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

Aparna Das Claire Mathieu

October 25, 2018

#### 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

In 1959, Dantzig and Ramser introduced the vehicle routing problem (VRP) and gave a linear programming based Algorithm whose "calculations may be readily performed by hand or automatic digital computing machine"[8]. Since its introduction, VRP has been widely studied by researchers in Operations Research and Computer Science. Several books (see [16], [10] and [9], among others) have been written on the VRP and it even has its own wikipedia page [19].

VRP is used 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. When Dantzig and Ramser first introduced VRP they stated that "no practical applications have been made as yet"[8]. Since that time, applications of various VRP problems have been identified in numerous industries where transportation costs matter such as food and beverage distribution, and package and newspaper delivery. Toth and Vigo report on businesses that saved between 5 % and 20% of total costs by solving VRP problems using computerized models [16].

The capacitated vehicle routing problem. We study the most basic form of the vehicle routing problem, the capacitated version (CVRP), where the input is n + 1 points representing the locations of n customers and one depot, and a vehicle of capacity k. The objective is to find a collection of tours, starting at the depot and visiting at most k customers, whose union covers all n customers, such that the sum of the lengths of the tours is minimized. CVRP is also referred to as the k-tours problem in the Computer Science literature [1, 4]. We focus on the 2*d*-Euclidean version of CVRP where customers and the depot are on the Euclidean plane.

**Previous work.** Several results are already known about the approximability of CVRP. When the capacity of the vehicle k is 2, it can be solved in polynomial time using minimum weight matching. The metric case was shown to be APX-complete for all  $3 \le k$ ; 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 [5]. Constant factor approximation with performance  $(1 + \alpha)$  with  $\alpha$  being the best constant factor approximation for TSP, were presented by Haimovich and Rinnooy Kan [11].

The existence of a PTAS for the 2*d*-Euclidean version remains an active area of research. Extending [11], Asano et al. presented a PTAS for the case  $k = O(\log n / \log \log n)$  [4]. Arora's work implies a PTAS for  $k = \Omega(n)$  [4].

**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 2d-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.

Where previous approaches fail. Our approximation scheme uses the divide and conquer approach that Arora used in designing a PTAS for Euclidean TSP [1]. Like Arora, we "divide" the problem using a randomized dissection that recursively partitions the region of input points into progressively smaller squares. We search for a solution that goes back and forth between adjacent squares a limited number of times and always through a small number of predetermined sites called *portals* that are placed along the boundary of squares. It is natural to attempt to extend the TSP structure theorem and show that there exists a near optimal solution that crosses the boundary of squares a small number of times, and then use dynamic programming. Unfortunately [2],

"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 squares, 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 squares it seems necessary to remember the number of points covered by tour segments, and that is is too much information to remember.

**Overview of our approach.** To get around this problem we remember *approximately* how many points are on each tour segment. Specifically we round the number of points on each tour segment down to a multiple  $(1 + \epsilon/\log n)$ , which means there are only  $O(\log k)$  possibilities for the approximate number of points on each tour segment. This enables us to deal with the difficulties described by Arora: now we have a small interface between adjacent squares, namely, for every pair of portals and every approximate number of points, we remember the number of tour segments that have this profile. The quasipolynomial running time of our dynamic program (DP) follows since the number of profiles is polylogarithmic and as there are at most n tour segments of each profile.

One challenge with only remembering the approximate number of points per tour segment is that the tours found by our dynamic program (DP) could now cover a little bit more than k points. To deal with this issue, whenever a tour of the DP solution covers more than k points we carefully choose enough points and drop them from the tour to make it feasible. Finally we compute a solution just on the dropped points using a 3-approximation [11] and prove that it has negligible length compared to OPT.

In section 2 we present our approximation scheme and prove its 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 2.5, proved in section 5). Section 3 presents the DP and section 6 the derandomization.

**Related Work.** Our work builds on the approach that Arora [1] used in designing a PTAS for the geometric traveling salesman problem. Similar techniques were also presented by Mitchell [13]. 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 [6], and K-Median [12] and quasipolynomial time schemes for Minimum Weight Triangulation [15] and Minimum Latency problems [3] among others. See [2] for a survey of these techniques and a discussion about generalizing them to other problems.

# 2 The Algorithm

An overview of our approximation scheme is given in Algorithm 1. We use a quasipolynomial time DP to find a near optimal solution,  $OPT^{DP}$ , which may include some tours that cover more than k points. We drop points from each infeasible tour of  $OPT^{DP}$ , choosing the points carefully using a randomized procedure to obtain the set of feasible *black* tours. A solution on the dropped points, the *red* tours, is obtained using a 3-approximation. We output the union of the red and black tours.

### Algorithm 1 CVRP approximation scheme

Input: *n* points in  $\mathbb{R}^2$  and an 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 tours and assign types to points using the randomized type assignment from subsection 2.4.
- 4: A point is *black* if has type -1 and *red* otherwise. Drop all red points from the  $OPT^{DP}$  tours.
- 5: Get a solution for the red points using the 3-approximation Algorithm from subsection 2.3.

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

### 2.1 Preprocessing [1]

**Perturbation.** Algorithm 1 will work on a perturbed instance. Following Arora's approach from [1], we perturb and scale so that all points fit into a square bounding box of side length L = O(n) where L is a power of 2, and so that all points have integer coordinates and the distance between any two points is either 0 or at least 4. A solution for the perturbed instance can be extended to the original instance using additional length  $O(\epsilon)$ OPT. See appendix A for the details. From this point, OPT will denote the optimal CVRP solution to the perturbed instance.

**Randomized Dissection.** Using Arora's divide and conquer paradigm from [1], we perform a randomized dissection of the input by recursively partitioning the bounding square into 4 smaller squares of equal size using one horizontal and one vertical dissection line. The recursion stops when the smallest squares have size  $1 \times 1$ . We define levels for the squares and the dissection lines. See appendix A for the details. The maximum level of any square or line is  $\ell_{\max} = O(\log n)$ . The probability that a line lbecomes a level  $\ell$  dissection line in the randomized dissection is

$$Pr(\operatorname{level}(l) = \ell) = 2^{\ell}/L \tag{1}$$

**Portals.** As in [1], we place points called *portals* on the boundary of dissection squares 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  $\ell$  dissection line for all  $\ell \leq \ell_{max}$ . Since a level  $\ell$  line forms the boundary of  $2^{\ell}$  level  $\ell$  squares there will be at most at 4m portals along the boundary of any dissection square b. As m and L are powers of 2, portals at lower level squares will also be portals in higher level squares.

**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 is portal respecting and crosses each side of a dissection square at most  $r = O(1/\epsilon)$  times.

Arora showed that for TSP there exists a near optimal solution which is light.

**Theorem 2.1.** [1] Let OPT(TSP) denote the optimal solution for an instance of Euclidean TSP and let D be its randomized dissection. With probability at least 1/2 there exists a salesman tour of length  $(1 + O(\epsilon))OPT$  that is light with respect to dissection D.

#### 2.2 The Structure Theorem

We formally define the solution computed by our DP. Recall that remembering the exact number of points on tour segments may require too many bits. Instead, we define  $O(\log k)$  thresholds that are multiples of  $(1 + \epsilon/\log n)$  in the range [1, k] and round down the number of points on each segment to its closest threshold. Given a tour segment covering a non-threshold number of points x with t being the closest threshold value less than x, to "round" the segment we set the type of exactly x - t points to indicate they should be dropped from the segment. When two tour segments each covering a threshold number of points are concatenated, the new segment may have to be "rounded" again as the sum of two thresholds do not necessarily add to another threshold. The DP works bottom-up in the dissection tree

and may have to "round" tour segments at each level of the tree. It will mark the points that need to be dropped at level  $\ell$  by setting their type to  $\ell$ . Points with a type in  $[0, \ell_{max}]$  are dropped from the final tours.

**Definition 2.2.** (Thresholds, Types, and Rounded Segments)

- Let  $\tau = \log_{(1+\epsilon/\log n)}(k/\epsilon)$ . We define a sequence of  $\tau + 1$  thresholds :  $t_0 = 1, t_1 = 1/\epsilon, \ldots, t_i = 1/\epsilon(1+\epsilon/\log n)^{i-1}, \ldots, t_{\tau} = k$ .
- Assume that each point has a type which is an integer in  $[-1, \ell_{max}]$ . We say that a point is active at level  $\ell$  if its type is strictly less than  $\ell$ .
- Let  $\pi = (\pi_i)$  be a set of tours. For any  $\pi_i$  and any dissection square b at level  $\ell$ , a segment is a connected component of  $\pi_i \cap b$ . A segment is either rounded or unrounded, with the following property: for each rounded segment s, there is a threshold  $t_i$  such that s covers exactly  $t_i$  active points.

The DP builds tours are allowed to cover more than k points and thus in one sense the DP solves a *relaxed* version of CVRP. To ensure that the additional cost of making the DP solution feasible small compared to OPT, we only round tour segments inside a dissection square when there are many, at least  $\gamma$  (defined below), segments entering the square in which case the cost of going from the depot to the dropped points in the square can be *charged* to OPT. If there are only a few, less than  $\gamma$  per threshold, tour segments entering the square, we can afford to remember the exact number of points per segment. The intuition behind the third part of definition 2.3 is to limit the number of points that are marked to 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. Let  $\gamma = \log^9 n/\epsilon^4$ . In any dissection square b for each threshold value  $t_i$  ( $i \leq \tau 1$ ) there are at most  $\gamma$  unrounded segments covering between  $[t_i, t_{i+1})$  points. The number of rounded segments in b covering exactly  $t_i$  points that are active at level(b) is an integer multiple of  $\gamma$ .
- 3. Let b be a dissection square and let s be a tour segment in b, which has t points that are active at level(b). Then segment s has at most  $t(1 + \epsilon/\log n)$  active points at level(b) + 1.

We apply the concept of being light to each tour of the solution individually rather than to the solution as a whole.

**Definition 2.4.** Let D be a dissection and let S be a CVRP solution consisting of tours  $(\pi_i)$ . S is called i-light if each tour  $\pi_i$  is light.

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

**Definition 2.5.** (Extended Objective Function) Let  $\pi = (\pi_i)$  be a set of tours. For every level  $\ell$  let  $c(\pi_i, \ell)$  be the number of times tour  $\pi_i$  crosses the boundary of level  $\ell$  squares, and  $d_{\ell} = L/2^{\ell}$  be the length of a level  $\ell$  square in the dissection. The extended objective function is:

$$F(\pi) = \sum_{i} length(\pi_i) + \frac{\epsilon}{\log^2 n} \sum_{level\ell} \sum_{i} c(\pi_i, \ell) \cdot d_\ell,$$
(2)

**Theorem 2.2.** (Structure Theorem) In expectation over the shifts of the random dissection, the *i*-light and relaxed CVRP solution which minimizes the extended objective function F given in equation 2 has length  $(1 + O(\epsilon))OPT$ .

We prove Theorem 2.2 in section 4. Let  $OPT^{DP}$  denote the relaxed and i-light CVRP solution that minimizes the extended objective function F given in equation 2. We present a DP to compute  $OPT^{DP}$  in section 3.

### 2.3 A constant factor approximation [11]

For a solution of the red points we use the constant factor approximation of Haimovich and Rinnooy Kan which partitions a traveling salesman tour of the points into tours that cover at most k points [11]. Algorithm 2 below is a version of the Algorithm presented by [11] and it is known to be a 3-approximation.

#### Algorithm 2 TSP Partitioning 3-approximation [11]

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

1: Let  $\pi$  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  $\pi$ .

3: Go around  $\pi$  starting at p, and every time k points are visited, take a detour to the depot. Output the resulting set of |n/k| + 1 tours as the CVRP solution.

**Theorem 2.3.** [11, 4] 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  $Rad(I) = 2/k \sum_{i \in I} d(i, o)$  and let  $TSP(I \cup \{o\})$  denote the length of the minimal tour of I and o. We have that:

- $Rad(I) \leq OPT$ ,
- $TSP(I \cup \{o\}) \leq OPT$  and
- in expectation Algorithm 2's solution has length  $Rad(I) + 2TSP(I \cup \{o\}) \leq 3OPT$ .

### 2.4 Assigning Types

The 3-approximation on the red points has small length if the Rad and TSP of the red points have small value. To ensure this is the case, when selecting y points to drop from segment S, we choose ysuch that length $(y) \leq O(\epsilon)$  length(S), and such that average distance of points in y to the depot is only a  $O(\epsilon)$  fraction of the average distance of points on segment S to the depot. Section 5 shows that both conditions hold if y is chosen as follows:

#### Algorithm 3 Randomized Type Assignment Procedure

Input: A tour segment S from a level  $\ell$  square b containing  $S_a$  active points and requiring y active points to be dropped

- 1: Select one active point p uniformly at random from  $S_a$
- 2: Starting at point p, select the next y-1 points from  $S_a$  that lie consecutively after p on the segment S; if the last active point on the segment before b's boundary is reached without having selected y points, wrap around and select active points from the other end of the tour segment (after S enters b)
- 3: Label each of the y chosen points with type  $\ell$ .

#### 2.5 Proof of Theorem 1.1

Lemma 2.4 proves the correctness and running time of a DP that computes  $OPT^{DP}$ . See section 3 for its proof.

**Lemma 2.4.** (Dynamic Program) Given the set of input points and a randomly shifted dissection, the dynamic program of Section 3 finds an *i*-light and relaxed CVRP solution that minimizes the objective F defined in Equation (2) in time  $n^{\log^{O(1/\epsilon)} n}$ .

The length of the solution output by Algorithm 1 will be the sum of the lengths of the black tours and red tours. Since the black tours are obtained by dropping points from the DP solution they have length at most  $OPT^{DP}$ , which by Theorem 2.2 is at most  $(1+O(\epsilon))OPT$ . Theorem 2.5, which is proved in section 5, shows that the length of the red tours is  $O(\epsilon)OPT$ . **Theorem 2.5.** 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

This section presents the quasi-polynomial time dynamic program and proves Lemma 2.4. The dynamic program (DP) finds  $OPT^{DP}$ , the relaxed and i-light solution that minimizes the extended objective from Equation (2).

The DP table. A configuration C of a dissection square b is a list of entries: for each pair of portals p, q, the configuration has two sublists of entries, one to record information about rounded tour segments and one to record information about unrounded tour segments:

- 1. first sublist:  $(r_1^{p,q}, \ldots, r_i^{p,q}, \ldots, r_{\tau}^{p,q})$ , 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. second sublist:  $(u_1^{p,q}, \ldots, u_j^{p,q}, \ldots, u_{\gamma\cdot\tau}^{p,q})$ , 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 square b and each configuration C of b. The table cell  $L_b[C]$  stores the minimum cost (according to objective F as defined in Equation 2) of placing tour segments in b in a way which is compatible with C and with the relaxed and i-light Definitions 2.3 and 2.4. OPT<sup>DP</sup> is the minimum table entry over all configurations of the root level square.

**Computing the table entries.** The table entries are computed in bottom-up order, in the following manner.

Inductively, let b be a square at level  $\ell$  and let  $b_1, b_2, b_3, b_4$  be the children of b at level  $\ell + 1$ . Since every tour is i-light, a tour segment inside b crosses the boundaries inside b between  $b_1, b_2, b_3, b_4$  at most 4r times, and always through portals. Thus the segment 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. By the relaxed definition applied to level  $\ell + 1$ , each piece is either rounded or one of the  $\gamma \tau$  unrounded tours inside a child of b. Thus every piece can be described by a tuple (p, q, x), where p, q are portals and x is either one of the rounded threshold values  $t_i$  for some  $i < \tau$  or a number  $j \leq \tau \gamma$  indicating it is the j-th unrounded tour in a child square of b. The profile  $\Phi = (p, m_1, n_2), (m_1, m_2, n_1), \dots (m_v, p', n_v)(f)$  of the segment is the list of those 4r+1 tuples (representing tour segment pieces), plus a flag f which is true iff the segment is rounded. Consider a profile  $\Phi$  of b with flag f = 1. Suppose that the concatenation of the pieces described by  $\Phi$  contains x active points. Let  $t_i$  be the threshold value defined by:  $t_i \leq x < t_i(1+\epsilon/\log n)$ . Then the DP counts this segment as having  $t_i$  active points.

Let D denote the number of possible profiles for a segment in square b. For each profile  $\Phi$ , let  $n_{\Phi}$  denote the number of tour segments in b with profile  $\Phi$ . 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 square *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  $\gamma \cdot \tau$  unrounded segment and if gluing  $C_1, C_2, C_3, C_4$  according to *I* yields configuration  $C_0$ .

The cost of a consistent combination is computed in the following way: The cost of configurations  $C_1, \ldots, C_4$  is stored in lookup tables  $L_{b_i}(C_i)$ ,  $1 \le i \le 4$ . Let  $c_b$  denote the total number of tour segments in b as specified by I (also specified by  $C_0$ ). The value of objective function F, as defined by equation 2 of  $(C_1, C_2, C_3, C_4, I)$  is the sum of the costs of  $C_i$  for child square  $b_i$ , plus  $(\epsilon/\log^2 n)2c_b(I)$ . The value of  $L_b(C_0)$  is given by the tuple  $(C_1, C_2, C_3, C_4, I)$  consistent with  $C_0$  with minimum cost.

Analysis of the dynamic program. How many possible configurations are there for a square b? There are  $O(\log^2 n)$  different pairs of portals (p,q); for each (p,q), there are  $\tau$  entries  $r_i^{p,q}$  in the first sublist and  $\tau\gamma$  entries  $u_i^{p,q}$  in the second sublist. Thus a configuration of b is a list of  $O(\tau\gamma\log^2 n)$ 

entries. Each entry (the number of segments  $r_i^{p,q}$  or number of points  $u_j^{p,q}$ ) is an integer between 0 and n, thus the total number of configurations for square b is  $n^{O(\tau \gamma \log^2 n)} = n^{O(\log^{12} n)}$ . As there are  $O(n^2)$  dissection squares, the DP table has size  $n^{O(\log^{12} n)}$  overall.

How many possible profiles are there for a segment in square b?  $\Phi$  has a list of O(r) tuples (p, p', x). There are  $O(\log^2 n)$  choices of portals p, p' and  $O(\tau \gamma)$  choices of x, so there are  $O(\tau \gamma \log^2 n)$  possibilities for each tuple. The flag doubles the number of profiles so there are  $D = (\log^{12} n)^{O(r)}$  possible profiles. As  $r = O(1/\epsilon)$ ,  $D = \log^{O(1/\epsilon)} n$ .

How many possible interfaces are there for a square b? At most  $n^D$ , since each  $n_{\Phi}$  is in [0, n]. This means we have only a quasi-polynomial number of possibilities for the interface vector I for square b.

Checking for consistency takes time polynomial in the size of the list of entries in I and  $C_i$ .

There are  $n^{\log^{O(1)} n}$  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 definition 2.3 but does not actually label points with a specific type. It merely records the number of active points that 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 during the same time that the tours of  $OPT^{DP}$  are constructed. For example while constructing  $OPT^{DP}$ , if we build a tour segment with x active points at level  $\ell$  but the DP's history recorded the segment as having t active points, we can choose any x - t active points from the segment and label them with type  $\ell$ . <sup>1</sup>

### 4 Proof of Structure Theorem

Let  $OPT^{L}$  denote the i-light solution of minimum length. In Lemma 4.2 we show that  $F(OPT^{DP}) \leq (1 + O(\epsilon))OPT^{L}$ , where F is defined in equation 2. As  $OPT^{DP}$  is at most  $F(OPT^{DP})$ , we get that  $OPT^{DP} \leq (1 + O(\epsilon))OPT^{L}$ . Then we apply Corollary 4.1, given below, to show that  $OPT^{DP} \leq (1 + O(\epsilon)OPT)$  and prove our structure Theorem. Corollary 4.1 is a simple generalization of Arora's structure Theorem that shows that  $OPT^{L}$  is near optimal.

**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 a set of tour  $\pi = \pi_1, \ldots, \pi_m$ . Apply Arora's structure Theorem 2.1 to each tour, sum, and use linearity of expectation.

**Lemma 4.2.** In expectation over the random shifted dissection,  $F(OPT^{DP}) \leq (1 + O(\epsilon))OPT^{L}$ .

*Proof.* To start comparing  $OPT^{DP}$  and  $OPT^{L}$ , we apply Lemma 4.3 to turn  $OPT^{L}$  into a solution that satisfies the relaxed definition 2.3. See appendix B for proof of Lemma 4.3

**Lemma 4.3.** Let S be a CVRP solution on input I. There exists an assignment of types to points, that turns S into a solution that satisfies definition 2.3. The tours of S are not modified and the length of S remains unchanged.

Now  $OPT^{DP}$  and  $OPT^{L}$  are both i-light and relaxed solutions. As  $OPT^{DP}$  also minimizes objective function F, we have

$$F(OPT^{DP}) \le F(OPT^L) = OPT^L + (\epsilon/\log^2 n) \sum_{\text{level } \ell} c(\pi^L, \ell) d_\ell$$

<sup>&</sup>lt;sup>1</sup>Labelling *any* active points on the segment with type l will satisfy the relaxed CVRP definition. But we use the randomized type assignment procedure 3 to ensure that a solution on these points will have small cost.

where  $\pi^L$  are the tours of  $OPT^L$ . Now we only need to show that the last term summing the number of crossings in  $OPT^L$  is  $O(\epsilon)OPT^L$  in expectation. Lemma 4.5 allows us to bound the number of crossings in  $OPT^L$  in terms of the number of crossings in OPT, and Lemma 4.4 allows us to charge each crossing to the length of OPT. See appendix C for the proofs.

**Lemma 4.4.** In expectation over the random dissection, for any level  $\ell$   $OPT \ge O(d_{\ell})E[(c(\pi, \ell)])$ .

**Lemma 4.5.** For a random dissection at any level  $\ell$ ,  $E[c(\pi^L, \ell)] \leq O(\log n)E[c(\pi, \ell)]$ .

Applying Lemma 4.5 we get,

$$(\epsilon/\log^2 n) \sum_{\text{level } \ell} E[c(\pi^L, \ell)] \cdot d_\ell \le (\epsilon/\log^2 n) \sum_{\text{level } \ell} O(\log n \cdot d_\ell) E[c(\pi, \ell)]$$
(3)

By Lemma 4.4,  $E[c(\pi, \ell)]O(d_{\ell})$  is at most OPT and as  $\ell_{max} = O(\log n)$ , Equation 3 is  $O(\epsilon)$ OPT. This proves the Lemma as  $OPT \leq OPT^{L}$ .

# 5 Proof of Theorem 2.5

Let R denote set of the points marked red by Algorithm 1. By Theorem 2.3, the 3-approximation on R has cost at most  $\text{Rad}(R) + \text{TSP}(R \cup \{o\})$ , where o is the depot. Lemmas 5.1 and 5.2 proves Theorem 2.5 by showing that in expectation both quantities are  $O(\epsilon)$ OPT.

**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  $TSP(R \cup \{o\}) = O(\epsilon)OPT$ 

#### 5.1 Properties of the randomized type assignment procedure

We state some properties that will be useful in proving the Lemmas 5.1,5.2. See appendix D for the proofs.

Let b be a level  $\ell$  square containing points labelled type  $\ell$  by Algorithm 1 and S be a a rounded tour segment inside b. Let  $R_s = r_1, r_2, \ldots r_d$  be the interval of points labelled type  $\ell$  on S ( $|R_s|$  may be zero) and  $S_a = s_1, s_2, \ldots s_x$  be the active points on S prior to rounding segment S. We now list some Properties of the rounded segment S.

**Property 5.3.**  $|R_s| \leq |S_a| \cdot O(\epsilon/\log n)$ 

**Property 5.4.** A point  $s \in S_a$  is in  $R_s$  with probability  $|R_s|/|S_a|$ .

**Property 5.5.**  $E[length(R_s)] \leq length(S_a) \cdot O(\epsilon/\log n)$ 

#### 5.2 Proof of Lemma 5.1

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

Partition  $R_{\ell}$  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  $R_{\ell}^j$  is the set of red points from tour segment  $S_j$ . By definition we have that

$$\operatorname{Rad}(I) \ge \frac{2}{k} \sum_{j=1}^{m} \sum_{x \in S_j} d(o, x)$$
(4)

As  $R_{\ell}$  is picked randomly, and by Properties 5.4 and 5.3,  $\Pr[x \in R_{\ell}^{j}] \leq O(\epsilon/\log n)$ , so we get

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

Combining this with equation 4 we get that  $E[\operatorname{Rad}(R_{\ell})] \leq O(\epsilon/\log n)\operatorname{Rad}(I)$ .

#### 5.3 Proof of Lemma 5.2

*Proof.* Let  $R_{\ell}$  be the points labeled type  $\ell$  at level  $\ell$ . We show that  $E[\text{TSP}(R_{\ell} \cup \{o\})] \leq O(\epsilon/\log n)\text{OPT}$ . This implies Lemma 5.2 since 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  $B_{\ell}$  be the squares at level  $\ell$  containing points of  $R_{\ell}$ . We consider the cost of  $\text{TSP}(R_{\ell} \cup \{o\})$  in two parts: the outside and inside costs, where intuitively the outside cost will be the cost to get to the squares  $B_{\ell}$  from the depot and the inside cost will be the cost of visiting the red points inside the square. focusing on the outside cost, let C be a set of points containing at least one portal from each square of  $B_{\ell}$  such that  $MST(C \cup \{o\})$  is minimized<sup>2</sup>. The optimal tour of  $R_{\ell} \cup \{O\}$  is at most  $2MST(C \cup \{o\})$ , thus

$$TSP(R_{\ell} \cup \{o\}) \le 2MST(C \cup \{o\}) + \sum_{b \in B_{\ell}} \text{ inside cost of } b$$
(5)

Claims 5.6 and 5.7 prove that the quantities on the right hand side of equation 5 are both  $O(\epsilon/\log n)$ OPT, proving that  $TSP(R_{\ell} \cup \{o\}) \leq O(\epsilon/\log n)$ .

**Claim 5.6.** In expectation over the random shifted dissection,  $E[2MST(C \cup \{o\})] \leq O(\epsilon/\log n)OPT$ .

*Proof.* We will show that,  $MST(C \cup \{o\}) \leq O(\epsilon/\log n) OPT^{DP}$ . The claim follows, as  $OPT^{DP} \leq (1 + O(\epsilon) OPT)$ , 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, A with value v on G.

$$v = \min_{x} (w_e \cdot x_e) \quad \text{s.t.} \quad \left\{ \begin{array}{ll} \sum_{e \in \delta(S)} x_e & \geq \gamma/4r \quad \forall S \subset V \\ x_e & \geq 0 \end{array} \right.$$

As each  $b \in B_{\ell}$  contains some points labelled  $\ell$  (i.e at least a group of rounded segments),  $OPT^{DP}$  contains are least  $\gamma$  tour segments crossing into b. Since each tour in  $OPT^{DP}$  is i-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,  $OPT^{DP} \ge v$ .

Now consider the linear program A' below. A' is the relaxation of the IP for MST. Let v' be the value of A' on graph G.

$$v' = \min_{x} (w_e \cdot x_e) \quad \text{s.t.} \quad \left\{ \begin{array}{ll} \sum_{e \in \delta(S)} x_e & \ge 1 & \forall S \subset V \\ x_e & \ge 0 \end{array} \right.$$

Observe that for any solution x of A,  $x' = x \cdot 4r/\gamma$  is a solution for A'. As A and A' have the same objective,  $v \cdot 4r/\gamma = v'$ . The MST relaxation A' is known to have integrality gap at most 2 [17], so that  $v' \ge 2$  MST  $(C \cup \{o\})$ . Thus we have that

$$OPT^{DP} \ge v = v' \cdot (\gamma/4r) \ge MST(C \cup \{o\}) \cdot (\gamma/2r)$$

Thus  $(2r/\gamma) \cdot \text{OPT}^{DP} \ge MST(C \cup \{o\})$  and as  $2r/\gamma = o(\epsilon/\log n)$ , the Claim is proved.

**Claim 5.7.** In expectation over the random dissection and the random type assignment the total inside cost at level  $\ell$  is at most  $O(\epsilon/\log n)OPT$ .

 $<sup>^{2}</sup>$  C is used only for the analysis and does not need to be found explicitly

*Proof.* The inside cost at level  $\ell$  is the sum of the inside costs of each square  $b \in B_{\ell}$ . The contribution of square  $b \in B_{\ell}$  to the inside cost, is the sum the length of the intervals of type  $\ell$  points inside b plus the cost of connecting these intervals to the boundary of b. In Claim 5.8 we show that, in expectation over the random type assignment, the sum over all squares in  $B_{\ell}$  of the length of intervals of type  $\ell$  points is  $O(\epsilon/\log n) \text{OPT}^{DP}$ . The type  $\ell$  intervals inside each  $b \in B_{\ell}$  must be connected to each other and to the boundary of their square. We refer to  $CC(\ell)$  as the total the connection cost at level  $\ell$ .  $CC(\ell)$  is the sum of the length of the boundaries of each square  $b \in B_{\ell}$  plus the cost of connecting the type  $\ell$  intervals inside each  $b \in B_{\ell}$  to the boundary of b. Claim 5.9 shows that  $CC(\ell) = O(\epsilon/\log n)F(\text{OPT}^{DP})$ . By Lemma 4.2 and Corollary 4.1  $F(\text{OPT}^{DP}) \leq (1 + O(\epsilon))\text{OPT}$  in expectation, proving this Claim. □

**Claim 5.8.** In expectation over the random type assignment the length of all intervals of type  $\ell$  is  $O(\epsilon/\log n)OPT^{DP}$ .

*Proof.* Sum the lengths of intervals of type  $\ell$  over all squares in  $B_{\ell}$  squares, use Property 5.5 and linearity. See appendix D for a detailed proof.

**Claim 5.9.** The total connection cost for level  $\ell$  is  $O(\epsilon/\log n)F(OPT^{DP})$ .

*Proof.* Let  $CC(\ell)$  denote the total connection cost which is the cost of connecting the type  $\ell$  intervals inside each  $b \in B_{\ell}$  to the boundary of the square. Focus on one square  $b \in B_{\ell}$ . As  $OPT^{DP}$  is relaxed, bhas  $g_b \cdot \gamma$  rounded segments for some integer  $g_b > 0$ . Consider a group of  $\gamma$  rounded segments. Let R'be a set containing one type  $\ell$  point from each of the  $\gamma$  segments in the group. The cost to connect all red intervals in the group with the boundary of the square is at most  $MST(R') + 5 \cdot d_{\ell}$ , where  $d_{\ell}$  is the side length of square b. We bound this using the following bound for TSP [11][4]. (See [11] for a proof).

**Theorem 5.10.** Let U be a set of points in 2d-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 5.10 we have that  $MST(R') + 5 \cdot d_{\ell} = O(d_{\ell} \cdot \sqrt{\gamma})$ . This holds for each of the  $g_b$  groups of rounded segments inside b thus we have that the total connection cost for b is  $O(g_b \cdot d_{\ell} \cdot \sqrt{\gamma})$ .

Let  $g_{\ell} = \sum_{b \in B_{\ell}} g_b$ , be the total number of groups of rounded segments at level  $\ell$ . The total connection cost for level  $\ell$  is the sum of the connection costs over all squares  $b \in B_{\ell}$ ,

$$CC(\ell) = O(g_{\ell} \cdot d_{\ell} \cdot \sqrt{\gamma}) \tag{6}$$

Each of the  $g_{\ell} \cdot \gamma$  rounded tour segments intersects twice with the boundary of a level  $\ell$  dissection square, thus:  $c(\pi^{DP}, \ell) \geq 2g_{\ell}\gamma$ , with  $\pi^{DP}$  being the tours of  $OPT^{DP}$ . Using equation 6,

$$O\left(d_{\ell}/\sqrt{\gamma}\right) \cdot c(\pi^{DP}, \ell) \ge CC(\ell) \tag{7}$$

For objective function F, (defined in 2), we have  $(\log^2 n/\epsilon) \cdot F(\text{OPT}^{DP}) \ge c(\pi^{DP}, \ell)d_\ell$ . Substituting it for  $c(\pi^{DP}, \ell)d_\ell$  in equation 7 we get,  $O(1/\sqrt{\gamma})(\log^2 n/\epsilon) \cdot F(\text{OPT}^{DP}) \ge CC(\ell)$ . As  $\gamma = \log^9 n/\epsilon^4$  this reduces to  $O(\epsilon/\log n) \cdot F(\text{OPT}^{DP}) \ge CC(\ell)$ , which proves this Claim.

### 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 [7, 14]. 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)  $\operatorname{Rad}(Y) \leq O(\epsilon/\log n)\operatorname{Rad}(S)$ and (2)length(Y)  $\leq O(\epsilon/\log n)\operatorname{length}(S)$ . In Lemma 5.1 and Property 5.5 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.

# References

- S. Arora. Polynomial-time approximation schemes for Euclidean TSP and other geometric problems. JACM, 45(5):753-782, 1998.
- S. Arora. Approximation schemes for NP-hard geometric optimization problems: A survey. Mathematical Programming, 97 (1,2) July 2003.
- [3] 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.
- [4] 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.
- [5] T. Asano, N. Katoh, H. Tamaki, and T. Tokuyama. Covering points in the plane by ktours: a polynomial approximation scheme for fixed k. IBM Tokyo Research Laboratory Research Report RT0162, 1996.
- [6] 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. 2008.
- [7] 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.
- [8] Dantzig, G.B.; Ramser, J.H. "The Truck Dispatching Problem". Management Science 6 (1): 80-91. (1959).
- [9] 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.
- [10] 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.
- [11] M. Haimovich and A.H.G Rinnooy Kan. Bounds and heuristic for capacitated routing problems. Mathematics of Operations Research, 10(4), 527-542, 1985.
- [12] S. Kolliopoulos and S. Rao. A nearly linear-time approximation scheme for the Euclidean k-media problem. SIAM J. Comput., 37(3):757782, 2007.
- [13] J. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomialtime approximation scheme for geometric TSP, k-MST, and related problems. SIAM J. Comput., 28(4):12981309, 1999.
- [14] S. Rao and W. Smith. Approximating geometrical graphs via "spanners" and "banyans". In 30th STOC, pages 540-550, 1998.
- [15] 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.
- [16] Paolo Toth , Daniele Vigo. The vehicle routing problem. Society for Industrial and Applied Mathematics. Philadelphia, PA, 2001.
- [17] Approximation Algorithms. Vijay Vazarani. chapter 20.
- [18] http://neo.lcc.uma.es/radi-aeb/WebVRP/
- [19] http://en.wikipedia.org/wiki/Vehicle\_routing\_problem

### A Technical Tools

**The Perturbation** Define a bounding square as the smallest square 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 square. Move every input point to the center of the grid square it lies in. Several points may map to the same grid square 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.

A solution for the perturbed instance can be extended to a solution for the original instance by taking detours from the grid centers to the locations of the points. The cost of such detours will be at most  $n \cdot \sqrt{2}d\epsilon/n$ . As the two farthest points must be visited from the depot we have that  $2d \leq \text{OPT}$ . Thus the total cost of the detours is  $\leq \epsilon \text{OPT}$  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 is  $4n/\epsilon$  which is O(n) for constant  $\epsilon$ .

**Randomized Dissection** A dissection of the bounding square is obtained by recursively partitioning a square into 4 smaller squares of equal size using one horizontal and one vertical dissection line. The recursion stops when the smallest squares have size  $1 \times 1$ . The bounding square has level 0, the 4 squares created by the first dissection have level 1, and since L = O(n) the level of the  $1 \times 1$  squares will be  $\ell_{\max} = O(\log n)$ . The horizontal and vertical dissection lines are also assigned levels. The boundary of the bounding square has level 0, the  $2^{i-1}$  horizontal and  $2^{i-1}$  vertical lines that form level *i* squares by partitioning the level i-1 squares are each assigned level *i*. A randomized dissection of the bounding square 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 0 horizontal line is moved from L/2 to  $a + L/2 \mod L$  and the level 0 vertical line is moved to  $b + L/2 \mod L$ . The dissection is "wrapped around" and wrapped around squares are treated as one region. The probability that a line *l* becomes a level  $\ell$  dissection line in the randomized dissection is

$$Pr(\operatorname{level}(l) = \ell) = 2^{\ell}/L \tag{8}$$

### **B** Relaxed CVRP

Proof. (Proof of Lemma 4.3) We give an type assignment procedure which initially assigns all points to type -1 and never sets any point to a type below -1. As S is a valid CVRP solution all tours in S contain at most k points of type -1 satisfying the first condition of definition 2.3. The procedure works in a bottom up fashion in the dissection tree from level  $\ell_{\max}$  to level 0. At the current level  $\ell$ consider each dissection square one at a time. For any threshold value  $t_i$  for  $i \leq \tau$  while square b has at least  $\gamma$  unrounded tour segments, select exactly  $\gamma$  such segments and perform a group-rounding as follows: Examine each of the  $\gamma$  segments one at a time. If the unrounded tour 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  $\ell$ . Perform as many group-rounding steps as necessary until square b has at most  $\gamma * \tau$  unrounded tours. Proceed similarly to the other squares at level  $\ell$ .

The type assignment procedure does not change the construction of any of the tour in S thus the cost of S is unchanged. Now we show that definition 2.3 is satisfied. While working at a level  $\ell$ , the procedure performs group-rounding on each square at level  $\ell$  until the square has at most  $\gamma * \tau$  unrounded segments. As the group-rounding rounds exactly  $\gamma$  segments together there will always be an integer multiple of  $\gamma$  rounded segments in each square. Condition (ii) continues to hold while the procedure works on levels  $j < \ell$  as in those levels the procedure only labels points with type  $j < \ell$  so the number of active points at level  $\ell$  remains the same. As for the third condition of definition 2.3, note that before rounding a segment at level  $\ell$ , all points on the segment have type -1 or a type greater than  $\ell$ . Thus prior to rounding the segment will have x active points at level  $\ell$  and at level  $\ell + 1$ . Let  $t_i \leq x \leq t_{i+1}$ . To round the segment at  $\ell$ , label  $x - t_i$  points with type  $\ell$ . This still leaves  $t_i$  active points at level  $\ell$  and x active points at level l + 1. As  $t_i(1 + \epsilon/\log n) = t_{i+1} > x$ , the third condition of definition 2.3 is satisfied.

### C Extensions of Arora's Results to CVRP

Let  $t(\pi_j, l)$  denote the number of times a tour  $\pi_j$  crosses dissection line l. Arora proved that the length $(\pi_j) \geq \frac{1}{2} \sum_{\text{line } l} t(\pi_j, l)$  [1]. For CVRP as  $\sum_j \pi_j = \text{OPT}$ , this implies

$$OPT \ge 1/2 \sum_{\text{line } l} t(\pi, l)$$
(9)

With probability  $2^{\ell+1}/L$  a dissection line l forms the boundary for some level  $\ell$  square <sup>3</sup>. For any level  $\ell$  as  $E(c(\pi, \ell)) = \sum_{l \mid n \in I} t(\pi, l) \cdot \Pr[l$  forms a boundary a level  $\ell$  square ], we have

$$E(c(\pi, \ell)) = \frac{2^{\ell+2}}{L} \sum_{\text{line } l} t(\pi, l)$$
(10)

*Proof.* (Proof of Lemma 4.4) Combine equations 9 and 10 to get  $E(c(\pi, \ell)) \leq \frac{2^{\ell+3}}{L}$  OPT. The Lemma follows from the fact that a level  $\ell$  square has side length  $d_{\ell} = L/2^{\ell}$ .

Let  $\pi$  denote the tours of OPT and  $\pi^L$  the tours of OPT<sup>L</sup>. Arora gives a procedure to modify  $\pi$  into the light tours  $\pi^L$  [1]. But the procedure may create new crossings in  $\pi^L$  with dissection squares not present in  $\pi$ . The next claim bounds the number of these new crossings.

*Proof.* (Proof of Lemma 4.5) To modify  $\pi$  into  $\pi^L$  Arora's procedure does bottom up patching to ensure that each tour crosses the edges of dissection squares at most O(r) times. The second step is to take detours (along the sides of squares) to make the tours portal respecting. Both steps, patching and detouring, can add new crossings to  $\pi^L$  which are not present in  $\pi$ . A patching on edge e of square b adds at most 6 new crossings to each child square inside b with edge e as a boundary and each detour adds at at most 2 new crossings to each such child square.

As Arora's procedure works bottom up the patching and detours taken at all levels  $j > \ell$  can add crossings in  $\pi^L$  with level  $\ell$  squares. As there are  $2^{\ell-j}$  level  $\ell$  squares along the edge of a level j square, a patching step on the edge of a level j square adds at most  $6 \cdot 2^{\ell-j}$  new crossings at level  $\ell$  and each detour along these edges add at most  $2 \cdot 2^{\ell-j-1}$  new crossings at level  $\ell$ . Thus we have that,

$$E[c(\pi^{L}, \ell)] \leq 6 \sum_{j \leq \ell} 2^{\ell-j} (\# \text{ patching at level } j) + 2 \sum_{j \leq \ell} 2^{\ell-j} (\# \text{ detours at level } j)$$

A patching step is performed only when a group of at least O(r) crossings are identified in  $\pi$ , thus an upper bound on the number of expected patching at level j is  $E[c(\pi, j)]/O(r)$ . Since each crossing at level j can require a detour, we get,

$$E[c(\pi^{L}, \ell)] \le 6 \sum_{j \le \ell} 2^{\ell - j} \frac{E[c(\pi, j)]}{O(r)} + 2 \sum_{j \le \ell} 2^{\ell - j} E[c(\pi, j)]$$

Writing out the definitions of  $E[c(\pi, j)]$  using equation 10 we see that for all  $j \leq \ell$ ,  $E[c(\pi, j)] = E[c(\pi, \ell)]/2^{\ell-j}$ . Substituting this into the above equation simplifies it to,

$$E[c(\pi^{L}, \ell)] \le 6 \sum_{j \le \ell} \frac{E[c(\pi, \ell)]}{O(r)} + \sum_{j \le \ell} E[c(\pi, \ell)]$$

As  $\ell \leq O(\log n)$ , we have that  $E[c(\pi^L, \ell)] \leq O(\log n)E[c(\pi, \ell)](1 + O(1/r))$ . This proves the Lemma as  $O(1/r) = O(\epsilon)$  is a constant.

<sup>&</sup>lt;sup>3</sup>The boundaries of level  $\ell$  squares are formed by lines at levels  $\leq \ell$ 

### **D** Properties of the randomized type assignment procedure

Let b be a level  $\ell$  square containing points labelled type  $\ell$  by Algorithm 1 and S be a a rounded tour segment inside b. Let  $R_s = r_1, r_2, \ldots r_d$  be the interval of points labelled type  $\ell$  on S ( $|R_s|$  may be zero) and  $S_a = s_1, s_2, \ldots s_x$  be the active points on S prior to rounding segment S. We now list some Properties of the rounded segment S.

*Proof.* (Proof of Property 5.3) In the DP's history, S has a profile  $\Phi$  with a flag set to true as S is a rounded segment. Suppose that after S is concatenated according to  $\Phi$  it has x active points. Then 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 we would need to set at most  $x - t_i \leq x(\epsilon/\log n)$  active points to  $\ell$ . Thus  $|R_s| \leq x\epsilon/\log n$  while  $|S_a| = x$ .

*Proof.* (Proof of Property 5.4) Each point  $s \in S_a$  belongs to  $|R_s|$  intervals as each interval consists of  $|R_s|$  consecutive points. There are a total of  $|S_a|$  different intervals, each starting at a different point in  $S_a$  and Algorithm 3 picks uniformly among them.

**Definition D.1.** (Length of interval) Let  $b_1, b_2$  be the points on the boundary of b where S enters and exits b. Then S visits  $s_1$  after entering at  $b_1$  and it visits  $s_x$  before exiting from  $b_2$  Let d(u, v) denote the distance between points u and v. If  $R_s$  does not contain both  $s_1$  and  $s_x$  then length( $R_s$ ) =  $\sum_{i=1}^{d} d(s_i, s_{i+1})$ . Otherwise let  $r_e = s_x$ , then  $r_{e+1} = s_1$  as Algorithm 3 wraps around and length( $R_s$ ) =  $\sum_{i=1}^{e-1} d(s_i, s_{i+1}) + d(s_x, b_2) + d(b_1, s_1) + \sum_{i=e+1}^{d} d(r_i, r_{i+1})$ .

*Proof.* (Proof of Property 5.5) Let  $b_1$  and  $b_2$  be the points where S enters and exits square b. Define  $z_x = d(b_1, s_1) + d(s_x, b_2)$  and  $z_i = d(s_i, s_{i+1})$ , for  $1 \le i < x$ . Then the length of  $S_a$  is  $\sum_{i=1}^x z_i$  and

$$E[\operatorname{length}(R_a)] = \sum_{i=1}^{x} z_i \operatorname{Pr}[z_i \text{ is counted inside } R_s]$$

For all *i* the probability that  $z_i$  is counted is the probability that  $s_i$  and its consecutive point,  $s_{i+1}$ , are both included in  $R_s$ . (The consecutive point of  $s_x$  is  $s_1$ ). A point  $s \in S_a$  belongs to exactly  $|R_s|$  intervals and the consecutive point of *s* appears in  $|R_s|-1$  of these intervals. Thus the  $\Pr[z_i$  is counted  $] = (|R_s|-1)/(|S_a|)$ . Applying Property 5.3,

$$E[\operatorname{length}(R_s)] = \sum_{i=1}^{x} z_i \frac{|R_s| - 1}{|S_a|} = O\left(\frac{\epsilon}{\log n}\right) \sum_{i=1}^{x} z_i = O\left(\frac{\epsilon}{\log n}\right) \operatorname{length}(S_a)$$

*Proof.* (Proof of Claim 5.8) Sum the lengths of intervals of type  $\ell$  over all squares in  $B_{\ell}$  squares, use Property 5.5 and linearity. Consider a square  $b \in B_{\ell}$  and the type  $\ell$  points inside  $b, R_b$ . 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  $\ell$  from tour segment  $s_j$ . By Property 5.5, 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  $\text{OPT}_b^{DP}$  denote the projection of  $\text{OPT}^{DP}$  inside square b.  $\text{OPT}_b^{DP} \geq \sum_{j=1}^m \text{length}(s_j)$ , and  $\text{OPT}^{DP}$  is at least the sum of  $\text{OPT}_b^{DP}$  over all squares  $b \in B_{\ell}$ . This implies that the total length of red intervals at level  $\ell$  is at most  $O(\epsilon/\log n) \text{OPT}^{DP}$ . □