{P_i^*,2}$. For the set D and each P_i^* , by Lemma 14, $\mathcal{E}_{P_i^*,2} \geq \sum_{C \in D} (\|P_i^* \cap C\| - \Delta_C)$. Thus $\mathcal{E}_{P^*,\mu} \geq \sum_{i=1}^{\mu-1} \sum_{C \in D} (\|P_i^* \cap C\| - \Delta_C)$, which is $\sum_{C \in D} (\|P^* \cap C\| - (\mu-1)\Delta_C)$. Recall from the proof of Theorem 13 that we modify P' when it is η -dense with respect to C (i.e. $\|P' \cap C\| > \frac{\log n}{\epsilon} \Delta_C$) and we always have $\|P' \cap C\| \leq (1+\epsilon)\|P^* \cap C\|$, which implies $\|P^* \cap C\| \geq \frac{\log n}{\epsilon(1+\epsilon)} \Delta_C$; in a sense P^* is also (almost) η -dense with respect to C . Since $\mu = \lceil \frac{1}{\epsilon} \rceil + 1$, thus $\|P^* \cap C\| - (\mu-1)\Delta_C \geq \frac{\|P^* \cap C\|}{2\epsilon}$ and $\mathcal{E}_{P^*,\mu} \geq \sum_{C \in D} \frac{\|P^* \cap C\|}{2\epsilon}$.

As in the proof Theorem 13, for any cluster $C \in T$, $\|P' \cap C\| \leq (1+\epsilon)\|P^* \cap C\|$. Thus $\|P'\| - \|P^*\| = \sum_{C \in D} (\|P' \cap C\| - \|P^* \cap C\|) \leq \sum_{C \in D} \epsilon \|P^* \cap C\| \leq 2\epsilon^2 \mathcal{E}_{P^*,\mu} \leq \epsilon \mathcal{E}_{P^*,\mu}$. ◀

We should note that we can generalize this proof slightly as follows. Suppose that we pick some arbitrary μ -jump of P^* (instead of the optimum) and consider $\tilde{P}_1^*, \dots, \tilde{P}_{\mu-1}^*$ which are the subpaths of P^* defined by that μ -jump. Then the same arguments show that $\sum_{i=1}^{\mu-1} \mathcal{E}_{\tilde{P}_i^*,2} \geq \sum_{i=1}^{\mu-1} \sum_{C \in D} (\|\tilde{P}_i^* \cap C\| - \Delta_C) = \sum_{C \in D} \|P^* \cap C\| - (\mu-1)\Delta_C$ which implies $\sum_{i=1}^{\mu-1} \mathcal{E}_{\tilde{P}_i^*,2} \geq \sum_{C \in D} \frac{\|P^* \cap C\|}{2}$. Using this one can show that at the end $\|P'\| \leq \|P^*\| + \epsilon \sum_{i=1}^{\mu-1} \mathcal{E}_{\tilde{P}_i^*,2}$. This slightly more general version will be used later when designing our algorithm for deadline TSP.

3.3 Finding A Near Optimum Solution For k -path

In this section we prove Theorem 3 by showing how we can find a near optimum solution for k -path as guaranteed by Theorem 15 using Dynamic Programming (DP). The DP is built on the γ -split tree Γ we compute. Consider an arbitrary cluster node C in Γ and the restriction

of the near optimum solution P' in the subgraph $G(C)$, denoted by P'_C . The set of edges of P'_C might be a collection of disjoint paths; one can imagine following along P' , it enters and exits C multiple times and each time it enters it follows a path in C until it exits again (assuming that s, t are not in C). If we denote the start-end of these subpaths in C as s_i, t_i , $1 \leq i \leq m$ (for some m) and if $|P'_C| = k_C$ then P'_C is a feasible solution for multi k -path with start-end pairs s_i, t_i and parameter k_C . This suggests a subproblem in our DP table will corresponds to an instance of multi-path k -path in a cluster C .

We define a subproblem in the DP as an instance of multi-path k -path with specified cluster node C , integer k_C and σ_C start-end pairs of vertices $\{(s_i, t_i)\}$ and the goal is to find a set of paths $\{P_i\}_{i=1}^{\sigma_C}$ such that P_i is a s_i - t_i path in C and $|\cup_{i=1}^{\sigma_C} P_i| = k_C$ while minimizing $\sum_{i=1}^{\sigma_C} |P_i|$. We use $A[C, k_C, (s_i, t_i)_{i=1}^{\sigma_C}]$ to denote the subproblem defined above and let the entry of the table store the optimal value of the solution to multi-path k -path for this subproblem. One should note that in the proof of Theorem 13, when we build P' from P^* , each time we go down a node of the split-tree, we might have a number of bridge-edges (when P' is η -sprase with respect to the current cluster C) or portal-edges (when P' is dense with respect to the current cluster C). When P' is sprase, the number of bridge-edges is at most $2\kappa' \frac{\log n}{\epsilon}$ (it was event Λ_s in the sparse case) and when P' is dense then the number of portal-edges is at most $(\frac{16\kappa'\delta}{\epsilon})^{2\kappa}$. Therefore, each step P' restricted to C might be chopped up into at most $(\frac{16\kappa'\delta}{\epsilon})^{2\kappa}$ subpaths, each with a new start-end point in a cluster C_i that is part of the partition of C . Given that the height of the split-tree is δ , the total number of start-end points for the instance of multi k -path at any cluster node C is at most $\delta(\frac{16\kappa'\delta}{\epsilon})^{2\kappa}$. This upper bounds σ in our subproblems.

Now we describe how to fill in the entries of the table. The base cases are when the cluster C has constant size $|C| = a$. Such instances can be solved using exhaustive search in $O(1)$ time.

Consider an arbitrary entry $A[C, k_C, (s_i, t_i)_{i=1}^{\sigma_C}]$ where for all split nodes children of C and every cluster children of them the entries of the table are computed. Consider any child split node s of C in Γ and let $C_1, C_2, \dots, C_g$ be the children cluster nodes of s in Γ . Recall π_s is the corresponding partition of C and R_{π_s} is the set of bridge-edges in C (crossing π_s). For each C_j let S_j be its portal set and recall R'_{π_s} is the set of edges crossing π_s only through $\{S_j\}$. We guess k_{C_j} for each C_j such that $\sum_{j=1}^g k_{C_j} = k_C$. We show how to guess start-end pairs $\{(s_i, t_i)_{i=1}^{\sigma_{C_j}}\}$ for each C_j and check the consistency of them. To do so we consider two cases: in the first case (meaning we guess we are in the sparse case) we guess a subset of R_{π_s} of size at most $2\kappa' \frac{\log n}{\epsilon}$; in the second case (meaning we assume we are in the dense case) we guess a subset of R'_{π_s} of size at most $(\frac{16\kappa'\delta}{\epsilon})^{2\kappa}$. Let E_{π_s} be the subset guessed in either case. Furthermore for each edge in E_{π_s} , we guess it is in which one of the σ_C paths with start-end pair $(s_i, t_i)_{i=1}^{\sigma_C}$ and for each path with start-end pair (s_i, t_i) we guess the order of the guessed edges appearing on the path. Specifically speaking, let $e_1, e_2, \dots, e_l$ be the edges guessed in the path with source-sink pair (s_i, t_i) appearing in this order. Let $C_{a_1}, C_{a_2}, \dots, C_{a_{l+1}}$ be the children cluster nodes of s that the path encounters following $e_1, e_2, \dots, e_l$, i.e. e_1 crosses between C_{a_1} and C_{a_2} , e_2 crosses between C_{a_2} and C_{a_3} , $\dots$, and e_l crosses between C_{a_l} and $C_{a_{l+1}}$. Then we set s_i and the endpoint of e_1 in C_1 to be a start-end pair in C_{a_1} , the endpoint of e_1 in C_{a_2} and the endpoint of e_2 in C_{a_2} to be a start-end pair in C_{a_2} , $\dots$, the endpoint of e_l in $C_{a_{l+1}}$ and t_i to be a start-end pair in $C_{a_{l+1}}$. By doing so we generate start-end pairs for each C_j and we sort them based on their ordering in s_i - t_i path and in the increasing order of i . This defines σ_{C_j} start-end pairs for each C_j .

► **Lemma 16.** *We can compute all entries $A[C, k_C, (s_i, t_i)_{i=1}^{\sigma_C}]$ in time $n^{O((\frac{\delta}{\epsilon})^{2\kappa+1})}$.*

The goal is to compute $A[c_0, k, (s, t)]$ where k and (s, t) are specified in the k -path instance. Proof of Theorem 3 follows from this lemma and Theorems 13 and 15.

Proof of Lemma 16. Formally, to compute $A[C, k_C, (s_i, t_i)_{i=1}^{\sigma_C}]$:

- Consider any child split node s of C , let $C_1, C_2, \dots, C_g$ be the children cluster nodes of s (where g can depend on s).
- Guess (i.e. try all possible values) k_{C_j} for each C_j such that $\sum_{j=1}^g k_{C_j} = k_C$.
- Let π_s be the corresponding partition of C and R_{π_s} be the bridge-edges of π_s . For each C_j let S_j be the portal set for it and let R'_{π_s} be the set of portal-edges of π_s . We consider both of the following two cases: 1) we guess a subset of R_{π_s} of size at most $2\kappa' \frac{\log n}{\epsilon}$; 2) we guess a subset of R'_{π_s} ; in both cases we denote the set of guessed edges as E_{π_s} .
- For each edge in E_{π_s} , we guess it is in which one of the σ_C paths with start-end pair $(s_i, t_i)_{i=1}^{\sigma_C}$ and for each path with start-end pair (s_i, t_i) we guess the order of the guessed edges appearing as described above. We generate $\{(s_i, t_i)\}_{i=1}^{\sigma_{C_j}}$ for each C_j accordingly. Then:
- We set $A[C, k_C, (s_i, t_i)_{i=1}^{\sigma_C}] = \min \sum_{j=1}^g A[C_j, k_{C_j}, (s_i, t_i)_{i=1}^{\sigma_{C_j}}] + \sum_{(u,v) \in E_{\pi_s}} d(u, v)$, where the minimum is taken over all tuples $s, k_{C_1}, \dots, k_{C_g}, (s_i, t_i)_{i=1}^{\sigma_{C_1}}, \dots, (s_i, t_i)_{i=1}^{\sigma_{C_g}}$ as described above.

Based on the structure theorem and the recurrence given above it is straightforward to see that this recurrence computes the best P' as guaranteed in Theorem 15.

Now we analyze the running time. First we show the time required to compute one entry of the DP table is $n^{O((\frac{\delta}{\epsilon})^{2\kappa+1})}$. In the recursion, for a cluster node C , there are $3 \log n$ children split nodes of C to consider. For a certain split node s , let $C_1, C_2, \dots, C_g$ be children cluster nodes of s , there are at most n^g guesses to for $\{k_{C_j}\}$ such that $\sum_{j=1}^g k_{C_j} = k_C$, which is at most $n^{2^{O(\kappa)}}$ because $g \leq 2^{O(\kappa)}$. For E_{π_s} : there are two cases, if $E_{\pi_s} \subset R_{\pi_s}$ then $|E_{\pi_s}| \leq 2\kappa' \frac{\log n}{\epsilon}$, there are at most $n^{(4\kappa' \frac{\log n}{\epsilon})}$ many possible E_{π_s} 's to consider; if $E_{\pi_s} \subset R'_{\pi_s}$, then because $|R'_{\pi_s}| \leq (\frac{16\kappa'\delta}{\epsilon})^{2\kappa}$ in this case there are at most $2^{(\frac{16\kappa'\delta}{\epsilon})^{2\kappa}} \leq n^{(\frac{16\kappa'\delta}{\epsilon})^{2\kappa}}$ many possible E_{π_s} 's to consider. To generate $(s_i, t_i)_{i=1}^{\sigma_{C_j}}$ for each C_j : for a certain E_{π_s} and for each edge in E_{π_s} we guess it is in which one of σ_C paths with start-end pair $(s_i, t_i)_{i=1}^{\sigma_C}$ and for each path with start-end pair (s_i, t_i) we guess the order of the edges appearing, which is at most $|E_{\pi_s}|!|E_{\pi_s}|^{\sigma_C}$ guessings. Note at each recursion it may increase at most $|E_{\pi_s}|$ many the number of start-end pairs and the depth of the recursion is δ . Thus $\sigma_C \leq \delta|E_{\pi_s}|$ and $|E_{\pi_s}|!|E_{\pi_s}|^{\sigma_C} \leq (\frac{16\kappa'\delta}{\epsilon})^{2\kappa} (2\kappa' \frac{\log n}{\epsilon})^{2\kappa} 2^{\kappa(\frac{16\kappa'\delta}{\epsilon})^{2\kappa}} \leq n^{O((\frac{\delta}{\epsilon})^{2\kappa+1})}$.

We show the size of the dynamic programming table is at most $n^{O((\frac{\delta}{\epsilon})^{2\kappa+1})}$. Recall an entry of the table is of the form $A[C, k_C, (s_i, t_i)_{i=1}^{\sigma_C}]$. For C , there are at most $(3 \log n 2^\kappa)^\delta$ cluster nodes in Γ because the size of Γ is at most $(3 \log n 2^\kappa)^\delta$. For k_C , there are at most n possible value of k_C to consider. For $\{(s_i, t_i)_{i=1}^{\sigma_C}\}$, there are at most $n^{2\sigma_C}$ start-end pairs to consider, which is at most $n^{2\delta(\frac{16\kappa'\delta}{\epsilon})^{2\kappa}}$ because σ_C is at most $\delta|E_{\pi_s}|$.

Therefore, computing the whole DP table and finding the near optimum path P' as in Theorem 15 takes at most $n^{O((\frac{\delta}{\epsilon})^{2\kappa+1})}$ time. $\blacktriangleleft$

3.4 Approximating P2P Orienteering on Doubling Metrics

In this section we prove Theorem 4 using the results of previous section for k -path.

► **Lemma 17.** *Let $G = (V, E)$ be graph with constant doubling dimension κ and given an instance of a P2P orienteering on G with specified budget B , start node $s \in V$ and end node $t \in V$. Let P^* be the optimal for this instance and $k = |P^*|$. Then we can get a s - t path that visits at least $(1 - \epsilon)k$ vertices in G with the length at most B .*

Proof. Let $\mu = \lceil \frac{1}{\epsilon} \rceil + 1$ and assume that $k \geq \mu^2$ (otherwise we can find P^* in polynomial time by exhaustive search in time $O(n^{1/\epsilon^2})$). We construct a subsequence of $1, \dots, k$ to define a μ -jump of P^* : Let $a_i = \lceil \frac{(i-1)(k-1)}{\mu-1} \rceil + 1$, $1 \leq i \leq \mu$. Note that $a_1 = 1$ and $a_\mu = k$. Let $P_1^*, P_2^*, \dots, P_{\mu-1}^*$ be the subpaths of P^* divided by $\{v_{a_1}, \dots, v_{a_\mu}\}$, i.e. P_i^* is a subpath of P^* with the source v_{a_i} and sink $v_{a_{i+1}}$.

For each P_i^* we consider the 2-excess of it and let P_j^* be the subpath with maximum 2-excess among $\{P_i^*\}_{i=1}^{\mu-1}$, i.e. $j = \arg \max_i \mathcal{E}_{P_i^*, 2}$. Note $|P_j^*| = a_{j+1} - a_j + 1 = (\lceil \frac{j(k-1)}{\mu-1} \rceil + 1) - (\lceil \frac{(j-1)(k-1)}{\mu-1} \rceil + 1) + 1 \leq \lceil \frac{k-1}{\mu-1} \rceil + 1$. Then let P' be the path exactly the same as P^* except P' directly connects v_{a_j} and $v_{a_{j+1}}$ in P_j^* . From the construction of P' ,

$$|P'| = k - |P_j^*| + 2 \geq k - \lceil \frac{k-1}{\mu-1} \rceil - 1 + 2 \geq k - \lfloor \frac{k-1}{\mu-1} \rfloor \geq (1 - \epsilon)k \quad \text{since } k \geq \mu^2, \text{ and} \quad (1)$$

$$\|P'\| = \|P^*\| - \mathcal{E}_{P_j^*, 2} = B - \mathcal{E}_{P_j^*, 2}. \quad (2)$$

We consider P' as a feasible solution for a k' -path instance with $k' = |P'| \geq (1 - \epsilon)k$ and $s, t \in V$. By Theorem 15, we can compute a (ϵ, μ) -approximation for this instance where $\mu = \lceil \frac{1}{\epsilon} \rceil + 1$, denoted as P'' :

$$|P''| \geq k' \geq (1 - \epsilon)k \quad \text{and} \quad (3)$$

$$\|P''\| \leq \|P'\| + \epsilon \mathcal{E}_{P', \mu} \leq B - \mathcal{E}_{P_j^*, 2} + \epsilon \mathcal{E}_{P', \mu} \leq B - \mathcal{E}_{P_j^*, 2} + \frac{1}{\mu-1} \mathcal{E}_{P', \mu}, \quad (4)$$

where the 2nd line uses (2). We consider the μ -jump of P' defined by $\langle v_{a_1}, \dots, v_{a_\mu} \rangle$. Note that this is also a w -jump of P^* . By the definition of excess and how we obtained P' from P^* (by short-cutting $P_{j,2}^*$):

$$\mathcal{E}_{P', \mu} \leq \|P'\| - \|\langle v_{a_1}, \dots, v_{a_\mu} \rangle\| = \sum_{i=1}^{\mu-1} \mathcal{E}_{P_i^*, 2} - \mathcal{E}_{P_j^*, 2}. \quad (5)$$

Thus, using (4) and (5): $\|P''\| \leq B - \mathcal{E}_{P_j^*, 2} + \frac{1}{\mu-1} (\sum_{i=1}^{\mu-1} \mathcal{E}_{P_i^*, 2} - \mathcal{E}_{P_j^*, 2}) \leq B$, where the last inequality follows from the fact that $\mathcal{E}_{P_j^*, 2}$ is the largest among all indices. $\blacktriangleleft$

However, for the P2P orienteering instance, P and k in Lemma 17 are unknown in advance. Therefore, we will consider all possible integers $1 \leq k \leq n$ and for each k we get the approximation for k -path on G with specified k and $s, t \in V$. We return the maximum k such that the length of path we get for k -path is at most B . This completes the proof of Theorem 4.

4 An Approximation Scheme for Deadline TSP in Doubling Metrics

In this section we prove Theorem 1. Our algorithm builds upon ideas from [20] for $O(1)$ -approximation for deadline TSP for general metrics combined with new ideas as well as those developed in the previous section to get an approximation scheme for deadline TSP on doubling metrics. Friggstad and Swamy [20] present an $O(1)$ -approximation for deadline TSP running in time $n^{O(\log n \Delta)}$ assuming that all distances are integers and at least 1. They use the notion of regret which is the same as 2-excess. Note if path P visits u and then v then short-cutting the subpath P_{uv} (replacing it with edge uv) will save a length which is exactly $\mathcal{E}_{P_{uv}, 2}$. They guess a sequence of vertices $v_0 = s, v_1, \dots, v_\ell$ of an optimal solution (with $\ell = O(\log \Delta)$) such that the 2-excess of the subpaths of optimal increase geometrically: $\mathcal{E}_{P_{v_i v_{i+1}}, 2} \geq \alpha^i$, where α is some constant satisfying $1 + \alpha \geq \alpha^2$. They consider a set of P2P orienteering instances with start node v_i , end node v_{i+1} , and length budget $d(v_i, v_{i+1}) + \alpha^i$.

These instances are not independent however, hence this is a more general problem that we call multi-path orienteering with group budget constraints (see [20]). They show given an β -approximation for P2P orienteering, at another $O(1)$ -factor loss, one can turn it into an $O(\beta)$ -approximation for multi-group multi-path orienteering. Then they concatenate these paths; these paths are not respecting the deadlines however. In order to make them deadline respecting, from every three consecutive paths they shortcut two of them, so another $O(1)$ -factor loss. The saving for the shortcutting of two paths is enough for the deadline of every vertex in the third path being satisfied. To obtain $O(1)$ -approximation for P2P orienteering instances with groups, they use known reductions from the problem of maximum coverage with group budgets to classic maximum coverage and use algorithm of [14] to get a constant approximation via a reduction to classic P2P orienteering (see [20] for more details). Putting everything together, to obtain an $O(1)$ -approximation for deadline TSP they lose $O(1)$ factor in three steps.

In our setting in order to get a $(1 + \varepsilon)$ -approximation for deadline TSP on graphs with bounded doubling dimension, we have to change all these steps so that we don't lose more than $(1 + \varepsilon)$ factor in any step. It turns out it becomes significantly more complex to maintain an at most $(1 + \varepsilon)$ loss in any step. As in [20], we assume distances are integer and ≥ 1 (this can be done as pointed out in [20] by scaling).

Overview of the proof. Suppose we have guessed a sequence of vertices $\langle v_0 = s, v_1, \dots, v_m \rangle$ of an optimum solution P^* where the μ -excess of the sub-path $P_{v_i, v_{i+1}}^*$ is (at least) α^i , where $\alpha = 1 + \varepsilon$ (for simplicity assume the increase in excess is exactly α^i). We also assume we have guessed the lengths of these sub-paths, say $\|P_{v_i, v_{i+1}}^*\| = B_i$. Note that the vertices visited in $P_{v_i, v_{i+1}}^*$ all must have a deadline at least as big as the visit time of v_i in P^* (which we have guessed since we know all previous B_j 's, $j < i$); let W_i denote this set of vertices. Let $\mathcal{I}_i$ be the P2P orienteering instance with start-end pair v_i, v_{i+1} , budget B_i , where the vertices allowed to visit are W_i . Note that the sub-paths $P_{v_i, v_{i+1}}^*$ form a solution to these instances $\mathcal{I}_i$ for $i \geq 1$. If we have v_i 's and B_i 's, we can try to solve these $O(\log \Delta)$ instances simultaneously (our DP described for P2P orienteering can be expanded to handle when we have $O(\log \Delta)$ instances to be solved on the same ground set simultaneously).

One problem is that the vertices visited in $\mathcal{I}_i$ might be violating their deadlines slightly. We will show that this violation will be small (using the assumption that the excess of subpath $P_{v_i, v_{i+1}}^*$ was α^i) and that by using a similar technique as we did to convert an approximation for k -path to an approximation for P2P orienteering we can drop a small fraction of vertices visited in all instances such that the total saving in time achieved for all $\mathcal{I}_j$ with $j < i$ is enough to ensure all the (remaining) vertices in $\mathcal{I}_i$ are visited before their deadline. Now we start explaining the details of the proof.

Let $G = (V, E)$ be a graph with constant doubling dimension κ , given a start node $s \in V$ and deadline $D(v)$ for all $v \in V$ as an instance $\mathcal{I}$ of deadline TSP on G . Suppose μ is a constant (will be fixed to be $\lfloor \frac{1}{\varepsilon} \rfloor + 1$ later and let $\alpha = (1 + \varepsilon)$). Let P^* be an optimum solution for this instance and $\langle v_0, v_1, \dots, v_m \rangle$ be a sequence of vertices in P^* satisfying the following properties:

- $v_0 = s$ is the start node of P^* .
- v_{i+1} is the first vertex in P^* after v_i with $\mathcal{E}_{P_{v_i, v_{i+1}}^*, \mu} > \alpha^i$, except possibly for v_m (the last vertex of P^*).

So each vertex v_{i+1} is the first vertex along the optimum after v_i such that the μ -excess of the subpath $P_{v_i, v_{i+1}}^*$ is at least α^i . We will be guessing these vertices v_i 's eventually and try to find (good) approximations of $P_{v_i, v_{i+1}}^*$. We can assume that $|P_{v_i, v_{i+1}}^*| \geq \mu^2$,

otherwise we can compute $P_{v_i v_{i+1}}^*$ exactly using exhaustive search. We also denote the vertex on P^* immediately before v_{i+1} by v_i' . Note $\|P^*\| \leq n\Delta_G$, thus $m \leq h\delta$ (where $\delta = \log \Delta$) for some constant $h = h(\varepsilon) > 0$. For each $0 \leq i < m$, we break $P_{v_i v_{i+1}}^*$ into $\mu - 1$ subpaths of (almost) equal sizes, denoted as $P_{i,j}^*$, $1 \leq j < \mu$, by selecting a μ -jump $J_i : v_i = u_i^1, u_i^2, u_i^3, \dots, u_i^{\mu-1}, u_i^\mu = v_{i+1}$ as follows. Assume $P_{v_i v_{i+1}}^* = \langle v_{i,1}, \dots, v_{i,k_i} \rangle$ where $v_i = v_{i,1}$ and $v_{i,k_i} = v_{i+1}$, let $a_j = \lceil \frac{(j-1)(k_i-1)}{\mu-1} \rceil + 1$ then $a_1 = 1$, $a_\mu = k_i$, and if we consider $v_{i,a_1}, \dots, v_{i,a_\mu}$ then we obtain J_i by letting $v_{i,a_j} = u_i^j$. Suppose $J_\mu^* = J_\mu^*(P_{v_i v_{i+1}}^*)$ is the optimum μ -jump of $P_{v_i v_{i+1}}^*$, which is the μ -jump with the maximum length. Recall that $\mathcal{E}_{P_{v_i v_{i+1}}^*, \mu}$ is the μ -excess of $P_{v_i v_{i+1}}^*$ and with $B_i = \|J_\mu^*(P_{v_i v_{i+1}}^*)\| + \mathcal{E}_{P_{v_i v_{i+1}}^*, \mu}$ we have $\|P_{v_i v_{i+1}}^*\| = B_i$. To simplify the notation we denote the μ -excess of subpath $P_{v_i v_{i+1}}^*$ by $\mathcal{E}_{i,\mu}^*$. We also use $\mathcal{E}_{i,\mu}'$ to denote the μ -excess of path $P_{v_i v_i'}^*$. From the definition of $\langle v_0, v_1, \dots, v_m \rangle$ it follows that $\mathcal{E}_{i,\mu}^* > \alpha^i$ and $\mathcal{E}_{i,\mu}' \leq \alpha^i - 1$. Let v be an arbitrary vertex in $P_{v_i v_{i+1}}^*$ that falls in between u_i^j and u_i^{j+1} . We use $\|J_i(v_i, u_i^j)\|$ to denote the length of J_i from v_i to u_i^j (i.e. following along J_i from the start node v_i to u_i^j). Define $L_{i,j} = \sum_{j=0}^{i-1} \|P_{v_j v_{j+1}}^*\| + \|J_i(v_i, u_i^j)\|$. Note that the visiting time of v in P^* (and hence the deadline of v) is lower bounded by $L_{i,j}$.

Let $N_{i,j} = \{v : D(v) \geq L_{i,j}\}$. Observe that if we consider $P_{v_i v_{i+1}}^*$ broken up into several legs $P_{u_i^j u_i^{j+1}}^*$, $1 \leq j < \mu$, then it is a P2P orienteering instance with start node v_i , end node v_{i+1} and given extra intermediate nodes u_i^j and the path is supposed to go through these intermediate nodes in this order; so it consists of $\mu - 1$ legs where leg j is between u_i^j, u_i^{j+1} and uses vertices in $N_{i,j} \subset V$ and total budget $B_i = \|J_\mu^*(P_{v_i v_{i+1}}^*)\| + \mathcal{E}_{i,\mu}^*$.

► **Definition 18** (multiple-groups-legs orienteering). *Let $G = (V, E)$ be a graph, given m groups each with $\mu - 1$ legs, where leg ℓ of group i ($0 \leq i < m$, $1 \leq \ell < \mu$) has start and end node pair $(s_{i,\ell}, t_{i,\ell})$ and can use vertices from $N_{i,\ell} \subseteq V$ (we have the property that end-node of leg ℓ is the same as start node of leg $\ell + 1$: $t_{i,\ell} = s_{i,\ell+1}$), total budget for all the legs of group i is B_i . The goal is to find a collection of paths $Q_{i,\ell}$, for $0 \leq i < m$, $1 \leq \ell \leq \mu - 1$, such that $Q_{i,\ell}$ is a $s_{i,\ell} - t_{i,\ell}$ path (and so concatenation of all legs of group i gives a single path from start node of the first leg to the last node of leg $\mu - 1$), such that $\sum_{\ell=1}^{\mu-1} \|Q_{i,\ell}\| \leq B_i$ and $|\cup_{i=0}^{m-1} \cup_{\ell=1}^{\mu-1} (Q_{i,\ell} \cap N_{i,\ell})|$ is maximized.*

Note $P_{u_i^j u_i^{j+1}}^*$ ($0 \leq i < m$, $1 \leq j < \mu$) is a feasible solution of the multiple groups-legs orienteering instance with groups $0 \leq i < m$ and legs $1 \leq j < \mu$, start and end node pairs (u_i^j, u_i^{j+1}) , budgets B_i and subset $N_{i,j}$. Consider the instance restricted to group i alone an instance of multiple-leg P2P orienteering where we have to find a $v_i v_{i+1}$ -path that goes through all u_i^j 's in order (so has $\mu - 1$ legs) and has budget B_i ; call this instance $\mathcal{I}_i$. Note $P_{v_i v_{i+1}}^*$ is a feasible solution to $\mathcal{I}_i$, thus the optimal of $\mathcal{I}_i$ visits at least $|P_{v_i v_{i+1}}^*|$ many vertices in $\cup_{j=1}^{\mu-1} N_{i,j}$. We consider all $\mathcal{I}_i$'s concurrently, i.e. an instance with pairs of start-end nodes $v_i v_{i+1}$ for group i where each group is also required to go through vertices of J_i in that order and thus has $\mu - 1$ legs where each leg has start-end pair u_i^j, u_i^{j+1} , total budget B_i for all the legs of group i , and the subset $N_{i,j} \subset V$, $0 \leq i < m$, $1 \leq j < \mu$ for leg j of group i . For a path Q , let $Q \cap N_{i,j}$ be the set of vertices Q visited in $N_{i,j}$. Using an argument similar to that of Theorem 13 we can show there is a $(1 + \epsilon)$ -approximation i.e. a set of paths $Q'_{i,j}$ such that $Q'_{i,j}$ is a u_i^j, u_i^{j+1} -path and if we define concatenation of different legs of group i by $Q'_i = Q'_{i,1} + \dots + Q'_{i,\mu-1}$ then:

$$\|Q'_i\| = \sum_{j=1}^{\mu-1} \|Q'_{i,j}\| \leq \|P_{v_i v_{i+1}}^*\| - \varepsilon \mathcal{E}_{i,\mu}^* \quad \text{and} \quad (6)$$

$$|\bigcup_{i=0}^{m-1} Q'_i| = |\bigcup_{i=0}^{m-1} \cup_{j=1}^{\mu-1} (Q'_{i,j} \cap N_{i,j})| \geq (1 - 4\epsilon) |P^*|. \quad (7)$$

We also show that if v is visited by $Q'_{i,j}$ then if $Q'_{i,j}(u_i^j, v)$ denotes the segment of path $Q'_{i,j}$ from u_i^j to v , then the length of the segment from v_i to v in Q'_i can be upper bounded:

$$\|Q'_i(v_i, v)\| = \sum_{\ell=1}^{j-1} \|Q'_{i,\ell}\| + \|Q'_{i,j}(u_i^j, v)\| \leq \|J_i(v_i, u_i^j)\| + (1 - \varepsilon)\mathcal{E}'_{i,\mu}. \quad (8)$$

We will prove the existence of such paths $Q'_{i,j}$ with a structure in the next theorem and also how to find these using a DP. For now suppose we have found such paths Q'_i as described above. We concatenate all these paths to obtain the final answer $\mathcal{Q} = Q'_0 + Q'_1 + \dots + Q'_{m-1}$. We show the vertices of $\mathcal{Q}$ are visited before their deadlines and hence we have an approximation for deadline TSP. Given the bounds given for the sizes of Q'_i 's in (6), the number of vertices visited overall (respecting their deadlines) is at least $(1 - 4\varepsilon)|P^*|$.

To see why the vertices in $\mathcal{Q}$ respect their deadlines consider an arbitrary node $v \in Q'_i$. Note that each Q'_i contains the vertices in J_i (as those are the vertices that define $\mu - 1$ legs of the i 'th group). Suppose v is visited in $Q'_{i,j}$, i.e. between u_i^j and u_i^{j+1} . Therefore, the visit time of v in $\mathcal{Q}$, i.e. $\|\mathcal{Q}_{sv}\|$ is bounded by:

$$\begin{aligned} \|\mathcal{Q}_{sv}\| &= \sum_{\ell=0}^{i-1} \|Q'_\ell\| + \|Q'_i(v_i, v)\| \\ &\leq \sum_{\ell=0}^{i-1} (\|P^*_{v_\ell v_{\ell+1}}\| - \varepsilon\mathcal{E}^*_{\ell,\mu}) + \|J_i(v_i, u_i^j)\| + (1 - \varepsilon)\mathcal{E}'_{i,\mu} \quad \text{using (6) and (8)} \\ &= L_{i,j} + (1 - \varepsilon)\mathcal{E}'_{i,\mu} - \varepsilon \sum_{\ell=0}^{i-1} \mathcal{E}^*_{\ell,\mu} \leq D(v), \end{aligned}$$

where the last inequality follows from the fact that $\mathcal{E}'_{i,\mu} \leq \alpha^i - 1$ and $\mathcal{E}^*_{\ell,\mu} > \alpha^\ell$ so $\varepsilon \sum_{\ell=0}^{i-1} \mathcal{E}^*_{\ell,\mu} \geq \varepsilon \sum_{\ell=0}^{i-1} \alpha^\ell = (\alpha^i - 1) \geq \mathcal{E}'_{i,\mu}$.

So we need to show the existence of Q'_i as described and how to find them. The road to get these paths Q'_i is an extension of what we had for multi-path orienteering in the previous section and is obtained by a DP that computes multi-group-legs with multiple paths (i.e. instances $\mathcal{I}_i$'s) concurrently. The proof of following theorem is an extension of that of Theorems 13 and 15.

► **Theorem 19** (multi-groups-legs multi-paths orienteering structure theorem). *Let $G = (V, E)$ be a graph with constant doubling dimension κ , given $s \in V$ and $D(v)$ for all v as an instance of deadline TSP and P^* be an optimal solution. Let v_i ($0 \leq i < m$), u_i^j ($0 \leq i < m$, $1 \leq j < \mu$), B_i , and $N_{i,j}$ as described above for $\mu = \lfloor \frac{1}{\varepsilon} \rfloor + 1$. Consider a multi-groups-legs orienteering instance with groups $0 \leq i < m$ and legs $1 \leq j < \mu$, start and end node pairs (u_i^j, u_i^{j+1}) , budgets B_i , and subset $N_{i,j}$. then we can construct a γ -split-tree (with $\gamma = n^{3h\delta}$) such that with probability at least $1 - \frac{1}{n}$ there exists a determining function Φ and a corresponding split-tree hierarchical decomposition $T = \Gamma|_\Phi$ of G and a structured solution of the multi-groups-legs orienteering instance $Q'_{i,j}$, ($0 \leq i < m$, $1 \leq j < \mu$) such that if we define $Q'_i = Q'_{i,1} + \dots + Q'_{i,\mu-1}$ for each $0 \leq i < m$, then:*

1. $|\cup_{i=0}^{m-1} \cup_{j=1}^{\mu-1} (Q'_{i,j} \cap N_{i,j})| \geq (1 - 4\varepsilon) |\cup_{i=0}^{m-1} \cup_{j=1}^{\mu-1} P^*_{u_i^j, u_i^{j+1}}| = (1 - 4\varepsilon)|P^*|$.
2. $\|Q'_i\| = \sum_{j=1}^{\mu-1} \|Q'_{i,j}\| \leq B_i - \varepsilon\mathcal{E}^*_{i,\mu} = \|J_\mu^*(P^*_{v_i, v_{i+1}})\| + (1 - \varepsilon)\mathcal{E}^*_{i,\mu}$ and for any vertex visited by Q'_i , say v visited in $Q'_{i,j}$ we have the length of the path from v_i to v in Q'_i , denoted by $\|Q'_i(v_i, v)\|$, satisfies: $\|Q'_i(v_i, v)\| \leq \|J_\mu^*(P^*_{v_i, v_i'})\| + (1 - \varepsilon)\mathcal{E}'_{i,\mu}$.
3. For any cluster $C \in T$ and a partition $\pi_{\Phi(C)}$ of C defining its children in T , let $R_{\pi_{\Phi(C)}}$, $R_{\pi_{\Phi(C)}}$ are the set of bridge-edges and portal-edges of C cut by the partition, we have either:

- $|Q'_i \cap R_{\pi_{\Phi(C)}}| \leq 2\kappa' \frac{\log n}{\epsilon}$ or
- $|Q'_i \cap R'_{\pi_{\Phi(C)}}| \leq (\frac{16\kappa'\delta}{\epsilon})^{2\kappa}$.

Using a more complex DP and we can find a near optimum structured solution as guaranteed by this Theorem. This DP tries to solve multi-groups-legs multi-paths orienteering simultaneously.

References

- 1 Esther M. Arkin, Joseph S. B. Mitchell, and Giri Narasimhan. Resource-constrained geometric network optimization. In Ravi Janardan, editor, *Proceedings of the Fourteenth Annual Symposium on Computational Geometry, Minneapolis, Minnesota, USA, June 7-10, 1998*, pages 307–316. ACM, 1998. doi:10.1145/276884.276919.
- 2 Sanjeev Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *J. ACM*, 45(5):753–782, 1998. doi:10.1145/290179.290180.
- 3 Sandip Banerjee, Yair Bartal, Lee-Ad Gottlieb, and Alon Hovav. Novel properties of hierarchical probabilistic partitions and their algorithmic applications. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*, pages 1724–1767. IEEE, 2024. doi:10.1109/FOCS61266.2024.00107.
- 4 Nikhil Bansal, Avrim Blum, Shuchi Chawla, and Adam Meyerson. Approximation algorithms for deadline-tsp and vehicle routing with time-windows. In László Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 166–174. ACM, 2004. doi:10.1145/1007352.1007385.
- 5 Nikhil Bansal and Viswanath Nagarajan. On the adaptivity gap of stochastic orienteering. *CoRR*, abs/1311.3623, 2013. arXiv:1311.3623.
- 6 Yair Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 184–193. IEEE Computer Society, 1996. doi:10.1109/SFCS.1996.548477.
- 7 Yair Bartal, Lee-Ad Gottlieb, and Robert Krauthgamer. The traveling salesman problem: Low-dimensionality implies a polynomial time approximation scheme. *SIAM J. Comput.*, 45(4):1563–1581, 2016. doi:10.1137/130913328.
- 8 MohammadHossein Bateni and Julia Chuzhoy. Approximation algorithms for the directed k -tour and k -stroll problems. *Algorithmica*, 65(3):545–561, 2013. doi:10.1007/S00453-011-9610-6.
- 9 Avrim Blum, Shuchi Chawla, David R Karger, Terran Lane, Adam Meyerson, and Maria Minkoff. Approximation algorithms for orienteering and discounted-reward tsp. *SIAM Journal on Computing*, 37(2):653–670, 2007. doi:10.1137/050645464.
- 10 Kevin Buchin, Mart Hagedoorn, Guangping Li, and Carolin Rehs. Orienteering (with time windows) on restricted graph classes. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 151–165. Springer, 2025. doi:10.1007/978-3-031-82670-2_12.
- 11 T.-H. Hubert Chan and Shaofeng H.-C. Jiang. Reducing curse of dimensionality: Improved PTAS for TSP (with neighborhoods) in doubling metrics. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 754–765. SIAM, 2016. doi:10.1137/1.9781611974331.CH54.
- 12 Kamalika Chaudhuri, Brighten Godfrey, Satish Rao, and Kunal Talwar. Paths, trees, and minimum latency tours. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 36–45. IEEE, 2003. doi:10.1109/SFCS.2003.1238179.
- 13 Chandra Chekuri, Nitish Korula, and Martin Pál. Improved algorithms for orienteering and related problems. *ACM Transactions on Algorithms (TALG)*, 8(3):1–27, 2012. doi:10.1145/2229163.2229167.

- 14 Chandra Chekuri and Amit Kumar. Maximum coverage problem with group budget constraints and applications. In *International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 72–83. Springer, 2004. doi:10.1007/978-3-540-27821-4_7.
- 15 Chandra Chekuri and Martin Pál. A recursive greedy algorithm for walks in directed graphs. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, pages 245–253. IEEE Computer Society, 2005. doi:10.1109/SFCS.2005.9.
- 16 Ke Chen and Sarel Har-Peled. The euclidean orienteering problem revisited. *SIAM Journal on Computing*, 38(1):385–397, 2008. doi:10.1137/060667839.
- 17 Vincent Cohen-Addad. Bypassing the surface embedding: approximation schemes for network design in minor-free graphs. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 343–356. ACM, 2022. doi:10.1145/3519935.3520049.
- 18 Boaz Farbstein and Asaf Levin. Deadline TSP. *Theor. Comput. Sci.*, 771:83–92, 2019. doi:10.1016/J.TCS.2018.11.016.
- 19 Zachary Friggstad and Chaitanya Swamy. Compact, provably-good lps for orienteering and regret-bounded vehicle routing. In Friedrich Eisenbrand and Jochen Könemann, editors, *Integer Programming and Combinatorial Optimization - 19th International Conference, IPCO 2017, Waterloo, ON, Canada, June 26-28, 2017, Proceedings*, volume 10328 of *Lecture Notes in Computer Science*, pages 199–211. Springer, 2017. doi:10.1007/978-3-319-59250-3_17.
- 20 Zachary Friggstad and Chaitanya Swamy. A constant-factor approximation for directed latency in quasi-polynomial time. *J. Comput. Syst. Sci.*, 126:44–58, 2022. doi:10.1016/J.JCSS.2021.12.001.
- 21 Jie Gao, Su Jia, Joseph S. B. Mitchell, and Lu Zhao. Approximation algorithms for time-window TSP and prize collecting TSP problems. In Ken Goldberg, Pieter Abbeel, Kostas E. Bekris, and Lauren Miller, editors, *Algorithmic Foundations of Robotics XII, Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics, WAFR 2016, San Francisco, California, USA, December 18-20, 2016*, volume 13 of *Springer Proceedings in Advanced Robotics*, pages 560–575. Springer, 2016. doi:10.1007/978-3-030-43089-4_36.
- 22 Naveen Garg, Sanjeev Khanna, and Amit Kumar. Hardness of approximation for orienteering with multiple time windows. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 2977–2990. SIAM, 2021. doi:10.1137/1.9781611976465.177.
- 23 Lee-Ad Gottlieb, Robert Krauthgamer, and Havana Rika. Faster algorithms for orienteering and k -tsp. *Theor. Comput. Sci.*, 914:73–83, 2022. doi:10.1016/J.TCS.2022.02.013.
- 24 Anupam Gupta, Robert Krauthgamer, and James R Lee. Bounded geometries, fractals, and low-distortion embeddings. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 534–543. IEEE, 2003. doi:10.1109/SFCS.2003.1238226.
- 25 Anupam Gupta, Ravishankar Krishnaswamy, Viswanath Nagarajan, and R. Ravi. Running errands in time: Approximation algorithms for stochastic orienteering. *Math. Oper. Res.*, 40(1):56–79, 2015. doi:10.1287/MOOR.2014.0656.
- 26 Aditya Jayaprakash and Mohammad R. Salavatipour. Approximation schemes for capacitated vehicle routing on graphs of bounded treewidth, bounded doubling, or highway dimension. *ACM Trans. Algorithms*, 19(2):20:1–20:36, 2023. doi:10.1145/3582500.
- 27 Haotian Jiang, Jian Li, Daogao Liu, and Sahil Singla. Algorithms and adaptivity gaps for stochastic k -tsp. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, volume 151 of *LIPICs*, pages 45:1–45:25. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.ITCS.2020.45.
- 28 Philip N. Klein. A linear-time approximation scheme for TSP in undirected planar graphs with edge-weights. *SIAM J. Comput.*, 37(6):1926–1952, 2008. doi:10.1137/060649562.

- 29 Joseph SB Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric tsp, k-mst, and related problems. *SIAM Journal on computing*, 28(4):1298–1309, 1999. doi:10.1137/S0097539796309764.
- 30 Viswanath Nagarajan and R. Ravi. The directed orienteering problem. *Algorithmica*, 60(4):1017–1030, 2011. doi:10.1007/S00453-011-9509-2.
- 31 Kinter Ren and Mohammad R. Salavatipour. Approximation schemes for orienteering and deadline tsp in doubling metrics, 2024. doi:10.48550/arXiv.2405.00818.
- 32 Kunal Talwar. Bypassing the embedding: algorithms for low dimensional metrics. In László Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 281–290. ACM, 2004. doi:10.1145/1007352.1007399.
- 33 Kunal Talwar. Bypassing the embedding: algorithms for low dimensional metrics. In *Proc. of 36th ACM Symp. on Theory of Computing (STOC)*, 2004. doi:10.1145/1007352.1007399.