# A PTAS for the Weighted Unit Disk Cover Problem

Jian Li \* Yifei Jin <sup>†</sup>

Institute for Interdisciplinary Information Sciences Tsinghua University, China

June 22, 2021

#### Abstract

We are given a set of weighted unit disks and a set of points in Euclidean plane. The minimum weight unit disk cover (WUDC) problem asks for a subset of disks of minimum total weight that covers all given points. WUDC is one of the geometric set cover problems, which have been studied extensively for the past two decades (for many different geometric range spaces, such as (unit) disks, halfspaces, rectangles, triangles). It is known that the unweighted WUDC problem is NP-hard and admits a polynomial-time approximation scheme (PTAS). For the weighted WUDC problem, several constant approximations have been developed. However, whether the problem admits a PTAS has been an open question. In this paper, we answer this question affirmatively by presenting the first PTAS for WUDC. Our result implies the first PTAS for the minimum weight dominating set problem in unit disk graphs. Combining with existing ideas, our result can also be used to obtain the first PTAS for the maximum lifetime coverage problem and an improved 4.475-approximation for the connected dominating set problem in unit disk graphs.

<sup>\*</sup>Email:lijian83@mail.tsinghua.edu.cn

<sup>&</sup>lt;sup>†</sup>Email: jin-yf13@mails.tsinghua.edu.cn

### 1 Introduction

The set cover problem is a central problem in theoretical computer science and combinatorial optimization. A set cover consists of a ground set U and collection S of subsets of U. Each set  $S \in S$  has a non-negative weight  $w_S$ . The goal is to find a subcollection  $C \subseteq S$  of minimum total weight such that  $\bigcup C$  covers all elements of U. The approximibility of the general SC problem is rather well understood: it is well known that the greedy algorithm is an  $H_n$ -approximation  $(H_n = \sum_{i=1}^n 1/i)$  and obtaining a  $(1 - \epsilon) \ln n$ -approximation for any constant  $\epsilon > 0$  is NP-hard [15, 23]. In the geometric set cover problem, U is a set of points in some Euclidean space  $\mathbb{R}^d$ , and S consists of geometric objects (e.g., disks, squares, triangles). In the geometric setting, we can hope for better-than-logarithmic approximations due to the special structure of S. Most geometric set cover problems are still NP-hard, even for the very simple classes of objects such as unit disks [11, 29] (see [9, 27] for more examples and exceptions). Approximation algorithms for geometric set cover have been studied extensively for the past two decades, not only because of the importance of the problem per se, but also its rich connections to other important notions and problems, such as VC-dimension [5, 12, 22],  $\epsilon$ -net, union complexity [10, 39, 40], planar separators [25, 35], even machine scheduling problems [3].

In this work, we study the geometric set cover problem with one of the simplest class of objects – unit disks. The formal definition of our problem is as follows:

**Definition 1.** Weighted Unit Disk Cover (WUDC): Given a set  $\mathcal{D} = \{D_1, \ldots, D_n\}$  of n unit disks and a set  $\mathcal{P} = \{P_1, \ldots, P_m\}$  of m points in Euclidean plane  $\mathbb{R}^2$ . Each disk  $D_i$  has a weight  $w(D_i)$ . Our goal is to choose a subset of disks to cover all points in  $\mathcal{P}$ , and the total weight of the chosen disks is minimized.

WUDC is the general version of the following minimum weight dominating set problem in unit disk graphs (UDG).

**Definition 2.** Minimum Weight Dominating Set (MWDS) in UDG: We are given a unit disk graph G(V, E), where V is a set of weighted points in  $\mathbb{R}^2$  and  $(u, v) \in E$  iff  $||u - v|| \leq 1$  for any  $u, v \in V$ . A dominating set S is a subset of V such that for any node  $v \notin S$ , there is some  $u \in S$  with  $(u, v) \in E$ . The goal of the minimum weight dominating set problem is to find a dominating set with the minimum total weight.

Given a dominating set instance with point set V, we create a WUDC instance by placing, for each point in  $v \in V$ , a point (to be covered) co-located with v and a unit disk centered at v, with weight equal to the weight of v. Thus, WUDC is a general version of problem MWDS in UDG. In this paper, we only state our algorithms and results in the context of WUDC.

### 1.1 Previous Results and Our Contribution

We first recall that a polynomial time approximation scheme (PTAS) for a minimization problem is an algorithm  $\mathcal{A}$  that takes an input instance, a constant  $\epsilon > 0$ , returns a solution SOL such that SOL  $\leq (1 + \epsilon)$ OPT, where OPT is the optimal value, and the running time of  $\mathcal{A}$  is polynomially in the size of the input for any fixed constant  $\epsilon$ .

WUDC is NP-hard, even for the unweighted version (i.e.,  $w(D_i) = 1$ ) [11]. For unweighted dominating set in unit disk graphs, Hunt et al. [31] obtained the first PTAS in unit disk graphs. For the more general disk graphs, based on the connection between geometric set cover problem

and  $\epsilon$ -nets, developed in [5,12,22], and the existence of  $\epsilon$ -net of size  $O(1/\epsilon)$  for halfspaces in  $\mathbb{R}^3$  [36] (see also [26]), it is possible to achieve a constant factor approximation. As estimated in [35], these constants are at best 20 (A recent result [6] shows that the constant is at most 13). Moreover, there exists a PTAS for unweighted disk cover and minimum dominating set via the local search technique [25,35].

For the general weighted WUDC problem, the story is longer. Ambühl et al. [2] obtained the first approximation for WUDC with a concrete constant 72. Applying the shifting technique of [28], Huang et al. [30] obtained a  $(6 + \epsilon)$ -approximation algorithm for WUDC. The approximation factor was later improved to  $(5 + \epsilon)$  [13], and to  $(4 + \epsilon)$  by several groups [14, 20, 42]. The current best ratio is 3.63. <sup>1</sup> Besides, the quasi-uniform sampling method [10, 40] provides another approach to achieve a constant factor approximation for WUDC (even in disk graphs). However, the constant depends on several other constants from rounding LPs and the size of the union complexity. Very recently, based on the separator framework of Adamaszek and Wiese [1], Mustafa and Raman [34] obtained a QPTAS (Quasi-polynomial time approximation scheme) for weighted disks in  $\mathbb{R}^2$  (in fact, weighted halfspaces in  $\mathbb{R}^3$ ), thus ruling out the APX-hardness of WUDC.

Another closely related work is by Erlebach and van Leeuwen [21], who obtained a PTAS for set cover on weighted unit squares, which is the first PTAS for weighted geometric set cover on any planar objects (except those poly-time solvable cases [9,27]). Although it may seem that their result is quite close to a PTAS for weighted WUDC, as admitted in their paper, their technique is insufficient for handling unit disks and "completely different insight is required".

In light of all the aforementioned results, it seems that we should expect a PTAS for WUDC, but it remains to be an open question (explicitly mentioned as an open problem in a number of previous papers, e.g., [2, 17, 19-21, 38]). Our main contribution in this paper is to settle this question affirmatively by presenting the first PTAS for WUDC.

**Theorem 3.** There is a polynomial time approximation scheme for the WUDC problem. The running time is  $n^{O(1/\epsilon^9)}$ .

Because WUDC is more general than MWDS, we immediately have the following corollary.

**Corollary 4.** There is a polynomial time approximation scheme for the minimum weight dominating set problem in unit disk graphs.

We note that the running time  $n^{\text{poly}(1/\epsilon)}$  is nearly optimal in light of the negative result by Marx [33], who showed that an EPTAS (i.e., Efficient PTAS, with running time  $f(1/\epsilon)\text{poly}(n)$ ) even for the unweighted dominating set in UDG would contradict the exponential time hypothesis.

Finally, in Section 9, we show that our PTAS for WUDC can be used to obtain improved approximation algorithms for two important problems in wireless sensor networks, the connected dominating set problem and the maximum lifetime coverage problem in UDG.

# 2 Our Approach - A High Level Overview

By the standard shifting technique [16], it suffices to provide a PTAS for WUDC when all disks lies in a square of constant size (we call it a block, and the constant depends on  $1/\epsilon$ ). This idea is formalized in Huang et al. [30], as follows.

<sup>&</sup>lt;sup>1</sup> The algorithm can be found in Du and Wan [17], who attributed the result to a manuscript by Willson et al.

**Lemma 5** (Huang ta al. [30]). Suppose there exists a  $\rho$ -approximation for WUDC in a fixed  $L \times L$ block, with running time f(n, L). Then there exists a  $(\rho + O(1/L))$ -approximation with running time  $O(L \cdot n \cdot f(n, L))$  for WUDC. In particular, setting  $L = 1/\epsilon$ , there exists a  $(\rho + \epsilon)$ -approximation for WUDC, with running time  $O(\frac{1}{\epsilon} \cdot n \cdot f(n, \frac{1}{\epsilon}))$ .

In fact, almost all previous constant factor approximation algorithms for WUDC were obtained by developing constant approximations for a single block of a constant size (which is the main difficulty <sup>2</sup>). The main contribution of the paper is to improve on the previous work [2, 13, 20, 30]for a single block, as in the following lemma.

**Lemma 6.** There exists a PTAS for WUDC in a fixed block of size  $L \times L$  for  $L = 1/\epsilon$ . The running time of the PTAS is  $n^{O(1/\epsilon^9)}$ 

From now on, the approximation error guarantee  $\epsilon > 0$  is a fixed constant. Whenever we say a quantity is a constant, the constant may depend on  $\epsilon$ . We use OPT to represent the optimal solution (and the optimal value) in this block. We use capital letters  $A, B, C, \ldots$  to denote points, and small letters  $a, b, c, \ldots$  to denote arcs. For two points A and B, we use |AB| to denote the line segment connecting A and B (and its length). We use  $D_i$  to denote a disk and  $D_i$  to denote its center. For a point A and a real r > 0, let D(A, r) be the disk centered at A with radius r. For a disk  $D_i$ , we use  $\partial D_i$  to denote its boundary. We call a segment of  $\partial D_i$  an arc.

First, we guess that whether OPT contains more than C disks or not for some constant C. If OPT contains no more than C disks, we enumerate all possible combinations and choose the one which covers all points and has the minimum weight. This takes  $O\left(\sum_{i=1}^{C} \binom{n}{i}\right) = O(n^{C})$  time, which is polynomial.

The more challenging case is whether OPT contains more than C disks. In this case, we guess (i.e., enumerate all possibilities) the set  $\mathcal{G}$  of the C most expensive disks in OPT. There are at most a polynomial number (i.e.,  $O(n^C)$ ) possible guesses. Suppose our guess is correct. Then, we delete all disks in  $\mathcal{G}$  and all points that are covered by  $\mathcal{G}$ . Let  $D_t$  (with weight  $w_t$ ) be the cheapest disk in  $\mathcal{G}$ . We can see that  $OPT \geq Cw_t$ . Moreover, we can also safely ignore all disks with weight larger than  $w_t$  (assuming that our guess is correct). Now, our task is to cover the remaining points with the remaining disks, each having weight at most  $w_t$ . We use  $\mathcal{D}' = \mathcal{D} \setminus \mathcal{G}$  and  $\mathcal{P}' = \mathcal{P} \setminus \mathcal{P}(\mathcal{G})$ to denote the set of the remaining disks and the set of remaining points respectively, where  $\mathcal{P}(\mathcal{G})$ denote the set of points covered by some disk in  $\mathcal{G}$ .

Next, we carefully choose to include in our solution a set  $\mathcal{H} \subseteq \mathcal{D}'$  of at most  $\epsilon C$  disks. The purpose of  $\mathcal{H}$  is to break the whole instance into many (still a constant) small pieces (substructures), such that each substructure can be solved optimally, via dynamic programming. <sup>3</sup> One difficulty is that the substructures are not independent and may interact with each other (i.e., a disk may appear in more than one substructure). In order to apply the dynamic programming technique to all substructures simultaneously, we have to ensure the orders of the disks in different substructures are consistent with each other. Choosing  $\mathcal{H}$  to ensure a globally consistent order of disks is in fact the main technical challenge of the paper.

Suppose we have a set  $\mathcal{H}$  which suits our need (i.e., the remaining instance  $(\mathcal{D}' \setminus \mathcal{H}, \mathcal{P}' \setminus \mathcal{P}(\mathcal{H}))$  can be solved optimally in polynomial time by dynamic programming). Let  $\mathcal{S}$  be the optimal

<sup>&</sup>lt;sup>2</sup> For the unweighted dominating set problem in a single block, it is easy to see the optimal number of disks is bounded by a constant, which implies that we can compute the optimum in poly-time. However, for the weighted dominating set problem or WUDC the optimal solution in a single block may consist of O(n) disks.

<sup>&</sup>lt;sup>3</sup> An individual substructure can be solved using a dynamic program similar to [2, 27].

solution of the remaining instance. Our final solution is  $SOL = \mathcal{G} \cup \mathcal{H} \cup \mathcal{S}$ . First, we can see that

$$w(\mathcal{S}) \leq w(\mathsf{OPT} - \mathcal{G} - \mathcal{H}) \leq \mathsf{OPT} - w(\mathcal{G}),$$

since  $\mathsf{OPT} - \mathcal{G} - \mathcal{H}$  is a feasible solution for the instance  $(\mathcal{D}' \setminus \mathcal{H}, \mathcal{P}' \setminus \mathcal{P}(\mathcal{H}))$ . Hence, we have that

$$\mathsf{SOL} = w(\mathcal{G}) + w(\mathcal{H}) + w(\mathcal{S}) \le \mathsf{OPT} + \epsilon C w_t \le (1 + \epsilon)\mathsf{OPT},$$

where the 2nd to last inequality holds because  $|\mathcal{H}| \leq \epsilon C$ , and the last inequality uses the fact that  $\mathsf{OPT} \geq w(\mathcal{G}) \geq Cw_t$ .

**Constructing**  $\mathcal{H}$ : Now, we provide a high level sketch for how to construct  $\mathcal{H} \subseteq \mathcal{D}'$ . First, we partition the block into *small squares* with side length  $\mu = O(\epsilon)$  such that any disk centered in a square can cover the whole square and the disks in the same square are close enough. Let the set of small squares be  $\Xi = \{\Gamma_{ij}\}_{1 \leq i,j \leq K}$  where  $K = L/\mu$ . For a small square  $\Gamma$ , let  $\mathsf{D}_{s_{\Gamma}} \in \Gamma$  and  $\mathsf{D}_{t_{\Gamma}} \in \Gamma$  be the furthest pair of disks (i.e.,  $|D_{s_{\Gamma}}D_{t_{\Gamma}}|$  is maximized). We include the pair  $\mathsf{D}_{s_{\Gamma}}$  and  $\mathsf{D}_{t_{\Gamma}}$  in  $\mathcal{H}$ , for every small square  $\Gamma \in \Xi$ , and call the pair the square gadget for  $\Gamma$ . See Figure 1 for an example. We only need to focus on covering the remaining points in the uncovered region  $\mathbb{U}(\mathcal{H})$ .

We consider all disks in a small square  $\Gamma$ . The uncovered portion of those disks defines two disjoint connected regions (See right hand side of Figure 1, the two shaded regions). We call such a region, together with all relevant arcs, a *substructure* (formal definition in Section 4). In fact, we can solve the disk covering problem for a single substructure optimally using dynamic programming (which is similar to the dynamic program in [2, 27]). It appears that we are almost done, since ("intuitively") all square gadgets have already covered much area of the entire block, and we should be able to use similar dynamic program to handle all such substructures as well. However, the situation is more complicated (than we initially expected) since the arcs are dependent. See Figure 2 for a "not-so-complicated" example. Firstly, there may exist two arcs (called *sibling arcs*) which belong to the same disk when the disk is centered in the *core-central area*, as show in Figure 1). The dynamic program has to make decisions for two sibling arcs, which belong to two different substructures (called R(emotely)-correlated substructures), together. Second, in order to carry out dynamic program, we need a suitable order of all arcs. To ensure such an order exists, we need all substructures interact with each other "nicely".

In particular, besides all square gadgets, we need to add into  $\mathcal{H}$  a constant number of extra disks. This is done by a series of "cut" operations. A cut can either break a cycle, or break one substructure into two substructures. To capture how substructures interact, we define an auxiliary graph, call substructure relation graph  $\mathfrak{S}$ , in which each substructure is a node. The aforementioned R-correlations define a set of blue edges, and geometrically overlapping relations define a set of red edges. Though the cut operations, we can make blue edges form a matching, and red edges also form a matching, and  $\mathfrak{S}$  acyclic (we call  $\mathfrak{S}$  an acyclic 2-matching). The special structure of  $\mathfrak{S}$ allows us to define an ordering of all arcs easily. Together with some other simple properties, we can generalize the dynamic program from one substructure to all substructures simultaneously.

# 3 Square Gadgets

We discuss the structure of a square gadget  $\mathsf{Gg}(\Gamma)$  associated with the small square  $\Gamma$ . Recall that the square gadget  $\mathsf{Gg}(\Gamma) = \mathsf{D}_s \cup \mathsf{D}_t$ , where  $\mathsf{D}_s$  and  $\mathsf{D}_t$  are the furthest pair of disks in  $\Gamma$ . We can see that for any disk  $\mathsf{D}_i$  in  $\Gamma$ , there are either one or two arcs of  $\partial \mathsf{D}_i$  which are not covered by



Figure 1: A square gadget.  $D_s$  and  $D_t$  are the furthest pair of disks in square  $\Gamma$  whose centers are  $D_s$  and  $D_t$ . On the left hand side, the blue region is the central area  $\mathfrak{C} = \mathsf{D}(D_s, r_{st}) \cap \mathsf{D}(D_t, r_{st})$ , where  $r_{st} = |D_s D_t|$ . The brown region is the core-central area  $\mathfrak{C}_o = \mathsf{D}(P, 1) \cap \mathsf{D}(Q, 1)$ . On the right hand side, the green area is the active regions, defined as  $(\bigcup_{i \in \mathfrak{C}_o} \mathsf{D}_i - (\mathsf{D}_s \cup \mathsf{D}_t)) \cap H^+$  and  $(\bigcup_{i \in \mathfrak{C}_o} \mathsf{D}_i - (\mathsf{D}_s \cup \mathsf{D}_t)) \cap H^-$ .

 $\mathsf{Gg}(\Gamma)$ . Without loss of generality, assume that  $D_s D_t$  is horizontal. The line  $D_s D_t$  divides the whole plane into two half-planes which are denoted by  $H^+$  (the upper half-plane) and  $H^-$  (the lower half-plane).  $\partial \mathsf{D}_s$  and  $\partial \mathsf{D}_t$  intersect at two points P and Q. We need a few definitions which are useful throughout the paper. Figure 1 shows an example of a square gadget.

1. (Central Area and Core-Central Area) Define the *central area* of  $\mathsf{Gg}(\Gamma)$  as the intersection of the two disks  $\mathsf{D}(D_s, r_{st})$  and  $\mathsf{D}(D_t, r_{st})$  in the square  $\Gamma$ , where  $r_{st} = |D_s D_t|$ . We use  $\mathfrak{C}$  to denote it. Since  $\mathsf{D}_s$  and  $\mathsf{D}_t$  are the furthest pair, we can see that every other disk in  $\Gamma$  is centered in the central area  $\mathfrak{C}$ .

We define the *core-central area* of  $\mathsf{Gg}(\Gamma)$  is the intersection of two unit disks centered at P, Q respectively. Essentially, any unit disk centered in the core-central area has four intersections with the boundary of gadget. Let us denote the area by  $\mathfrak{C}_o$ .

2. (Active Region) Consider the regions

$$\left(\bigcup_{D_i \in \mathfrak{C}_o} \mathsf{D}_i - (\mathsf{D}_s \cup \mathsf{D}_t)\right) \cap H^+ \quad \text{and} \quad \left(\bigcup_{D_i \in \mathfrak{C}_o} \mathsf{D}_i - (\mathsf{D}_s \cup \mathsf{D}_t)\right) \cap H^-$$

We call each of them an *active region* associated with square  $\Gamma$ . An active region can be covered by disks centered in the core-central area. We use Ar to denote an active region.

### 4 Substructures

Initially,  $\mathcal{H}$  includes all square gadgets. In Section 7, we will include in  $\mathcal{H}$  a constant number of extra disks. For a set S of disk, we use  $\mathbb{R}(S)$  to denote the region covered by disks in S (i.e.,  $\cup_{\mathsf{D}_i \in S} \mathsf{D}_i$ ). Assuming a fixed  $\mathcal{H}$ , we now describe the basic structure of the uncovered region  $\mathbb{R}(\mathcal{D}') - \mathbb{R}(\mathcal{H})$ .

<sup>&</sup>lt;sup>4</sup> Recall that  $\mathcal{D}' = \mathcal{D} \setminus \mathcal{G}$  where  $\mathcal{G}$  is the *C* most expensive disks in OPT.



Figure 2: The general picture of the substructures in a block. The red points are the grid points of small squares. Dash green disks are what we have selected in  $\mathcal{H}$ . There are five substructures in the block.

For ease of notation, we use  $\mathbb{U}(\mathcal{H})$  to denote the uncovered region  $\mathbb{R}(\mathcal{D}') - \mathbb{R}(\mathcal{H})$ . Figure 2 shows an example. Intuitively, the region consists of several "strips" along the boundary of  $\mathcal{H}$ . Now, we define some notions to describe the structure of those strips.

- 1. (Arcs) Consider a disk  $D \in \mathcal{D}'$  and suppose the center of D is in the square  $\Gamma$ . Let  $D_sD_t$  be the square gadget  $\mathsf{Gg}(\Gamma)$ , and without loss of generality assume the line  $D_sD_t$  is horizontal and divides the plane into two halfplanes  $H^+$  and  $H^-$ . D may contribute at most two uncovered arcs, one in  $H^+$  and one in  $H^-$ . Let us first focus on  $H^+$ .  $\partial D$  intersects  $\partial \mathcal{H}$  at several points  ${}^5$  in  $H^+$ . The uncovered arc is the segment of  $\partial D$  starting from the first intersection point and ending at the last intersection point. <sup>6</sup> We can define the uncovered arc for  $H^-$  in the same way (if  $|\partial D \cap \partial \mathcal{H}| \neq 0$ ). Figure 3 illustrates why we need so many words to define an arc. Essentially, some portions of an arc may be covered by some other disks in  $\mathcal{H}$ , and the arc is broken into several pieces. Our definition says those pieces should be treated as a whole. In this paper, when we mention an arc, we mean an entire uncovered arc (w.r.t. the current  $\mathcal{H}$ ). Note that both endpoints of an arc lie on the boundary of  $\mathcal{H}$ .
- 2. (Subarcs) For an arc a, we use a[A, B] to denote the closed subarc of arc a from point A to point B. Similarly, we only write a(A, B) to denote the corresponding open subarc (with endpoints A and B excluded).
- 3. (Central Angle) Suppose arc *a* is part of  $\partial D$  for some disk D with center *D*. The central angle of *a*, denoted as  $\angle(a)$  is the angle whose apex (vertex) is *D* and both legs (sides) are the radii connecting *D* and the endpoints of *a*. We can show that  $\angle(a) < \pi$  for any arc *a* (See Lemma 39 in Appendix A.1)

<sup>&</sup>lt;sup>5</sup> The number must be even.

<sup>&</sup>lt;sup>6</sup> Note that an uncovered arc may not entirely lie in the uncovered region  $\mathbb{U}(\mathcal{H})$  (some portion may be covered by some disks in  $\mathcal{H}$ ).



Figure 3: The figure gives an example of an arc. The blue curves are part of the boundary of  $\mathcal{H}$ . The red curve is an uncovered arc.



Figure 4: A substructure. The baseline **b** consists of the red arcs which are the part of consecutive boundary of  $\partial \mathcal{H}$ .  $Q_s, Q_t$  are the endpoints of **b**. The black curves are are uncovered arcs. The bold black arcs form the envelope. The arc  $a \prec c$  because  $A \prec C$  and  $B \prec D$ .

- 4. (Baseline) We use  $\partial \mathcal{H}$  to denote to be the boundary of  $\mathcal{H}$ . Consider an arc *a* whose endpoints  $P_1, P_2$  are on  $\partial \mathcal{H}$ . We say the arc *a* cover a point  $P \in \partial \mathcal{H}$ , if *P* lies in the segment between  $P_1$  and  $P_2$  along  $\partial \mathcal{H}$ . We say a point  $P \in \partial \mathcal{H}$  can be covered if some arc in  $\mathcal{D}'$  covers *P*. A baseline is a consecutive maximal segment of  $\partial \mathcal{H}$  that can be covered. We usually use **b** to denote a baseline.
- 5. (Substructure) A substructure St(b, A) consists of a baseline b and the collection A of arcs which can cover some point in b. The two endpoints of each arc  $a \in A$  are on b and  $\angle(a)$  is less than  $\pi$ . Note that every point of b is covered by some arc in A. Figure 4 illustrates the components of an substructure.

Occasionally, we need a slightly generalized notion of substructure. For a set  $\mathcal{A}$  of uncovered arcs, if they cover a consecutive segment of the boundary of  $\mathcal{H}$ ,  $\mathcal{A}$  also induces a substructure denoted as  $St[\mathcal{A}]$ .

Arc Order: Now we switch our attention to the order of the arcs in a substructure St(b, A). Suppose the baseline **b** starts at point  $Q_s$  and ends up at point  $Q_t$ . Consider any two points  $P_1$  and  $P_2$  on the baseline **b**. If  $P_1$  is closer to  $Q_s$  than  $P_2$  along the baseline **b**, we say that  $P_1$  appears earlier than  $P_2$  (denoted as  $P_1 \prec P_2$ ). Consider any two arcs a and c in A. The endpoints of arc a are A and B, and the endpoints of arc c are C and D. All of points A, B, C, D are on the baseline **b**. Without loss of generality, we assume that  $A \prec B$ ,  $C \prec D$  and  $A \prec C$ . If  $B \prec D$ , We say arc a appears earlier than arc c (denoted as  $a \prec c$ ). Otherwise, we say a and c are incomparable. See Figure 4 for an example. It is easy to see that  $\prec$  defines a partial order.

**Adjacency:** Consider two arcs a (with endpoints  $A \prec B$ ) and c (with endpoints  $C \prec D$ ). If  $a \prec c$  and  $C \prec B$ , we say that a and b are *adjacent* (we can see that they must intersect exactly once), and

c is the *adjacent successor* of a. Similarly, we can define the adjacent successor of subarc  $a[P_1, P_2]$ . If c is the adjacent successor of a, meanwhile c intersects with subarc  $a[P_1, P_2]$ , we say that c is the *adjacent successor* of subarc  $a[P_1, P_2]$ . Among all adjacent successors of  $a[P_1, P_2]$ , we call the one whose intersection with  $a[P_1, P_2]$  is closest to  $P_1$  the *first adjacent successor* of  $a[P_1, P_2]$ .

In order to carry out the dynamic program in Section 6, we need to properly orient the baseline for each substructure so that the (partial) order or the arcs is well defined. Our final solution in each substructure can be represented as a path (which is a segment of the boundary of the union of chosen disks). Our dynamic program essentially needs to determine such a path. To be precise, we provide a formal definition of a *valid path*, as follows.

**Definition 7** (A Valid Path). Consider a substructure St(b, A). Suppose the baseline b is oriented from  $Q_s$  to  $Q_t$ . A valid path is a path from  $Q_s$  to  $Q_t$  which consists of a sequence of subarcs  $\{a_1[Q_s, Q_1], a_2[Q_1, Q_2], \ldots, a_k[Q_{k-1}, Q_t]\}$  (baseline segments are considered as subarcs as well). For any  $a_i$ ,  $a_{i+1}$  is its adjacent successor (so  $a_i \prec a_{i+1}$ ).  $Q_i$  is the intersection points of the arcs  $a_i$  and  $a_{i+1}$ .

Note that the baseline from  $Q_s$  to  $Q_t$  is a trivial valid path (we do not consider any coverage requirement yet). Among all the valid paths in a substructure, there is one that is maximal in terms of the coverage ability, which we call *the envelope* of the substructure.

**Definition 8** (Envelope of a Substructure). Consider a substructure St(b, A). Suppose the baseline b is oriented from  $Q_s$  to  $Q_t$ . The envelope of St is the valid path  $\{a_1[Q_s, Q_1], a_2[Q_1, Q_2], \ldots, a_k[Q_{k-1}, Q_t]\}$  where  $a_{i+1}$  is the first adjacent successor of  $a_i$  for all  $i \in [k]$ .

**Coverage:** Consider a substructure St(b, A). Consider an arc a with endpoints A and B on baseline b. We use b[A, B] to denote the segment of b that is covered by a. We say that the region surrounded by the arc a and b[A, B] is *covered* by arc a and use  $\mathbb{R}(a)$  to denote the region. We note that the covered region  $\mathbb{R}(a)$  is with respected to the current  $\mathcal{H}$ . Similarly, consider a valid path Path. The region covered by Path is  $\bigcup_{a \in Path} \mathbb{R}(a)$  (the union is over all arcs in Path) and is denoted by  $\mathbb{R}(Path)$ . Finally, we define the region covered by the substructure St, denoted by  $\mathbb{R}(St)$ , to be the region covered by the envelope of St.

### 5 Simplifying the Problem

The substructures may overlap in a variety of ways. As we mentioned in Section 2, we need to include in  $\mathcal{H}$  more disks in order to make the substructures amenable to the dynamic programming technique. However, this step is somewhat involved and we decide to postpone it to the end of the paper (Section 7). Instead, we present in this section what the organization of the substructures and what properties we need *after* including more disks in  $\mathcal{H}$  for the final dynamic program.

**Self-Intersections:** In a substructure St, suppose there are two arcs a and c in  $\mathcal{A}$  with endpoints A, B and C, D respectively. If  $A \prec B \prec C \prec D$  and a and c cover at least one and the same point in  $\mathcal{P}$ , we say the substructure is *self-intersecting*. In other words, there exists at least one point covered by two non-adjacent arcs in a self-intersecting substructure. See Figure 9 for an example. Self-intersections are troublesome obstacles for the dynamic programming approach. So we will eliminate all self-intersections in Section 7. In the rest of the section, we assume all substructures are *non-self-intersecting* and discuss their properties.

**Lemma 9** (Single Intersection Property). For any two arcs in a non-self-intersecting substructure, they have at most one intersection in the interior of the substructure.

*Proof.* We prove by contradiction. Suppose  $a_i$  and  $a_j$  belong to the same substructure.  $a_i$  and  $a_j$  intersect at point A and B. Since the substructure is non-self-intersecting,  $a_i$  and  $a_j$  lie on the same side of line A and B. Because the two radii of  $a_i$  and  $a_j$  are equivalent, the sum of central angles of  $a_i$  and  $a_j$  equals  $2\pi$ . Thus at least one central angle of  $a_i$  and  $a_j$  is no less than  $\pi$ , rendering a contradiction to the fact that the central angle (defined in Sec. 4) of any arc is less than  $\pi$ .

Base on the single intersection property, we can easily get the following property.

**Lemma 10.** In a non-self-intersecting substructure, if a point is covered by two arcs a and b, a is adjacent to b.

**Order Consistency:** There are two types of relations between substructures which affect how the orientations should be done. One is the *overlapping relation* and the other is *remote-correlation*. See Figure 2 for some examples.

As we alluded in Section 2, the two substructures which contain different related active regions of the same gadget interact with each other, i.e.,

**Definition 11** (Remote correlation). Consider two substructures  $St_u$  and  $St_l$  which are not overlapping. They contain two different active regions of the same gadget respectively (recalling that one gadget may have two different active regions, one in  $H^+$ , one in  $H^-$ ). We say that they are remotely correlated or R-correlated. See Figure 2.

There are two possible baseline orientations for each substructure (clockwise or anticlockwise around the center of the arc), which gives rise to four possible ways to orient both  $St_u$  and  $St_l$ . However, there are only two (out of four) of them are consistent (thus we can do dynamic programming on them). More formally, we need the following definition:

**Definition 12** (Local Order Consistency). Consider two substructures  $\mathsf{St}_u(\mathsf{b}_u, \mathcal{A}_u)$  and  $\mathsf{St}_l(\mathsf{b}_l, \mathcal{A}_l)$ . There is an orientation for each substructure, such that the partial orders of the disks for both substructures are consistent in the following sense: It can not happen that  $a_i \prec b_i$  in substructure  $\mathsf{St}_u$  but  $a_j \prec b_j$  in  $\mathsf{St}_l$ , where  $a_i, a_j \in \partial \mathsf{D}_a, b_i, b_j \in \partial \mathsf{D}_b$  and  $a_i, b_i \in \mathcal{A}_u, a_j, b_j \in \mathcal{A}_l$ .

We show in the following simple lemma that the local order consistency can be easily achieved for the two substructures containing different active regions of the same gadget.

**Lemma 13** (Local Order Consistency). Consider two substructures  $St_u$  and  $St_l$  which are *R*-correlated. Each of them contains only one active region. Then the local order consistency holds for the two substructures  $St_u$  and  $St_l$ .

*Proof.* We consider two substructures  $St_u$  and  $St_l$ . The arcs not in the active regions have no influence on the order consistency since each of them only appears in one substructure. So, we only need to consider the order of the arcs in two active regions. We use the same notations as those on the RHS of Figure 1. We orient the upper baseline from  $Q_s$  to  $Q_t$ , and the lower baseline from  $P_s$  to  $P_t$ . Suppose two arcs  $c_1, c_3$  belong to disk  $D_u$ , and two arcs  $c_2, c_4$  belong to the disk  $D_v$ . Assume  $c_4 \prec c_3$  in substructure  $St_u$  and  $c_1 \prec c_2$  in substructure  $St_l$ . There must exist another intersection point on each side of the line connecting the two intersections of  $(c_1, c_2), (c_3, c_4)$ . This contradicts the fact that two unit disks have at most two intersections.

Then, we discuss the situation where two substructures overlap. We first need a few notations. For an arc a, we use D(a) to denote the disk associated with a. For a substructure St(b, A), we let D(b) be the set of disks that contributes an arc to the baseline b.

**Definition 14** (Overlapping Relation). Consider two substructures  $St_1(b_1, A_1)$  and  $St_2(b_2, A_2)$  and the point set  $\mathcal{P}$ . We say  $St_1$  and  $St_2$  overlap when there are two arcs  $a \in A_1$  and  $b \in A_2$  such that both a and b can cover at least one and the same point of  $\mathcal{P}$ .

The dynamic program need the following property.

**Proposition 15.** The orientations of the two overlapping substructures should be different (i.e., if one is clockwise, the other should be anticlockwise).

As different substructures may interact with each other, we need a dynamic program which can run over all substructures simultaneously. Hence, we need to define a globally consistent ordering of all arcs.

**Definition 16** (Global Order Consistency). We have global order consistency if there is a way to orient the baseline of each substructure, such that the partial orders of the disks for all substructures are consistent in the following sense: It can not happen that  $a_i \prec b_i$  in substructure  $\mathsf{St}_i(\mathsf{b}_i, \mathcal{A}_i)$  but  $a_j \prec b_j$  in  $\mathsf{St}_j(\mathsf{b}_j, \mathcal{A}_j)$ , where  $a_i, a_j \in \partial \mathsf{D}_a, b_i, b_j \in \partial \mathsf{D}_b$  and  $a_i, b_i \in \mathcal{A}_i, a_j, b_j \in \mathcal{A}_j$ .

**Substructure Relation Graph**  $\mathfrak{S}$ : we construct an auxiliary graph  $\mathfrak{S}$ , called the *substructure* relation graph, to capture all R-correlations and overlapping relations. Each node in  $\mathfrak{S}$  represents a substructure. If two substructures are R-correlated, we add a blue edge between the two substructures. If two substructures overlap, we add a red edge.

Consider a red edge between  $St_1(b_1, A_1)$  and  $St_2(b_2, A_2)$ . If baseline  $b_1$  is oriented clockwise (around the center of any of its arc), then  $b_2$  should be oriented counterclockwise, and vise versa. The blue edge represents the same orientation relation, i.e., if  $St_1(b_1, A_1)$  and  $St_2(b_2, A_2)$  are Rcorrelated,  $b_1$  and  $b_2$  should be oriented differently.

It is unclear how to orient all baselines if  $\mathfrak{S}$  is an arbitrary graph. So we need to ensure that  $\mathfrak{S}$  has a nice structure.

**Definition 17** (Acyclic 2-Matching). We say the substructure relation graph  $\mathfrak{S}$  is an acyclic 2-matching, if  $\mathfrak{S}$  is acyclic and is composed by a blue matching and a red matching. In other words,  $\mathfrak{S}$  contains only paths, and the red edges and blue edges appear alternately in each path.

If  $\mathfrak{S}$  is a acyclic 2-matching, we can easily assign each substructure a global arc order consistent orientations.

**Point-Order Consistency:** Similarly to the arc order consistency, we also need define the *point-order consistency*, which is also crucial for our dynamic program.

**Definition 18** (Point Order Consistency). Suppose a set  $\mathcal{P}_{co}$  of points is covered by both of two overlapping substructures  $\mathsf{St}_1(\mathsf{b}_1, \mathcal{A}_1)$  and  $\mathsf{St}_2(\mathsf{b}_2, \mathcal{A}_2)$ . Consider any two points  $P_1, P_2 \in \mathcal{P}_{co}$  and four arcs  $a_1, a_2 \in \mathcal{A}_1, b_1, b_2 \in \mathcal{A}_2$ . Suppose  $P_1 \in \mathbb{R}(a_1) \cap \mathbb{R}(b_1)$  and  $P_2 \in \mathbb{R}(a_2) \cap \mathbb{R}(b_2)$ . But  $P_1 \notin \mathbb{R}(a_2) \cup \mathbb{R}(b_2)$  and  $P_2 \notin \mathbb{R}(a_1) \cup \mathbb{R}(b_1)$ . We say  $P_1$  and  $P_2$  are point-order consistent if  $a_1 \prec a_2$  in  $\mathsf{St}_1$  and  $b_1 \prec b_2$  in  $\mathsf{St}_2$ . We say the points in  $\mathcal{P}_{co}$  satisfy point order consistency if all pairs of points in  $\mathcal{P}_{co}$  are point-order consistent.

After introducing all relevant concepts, we can finally state the set of properties we need for the dynamic program.

**Lemma 19.** After choosing  $\mathcal{H}$ , we can ensure the following properties holds:

- P1. (Active Region Uniqueness) Each substructure contains at most one active region.
- P2. (Non-self-intersection) Every substructure is non-self-intersecting.
- P3. (Acyclic 2-Matching) The substructure relation graph  $\mathfrak{S}$  is an acyclic 2-matching, i.e.,  $\mathfrak{S}$  consists of only paths. In each path, red edges and blue edges appear alternately.
- P4. (Point Order Consistency) Any point is covered by at most two substructures. The points satisfy the point order consistency.

How to ensure all these properties will be discussed in detail in Section 7. Now, everything is in place to describe the dynamic program.

## 6 Dynamic Programming

Suppose we have already constructed the set  $\mathcal{H}$  such that Lemma 19 holds (along with an orientation for each substructure). Without loss of generality, we can assume that the remaining disks can cover all remaining points (otherwise, either the original instance is infeasible or our guess is wrong). In fact, our dynamic program is inspired, and somewhat similar to those in [2, 20, 27].

**DP for Two Overlapping Substructures:** For ease of description, we first handle the case where there are only two overlapping substructures. We will extend the DP to the general case shortly. Suppose the two substructures are  $St_1(b_1, A_1)$  and  $St_2(b_2, A_2)$ ,  $b_1$  is oriented from  $P_s$  to  $P_t$  and  $b_2$  is oriented from  $Q_s$  to  $Q_t$ 

A state of the dynamic program is a pair  $\Phi = (P, Q)$  where P is an intersection point of two arcs in substructure  $\mathsf{St}_1$  and Q is an intersection point of two arcs in substructure  $\mathsf{St}_2$ . Fix the state  $\Phi = (P, Q)$  and consider  $\mathsf{St}_1$ . Let  $b_P$  and  $t_P$  be the two arcs intersecting at P. Suppose  $b_P \prec t_P$ with endpoints (A, B), (C, D) respectively. We call arc  $b_P$  the *base-arc* and  $t_P$  the *top-arc* for point P. <sup>7</sup> Our DP maintains that the base-arc is always paid in the subproblem, and intermediate state.

Given state  $\Phi = (P, Q)$ , now we describe the subproblem associated with the state. Again first focus on St<sub>1</sub>. Intuitively, a feasible solution to the subproblem restricted to St<sub>1</sub> (resp. St<sub>2</sub>) is a valid path starting from point P (resp. Q) and terminating at P<sub>t</sub> (resp. Q<sub>t</sub>). More specifically, we construct a substructure St<sub>1</sub><sup>[P]</sup>(b<sub>1</sub>[P],  $A_1[P]$ ) (See Figure 5):

- $b_1[P]$  is the concatenation of subarc  $b_P[P, B]$  and the original baseline segment  $b_1[B, P_t]$ . All arcs in  $b_1[P]$  have cost zero.
- $\mathcal{A}_1[P]$  consists of all arcs  $a' \in \mathcal{A}_1$  such that  $b_P \prec a'$  (of course, with the portion covered by  $\mathbf{b}_1[P]$  subtracted). The cost each such arc is the same as its original cost.

<sup>&</sup>lt;sup>7</sup> If P is the tail endpoint of an arc (so P is on the baseline), P only has a base-arc (no top-arc), which is the baseline arc it lies on.



Figure 5: The figure explains the dynamic program of two overlapping substructures. The left figure shows the subproblem  $\mathsf{OPT}(P,Q)$ . The goal of  $\mathsf{OPT}(P,Q)$  is to find minimum valid paths for  $PP_t$  and  $QQ_t$  respectively in set  $\mathcal{A}_1[P] \cup \mathcal{A}_2[Q]$  such that the paths cover all points of  $\mathcal{P}[P,Q]$ . The right figure illustrates one of its four smaller subproblems  $\mathsf{OPT}(P^t,Q)$ .

Similarly, we consider  $\mathsf{St}_2$  and the intersection point Q, and construct  $\mathsf{St}_2[Q]$  with baseline  $\mathsf{b}_2[Q]$ and arc set  $\mathcal{A}_2[Q]$ . We use  $\mathcal{P}(a)$  (or  $\mathcal{P}(\mathcal{A})$ ) to denote the points covered by a (or  $\mathcal{A}$ ) (w.r.t. the original baseline). Let the point set  $\mathcal{P}[P,Q]$  that we need to cover in the subproblem  $\Phi(P,Q)$  be

$$\mathcal{P}[P,Q] = \mathcal{P}(\mathcal{A}_1[P]) + \mathcal{P}(\mathcal{A}_2[Q]) - \mathcal{P}(b_P) - \mathcal{P}(b_Q).$$
(1)

We note that the minus term  $-\mathcal{P}(b_P) - \mathcal{P}(b_Q)$  is not vacuous as  $b_P$  (resp.  $b_Q$ ) may cover some points in  $\mathcal{A}_2[Q]$  (resp.  $\mathcal{A}_1[P]$ ), and it is important that we do not have to cover those point (this subtlety is crucial in the correctness proof of the DP). The goal for the subproblem  $\Phi(P,Q)$  is to find two valid paths with minimum total weight, one from P to  $P_t$  and one from Q to  $Q_t$ , such that they together cover all points in  $\mathcal{P}[P,Q]$ . Note that the weights of both base-arcs  $b_P$  and  $b_Q$ should be included in any feasible solution as well.

Suppose  $b_P(P, B]$  intersects its first successor at  $P^b$  (called *base-adjacent point*) and  $t_P(P, D]$  intersects its first successor at  $P^t$  (called *top-adjacent point*). Similarly, we define  $Q^b$ ,  $Q^t$  in  $St_2$  in exactly the same way.

Now, computing the optimum for subproblem  $\Phi(P,Q)$  reduces to computing the optima for four smaller subproblems  $\mathsf{OPT}(P^b,Q)$ ,  $\mathsf{OPT}(P^t,Q)$ ,  $\mathsf{OPT}(P,Q^b)$  and  $\mathsf{OPT}(P,Q^t)$ . We define two Boolean variables  $I_P$  (reps.  $I_Q$ ) indicating whether we can move from  $\mathcal{P}_{P^b,Q}$  to  $\mathcal{P}_{P,Q}$  without choosing a new arc. Formally, if  $\mathcal{P}_{P,Q} = \mathcal{P}_{P^b,Q}$ ,  $I_P = 0$ . Otherwise,  $I_P = 1$ . Similarly, if  $\mathcal{P}_{P,Q} = \mathcal{P}_{P,Q^b}$ ,  $I_Q = 0$ . If not,  $I_Q = 1$ . The dynamic programming recursion is:

$$\mathsf{OPT}(P,Q) = \min \begin{cases} \mathsf{OPT}(P,Q^b) + I_Q \cdot \infty, & \text{add no new arc };\\ \mathsf{OPT}(P,Q^t) + w[b_Q], & \text{add base-arc } b_Q;\\ \mathsf{OPT}(P^b,Q) + I_P \cdot \infty, & \text{add no new arc };\\ \mathsf{OPT}(P^t,Q) + w[b_P], & \text{add base-arc } b_P. \end{cases}$$
(2)

The optimal value we return is  $OPT(P_s, Q_s)$ . Now, we prove the correctness of the DP in the following theorem. We note that both the point-order consistency and the subtlety mentioned above play important roles in the proof.

**Theorem 20.** Suppose that we have two overlapping substructures  $St_1(b_1, A_1)$  and  $St_2(b_2, A_2)$ . Further suppose that  $b_1$  and  $b_2$  are oriented in a way such that the point-order consistency holds. Then, the cost of the optimal solution equals to  $OPT(P_s, Q_s)$  (which is computed by (2)).

*Proof.* Consider subproblem  $\mathsf{OPT}(P,Q)$ . As we know the optimal solution of  $\mathsf{OPT}(P,Q)$  should be two valid paths. One is from P to  $P_t$  and the other is from Q to  $Q_t$ . Suppose they are  $\mathsf{Path}_1 = \{a_1[P, P_1], a_2[P_1, P_2], \ldots, a_i[P_{i-1}, P_i], \ldots, a_k[P_{k-1}, P_t]\}$  and  $\mathsf{Path}_2 = \{b_1[Q, Q_1], b_2[Q_1, Q_2], \ldots, b_i[Q_{i-1}, Q_i], \ldots, b_i[Q_{l-1}, Q_t]\}$ . We can see that it suffices to prove that at least one of the two statements is true.

- The pair of paths  $(\mathsf{Path}_1 \{a_1[P, P_1)\}, \mathsf{Path}_2)$  is the optimal solution to subproblem  $\mathsf{OPT}(P_1, Q)$ .
- The pair of paths ( $\mathsf{Path}_1, \mathsf{Path}_2 \{b_1[Q, Q_1)\}$ ) is the optimal solution to subproblem  $\mathsf{OPT}(P, Q_1)$ .

We prove by contradiction. Assume that both of the above statements are wrong. Suppose  $b_P$  and  $b_Q$  are the base-arcs for state  $\Phi(P,Q)$ , i.e.,  $b_P$  intersects with  $a_1$  at point P and  $b_Q$  intersects with  $b_1$  at point Q. We use  $\mathcal{P}(\mathsf{Path})$  to denote the point set which is covered by path Path. Recall that  $\mathcal{P}[P,Q] = \mathcal{P}(\mathcal{A}_1[P]) + \mathcal{P}(\mathcal{A}_2[Q]) - \mathcal{P}(b_P) - \mathcal{P}(b_Q)$ . Since  $\mathsf{Path}_1 \cup \mathsf{Path}_2$  is the optimal solution for  $\mathsf{OPT}(P,Q)$  (hence feasible), we have that  $\mathcal{P}[P,Q] = \mathcal{P}(\mathsf{Path}_1) + \mathcal{P}(\mathsf{Path}_2) - \mathcal{P}(b_P) - \mathcal{P}(b_Q)$ . Then, the pair of paths  $(\mathsf{Path}_1 - \{a_1[P, P_1]\}, \mathsf{Path}_2)$  is the optimal solution for the subproblem in which we need to pick one path from  $P_1$  to  $P_t$  and one from Q to  $Q_t$  to cover the points in

$$\mathcal{P}(\mathsf{Path}_1) + \mathcal{P}(\mathsf{Path}_2) - \mathcal{P}(b_P) - \mathcal{P}(b_Q) - \mathcal{P}(a_1) = \mathcal{P}[P,Q] - \mathcal{P}(a_1).$$

If not, we can get a contradiction by replacing  $\mathsf{Path}_1 - a_1[P, P_1]$  and  $\mathsf{Path}_2$  with the optimal solution of the above subproblem, resulting in a solution with less weight than  $\mathsf{OPT}(P,Q)$ . Since the first statement is wrong, we must have that

$$\mathcal{P}[P,Q] - \mathcal{P}(a_1) \neq \mathcal{P}[P_1,Q]$$

(otherwise  $(\mathsf{Path}_1 - \{a_1[P, P_1)\}, \mathsf{Path}_2)$  is optimal for  $\Phi(P_1, Q)$ ). We note that the LHS  $\subseteq$  RHS. Plugging the definition (1), we have that

$$\mathcal{P}(\mathcal{A}_1[P]) + \mathcal{P}(\mathcal{A}_2[Q]) - \mathcal{P}(b_P) - \mathcal{P}(b_Q) - \mathcal{P}(a_1) \neq \mathcal{P}(\mathcal{A}_1[P_1]) + \mathcal{P}(\mathcal{A}_2[Q]) - \mathcal{P}(a_1) - \mathcal{P}(b_Q).$$

A careful (elementwise) examination of the above inequality shows that it is only possible if

$$\mathcal{P}(b_P) \cap (\mathcal{P}(\mathcal{A}_2[Q]) - \mathcal{P}(b_Q)) \neq \emptyset.$$

Repeating same argument, we can see that if the second statement is wrong, we have that

$$\mathcal{P}(b_Q) \cap (\mathcal{P}(\mathcal{A}_1[P]) - \mathcal{P}(b_P)) \neq \emptyset.$$

Hence, there exist  $b_i \in \mathcal{A}_2[Q]$  and  $a_j \in \mathcal{A}_1[P]$  such that  $\mathcal{P}(b_P) \cap \mathcal{P}(b_i) \neq \emptyset$ , and  $\mathcal{P}(b_Q) \cap \mathcal{P}(a_j) \neq \emptyset$ . However, this contradicts the point-order consistency because of  $b_P \prec a_j$  and  $b_Q \prec b_i$ .

Thus, one of the two statements is true. W.l.o.g, suppose  $(\mathsf{Path}_1 - \{a_1[P, P_1]\}, \mathsf{Path}_2)$  is the optimal solution to subproblem  $\mathsf{OPT}(P_1, Q)$ . If  $P_1$  is the top-adjacent point of P, through  $\mathsf{OPT}(P,Q) = \mathsf{OPT}(P_1,Q) + w[b_P]$  in  $\mathsf{DP}(2)$ , we can get the optimal solution of subproblem  $\mathsf{OPT}(P,Q)$ . If not, suppose  $P^t, Q^t$  is the top-adjacent point of P, Q, using the same argument, we can prove that  $(\mathsf{Path}_1, \mathsf{Path}_2)$  is the optimal solution to at least one of the subproblems  $\mathsf{OPT}(P,Q^t)$ or  $\mathsf{OPT}(P^t,Q)$ . Thus, we can get the optimal solution for  $\mathsf{OPT}(P,Q)$  by our DP.

**DP for the general problem:** Then, we handle all substructures together. Our goal is find a valid path for each substructure such that minimizing the total weight of all the paths. We can see that we only need to handle each path in the substructure relation graph  $\mathfrak{S}$  separately (since different paths have no interaction at all). Hence, from now on, we simply assume that  $\mathfrak{S}$  is a path.

Suppose the substructures are  $\{\mathsf{St}_k(\mathsf{b}_k, \mathcal{A}_k)\}_{k \in [m]}$ . We use  $A_k$  and  $B_k$  to denote two endpoints of  $\mathsf{b}_k$ . Generalizing the previous section, a state for the general DP is  $\Phi = \{P_k\}_{k \in [m]}$ , where  $P_k$  is an intersection point in substructure  $\mathsf{St}_k$ . We use  $b_{P_k}$ ,  $t_{P_k}$ ,  $P_k^b$ ,  $P_k^t$  to denote the base-arc, top-arc, base-adjacent point, top-adjacent point (w.r.t.  $P_k$ ) respectively. For each  $k \in [m]$ , we also define  $\mathsf{St}_k^{[P_k]}(\mathsf{b}_k[P_k], \mathcal{A}_k[P_k])$  in exactly the same way as in the previous section. Let  $\mathcal{P}\left[\{P_k\}_{k \in [m]}\right]$  be the point set we need to cover in the subproblem:

$$\mathcal{P}\left[\{P_k\}_{k\in[m]}\right] = \bigcup_{k\in[m]} \mathcal{P}(\mathcal{A}_k[P_k]) - \bigcup_{k\in[m]} \mathcal{P}(b_{P_k}).$$

The subproblem  $\mathsf{OPT}(\{P_k\}_{k\in[m]})$  is to find, for each substructure  $\mathsf{St}_k$ , a valid path from  $P_k$  to  $B_k$ , such that all points in  $\mathcal{P}[\{P_k\}_{k\in[m]}]$  can be covered and the total cost is minimized.

The additional challenge for the general case is caused by R-correlations. If two arcs (in two different substructures) belong to the same disk, we say that they are *siblings* of each other. If we processed each substructure independently, some disks would be counted twice. In order to avoid double-counting, we should consider both siblings together, i.e., select them together and pay the disk only once in the DP.

In order to implement the above idea, we need a few more notations. We construct an auxiliary bipartite graph  $\mathfrak{B}$ . The nodes on one side are all disks in  $\mathcal{D}' \setminus \mathcal{H}$ , and the nodes on the other side are substructures. If disk  $\mathsf{D}_i$  has an arc in the substructure  $\mathsf{St}_j$ , we add an edge between  $\mathsf{D}_i$  and  $\mathsf{St}_j$ . Besides, for each arc of baselines, we add a node to represent it and add an edge between the node and the substructure which contains the arc. Because the weight of any arc of baselines is zero, it shall not induce contradiction that regard them as independent arcs. In fact, there is a 1-1 mapping between the edges in  $\mathfrak{B}$  and all arcs. See Figure 6 for an example.

Fix a state  $\Phi = \{P_k\}_{k \in [m]}$ . For any arc *a* in  $\mathsf{St}_k$  (with intersection point  $P_k$  and base-arc  $b_{Pk}$ ), *a* has three possible positions:

- 1.  $a \prec b_{Pk}$ : we label its corresponding edge with "unprocessed";
- 2.  $a = b_{Pk}$ : we label its corresponding edge with "processing";
- 3. Others: we label its corresponding edge with "done".

As mentioned before, we need to avoid the situation where one arc becomes the base-arc first (i.e., being added in solution and paid once), and its sibling becomes the base-arc in a later step (hence being paid twice). With the above labeling, we can see that all we need to do is to avoid the states in which one arc is "processing" and its sibling is "unprocessed". If disk D is incident on at least one "processing" edge and not incident on any "unprocessed" edge, we say the D is *ready*. Let  $\mathcal{R}$  be the set of ready disks. For each ready disk D, we use  $N_{\rm p}(D)$  to denote the set of neighbors (i.e., substructures) of D connected by "processing" edges. We should consider all substructures in  $N_{\rm p}(D)$  together.

Again, we need in our DP indicator variables to tell us whether a certain transition is feasible: Formally, if  $\mathcal{P}[\{P_k\}_{k\in[m]}] = \mathcal{P}[[\mathcal{P}_k][P_i^b]_{\{i\}}]$ , let  $I_i = 0$ . Otherwise, let  $I_i = 1$ . For ease of notation,



Figure 6: The bipartite graph which is used for marking the *ready* disks. The nodes on upper side represent the disks. The nodes on the lower side represent the substructures. If  $D_i$  has an arc in  $St_i$ , we add an arc between them.

for a set  $\{e_k\}_{k\in[m]}$  and  $S\subseteq[m]$ , we write  $[e_k][e'_i]_S=\{e_k\}_{k\in[m]\setminus S}\cup\{e'_i\}_{i\in S}$ . Hence,

$$[P_k][P_i^b]_{\{i\}} = \{P_k\}_{k \in [m] \setminus i} \cup P_i^b \text{ and}$$
$$[P_k][P_i^t]_{N_{\rm p}({\rm D})} = \{P_k\}_{k \in [m] \setminus N_{\rm p}({\rm D})} \cup \{P_i^t\}_{i \in N_{\rm p}({\rm D})}$$

Then we have the dynamic program as follows:

$$\mathsf{OPT}\left(\{P_k\}_{k\in[m]}\right) = \min\left\{\begin{array}{l}\min_{i\in[m]}\left\{\mathsf{OPT}\left([P_k][P_i^b]_{\{i\}}\right) + I_i \cdot \infty\right\}, & \text{add no disk}\\ \min_{\mathsf{D}\in\mathcal{R}}\left\{\mathsf{OPT}\left([P_k][P_i^t]_{N_{\mathsf{P}}(\mathsf{D})}\right) + w_{\mathsf{D}}\right\}, & \text{add disk }\mathsf{D}\end{array}\right.$$
(3)

Note that in the second line, the arc(s) in  $N_p(\mathsf{D})$  are base-arcs (w.r.t. state  $\Phi(\{P_k\}_{k\in[m]})$ ).

In the rest of the section, we prove the correctness of the dynamic program. If we use the solution of a smaller subproblem  $\mathsf{OPT}(\Phi')$  to compute subproblem  $\mathsf{OPT}(\Phi)$ , we say  $\Phi$  can be reached from  $\Phi'$  (denoted as  $\Phi' \to \Phi$ ). If  $\Phi$  can be reached from initial state  $\Phi_0(\{P_k \mid P_k = B_k\}_{k \in [m]})$ , we say the state is reachable, which is denoted by  $\Phi_0 \to \Phi$ .

We start with a simple consequence of our DP: there is no double-counting.

**Lemma 21.** If two arcs in the solution belong to the same disk, their weights are counted only once in the DP(3).

*Proof.* From the DP, we can see that the weight of an arc is counted only when it becomes a basearc (its label changes from "unprocessed" to "processing"). If the two sibling arcs a, b (belonging to disk D) in the solution are counted twice, there exist two states  $\Phi_1$  and  $\Phi_2$  such that  $(1)\Phi_1 \rightarrow \Phi_2$ , (2)a is a base-arc in  $\Phi_1$ , but b is not a base-arc in  $\Phi_1$ , (3) b is a base-arc in  $\Phi_2$ . So, in  $\Phi_1$  or any state before that, arc b is "unprocessed". However, a can become a base-arc only when D is ready, rendering a contradiction.

Now, we prove the correctness of the dynamic program. The proof is a generalization of Theorem 20. **Theorem 22.** Suppose  $\mathfrak{S}$  is a path. All baselines are oriented such that all properties in Lemma 19 hold. Then, the optimal cost for the problem equals to  $\mathsf{OPT}(\{A_k\}_{k\in[m]})$  (computed by our DP (3)).

Proof. Suppose the set  $\{\mathsf{Path}_k\}_{k\in[m]}$  of paths is the optimal solution. We need to prove (1) the final state  $\Phi(\{A_k\}_{k\in[m]})$  is reachable, (2) for any reachable state  $\Phi = \{P_k\}_{k\in[m]}$ ,  $\mathsf{OPT}(\{P_k\}_{k\in[m]})$  computes the optimal solution for the corresponding subproblem (that is to find one valid path from  $P_k$  to  $B_k$  for each substructure  $\mathsf{St}_k^{[P_k]}$  to cover all point in  $\mathcal{P}[\{P_k\}_{k\in[m]}]$ , such that the total cost is minimized).

We first prove the first statement. Suppose the state  $\Phi$  is reachable, we prove it can reach another state if  $\Phi$  is not the final state (i.e., we do not get stuck at  $\Phi$ ). Let us prove it by contradiction. Assume we get stuck at state  $\Phi$ . That means there is no ready disk in  $\Phi$ . Note that each substructure, say St, is incident on exactly one "processing" arc, say arc *a* (which is the base-arc in St). *a*'s sibling, say *b* (in St'), must be labeled "unprocessed" (otherwise the disk would be ready). Consider the base-arc (or "processing" arc), say *a*', in St'. So we have  $a' \prec b$  in St'. Again the sibling *b*' of *a*' must be an "unprocessed" arc in St. <sup>8</sup> So we have  $b' \prec a$  in St, which contradicts the global arc-order consistency.

Now, we prove the second statement. We consider state  $\Phi = \{P_k\}_{k \in [m]}$ . Suppose the optimal solution for subproblem  $\Phi\{P_k\}_{k \in [m]}$  is the set of paths  $\mathsf{PS} = \{\mathsf{Path}_k\}_{k \in [m]}$  where  $\mathsf{Path}_k = (a_{k_1}, a_{k_2}, \ldots, a_{k_n})$ . Consider the states  $\Phi_{N_p(\mathsf{D})} := [P_k][P_i^t]_{N_p(\mathsf{D})}$  for all  $\mathsf{D} \in \mathcal{R}$ . Obviously, we can see from our DP that  $\Phi_{N_p(\mathsf{D})} \to \Phi$ . Define for each  $\mathsf{D} \in \mathcal{R}$ , a set of paths

$$\mathsf{PS}_{N_{\mathsf{P}}(\mathsf{D})} = \{\mathsf{Path}_k\}_{k \in [m] - N_{\mathsf{P}}(\mathsf{D})} \cup \{\mathsf{Path}_i - \{a_{i_1}\}\}_{i \in N_{\mathsf{P}}(\mathsf{D})}.$$

It suffices to prove that there exists at least one  $\mathsf{D} \in \mathcal{R}$  such that  $\mathsf{PS}_{N_{\mathsf{p}}(\mathsf{D})}$  is the optimal solution for  $\mathsf{OPT}(\Phi_{N_{\mathsf{p}}(\mathsf{D})})$ .

Consider a substructure  $\mathsf{St}_i$ . Suppose the intersection point in  $\mathsf{St}_i$  of  $\Phi$  is  $P_i$  and the base-arc at point  $P_i$  is  $b_{P_i}$ . For each  $i \in [m]$ , let  $\mathsf{St}_{\pi(i)}$  be the only substructure (if any) overlapping with  $\mathsf{St}_i$ . So  $b_{P\pi(i)}$  is the base-arc of  $\mathsf{St}_{\pi(i)}$ . Using exactly the same exchange argument in Theorem 20, we can show that if  $\mathsf{PS}_{N_p(\mathsf{D})}$  is not the optimal solution for  $\mathsf{OPT}(\Phi_{N_p(\mathsf{D})})$ , there exists some  $i \in N_p(\mathsf{D})$  such that  $\mathcal{E}_i$  happens, where  $\mathcal{E}_i$  is the following event: there exists an arc  $\beta_{\pi(i)}$  in  $\mathsf{St}_{\pi(i)}$  with  $\beta_{\pi(i)} \succ b_{P\pi(i)}$ such that

$$\mathcal{P}(b_{P_i}) \cap \mathcal{P}(\beta_{\pi(i)}) \neq \emptyset.$$

We use  $\mathcal{E}_i$  to denote the above event. Thus if there is no  $\mathsf{D} \in \mathcal{R}$  such that  $\mathsf{PS}_{N_p(\mathsf{D})}$  is the optimal solution for  $\mathsf{OPT}(\Phi_{N_p(\mathsf{D})})$ , we have

$$\bigwedge_{N_{\rm p}({\rm D})\in\mathcal{R}} \left(\bigvee_{i\in N_{\rm p}({\rm D})} \mathcal{E}_i\right) = \text{True.}$$
(4)

Converting the conjunctive normal form (CNF) to the disjunctive normal form (DNF), we get

$$\bigvee_{(k_1,\ldots,k_{|\mathcal{R}|})\in\Pi_{\mathsf{D}\in\mathcal{R}}N_{\mathsf{P}}(\mathsf{D})}\left(\bigwedge_{i\in|\mathcal{R}|}\mathcal{E}_{k_i}\right) = \mathrm{True},$$

<sup>&</sup>lt;sup>8</sup>To see that b' is in St, note that  $\mathfrak{S}$  is a path and St is only R-correlated with St' (and vice versa).

where  $\Pi_{\mathsf{D}\in\mathcal{R}}N_{\mathsf{p}}(\mathsf{D})$  means the Cartesian product of all  $N_{\mathsf{p}}(\mathsf{D})$  in  $\mathcal{R}$ . We call each  $\bigwedge_{i\in|\mathcal{R}|}\mathcal{E}_{k_i}$  a clause (note that  $k_i$  indexes a substructure). If we can prove that every clause is false, then obviously, (4) is wrong, resulting in a contradiction.

Now, we show that every clause is false. First, we consider the case that both end nodes of  $\mathfrak{S}$  are incident to red edges. W.l.o.g., suppose the two nodes of  $\mathfrak{S}$  are  $\mathsf{St}_1$  and  $\mathsf{St}_2$ . Thus, they are R-correlated with other substructures. We know if one substructure  $\mathsf{St}_i$  is not R-correlated with others, every clause must contain the corresponding event  $\mathcal{E}_i$  (since the disk corresponding to its base-arc must be ready and in  $\mathcal{R}$ ). Hence, every clause contains  $\mathcal{E}_1$  and  $\mathcal{E}_2$ . Moreover, for each pair ( $\mathsf{St}_i, \mathsf{St}_j$ ) of R-correlated substructures, each clause should contain either  $\mathcal{E}_i$  or  $\mathcal{E}_j$ . Suppose the length of the path  $\mathfrak{S}$  is  $\ell$ . Because red and blue edges alternates, there are  $\frac{\ell-1}{2}$  R-correlated substructure pairs and  $\frac{\ell+1}{2}$  overlapping substructure pairs. We should select  $\frac{\ell-1}{2} + 2$  terms in each clause. Because  $\frac{\ell-1}{2} + 2 > \frac{\ell+1}{2}$ , there exists a pair of overlapping substructures ( $\mathsf{St}_i, \mathsf{St}_{\pi(i)}$ ) such that both  $\mathcal{E}_i$  and  $\mathcal{E}_{\pi(i)}$  appear in the clause. To make the clause true, we must have

$$\mathcal{E}_i = (\mathcal{P}(b_{P_i}) \cap \beta_{\pi(i)} \neq \emptyset) = \text{True} \text{ and } \mathcal{E}_{\pi(i)} = (\mathcal{P}(b_{P_{\pi(i)}}) \cap \beta_i \neq \emptyset) = \text{True},$$

where  $\beta_{\pi(i)} \succ b_{P\pi(i)}$  and  $\beta_i \succ b_{Pi}$ . It yields a contradiction to the point-order consistency.

Next, we consider the remaining case where at least one end of the path  $\mathfrak{S}$  is a blue edge, meaning the substructure on the end does not overlap with any other substructure. W.l.o.g., suppose the end node is  $\mathsf{St}_1$  and it is R-correlated with  $\mathsf{St}_2$ . The event  $\mathcal{E}_1 = \mathcal{P}(b_{P1}) \cap \mathcal{P}(\beta_{\pi(1)}) \neq \emptyset$ ) is always false since  $\beta_{\pi(1)}$  does not exist. So all clause containing  $\mathcal{E}_1$  is false. To make each of the remaining clauses true,  $\mathcal{E}_2$  must be true, and the case reduces to the previous case (by simply omitting node  $\mathsf{St}_1$ ). So the same argument again renders a contradiction. This completes the proof of the theorem.

### 7 Constructing $\mathcal{H}$

In this section, we describe how to construct the set  $\mathcal{H}$  in details. We first include in  $\mathcal{H}$  all square gadgets. The boundary of  $\mathcal{H}$  consists of several closed curves, as shown in Figure 2.  $\mathcal{H}$  and all uncovered arcs define a set of substructures.

First, we note that there may exist a closed curve that all points on the curve are covered by some arcs (or informally, we have a cyclic substructure, with the baseline being a cycle). We need to break all such baseline cycles by including a constant number of extra arcs into  $\mathcal{H}$ . This is easy after we introduce the label-cut operation in Section 7.1, and we will spell out all details then. Note that we cannot choose some arbitrary envelope cycle since it may ruin some good properties we want to maintain.

From now on, we assume that all baselines are simple paths. Now, each closed curve contains one or more baselines. So, we have an initial set of well defined substructures. The main purpose of this section is to cut these initial substructures such that Lemma 19 holds.

We will execute a series of operations for constructing  $\mathcal{H}$ . We first provide below a high level sketch of our algorithm, and outline how the substructures and the substructure relation graph  $\mathfrak{S}$  evolve along with the operations.

• (Section 7.1) First, we deal with active regions. Sometimes, two active region may overlap significantly and become inseparable (formally defined later), they essentially need to be dealt

as a single active region. In this case, we merge the two active regions together (we do not need to do anything, but just to pretend that there is only one active region). We can also show that one active region can be merged with at most one other active region. For the rest of cases, two overlapping active region are separable, and we can cut them into at most two non-overlapping active regions, by adding a small number of extra disks in  $\mathcal{H}$ . After the merging and cutting operations, each substructure contains at most one active region. Hence, the substructures satisfy the property (P1) in Lemma 19. Moreover, we show that if any substructure contains an active region, the substructure is limited in a small region.

- (Section 7.2) We ensure that each substructure is non-self-intersecting by a simple greedy algorithm. After this step, (P2) is satisfied.
- (Section 7.3) In this step, we ensure that substructure relation graph  $\mathfrak{S}$  is a acyclic 2-matching (P3). The step has three stages. First, we prove that the set of blue edges forms a matching. Second, we give an algorithm for cutting the substructures which overlap with two or more other substructures. After the cut, each substructure overlaps with no more than one other substructure. So after the first two stages, we can see that  $\mathfrak{S}$  is composed of a blue matching and a red matching. At last, we prove that the blue edges and red edges cannot form a cycle, establishing  $\mathfrak{S}$  is acyclic.
- (Section 7.4) The goal of this step is to ensure the point-order consistency (P4). We first show there does not exist a point covered by more than two substructures, when  $\mathfrak{S}$  is an acyclic 2-matching. Hence, we only need to handle the case of two overlapping substructures. We show it is enough to break all cycles in a certain planar directed graph. Again, we can add a few more disks to cut all such cycles.
- (Section 7.5) Lastly, we show that the number of disks added in  $\mathcal{H}$  in the above four steps is  $O(K^2)$ , where  $K = \frac{L}{\mu}$  and L and  $\mu$  are side lengths of block and small square respectively.

### 7.1 Merging and Cutting Active Regions

If two active regions overlap in the same substructure, we need to either merge them into a new one or cut them into two non-overlapping new ones. As we know, each gadget may have two active regions. Suppose active regions  $Ar_1$  and  $Ar_2$  belong to the same gadget Gg, while  $Ar'_1$  and  $Ar'_2$  belong to a different gadget Gg'. Due to R-correlations, we need consider the four active regions together.

First, we consider the case where  $Ar_1$  overlaps with  $Ar'_1$ , and  $Ar_2$  overlaps with  $Ar'_2$ . We need the following important concept *order-separability*, which characterizes how the two sets of arcs overlap.

**Definition 23** (Order-separability). Consider a substructure St(b, A).  $A_1, A_2$  are two disjoint subsets of A. If  $A_1, A_2$  satisfy that

$$a \prec b$$
, for any  $a \in \mathcal{A}_1$  and  $b \in \mathcal{A}_2$ , (5)

we say that  $\mathcal{A}_1, \mathcal{A}_2$  are order-separable.

We use  $A_1$ ,  $A'_1$ ,  $A_2$ ,  $A'_2$  to denote the set of arc associated with active regions  $Ar_1$ ,  $Ar'_1$ ,  $Ar_2$ ,  $Ar'_2$ , respectively. If  $A_1$  and  $A'_1$  are not order-separable, we say the pair  $(Ar_1, Ar'_1)$  is a *mixture*. If

both of  $(Ar_1, Ar'_1)$  and  $(Ar_2, Ar'_2)$  are mixtures, we say the two pairs form a *double-mixture*. When they are double-mixture, we merge them simultaneously. It only means that we regard the two active regions  $(Ar_1, Ar'_1)$  as a new single active region, and  $(Ar_2, Ar'_2)$  as another single active region.

To show an active region cannot grow unbounded, we prove that the merge operations do not generate chain reactions. The rough idea is that if two active regions form a mixture, their corresponding small squares must be adjacent and their core-central areas must overlap. Due to the special shape of core-central areas (a narrow spindle shape), the overlapping can only happen between two of them, not more.

**Lemma 24.** Consider two non-empty small squares  $\Gamma$ ,  $\Gamma'$ . Suppose the square gadgets in the two squares are  $\mathsf{Gg}(\Gamma) = (\mathsf{D}_s, \mathsf{D}_t)$  and  $\mathsf{Gg}'(\Gamma') = (\mathsf{D}'_s, \mathsf{D}'_t)$ . The active region pairs  $(\mathsf{Ar}_1, \mathsf{Ar}_2)$  and  $(\mathsf{Ar}'_1, \mathsf{Ar}'_2)$  are associated with gadget  $\mathsf{Gg}(\Gamma)$  and  $\mathsf{Gg}'(\Gamma')$  respectively.  $(\mathsf{Ar}_1, \mathsf{Ar}_2)$  and  $(\mathsf{Ar}'_1, \mathsf{Ar}'_2)$  form a double-mixture. Then, the following statements hold:

- 1. Their corresponding squares  $\Gamma, \Gamma'$  are adjacent;
- 2. The core-central areas of  $\mathsf{Gg}(\Gamma)$  and  $\mathsf{Gg}'(\Gamma')$  overlap;
- 3. The angle between  $D_s D_t$  and  $D'_s D'_t$  is  $O(\epsilon)$
- 4. None of the two core-central areas can overlap with any small squares other than  $\Gamma$  and  $\Gamma'$ .

Next, we consider the case where only one of  $(Ar_1, Ar'_1)$  and  $(Ar_2, Ar'_2)$  is a mixture. However, we show that it is impossible as follows. We use notation  $Ar(\mathcal{A})$  to denote an active region with arc set  $\mathcal{A}$ .

**Lemma 25.** Suppose active region pairs  $(Ar_1(\mathcal{A}_1), Ar_2(\mathcal{A}_2))$  and  $(Ar'_1(\mathcal{A}'_1), Ar'_2(\mathcal{A}'_2))$  are associated with gadget Gg and Gg' respectively. If  $\mathcal{A}_1$  and  $\mathcal{A}'_1$  are order-separable, then  $\mathcal{A}_2$  and  $\mathcal{A}'_2$  are also order-separable.

The proofs of Lemma 24 and Lemma 25 are elementary planar geometry and we defer them to Appendix A.2.

**Cutting Overlapping Active Regions:** After the merging stage, even if any two active regions overlap in the same substructure, their arcs are order-separable. We define *label-cut* operation as below to further separate them such that the baselines of all the active regions are non-overlapping.

We consider two overlapping active regions  $\operatorname{Ar}_1(\mathcal{A}_1)$  and  $\operatorname{Ar}_2(\mathcal{A}_2)$  in a substructure  $\operatorname{St}(\mathcal{A})$ . Suppose  $\mathcal{A}_1, \mathcal{A}_2 \subset \mathcal{A}$  and  $\mathcal{A}_1, \mathcal{A}_2$  are order-separable. We can cut St into two substructures  $\operatorname{St}_1$  and  $\operatorname{St}_2$  with disjoint baselines, such that  $\mathbb{R}(\operatorname{Ar}_1) \subset \mathbb{R}(\operatorname{St}_1)$  and  $\mathbb{R}(\operatorname{Ar}_2) \subset \mathbb{R}(\operatorname{St}_2)$ . In other words, if we assign arcs in  $\mathcal{A}_1$  one kind of label and arcs in  $\mathcal{A}_2$  a different kind of label, after the label-cut operation, the arcs with different labels belong to different new substructures. We define label-cut formally as follows.

**Definition 26** (Label-cut). Consider a substructure St(b, A). Suppose  $A'_1, A'_2$  are two subsets of A.  $A'_1$  and  $A'_2$  are order-separable. We can add two consecutive arcs in the envelope of A into H. After that, St can be separated into two new substructures  $St_1(b_1, A_1), St_2(b_2, A_2)$ , such that subarcs in  $A'_1$  only belong to  $A_1$ , and subarcs in  $A'_2$  only belong to  $A_2$ .

**Lemma 27.** Consider a substructure St(b, A) and two subsets  $A'_1, A'_2$  of A. There exists a label-cut when  $A'_1$  and  $A'_2$  are order-separable, and we can perform the label-cut in polynomial time.



Figure 7: The example of label-cut. The left hand side illustrates the whole substructure before cutting. The arcs have two different labels. One is green and the other is brown. The bold black subarcs are what we select in the envelope. The the right hand side illustrates that each of the two separable substructures induced by the label-cut operation only contains arcs with the same label.



Figure 8: The farthest disk pair of square  $\Gamma$  is  $(D_s, D_t)$ . Suppose  $D_s$  and  $D_t$  intersect at point P, Q. The disk D(P, 2) is tangent to  $D_s$  and  $D_t$  at point  $Q_s$  and  $Q_t$  respectively.  $D(Q_s, 1)$  and  $D(Q_t, 1)$  intersect at D. The active region of  $\Gamma$  in  $H^+$  is totally covered by  $Dom(\Gamma^+)$ .

Figure 7 illustrates the process of construction. We defer its proof in Appendix A.2.

After the label-cut operation, in each substructure, the baselines for all active regions are not overlapping. Thus, if any substructure contains more than one active region, consider any two of them, say  $\mathcal{A}_1$  and  $\mathcal{A}_2$ . We can add into  $\mathcal{H}$  one arc *a* along the envelope which satisfies  $a_1 \prec a \prec a_2, \forall a_1 \in \mathcal{A}_1, a_2 \in \mathcal{A}_2$ . After the addition of arc *a*,  $\mathcal{A}_1$  and  $\mathcal{A}_2$  are separated into two different new substructures. Repeat the above step whenever one substructure contains more than one active region. This establishes the active region uniqueness property (P1).

Limiting the size of substructure which contains an active region: Now, we discuss how to make substructure which contains an active region bounded inside a small region. This property is particular useful later when we show the substructure relation graph  $\mathfrak{S}$  is acyclic.

Suppose the gadget of square  $\Gamma$  is  $(\mathsf{D}_s, \mathsf{D}_t)$ . The line  $D_s D_t$  divides the plane into two halfplanes  $H^+$  and  $H^-$ .  $\mathsf{D}_s$  and  $\mathsf{D}_t$  intersect at point P and Q. The boundary of disk  $\mathsf{D}(P,2)$  is tangent to  $\mathsf{D}_s$  and  $\mathsf{D}_t$  at point  $Q_s$  and  $Q_t$  respectively.  $\mathsf{D}(Q_s, 1)$  and  $\mathsf{D}(Q_t, 1)$  intersect at point D. See Figure 8. We call D the dome-point of gadget  $\mathsf{Gg}(\Gamma)$  and use  $\mathsf{Dom}(\Gamma^+)$  to denote the region  $(\mathsf{D}(D,1)) - \mathsf{D}_s - \mathsf{D}_t) \cap H^+$ .

**Lemma 28.** Consider the substructure St which contains an active region of square  $\Gamma$ . The substructure can be cut into at most three smaller substructures, by doing label-cut twice. At most one of them contains the active region. Moreover, this new substructure (if any) is bounded by the region  $\mathsf{Dom}(\Gamma^+)$  (or  $\mathsf{Dom}(\Gamma^-)$ ) associated with  $\Gamma$ .

We defer the proof to Appendix A.2.

There may exist a substructure which contains a merged active region (i.e. an active region which is the union of two initial active regions). Based on Lemma 24, the arcs of the two initial active regions belong to two adjacent small squares. Suppose the two squares are  $\Gamma$ ,  $\Gamma'$ , with square gadgets  $(\mathsf{D}_s, \mathsf{D}_t)$  and  $(\mathsf{D}'_s, \mathsf{D}'_t)$  respectively. The dome-points of  $\Gamma$  and  $\Gamma'$  in  $H^+$  are respectively D and D'. We apply the operations in Lemma 28 for each initial active region. Since the angle between  $D_s D_t$  and  $D'_s D'_t$  is  $O(\epsilon)$ , obviously, the substructure containing the merged active region is small as well, i.e. bounded in  $\mathsf{Dom}(\Gamma'^+) \cup \mathsf{Dom}(\Gamma'^+)$ .

Since each substructure which contains an active region is small enough, we can show the following lemma, which will be useful for proving the acyclicity of the substructure relation graph  $\mathfrak{S}$ .

**Lemma 29** (Highly parallel arcs). Suppose substructure St(b, A) contains an active region. The central angle of any arc in St is no more than  $O(\epsilon)$ . Meanwhile, there exists a line l such that the angle between l and the tangent line at any point of any arc  $a \in A$  is at most  $O(\epsilon)$ .

Actually, for the initial active region, the line  $D_sD_t$  satisfy the property. For the merged active region, we know the angle between  $D_sD_t$  and  $D'_sD'_t$  is  $O(\epsilon)$  based on Lemma 24. Hence, we can still see that the line  $D_sD_t$  satisfies the property.

To summarize, after all operations in this subsection, we can ensure that each substructure contains at most one active region (i.e., (P1) in Lemma 19). Moreover, we have that each substructure which contains an active region is small enough (so that Lemma 29 holds).

Handling cyclic substructures: At the end of this subsection, we deal with the problem we left in the very beginning of Section 7, to break all cyclic baselines. Note that this step should be done in the beginning. First, we consider that case that there exists a point on the baseline which cannot be covered by any arc of any active region. We can include any envelope arc that covers the point into  $\mathcal{H}$ , which is enough to break the cycle. This is essentially a label-cut and do not separate any single connected active region into disconnected pieces. Then we consider the case where every point on the baseline is covered by some arc of active regions. Note that the merge operation only depends on the local property of two active regions, thus, we can merge active regions even in "cyclic substructure". Assume that we have done all merge operations. Based on Lemma 29, we know one active region is very small and thus cannot cover all points on a closed curve. We pick one active region. Using the same operation as Lemma 28 (do two label-cuts), we can essentially isolate the active region and cut the original cyclic baseline to two new baselines.

#### 7.2 Eliminating Self-intersections

The goal of this part is to add a few more disks into  $\mathcal{H}$  so that any substructure is non-selfintersecting (Recall the definition in Lemma 9). Note that the substructures which contain active region are non-self-intersecting after the process in Section 7.1. We just need to process the substructures without active regions in this part.

We use a simple greedy approach. We consider one substructure  $\mathsf{St}(\mathsf{b}, \mathcal{A})$ . Suppose the endpoints of  $\mathsf{b}$  are  $Q_s$  and  $Q_t$ , and the envelope is  $\{a_1[Q_s, Q_1], \ldots, a_k[Q_{k-1}, Q_t]\}$ , where  $Q_i$  is the intersection point of  $a_{i-1}$  and  $a_i$ . We denote the endpoints of  $a_i$  on the baseline  $\mathsf{b}$  by  $P_i$  and  $P'_i$  (Note that  $P_1 = Q_s$ ). Let arc set  $\mathcal{A}_i$  be  $\{a_1, a_2, \ldots, a_i\}$ . Then we have a set  $\{\mathsf{St}_i[\mathcal{A}_i]\}_{i \in [k]}$  of substructures, where  $\mathsf{St}_i[\mathcal{A}_i]$  is the substructure induced by arc set  $\mathcal{A}_i$ . We consider the arcs lying on the envelope one by one and check whether we should add it into  $\mathcal{H}$  or not. Concretely, we add  $\mathsf{D}(a_{i+1})$  in  $\mathcal{H}$  if the following condition holds:



Figure 9: The process to avoid self-intersection. The left hand side is a self-intersection substructure. We search from point  $Q_s$  along the envelope. Let arc set  $\mathcal{A}_i$  be  $\{a_1, a_2, \ldots, a_i\}$ . Then we have a set  $\{\mathsf{St}_i[\mathcal{A}_i]\}_{i\in[k]}$  of substructures. If  $\mathsf{St}_i$  is non-self-intersecting but  $\mathsf{St}_{i+1}$  is self-intersecting. We add arc  $a_{i+1}$  in  $\mathcal{H}$ . The right hand side illustrates the two new substructures after the cut.

•  $St_i$  is non-self-intersecting, but  $St_{i+1}$  is self-intersecting,

The addition of  $D(a_{i+1})$  cuts the substructure into two smaller substructures. One of which is certainly non-self-intersecting. The other is induced by arc set  $\mathcal{A} = \{a_{i+2}, \ldots, a_k\}$ . See Figure 9. We repeat the above process until there is no self-intersection in all substructures. Furthermore, we can easily prove the following nice property. The proof can be found in Appendix A.3.

**Lemma 30.** In each of the above iterations, one substructure St(b, A) is cut into at most two new substructures. Any original arc in A cannot be cut into two pieces, each of which belongs to a different new substructure.

To summarize, we have obtained the non-self-intersection property ((P2) in Lemma 19).

### 7.3 Ensuring that $\mathfrak{S}$ is an Acyclic 2-Matching

We discuss how to add some extra disks in  $\mathcal{H}$  to make  $\mathfrak{S}$  an acyclic 2-matching ((P3) in Lemma 19).

Blue edges: First we show that the set of blue edges form a matching.

**Lemma 31.** Two blue edges cannot be incident to the same node.

*Proof.* Before the merge operation, the set of active region pairs forms a matching. To see this, note that our merge operations always apply to a double-mixture (which corresponds to merging two blue edges into one). Moreover, any cut operation cannot break one active region into two, thus has no effect on any blue edge. Hence, after all merge and cut operations, the set of active region pairs is still a matching.  $\Box$ 

**Red edges:** Then, we prove that any node which has more than one incident red edges can be cut such that each new node(i.e., substructure) only has at most one incident red edge.

First, we prove a simple yet useful geometric lemma stating that a point cannot be covered by three or more substructures. Note that from now on, all substructures have no self-intersections.

**Lemma 32.** We are given a substructure St(b, A).  $a \in A$  and two arcs  $b_1, b_2 \notin A$ . If  $b_1, b_2$  cover the same point on  $a, b_1, b_2$  should belong to the same substructure.

Intuitively, if the two disks corresponding to  $b_1$  and  $b_2$  cover the same point, they should be close enough such that their corresponding square gadgets overlap (which implies  $b_1$  and  $b_2$  share the



Figure 10: The arcs in substructures  $St_1$  and  $St_2$  cut the envelope of St into 7 segments. The sequence of the labels for those segments is 0101020. The compressed label sequence is 01020. So we have 5 l-segments.

same baseline). First we prove that the minimum distance between any two disks in two different substructure should not be too small, i.e., their overlapping region cannot be too large. The proof can be found in Appendix A.4.

In fact, essentially the same proof can be used to prove that any two different substructures cannot both intersect with subarc whose central angle is  $O(\epsilon)$ , as in the following corollary.

**Corollary 33.** Suppose St(b, A) is a substructure without any self-intersection. Consider an arc  $a \in A$  and two arcs  $b_1, b_2 \notin A$ . Suppose a' is a subarc of a whose central angle is  $O(\epsilon)$ . If both  $b_1, b_2$  cover some part of a',  $b_1, b_2$  should belong to the same substructure.

Combining with Lemma 29, we can easily see the following lemma.

**Lemma 34.** Any substructure which contains an active region cannot overlap with two or more different substructures.

Then we show how to cut the substructure  $\mathsf{St}(\mathsf{b}, \mathcal{A})$  which overlaps with more than one other substructures. Note that such substructure does not contain an active region based on Lemma 34. Suppose the envelope of  $\mathsf{St}$  is  $\{a_1[Q_s, Q_1], \ldots, a_k[Q_{k-1}, Q_t]\}$ . St overlaps with k substructures  $\mathsf{St}_i(\mathsf{b}_i, \mathcal{A}_i), i = 1, 2, \ldots, k$ .

If  $St_i$  overlaps with St, there exists an arc  $a \in St_i$  intersecting some envelope arc of St. Thus, the envelope St is subdivided into several segments by those intersection points. We can label those segments as follows:

- If the segment is covered by some arc in  $St_i$ , we label it as 'i'.
- If there is no arc in any  $St_i$  covering the segment, we label it as '0'.

See Figure 10 for an example. According to the Lemma 32, we know there is no point on the envelope covered by two substructures. Thus the above labeling scheme is well defined.

Traversing those segments along the envelope, we obtain a label sequence. First, for each label i, we identify those maximal consecutive subsequence, which consists of only letter 0 and i, and starts with and end with i, and replace the subsequence by a single letter i. We obtain a compressed sequence. In fact, each letter, say i ( $i \neq 0$ ), in the compressed sequence corresponds to one or more segments labeled with either i or 0, and the first and last one must be labeled with i. We call the concatenation of those segments an l(ong)-segment. Of course, a letter 0 in the compressed sequence is Davenport-Schinzel sequence of order 2 [37], because two different labels cannot intersect (because the baselines of two

substructures cannot intersect). For example, the pattern "1212" should never appear. Thus, the length of the compressed sequence, i.e., the number of l-segments, is at most O(k).

Now, we discuss how to cut St into several new ones based on l-segments. Keep in mind that our goal is to make sure each new substructure only overlap with one substructure of  $\{St_i\}_{i\in[k]}$ . The cut operation is again a simple greedy procedure. Consider two consecutive l-segments. Suppose they are  $\{a_i[Q_{i-1}, Q_i], a_{i+1}[Q_i, Q_{i+1}], \ldots, a_j[Q_{j-1}, Q_j]\}$  and  $\{a_{j+1}[Q_j, Q_{j+1}], \ldots, a_k[Q_{k-1}, Q_k]\}$ . We add into  $\mathcal{H}$  the last arc  $a_j$  of former l-segment and the first arc  $a_{j+1}$  of the later l-segment. St is thus cut into two new ones. Repeat the above step for all two consecutive l-segments in order.

We still need to show that after the cut, every new substructure overlap at most one of  $\{St_i\}_{i \in [k]}$ . Consider one new substructure. Notice such an original arc in  $\mathcal{A}$  can only belong to one new substructure. By our cut operation, all envelope arcs of the new substructure can intersect at most one of  $\{St_i\}_{i \in [k]}$ . Hence, the new substructure can overlap at most one  $St_i$ .

Blue edges and red edges: After the above operations, the set of all blue edges forms a matching, while the set of all red edges also forms a matching. To show  $\mathfrak{S}$  is an acyclic 2-matching, it suffices to prove  $\mathfrak{S}$  contains no cycle. So, the rest of the section is devoted to prove the following lemma.

**Lemma 35.** Suppose the side length of square is  $\mu$ , where  $\mu = O(\epsilon)$  and the block contains  $K \times K$  small squares, where  $K = \frac{C_0}{\epsilon^2}$  and  $C_0$  is an appropriate constant. Then, after all operations stated in this section, there is no cycle in  $\mathfrak{S}$ .

If there is a cycle Cyc in  $\mathfrak{S}$ , the red edges and blue edges alternate in Cyc, which correspond to a sequence of substructures, each containing an active region (since it is R-correlated with another active region). Now, we provide a high level explanation why Lemma 35 should hold. Each active region is associated with a small square. A small square is very small (comparing to a unit disk or the whole block), and an active region is also very small. We pick a point in each small square and each substructure. If two substructures in Cyc overlap, the two points in them are also very close (i.e.,  $O(\epsilon)$ ). So we can pretend the two points as one point (or we just pick a point in their overlapping region). For each active region Ar, we connect the point in Ar and the point in the small square associated with Ar. Since Both the small square and the active region are very small, the distance between two point is about  $1 - O(\epsilon)$ . Thus, a cycle Cyc would present itself geometrically as a polygon. We can show the angle between two adjacent edges of the polygon is close to  $\pi$ . So the size of the polygon cannot be not small (it takes a lot of edges to wrap a loop). However, the polygon cannot be much larger than the block. By contradiction, we prove that there is no cycle in  $\mathfrak{S}$ .

Now, we formally prove Lemma 35. We first prove a geometric lemma which will be useful for bounding the angle between the two adjacent edges of the aforementioned polygon.

**Lemma 36.** Consider two substructures  $St_1(b_1, A_1)$  and  $St_2(b_1, A_2)$  in Cyc. Suppose  $St_1$  and  $St_2$ overlap (there is a red edge between them). For any two arcs  $a \in A_1$  and  $b \in A_2$ , suppose that D(a)and D(b) overlap and their intersection points are A and B. The central angles of a, b are  $\theta_a, \theta_b$ respectively. The centers of D(a), D(b) are  $D_a, D_b$ . Then  $\angle AD_aB$  (or  $\angle AD_bB$ ) is at most ( $\theta_a + \theta_b$ ).

*Proof.* We distinguish a few cases depending on whether the intersection points A and B lie on a or b or none of them. All cases are depicted in Figure 11.

• Both A and B lie on one arc (see Figure (a)(b)). W.l.o.g., suppose they lie on arc a. Obviously,  $\angle AD_aB$  is no more than  $\theta_a$  (or  $\theta_b$ ).



Figure 11: The four kinds of cases for two overlapping arcs.

- If one intersection is on neither a nor b (see Figure (c)), we prove that the case cannot happen. Suppose A is on neither of a and b. The endpoint  $A_1$  of a is covered by disk D(b) and the endpoint  $A_2$  of a is covered by disk D(a). Thus the baseline  $b_1$  must intersect with  $b_2$  which contradicts the fact that  $St_1$  and  $St_2$  are two different substructures.
- If one intersection point is on a but not on b and the other intersection point is on b but not on a (see Figure (d)), it is easy to see that  $B_1$  is covered by arc b, and  $A_2$  is covered by asince their baselines do not intersect. Thus, the length of arc AB (w.r.t. D(a)) is no more than the sum of lengths of a and b. So  $\angle AD_aB$  is at most  $\theta_a + \theta_b$ .

The above cases are exhaustive, thus our proof is completed.

Based on Lemma 29, Lemma 36, we can prove Lemma 35 below:

Proof of Lemma 35: Consider two substructures  $St_1$  and  $St_2$ . Suppose  $St_1$  contains the active region  $Ar_1$ , and  $St_2$  contains the active region  $Ar_2$ . The centers of arcs of  $Ar_1$  and  $Ar_2$  locate in small squares  $\Gamma_1$  and  $\Gamma_2$  respectively. The square gadgets of the two squares are  $Gg(D_{s_1}, D_{t_1})$  and  $Gg(D_{s_2}, D_{t_2})$  respectively. If  $St_1$  and  $St_2$  overlap, their must exist an arc *a* in  $St_1$  and an arc *b* in  $St_2$  such that *a* and *b* intersect. Suppose D(a) and D(b) intersect at points *A* and *B*. The center of D(a) and D(b) are  $D_a$  and  $D_b$ .

Based on Lemma 29, we know both central angle of a and b are no more than  $O(\epsilon)$ . According to Lemma 36, the central angle of the arc AB is at most  $O(\epsilon)$ . It means angle between the tangent lines of D(a) and D(b) at point A is no more than  $O(\epsilon)$ . Thus,  $\angle D_a A D_b$  is at least  $\pi - O(\epsilon)$ .

We know that all disks in the same active region are centered in one small square or two adjacent small squares. Moreover Lemma 29 implies all disk centers should lies in one or two squares. Hence, the distance between (any point in) the square and (any point in) its active region is at least  $1 - O(\epsilon)$ . Construct the aforementioned polygon. Consider two adjacent edges XY and YZ in the polygon. We consider two cases:

1. Y is in the intersection of two substructures  $\mathsf{St}_1$  and  $\mathsf{St}_2$ . We can easily see that (1)  $|YA| = O(\epsilon)$ ; (2)  $|XD_a| = O(\epsilon)$ ; (3)  $|ZD_b| = O(\epsilon)$ . Hence, we can see  $\angle XYZ$  is at least  $\pi - O(\epsilon)$ .



Figure 12: Substructure  $\mathsf{St}_1(\mathsf{b}_1, \mathcal{A}_1)$  and  $\mathsf{St}_2(\mathsf{b}_2, \mathcal{A}_2)$  are overlapping.  $\mathsf{b}_1$  starts from  $Q_s$  and ends up with  $Q_t$  and  $\mathsf{b}_2$  starts from  $P_s$  and ends up with  $P_t$ . There are two paths forming a cycle.

2. Y is in a small square  $\Gamma$  and X and Z are in the two substructures associated with  $\Gamma$ . Since both substructures are bound in an  $O(\epsilon)$  size region (by Lemma 29), we can see that  $\angle XYZ$  is at least  $\pi - O(\epsilon)$  as well.

Hence, we can see the polygon contains at least  $\Omega(2\pi/\epsilon)$  nodes and the diameter of the polygon is at least  $\Omega(1/\epsilon)$ . But this cannot be larger than the diameter of a block, rendering a contradiction.

To summarize, we have ensured that  $\mathfrak{S}$  is an acyclic 2-matching ((P3) in Lemma 19).

#### 7.4 Ensuring Point Order Consistence

We have ensured that the set of red edges is a matching in  $\mathfrak{S}$ . Hence, one substructure can overlap with at most one other substructure. Therefore, if we can guarantee that the points which are covered by any pair of overlapping substructure are order consistent, then all points in  $\mathcal{P}$  are order consistence (after all, point-order consistency is defined over a pair of substructures).

Consider two overlapping substructures  $St_1(b_1, A_1)$  and  $St_2(b_2, A_2)$  and a set  $\mathcal{P}_{co}$  of points covered by  $\mathcal{A}_1 \cup \mathcal{A}_2$ . Suppose  $P_1, P_2 \in \mathcal{P}_{co}$  and  $a_1, a_2 \in \mathcal{A}_1, b_1, b_2 \in \mathcal{A}_2$ . Recall point-order consistency requires that the following conditions are met

- $P_1 \in \mathbb{R}(a_1) \cap \mathbb{R}(b_1)$  and  $P_2 \in \mathbb{R}(a_2) \cap \mathbb{R}(b_2)$
- $P_1 \notin \mathbb{R}(a_2) \cup \mathbb{R}(b_2)$  and  $P_2 \notin \mathbb{R}(a_1) \cup \mathbb{R}(b_1)$ .
- $(a_1 \prec a_2) \Leftrightarrow (b_1 \prec b_2).$

It is helpful to consider the following directed planar graph  $\mathfrak{D}$  induced by all arcs in  $\mathcal{A}_1 \cup \mathcal{A}_2$ in the uncovered region  $\mathbb{U}(\mathcal{H})$ . Regard each intersection point of arcs in  $\mathcal{A}_1 \cup \mathcal{A}_2$  as a node. Each subarc is an directed edge with a direction consistent with its baseline. We use  $A \to B$  to denote that there is a directed edge between nodes A and B in  $\mathfrak{D}$ . If there is no directed cycle in  $\mathfrak{D}$ , we can verify that all conditions listed above hold. Indeed, suppose the condition is not satisfied, which means  $a_1 \prec a_2$ ,  $b_1 \succ b_2$ ,  $a_1, a_2 \in \mathcal{A}_1$  and  $b_1, b_2 \in \mathcal{A}_2$ . Suppose  $a_1, b_1$  intersect at  $(A_1, B_1)$  and  $A_1 \to B_1$ , meanwhile  $a_2, b_2$  intersect at  $(A_2, B_2)$  and  $A_2 \to B_2$ . Since  $a_1 \prec a_2$ , there exists a path in  $St_1$  which goes from  $B_1$  to  $A_2$ . Similarly, there exists a path in  $St_2$  from  $B_2$  to  $A_1$ . Thus, the two paths and  $a_1, a_2$  form a directed cycle.

So all we have to do is to break all cycles in  $\mathfrak{D}$ . When  $\mathfrak{D}$  contains a cycle, we can cut the cycle through adding an arc on the envelope into  $\mathcal{H}$ . See Figure 12 for an example. Only arcs  $\mathcal{A}_1$  (or in  $\mathcal{A}_2$ ) cannot form a cycle. So if there is a cycle, the cycle must pass through the envelope of  $\mathcal{A}_1$  and  $\mathcal{A}_2$ . Moreover, based on Lemma 29, if St<sub>1</sub> and St<sub>2</sub> form a cycle, either St<sub>1</sub> or St<sub>2</sub> does not contains any active region. W.l.o.g., suppose it is St<sub>2</sub>. Based on these observations, we have our algorithms as follows:

Suppose the envelope of  $\mathsf{St}_1$  is  $\mathsf{Path}_1 = \{a_1, a_2, \ldots, a_k\}$ .  $a_i$  and  $a_j$  are the first and last arcs respectively which intersect  $\mathsf{St}_2$ . Suppose the arc  $b_{i'} \in \mathcal{A}_2$  overlaps with  $a_i$ , (if there is more than one such arc, we select a minimal one, w.r.t. the arc ordering) and the arc  $b_{j'} \in \mathcal{A}_2$  overlaps with  $a_j$  (if there is more than one such arc, we select a maximal one). Since  $a_i \prec a_j$  and they do not satisfy point-order consistency, we have  $b_{i'} \succ b_{j'}$ . We can see that all arcs between  $b_{i'}$  and  $b_{j'}$  cannot intersect with  $\mathsf{Path}_1$ . So we can select one arc between  $b_{i'}$  and  $b_{j'}$  to add in  $\mathcal{H}$  for cutting  $\mathsf{St}_2$  into two. After the cut, any cycle in  $\mathfrak{D}$  can be broken.

After cutting  $St_2$ ,  $St_1$  overlaps with both of the new substructures obtained from  $St_2$ . Then, we encounter the same situation as in Section 7.3 (a node in  $\mathfrak{S}$  has two incident red edges). We can apply the operation in Section 7.3 to cut  $St_1$  such that the set of red edges in  $\mathfrak{S}$  is still a matching.

#### 7.5 The number of disks in $\mathcal{H}$

Finally, we count collectively the total number of disks that we have added in  $\mathcal{H}$ . First, we add the square gadget for each nonempty small square in  $\mathcal{H}$ . The number of the disks is  $O(K^2)$ , where  $K = L/\mu = O(1/\epsilon^2)$ . In order to cut overlapping active regions, the number of disks we add in  $\mathcal{H}$ is bounded by the number of active regions. Since there are  $O(K^2)$  gadgets, we add  $O(K^2)$  disks in Section 7.1. In Section 7.2, to ensure that each substructure is non-self-intersecting and contains at most one active region, we design a greedy algorithm. We can see that each arc we added in  $\mathcal{H}$ covers at least one intersection point of two disks in  $\mathcal{H}'$ , where  $\mathcal{H}'$  is the set  $\mathcal{H}$  before this step. The algorithm guarantees that each arc which we add in  $\mathcal{H}$  do not cover the same intersection point on the boundary of  $\mathcal{H}'$ . Since the union complexity of unit disks is linear [32] and  $\mathcal{H}'$  contains at most  $O(K^2)$  disks, there are at most  $O(K^2)$  intersection points on  $\partial \mathcal{H}'$ . So, we add at most  $O(K^2)$  in this step. In Section 7.3, we break the cycles in  $\mathfrak{S}$ . The number of disks we add  $\mathcal{H}$  is proportional to the number of substructures. So, again, we add at most  $O(K^2)$  disks. Similarly, in Section 7.4, we also add at most  $O(K^2)$  disks. To summarize, we have added at most  $O(K^2)$  disks in  $\mathcal{H}$ .

### 8 Time Complexity

The time complexity contains three parts. The first part is to enumerate all combination of  $\mathcal{G}$ . We set  $C = O(\frac{K^2}{\epsilon})$  (since we need  $C > |\mathcal{H}|/\epsilon$ ). Since  $K = O(\frac{1}{\epsilon^2})$ , the number of combinations is bounded by  $n^C = n^{O(1/\epsilon^5)}$ . The second part is the construction of set  $\mathcal{H}$ . It is easy to see that the time cost for each operation (i.e., label-cut) is no more than  $O(n^2)$ . Thus, the time cost is  $O(K^2n^2) = O(n^2/\epsilon^4)$ . The last part is the dynamic program. There are at most  $O(K^2)$ substructures and at most  $O(n^2)$  intersection points in each substructure. Thus, the number of total states is at most  $O((n^2)^{K^2})$ . For each recursion, the time cost is at most O(n). Thus, the overall time complexity of the dynamic program is  $O(n^{2K^2+1}) = n^{O(1/\epsilon^4)}$ . Overall, the total time cost is  $n^{O(1/\epsilon^5)} \cdot \max\{n^2/\epsilon^4, n^{O(1/\epsilon^4)}\}) = n^{O(1/\epsilon^9)}$ . This finishes the proof of Lemma 6.

# 9 Applications

The weighted dominating set problem (MWDS) in unit disk graphs has numerous applications in the areas of wireless sensor networks [17]. In this section, we show that our PTAS for WUDC can be used to obtain PTASs for two important problems in this domain.

#### 9.1 Connected Dominating Set in UDG

The goal for the *minimum-weighted connected dominating set* problem (MWCDS) is to find a dominating set which induces a connected subgraph and has the minimum total weight. Clark et al. [11] proved that MWCDS in unit disk graphs is NP-hard. Ambühl et al. [2] obtained the first constant factor approximation algorithm for MWCDS(the constant is 94). The ratio was subsequently improved in a series of papers [13, 20, 30]. The best ratio known is 7.105 [17, pp.78].

One way to compute an approximation solution for MWCDS is to first compute minimum weighted dominating set (MWDS) and then connect the dominating set using a *node-weighted* steiner tree (NWST) [30, 41]. The optimal MWDS value is no more than the optimal MWCDS value. After zeroing out the weight of all terminals, the optimal NWST value (for any set of terminals) is also no more than the optimal MWCDS value. Hence, if there is an  $\alpha$ -approximation for MWDS(or equivalently WUDC) and a  $\beta$ -approximation for NWST, then there is an  $\alpha + \beta$  factor approximation algorithm for MWCDS.

Zou et al. [41] show that there exists a  $2.5\rho$ -approximation for NWST if there exists a  $\rho$ -approximation for the classical edge-weighted *minimum steiner tree problem*. The current best ratio for minimum steiner tree is 1.39 [7]. Thus, there exists a 3.475-approximation for NWST. Combining with our PTAS for WUDC, we obtain the following improved result for MWCDS.

**Theorem 37.** There exists a polynomial-time  $(4.475 + \epsilon)$ -approximation for MWCDS for any fixed constant  $\epsilon > 0$ .

#### 9.2 Maximum Lifetime Coverage in UDG

The maximum lifetime coverage problem (MLC) is a classical problem in wireless sensor networks: Given n targets  $t_1, \ldots, t_n$  and m sensors  $s_1, \ldots s_m$ , each covering a subset of targets, find a family of sensor cover  $S_1, \ldots, S_p$  with time lengths  $\tau_1, \ldots, \tau_p$  in [0, 1], respectively, to maximize  $\tau_1 + \ldots + \tau_p$ subject to that the total active time of every sensor is at most 1. MLC is known to be NP-hard [8]. Berman et al. [4] reduced MLC to the minimum weight sensor cover (MSC) problem through Garg-Könemann technique [24]. In particular, they proved that if MSC has a  $\rho$ -approximation, then MLC has a  $(1 + \epsilon)\rho$ -approximation for any  $\epsilon > 0$ . Ding et al. [14] noted that, if all sensors and targets lie in the Euclidean plane and all sensors have the same covering radius, any approximation result for WUDC can be converted to almost the same approximation for MLC. Hence, the current best known result for MLC is a 3.63-approximation. Using our PTAS, we obtained the first PTAS for MLC.

**Theorem 38.** There exists a PTAS for MLC when all sensors and targets lie in the Euclidean plane and all sensors have the same covering radius.

Let us mention one more variant of MLC, called maximum lifetime connected coverage problem, studied by Du et al. [18]. The problem setting is the same as MLC, except that each sensor cover  $S_i$  should induce a connected subgraph. They obtained a  $(7.105 + \epsilon)$ -approximation when the communication ratios  $R_c$  is no less two times the sensing radius  $R_s$ . Essentially, they showed that an  $\alpha$ -approximation for WUDC and a  $\beta$ -approximation for NWST imply an  $\alpha + \beta$ -approximation algorithm for the connected MLC problem. Using our PTAS, we can improve the approximation ratio to  $(4.475 + \epsilon)$ .

# 10 Final Remarks

Much of the technicality comes from the fact that the substructures interact each other in a complicated way and it is not easy to ensure a globally consistent order. The reader may wonder what if we choose more than two disks (but still a constant) in a small square, hoping that the uncovered regions become separated and more manageable. We have tried several other ways, like choosing a constant number of disks in the convex hull of the centers in a small square. However, these seemed to only complicate, not to simplify, the matter.

We believe our result and insight are useful to tackle other problems involving unit disks or unit disk graphs. On the other hand, our approach strongly relies on the special properties of unit disks and does not seem to generalize to arbitrary disks with disparate radius. Obtaining a PTAS for the weighted disk cover problem with arbitrary disks is still a central open problem in this domain. An interesting intermediate step would be to consider the special case where the ratio between the longest radius and the shortest radius is bounded.

## References

- ADAMASZEK, A., AND WIESE, A. Approximation schemes for maximum weight independent set of rectangles. In Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on (2013), IEEE, pp. 400–409.
- [2] AMBÜHL, C., ERLEBACH, T., MIHALÁK, M., AND NUNKESSER, M. Constant-factor approximation for minimum-weight (connected) dominating sets in unit disk graphs. In Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, J. Díaz, K. Jansen, J. Rolim, and U. Zwick, Eds., vol. 4110 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, pp. 3–14.
- [3] BANSAL, N., AND PRUHS, K. The geometry of scheduling. SIAM Journal on Computing 43, 5 (2014), 1684–1698.
- [4] BERMAN, P., CALINESCU, G., SHAH, C., AND ZELIKOVSKY, A. Efficient energy management in sensor networks. In Ad Hoc and Sensor Networks. Nova Science Publishers (2005), Nova Science Publisher.
- [5] BRÖNNIMANN, H., AND GOODRICH, M. Almost optimal set covers in finite vc-dimension. Discrete & Computational Geometry 14, 1 (1995), 463–479.
- [6] BUS, N., GARG, S., MUSTAFA, N. H., AND RAY, S. Tighter estimates for epsilon-nets for disks. arXiv preprint arXiv:1501.03246 (2015).

- [7] BYRKA, J., GRANDONI, F., ROTHVOSS, T., AND SANITÀ, L. An improved lp-based approximation for steiner tree. In *Proceedings of the forty-second ACM symposium on Theory of computing* (2010), ACM, pp. 583–592.
- [8] CARDEI, M., THAI, M., LI, Y., AND WU, W. Energy-efficient target coverage in wireless sensor networks. In INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE (March 2005), vol. 3, pp. 1976–1984 vol. 3.
- [9] CHAN, T. M., AND GRANT, E. Exact algorithms and apx-hardness results for geometric packing and covering problems. *Computational Geometry* 47, 2, Part A (2014), 112 – 124. Special Issue: 23rd Canadian Conference on Computational Geometry (CCCG11).
- [10] CHAN, T. M., GRANT, E., KÖNEMANN, J., AND SHARPE, M. Weighted capacitated, priority, and geometric set cover via improved quasi-uniform sampling. In *Proceedings of* the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms (2012), SODA '12, SIAM, pp. 1576–1585.
- [11] CLARK, B. N., COLBOURN, C. J., AND JOHNSON, D. S. Unit disk graphs. Discrete Math. 86, 1-3 (Jan. 1991), 165–177.
- [12] CLARKSON, K. L., AND VARADARAJAN, K. Improved approximation algorithms for geometric set cover. Discrete & Computational Geometry 37, 1 (2007), 43–58.
- [13] DAI, D., AND YU, C. A  $5 + \epsilon$ -approximation algorithm for minimum weighted dominating set in unit disk graph. Theoretical Computer Science 410, 8 (2009), 756–765.
- [14] DING, L., WU, W., WILLSON, J., WU, L., LU, Z., AND LEE, W. Constant-approximation for target coverage problem in wireless sensor networks. In *INFOCOM* (2012), IEEE, pp. 1584– 1592.
- [15] DINUR, I., AND STEURER, D. Analytical approach to parallel repetition. In Proceedings of the 46th Annual ACM Symposium on Theory of Computing (2014), ACM, pp. 624–633.
- [16] DU, D.-Z., L KO, K., AND HU, X. Design and Analysis of Approximation Algorithms. Springer, 2011.
- [17] DU, D.-Z., AND WAN, P.-J. Connected Dominating Set: Theory and Applications, vol. 77. Springer Science & Business Media, 2012.
- [18] DU, H., PARDALOS, P., WU, W., AND WU, L. Maximum lifetime connected coverage with two active-phase sensors. *Journal of Global Optimization* 56, 2 (2013), 559–568.
- [19] ERLEBACH, T., GRANT, T., AND KAMMER, F. Maximising lifetime for fault-tolerant target coverage in sensor networks. In *Proceedings of the twenty-third annual ACM symposium on Parallelism in algorithms and architectures* (2011), ACM, pp. 187–196.
- [20] ERLEBACH, T., AND MIHALÁK, M. A (4+ε)-approximation for the minimum-weight dominating set problem in unit disk graphs. In Approximation and Online Algorithms, E. Bampis and K. Jansen, Eds., vol. 5893 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, pp. 135–146.

- [21] ERLEBACH, T., AND VAN LEEUWEN, E. Ptas for weighted set cover on unit squares. In Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, M. Serna, R. Shaltiel, K. Jansen, and J. Rolim, Eds., vol. 6302 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, pp. 166–177.
- [22] EVEN, G., RAWITZ, D., AND SHAHAR, S. M. Hitting sets when the vc-dimension is small. Information Processing Letters 95, 2 (2005), 358–362.
- [23] FEIGE, U. A threshold of ln n for approximating set cover. Journal of the ACM (JACM) 45, 4 (1998), 634–652.
- [24] GARG, N., AND KÖNEMANN, J. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. SIAM J. Comput. 37, 2 (May 2007), 630–652.
- [25] GIBSON, M., AND PIRWANI, I. A. Algorithms for dominating set in disk graphs: breaking the logn barrier. In ESA. Springer, 2010, pp. 243–254.
- [26] HAR-PELED, S., KAPLAN, H., SHARIR, M., AND SMORODINSKY, S. Epsilon-nets for halfspaces revisited. arXiv preprint arXiv:1410.3154 (2014).
- [27] HAR-PELED, S., AND LEE, M. Weighted geometric set cover problems revisited. JoCG 3, 1 (2012), 65–85.
- [28] HOCHBAUM, D. S., AND MAASS, W. Approximation schemes for covering and packing problems in image processing and vlsi. *Journal of the ACM (JACM) 32*, 1 (1985), 130–136.
- [29] HOCHBAUM, D. S., AND MAASS, W. Fast approximation algorithms for a nonconvex covering problem. Journal of Algorithms 8, 3 (1987), 305 – 323.
- [30] HUANG, Y., GAO, X., ZHANG, Z., AND WU, W. A better constant-factor approximation for weighted dominating set in unit disk graph. *Journal of Combinatorial Optimization 18*, 2 (2009), 179–194.
- [31] HUNT, H. B., III, MARATHE, M. V., RADHAKRISHNAN, V., RAVI, S. S., ROSENKRANTZ, D. J., AND STEARNS, R. E. Nc-approximation schemes for np- and pspacehard problems for geometric graphs, 1997.
- [32] KEDEM, K., LIVNE, R., PACH, J., AND SHARIR, M. On the union of jordan regions and collision-free translational motion amidst polygonal obstacles. *Discrete & Computational Ge*ometry 1, 1 (1986), 59–71.
- [33] MARX, D. On the optimality of planar and geometric approximation schemes. In Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on (2007), IEEE, pp. 338–348.
- [34] MUSTAFA, N. H., RAMAN, R., AND RAY, S. Qptas for geometric set-cover problems via optimal separators. arXiv preprint arXiv:1403.0835 (2014).
- [35] MUSTAFA, N. H., AND RAY, S. Ptas for geometric hitting set problems via local search. In Proceedings of the twenty-fifth annual symposium on Computational geometry (2009), ACM, pp. 17–22.

- [36] PYRGA, E., AND RAY, S. New existence proofs  $\varepsilon$ -nets. In Proceedings of the twenty-fourth annual symposium on Computational geometry (2008), ACM, pp. 199–207.
- [37] SHARIR, M. Davenport-schinzel sequences and their geometric applications. In *Theoretical Foundations of Computer Graphics and CAD*, R. Earnshaw, Ed., vol. 40 of *NATO ASI Series*. Springer Berlin Heidelberg, 1988, pp. 253–278.
- [38] VAN LEEUWEN, E. J. Optimization and approximation on systems of geometric objects. Phd thesis, University of Amsterdam, 2009.
- [39] VARADARAJAN, K. Epsilon nets and union complexity. In Proceedings of the Twenty-fifth Annual Symposium on Computational Geometry (New York, NY, USA, 2009), SCG '09, ACM, pp. 11–16.
- [40] VARADARAJAN, K. Weighted geometric set cover via quasi-uniform sampling. In *Proceedings* of the forty-second ACM symposium on Theory of computing (2010), ACM, pp. 641–648.
- [41] ZOU, F., LI, X., GAO, S., AND WU, W. Node-weighted steiner tree approximation in unit disk graphs. *Journal of Combinatorial Optimization* 18, 4 (2009), 342–349.
- [42] ZOU, F., WANG, Y., XU, X.-H., LI, X., DU, H., WAN, P., AND WU, W. New approximations for minimum-weighted dominating sets and minimum-weighted connected dominating sets on unit disk graphs. *Theoretical Computer Science* 412, 3 (2011), 198–208.

## A Missing Proofs

#### A.1 Missing Proofs in Section 4

**Lemma 39.** The central angle of any uncovered arc is less than  $\pi$ .

Proof. We only need prove the arc of a gadget is less than  $\pi$ . As we union all gadgets and add more and more disks in  $\mathcal{H}$ , the central angle of an arc only becomes smaller. So, now we fix a square gadget  $\mathsf{Gg}(\Gamma)$ . Consider the substructure above line  $D_s D_t$ . See the right hand side of Figure 1 for an example. Suppose an arc a with endpoints M and M' on the boundary of  $\mathsf{Gg}$ . The center of the arc is in the central area of  $\mathsf{Gg}$ . If the central angle  $\angle(a) \ge \pi$ , its center should lie in the cap region bounded by a and the chord MM'. Without loss of generality, we suppose M is closer to the line  $D_s D_t$  than M'. Draw a auxiliary line at M which is parallel to  $D_s D_t$ . If the line does not intersect the central area, the center a locates below the line MM' (Hence, outside the cap region), thus the central angle is less than  $\pi$ . If the auxiliary line intersects the boundary of central area at point N. Then, we can see that

$$|MN| = \sqrt{|MD_s|^2 - x^2} + |D_sD_t| - \sqrt{|ND_t|^2 - x^2},$$

where x is the vertical distance between point M and line  $D_s D_t$ . It is not difficult to see that |MN| > 1. It means that the center of the arc locates below the line MN (Otherwise, the distance from the center to M is larger than |NM|, which is larger than 1, rendering a contradiction). So, the central angle of any arc is less than  $\pi$ .

#### A.2 Missing Proofs in Section 7.1

**Lemma 24** Consider two non-empty small squares  $\Gamma$ ,  $\Gamma'$ . Suppose the square gadgets in the two squares are  $\mathsf{Gg}(\Gamma) = (\mathsf{D}_s, \mathsf{D}_t)$  and  $\mathsf{Gg}'(\Gamma') = (\mathsf{D}'_s, \mathsf{D}'_t)$ . The active region pairs  $(\mathsf{Ar}_1, \mathsf{Ar}_2)$  and  $(\mathsf{Ar}'_1, \mathsf{Ar}'_2)$  are associated with gadget  $\mathsf{Gg}(\Gamma)$  and  $\mathsf{Gg}'(\Gamma')$  respectively.  $(\mathsf{Ar}_1, \mathsf{Ar}_2)$  and  $(\mathsf{Ar}'_1, \mathsf{Ar}'_2)$  form a double-mixture. Then, the following statements hold:

- 1. Their corresponding squares  $\Gamma, \Gamma'$  are adjacent;
- 2. The core-central areas of  $\mathsf{Gg}(\Gamma)$  and  $\mathsf{Gg}'(\Gamma')$  overlap;
- 3. The angle between  $D_s D_t$  and  $D'_s D'_t$  is  $O(\epsilon)$
- 4. None of the two core-central areas can overlap with any small squares other than  $\Gamma$  and  $\Gamma'$ .

*Proof.* Suppose the core-central areas of  $\mathsf{Gg}$  and  $\mathsf{Gg}'$  are  $\mathfrak{C}_o$  and  $\mathfrak{C}'_o$  respectively. Because  $(\mathsf{Ar}_1, \mathsf{Ar}_2)$  and  $(\mathsf{Ar}'_1, \mathsf{Ar}'_2)$  form a double-mixture. There exists at least one disk centered in  $\Gamma'$  which appears in both  $\mathsf{Ar}_1$  and  $\mathsf{Ar}_2$ . Thus, the disk is centered in  $\mathfrak{C}_o$ . It means that  $\mathfrak{C}_o$  and  $\mathfrak{C}'_o$  overlap. Hence,  $\Gamma$  and  $\Gamma'$  are adjacent. See Figure 13.

It is easy to see that any core-central area can overlap with at most two squares. Since  $\mathfrak{C}_o$  and  $\mathfrak{C}'_o$  overlap, at least one of them overlaps with both  $\Gamma$  and  $\Gamma'$ . Without loss of generality, suppose  $\mathfrak{C}_o$  overlap with both  $\Gamma$  and  $\Gamma'$ . We only need to prove  $\mathfrak{C}'_o$  cannot overlap with other squares other than  $\Gamma$  and  $\Gamma'$ .

Our proof needs a useful notion, called *apex angle*. Consider a square gadget. See the left one of Figure 1 (and we use the notations there). The line  $D_s E$  and  $D_s F$  are the tangent lines to the



Figure 13: The case that arcs in two different active regions are not order-separable. The two adjacent squares are  $\Gamma = A_1 A_2 A_5 A_6$  and  $\Gamma' = A_2 A_3 A_4 A_5$ . ( $\mathsf{D}_s, \mathsf{D}_t$ ) is the square gadget in  $\Gamma$  and ( $\mathsf{D}'_s, \mathsf{D}'_t$ ) is the square gadget in  $\Gamma'$ . The active regions  $\mathsf{Ar}_1$  and  $\mathsf{Ar}_2$  belong to the  $\mathsf{Gg}(\Gamma)$ , while the  $\mathsf{Ar}'_1$  and  $\mathsf{Ar}'_2$  belong to the  $\mathsf{Gg}'(\Gamma')$ .

boundary of core-central area at point  $D_s$ . We define *apex angle*  $\theta_{\mathfrak{C}_o}$  to be  $\angle ED_sF$ . We notice that  $\theta_{\mathfrak{C}_o}$  only depends on the distance  $|D_sD_t|$ . When the side length of a small square is  $\epsilon$ ,  $\theta_{\mathfrak{C}_o}$  is  $O(\epsilon)$ . In fact, even the core-central area is completely determined by  $D_s$  and  $D_t$ . Thus, we can generalize the concept *core-central area* by using any two overlapping disks in  $\mathcal{H}$ .

Now go back to our proof. See Figure 13. The segments  $D_sD_t$  and  $D'_sD'_t$  cannot intersect because they belong to different small squares. Suppose the squares  $\Gamma$  and  $\Gamma'$  are  $A_1A_2A_5A_6$  and  $A_2A_3A_4A_5$ , respectively.  $A_2A_5$  is the common side. W.l.o.g, suppose  $D'_s$  is closer to line  $A_2A_5$ than  $D'_t$ . If any disk centered in  $\mathfrak{C}'_o$  can appear in  $\operatorname{Ar}_1$  but outside the disk  $D'_s$  (i.e., outside the disks  $D_s \cup D_t \cup D'_s$ ), the core-central areas defined by  $D'_sD_t^{-9}$  and  $\mathfrak{C}'_o$  should overlap nontrivial (not only touch at point  $D'_s$ ). Thus, the angle  $\angle D_tD'_sD'_t$  should less than  $O(\epsilon)$ . It means angle between  $D_sD_t$  and  $D'_sD'_t$  is at most  $O(\epsilon)$ . Moreover, angle between line  $D'_sD'_t$  and  $A_5A_4$  is no less than  $\frac{\pi}{2} - O(\epsilon) > O(\epsilon)$ . Thus,  $\mathfrak{C}'_o$  cannot intersect  $A_5A_4$ . Hence,  $\mathfrak{C}'_o$  cannot overlap other squares.  $\Box$ 

**Lemma 25** Suppose active region pairs  $(Ar_1(A_1), Ar_2(A_2))$  and  $(Ar'_1(A'_1), Ar'_2(A'_2))$  are associated with gadget Gg and Gg' respectively. If  $A_1$  and  $A'_1$  are order-separable, then  $A_2$  and  $A'_2$  are also order-separable.

*Proof.* We prove the lemma by contradiction. We only consider the arcs which have siblings. Suppose  $\mathcal{A}_1$  and  $\mathcal{A}'_1$  are order-separable but  $\mathcal{A}_2$  and  $\mathcal{A}'_2$  are not order separable. W.l.o.g., assume  $a \prec a', \forall a \in \mathcal{A}_1, \forall a' \in \mathcal{A}'_1$ . Since  $\mathcal{A}_2$  and  $\mathcal{A}'_2$  are not order-separable, there exist arcs  $b_1, b_2 \in \mathcal{A}_2$  and  $b' \in \mathcal{A}'_2$  such that  $b_1 \prec b' \prec b_2$ . (If not, there exist arcs  $b \in \mathcal{A}_2$  and  $b'_1, b'_2 \in \mathcal{A}'_2$  such that  $b'_1 \prec b \prec b'_2$ .) Suppose  $a_2$  is the sibling of  $b_2$ , a' is the sibling of b'. Thus,  $a' \prec a_2$  based on the same proof to Lemma 13. It yields a contradiction to the assumption that  $a \prec a', \forall a \in \mathcal{A}_1, \forall a' \in \mathcal{A}'_1$ .  $\Box$ 

 $<sup>{}^{9}</sup>D'_{s}$  and  $D_{t}$  do not locate in the same square. This is core-central area in the generalized sense.

**Lemma 28** Consider the substructure St which contains an active region of square  $\Gamma$ . The substructure can be cut into at most three smaller substructures, by doing label-cut twice. At most one of them contains the active region. Moreover, this new substructure (if any) is bounded by the region  $\mathsf{Dom}(\Gamma^+)$  associated with  $\Gamma$ .

Proof. See Figure 8. We use the same notations defined in Section 7.1. The entire active region  $[\bigcup_{i:D\in\mathfrak{C}_o} \mathsf{D}_i - (\mathsf{D}_s \cup \mathsf{D}_t)] \cap H^+$  is a sub-region of  $(\mathsf{D}(P,2) - \mathsf{D}_s - \mathsf{D}_t) \cap H^+$ . Thus, any disk centered in  $\mathsf{D}(Q_s,1) \cap \mathsf{D}(Q_t,1) \cap H^+$  can cover the active region. So, if there is a disk centered in  $\mathsf{D}(Q_s,1) \cap \mathsf{D}(Q_t,1) \cap H^+$ , we add it into  $\mathcal{H}$  such that the entire active region is covered. If not, we prove we can cut St such that the new substructure contains the active region is bounded by the region  $(\mathsf{D}(D,1) - \mathsf{D}_s - \mathsf{D}_t) \cap H^+$ . The disks centered in  $\mathbb{R}(\Gamma) \cap \mathsf{D}(Q_s,1) - \mathsf{D}(Q_s,1) \cap \mathsf{D}(Q_t,1)$ ) cover point  $Q_s$ . Thus, the arcs of these disks are order-separable to the arcs in active region. Similarly, the arcs of disks centered in  $\mathbb{R}(\Gamma) \cap \mathsf{D}(Q_t,1) - \mathsf{D}(Q_s,1) \cap \mathsf{D}(Q_s,1) \cup \mathsf{D}(Q_t,1)$  be arcs in active region. Thus, we can add two disks in  $\mathcal{H}$  to label-cut St into at most three new substructures. Obviously, there is only one of them containing arcs centered in  $\Gamma - \mathsf{D}(Q_s,1) \cup \mathsf{D}(Q_t,1)$ . Hence, this new substructure is bounded in the region  $(\mathsf{D}(D,1) - \mathsf{D}_s - \mathsf{D}_t) \cap H^+$ .

#### A.3 Missing proofs in Section 7.2

**Lemma 30** In each of the above iterations, one substructure St(b, A) is cut into at most two new substructures. Any original arc in A cannot be cut into two pieces, each of which belongs to a different new substructure.

*Proof.* Figure 9 gives an explanation about the change of substructures before and after the process. Suppose the envelope is  $\{a_1[Q_s, Q_1], \ldots, a_k[Q_{k-1}, Q_t]\}$ . After we add the disk  $\mathsf{D}(a_i)$  in  $\mathcal{H}$ , the baselines of the two new substructures are  $\mathsf{b}_1 = \mathsf{b}[Q_s, P_{i+1}] \cup a_{i+1}[P_{i+1}, Q_i]$  and  $\mathsf{b}_2 = a_{i+1}[Q_{i+1}, P'_{i+1}] \cup \mathsf{b}[P'_{i+1}, Q_t]$ . Since  $a_{i+1}$  is an arc lying on the envelope, there does not exist an arc b with endpoints (A, B) such that  $A \prec P_{i+1} \prec P'_{i+1} \prec B$ . So the endpoints of any arc in  $\mathsf{St}_1$  cannot be in  $\mathbb{R}(\mathsf{St}_2)$ . Thus, any arc cannot be separated into two substructures.

#### A.4 Missing proofs in Section 7.3

**Lemma 40.** Consider two disks  $D_u$  and  $D_l$  in squares  $\Gamma_u$ ,  $\Gamma_l$  respectively. Suppose  $D_u \notin \mathsf{Gg}(\Gamma_u), D_l \notin \mathsf{Gg}(\Gamma_l)$  and  $D_u$  and  $D_l$  intersect at points A and B. Suppose the side length of square is  $\mu$ . If  $\angle AD_uB$  (or  $\angle AD_lB$ ) is more than  $2\sqrt{2\mu}$ ,  $\mathsf{Gg}(\Gamma_u)$  and  $\mathsf{Gg}(\Gamma_l)$  should overlap. Moreover, if intersection A (resp. B) is in  $\mathbb{U}(\mathcal{H})$ , the two arcs which intersect at point A (resp. B) are in the same substructure.

Proof. Suppose  $D_u$  and  $D_l$  intersect at point A and B.  $D_s \in \mathsf{Gg}(\Gamma_u)$ ,  $D'_s \in \mathsf{Gg}(\Gamma_l)$ . Thus,  $D_u, D_s$  locate in the same square. The line  $D_u D_l$  intersect disk  $D_s$  at point E and intersect disk  $D'_s$  at point F.  $|D_u E|$  and  $|D_l F|$  are no less than  $1 - \mu$  according to the triangle inequality. Suppose  $\angle AD_u B$  is  $\theta$ . Then the length  $|D_u D_l|$  equals  $2\cos(\frac{\theta}{2})$ . Since  $\theta > 2\sqrt{2\mu}$ , we have  $|D_u E| + |D_l F| > |D_u D_l|$ , i.e.  $D_s$  and  $D'_s$  overlap. Figure 14 illustrates the situation. If A (or B) is in  $\mathbb{U}(\mathcal{H})$ , it is obvious that the two arcs which intersect at A (or B) can cover the same point on the baseline (i.e., the intersection point of  $D_s$  and  $D'_s$ ) which means they are in the same substructure.

Based on the Lemma 40, we prove Lemma 32 as follows.



Figure 14:  $D_u$  and  $D_l$  intersect at point A and B. Suppose the side length of square is  $\mu$ . If the central angle of arc a[A, B] is more than  $2\sqrt{2\mu}$ .  $D_s, D'_s$  should overlap, where  $D_s \in \mathsf{Gg}(\Gamma_u), D'_s \in \mathsf{Gg}(\Gamma_l)$ .

**Lemma 32** We are given a substructure St(b, A).  $a \in A$  and two arcs  $b_1, b_2 \notin A$ . If  $b_1, b_2$  cover the same point on  $a, b_1, b_2$  should belong to the same substructure.

Proof. We prove the lemma by contradiction. Suppose the arcs  $a, b_1, b_2$  are parts of  $D, D_1, D_2$  respectively. Consider a point P on D that can be covered by all of  $a, b_1, b_2$ . D and  $D_1$  intersect at  $A_1$  and  $B_1$ , meanwhile D and  $D_2$  intersect at  $A_2$  and  $B_2$ . See Figure 15. Since D and  $D_1$  belong to different substructures, we know  $\angle A_1 DB_1$  is less than  $2\sqrt{2\mu}$  based on Lemma 40. Similarly,  $\angle A_2 DB_2$  is less than  $2\sqrt{2\mu}$ . It is easy to see that  $\angle D_1 DD_2$  is less than  $2\sqrt{2\mu}$ . When  $\angle D_1 DD_2$  is  $O(\sqrt{\epsilon})$  when  $\mu = O(\epsilon)$ .  $|D_1 D_2|$  is less than 1 as  $|DD_1|$  and  $|DD_2|$  are more than 1 and less than 2. Suppose  $A_3$  and  $B_3$  are the two intersection points of  $D_1$  and  $D_2$ . Thus, we have  $\angle A_3 D_1 B_3 > \pi/3 > 2\sqrt{2\mu}$ . Moreover, P is in  $\mathbb{U}(\mathcal{H})$ . By Lemma 40, we can see that it contradicts the fact that  $D_1$  and  $D_2$  belong to two different substructures.



Figure 15: Consider a point P on D.  $a, b_1, b_2$  which belong to disks  $D, D_1, D_2$  respectively, can cover the point P. D and  $D_1$  intersect at  $A_1$  and  $B_1$ , meanwhile D and  $D_2$  intersect at  $A_2$  and  $B_2$ . Then,  $D_1$  and  $D_2$  belong to the same substructure.