A quasi-polynomial time approximation scheme for Euclidean capacitated vehicle routing*

Aparna Das   Claire Mathieu†

Abstract

In the capacitated vehicle routing problem, introduced by Dantzig and Ramser in 1959, we are given the locations of n customers and a depot, along with a vehicle of capacity k, and wish to find a minimum length collection of tours, each starting from the depot and visiting at most k customers, whose union covers all the customers. We give a quasi-polynomial time approximation scheme for the setting where the customers and the depot are on the plane, and distances are given by the Euclidean metric.

1 Introduction

Dantzig and Ramser introduced the vehicle routing problem (VRP) in 1959 and gave a linear programming based algorithm whose *"calculations may be readily performed by hand or automatic digital computing machine"*[10]. Since its introduction, VRP has come to describe a class of problems where the objective is to find low cost delivery routes from depots to customers using a vehicle of limited capacity. The VRP has been widely studied by researchers in Operations Research and Computer Science and several books (see [22], [14] and [12], among others) have been written on the problems. VRP problems have direct application to business delivery routing in various industries where transportation costs matter such as food and beverage distribution, and package and newspaper delivery. Toth and Vigo report on several businesses that have saved between 5 and 20% of total costs by solving VRP problems via computerized models [22].

Capacitated vehicle routing problem. We study the most basic form of the vehicle routing problem, the capacitated version (CVRP), where the input consists of an integer k representing the capacity of the vehicle, and $n+1$ points representing the locations of n customers and one depot. The objective is to find a collection of tours, each starting at the depot and visiting at most k customers, whose union cover all n customers, such that the sum of the lengths of the tours is minimized. The CVRP is also called the k-tours problem in the Computer Science literature [2, 5]. We study the Euclidean version of the problem where customers and the depot are on the Euclidean plane.

Popular CVRP heuristics. The CVRP has several well-known heuristics, each with many variations. In its basic version, the "savings" algorithm of [8] starts with the simplest feasible solution, a set of n tours each visiting a single point and repeatedly chooses two tours and "merges" them, going directly from the last location of the first tour to the first location of the second tour, thus shortcutting one trip to the depot. The algorithm is greedy; each iteration merges the two tours yielding the largest savings. The "sweep" heuristic of [11] is similar to the Jarvis convex hull algorithm. The "seeding" procedure of [13] places seeds at well-chosen locations in the plane, associates at most k locations to each seed so as to minimize the total distance from locations to associated seeds, and finally builds one tour for each seed. General-purpose heuristics such as local search, Tabu search, genetic algorithms, neural networks and ant colony optimization schemes have also been applied to this problem. See [17, 22] for details.

The above heuristics seem to perform well on common test beds [17] however their worst case behavior (i.e. approximation factors) have not been pinned down yet. Simple examples show that the basic version of the above heuristics are not approximation schemes even on the Euclidean plane. Larson and Odoni [18] give an example (figure 6.3.2) where the savings algorithm produces a solution that is 11% more than OPT. Figure 1 shows an example where the sweep heuristic performs arbitrarily worse than OPT and one where the seeding algorithm's tour has length 50% more than OPT.

Approximation algorithms for CVRP. Partial results are known about the approximability of CVRP. When the capacity of the vehicle k is 2, the problem can be solved using minimum weight matching. The metric case was shown to be APX-complete for all $k \geq 3$. Asano et al. presented a reduction from H-matching for $k = O(1)$ and there is a simple reduction from the traveling salesman problem (TSP) for larger k [6]. Constant factor approximation with performance $(1+\alpha)$

*Both authors supported by NSF grant CCF-0728816.

†Both authors from Brown University-Computer Science.

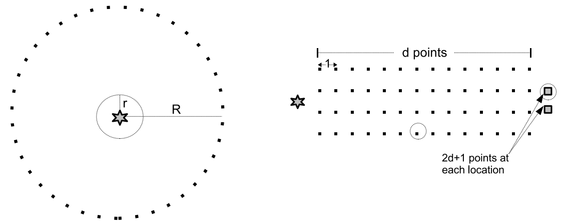

Figure 1: Bad example for Sweep and Seeding. The depot is the star. (a) There are $n/2$ customers on both the inside and outside circles. Take $k = n$. The sweep tour uses many edges between the two circles and has length $\approx R \cdot n/2$. OPT places all customers on the inside circle at the end of its tour and has cost at most $4\pi(R + r)$. (b) Take $k = 4d + 1$, d large and let dashed circles be the seeds. OPT has length about $4d$, using two tours each covering $2d$ points from the middle and $2d + 1$ points from the far right. Seeding has length about $6d$; at most $k = 4d + 1$ customers from the far right are assigned to the far right seed and covered by tour of length $2d$ and the remaining customers are covered by a tour of length about $4d$.

where α is the best approximation factor for TSP, were presented by Haimovich and Rinnooy Kan [15].

The hardness of the CVRP is closely related to that of TSP. Since the TSP has a polynomial time approximation schemen (PTAS) in the Euclidean plane, it has naturally been conjectured that the CVRP also has a PTAS in that setting [3]. Indeed, in the case of very large capacity, $k = \Omega(n)$, Arora's PTAS for TSP easily extends to a PTAS for CVRP. In the case of small capacity, $k = O(\log n / \log \log n)$, Asano et al. [5] presented a PTAS extending [15]; and very recently, for slightly larger capacity, $k \leq 2^{\log^{\delta} n}$ (where δ a function of ϵ), Adamaszek et al. presented a PTAS using the algorithm of this paper as a black box [1].

Our result. We present a quasi-polynomial time approximation scheme for the entire range of k.

Theorem 1.1. *(Main Theorem) Algorithm 1 is a randomized quasi-polynomial time approximation scheme for the two dimensional Euclidean capacitated vehicle routing problem. Given $\epsilon > 0$, it outputs a solution with expected length $(1+O(\epsilon))OPT$, in time $n^{\log^{O(1/\epsilon)} n}$. The Algorithm can be derandomized.*

Our running time is quasi-polynomial, and it is seriously super-polynomial, so in itself it's unlikely to lead to much in the way of practical improvements. Our attempts to get truly polynomial running time have been unsuccessful so far; one possible direction might be to study the easier version of the problem with soft capacity constraints, where OPT is required to use tours of capacity k but the algorithm is allowed tours of capacity $k(1 + \epsilon)$.

Extensions and open problems. There are many variations of the CVRP problem. In the most common variation, not only is the vehicle capacity fixed, but the total number of vehicles is also bounded by some number m. This happens in settings where the tours must occur simultaneously. In another variation, the primary objective is to minimize, not the sum of tour lengths, but the maximum tour length (for example, all garbage must be picked up by a certain time.) In yet another variation, each point has a "demand" (less than or equal to k), and the solution must deliver the entire demand using one tour (in other words, split deliveries are not allowed). This models grocery deliveries for example. In more complicated variations, each point has a time window in which it must be visited. In addition, there can be more than one depot, and the tour could be required to perform a mixture of pickups and deliveries. None of those problems have approximation schemes, not even quasi-polynomial time approximation schemes.

In this work we designed the first quasi-polynomial time approximation scheme for the most basic CVRP problem. We hope that our work will stimulate new research for designing approximation schemes for some of the above variants. In particular it is not hard to see that our method can be applied to solve the common variant where the number of vehicles is bounded by m and the capacity constraint is soft: if L is the value of the optimal solution that is constrained to use at most m tours that each pick up at most k points, then our approach can be extended to construct at most m tours, each picking up at most $k(1+\epsilon)$ points, with total length at most $L(1 + \epsilon)$.

Where previous approaches fail. Our approximation scheme uses the divide and conquer approach that Arora used in designing a PTAS for Euclidean TSP [2]. Like Arora, we "divide" the problem using a randomized dissection that recursively partitions the region of input points into progressively smaller boxes. We search for a solution that goes back and forth between adjacent boxes a limited number of times and always through a small number of predetermined sites called *portals* that are placed along the boundary of boxes. It is natural to attempt to extend the TSP structure theorem to show that there exists a near optimal solution that crosses the boundary of boxes a small number of times, and then use dynamic programming. Unfortunately, as noted by Arora [3],

> *"we seem to need a result stating that there is a near-optimum solution which enters or*

leaves each area a small number of times. This does not appear to be true. [...] The difficulty lies in deciding upon a small interface between adjacent boxes, since a large number of tours may cross the edge between them. It seems that the interface has to specify something about each of them, which uses up too many bits."

Indeed, to combine solutions in adjacent boxes it seems necessary to remember the number of points covered by each tour segment and that is too much information to remember.

Overview of our approach. To get around this problem we introduce a new trick which allows us to remember *approximately* how many points are on each tour segment. We design a simple randomized technique that drops points from tour segments. Our technique ensures that the dropped points can be covered at low cost with additional tours, and we simply use a 3-approximation [15] to cover them in the end. (See Figure 2.) Thanks to dropping points, we may assume that the number of points on each tour segment is a power of $(1 + \epsilon/\log n)$, so there are only $O(\log n \log k)$ possibilities. This is a huge saving (when k is $\Omega(\log n)$) compared to the k possibilities that would be required to remember the number of points exactly and it enables us to deal with the difficulties described by Arora: now we have a small interface between adjacent boxes, namely, for every pair of portals and every threshold number of points, we remember the number of tour segments that have this profile. The quasipolynomial running time of our dynamic program (DP) follows as the number of profiles is polylogarithmic and there are at most n tour segments of each profile.

The main technical difficulty consists in showing that the dropped points can be covered at low cost. That cost is split in several components, analyzed separately using a variety of techniques. Let us point out an idea used to analyze one of the components: consider an instance of the problem such that the optimal solution crosses each cut at least δ times. Then OPT has value at least δ times the cost of the minimum spanning tree (see proof of Proposition 5.6.). This lower bound is simple, but new, and crucial in analyzing boxes of a dissection that are visited by many tour segments.

Obstacles for extension to PTAS. Since we describe a tour segment by the pair of portals it uses and the threshold number of points it covers, and there are $O(\log n)$ portals and $O(\log^2 n)$ thresholds we get a polylogarithmic number of profiles. The quasipolynomial running time of our method follows as there can be at most n tour segments of each profile. To reduce the number of tour profiles we seem to require a result show-

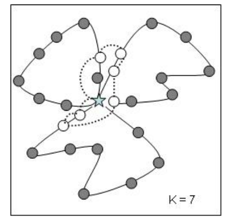

Figure 2: A solution computed by Algorithm 1 for $k = 7$. The star is the depot, the solid circles are the "black" points and the empty circles are the "red" points. The solid tours are computed by the DP in step 2, each covering $\leq k$ "black" points. The dotted tour covers the "red" points and is computed in step 5 using the 3-approximation.

ing the existence of a near optimal solution that uses a small number of portals in each box. We also need to be able to reduce the number of thresholds while still maintaining that each tour covers no more than $(1+\epsilon)k$ points. Additionally a few factors of our quasipolynomial running time comes from accumulating cost over the $O(\log n)$ levels of the randomized dissection tree and a more global accounting of cost could help reduce the running time.

Related Techniques. Our work builds on the approach that Arora [2] used in designing a PTAS for the geometric traveling salesman problem. Similar techniques were also presented by Mitchell [19]. Recently these techniques have been applied to design approximation schemes for several NP-Hard geometric problems, including the polynomial time approximation schemes for Steiner Forest [7], and k-Median [16] and quasipolynomial time schemes for Minimum Weight Triangulation [21] and Minimum Latency problems [4] among others. See [3] for a survey of these techniques.

Outline. Section 2 presents our approximation scheme and the proof of correctness under the assumptions that the DP solution is near optimal (Theorem 2.2, proved in Section 4) and that the 3-approximation solution on the dropped points has length at most $O(\epsilon)$OPT (Theorem 2.4, proved in Section 5). Section 3 presents the DP and Section 6 the derandomization.

2 The Algorithm

Algorithm 1 is an overview of our approximation scheme. A quasipolynomial time DP is used to find a near optimal solution, OPT^{DP}, that includes some tours that cover more than k points. A set of feasible *black* tours is obtained by dropping points from the infeasible tours of OPT^{DP}. The dropped points are chosen carefully using a randomized procedure and are colored red. A 3-approximation is used to construct a solution all red points. The final output is the union of the red and black tours. See Figure 2 for an example.

Algorithm 1 CVRP approximation scheme

Input: n points $\in \mathbb{R}^2$ and integer k

1: Perturb instance, perform random dissection and place portals as described in Section 2.1.
2: Use the DP from Section 3 to find OPT^{DP} which is defined in Subsection 2.2.
3: Trace back in the DP's history to construct *black* tours and assign types to points using the randomized type assignment from Subsection 2.4.
4: Color a point *black* if it has type -1 and *red* otherwise. Drop all red points from the black tours.
5: Use the 3-approximation Algorithm from Subsection 2.3 to get solution on just the red points.

Output: the union of the red tours on the red points and the black tours on the black points.

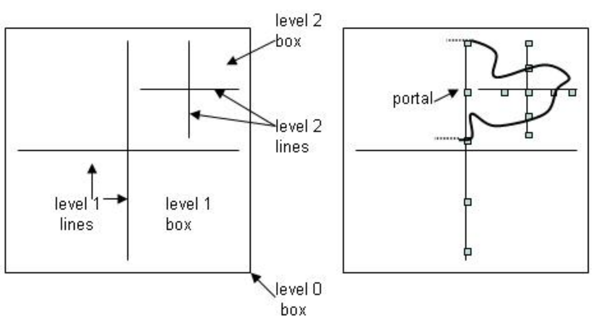

Figure 3: A randomized dissection. The figure on the left shows a dissection with lines and boxes and their levels. The figure on the right shows a portal respecting light tour which crosses the boundaries of boxes only at portals and at most $r = O(1/\epsilon)$ times.

2.1 Preprocessing [2]

Algorithm 1 works on a perturbed instance which we obtain following Arora's approach [2]. A solution for the perturbed instance can be extended to the original instance using additional length $O(\epsilon)$OPT. After the perturbation step we use OPT to denote the optimal CVRP solution of the perturbed instance. We build a randomized dissection of the perturbed instance and place portals along the boundaries of the dissection boxes as done in Arora's work. With the help of Arora's structure theorem 2.1, we can restrict ourselves to look for *portal respecting and light* tours (Definition 2.1). Details are below.

Perturbation. Define a bounding box as the smallest box whose side length L is a power of 2 that contains all input points and the depot. Let d denote the maximum distance between any two input points. Place a grid of granularity $d\epsilon/n$ inside the bounding box. Move every input point to the center of the grid box it lies in. Several points may map to the same grid box center and we will treat these as multiple points which are located at the same location. Finally scale distances by $4n/(\epsilon d)$ so that all coordinates become integral and the minimum non-zero distance is least 4.

Solution to original instance. The solution for the perturbed instance is extended to a solution for the original instance by taking detours from the grid centers to the locations of the points. The total cost of these detours is at most $n \cdot \sqrt{2}d\epsilon/n$. As the two farthest points must be visited from the depot we have that $2d \leq$ OPT. Thus the total cost of detours is $\leq \epsilon$OPT and is negligible compared to OPT. Note also that scaling does not change the structure of the optimal solution. After scaling the maximum distance between points L will be $O(n)$.

Randomized Dissection. A dissection of the bounding box is obtained by recursively partitioning a box into 4 smaller boxes of equal size using one horizontal and one vertical dissection line. The recursion stops when the smallest boxes have size 1×1. The bounding box has level 0, the 4 boxes created by the first dissection have level 1, and since $L = O(n)$ the level of the 1×1 boxes will be $\ell_{\max} = O(\log n)$. The horizontal and vertical dissection lines are also assigned levels. The boundary of the bounding box has level 0, the 2^{i-1} horizontal and 2^{i-1} vertical lines that form level i boxes by partitioning the level $i-1$ boxes are each assigned level i. See Figure 3. A randomized dissection of the bounding box is obtained by randomly choosing integers $a, b \in [0, L)$, and shifting the x coordinates of all horizontal dissection lines by a and all vertical dissection lines by b and reducing modulo L. For example the level 1 horizontal line is moved from $L/2$ to $a + L/2 \mod L$ and the level 1 vertical line is moved to $b + L/2 \mod L$. The dissection is "wrapped around" and wrapped around boxes are treated as one region. The crucial property is that the probability that a line l becomes a level ℓ dissection line in the randomized dissection is

$$Pr(\text{level}(l) = \ell) = 2^\ell/L \tag{2.1}$$

Portals. As in [2], we place points called *portals* on the boundary of dissection boxes that will be the entry and exit points for tours. Let $m = O(\log n/\epsilon)$ and a power of 2. Place $2^\ell m$ portals equidistant apart on each level ℓ dissection line for all $\ell \leq \ell_{max}$. Since a level ℓ line forms the boundary of 2^ℓ level ℓ boxes there will be at most a $4m$ portals along the boundary of any dissection box b. As m and L are powers of 2, portals at lower level boxes will also be portals in higher level boxes.

Definition 2.1. *(Portal respecting and light) A tour is* portal respecting *if it crosses dissection lines only* *at portals. A tour is* light *if it crosses each side of a dissection box at most $r = O(1/\epsilon)$ times.*

See Figure 3. Arora proved there exists a near optimal TSP solution that is portal respecting and light.

THEOREM 2.1. *[2](Arora's Structure Theorem) Let $OPT(TSP)$ denote the optimal solution for an instance of Euclidean TSP and let D be a randomized dissection. With probability $\geq 1/2$ there exists a portal respecting and light tour with respect to D of length $(1 + O(\epsilon))OPT(TSP)$.*

2.2 The Structure Theorem We define $O(\log n \log k)$ *thresholds* in the range $[1, k]$. Instead of remembering the exact number of points on a segment we remember its threshold number. A tour segment is called "rounded" if it covers a threshold number of points. To "round" a tour segment covering x points we find, the largest threshold value $t < x$ and set the *type* of exactly $x - t$ points to indicate that they should be dropped from the segment. As the DP works bottom-up in the dissection tree it rounds tour segments at each level of the tree. To drop a point at level ℓ it sets the type of the point to ℓ. In the end, all points with type between $[0, \ell_{\max}]$ are dropped from the tours.

DEFINITION 2.2. *(Thresholds, types, rounded segments)*

- *Let $\tau = \log_{(1+\epsilon/\log n)}(k\epsilon) + 1/\epsilon$. The sequence of $\tau + 1/\epsilon$* thresholds *are $1, 2, 3, \ldots, 1/\epsilon, t_1 = 1/\epsilon(1 + \epsilon/\log n), \ldots, t_i = 1/\epsilon(1 + \epsilon/\log n)^i, \ldots, t_\tau = k$.*
- *The* type *of a point is an integer in $[-1, \ell_{\max}]$. A point is* active *at level ℓ if its type is strictly less than ℓ.*
- *Let $\Pi = (\pi_i)$ be a set of tours. For any dissection box b at level ℓ, a* segment *is a piece of a tour π_i that enters and exits b at most once. A segment is* rounded *at level ℓ if it covers exactly a threshold, t_i, number of active points otherwise it is called* unrounded. *See Figure 4 for an example of rounded and unrounded segments.*
- *Let $\gamma = \lceil \log^4 n/\epsilon^4 \rceil$. We will always round tour segments in groups of size γ.*

The DP builds tours that may cover more than k points and thus in one sense solves a *relaxed* version of CVRP. To ensure that the DP solution can be made feasible at small cost, tour segments inside a dissection box are only rounded when there are many, at least γ (defined above), segments entering the box in which case

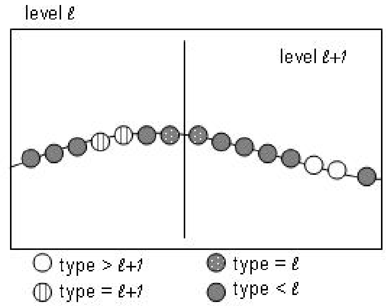

Figure 4: The figure shows boxes at levels $\ell + 1$ and ℓ, and four types of points. The points of type $> \ell + 1$ (white) and type $= \ell + 1$ (stripped) are inactive in all boxes shown. Points with type $= \ell$ (dotted) are active at level $\ell + 1$. Points of type $< \ell$ (solid) are active in all shown boxes. Assume thresholds are $5, 9$. In level ℓ, segment S is rounded as it covers 9 active points. In level $\ell + 1$, segment S is rounded in the left box covering 5 active points and unrounded in the right box covering 6 active points.

the cost of going from the depot to the dropped points in the box can be *charged* to OPT. If a box has less than γ segments, we can afford to remember the exact number of points on each segment. The third part of definition 2.3 limits the number of points that can be dropped from each tour.

DEFINITION 2.3. *(Relaxed CVRP) A* relaxed CVRP *is a set of tours such that there exists an assignment of types to the points with the following properties:*

1. *Each tour covers the depot, at most k points of type $= -1$, and possibly some points of type > -1. The union of the tours covers all n points.*
2. *Each dissection box contains some integer multiple of γ rounded tour segments and at most γ unrounded tour segments.*
3. *Let b be a dissection box and let s be a tour segment in b, which has t active points at level(b). Then segment s has at most $t(1 + \epsilon/\log n)$ active points at level$(b) + 1$.*

The DP will find a *structured* CVRP solution.

DEFINITION 2.4. *(Structured CVRP) Let D be a dissection and let S be CVRP solution consisting of tours $\Pi = (\pi_i)$. S is called* structured *if S is relaxed and each tour π_i is portal respecting and light.*

We extend the objective function to include a penalty for the number of tour crossings:

DEFINITION 2.5. *(Extended Objective Function) Let $\Pi = (\pi_i)$ be a set of tours. For every level ℓ let $c(\pi_i, \ell)$ be the number of times tour π_i crosses the boundary of*

level ℓ boxes, and $d_\ell = L/2^\ell$ denote the length of a level ℓ dissection box. The extended objective function is:
(2.2)
$$F(\Pi) = \sum_i length(\pi_i) + \frac{\epsilon}{\log n} \sum_{level \ell} \sum_i c(\pi_i, \ell) \cdot d_\ell,$$

THEOREM 2.2. *(Structure Theorem) In expectation over shifts of the random dissection, there exists a structured CVRP solution of length $(1+O(\epsilon))OPT$ that minimizes objective function F of Equation 2.2.*

Theorem 2.2 is proved in Section 4. Let OPT^{DP} denote the structured CVRP solution that minimizes the extended objective function F of Equation 2.2. Section 3 describes the DP to compute OPT^{DP} .

2.3 A constant factor approximation [15] To make the structured CVRP solution feasible we drop points from tours containing more than k points and we color dropped points red. A solution for red points is built using Haimovich and Rinnooy Kan's algorithm (Algorithm 2) which partitions a TSP of the red points into tours that cover at most k points [15]. Theorem 2.3 shows the algorithm is a 3-approximation.

THEOREM 2.3. *[15, 5] Let I denote the set of input points, o the depot, and $d(i,o)$ denote the distance of point i from the depot. Define $Rad(I) = \frac{2}{k} \cdot \sum_{i \in I} d(i,o)$ and let $TSP(I \cup o)$ denote the length of the minimal tour of I and o. Then we have,*

- $Rad(I) \leq OPT$
- $TSP(I \cup o) \leq OPT$
- *In expectation the solution of Algorithm 2 has length $Rad(I) + 2 \cdot TSP(I \cup o) \leq 3OPT$.*

Algorithm 2 TSP Partitioning 3-approximation [15]

Input: n points, depot, and integer k
1: Let π denote a tour of input points and the depot obtained using a 2-approximation of TSP.
2: Choose a point p uniformly at random from π.
3: Go around π starting at p, and every time k points are visited, take a detour to the depot.

Output the resulting set of $\lfloor n/k \rfloor + 1$ tours.

2.4 Assigning Types The 3-approximation solution of the red points has small value when Rad and TSP of the red points have small value. To ensure this our type assignment procedure, Algorithm 3, drops points from segment S such that the length of the interval connecting dropped points is small, $O(\epsilon)$length(S), and such that the average distance of the dropped points to the depot is an $O(\epsilon)$ fraction of the average distance of points on segment S to the depot. Figure 5 illustrates Algorithm 3.

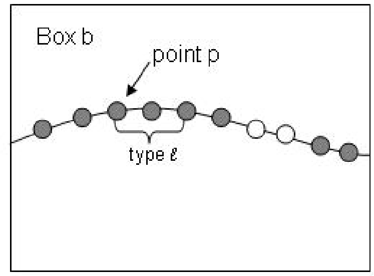

Figure 5: b is a level ℓ box with $|S_a| = 8$ active points (the dark circles) and two inactive points (the white circles). If the closest threshold to 8 is 5, $y = 3$ points are marked to be dropped. Here p and the next two active points are labelled type ℓ.

Algorithm 3 Randomized Type Assignment Procedure

Input: A tour segment S from a level ℓ box b containing active points S_a and requiring y active points to be dropped
1: Select an active point p uniformly from S_a
2: Select p and the $y-1$ consecutive points after p which are all in S_a; if there are less than $y-1$ active points on the segment after p, wrap around and select active points from the other end of the segment.
3: Label each of the y chosen points with type ℓ.

2.5 Proof of Main Theorem 1.1 We use the DP of Lemma 2.1 to compute the structured CVRP solution. Section 3 proves Lemma 2.1.

LEMMA 2.1. *(Dynamic Program) Given the set of input points and a randomly shifted dissection, the dynamic program of Section 3 finds a structured CVRP solution that minimizes the objective F defined in Equation (2.2) in time $n^{\log^{O(1/\epsilon)} n}$.*

The output of Algorithm 1 has length equal to the lengths of the black tours plus the lengths of red tours. The black tours have length at most OPT^{DP} as they are obtained by dropping points from the DP solution. By Theorem 2.2 $\mathrm{OPT}^{D}P$ has length at most $(1+O(\epsilon))$OPT. Theorem 2.4, which is proved in Section 5, shows that the length of the red tours is $O(\epsilon)$OPT.

THEOREM 2.4. *In expectation over the random shifts of the dissection and the random type assignment,* *the length of the red tours output by Algorithm 1 is $O(\epsilon)OPT$.*

Thus the solution output by Algorithm 1 has total length $(1 + O(\epsilon))$OPT. The DP dominates the running time. The derandomization of the Algorithm is discussed in Section 6.

3 The Dynamic Program

The DP table. A *configuration* C of a dissection box b is a list of entries describing the tour segments inside b. A configuration is described by two sublists, one that records information about rounded tour segments and other about unrounded tour segments:

1. Rounded sublist: $(r_1^{p,q}, \dots, r_i^{p,q}, \dots r_\tau^{p,q})$, where $r_i^{p,q}$ is the number of rounded tour segments that use portals p and q and cover exactly t_i active points.
2. Unrounded sublist: $(u_1^{p,q}, \dots, \dots, u_\gamma^{p,q})$, where $u_j^{p,q}$ is the number of active points covered by the j-th unrounded tour segment that uses portals p and q.

The DP has a table entry $L_b[C]$ for each dissection box b and each configuration C of b. $L_b[C]$ is the minimum cost[1] of placing tour segments in b which are compatible with C and are structured as defined by Definition 2.4. OPT^{DP} is the minimum table entry over all configurations of the root level box.

Computing the table entries. Compute table entries in bottom-up order. Inductively, let b be a box at level ℓ and let b_1, b_2, b_3, b_4 be the children of b at level $\ell + 1$. As every tour is structured (and in particular portal respecting and light) , a tour segment in b crosses the boundaries of boxes b_1, b_2, b_3, b_4 inside b, at most $4r$ times, and always through portals. Thus the segment in b is the concatenation of at most $4r + 1$ pieces, where a piece goes from some portal m_i to some portal m_{i+1} in one of the children of b. As the tour must be structured, each piece is either rounded or one of the at most γ unrounded tours inside a child of b. Every piece can be described by a tuple (p, q, x), where p, q are portals and x is either a threshold t_i for some $i < \tau$ or x is a number $j \leq \gamma$ indicating it is the j-th unrounded tour in a child box of b. The *concatenation profile* $\Phi = (p, m_1, n_1), (m_1, m_2, n_2), \dots (m_v, p', n_v)$ of the segment is the list of those $4r+1$ tuples, representing tour segment pieces. Suppose that the concatenation of the pieces described by Φ contains x active points. If b is described as an unrounded box by C then the DP counts this segment as having x active points. Otherwise if b is a rounded box, the DP counts the segment as having t_i active points where t_i is the largest threshold less than x i.e, $t_i \leq x < t_i(1 + \epsilon/\log n)$.

[1] Cost is computed using objective F defined in Equation 2.2.

Let D denote the number of possible concatenation profiles for a segment in box b. For each Φ, let n_Φ denote the number of tour segments in b with concatenation profile Φ. An *interface vector* $I = (n_\Phi)_\Phi$ is a list of D entries. Intuitively, the vector I provides the interface between how tour segments in b are formed by concatenating the segments of b's children.

Let C_0 be a configuration for box b. The calculation of $L_b(C_0)$ is done in a brute force manner by iterating through all possible values of the interface vector I and all possible combinations of configurations in b's children, C_1, C_2, C_3, C_4. A combination $C_0, I, C_1, C_2, C_3, C_4$ is *consistent* if I describes at most γ unrounded segment and if gluing C_1, C_2, C_3, C_4 according to I yields configuration C_0.

The cost of configurations $C_1, \dots C_4$ is stored in lookup tables $L_{b_i}(C_i)$, $1 \leq i \leq 4$. Let $c_b(I)$ be the total number of tour segments in b as specified by I. The value of objective function F, defined in Equation 2.2, of (C_1, C_2, C_3, C_4, I) is $(\epsilon/\log n) \cdot 2c_b(I)$ plus the sum of the costs of C_i for child box b_i. Entry $L_b(C_0)$ stores the cost of the tuple (C_1, C_2, C_3, C_4, I) that is consistent with C_0 and minimizes objective function F.

Running time of dynamic program. How many possible configurations are there for a box b? A configuration of b is a list of $O((\tau + \gamma)\log^2 n)$ entries; there are $O(\log^2 n)$ different pairs of portals (p, q); for each (p, q), there are τ entries in the rounded sublist and γ entries in the unrounded sublist. Each entry of the list (the $r_i^{p,q}$ and $u_j^{p,q}$) is an integer less than n, thus the total number of configurations for box b is $n^{O((\tau+\gamma)\log^2 n)} = n^{O(\log^6 n)}$. As there are $O(n^2)$ dissection boxes, the DP table has size $n^{O(\log^6 n)}$ overall.

How many possible concatenation profiles are there for a segment in box b? Each Φ has a list of $O(r)$ tuples (p, p', x). There are $O(\log^2 n)$ choices of portals p, p' and $\gamma + \tau$ choices of x, so there are $O((\tau + \gamma)\log^2 n) = O(\log^6 n)$ possibilities for each tuple. Thus there are $D = (\log^6 n)^{O(r)}$ possible values of Φ. As $r = O(1/\epsilon)$, $D = \log^{O(1/\epsilon)} n$.

How many possible interfaces are there for a box b? At most n^D, as each n_Φ is an integer less than n. Thus we have a quasi-polynomial number of possibilities for the interface vector I for box b.

Checking for consistency takes time polynomial in the size of the list of entries in I and configurations C_i for $0 \leq i \leq 4$. There are $n^{\log^6 n}$ possible values for each C_i and $n^{\log^{O(1/\epsilon)} n}$ possible values for I. Thus in total it takes time polynomial in $n^{\log^{O(1/\epsilon)} n}$ to run through all combinations of I, C_1, C_2, C_3, C_4 and to compute the lookup table entry at $L_b[C_0]$.

Remark. The DP verifies the existence of a type-assignment satisfying the relaxed CVRP Definition 2.3 but does not actually label points with a specific type. It merely records the number of active points the tour segments it constructs should have. Once the cost of OPT^{DP} solution is found, we can trace through DP solution's history, and find a valid type assignment by looking at the decisions made by the DP. In fact the type assignment can be done while the tours of OPT^{DP} are constructed. If we construct a tour segment with x active points at level ℓ that the DP's history recorded as having t active points, any $x - t$ active points can be chosen from the segment and labeled with type ℓ. Labelling *any* active points on the segment with type l will satisfy the relaxed CVRP definition. But we use the randomized type assignment Algorithm 3 instead to ensure that the labelled points, which will be dropped later, can all be covered with small cost.

4 Proof of the Structure Theorem

Let OPT^L denote the CVRP solution of minimum length consisting of tours that are each portal respecting and light. $F(\mathrm{OPT}^{DP}) \leq (1 + O(\epsilon))\mathrm{OPT}^L$ by Lemma 4.1, where F the extended objective function of Equation 2.2. As $\mathrm{OPT}^{DP} \leq F(\mathrm{OPT}^{DP})$, this immediately implies, $\mathrm{OPT}^{DP} \leq (1+O(\epsilon))\mathrm{OPT}^L$. The structure theorem follows by Corollary 4.1 which shows that OPT^L is near optimal.

LEMMA 4.1. *In expectation over the random shifts of the dissection, $F(OPT^{DP}) \leq (1 + O(\epsilon))OPT^L$.* (Proof given below.)

COROLLARY 4.1. *(Generalization of Arora) In expectation over the random shifts of the dissection, $E[OPT^L] \leq (1 + O(\epsilon))OPT$*

Proof. Let OPT^L consist of the set of tours $\Pi^L = \pi_1, \dots \pi_m$. Apply Arora's structure Theorem 2.1 to each tour, sum, and use linearity of expectation.

4.1 Proof of Lemma 4.1

Proof. To compare OPT^{DP} and OPT^L, we apply Lemma 4.2 to turn OPT^L into a solution that satisfies the relaxed Definition 2.3.

LEMMA 4.2. *Let S be any CVRP solution. There exists a type assignment, such that S becomes a relaxed CVRP solution satisfying Definition 2.3 and the length of S is unchanged.*

The proof of Lemma 4.2 is given below. Using the type assignment of Lemma 4.2, OPT^L turns into a relaxed tour without increasing its cost. OPT^L contains only portal respecting and light tours thus it is now a structured tour so we can compare its cost to OPT^{DP}. Let Π^L denote the set of tours of OPT^L and Π the tours of OPT. As OPT^{DP} minimizes objective function F, we have $F(\mathrm{OPT}^{DP}) \leq F(\mathrm{OPT}^L)$ which is equal to, [2]

$$F(\mathrm{OPT}^L) = \mathrm{OPT}^L + \frac{\epsilon}{\log n} \sum_{\text{level } \ell} c(\Pi^L, \ell) \cdot d_\ell \tag{4.3}$$

Our goal now is to show that the last term of Equation 4.3 is $O(\epsilon)\mathrm{OPT}^L$ in expectation. Lemma 4.3 lets us to bound the number of crossings in Π^L in terms of the number of crossings in Π. Lemma 4.4 lets us to charge each crossing of Π to the length of OPT.

LEMMA 4.3. *For a random dissection at any level ℓ, $E[c(\Pi^L, \ell)] \leq (2 + O(\epsilon))E[c(\Pi, \ell)]$.*

LEMMA 4.4. *In expectation over the random dissection, for any level ℓ, $O(d_\ell)E[(c(\Pi, \ell)] \leq OPT$.*

Using Lemma 4.3 and 4.4 (proofs below) we have that the last term of Equation 4.3 is,

$$\begin{aligned}
&\frac{\epsilon}{\log n} \sum_{\text{level } \ell} E[c(\Pi^L, \ell)] \cdot d_\ell \\
(4.4)\qquad &\leq \frac{\epsilon}{\log n} \sum_{\text{level } \ell} O(2 + \epsilon) \cdot E[c(\Pi, \ell)] \cdot d_\ell \\
(4.5)\qquad &\leq \frac{\epsilon}{\log n} \sum_{\text{level } \ell} \mathrm{OPT} \\
(4.6)\qquad &\leq \frac{\epsilon}{\log n} \cdot \ell_{\max} \cdot \mathrm{OPT} = O(\epsilon)\mathrm{OPT}
\end{aligned}$$

Equation 4.4 follows by Lemma 4.3, Equation 4.5 follows by Lemma 4.4 and Equation 4.6 follows as there are at most $\ell_{\max} = O(\log n)$ levels.

4.2 Proof of Lemma 4.2 We describe a type assignment procedure to prove the Lemma. Initially assign type $= -1$ to all points. Work in a bottom up fashion in the dissection tree from level $\ell_{\max}$ to level 0. At current level ℓ consider each dissection box b one at a time. While box b has more than γ unrounded tour segments, select exactly γ unrounded segments and round the γ segments as a group as follows: Consider each of the γ segments one at a time. If the segment has x active points with $t_i < x < t_{i+1}$, pick any $x - t_i$ of these active points and label them as type ℓ. Perform as many group-rounding steps as necessary until there are at most γ unrounded tours left in box b. Proceed similarly to the other boxes at level ℓ.

[2] To ease notation, $c(\Pi^L, \ell)$ means $\sum_{\pi \in \Pi^L} c(\pi_i^L, \ell)$.

The type assignment procedure does not change any tour in S thus the cost of S is unchanged. Now we verify that the relaxed CVRP Definition 2.3 will be satisfied at the end of the procedure. As S is initially a valid CVRP solution, each tour in S visits the depot and visits at most k other points. The procedure only assigns types ≥ -1. Thus each tour will contain at most k points of type -1 satisfying the first condition of definition 2.3.

Consider a box b at level ℓ. We perform rounding in γ sized groups in b until it has at most γ unrounded segments which implies that b will always contains an integer multiple of γ rounded segments. Thus the second condition holds for b right after the procedure has finished working on level ℓ. On levels $j < \ell$ points are labelled with types $j < \ell$. Thus the number of active points (and hence the number of rounded and unrounded segments) remains the same at level ℓ, and condition two continues to hold in box b while the procedure works on levels $< \ell$.

Consider a segment prior to and after it is rounded at level ℓ. Prior to rounding all points on the segment either have type $= -1$ or a type strictly greater than ℓ, so the segment has the same number of active points, at level ℓ and at level $\ell + 1$. Let x be the number of active points prior to rounding such that $t_i \leq x \leq t_{i+1}$. After rounding the segment has $x - t_i$ points labelled with type ℓ which leaves t_i active points at level ℓ and x active points at level $\ell+1$. As $t_i(1+\epsilon/\log n) = t_{i+1} > x$, the third condition of Definition 2.3 is satisfied.

4.3 Proofs of Lemma 4.3 and 4.4 We list some useful properties required for the proofs of Lemmas 4.3 and 4.4. Let $t(\pi_j, l)$ denote the number of times a tour π_j crosses dissection line l. Arora proved the following useful property relating the length of π_j to $t(\pi_j, l)$.

PROPERTY 4.1. *[2] length*$(\pi_j) \geq \frac{1}{2}\sum_{line\ l} t(\pi_j, l)$

Let $\Pi = (\pi_i)$ be the tours of the optimal CVRP solution. As $\sum_j \pi_j = \text{OPT}$, Arora's Property 4.1 implies that

$$\text{OPT} \geq \frac{1}{2} \sum_{\text{line } l} t(\Pi, l) \tag{4.7}$$

We can write the expected number of crossings on level ℓ boxes in terms $t(\Pi, l)$. We have,

$$E(c(\pi, \ell)) = \sum_{\text{line } l} t(\Pi, l) \cdot \Pr[l \text{ is boundary of level } \ell \text{ box }]$$

The boundaries of level ℓ boxes are formed by lines at levels $\leq \ell$ and by Equation 2.1 the probability that a line is at level $\leq \ell$ is $2^{\ell+1}/L$. Thus for any level ℓ,

$$E(c(\Pi, \ell)) = \frac{2^{\ell+2}}{L} \sum_{\text{line } l} t(\Pi, l) \tag{4.8}$$

Proof. (Proof of Lemma 4.3) To modify the OPT tours, Π, into the OPT^L tours, Π^L, Arora's procedure first does bottom up patching to ensure that the boundary of each dissection box is crossed at most $r = O(1/\epsilon)$ times per tour. Second, it takes detours (along the sides of boxes) to make the tours are portal respecting. Both patching and detouring, may add new crossings to Π^L that are not present in Π.

New crossings from patching. Focus on a box on some level ℓ. Let l be a line such that level$(l) = i$ for $i < \ell$ and let $p_{l,j}$ denote the number segments of l that will be patched at levels $j \geq i$. Each application of patching replaces at least $r-4$ crossings with at most 4 crossings. Thus we have

$$\sum_{j \geq i} p_{l,j} \leq \frac{t(\Pi, l)}{r-3} \tag{4.9}$$

Patching on segments of line l at levels $j < \ell$ introduces $6 \cdot 2^{\ell-j-1}$ crossings at level ℓ. This follows as there are $2^{\ell-j}$ level ℓ boxes along the level j segment of line l for all $j \leq \ell$ and each of Arora's patchings add at most 6 new crossings to each child box along the segment. Thus the number of new crossings introduced at level ℓ from patching on line l is $\sum_{j \geq i} p_{l,j} \cdot 6 \cdot 2^{\ell-j}$. Whether the crossings get added depends on whether level$(l) = i$. Thus the expected number of additional crossings from patching on line l is

$$\begin{aligned}
&E(\text{new crossings from patching on line } l) \\
&\quad = \sum_{\ell > i \geq 1} \Pr[\text{level}(l) = i] \cdot \sum_{\ell > j \geq i} \left(p_{l,j} \cdot 6 \cdot 2^{\ell-j}\right) \\
&\quad \leq 6 \sum_{\ell > j \geq 1} p_{l,j} 2^{\ell-j} \sum_{i \leq j} \Pr[\text{level}(l) = i] \\
(4.10)&\quad = 6 \sum_{\ell > j \geq 1} p_{l,j} 2^{\ell-j} \sum_{i \leq j} \frac{2^i}{L} \\
&\quad = 6 \sum_{\ell > j \geq 1} p_{l,j} 2^{\ell-j} \cdot \frac{2^{j+1}}{L} \\
(4.11)&\quad \leq 6 \frac{t(\Pi, l)}{r-3} \frac{2^{\ell+1}}{L}
\end{aligned}$$

where Equation 4.10 follows by equation 2.1 and Equation 4.11 follows from Equation 4.9. Summing over all dissection lines l, the expected number of additional crossing in Π^L at level ℓ from patching is:

$$\begin{aligned}
E[c(\Pi^L, \ell)] &= \sum_{\text{line } l} 6 \cdot \frac{t(\Pi, l)}{r-3} \cdot \frac{2^{\ell+1}}{L} \\
&\leq 6 \cdot \frac{E[c(\Pi, \ell)]}{r-3} \\
&\leq O(\epsilon)) E[(c(\Pi, \ell)]
\end{aligned}$$

where the second to the last inequality follows by Equation 4.8.

New crossings from detours. A detour on a line at level $j < \ell$ adds at most $(L/2^j \cdot m)/L/2^\ell = 2^{\ell-j})/m$ new crossings to level ℓ boxes, where $m = O(\log n)$. This follows as the length of the detour i.e the distance between portals on a level j line is $L/2^j$ and there are at least $L/2^\ell$ boxes along this detour distance. As the number of detours taken at level j is at most $E[(c(\Pi, j)]$ we have that the number of additional crossings added to level ℓ boxes from detours is

$$\begin{aligned} &\text{additional crossing from detours on } \Pi \\ &= \sum_{j\le\ell}(\#\text{ detours at level } j)\cdot\frac{2^{\ell-j}}{m} \\ &\le \sum_{j\le\ell}E[(c(\Pi,j)]\cdot\frac{2^{\ell-j}}{m} \\ &= \sum_{j\le\ell}E[(c(\Pi,\ell)]\cdot 2^{j-\ell}\cdot\frac{2^{\ell-j}}{m} \qquad (4.12) \\ &\le O(\log n)\frac{E[c(\Pi,\ell)]}{m} = E[c(\Pi,\ell)] \qquad (4.13) \end{aligned}$$

Equation 4.12 follows from two implications of Equation 4.8: $E[c(\Pi, \ell)] = 2^{\ell+2}/L \sum_{\text{line } l} t(\Pi, l)$ and $E[c(\Pi, j)] = 2^{j+2}/L \sum_{\text{line } l} t(\Pi, l)$. Equation 4.13 follows as there are at most $O(\log n)$ levels, so $O(\log n)$ possible values of $j \le \ell$.

Total crossings. The expected number of crossings on the light tours at level ℓ is at most the original crossing at level ℓ plus the additional crossings added from patching and detours.

$$\begin{aligned} E[c(\Pi^L, \ell) &\le E[c(\Pi,\ell)] + O(\epsilon)E[c(\Pi,\ell)] + E[c(\Pi,\ell)] \\ &= (2 + O(\epsilon))E[c(\Pi,\ell)] \end{aligned}$$

Proof. (Proof of Lemma 4.4) Combine Equations 4.7 and 4.8 to get $E(c(\Pi, \ell)) \le \frac{2^{\ell+3}}{L}\text{OPT}$. The Lemma follows from the fact that a level ℓ box has side length $d_\ell = L/2^\ell$.

5 Proof of Theorem 2.4

Let R denote the points marked red by Algorithm 1. By Theorem 2.3 the 3-approximation on R has cost at most $\text{Rad}(R) + 2\text{TSP}(R \cup o)$. Lemmas 5.1 and 5.2 show in expectation both quantities are $O(\epsilon)\text{OPT}$, proving Theorem 2.4.

LEMMA 5.1. *In expectation over the random type assignment, $Rad(R) = O(\epsilon)OPT$*

LEMMA 5.2. *In expectation over the random dissection and type assignment $TSP(R \cup o) = O(\epsilon)OPT$*

We begin by listing some properties that will be useful in proving Lemmas 5.1, and 5.2.

Let b be a level ℓ box containing points of type ℓ and consider a segment S which is rounded in b by Algorithm 1. Let S_a denote the set of active points on S prior to its rounding and let R be the interval of points labelled type ℓ in S. Let $|R|$ denote the number of points labelled type ℓ and $|S_a|$ the size of S_a.

PROPERTY 5.1. $|R| \le |S_a| \cdot O(\epsilon/\log n)$.

Proof. In the DP's history S has a concatenation profile Φ with its rounded flag set to true as S is a rounded segment. Suppose S has x active points once it is concatenated according to Φ. The DP counts S as a rounded segment having exactly t_i active points for the unique threshold value t_i lying in the interval $[x/(1 + \epsilon/\log n), x]$. To get exactly t_i active points on S at most $x - t_i \le x(\epsilon/\log n)$ active points are set to type ℓ. Thus $|R| \le x\epsilon/\log n$ while $|S_a| = x$.

DEFINITION 5.1. *(Length of interval) Denote the points in R as $r_1, r_2, \ldots r_d$ and the points in S_a as $s_1, s_2, s_3, \ldots s_x$. Let b_1, b_2 be the points on the boundary of b where S enters and exits b. Thus S visits s_1 after entering at b_1 and it visits s_x before exiting from b_2. If R does not contain both s_1 and s_x then length$(R) = \sum_i^d d(s_i, s_{i+1})$, where $d(u, v)$ is the distance between points u and v. Otherwise let $r_e = s_x$, then $r_{e+1} = s_1$ as Algorithm 3 wraps around and length$(R) = \sum_{i=1}^{e-1} d(s_i, s_{i+1}) + d(s_x, b_2) + d(b_1, s_1) + \sum_{i=e+1}^{d} d(r_i, r_{i+1})$.*

PROPERTY 5.2. *$E[length(R)] \le length(S_a) \cdot O(\epsilon/\log n)$*

Proof. Let b_1 and b_2 be the points where S enters and exits box b. Define $z_x = d(b_1, s_1) + d(s_x, b_2)$ and $z_i = d(s_i, s_{i+1})$, for $1 \le i < x$. The length of S_a is $\sum_{i=1}^{x} z_i$ and $E[\text{length}(R)] = \sum_{i=1}^{x} z_i \Pr[z_i \text{ is counted in } R]$. For all i the probability that z_i is counted is the probability that s_i and its consecutive point[3], s_{i+1}, are both included in R. A point $s \in S_a$ belongs to exactly $|R|$ intervals and the consecutive point of s appears in $|R| - 1$ of these intervals. Thus the $\Pr[z_i \text{ is counted }] = (|R|-1)/(|S_a|)$. Applying Property 5.1, $E[\text{length}(R)] = \sum_{i=1}^{x} z_i(|R| - 1)/(|S_a|) = O(\epsilon/\log n)\sum_{i=1}^{x} z_i$, which proves the property as $\sum_{i=1}^{x} z_i = \text{length}(S_a)$.

PROPERTY 5.3. *A point $s \in S_a$ is in R with probability $|R|/|S_a|$.*

Proof. Each point $s \in S_a$ belongs to $|R|$ intervals as each interval consists of $|R|$ consecutive points. There

[3] The consecutive point of s_x is s_1

are a total of $|S_a|$ different intervals, each starting at a different point in S_a and Algorithm 3 picks uniformly among them.

5.1 Proof of Lemma 5.1 Recall that, $\text{Rad}(R) = 2/k \sum_{x\in R} d(x,o)$, where $d(x,o)$ is the distance of point x from the depot. By Theorem 2.3 $\text{Rad}(I) \le \text{OPT}$, so it is sufficient to show that $\text{Rad}(R) \le O(\epsilon)\text{Rad}(I)$. Fix any level ℓ of the dissection and let R_ℓ be the set of points which were assigned type ℓ. We show that in expectation $\text{Rad}(R_\ell) \le O(\epsilon/\log n)\text{Rad}(I)$. The proposition follows by linearity of expectation (over all levels) since $\text{Rad}(R) = \sum_{\text{level } l} \text{Rad}(R_\ell)$.

Partition R_ℓ according to the tour segment it is from: $R_\ell^1 \subset S_1, R_\ell^2 \subset S_2, \ldots R_\ell^m \subset S_m$ where R_ℓ^j is the set of red points from tour segment S_j. By definition we have that

$$\text{Rad}(I) \ge \frac{2}{k}\sum_{j=1}^{m}\sum_{x\in S_j} d(o,x) \tag{5.14}$$

As R_ℓ is picked randomly, and by Properties 5.3 and 5.1, $\Pr[x \in R_\ell^j] \le O(\epsilon/\log n)$, so we get

$$\begin{aligned} E[\text{Rad}(R_\ell)] &= \frac{2}{k}\sum_{j=1}^{m}\sum_{x\in S_j} d(o,x)\Pr[x\in R_\ell^j] \\ &\le \frac{2}{k}\sum_{j}^{m}\sum_{x\in S_j} d(o,x)\cdot O(\epsilon/\log n) \end{aligned}$$

Combining this with Equation 5.14 we get $E[\text{Rad}(R_\ell)] \le O(\epsilon/\log n)\text{Rad}(I)$.

5.2 Proof of Lemma 5.2

Proof. Let R_ℓ be the points labeled type ℓ at level ℓ. We show $E[\text{TSP}(R_\ell \cup \{o\})] \le O(\epsilon/\log n)\text{OPT}$ which implies Lemma 5.2 as the tours of $\{R_\ell \cup o\}$ from all levels can be pasted together at the depot to yield a tour of $(R \cup o)$.

Let B_ℓ be the boxes at level ℓ containing points of R_ℓ. Consider the cost of $\text{TSP}(R_\ell \cup o)$ in two parts: the outside part, which is the cost to reach the boxes of B_ℓ from the depot, and the inside costs, which is the cost of visiting the red points inside the boxes of B_ℓ. To analyze the outside cost, let C be a set of points containing at least one portal from each box of B_ℓ such that the minimum spanning tree, $MST(C \cup o)$ is minimized[4]. The optimal tour of $R_\ell \cup \{o\}$ is at most,

$$TSP(R_\ell \cup o) \le 2MST(C\cup o) + \sum_{b\in B_\ell} \text{inside cost of } b \tag{5.15}$$

[4] C is used only for the analysis and does not need to be found explicitly.

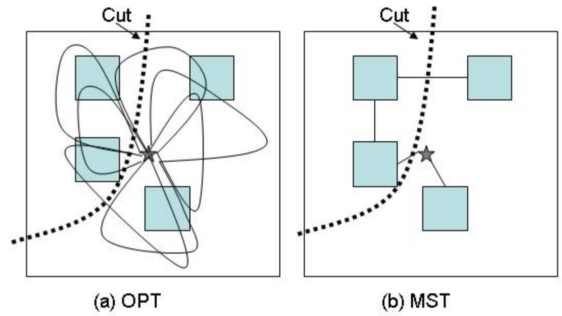

Figure 6: The shaded boxes are the boxes of B_ℓ. (a) Given that OPT has at least 3 tours entering each box, OPT crosses all non-trivial cuts at least 6 times. This is made explicit in Equation 5.16. (b) The MST crosses all non-trivial cuts at least once as expressed in Equation 5.17

Proposition 5.1 shows that the first term of 5.15 is $O(\epsilon/\log n)$OPT Proposition and 5.2 shows the same for the second term, which proves the Lemma.

Proposition 5.1. *In expectation over the random shifted dissection, $E[2MST(C\cup o)] \le O(\epsilon/\log n)OPT$.*

Proof. We show that, $MST(C \cup o) \le O(\epsilon/\log n)\text{OPT}^{DP}$ which implies the proposition as OPT^{DP} is at most $(1+O(\epsilon))$OPT by the structure Theorem 2.2.

Consider the fully connected graph G with one vertex for each point in C and one more for the depot. Define the weight of an edge of G to be the distance between the two vertices connected by that edge. Consider the following linear program 5.16 with value v on G.

$$v = \min_x (w_e \cdot x_e) \text{ s.t. } \begin{cases} \sum_{e\in\delta(S)} x_e \ge \gamma/4r & \forall S \subset V \\ & S \ne \emptyset \\ & S \ne V \\ x_e \ge 0 & \end{cases} \tag{5.16}$$

As each $b \in B_\ell$ contains points labelled ℓ (i.e at least γ rounded segments), OPT^{DP} contains at least γ tour segments crossing into b. See Figure 6. As each tour in OPT^{DP} is structured (and in particular light), there are at least $\gamma/4r$ tours entering b. Thus OPT^{DP} has at least $\gamma/4r$ edges crossing any cut separating the depot from a point in C. As v is the minimum cost way to have at least $\gamma/4r$ edges cross all such cuts, $\text{OPT}^{DP} \ge v$.

Now consider the linear program 5.17 which is the

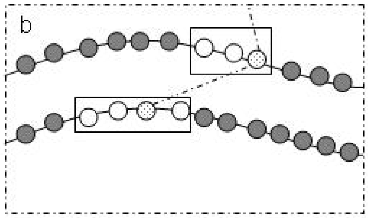

Figure 7: The figure shows a box $b \in B_\ell$. The white points have type ℓ and were dropped. Proposition 5.3 shows that total length of the white intervals (boxed segments) is small. Proposition 5.4 shows that the cost of connecting the white intervals to the boundary is small (dashed lines).

IP relaxation of MST. Let v' be its value on G.
(5.17)

$$v' = \min_x (w_e \cdot x_e) \quad \text{s.t.} \quad \begin{cases} \sum_{e \in \delta(S)} x_e \geq 1 & \forall S \subset V \\ & S \neq \emptyset \\ & S \neq V \\ x_e \geq 0 \end{cases}$$

See Figure 6. Observe that for any solution x of 5.16, $x' = x \cdot 4r/\gamma$ is a solution for 5.17. As both linear programs have the same objective, $v \cdot 4r/\gamma = v'$. The MST relaxation 5.17 is known to have integrality gap at most 2 [23], so that $v' \geq \frac{1}{2} \cdot$ MST $(C \cup o)$. Thus we have that

$$\text{OPT}^{DP} \geq v = v' \cdot (\gamma/4r) \geq MST(C \cup o) \cdot (\gamma/8r)$$

Thus $(8r/\gamma) \cdot \text{OPT}^{DP} \geq MST(C \cup o)$ and as $8r/\gamma = o(\epsilon/\log n)$, the Proposition is proved.

Now we analyze the inside cost.

Proposition 5.2. *In expectation over the random dissection and the random type assignment the total inside cost at level ℓ is at most $O(\epsilon/\log n)OPT$.*

Proof. The inside cost at level ℓ is the sum of the inside costs of each box $b \in B_\ell$. The contribution of box $b \in B_\ell$, is the sum the length of the intervals of type ℓ points inside b plus the cost of connecting these intervals to the boundary of b. Proposition 5.3 shows that in expectation over the random type assignment the sum of the length of intervals of type ℓ points over all boxes in B_ℓ is $O(\epsilon/\log n)\text{OPT}^{DP}$.

The type ℓ intervals inside each $b \in B_\ell$ must be connected to each other and to the boundary of their box. We refer to this as the total connection cost at level ℓ and denote it as $CC(\ell)$. $CC(\ell)$ is the sum of the length of the boundaries of each box $b \in B_\ell$ plus the cost of connecting the type ℓ intervals inside each $b \in B_\ell$ to the boundary of b. Proposition 5.4 shows that $CC(\ell) = O(\epsilon/\log n)F(\text{OPT}^{DP})$. See Figure 7. This proves the proposition as Lemma 4.1 and Corollary 4.1 imply that $F(\text{OPT}^{DP}) \leq (1 + O(\epsilon))\text{OPT}$.

Proposition 5.3. *In expectation over the random type assignment the length of all intervals of type ℓ is $O(\epsilon/\log n)OPT^{DP}$.*

Proof. The main idea is to sum the lengths of intervals of type ℓ over all boxes in B_ℓ boxes, use Property 5.2 and linearity.

Consider a box $b \in B_\ell$ and denote the set of its type ℓ points as R_b . Partition the points in R_b according to the segments of b they come from: $r_1 \subset s_1, r_2 \subset s_2, \ldots r_m \subset s_m$ such that r_j is the set of points labelled type ℓ from tour segment s_j. By Property 5.2, in expectation over the random type-assignment the length of r_j is at most $O(\epsilon/\log n)$ times the length of s_j. By linearity, $\sum_j^m E[\text{length}(r_j)] \leq O(\epsilon/\log n)\sum_{j=1}^m \text{length}(s_j)$. Let OPT_b^{DP} denote the projection of OPT^{DP} inside box b. $\text{OPT}_b^{DP} \geq \sum_{j=1}^m \text{length}(s_j)$, and OPT^{DP} is at least the sum of OPT_b^{DP} over all boxes $b \in B_\ell$. This implies that the total length of red intervals at level ℓ is at most $O(\epsilon/\log n)\text{OPT}^{DP}$.

Proposition 5.4. *The total connection cost for level ℓ is $O(\epsilon/\log n)F(OPT^{DP})$.*

Proof. Let $CC(\ell)$ denote the total connection cost at level ℓ, which is the cost of connecting the type ℓ intervals inside each $b \in B_\ell$ to the boundary of its box. Focus on one box $b \in B_\ell$. To simplify the analysis we add in the length of the boundary of b which is $4d_\ell$ where d_ℓ is side length of a level ℓ box. As b is a rounded box it has at least γ rounded tour segments. Partition the segments of b into groups of size γ. This yields g_b groups: each containing exactly γ rounded segments. Consider any group and let R' be a set containing one type ℓ point from each of the segments in the group. As we have already added the entire boundary of b to the connection cost, the additional cost to connect all the type ℓ intervals in the group to the boundary of the box is at most $MST(R') + d_\ell/2$. We bound $MST(R')$ using the following bound for TSP [15][5]. (See [15] for a proof).

Theorem 5.1. *[15][5] Let U be a set of points in two dimensional Euclidean space. Let d_{max} be the max distance between any two points of U. Then $TSP(U) = O(d_{max}\sqrt{|U|})$*

In our context, $d_{max} = d_\ell$, and $U = R'$. Since $|R'| = \gamma$, $|U| = \gamma$. By Theorem 5.1 we have that $MST(R') + d_\ell = O(d_\ell \cdot \sqrt{\gamma})$. This holds for each of the g_b groups of rounded segments thus we have that the total contribution of box b to the connection cost is $g_b \cdot O(d_\ell \cdot \sqrt{\gamma}) + 4d_\ell = O(g_b \cdot d_\ell \cdot \sqrt{\gamma})$. Summing up the contribution of each box $b \in B_l$ we have that the total connection cost at level ℓ is,

$$(5.18)\quad CC(\ell) = \sum_{b \in B_\ell} O(g_b \cdot d_\ell \cdot \sqrt{\gamma}) = O(d_\ell \cdot \sqrt{\gamma}) \sum_{b \in B_\ell} g_b$$

Each rounded tour segment in level ℓ has two crossings with the boundary of a level ℓ dissection box, thus: $c(\Pi^{DP}, \ell) \geq 2\gamma \sum_{b \in b_l} g_b$, where Π^{DP} are the tours of OPT^{DP}. Using Equation 5.18,

$$(5.19)\qquad O\left(d_\ell/\sqrt{\gamma}\right) \cdot c(\Pi^{DP}, \ell) \geq CC(\ell)$$

For objective function F, (defined in 2.2), we have $(\log n/\epsilon) \cdot F(\text{OPT}^{DP}) \geq c(\Pi^{DP}, \ell) d_\ell$. Substituting it for $c(\Pi^{DP}, \ell) d_\ell$ in Equation 5.19 we get, $O(1/\sqrt{\gamma})(\log n/\epsilon) \cdot F(\text{OPT}^{DP}) \geq CC(\ell)$. As $\gamma = \log^4 n/\epsilon^4$ this reduces to $O(\epsilon/\log n) \cdot F(\text{OPT}^{DP}) \geq CC(\ell)$, which proves this Proposition.

6 Derandomization

Arora's dissection can be derandomized by trying all choices for the shifts a and b. More efficient derandomizations are given in Czumaj and Lingas and in Rao and Smith [9, 20]. As for the randomized type assignment Algorithm 3, to guarantee that the cost of the dropped points is small, when selecting an interval Y to drop from a segment S, we only need to pick Y such that (1) $\text{Rad}(Y) \leq O(\epsilon/\log n)\text{Rad}(S)$ and (2)$\text{length}(Y) \leq O(\epsilon/\log n)\text{length}(S)$. In Lemma 5.1 and Property 5.2 we prove that these two conditions hold at the same time, in expectation when Y is chosen by first selecting a point uniformly from S and then selecting the next $|Y| - 1$ consecutive points. To derandomize we can test the at most $|S|$ intervals of length $|Y|$ in S, (each starting from a different point in S), and select any interval that satisfies these two conditions.

7 Acknowledgments

We like to thank Shay Mozes his close reading of our paper and for pointing out corrections.

References

[1] A. Adamaszek, A. Czumaj, and A. Lingas. PTAS for k-tour cover problem on the plane for moderately large values of k. CoRR. 2009. http://arxiv.org/abs/0904.2576

[2] S. Arora. Polynomial-time approximation schemes for Euclidean TSP and other geometric problems. JACM, 45(5):753-782, 1998.

[3] S. Arora. Approximation schemes for NP-hard geometric optimization problems: A survey. Mathematical Programming, 97 (1,2) July 2003.

[4] S. Arora and G. Karakostas. Approximation schemes for minimum latency problems. Proceedings of the thirty-first annual ACM symposium on Theory of computing. pages 688–693. 1999.

[5] T. Asano, N. Kathoh, H. Tamaki, T. Tokuyama. Covering Points in the plane by CVRP: towards a polynomial time approximation scheme for general k. STOC 1997.

[6] T. Asano, N. Katoh, H. Tamaki, and T. Tokuyama. Covering points in the plane by k-tours: a polynomial approximation scheme for fixed k. IBM Tokyo Research Laboratory Research Report RT0162, 1996.

[7] G. Borradaile, P. Klein, and C. Mathieu. A polynomial time approximation scheme for Euclidean Steiner forest. Proceedings of the forty-ninth annual IEEE Symposium on Foundations of Computer Science, pp. 115-124. 2008.

[8] G. Clarke, and J.V.Wright. Scheduling of vehicles from a central depot to a number of delivery points, Operations Research 12 (1964), pages 568581.

[9] A. Czumaj and A. Lingas. A polynomial time approximation scheme for Euclidean minimum cost k-connectivity. Proceedings of 25th Annual International Cololoquium on Automata, Languages and Programming, LNCS, Springer Verlag 1998.

[10] Dantzig, G.B.; Ramser, J.H. The Truck Dispatching Problem. Management Science 6 (1): 80-91. (1959).

[11] B.E. Gillett, and L.R. Miller. A heuristic algorithm for the vehicle dispatch problem, Operations Research 22 (1974), pages 340349.

[12] M.L. Fischer. Vehicle Routing. in Network Routing, Handbooks in Operations Research and Management Science, 8, Ball, M. O., T. L. Magnanti, C. L. Monma and G. L. Nemhauser (Eds.), Elsevier Science, Amsterdam, 1-33, 1995.

[13] M.L. Fisher, and R. Jaikumar. A generalized assignment heuristic for the vehicle routing problem, Networks 11 (1981), 109124.

[14] Golden, Bruce; Raghavan, S.; Wasil, Edward (Eds.) The Vehicle Routing Problem: Latest Advances and New Challenges. Operations Research/Computer Science Interfaces Series, Vol. 43. 2008.

[15] M. Haimovich and A.H.G Rinnooy Kan. Bounds and heuristic for capacitated routing problems. Mathematics of Operations Research, 10(4), 527-542, 1985.

[16] S. Kolliopoulos and S. Rao. A nearly linear-time approximation scheme for the Euclidean k-media problem. SIAM J. Comput., 37(3):757782, 2007.

[17] G. Laporte. What you should know about the vehicle routing problem. Naval Research Logistics, vol 54, issue 8. pages 811-819. 2007.

[18] R.C. Larson and A.R. Odoni. Urban Operations Research. Prentice Hall, NJ. 1981.

[19] J. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial- time approxi-

mation scheme for geometric TSP, k-MST, and related problems. SIAM J. Comput., 28(4):12981309, 1999.

[20] S. Rao and W. Smith. Approximating geometrical graphs via "spanners" and "banyans". In 30th STOC, pages 540-550, 1998.

[21] J. Remy and A. Steger. A quasi-polynomial time approximation scheme for minimum weight triangulation. Proceedings of the 38th ACM Symposium on Theory of Computing, pages 316–325. 2006.

[22] Paolo Toth, Daniele Vigo. The vehicle routing problem. Society for Industrial and Applied Mathematics. Philadelphia, PA. 2001.

[23] Approximation Algorithms. Vijay Vazirani. Chapter 20. Berlin: Springer. 2003.

[24] http://neo.lcc.uma.es/radi-aeb/WebVRP/