LA-UR-99-4101
Approved for public release;
distribution is unlimited.

Title:

Point Set Labeling with Specified Positions

Author(s):

Srinivas R. Dotti, Madhav V. Marathe,
and Bernard M. Moret

Submitted to:

http://lib-www.lanl.gov/cgi-bin/getfile?00796865.pdf

International Journal of Foundations of Computer Science
© World Scientific Publishing Company

Point Set Labeling with Specified Positions

Srinivas R. Doddi*
Timbre Technologies, Inc, A TEL Company
Fremont, CA, 94538, USA

Madhav V. Marathe†
Basic and Applied Simulation Sciences (D-2), Los Alamos National Laboratory,
MS M997, Los Alamos, NM, 87544, USA

and

Bernard M. Moret‡
Depatartment of Computer Science, University of New Mexico,
Albuquerque, NM, 87131, USA

Received (received date)
Revised (revised date)
Communicated by Editor's name

ABSTRACT

Motivated by applications in cartography and computer graphics, we study a version of the map-labeling problem that we call the *k-Position Map-Labeling Problem*: given a set of points in the plane and, for each point, a set of up to k allowable positions, place uniform and non-intersecting labels of maximum size at each point in one of the allowable positions. This version combines an aesthetic criterion and a legibility criterion and comes close to actual practice while generalizing the fixed-point and slider models found in the literature.

We present a general heuristic that given an $\epsilon > 0$, runs in time $O(n \log n + n \log(R^*/\epsilon)\log(k))$, where R^* is the size of the optimal label, and guarantees a constant approximation for any regular labels. For circular labels, our technique yields a $(3.6 + \epsilon)$-approximation, improving in the case of arbitrary placement over the previous bound of approximately 19.5 obtained by Strijk and Wolff [28]. We then extend our approach to arbitrary positions, obtaining an algorithm that is easy to implement and also substantially improves the best approximation bounds. Our technique combines several geometric and combinatorial properties, which may be of independent interest.

1. Introduction and Summary of Results

[*] srini@timbrecom.com;most of work was done while the author was a graduate student in the Dept. of Computer Science at the University of New Mexico and a research associate at Los Alamos National Laboratory.

[†] marathe@lanl.gov; research supported by the Department of Energy under Contract W-7405-ENG-36.

[‡] moret@cs.unm.edu; research supported in part by ONR grant N00014-97-1-0244 and NSF grant ITR 00-81404.

The problem of automated label placement has received considerable attention in the computational geometry community [2,13,22,24,26,28] due to its applications in the areas of cartography [3,10,14,20] and computer graphics [4,5]. For example, the ACM Computational Geometry Task Force [2] has targeted it as one of the important areas of research in Discrete Computational Geometry. We refer the reader to the Map Labeling Bibliography website [35], for comprehensive information on this subject.

As pointed out by Marks and Shieber [24], four concerns are of particular importance in placing labels: (1) the degree to which labels overlap with each other—non-intersecting labels are the most fundamental requirement [13]—and obscure cartographic features; (2) the degree to which labels are clearly and unambiguously associated with the features they identify; (3) *a priori* preferences among a canonical set of potential label positions; and (4) the number of features left unlabeled. In general, cartographers first obtain computer solutions and then use their know-how to refine the solutions in order to improve aesthetics—a painstaking and time-consuming task.

Models for the study of label-placement problems can be broadly classified into three types: fixed-position models, slider models, and arbitrary-orientation models. (For more details, see [1,8,22].) We generalize these models with a model in which the user can specify a set of k allowable positions for each point. It is crucial to note that k is not fixed in advance, but can be specified by the user, so that the k-position model indeed generalizes fixed-position and slider models and, for arbitrarily large k (specified using a suitable succinct specification), also subsumes the arbitrary-position models. Formally, an instance of the *k-Position Map-Labeling (KPML)* problem consists of a set of points, and, for each point, a set of k allowable label placements. The goal is to place a label for each point (with the point lying on the boundary of the label) in one of the allowable placements so as to maximize the size of the labels. In the slider model there is no constraint on k; in the fixed-position models, k is fixed. In contrast, the model proposed here allows the k positions to be specified by the user; in particular, by allowing $k = \infty$ we get the slider model and by setting $k = 4$ we get the model of Formann and Wagner's.

Our model reflects minimal constraints on aesthetics and association of labels with point features (as expressed by the allowed placements), while maximizing legibility (as expressed by overall size). For expository purposes, we focus on uniform circular labels; in a later section, we briefly outline how our technique extends directly to any regular polygonal labels (albeit with slightly worse performance guarantees).

Recall that an approximation algorithm for a maximization problem Π provides a *performance guarantee* of $\rho \geq 1$ if, for every instance I of Π, the solution value returned by the approximation algorithm is at least $1/\rho$ times the optimal value of I. Our main result is an efficient and easily implementable polynomial-time approximation algorithm with a performance guarantee of 3.6 for the KPML problem restricted to circular labels. This result has two important extensions that are further discussed in Section 9.

- As our analysis shows, our algorithm works even for unbounded k without any loss in performance, yielding a substantial improvement over the previous bounds of roughly 30 by Doddi *et al.* [8] and of approximately 19.5 by Strijk and Wolff [28].

- By using a circumscribed regular polygon and an inscribed regular polygon as lower and upper bounds, the algorithm yields a polynomial-time approximation (with slightly worse performance guarantees) for the KPML problem when restricted to any regular polygon. In fact, the algorithm works even if we are allowed a fixed set of regular polygons as surrogates for labels, with each point having a different set of allowable positions.

Our approach is motivated by a similar approach taken by Formann and Wagner [13] to transform a 4-position map-labeling problem to instances of *2SAT*; in Section 3 we discuss why their idea cannot be extended directly to apply to our problem. Our technique combines several combinatorial and geometric properties that characterize the structure of label placements, properties that may be of independent interest.

Our paper is organized as follows. In Section 2, we briefly review past results. In Section 3, we present the basic idea behind the algorithm. Section 4 gives definitions and notation as well as two crucial results—which allow us to conduct local searches only. Section 5 develops a number of lemmata on the geometric relationships inherent in the problem. Section 6 gives structural characterizations of the problem and relates them to the geometry of the problem. In Section 7, we use all of these results to develop an algorithm that selects two positions for each point and show that the selection always contains a feasible solution if any exists. In Section 8, we give the main algorithm. Finally, we present extensions of our results in Section 9 and concluding remarks in Section 10.

2. Related Literature

Automated map labeling has been studied for nearly three decades in the cartography community. Current practical approaches typically include combinations of techniques such as mathematical programming, gradient descent, simulated annealing, etc.; a comprehensive survey can be found in Christensen *et al.* [4]. A comprehensive bibliography and other pertinent information on map labeling can be found at the web site [35].

Formann and Wagner [13] studied the problem of labeling n points with uniform and axis-aligned squares. They gave a $O(n \log n)$ algorithm with performance guarantee of 2 and showed that this guarantee cannot be improved unless we have $\mathsf{P} = \mathsf{NP}$. Kucera *et al.* [23] gave exact algorithms to solve this problem; one of their algorithms runs in time $O(4^{\sqrt{n}})$ and returns an optimal solution.

Doddi *et al.* [8] considered two label-placement problems: maximizing label size and maximizing the number of labeled points. For the problem of maximizing label size, they gave constant-factor approximation algorithms with performance guarantee of $8(2+\sqrt{3})$ (~ 30) for circular labels and $8\sqrt{2}/\sin(\pi/10)$ (~ 37) for square labels. For the problem of maximizing the number of labeled points subject to placing labels of specified size, they developed a bicriteria approximation in which at least $(1-\epsilon)\cdot n$ labels are placed, each of size at least $(1-c\cdot\epsilon)$ times the optimal label, for some positive constant c. Strijk and Wolff [28] improved the algorithm of Doddi *et al.* for circular labels, obtaining an approximation ratio of approximately 19.5; they also proved that the problem is APX-hard.[a] Informally, the results in [28] and [8] is based on the following observation: letting D_3 denote the smallest diameter of any three point subset of the input points, it is possible to labeling the input points using labels that are a constant fraction of D_3. The above observation allows us to label a point using a local search, i.e. considering only points within certain neighborhood distance. The basic difference between the work of [28] and [8] is in the way the labels are placed. Intuitively, while placing labels for two points p and q with minimal distance, [8] place labels for p and q that are as far apart as possible, [28] place labels for these points that are as close as possible.

While these efforts aimed at maximizing label size, a number of researchers have also worked on maximizing the number of points that can be simultaneously labeled by labels of a given size—the dual version of the size problem. This problem can be cast as an independent set problem in an appropriately defined intersection graph, so that results for finding independent sets in such graphs immediately yield analogous results for maximizing the number of labeled points [1,8,11,18,22]. Agarwal *et al.* [1] presented $O(\log n)$- and $(1+\epsilon)$-approximations when the labels are, respectively, nonuniform and axis-aligned with uniform height. Kreveld *et al.* [22] proposed the slider model, a direct extension of the model proposed in [8], and presented two algorithms for maximizing the number of labelled points under this model: a fast heuristic with a performance guarantee of 2 and a polynomial-time approximation scheme, both restricted to the case of rectangular labels of the same height. Recently Erlebach, Jansen and Seidel [11] devised an elegant polynomial-time approximation scheme for finding the maximum independent set of a set of circles; as observed above, such an algorithm immediately yields one for maximizing the number of points labelled with arbitrarily-sized circular labels.

Finally, a number of research papers have addressed extensions of the basic model. These extensions include: (i) placing multiple labels per point; (ii) labeling edge and point features; and (iii) associating weights with points and maximizing a weighted objective. We refer the readers to [19,27,26,34,29] and the references therein.

3. The Basic Idea

Definition 1 *Given a set S of points in the plane and, for each point $a \in S$, a set X_a (with $|X_a| \leq k$) of possible label placements, the k-*Position Map Labeling (KPML) *problem is to identify the largest $R > 0$ such that, for each point $a \in S$, a label of size*[b] *R can be placed at one of the positions in X_a such that no two such labels intersect.*

For the rest of the paper, we will assume that our labels form topologically open

[a] Informally speaking an APX-hard problem cannot have a polynomial-time approximation scheme unless P = NP. See [25] for more details.

[b] In the case of circular labels, size is taken to be the radii of the circles; in the case of regular polygons, size is taken to be the length of the sides.

sets. The position of a circular label of a given size that must include a given point on its perimeter is fully specified by the angle made by the line passing through the point and the center of the circle. Thus we shall assume that positions are given as angles measured counterclockwise with respect to (w.r.t.) the positive abscissa. Note that, in our development, a position, unless otherwise specified, can be any angle whatsoever—it need not be limited to the *allowable* positions specified in the input.

Our definition can be extended to regular polygons. In such a case, we need an angle and also allowable orientations for the polygonal label. As stated earlier, we focus here on circular labels in order to simplify the exposition.

The main result of this paper can be viewed as a polynomial-time approximate reduction to the *2SAT* problem. Thus our technique generalizes the idea of Formann and Wagner [13], who reduced the problem of placing uniform and axis-aligned squares to the *2SAT* problem.

Let S denote the given set of points, R^* the size of labels (radius or edge length) in an optimal solution, and $\rho > 1$ some constant.

- A candidate label of size σ labeling point $a \in S$ is called *ρ-dead* if a label of size $\rho \cdot \sigma$ placed in the same position contains some other point $b \in S$, $b \neq a$. If we have $\rho \cdot \sigma \leq R^*$ and a candidate label of size σ is ρ-dead, then the position used by that label cannot be used in an optimal solution.
- A candidate label of size σ labeling point $a \in S$ is called *safe* if it does not intersect any label of equal size labeling (in any position) a different point of S. Clearly, if there exists a safe label, then it can be added to the solution without worrying about the placement of labels at other points.
- A candidate label of size σ labeling point $a \in S$ is called *ρ-pending* if it is neither ρ-dead nor safe. A ρ-pending label of size σ labeling point $a \in S$ may intersect only with ρ-pending labels of other points in S.

The approximation algorithm uses the concept of a *ρ-relaxed procedure* and the corresponding certificates of failure as formulated by Hochbaum and Shmoys [16]. Informally speaking, a polynomial-time $(\rho + \epsilon)$-relaxed procedure TEST for a maximization problem Π (where the optimal value for instance I is denoted by R^*) has the following structure: given a candidate solution with value M, TEST either outputs a "certificate of failure," implying $R^* < (\rho + \epsilon) \cdot M$, or succeeds, implying that the heuristic solution value is at least M. Such a procedure can then be used to obtain a ρ-approximation for a combinatorial problem in the following way. Let the optimal solution take a value from the range $[LB, UB]$. A binary search over $[LB, UB]$, in steps of ϵ',[c] yields the largest value M such that TEST with candidate solution value M succeeds but TEST with candidate solution value $(M + \epsilon')$ fails. Letting $HEU(I)$ denote the heuristic solution, we can thus write

[c] In their original paper [16] applied the technique to graph problems. In their case, they needed to search an ordered domain with polynomially many values. Thus they did not have to do binary search. In our case the domain can be exponentially large. Hence we need to do binary search over the given ordered domain in steps of ϵ' to obtain an efficient polynomial algorithm.

$HEU(I) = M$ and $R^* \leq \rho \cdot (M + \epsilon')$, yielding a performance of $\rho + \epsilon$, for any desired ϵ value (at the expense of an appropriate increase in the running time). This procedure is a polynomial-time algorithm for the KPML problem, but it is not strongly polynomial-time, because because its running time depends on ϵ. Section 8.1 analyzes the exact running time of our procedure.

Although Formann and Wagner [13] do not state it explicitly (nor use the terminology presented above), their algorithm can be viewed as an instance of this general schema. Their algorithm starts by placing infinitesimally small and equally-sized candidate labels at each of the four positions of each point. At each step, the size of each label is uniformly increased; any ρ-dead label is removed and its corresponding position eliminated. In the case of square, axis-aligned labels that must touch the labeled point at one corner, Formann and Wagner showed that, for $\rho \geq 2$, there are at most two ρ-pending labels. Using this fact, a *2SAT* instance is constructed and solved. The process is repeated until the *2SAT* instance is no longer satisfiable; the last feasible solution found is then returned. The transformation to a *2SAT* formula combined with a procedure for solving the *2SAT* problem forms a 2-relaxed procedure in the sense of Hochbaum and Shmoys and yields a performance guarantee of 2.

The *2SAT* instance itself simply describes, using implications, the possible intersections among ρ-pending labels. Since there are at most two possible positions per point, the choice at each point can be encoded by a single Boolean variable. Let x_a and x_b denote the variables corresponding to points $a \in S$ and $b \in S$, respectively, where x_a is set to true iff the first of the two ρ-pending labels for point a is chosen (and similarly for point b). If, say the first ρ-pending label for a intersects with the second ρ-pending label for b, this is encoded with the implication $x_a \to x_b$, or, in *2SAT* form, the clause $\{\overline{x_a}, x_b\}$. It is easily verified that a feasible solution exists for the labeling problem iff the constructed *2SAT* instance is satisfiable.

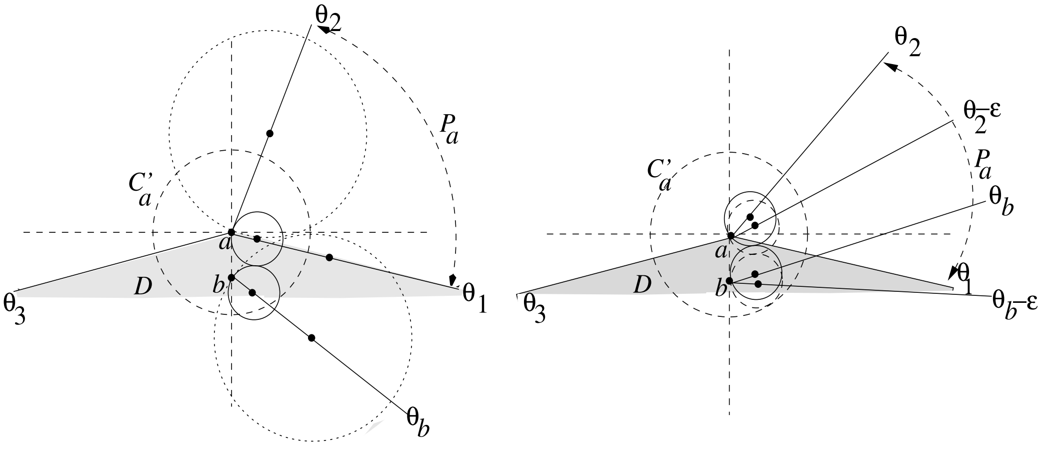
Our main algorithm also uses the idea of reduction to *2SAT*. However, the number of ρ-pending positions for the KPML problem is much larger than 2—and, with just $k = 3$, a straightforward application of Formann and Wagner's technique will yield an instance of *3SAT*, which is of course NP-hard. Our main contribution is a selection technique that combines several geometric and combinatorial properties to select at most 2 feasible positions for each point—at the cost of using a slightly larger ρ (in the case of circular labels, we use $\rho < (3.6 + \epsilon)$). The selection procedure combined with an algorithm for solving *2SAT*, yields the required ρ-relaxed procedure.

In broad outline, our selection procedure is based on three crucial geometric properties of (sub)optimal solutions. We call a position *dead*, *safe*, *pending* if the label placed at that position is dead, safe, or pending, respectively.

Packing property: The first property concerns the packing of labels in (sub)optimal solutions to the KPML problem. For the sake of analysis, we can ignore safe positions. (This is due to the fact that we can always place a label at a safe position regardless of the placement of labels at other points.) Pick $a \in S$ and let C'_a denote the circle of radius $0.8 \cdot R^*$ centered at a. Let $S'_a \subset S$ denote the set of all points of S that lie inside C'_a. We show that, while placing labels at a, we can ignore any point of S that lies outside C'_a. This is a crucial result, since it allows us to conduct a local search within S'_a only; using a packing argument, we further show $|S'_a| \leq 4$. Sections 4 and 6 provide the necessary details.

Bounding dead and pending regions: The second property allows us to combine contiguous dead, or pending, positions into regions and allows us to bound the number of regions of each type. To begin with, we identify and eliminate all dead positions of a. We call a contiguous set of dead positions a *dead region*. We consider only maximal dead regions, in the sense that no two such regions share a dead position. Thus any two dead regions must be separated by a region of pending positions, which we call a *pending region*. We calculate the minimum angle of a dead region and show that the number of dead regions (and thus also of pending regions) is at most 2. Our aim is to select at most one position from each pending region, thereby allowing us to encode the problem as an instance of *2SAT*. Section 5 discusses this part in detail.

Equivalent pending positions: Intuitively, the third property allows us to define "equivalent positions" in pending regions, thereby allowing us to choose any one of these positions for constructing the *2SAT* instance. Let $b \in C'_a$ as in Figure 1.



(a)

(b)

Figure 1: A clique-region and a uniform-region of point a.

As previously discussed, b must be surrounded by a dead region of a, which we denote by $D = Cone(a, \theta_3, \theta_1 - \varepsilon)$, where $Cone(a, \theta_3, \theta_1)$ denotes a contiguous set of ρ-dead positions between θ_3 and $\theta_1 - \varepsilon$ for infinitesimally small ε (using anti-clockwise angles measured from the positive abscissa, with $\theta_3 \leq \theta_1$). Let $P_a = Cone(a, \theta_1, \theta_2)$ be a pending region, a conical section of ρ-pending positions between θ_1 and θ_2 in anti-clockwise direction. P_a has one of its boundaries, θ_1, adjacent to the dead region D surrounding b. We show that the positions in the conical region P_a form one of two types of equivalence classes, a *clique-region* or a *uniform-region*.

- The set of positions in the conical region P_a forms a *clique-region* w.r.t. b if, for each ρ-pending position θ_b of b, a label of size R^*/ρ placed at θ_b touches (i.e., intersects with an intersection of null area) a label of the same size placed at some $\theta_1 \in P_a$, while a ρ-enlarged label (of size R^*) placed at θ_b *intersects* (with nonzero area) every ρ-enlarged label placed at any position inside P_a.
- Similarly, the set of positions in the conical region P_a forms a *uniform-region* w.r.t. b if there exists a ρ-pending position θ_b at b such that a label of size R^*/ρ placed at θ_b intersects a label of same size placed at any position in P_a.

Since each such set is adjacent to a dead region, for each point b in C'_a we have two *clique-regions* of a w.r.t. b and two *uniform-regions* of a w.r.t. b—one on either side of the vertical line as shown in Figure 1. Note that

> No optimal solution can simultaneously place a label at positions θ_b and $\theta_1 \in P_a$ since they intersect each other.

It is crucial to note the difference between these two types of regions.

- A clique-region is defined w.r.t. two positions, θ_1 of a and θ_b of b. Consequently, the entire P_a can be treated as a single position and hence any position in P_a can be selected and the remaining positions can be ignored w.r.t. b.
- In contrast, a uniform-region P_a is defined w.r.t. only θ_b of b, so that an optimal solution may place a label at $\theta_b - \varepsilon$ for the point b and a label at θ_2 for the point a (refer to Figure 1). Thus only θ_2 can be selected and the remaining positions can be ignored w.r.t. b.

Selection procedure: Our selection criterion depends on whether a set is a clique-region or a uniform-region. Since, by definition, both clique-regions and uniform-regions share a boundary position (the position θ_1), one of these two sets of a w.r.t. b must properly contain the other. Therefore, θ_2 is always a feasible position w.r.t. b. In Section 5, we show that, if we have $\delta(a,b) > 0.4 \cdot R^*$, where $\delta(a,b)$ denotes the Euclidean distance between a and b, then the corresponding clique-region of a w.r.t. b properly contains the uniform-region of a w.r.t. b. This is an important observation, since it allows us to treat a uniform-region of a w.r.t. b as a clique-region of a w.r.t. b.

Suppose C'_a has one more input point, say c. There are two possibilities: c lies in the same dead region that is surrounding b or c is surrounded by another dead region. In the first case, it is easy to select two positions that are feasible w.r.t. both b and c, as the clique-region and uniform-region of a w.r.t. b and the clique-region and uniform-region of a w.r.t. c share the same boundary position, namely θ_1. However, when c lies in a different dead region, then the positions of a that are considered feasible w.r.t. b may not be feasible w.r.t. c—this is the case where we need a selection procedure.

Since uniform-regions of a are defined w.r.t. a given point, each point could have a number of uniform-regions associated with it; the same holds for clique-regions. It turns out that adjacent clique-regions or adjacent uniform-regions can be merged to form larger conical regions. Informally, a **P**-region of a is the union of all clique-regions and uniform-regions of a that share the same boundary position (θ_1), as illustrated in Figure 2. A **P**-region of a is thus the superset of least angle covering all of the clique-regions and uniform-regions of a w.r.t. to the the points that lie inside the same dead region (D). Thus a **P**-region must be a clique-region, a uniform-region, or both w.r.t. at least one of these input points; we call this point a *reference point* of the **P**-region. Furthermore, a **P**-region is maximal in the sense that it is not properly contained in any **P**-region, dead region, clique-region, or uniform-region. If $Cone(a, \theta_1, \theta_2)$ is a **P**-region, then θ_2 can be considered as a feasible position w.r.t. to the points that lie inside D.

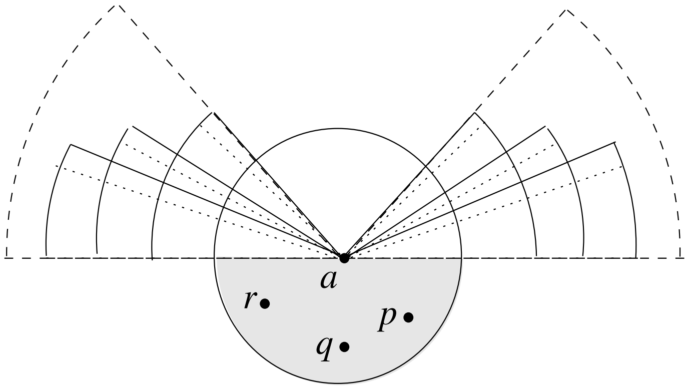


Figure 2: Schematic diagram illustrating **P**-regions. The shaded region corresponds to the **D**-region of a obtained by combining the dead regions w.r.t. p, q, and r. The dashed cones represent the final **P**-regions obtained by combining the maximal uniform/clique-regions w.r.t. p, q, and r.

Since, by definition, a **P**-region is adjacent to a dead region, the number of **P**-regions of a depends on the number of its dead regions. If a has one dead region, it can have at most two **P**-regions, and if a has two dead regions, it can have at most four **P**-regions—see Figure 3. We study the properties of **P**-regions in Section 5. Section 7 contains a detailed discussion of how to select two positions from the **P**-regions that are feasible w.r.t. every input point that lies inside C'_a.

4. Definitions and Preliminaries

In this section, we describe our notation and preliminary results used in the rest of the paper. Figure 4 illustrates our notation.

$\delta(a, b)$ denotes the Euclidean distance between points $a, b \in S$.

$C(a, \theta, R)$ denotes the labeling circle of radius R labeling point $a \in S$ in position θ (which may or may not be in the allowed set of positions X_a).

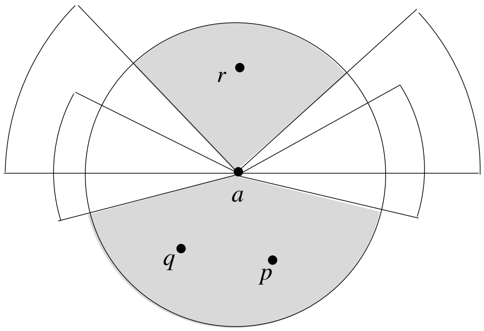


Figure 3: Schematic diagram illustrating **P**-regions and **D**-regions. We have two **D**-regions (shaded): the top one is w.r.t. r and the bottom w.r.t. p and q.

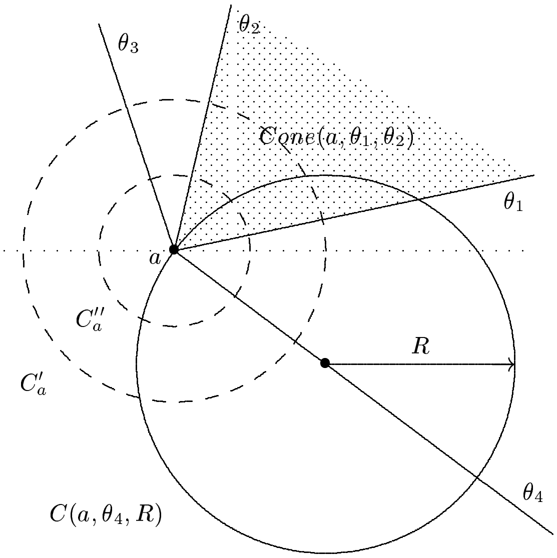


Figure 4: Our notation

R^* denotes the radius of the labeling circles in the optimal solution.

C'_a denotes the circle (not a label) of radius $0.8R^*$ centered at $a \in S$; similarly C''_a and C'''_a denote the circles of radius $0.4R^*$ and $0.1R^*$ centered at $a \in S$ respectively. (The constants produce the desired bounds in later lemmata.)

$N(C'_a)$, $N(C''_a)$ and $N(C'''_a)$ denote the number of points (other than a) of S that lie inside C'_a, C''_a and C'''_a respectively.

$Angle(\theta_i, \theta_j)$ denotes the angle between θ_i and θ_j in counterclockwise direction, starting from θ_i.

$Cone(a, \theta_1, \theta_2)$ denotes the conical region containing positions between θ_1 and θ_2, including θ_1 and θ_2, such that $\theta_1 < \theta < \theta_2$.

ε denotes an infinitesimally small positive value.

We now formalize the definitions introduced in Section 3. We use $\rho > 1$ to denote the approximation ratio; later, we shall fix $\rho = 3.6$.

Definition 2 *Let $a \in S$ and let θ be a position with respect to a (not necessarily in X_a). We call θ* dead *if there exists $b \in S - \{a\}$ such that $C(a, \theta, R^*)$ contains $b \in S$. We call θ* ρ-safe *if for all $b \in S - \{a\}$ and for any angle γ, $C(a, \theta, R^*/\rho)$ does not intersect the circle $C(b, \gamma, R^*/\rho)$ (a circle placed at any point $b \in S$ distinct from a). We call θ* ρ-pending *if it is neither dead nor ρ-safe.*

A position θ is *dead* if an optimal solution (using labeling circles of size R^*) cannot use it. In contrast, an approximation algorithm with performance ρ can safely place a labeling circle of size R^*/ρ at a *ρ-safe* position, regardless of chosen positions of labeling circles of equal size labeling other points. Finally, *ρ-pending* positions are those that may be used to place a labeling circle of size R^*/ρ only for certain placements of other labeling circles (of the same size) at other points. Note the different quantifiers in the definitions of dead and pending positions: if a position is dead, it does not matter which point in the input was responsible, but pending positions are defined with respect to particular points and positions.

We now show that there is a minimum separation beyond which two points can be handled independently of each other in an approximate solution. From here on, in order to simplify our notation, we assume without loss of generality that points a and p share the same ordinate.

Lemma 1 *Assume $\rho \geq 3.6$. Let $a, p \in S$ with $p \notin C'_a$ and let θ_a be a ρ-pending position of a. Then any position θ_p of p such that $C(p, \theta_p, R^*/\rho)$ intersects $C(a, \theta_a, R^*/\rho)$ is a dead position.*

Proof. Refer to Figure 5. Let a' and a'' denote the centers of $C(a, \theta_a, R^*/\rho)$ and $C(a, \theta_a, R^*)$ respectively, and let p' and p'' denote the centers of $C(p, \theta_p, R^*/\rho)$ and $C(p, \theta_p, R^*)$ respectively. We proceed to show that, for any $\delta(a, p) \geq 0.8R^*$, we have $\delta(a, p'') \leq R^*$, which implies that θ_p is a dead position.

$\delta(a, p'')$ is maximized by maximizing θ_a and minimizing θ_p. In turn, θ_a is maximized just as the position that it denotes becomes dead, so that we can assume that θ_a is ε away from being dead, for arbitrary small $\varepsilon > 0$. Therefore p lies just outside $C(a, \theta_a, R^*)$; since ε is infinitesimal,[d] we simply assume that p lies on the perimeter of $C(a, \theta_a, R^*)$. The triangle $aa''p$ is thus isosceles; note that, if the line pp'' intersects that triangle, we are done, since we must then have $\delta(a, p'') \leq \delta(p, p'') = R^*$. (Equality occurs when we actually have $a'' = p''$.) Thus we need only show that, whenever the line pp'' lies outside that triangle, no intersection of the two ρ-scaled labels can occur.

The farthest extent of $C(a, \theta_a, R^*/\rho)$ when projected onto the ap segment is one radius (or $5R^*/18$ with our choice of ρ) plus the projection of the segment aa', or $2R^*/18$; similarly, the farthest extent of $C(p, \theta_p, R^*/\rho)$ when projected onto the ap segment occurs when the line pp'' is (nearly) aligned with pa'' and is then also one radius plus the projection of the segment pp' (minus some infinitesimal constant), for a contribution of $7R^*/18$. Thus the projection of the two circles onto

[d] Many of the sets we define in this paper are open sets; in all cases, we treat them as closed sets in order to derive bounds.

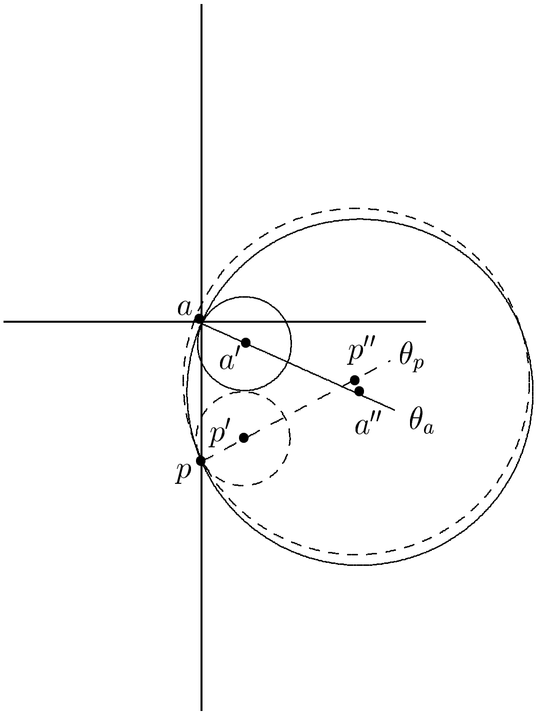


Figure 5: Illustration of Lemma 1.

the segment ap (which has length $0.8R^*$) spans at most $14R^*/18 < 0.8R^*$, so that the two circles do not intersect. This implies that if θ_p is a position such that $C(p, \theta_p, R^*/\rho)$ intersects $C(a, \theta_a, R^*/\rho)$, then $\delta(a, p'') < R^*$. □

The lemma is stated around a point a, but implies a dead position for point p. This lemma is crucial in our development, as it implies that, while placing a label (circle) of size R^*/ρ ($\rho = 3.6$) at point a, we can safely ignore any points outside C'_a and thus restrict our scope to a strictly local search. We formalize this important observation in the next theorem.

Theorem 1 *Let $a, p \in S$ with $p \notin C'_a$. Then we can ignore p while selecting feasible positions $\theta_a \in X_a$ to place a label (circle) of radius R^*/ρ ($\rho = 3.6$).*

Proof. We need to consider two possibilities: (i) for all $\theta_p \in X_p$, $C(p, \theta_p, R^*/\rho)$ does not intersect $C(a, \theta_a, R^*/\rho)$ or (ii) there exists a $\theta_p \in X_p$ such that $C(p, \theta_p, R^*/\rho)$ intersects $C(a, \theta_a, R^*/\rho)$. In the first case, since the label $C(a, \theta_a, R^*/\rho)$ does not intersect with any possible label w.r.t. p, p can be ignored. In the second case, consider a position $\theta_p \in X_p$ satisfying the condition; then by Lemma 1, this position is a dead position and hence can never be used for placing a label. Thus all such positions can be safely ignored. □

Consider a point $p \in C'_a$. Suppose there exists no pending position $\theta_p \in X_p$ such that the corresponding circle $C(p, \theta_p, R^*/\rho)$ intersects the circle $C(a, \theta_a, R^*/\rho)$, for any pending position $\theta_a \in X_a$. Then the point p can also be ignored, as it does not affect the placement of a label of size R^*/ρ at a. From here on, we assume that, for each point $p \in C_a$, there exists a pending position θ_p such that $C(p, \theta_p, R^*/\rho)$ intersects the circle $C(a, \theta_a, R^*/\rho)$ for some pending position $\theta_a \in X_a$.

In the remaining sections, we assume $\rho = 3.6$ (and thus drop the ρ from terms like safe or pending, although we still use it in some equations in order to show where the constants come from) and, when working on the labeling of point a, restrict our attention to points within C'_a—i.e., to points within $0.8R^*$ of a.

5. Some Interesting Conical Regions

This section elaborates on the second and third geometric ideas discussed in Section 3, namely combining contiguous dead or pending positions into regions and obtaining upper bounds on the number of regions of each type. In subsection 5.1 we define formally **D**-regions—obtained by combining overlapping dead regions created by distinct points in the input; the main result of this subsection is summarized in Corollary 1: for any given point $a \in S$, there exist at most two **D**-regions. Subsection 5.2 formally defines clique-regions and uniform-regions. Just as dead regions created by distinct points can be combined to form **D**-regions, we can combine overlapping uniform-regions or clique-regions w.r.t. distinct points to form **P**-regions.

5.1. The **D***-regions and their Properties*

We extend Definition 2 to a conical region $Cone(a, \theta_1, \theta_2)$. We first consider a region formed by a contiguous set of dead positions.

Definition 3 *Pick $a \in S$. Then $Cone(a, \theta_1, \theta_2)$ is a* maximal dead conical region *or* **D***-region, whenever*

1. *every position θ, $\theta_1 \leq \theta \leq \theta_2$, is dead w.r.t. to some $p \in C'_a$; and*
2. *neither $\theta_1 - \varepsilon$ nor $\theta_2 + \varepsilon$ is dead.*

The second condition indicates that any two **D**-regions are separated by at least one non-dead position. If some point p is located within C'_a, then it must be surrounded by a **D**-region, as illustrated in Figure 6.

We now determine the minimum angle of a **D**-region, which will enable us to bound the number of **D**-regions and other types of regions that can exist for a point in S.

Lemma 2 *The minimum angle of a* **D***-region is* $132.8°$.

Proof. Pick $a, p \in S$ with $p \in C'_a$ and denote by $D = Cone(a, \theta_1, \theta_2)$ the **D**-region of a w.r.t. p (see Figure 6). For any $\varepsilon > 0$, the point p must lie outside both $C(a, \theta_1 - \varepsilon, R^*)$ and $C(a, \theta_2 + \varepsilon, R^*)$. It is easily seen that, as $\delta(a, p)$ increases, $Angle(\theta_1, \theta_2)$ decreases, so that $Angle(\theta_1, \theta_2)$ is minimized when p lies on the perimeter of C'_a. Let a' denote the center of $C(a, \theta_1 - \varepsilon, R^*)$, for any fixed $\varepsilon > 0$; note that we have $\delta(p, a') > \delta(a, a') = R^*$. By the law of cosines, we can write

$$\cos(\angle a'ap) = \frac{\delta(a,p)^2 + \delta(a,a')^2 - \delta(a',p)^2}{2\delta(a,p)\delta(a,a')} \tag{1}$$

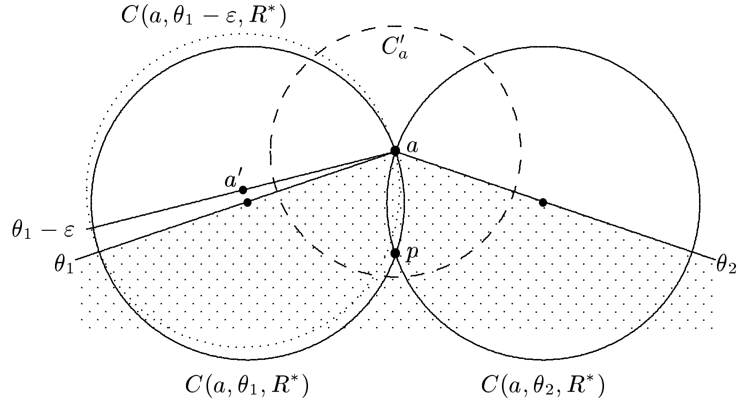


Figure 6: A **D**-region (shown in shaded) of a surrounding p

Substituting known values yields

$$\cos(\angle a'ap) < \frac{\delta(a,p)^2 + {R^*}^2 - {R^*}^2}{2\delta(a,p)R^*} = \frac{\delta(a,p)}{2R^*} \tag{2}$$

Because p lies in C'_a, we have $\delta(a,p) \leq 0.8R^*$; substituting, we get $\cos(\angle a'ap) < 0.4$ and thus $\angle a'ap > 66.4°$. By symmetry, the minimum angle of a **D**-region is $132.8°$. □

Corollary 1 *For any given point $a \in S$, there exist at most two* **D***-regions.*

5.2. Pending Regions and their Properties

We now consider conical sections formed only by pending positions for a given point. By Corollary 1, a point a can have at most two **D**-regions and thus also two disjoint regions of largest possible angle of pending positions. We formalize the two types of equivalence classes introduced earlier: clique-regions and uniform-regions. We then discuss the conditions under which such sets can be combined to form **P**-regions.

5.2.1. The clique-regions

Pick $a, p \in S$ with $p \in C'_a$, so that p must be surrounded by a **D**-region. Let p lie directly below a. Let $Cone(a,\theta_1,\theta_2)$ be a conical section of pending positions, with θ_1 adjacent to the **D**-region surrounding p. Suppose there exists a position θ_p (not necessarily in X_p) at point p such that $C(p,\theta_p,R^*/\rho)$ touches $C(a,\theta_1,R^*/\rho)$. If we enlarge the size of the labeling circles to the optimal value, then $C(p,\theta_p,R^*)$ will intersect potential labeling circles for a placed at positions closer to θ_2; consider the case where it intersects $C(a,\theta_2,R^*)$ itself. Then $C(p,\theta_p,R^*)$ intersects every $C(a,\theta,R^*)$, for $\theta_1 \leq \theta \leq \theta_2$. Clearly, no optimal solution can simultaneously place a labeling circle for point a at position θ and one for point p at position θ_p, since $C(p,\theta_p,R^*)$ and $C(a,\theta,R^*)$ intersect. Therefore, $Cone(a,\theta_1,\theta_2)$ is an equivalence class of positions w.r.t. p, so that, while selecting a position that is feasible w.r.t. p, we can ignore all but one position in $X_a \in Cone(a, \theta_1, \theta_2)$.

In Lemma 3, we derive expressions for the positions θ_1, θ_2 and θ_p Definition 4. The larger the angle of $Cone(a, \theta_1, \theta_2)$, the more positions can be treated as equivalent. Corollary 2 shows that, as $\delta(a, p)$ increases, θ_2 increases monotonically in the first and second quadrants; since this situation corresponds to a clique in a geometric intersection graph, we call $Cone(a, \theta_1, \theta_2)$ a *clique-region.*

Definition 4 *Assume $a, p \in S$. Let $Cone(a, \theta_1, \theta_2)$ denote a conical section such that θ_1 is adjacent to the* **D***-region of a w.r.t. p. We call $Cone(a, \theta_1, \theta_2)$ a* clique-region of a w.r.t. p *whenever there exists a position θ_p such that:*

1. *$C(p, \theta_p, R^*/\rho)$ touches $C(a, \theta_1, R^*/\rho)$;*
2. *$\forall\theta$, $\theta_1 \leq \theta \leq \theta_2$, $C(p, \theta_p, R^*)$ intersects $C(a, \theta, R^*)$.*

A maximal clique-region of a w.r.t. p *is a clique-region of a w.r.t. p that is not properly contained in any clique-region of a w.r.t. p.*

Figure 7 illustrates the basic tenets of the definition. We summarize the above definition as follows. Noting Definition 4, it is clear that a maximal clique-region is adjacent to a **D**-region, so that a point $a \in S$ has at most two maximal clique-regions w.r.t. some given point $p \in C'_a$.

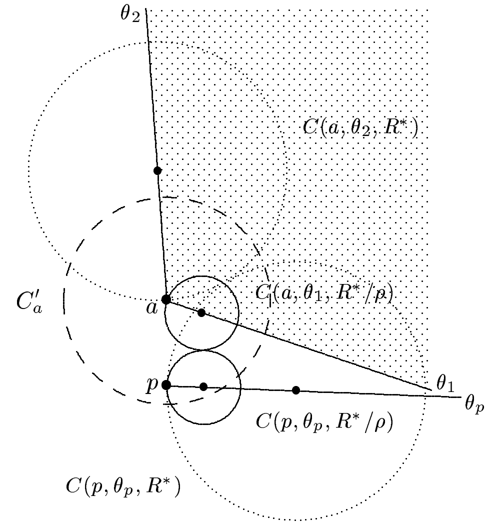


Figure 7: $Cone(a, \theta_1, \theta_2)$ is a maximal clique-region of a w.r.t. p.

Lemma 3 *Pick $a, p \in S$ with $p \in C'_a$ and assume that no other point of S lies within C'_a. Further assume that a and p share the same ordinate. Let $Cone(a, \theta_1, \theta_2)$ and $Cone(a, \theta_3, \theta_4)$ denote two maximal clique-regions of a w.r.t. p. Let θ_p be as in Definition 4 and let a'' and p'' denote the centers of $C(a, \theta_1, R^*)$ and $C(p, \theta_p, R^*)$, respectively. We then have*

1. $\theta_1 = -\arcsin \frac{\delta(a,p)}{2R^*}$

2. *$\theta_2 = \arccos(\frac{\delta(a,p)^2+\delta(a,p'')^2-{R^*}^2}{2\delta(a,p)\delta(a,p'')}) + \arccos(\frac{\delta(a,p'')^2+{R^*}^2-4{R^*}^2}{2\delta(a,p'')R^*}) - \frac{\pi}{2}$*
3. *$\theta_3 = \pi - \theta_2$ and $\theta_4 = \frac{3\pi}{2} - \theta_1$*
4. *$\theta_p = \frac{\pi}{2} - \arccos(\frac{(2\rho-1)\delta(a,p)}{2\sqrt{\rho(\rho-1)\delta(a,p)^2+{R^*}^2}}) - \arccos(\frac{\rho(\rho-1)\delta(a,p)^2-2{R^*}^2}{2R^*\sqrt{\rho(\rho-1)\delta(a,p)^2+{R^*}^2}})$*
5. *θ_2 monotonically increases as $\delta(a,p)$, whenever $\delta(a,p) > 0.4R^*$ increases.*

Figure 8 illustrates the situation; note that two maximal clique-regions may overlap.

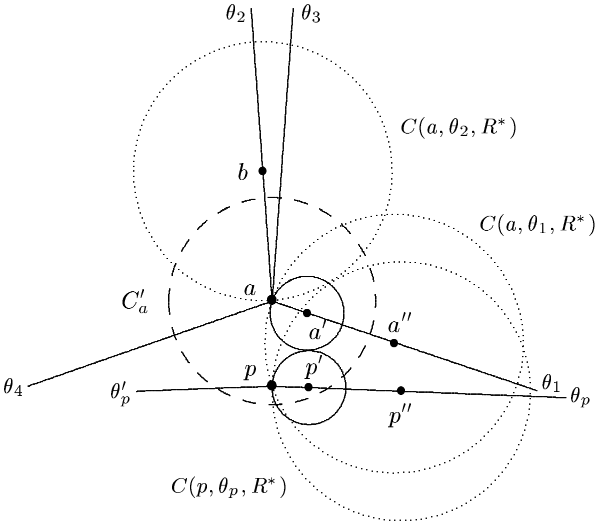


Figure 8: The geometry of Lemma 3. $Cone(a,\theta_1,\theta_2)$ and $Cone(a,\theta_3,\theta_4)$ denote two maximal clique-regions of a w.r.t. p.

Proof. The first relationship follows easily from considering the isosceles triangle $aa''p$; the second from writing $\theta_2 = \angle pap'' + \angle p''ab - \frac{\pi}{2}$; the third from symmetry along the ap axis; and the last from writing $\theta_p = \pi/2 - \angle p'pa' - \angle a'pa$, where a' and p' are the centers of $C(a,\theta_1,R^*/\rho)$ and $C(p,\theta_p,R^*/\rho)$, respectively, and noting that the equality $\delta(p,a')^2 = \left({R^*}^2 + \rho(\rho-1)\delta(a,p)^2\right)/\rho^2$ (derived below).

Consider the triangle paa'. By law of cosines, we have:

$$cos(\angle paa') = \frac{\delta(a,p)^2 + \delta(a,a')^2 - \delta(p,a')^2}{2\delta(a,p)\delta(a,a')} \tag{3}$$

We now determine the value of $\cos(\angle paa')$ in terms of $\delta(a,p)$. Since the points a and p share the same ordinate, the position θ_1 lies in the fourth quadrant. Therefore, we can get that

$$\theta_1 = 270^\circ + \angle paa' \Rightarrow cos(\angle paa') = -\sin\theta_1 \tag{4}$$

Using expression 1 in Lemma 3, we obtain the value of $\sin\theta_1$ as follows:

$$\sin\theta_1 = -\delta(a,p)/2R^* \tag{5}$$

Since the value of aa' corresponds to the radius of smaller circle, we have $\delta(a,a') = R^*/\rho$. Substituting these values in the expression for $\cos(\angle paa')$, we get $\delta(p,a')^2 = \left(R^{*^2} + \rho(\rho-1)\delta(a,p)^2\right)/\rho^2$.

We now prove the monotonicity property. Using the law of cosines in $\Delta pp''a$, we can write

$$\delta(a,p'')^2 = R^{*^2} + \delta(a,p)(\delta(a,p) - 2R^* \cos(\angle app'')) \tag{6}$$

Substituting in the expression for θ_2, we conclude that θ_2 monotonically increases as $\delta(a,p)$ increases (see Figure 9). □

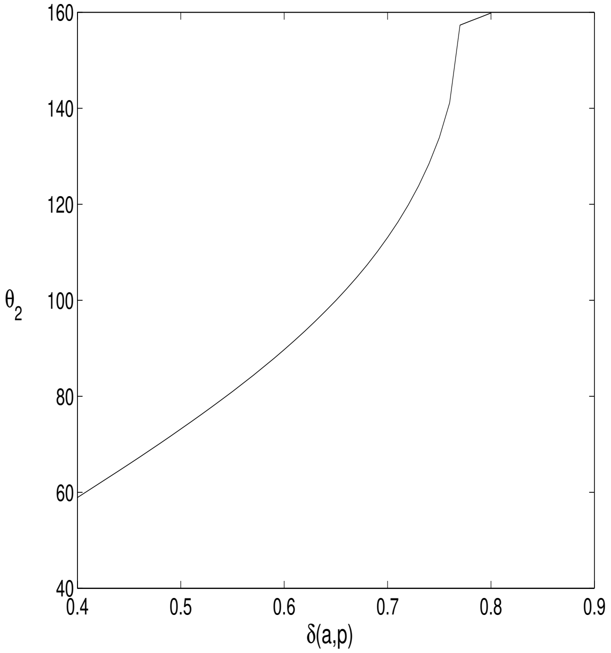


Figure 9: The monotonicity of maximal clique-regions. As $\delta(a,p)$ increases from $0.4R^*$ to $0.8R^*$, the size of the maximal clique-region $Cone(a,\theta_1,\theta_2)$ increases, as illustrated here by observing how θ_2 varies with increasing $\delta(a,p)$.

In Lemma 3, the expressions for positions θ_1 and θ_2 are derived based on the assumption that the points a and p share the same ordinate. Therefore, the position θ_1 lies in the fourth quadrant and the position θ_2 lies either in the first quadrant or the second quadrant of the point a. As $\delta(a,p)$ increases, the corresponding value of θ_1 decreases and the corresponding value of θ_2 increases. Therefore, the value of the clique-region, denoted by $Cone(a,\theta_1,\theta_2)$, increases.

Since the positions θ'_p, θ_3 and θ_4 are mirror images of the positions θ_p, θ_2 and θ_1 respectively, the clique-region of a, denoted by $Cone(a,\theta_3,\theta_4)$ also increases. (Note that the expression for θ'_p is not equal to the expression for θ_p and it needs to be derived based on the positions θ_3 and θ_4 respectively.)

Since θ_1 monotonically decreases (or alternatively θ_4 increases) as $\delta(a,p)$ increases we can show that a maximal clique-region of a w.r.t. p increases as $\delta(a,p)$ increases, which is summarized below.

Corollary 2 *Assume $p \in C'_a - C''_a$ (i.e., $0.8R^* \geq \delta(a,p) \geq 0.4R^*$); then we have: (i) $Angle(\theta_2,\theta_4) < 132.6°$; (ii) $Angle(\theta_1,\theta_3) < 132.6°$; and (iii) $\theta_2 > 58°$.*

The bound of 132.6° is the reason for our specific choice of ρ: our proof of Lemma 8 will need these angles to be no larger than 132.8°, the minimum angle of a **D**-region. We summarize the important features of properties of clique-regions as follows.

1. A clique-region of $a \in S$ is defined w.r.t. some input point $p \in C'_a$ and a ρ-pending position θ_p (not necessarily in X_p) of p. X_p).

2. A maximal clique-region of $a \in S$ is defined w.r.t. an input point $p \in C'_a$ (as opposed to a position).

3. A maximal clique-region of a w.r.t. $p \in C'_a$ is adjacent to the **D**-region that is surrounding p.

4. The input point $a \in S$ has two maximal clique-regions w.r.t. $p \in C'_a$.

5. Suppose $Cone(a,\theta_1,\theta_2)$ denotes the **D**-region surrounding the input point p. Then one maximal clique-region of a w.r.t. p is adjacent to θ_1 and the other to θ_2.

5.2.2. The uniform-regions

Suppose now that there exists $p \in S$ and θ_p such that $C(p,\theta_p,R^*/\rho)$ intersects both $C(a,\theta_1,R^*/\rho)$ and $C(a,\theta_2,R^*/\rho)$. Clearly, $C(p,\theta_p,R^*/\rho)$ uniformly intersects every $C(a,\theta,R^*/\rho)$, for $\theta_1 \leq \theta \leq \theta_2$. We call the region $Cone(a,\theta_1,\theta_2)$ a *uniform-region.* Clearly, no optimal solution can simultaneously place a label at θ_2 for the point a at θ_p for the point p. However, unlike a clique-region, where every position can be treated as a feasible position, in this case, only the position θ_2 can be considered feasible. This is due to the fact that an optimal solution may place a label at a position in X_a which is $\theta_p - \varepsilon$ and place a label at position θ_2.

In Lemma 4, we derive expressions for the positions θ_1 and θ_2. As a corollary of Lemma 4, we show that θ_2 decreases, in the first quadrant, as $\delta(a,p)$ increases. This is a crucial observation as, for any $\delta(a,p) \geq 0.4R^*$, it allows us to treat a uniform-region of a w.r.t. p as a clique-region of a w.r.t. p. In Corollary 3, we determine the minimum value of θ_2 for any $\delta(a,p) > 0.4R^*$.

Definition 5 *$Cone(a,\theta_1,\theta_2)$ is a (ρ-)*uniform-region *of a w.r.t. p and pending position θ_p whenever $Cone(a,\theta_1,\theta_2)$ is a region of only pending positions and $C(p,\theta_p,R^*/\rho)$ intersects $C(a,\theta,R^*/\rho)$, for every $\theta \in Cone(a,\theta_1,\theta_2)$. A* maximal uniform-region *of a w.r.t. p and θ_p is a uniform-region of a of largest angle w.r.t. p and θ_p. A* maximal uniform-region *of a w.r.t. p is a maximal uniform-region of a w.r.t. p and θ'_p, where θ'_p is the largest angle preserving $a \notin C(p,\theta'_p,R^*)$.*[e]

[e] Here we assume that θ'_p to be on the right of the line joining a and p.

Figure 10 illustrates the second part of the definition; note that uniform-regions, like clique-regions, are contiguous regions of pending positions, so that, even though $Cone(a, \theta_0, \theta_1)$ meets the intersection requirements, it is not a uniform region, since all of $Cone(a, \theta_0, \theta_1)$ falls within a dead region.

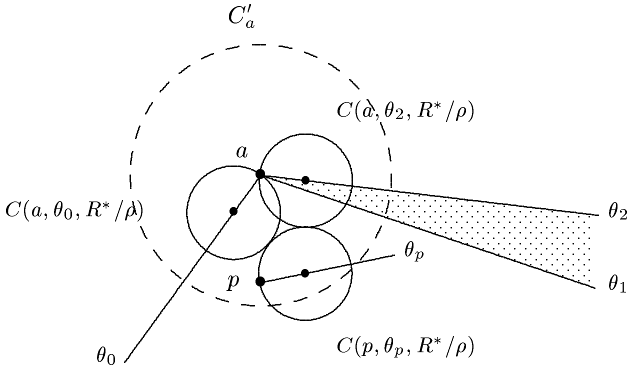

C'_a
$C(a, \theta_2, R^*/\rho)$
a
$C(a, \theta_0, R^*/\rho)$
θ_2
θ_p
p
θ_1
θ_0
$C(p, \theta_p, R^*/\rho)$

Figure 10: $Cone(a, \theta_1, \theta_2)$ is a maximal uniform-region of a w.r.t. p and θ_p.

At this stage, it is important to make a distinction between a maximal uniform-region w.r.t. both point p and position θ_p and a maximal uniform-region w.r.t. just a point p. The defintion of a maximal uniform-region w.r.t. p and θ_p is specific to a particular position—the position θ_p in this case. Suppose $Cone(a, \theta_1, \theta_2)$ denotes a maximal uniform-region w.r.t. p and θ_p. In this case, although every region $Cone(a, \theta_1, \theta'_2) \subset Cone(a, \theta_1, \theta_2)$ is a uniform-region w.r.t. p and θ_p, it cannot be called maximal w.r.t. p and θ_p, since it is properly contained in $Cone(a, \theta_1, \theta_2)$. However, it is possible that $Cone(a, \theta_1, \theta'_2)$ may be a maximal uniform-region w.r.t. p and position θ'_p, where $\theta'_p \neq \theta_p$.

Consider a maximal uniform-region w.r.t. just a point p. It is associated with all label positions of p, meaning it is not properly contained in any uniform-region w.r.t. p and θ_p, for any θ_p. Therefore, $Cone(a, \theta_1, \theta_2)$ may not be a maximal uniform-region w.r.t. p, since it may be properly contained in some other maximal uniform-region of a w.r.t. p.

Since a maximal uniform-region of a w.r.t. p cannot be a subset of any uniform-region of a w.r.t. p, it must be adjacent to a **D**-region. In Figure 10, the position θ_1 delimits both a **D**-region of a w.r.t. p and a maximal uniform-region of a w.r.t. p and θ_p. Thus we already know one of the angles from Lemma 3. The other angle is also easy to compute: denote by p' the center of $C(p, \theta_p, R^*/\rho)$ and by a' the center of $C(a, \theta_2, R^*/\rho)$ and write $\theta_2 = \angle pap' + \angle p'aa' - \frac{\pi}{2}$. Maximizing the angle θ_p gives a situation similar to that of Lemma 3 and allows us to write $\delta(a, p')^2 = \left(R^{*^2} + \rho(\rho-1)\delta(a, p)^2\right)/\rho^2$. The following lemma can be proved similarly as Lemma 3. For the monotonicity of θ_2, see Figure 11.

Lemma 4 *Let $Cone(a, \theta_1, \theta_2)$ denote a maximal uniform-region w.r.t. p and let θ_1 be adjacent to the* **D***-region surrounding p. We have*

1. $\theta_1 = -\arcsin \frac{\delta(a,p)}{2R^*}$

2. $\theta_2 = \arccos(\frac{(2\rho-1)\delta(a,p)}{2\sqrt{\rho(\rho-1)\delta(a,p)^2+{R^*}^2}}) + \arccos(\frac{\rho(\rho-1)\delta(a,p)^2-2{R^*}^2}{2R^*\sqrt{\rho(\rho-1)\delta(a,p)^2+{R^*}^2}}) - \frac{\pi}{2}$

3. θ_2 *monotonically decreases as* $\delta(a,p)$ *increases.*

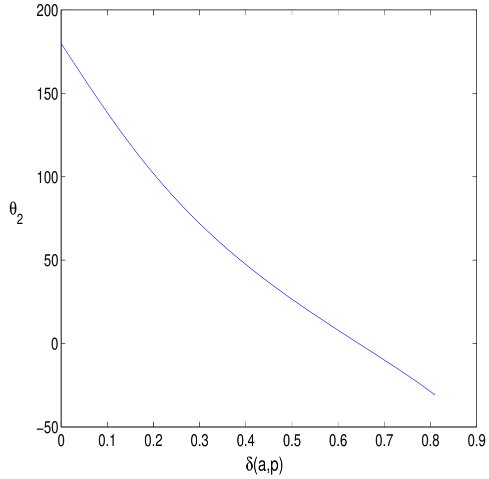


Figure 11: Monotonicity of maximal uniform-regions. In contrast to maximal clique-regions, the size of maximal uniform-regions as denoted by the conical region $Cone(a, \theta_1, \theta_2)$ decreases as $\delta(a, p)$ increases.

Corollary 3 *Let* $Cone(a, \theta_1, \theta_2)$ *be a maximal uniform-region w.r.t.* p*, with* $p \in (C'_a - C''_a)$*; then we have* $\theta_2 < 48°$*.*

We summarize the important properties of uniform-regions as follows.

1. A uniform-region of $a \in S$ is defined w.r.t. some input point $p \in C'_a$ and its ρ-pending position say θ_p (not necessarily in X_p).

2. A maximal uniform-region of $a \in S$ is defined w.r.t. an input point $p \in C'_a$ (as opposed to a position).

3. A maximal uniform-region of a w.r.t. $p \in C'_a$ is adjacent to the **D**-region surrounding p.

4. The input point $a \in S$ has two maximal uniform-regions w.r.t. $p \in C'_a$.

5. Suppose $Cone(a, \theta_1, \theta_2)$ denotes the **D**-region surrounding the input point p. Then one maximal uniform-region of a w.r.t. p is adjacent to θ_1 and the other uniform-region of a w.r.t. p is adjacent to θ_2.

5.3. **P***-regions and their Properties*

Let D be a **D**-region of a with limiting angle θ_1 and let $Cone(a, \theta_1, \theta_{21})$ denote a maximal clique-region w.r.t. p and $Cone(a, \theta_1, \theta_{22})$ denote a maximal uniform-region w.r.t. p—in both conical sections, θ_1 is adjacent to D. Pick $p \in C'_a - C''_a$; by

Corollaries 2 and 3, we have $\theta_{21} > \theta_{22}$, so that $Cone(a, \theta_1, \theta_{22})$ is also a *clique-region* w.r.t. p.

Lemma 5 *Pick $a, p \in S$ with $0.4R^* \leq \delta(a,p) \leq 0.8R^*$; then a maximal uniform-region of a w.r.t. is also a clique-region of a w.r.t. p.*

Suppose p is the only point in C'_a, i.e., we have $N(C'_a) = 1$. Let $Cone(a, \theta_4, \theta_1)$ denote the **D**-region of a surrounding p. Then a has two maximal uniform-regions, where one is adjacent to θ_1 and the other to θ_4. Let $Cone(a, \theta_1, \theta_2)$ and $Cone(a, \theta_3, \theta_4)$ denote these two sets. By definition, at least one pending position from X_a must be present in these two sets, as otherwise R^* cannot be optimal.[f] Then, clearly, we can select (one or) two positions, one in $Cone(a, \theta_1, \theta_2) \cap X_a$ and the other one in $(Cone(a, \theta_1, \theta_2) - Cone(a, \theta_3, \theta_4)) \cap X_a$, and ignore the remaining positions, so that any feasible solution guarantees to place a label of size R^*/ρ for a at one of these two positions. Now suppose that we have $N(C'_a) > 1$. Then the selection of the two positions must be such that they form a pair of feasible positions w.r.t. the other input points in C'_a. However, this situation can get complicated due to the fact that, for each input point say $p \in C'_a$, we have two maximal uniform-regions and two maximal clique-regions of a w.r.t. p, resulting in a large number of maximal uniform-regions and maximal clique-regions for a. We now present a number of results that allow us to select a pair of feasible positions w.r.t. all of the input points in C'_a.

We first recall that maximal clique-regions and maximal uniform-regions are adjacent to a **D**-region. Therefore, maximal uniform-regions and maximal clique-regions of a w.r.t. the points that lie in the same **D**-region must share a boundary position adjacent to it. We use this idea in Lemma 5 to show that, for $\delta(a,p) > 0.4R^*$, a maximal uniform-region is also a maximal clique-region. Lemma 5 is useful only for $N(C'_a) = 1$. We need a similar result for $N(C'_a) > 1$. Lemma 6 generalizes Lemma 5 when the number of **D**-regions at a is one, $N(C'_a)$ is larger than 1, and all the points lie outside C''_a but inside C'_a. That is, Lemma 6 shows that all maximal uniform-regions of a are also clique-regions of a w.r.t. the points in C'_a. Therefore, in this case, we can select a pair of positions that are feasible w.r.t. to all of the input points in C'_a. In order to deal with situations where C''_a also has points and a has more than one **D**-region, we use the concept of a **P**-region, formed by the union of all maximal clique-regions and maximal uniform-regions that share the same boundary position. Our aim is to select at most one position in each **P**-region. Since each **P**-region is adjacent to a **D**-region, a can have at most four **P**-regions. In Lemma 8, we show that a has at most two **P**-regions when we have $N(C'_a) > N(C''_a)$, a crucial result for our selection algorithm.

We now allow more than one point in $(C'_a - C''_a)$. Let $p \in S$ and $q \in S$ be located within $(C'_a - C''_a)$ and within D, a **D**-region of a. (The three points a, p, and q of S are distinct.) Let $Cone(a, \theta_1, \theta_2)$ denote the conical section of minimum angle surrounding the maximal uniform-regions of a w.r.t. p and q. (Assume that position θ_1 is adjacent to D.)

[f] It is possible that at most one of the sets $Cone(a, \theta_1, \theta_2) \cap X_a$ and $Cone(a, \theta_3, \theta_4) \cap X_a$ can be empty.

Lemma 6 *Let p, q and $Cone(a, \theta_1, \theta_2)$ be defined as above. Suppose the minimum angle of each of the maximal uniform-regions of a w.r.t. p and q is greater than zero. Then $Cone(a, \theta_1, \theta_2)$ is a clique-region w.r.t. both p and q.*

Proof. We assume $\delta(a, q) \geq \delta(a, p)$. Let $Cone(a, \theta_1, \theta_2')$ and $Cone(a, \theta_1, \theta_2'')$ be the maximal uniform-regions of a w.r.t. p and q respectively—by assumption, we have $Angle(\theta_2', \theta_1) > 0$ and $Angle(\theta_2'', \theta_1) > 0$.

Let θ_p' be a pending position of p such that $C(p, \theta_p', R^*/\rho)$ almost intersects $C(a, \theta_2, R^*/\rho)$, i.e., θ_p' is ε away from being a dead position. Let θ_p'' be a pending position of p of least absolute angle such that $C(p, \theta_p'', R^*/\rho)$ intersects $C(a, \theta_1, R^*/\rho)$, i.e., $Cone(p, \theta_p'', \theta_p')$ is a maximal uniform-region of p w.r.t. a. Let θ_q be a pending position at q such that $C(q, \theta_q, R^*/\rho)$ intersects $C(a, \theta_1, R^*/\rho)$—in order for our assumption, i.e., $Angle(\theta_2'', \theta_1) > 0$, to hold, such a θ_q must exist.

We claim that q cannot lie inside $C(p, \theta_p', R^*)$ or outside $C(p, \theta_p'', R^*/\rho)$. Suppose that q lies inside $C(p, \theta_p', R^*)$. It is easily verified that every $\theta_p \in Cone(a, \theta_p'', \theta_p')$ becomes a dead position, implying that $Cone(a, \theta_1, \theta_2')$ is not a maximal uniform set w.r.t. p, a contradiction. Suppose now that q lies outside $C(p, \theta_p'', R^*/\rho)$. Then $C(q, \theta_q, R^*/\rho)$ cannot intersect $C(a, \theta_1, R^*/\rho)$, implying that $Cone(a, \theta_1, \theta_2'')$ is not a maximal uniform-region w.r.t. q, another contradiction. Thus we can establish $\theta_2' > \theta_2''$. By Corollaries 2 and 3, we can further verify that $Cone(a, \theta_1, \theta_2)$ is a clique-region of a w.r.t. both p and q. □

Let D denote a given **D**-region. We know that each boundary position of D is adjacent to a maximal clique-region and to a maximal uniform-region. Given a maximal clique-region w.r.t. p and a maximal uniform-region w.r.t. q, both adjacent to the same boundary position of D, one must contain the other, which allows us to combine them.

Definition 6 *$Cone(a, \theta_1, \theta_2)$ is a* **P**-region *if it is not contained in any maximal clique-region or maximal uniform-region of a w.r.t. p, for any point $p \in S$ within C_a'; if this region is a clique-region or uniform-region w.r.t. p, then we call p the* reference point *of the* **P**-region*.*

A **P**-region can be a maximal uniform-region, a maximal clique-region, or both (with more than one reference point). The maximality condition of a clique-region or uniform-region is preserved in the definition of a **P**-region: neither $Cone(a, \theta_1 - \varepsilon, \theta_2)$ nor $Cone(a, \theta_1, \theta_2 + \varepsilon)$ is a **P**-region. The following lemma is easily proved.

Lemma 7 *Let P be a* **P***-region for a with reference point p. If p belongs to $C_a' - C_a''$, then P is a clique-region.*

We now study several useful properties of **P**-regions. We first note that the region, excluding any **D**-regions, surrounding a given point $a \in S$ can be partitioned into **P**-regions. Our aim is to select one allowable position from each **P**-region and eliminate all others. Assuming that we can select an allowable position from each **P**-region, then we need to find an upper bound on the number of **P**-regions that can exist for any point. A simple upper bound is 4, since each **P**-region is adjacent to a **D**-region. However, in order to construct a *2SAT* instance, we need to select at most 2 positions for each point.

Lemma 8 *Let $a \in S$ denote a point with $N(C_a') > N(C_a'')$. Then the number of*

P-*regions at a is at most* 2.

Proof. If the number of **D**-regions at a is one, then the number of **P**-regions is at most two. Consider then the case where there are two **D**-regions, D_1 and D_2; note that they must be non-intersecting. Since each D_i is determined by a different point of S within C'_a, we must have $N(C'_a) \geq 2$. Pick $p, q \in S$ such that p lies inside D_1 and q lies inside D_2. Since $N(C'_a)$ is larger than $N(C''_a)$, assume without loss of generality that p lies outside C''_a.

Consider adding points p and q in that order to the neighborhood of a. After adding p, we have two **P**-regions, each adjacent to one border position of D_1; call them $Cone(a, \theta_1, \theta_2)$ and $Cone(a, \theta_3, \theta_4)$—assume that θ_1 and θ_4 are adjacent to D_1. (Note that these two **P**-regions may intersect.) By Lemma 7, these two **P**-regions are maximal clique-regions; furthermore, by Corollary 2, we have $Angle(\theta_1, \theta_3) < 132.6°$ and $Angle(\theta_2, \theta_4) < 132.6°$. Adding q creates the **D**-region D_2, which has an angle of at least $132.8°$. This implies that D_2 must include at least one of the following three pairs of positions: (i) (θ_1, θ_3), (ii) (θ_2, θ_4), or (iii) (θ_2, θ_3). In the first two cases, at least one of the two existing **P**-regions vanishes, thus preserving our conclusion. When D_2 intersects both θ_2 and θ_3, the **P**-regions w.r.t. p simply shrink and thus remain maximal clique-regions w.r.t. p. Any **P**-region caused directly by the addition of q is a subset of either $Cone(a, \theta_1, \theta_2)$ or $Cone(a, \theta_3, \theta_4)$—so that no new **P**-region gets created. By the same reasoning, adding a third or even a fourth point of S within C'_a simply causes further shrinking of the **P**-regions without increasing their number. □

Corollary 4 *If the number of* **D**-*regions is* 2 *and we have* $N(C'_a) > N(C''_a)$, *then each* **P**-*region is a maximal clique-region w.r.t.* $p \in (C'_a - C''_a)$.

Corollary 4 implies that we can select a pair of positions, one from each **P**-regions, so that they are feasible w.r.t. the points in $C'_a - C''_a$. We summarize the important properties of **P**-regions as follows.

1. A **P**-region of $a \in S$ is a union of all the maximal clique-regions and maximal uniform-regions that share the same boundary adjacent to **D**-region.

2. A **P**-region of $a \in S$ has a reference point say $p \in C'_a$.

3. A **P**-region of $a \in S$ can be a maximal clique-region, a maximal uniform-region or both.

4. A **P**-regions of a is adjacent to a **D**-region.

6. Structural Packing Properties

In this section we provide geometric lemmata that capture the structural properties of the KPML problem and relate them to the conical regions described in the previous section. In particular, we determine upper bounds on the number of input points that lie inside C'_a and C''_a.

We begin by bounding the number of points that can appear within various radii of a given point. We use the well-known packing result given below.

Proposition 1 *Let C be a circle of radius r and let S be a set of circles of radius r such that every circle in S intersects C and no two circles in S intersect each other. Then we have $|S| \leq 5$.*

Proposition 1 implies that C'_a can have at most five input points, including a, since each of these points needs to be labeled with a label of size R^*. Thus a simple upper bound on $N(C'_a)$ is four. Let B'_a, B''_a, and B'''_a denote the circles centered at $a \in S$ (not labels) of radius $0.8R$, $0.4R$, and $0.1R$ respectively. Furthermore, let $N(B'_a)$, $N(B''_a)$, and $N(B'''_a)$ denote the numbers of points of S (other than a) that lie inside B'_a, B''_a, and B'''_a, respectively, with the condition that each such point can be labeled using a circular label of size R. Our bounds can be summarized as follows.

Lemma 9 *Let $a \in S$ and let C'_a, C''_a, C'''_a, $N(C'_a)$, $N(C''_a)$, and $N(C'''_a)$ be as defined in Section 4.*

1. $N(C'_a) \leq 4$
2. $N(C''_a) \leq 2$
3. $N(C''_a) > 0 \Longrightarrow N(C'_a) \leq 3$.
4. $N(C''_a) = 2 \Longrightarrow N(C'_a) = N(C''_a) = 2$
5. $N(C'''_a) = 1 \Longrightarrow N(C''_a) = N(C'''_a) = 1$.
6. *Let B'_a, B''_a, B'''_a, $N(B'_a)$, $N(B''_a)$ and $N(B'''_a)$ be as defined above. Then we have (i) $N(B'_a) \leq 4$, (ii) $N(B''_a) \leq 2$, (iii) $N(B''_a) > 0 \Longrightarrow N(B'_a) \leq 3$, (iv) $N(B''_a) = 2 \Longrightarrow N(B'_a) = N(B''_a)$, and (v) $N(B'''_a) = 1 \Longrightarrow N(B''_a) = N(B'''_a)$.*

Figure 12 informally shows why a labeling circle associated with a third point q cannot be forced within C''_a or even within C'_a if two other points (p and r) lie within C''_a—these are the second and fourth assertions of the lemma.

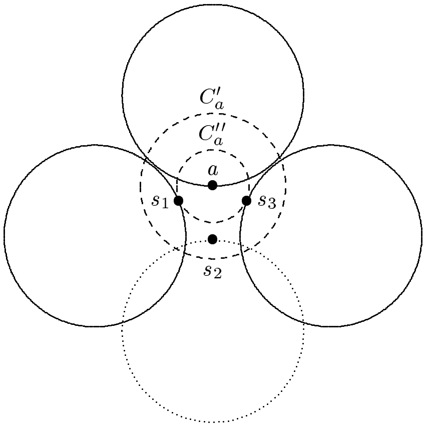


Figure 12: Illustration of parts 2 and 4 of Lemma 9.

Proof.

1. If we had $N(C'_a) \geq 5$, C'_a would contain at least 6 points, contradicting Proposition 1.

2. Assume $N(C''_a) \geq 3$, with three input points denoted by s_1, s_2, and s_3, and set $s_0 = a$, $s_1 = p$, $s_2 = q$ and $s_3 = r$. There must be a placement of non-intersecting circles, of radii R^*, at these four points. With out loss of generality, let us assume that a circle of radius R^* is placed at the point s_0, as shown Figure 12. Thus we have three points s_1, s_2 and s_3 which must lie outside the circle placed at s_0. With out loss of generality, let us further assume that these three points, s_1, s_2 and s_3, lie in the order of anticlockwise direction starting from s_1, as shown in Figure 12.

   For each i, $0 \leq i \leq 3$, let s'_i denote the center of circle labeling the point s_i and let ϕ_i denote the angle between the rays $s_0 s'_i$ and $s_0 s'_{i+1}$ (using the modulo 4). Thus these angles ϕ_i discretize the total angle around the point a, and hence we have $\sum_{i=0}^{3} \phi_i = 360°$. Since the radius of optimal label is R^* we have $\delta(s_i, s'_i) = R^*$ and $\delta(s_0, s'_i) < 1.4R^*$. Since no two optimal labels intersect, we have $\delta(s'_i, s'_{i+1}) \geq 2R^*$.

   By the law of cosines, we have

$$\cos(\phi_0) = \frac{\delta(a,a')^2 + \delta(a,s'_1)^2 - \delta(a',s'_1)^2}{2\delta(a,a')\delta(a,s'_1)} \Longrightarrow \phi_0 > 111° \tag{7}$$

$$\cos(\phi_1) = \frac{\delta(a,s'_1)^2 + \delta(a,s'_2)^2 - \delta(s'_1,s'_2)^2}{2\delta(a,s'_1)\delta(a,s'_2)} \Longrightarrow \phi_1 > 91° \tag{8}$$

$$\cos(\phi_2) = \frac{\delta(a,s'_2)^2 + \delta(a,s'_3)^2 - \delta(s'_2,s'_3)^2}{2\delta(a,s'_2)\delta(a,s'_3)} \Longrightarrow \phi_2 > 91° \tag{9}$$

   and

$$\cos(\phi_3) = \frac{\delta(a,s'_3)^2, \delta(a,a')^2 - \delta(a',s'_3)^2}{2\delta(a,s'_3)\delta(a,a')} \Longrightarrow \phi_3 > 111° \tag{10}$$

   The total angle is $\sum_{i=0}^{3} \phi_i = 111°+91°+91°+111° = 404°$. This implies that at least one of the two assumptions, i.e., $\delta(s'_i, s'_{i+1}) \geq 2R^*$ and $\delta(a, s'_i) \leq 1.4R^*$ must be false. Suppose $\delta(s'_i, s'_{i+1}) \geq 2R^*$ is false. Then the labels intersect implying that radius of optimal label must be strictly less than R^*. Suppose $\delta(a, s'_i) > 1.4R^*$. Then $N(C''_a) < 3$, implying that the assumption $N(C''_a) \geq 3$ is false. In each case, we have the desired contradiction.

3. Similar to part 4.

4. For the sake of contradiction, let us assume $N(C'_a) = 3$. As shown in Figure 12 and Figure 13, we have two possible cases. We consider the second case. Let p and q denote the input points inside C''_a and a', p', and q' denote the centers of the circles each of size R^* placed at the points a, p, and q respectively. Pick $r \in (C'_a - C''_a)$ and let r' denote the center of the circle of size R^* placed at r. We now determine the minimum value of $\delta(a, r')$—if it is less than $1.8R^*$, then we have $N(C'_a) > N(C''_a)$. By the law of cosines we have:

$$\delta(a,r')^2 = \delta(a,q')^2 + \delta(q',r')^2 - 2\cos(\angle aq'r')\delta(a,q')\delta(q',r') \tag{11}$$

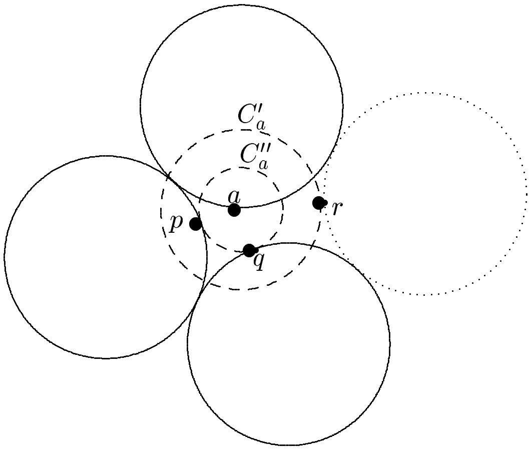


Figure 13: Illustration of part 4 in Lemma 9.

We have $\delta(a, q') \leq 1.4R^*$ and $\delta(q', r') = 2R^*$; so we need to compute $\cos(\angle aq'r')$. We can easily verify $\angle a'aq' < 158°$. From the known value of $\angle a'aq'$, we conclude $\delta(a', r') < 2.36R^*$. By applying the law of cosines we next obtain $\angle aq'r' > 63°$. By substituting these values in the above expression, we finally get $\delta(a, r') > 1.84R^*$, so that r cannot lie inside C'_a. The analysis of the other case is similar. In both cases, we conclude $\delta(a, r') > 1.8R^*$, the desired contradiction.

5. Similar to part 4 (see Figure 14).
6. Similar to parts 1 through 5.

□

Corollary 5 *Let $a, p, q \in S$ be three points with $p, q \in C''_a$ obeying $\delta(p, q) < 0.4R^*$. Then, for any point $r \in S$ distinct from a, p, and q, we have $r \notin C'_a$, $r \notin C'_p$ and $r \notin C'_q$.*

Proof. By Lemma 9, we have $r \notin C'_a$. Since we have $\delta(p, a) < 0.4R^*$ and $\delta(p, q) < 0.4R^*$, we also have $N(C''_p) > 1$. Therefore, by Part 3 of Lemma 9, we have $N(C'_p) = N(C''_p)$, implying $r \notin C'_p$. *Mutatis mutandis*, the same reasoning proves the result for q as well. □

Corollary 5 indicates that the points a, p, and q can be labeled separately from the rest of the points in S.

7. Selecting Feasible Positions

We combine the results in the preceding sections to design an efficient procedure to select two positions per point that guarantees a feasible solution if one exists; we call such positions feasible. By Lemma 1, only the points inside C'_a need to be considered to establish the feasibility of positions at a.

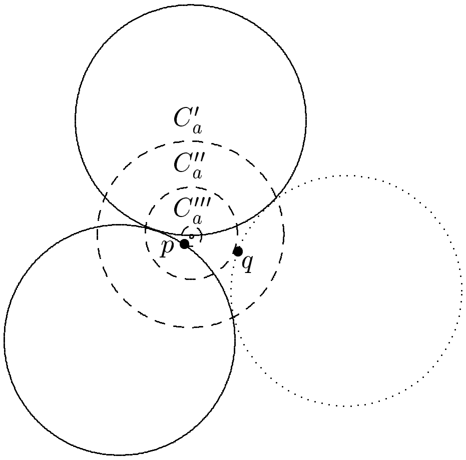


Figure 14: Illustration of part 5 in Lemma 9.

7.1. Overall Idea

Consider a point $a \in S$ and let $Cone(a, \theta_1, \theta_2)$ be its **P**-region with a reference point $p \in S$. (Assume that p lies in the **D**-region adjacent to θ_1.) Let α_p be a pending position at p with least absolute angle such that $C(p, \alpha_p, R^*/\rho)$ intersects $C(a, \theta_1, R^*/\rho)$. If we have $p \in (C'_a - C''_a)$, then, by Lemma 7, $Cone(a, \theta_1, \theta_2)$ is a clique-region w.r.t. p. Observe that no optimal solution can simultaneously contain labels $C(p, \alpha_p, R^*)$ and $C(a, \theta, R^*)$, for any $\theta \in Cone(a, \theta_1, \theta_2)$, as they intersect each other. Now suppose we have $p \in C''_a$; then by Lemma 7, $Cone(a, \theta_1, \theta_2)$ is a uniform-region w.r.t. p. Let β_p be a position of largest absolute angle at p such that $C(p, \beta_p, R^*/\rho)$ intersects $C(a, \theta_2, R^*/\rho)$. Clearly, $C(p, \beta_p, R^*/\rho)$ also intersects $C(a, \theta, R^*/\rho)$, for every $\theta \in Cone(a, \theta_1, \theta_2)$. Thus it is sufficient to consider θ_2 and ignore the remaining positions inside $Cone(a, \theta_1, \theta_2)$. In both cases, position θ_2 is feasible w.r.t. p; however, it may be possible that θ_2 is infeasible w.r.t. some other point, say $q \in C'_a$. This situation may arise when a has more than two **P**-regions and q lies in a **D**-region that is different from the **D**-region associated with p. We show that, regardless of the positions of points p and q in C'_a, we can always find two feasible positions for a.

7.2. Selecting Positions for One Point

By Lemma 9, we know that $N(C''_a) \leq 2$. Thus we need to consider three distinct cases: (i) $N(C''_a) = 0$, (ii) $N(C''_a) = 1$ and (iii) $N(C''_a) = 2$. Lemma 10 handles the first two cases while Lemma 11 handles the third case.

Lemma 10 *Let P_1 and P_2 denote the* **P***-regions of point $a \in S$. Assume $N(C''_a) = 0$ or $N(C''_a) = 1$ and let p be defined as above. Then, given the sets $X_a \cap (P_1 - P_2)$ and $X_a \cap (P_2 - P_1)$, we can select two feasible positions $\theta'_a, \theta''_a \in\in X_a$ $O(1)$ time.*

Proof. By Lemma 9, we have $N(C'_a) > N(C''_a)$.[g] Thus, by Lemma 8, the number of **P**-regions at a is at most 2. Let $P_1 = Cone(a, \theta_1, \theta_2)$ and $P_2 = Cone(a, \theta_3, \theta_4)$ denote the two **P**-regions at a, and assume that θ_1 and θ_4 are adjacent to a **D**-region of a. (If the number of **D**-regions at a is 2, then all four positions θ_i are adjacent to a D-region of a.) Without loss of generality, assume $\theta_i \in X_a$, for $1 \leq i \leq 4$. Finally, let $U_i \subseteq P_i$ be the uniform-region with maximum angle, i.e., among all maximal uniform-regions that lie inside of P_i, U_i has the largest angle. We now consider two cases.

Case 1: $N(C''_a) = 0$.

By Lemma 8, each P_i must be a clique-region w.r.t. each point in C'_a. (This is also true when a has only one **D**-region, since, by assumption, there are no safe positions.) Thus we can select $\theta'_a \in X_a \cap (P_1 - P_2)$ and $\theta''_a \in X_a \cap (P_2 - P_1)$. (If either $X_a \cap (P_1 - P_2)$ or $X_a \cap (P_2 - P_1)$ is empty, we select just one pending position θ'_a from a nonempty set $X_a \cap P_i$.) It is easily verified that the positions θ'_a and θ''_a are feasible. In this case, we can select *any* position $\theta'_a \in X_a \cap (P_1 - P_2)$ and *any* position $\theta''_a \in X_a \cap (P_2 - P_1)$.

Case 2: $N(C''_a) = 1$.

Pick $p \in C''_a$ and denote its associated **D**-region by $Cone(a, \theta_4, \theta_1)$. Assume that p is vertically below a, so that P_1 and P_2 lie on the right and left of $\overline{ap}$, respectively. Let q and r be the points of S that lie inside $(C'_a - C''_a)$. If a has two **D**-regions, we set $\theta'_a = \theta_2$ and $\theta''_a = \theta_3$, positions that are easily verified to be feasible. If a has a single **D**-region, call it $Cone(a, \theta_4, \theta_1)$, the points p, q and r must all lie inside that region. We then have two sub-cases to consider.

Case 2(a): p is reference point of at most one P_i.

Suppose p is a reference point of at most one P_i; let it be P_2. Thus P_2 is a maximal uniform-region w.r.t. p and we have $U_2 = P_2$. Let the reference point of P_1 be q. By Lemma 6, U_1 must be a clique-region w.r.t. both q and r. (By Lemma 6, U_1 is a clique-region w.r.t. r if and only if r has a pending position θ_r such that $C(r, \theta_r, R^*/\rho)$ intersects $C(a, \theta_1, R^*/\rho)$. If no such position θ_r exists, then every position inside P_1 is feasible w.r.t. r; thus we can still treat U_1 as a clique-region w.r.t. r.) We can ignore r, as positions feasible w.r.t. p and q must also be feasible w.r.t. r. Choosing $\theta'_a \in X_a \cap (U_1 - U_2)$ and $\theta''_a \in X_a \cap (U_2 - U_1)$ allows us to obtain the required feasible positions (see Figure 15).

Case 2(b): p is reference point of both P_1 and P_2.

Suppose p is a reference point of both P_1 and P_2, i.e., we have $P_i = U_i$. This can happen only when a has one **D**-region. (This is due to the fact that q or r cannot lie in a **D**-region different from the **D**-region containing p as, by Lemma 8, it would also imply that both P_1 and P_2 as clique-regions w.r.t. q and r.) Furthermore, this can happen when p has two **D**-regions and a lies in a **D**-region different from the **D**-region containing q and r. (As shown in Figure 16, in order for p to be a reference point of both P_1 and P_2, there must exist two pending positions θ'_p and

[g] Although it is possible that we have $N(C'_a) = N(C''_a) = 0$ and $N(C'_a) = N(C''_a) = 1$, we can treat these as special cases of Lemma 11.

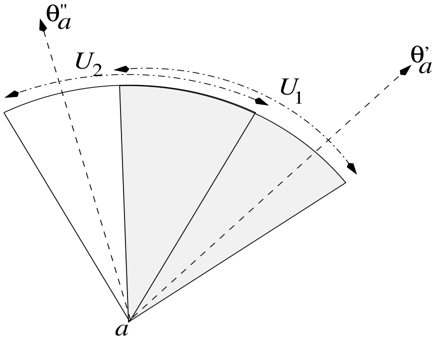


Figure 15: Illustration of the selection of positions in Case 2a

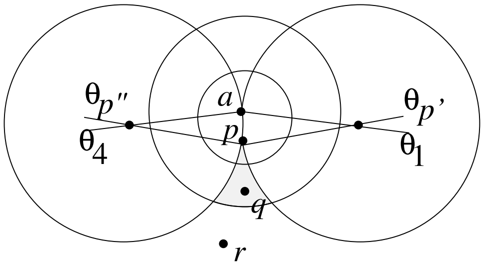


Figure 16: Possible positions of q and r

θ''_p. The points q and r cannot lie in $C(p, \theta'_p, R^*)$ and $C(p, \theta''_p, R^*)$, as the positions θ'_p and θ''_p will become dead. Therefore, the points must lie in the shaded region. Using a packing argument, we can further verify that both q and r cannot lie inside the shaded region.) Furthermore, using a simple packing argument, we can verify that q or r must lie outside C'_a. Say r lies outside C'_a; we have two possibilities: (i) $\delta(p,q) \leq 0.4R^*$ and (ii) $\delta(p,q) > 0.4R^*$. In the first case, by Lemma 11, the points a, p, and q can be labeled separately. In the second case, p has two **D**-regions. By Lemma 8, **P**-regions at p are clique-regions w.r.t. q. Therefore, every pending position of p is a feasible position w.r.t. q. Thus we select two pending positions, each from a different **P**-region of p, that are farthest away from a. For each of the two feasible positions selected for the point p, we can select a feasible position for a. Thus we have two feasible positions for a. At least one such feasible position for a must exist—otherwise, R^* would not be optimal.

Important Remark: In all the above cases selection of positions is limited to conical sections $P_1 - P_2$ and $P_2 - P_1$. We have two cases to consider: (i) $k = O(|S|)$[h] and (ii) $k = \infty$.

$k = O(|S|)$: In this case, the selection of positions θ'_a and θ''_a must be restricted to the sets $X_a \cap (P_1 - P_2)$ and $X_a \cap (P_2 - P_1)$. As discussed above, any position from $X_a \cap (P_1 - P_2)$ and any position from $X_a \cap (P_2 - P_1)$ form a pair of feasible

[h] Any polynomial function of $|S|$ would suffice for the argument to hold. On the other hand if k is fixed (independent of input size) then the selection can be seen to take $O(1)$ time.

positions. Thus, given the sets $X_a \cap (P_1 - P_2)$ and $X_a \cap (P_2 - P_1)$ a pair of positions can be selected in $O(1)$ time. (Note, however, that determining the sets $X_a \cap (P_1 - P_2)$ and $X_a \cap (P_2 - P_1)$ involves identifying and eliminating the positions in X_a that do not belong to the sets $X_a \cap (P_1 - P_2)$ and $X_a \cap (P_2 - P_1)$. In Theorem 2, we show that it takes $O(\log k)$ time determine the sets $X_a \cap (P_1 - P_2)$ and $X_a \cap (P_2 - P_1)$.)

$k = \infty$: In this case, any position from $P_1 - P_2$ and any position from $P_2 - P_1$ can be selected. This can be easily done in $O(1)$ time. (In Theorem 2, we show that it takes only $O(1)$ time to update these conical sections.)

□

to specification), Lemma 10 implies that, regardless of the selection of positions at p, q and r, a feasible solution exists that places a circle of size R^*/ρ at a. We now handle the third case.

Lemma 11 *Let P_1 and P_2 denote the* **P***-regions of point $a \in S$. Pick $p, q \in S$ such that we have $p, q \in C''_a$ and thus $N(C''_a) = 2$. Then, given the sets $X_a \cap (P_1 - P_2)$ and $X_a \cap (P_2 - P_1)$, two feasible positions for each of the points a, p and q can be selected in $O(1)$ time.*

Proof. By Part 3 of Lemma 9, we have $N(C'_a) = N(C''_a) = 2$. Thus we assume $N(C'_a) = N(C''_a)$. (Note that we can treat the cases with $N(C'_a) = N(C''_a) = 0$ and $N(C'_a) = N(C''_a) = 1$ as special cases of this lemma, by ignoring the corresponding points in $N(C'_a)$.) Additionally by Part 5 of the same lemma, we get $N(C'''_a) = 0$. Thus we have $\delta(a,p) > 0.1R^*$ and $\delta(a,q) > 0.1R^*$. We break the analysis into two subcases.

Case 1: $\delta(p,q) \leq 0.4R^*$.

In this case, Corollary 5 implies that a, p, and q can be labeled separately from the rest of the points.

Case 2: $\delta(p,q) > 0.4R^*$.

Let us first consider the point p. We have $a \in C''_p$, $a \in C''_q$, and $q \in C'_p$. Without loss of generality, assume that points p and a share the same ordnate. Since we have $N(C''_p) > 0$, by Lemma 9, we also have $N(C'_p) \leq 3$. Thus $N(C'_p)$ can contain at most one other input point; denote it by $r \in S - \{a, p, q\}$. Since we have $N(C''_a) = 2$, we can write $\delta(a,r) > 0.8R^*$.

Let us now determine the possible positions of the input points q and r. First consider the point q. Since we have $\delta(p,q) > 0.4R^*$ and $\delta(a,q) < 0.4R^*$, we can conclude $\angle apq < 90°$ (see Figure 17). Consider the point r. Since we have $\delta(a,r) > 0.8R^*$, r cannot lie inside C''_p. As shown in Figure 17, the shaded region represents the possible position of r inside C'_p. We can easily verify $\angle apr > 70°$. Without loss of generality, let us assume that r lies on the right side.

Since we have $N(C'_p) > 0$, p must have at least one **D**-region. By Corollary 1, the number of **D**-regions at p cannot exceed two. Thus we distinguish between two sub-cases: (i) p has one **D**-region and (ii) p has two **D**-regions.

Case 2(a): p has one D-region.

Let us first consider the case where p has one **D**-region; we let this **D**-region be

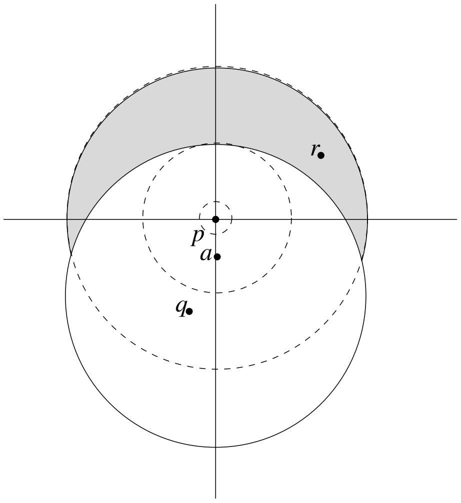


Figure 17: The possible positions of r.

denoted by $D = Cone(p, \theta_{p4}, \theta_{p1})$, with $\theta_{p4} > \theta_{p1}$. Since r lies on the right-hand side, θ_1 lies in the first or second quadrants, i.e., we have $\theta_1 < 180°$. In this case, the points a, q, and r lie in D. Thus the region $Cone(p, \theta_{p1}, \theta_{p4})$ represents a contiguous set of pending positions. Since each **P**-region must be adjacent to a **D**-region, p can have at most two **P**-regions. Let us denote them by $P_1 = Cone(p, \theta_{p1}, \theta_{p2})$ and $P_2 = Cone(p, \theta_{p3}, \theta_{p4})$. Our aim is to show that we can select one position from each of these **P**-regions resulting in a pair of feasible positions.

Let $U_{pa,i}, U_{pq,i}, U_{pr,i} \subseteq P_i$, $1 \leq i \leq 2$, denote the maximal uniform-regions of p w.r.t. a, q, and r, respectively, with $U_{pa,1}$, $U_{pq,1}$ and $U_{pr,1}$ are adjacent to θ_1. Because r lies on the right side and we have $\delta(a,p) > 0.1R^*$, $U_{pa,1}$ and $U_{pr,2}$ are empty—they do not exist. (In other words, for any θ_a and any position $\theta_p \in U_{pa,1} - U_{pa,2}$, if $C(a, \theta_a, R^*/\rho)$ intersects $C(p, \theta_p, R^*/\rho)$ then the ρ-enlarged circle $C(a, \theta_a, R^*)$ contains the point r and thus θ_a becomes a dead position.) Let U_{p1} denote conical region of minimum angle surrounding both $U_{pq,1}$ and $U_{pr,1}$. Suppose the minimum angles of both $U_{pq,1}$ and $U_{pr,1}$ are greater than zero. By Lemma 6, U_{p1} is a clique-region of both q and r. Clearly, any position in $U_{p1} \cap X_p$ and any position in $P_2 - U_{p1}$ form a pair of feasible positions w.r.t. both q and r. Consider the point a. Since $U_{pa,1}$ is empty, every position in $P_1 - U_{pa,2} \cap X_p$ is feasible w.r.t. a. Thus we can select two positions, one in each P_i, that are feasible w.r.t. a, q, and r.

Case 2(b): p has two D-regions.

In this case, by Lemma 8, we know that both P_1 and P_2 are clique-regions w.r.t. r. Thus we can ignore the point r, as any position in P_i, $1 \leq i \leq 2$, is a feasible position w.r.t. r. Since this reasoning also holds for q, we can conclude that both p and q can be labeled separately from the rest of the points in $S - \{a\}$.

In all the above cases, the selection of feasible positions θ'_a and θ''_a is similar to

Lemma 10. Hence two feasible positions can be selected in $O(1)$ time. □

7.3. The position Selection Procedure

Now our selection algorithm is clear. We assume that the sets P_1 and P_2 are given as part of the input—they can be computed in polynomial time. The selection algorithm first selects positions for each point $a \in S$ obeying $N(C''_a) \leq 1$ as discussed in Lemma 10; it then selects two positions for each point a and the points that lie in C''_a obeying $N(C''_a) = N(C'_a) = 2$ using local search; let H denote these positions. The selection procedure is summarized in Figure 18. Note that the running time of the selection procedure is independent of the number of positions k, if the labels can be placed in any position. It does depend on k if the set of positions is given as a part of input. These facts will be used in the next two sections for analyzing the running time of the algorithm.

```
PROCEDURE SELECT

  • Input: A label radius bound $\Delta$, $S = (S_1, S_2)$ and P-regions associated with
    each point.
    Remark: The partition of points in $S$ into $S_1$ and $S_2$ is on the basis of
    Lemmata 11 and 10; i.e., $a \in S_1 \iff N(C''_a) \leq 1$, and, $a \in S_2 \iff N(C''_a) =$
    $2$,

  • Output: $H = a$ set of two feasible positions $\{\theta'_a, \theta''_a \;\; |a \in S\}$, for each point
    in $S$.

  1. $S'_1 \leftarrow S_1$, $S'_2 \leftarrow S_2$, and $H \leftarrow \phi$.
  2. While($|S'_2| > 0$)
       (a) Let $a \in S'_2$ and $p, q \in C''_a$.
       (b) Select pairs $\theta'_a$ and $\theta''_a$, $\theta'_p$ and $\theta''_p$, and $\theta'_q$ and $\theta''_q$ for the points
           $a$, $p$, and $q$ respectively (they must exist by Lemma 11).
       (c) $H \leftarrow H \cup \{\theta'_a, \theta''_a, \theta'_p, \theta''_p, \theta'_q, \theta''_q\}$,    $S'_2 \leftarrow S'_2 - \{a, p, q\}$
  3. While($|S'_1| > 0$)
       (a) Let $a \in S'_1$.
       (b) Select $\theta'_a$ and $\theta''_a$ as in Lemma 10.
       (c) $H \leftarrow H \cup \{\theta'_a, \theta''_a\}$,    $S'_1 \leftarrow S'_1 - \{a\}$
```

Figure 18: Description of the selection procedure.

8. Overall Algorithm and Its Analysis

Before we describe our algorithm, we need one last concept.

Definition 7 *Given a set S of points, the* diameter *of any subset of points is the maximum distance between any two points in the subset. The* minimum 3-diameter *of S, denoted $D_3(S)$, is the smallest value of the diameter among all subsets of S*

of size 3.
Two important facts about $D_3(S)$ in our context are:
Proposition 2 *The minimum* 3*-diameter* $D_3(S)$ *of a set of points* S *can be computed in* $O(n \log n)$ *time* [6,12]*; and* $D_3(S)/5.26 \leq R^* \leq 3.74 D_3(S)$ [28].
Let Δ denote the current label radius. Using the known bounds on the size of optimal labels in terms of $D_3(S)$ (Proposition 2), we do a binary search in steps of ϵ_1 over the interval $[\frac{D_3}{3.6 \times 5.26}, 3.74 D_3]$ [i] to find the largest value of Δ for which we can place non-intersecting labels at each point and return this as our final solution. The algorithm is formally described in Figure 19. Procedure Test-Satisfiable decides whether labels of size Δ can be placed at each point. We start with two **P**-regions for each point. At each step, we update the **P**-regions based on the new Δ. We then call Procedure Select, which selects two feasible positions for each point, transform the resulting collection into an instance of the *2SAT* problem, and check whether the resulting instance is satisfiable. Figure 20 summarizes these steps.

Algorithm Main

- **Input:** A set S of points along with a set of positions X_a for each point $a \in S$ and a required precision $\epsilon > 0$.
- **Output:** A set of uniform circular labels, one for each point in S. The radius of the labels is at least $R^*/(3.6 + \epsilon)$, where R^* denotes the radius of labels in an optimal solution.

1. Set $\Delta = 0$ and $i = 0$. Compute $D_3(S)$. For each $a \in S$, compute the set B_a of its 4 closest neighbors.
2. Perform a binary search over the interval $\mathcal{I} = [D_3(S)/(3.6 \times 5.26), 3.74 D_3(S)]$ with a spacing of $\epsilon_1 = \epsilon/3.6$ to find maximum value of Δ (denoted Δ_h) such that Procedure Test-Satisfiable(Δ, B_a) returns **true**.
3. Return the labels and their sizes as used in the construction of instance I_Δ in Procedure Test-Satisfiable(Δ_h, B_a).

Figure 19: Description of the main algorithm.

8.1. Analysis

We now analyze the performance guarantee and the running time of Algorithm Main.
Theorem 2 *Given a set* S *of* n *of points, presorted lists* X_a *of positions for each point* $a \in S$*, and for any* $\epsilon > 0$*,* Algorithm Main *labels all points in* S *with non-intersecting uniform circular labels of radius at least* $R^*/(3.6 + \epsilon)$ *in* $O(n \log n + n \log(R^*/\epsilon) \log(k))$ *time.*
Proof. Correctness of Binary Search: Observe that we do a binary search in the interval $[\frac{D_3}{3.6 \times 5.26}, 3.74 D_3]$. In order for the binary search to work, the procedure

[i] Performing a search over an interval $[a, b]$ in steps of ϵ means searching at all values $a + k\epsilon$, $k = 0, \dots, \lceil \frac{a-b}{\epsilon} \rceil$.

```
PROCEDURE TEST-SATISFIABLE

  • Input: the set of input points $S$, an estimate $\Delta$ on the radius of the circular
    label and sets $B_a$ for all $a \in S$.
  • Output: Return true if the set $S$ can be labeled with labels of size $\Delta$, false
    otherwise.

  1. Set the radii $C'_a$ and $C''_a$ to $0.4\Delta$ and $0.8\Delta$ respectively. Determine $N(C'_a)$
     and $N(C''_a)$ for each $a$ from its set of closest neighbors.
  2. Construct the P-regions for each point $a \in S$ using $\Delta$ as label radius and
     using the data structure computed in Step 1 of ALGORITHM MAIN.
  3. $H \leftarrow$ SELECT$(S, \Delta, \mathbf{P})$.
  4. Construct a 2SAT instance $I_\Delta$ from $H$.
  5. If $I_\Delta$ is satisfiable, then return true, the instance $I_\Delta$ and $H$, else return false.
```

Figure 20: The testing procedure

Test-Satisfiable must output **true** for all $R \leq R^*/3.6$ and **false** for $R > R^*$. The procedure Test-Satisfiable returns **false** if $R > R^*$, since R^* is the optimal labeling size. We argue that all the results in Sections 5 through 7 hold for any $R \leq R^*$. First consider Lemma 1 and consider an input point p such that $\delta(a,p) > 0.8R^*$. For any $R < R^*$, we still have $\delta(a,p) > 0.8R$. Since C'_a and C''_a are defined w.r.t. $R = R^*$, the input point p still lies outside C'_a for any $R < R^*$. This implies that Lemma 1 still holds for any $R < R^*$.

We derived the results in the Sections 5 and 6, by setting $R = R^* = 1$ and keeping $\delta(a,p)$ as a fraction of R, i.e., a fraction of 1. Thus for any $R < R^*$, the value of $\delta(a,p)$, i.e., the fraction, will only increase, implying $N(C'_a)$ and thus $N(C''_a)$ is non-increasing. Consider a and p such that $\delta(a,p) < R^*$ and consider the expressions derived in Section 5. For any smaller $R < R^*$, we have two possibilities: p lies either (i) inside C'_a or (ii) outside of C'_a. Consider the first case. Since in this case $\delta(a,p)$ increases, the angle of a maximal clique-region of a w.r.t. p increases and the angle of maximal uniform-region of a w.r.t. p decreases. Therefore, as R gets smaller, we have potentially more feasible positions. Similarly, in Section 6, Lemma 9 holds under general conditions. Now consider the second possibility, i.e., p lying outside C'_a. In this case, by Lemma 1, we can ignore p. Combining these facts we can conclude that Test-Satisfiable must output **true**, for all $R \leq R^*/3.6$.

Performance guarantee: By Steps 2 and 3 of ALGORITHM MAIN, letting Δ_h be the heuristic label size and R^* be the optimal label size for the point set S, we get that $R^* \leq 3.6(\Delta_h + \epsilon/3.6) = 3.6\Delta_h + \epsilon$, the required performance guarantee.

Running time: For each point a, the points of S that lie in C'_a must be determined. Since we have $N(C'_a) \leq 5$, these points can be computed in $O(n \log n)$ time with the algorithm of Dickerson *et al.* [7], after which the algorithm takes only $O(n)$ time to compute **P**- and **D**-regions for all points. We first determine the running times of Procedure Select and Procedure Test-Satisfiable.

The selection of feasible positions in Lemmata 10 and 11, Procedure Select requires the computation of $P_1 - P_2$ and $P_2 - P_1$, which can be done in $O(n)$ time for all the points.

We have two cases to consider: (i) $k = O(|S|)$ and (ii) $k = \infty$.

$k = O(|S|)$: This case requires the determination of two sets $X_a \cap (P_1 - P_2)$ and $X_a \cap (P_2 - P_1)$. That is to idenitify sets of positions of X_a that lie within the conical sections $(P_1 - P_2)$ and $(P_2 - P_1)$. Since X_a is a sorted list of positions, the selection can be done efficiently. First identify the extreme positions of the sets $P_1 - P_2$ and $P_2 - P_1$. This takes only $O(1)$ time per point. Then we do a binary search of these extreme positions on the list X_a to determine the sublists that form the sets $X_a \cap (P_1 - P_2)$ and $X_a \cap (P_2 - P_1)$. Since we have only four extreme positions, the binary search takes only $O(\log k)$ time per point assuming X_a is presorted. Thus this step takes $O(n \log k)$ time for all the points.

$k = \infty$: In this case, we need to compute only the conical sections $(P_1 - P_2)$ and $(P_2 - P_1)$. Hence this step takes only $O(n)$ time.

follows.

Next consider Procedure Test-Satisfiable. Step 1 of the procedure takes $O(n)$ time. Step 2 makes one call to Procedure Select and thus takes $O(\log k)$ time. Step 3 takes $O(n)$ time since we create at most one variable and a constant number of clauses per point (because we need to consider the interaction of this point with no more than its four closest neighbors). Step 4 takes $O(n)$ time using the well-known algorithm for solving *2SAT* [15,17], because the instance produced in Step 3 has a linear number of variables and clauses. Thus one call to Procedure Test-Satisfiable takes $O(n)$ time to compute.

Let us now complete the argument by determining the execution time of Algorithm Main. By Proposition 2, $D_3(S)$ can be computed in $O(n \log n)$ time. Using the algorithm of [7], the five nearest neighbors of each point can be calculated in advance in $O(n \log n)$ time. Thus the total running time of Step 1 of Algorithm Main is $O(n \log n)$. By Proposition 2, we need $O(\log(R^*/\epsilon))$ iterations of Step 2; and each iteration of Step 2 makes one call to Procedure Test-Satisfiable. Thus the total running time of Step 2 is $O(n \log(R^*/\epsilon) \log(k))$. Step 3 takes $O(n)$ time. Thus the total running time of the algorithm is $O(n \log n + n \log(R^*/\epsilon) \log(k))$. For $k = \infty$, our algorithm runs in time $O(n \log n + n \log(R^*/\epsilon))$. □

This theorem assumes that k in the KPML problem is polynomial in the number of points. It also does not deal with potential time savings resulting from the maintenance of **P**-regions from iteration to iteration, something easily done since **P**-regions must decrease monotonically as the label size increases.

9. Extensions

First consider placing regular polygonal labels at each point. By using appropriate lower and upper bounds obtained via inscribed and circumscribed circles, it is easy to see that, for all $\epsilon > 0$, we obtain an $O(n \log n + n \log(R^*/\epsilon))$ time approximation algorithm for the KPML problem when restricted to any regular polygon. The performance guarantee is upper bounded by $(3.6\sqrt{2}+\epsilon)$. In fact, the algorithm even works if we are allowed a fixed set of regular polygons as surrogates for labels, with each point having a different set of allowable positions. Note that, for regular polygons, an allowable position is characterized by the angular displacement of the position as well as the orientation of the polygon. Note also that we have not attempted to optimize the bounds for regular polygons.

Now consider the case where a label can be placed in any position. This model is also known as arbitrary orientation model and was considered in [8] and [28]. In this case, we have $k = \infty$. A naive application of Theorem 2 will yield an algorithm whose running time is not polynomially bounded due to the nature of its dependence on k. Nevertheless, as discussed earlier (section 7.3) the selection procedure in fact takes $O(1)$ time per point. that The rest of the details are straightforward. We thus get the following result.

Theorem 3 *In the arbitrary orientation model, given a set S of n points, and for any $\epsilon > 0$,* Algorithm Main *labels each point at in S with uniform circular labels of size at least $R^*/(3.6 + \epsilon)$ in $O(n \log n + n \log(R^*/\epsilon))$ time.*

Thus, as mentioned earlier, Theorem 3 improves upon the previous performance guarantees for the map-labeling problem considered by [8] and [28].

10. Conclusions

We have described an efficient approximation algorithm for the *k-Position Map-Labeling Problem*: given a set of points in the plane and, for each point, a set of up to k allowable positions, place uniform and non-intersecting labels of maximum size at each point in one of the allowable positions. The approximation algorithm can be easily extended to obtain an efficient algorithm for placing labels in arbitrary positions.

Based on the problem considered in this paper and the recent interest in placing multiple labels for each point, we propose the following general map-labeling problem for further investigation:

Multilabel, multiposition map-labeling: given a set S of points in the plane, a set of positions Θ, a collection $\{\mathcal{R}_i\}$ of subsets $\mathcal{R}_i \subseteq \Theta^{r_i}$ of allowable combinations of positions,[j] one for each point $s_i \in S$, find a placement of r_i uniform labels of maximum size at each point s_i so that (i) each label occupies a distinct position and (ii) the set of positions P_i occupied by these labels belongs to the allowable position combinations $\mathcal{R}_i$.

This problem formulation can be easily extended to the weighted case and can also be extended to specify relations that capture allowable positions between sets of points rather than for a given point. The general problem allows much more flexibility to the cartographer in terms of label placement. One motivation for this comes from a similar formulation of general constraint satisfaction problems that

[j] These positions may or may not be intersecting, depending on the cartographer's requirements.

have been studied in computer science. We have begun a preliminary study of this problem in [9]. See [33] for similar formulations motivated by constraint satisfaction problems.

Given the applications of automated map-labeling in Geographic Information Systems, in which some of the points may represent mobile objects, a separate problem of significant interest is to devise dynamic algorithms for the maintenance of a collection of labels, either within a conventional framework of dynamic algorithms or within a kinetic framework.

11. Acknowledgments

We sincerely thank the two anonymous referees for an extremely careful and thorough reading of the manuscript. Their suggestions have substantially improved our presentation and also helped us correct a number of inconsistencies and errors in the earlier version of the paper. We thank Professor Mark de Berg (Utrecht University) for his help during the review process. The first author thanks Dr. Xinhui Niu and Dr. Nickhil Jakatdar of Timbre Technologies for their support.

References

1. P. Agarwal, M. van Kreveld, and S. Suri. "Label placement by maximum independent set in rectangles" *Computational Geometry: Theory and Applications,* (1998), 11:209-218.
2. ACM Computational Geometry Impact Task Force. "Application challenges to computational geometry," Princeton U. Technical Report TR-521-96, available on-line at `http://www.cs.princeton.edu/~chazelle/pubs/CGreport.ps.Z` Journal version in *Advances in Discrete and Computational Geometry* B. Chazelle, J. Goodman and R. Pollack Eds., 223:407-463, 1999.
3. D. Beus and D. Crockett, "Automated production of 1:24,000 scale quadrangle maps," *Proc. 1994 ASPRS/ACSM Ann. Convention and Exposition* **1** (1994), 94–99.
4. J. Christensen, J. Marks, and S. Shieber, "Algorithms for cartographic label placement," *Proc. ASPRS/ACSM Convent. & Exp.* (1993), 75–89.
5. J. Christensen, J. Marks, and S. Shieber, "An Empirical Study of Algorithms for Point-Feature Label Placement," *ACM Transactions on Graphics* **14** (1995), 203–232.
6. A. Datta, P. Lenhof, C. Schwarz and M. Smid, "Static and dynamic algorithms for k-point clustering problems," *Proc. 3rd Workshop on Algs. and Data Structures WADS-93*, Springer-Verlag LNCS **709** (1993), 265–276.
7. M.T. Dickerson, R.L. Drysdale, and J.R. Sack, "Simple algorithm for enumerating interpoint distances and finding k nearest neighbors," *Int'l J. Comput. Geom. and Appl.* **2**, 2 (1992), 221–239.
8. S. Doddi, M. Marathe, A. Mirzaian, B.M.E. Moret, and B. Zhu, "Map labeling and its generalizations," *Proc. 8th ACM-SIAM Symp. on Disc. Algs. SODA-97*, SIAM Press (1997), 148–157.
9. S. Doddi, M. Marathe and B.M.E. Moret. "Multiposition, multilabel map labeling," Unpublished manuscript, February 2001.
10. J. Doerschler and H. Freeman, "A rule-based system for cartographic name place-

ment," *Commun. ACM* **35** (1992), 68–79.

11. T. Erlebach, K. Jansen and E. Seidel, Polynomial-time approximation schemes for geometric graphs. *Proc. 12th ACM-SIAM Symp. on Disc. Algs. SODA-01*, SIAM Press (2001), 671–679.
12. D. Eppstein and J. Erickson, "Iterated nearest neighbors and finding minimal polytopes," *Discrete & Comput. Geom.* **11** (1994), 321–350.
13. M. Formann and F. Wagner, "A packing problem with applications to lettering of maps," *Proc. 7th ACM Symp. Comput. Geom. SoCG-91*, ACM Press (1991), 281–288.
14. S. Hirsch, "An algorithm for automatic name placement around point data," *The American Cartographer* **9**, 1 (1982), 5–17.
15. M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco CA, 1979.
16. D. Hochbaum and D. Shmoys, "A unified approach to approximation algorithms for bottleneck problem," *J. ACM* **33**, 3 (1986), 533–550.
17. D. Hochbaum (ed.). *Approximation Algorithms for NP-Hard Problems.* PWS Publishing Company, Boston, MA, 1997.
18. H. Hunt III, M. Marathe, V. Radhakrishnan, S. Ravi, D. Rosenkrantz, and R. Stearns. "NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs," *J. Algorithms* **26** (1998), 238–274.
19. C. Iturriaga. *Map Labeling Problems.* PhD thesis, University of Waterloo, 1999.
20. E. Imhof, "Positioning names on maps," *The American Cartographer* **2** (1975), 128–144.
21. C. Jones, "Cartographic name placement with Prolog," *IEEE Computer Graphics and Applications* **5** (1989), 36–47.
22. M. van Kreveld, T. Strijk, and A. Wolff. "Point set labeling with sliding labels," *Proc. 14th ACM Symp. Comput. Geom. SoCG-98*, ACM Press (1998), 337–346. Final version appears in *Comput. Geometry: Theory and Appls.* **13** (1999), 21–47.
23. L. Kucera, K. Mehlhorn, B. Preis, and E. Schwarzenecker, "Exact algorithms for a geometric packing problem," *Proc. 10th Symp. on Theor. Aspects of CS STACS-93*, in LNCS **665**, Springer-Verlag (1993), 317–322.
24. J. Marks and S. Shieber, "The Computational Complexity of Cartographic Label Placement," *TR-05-91, Advanced Research in Computing Technology, Harvard University*, March (1991).
25. B.M.E. Moret, *The Theory of Computation.* Addison-Wesley, Reading, MA, 1998.
26. C.K. Poon, B. Zhu and F. Chin, "A polynomial-time solution for labeling a rectilinear map," *Proc. 13th Ann. ACM Sympos. Comput. Geom. SoCG-97*, ACM Press (1997), 451–453.
27. S. Poon, C. Shin, T. Strijk, and A. Wolff, "Labeling points with weights," *Proc. 17th European Workshop of Comput. Geometry EuroCG-01* (2001), 97–100.
28. T. Strijk and A. Wolff, "Labeling points with circles," *Int'l J. Comput. Geometry and Appls.* **11(2)** *(April 2001) 181–195.*
29. T. Strijk. "Geometric Algorithms for Cartographic Label Placement," Ph.D. thesis, Utrecht University, Department of Computer Science, January 2001.
30. F. Wagner, "Approximate map labeling is in $\Omega(n \log n)$," *Inform. Process. Lett.* **52** (1994), 161–165.

31. F. Wagner and A. Wolff, “Map labeling heuristics: provably good and practically useful,” *Proc. 11th Ann. ACM Sympos. Comput. Geom. SoCG-95*, ACM Press (1995), 109–118.

32. A. Wolff. “Map Labeling,” *Diploma Report, Freie University, Berlin, Department of Mathematics.* (1995).

33. A. Wolff. “Automated Label Placement in Theory and Practice,” *Ph.D. thesis, Freie University, Berlin, Department of Mathematics and Computer Science.* May (1999).

34. Z. Qin, A. Wolff, Y. Xu, and B. Zhu, ”New algorithms for two-label point labeling,” *Proc. 8th Annual Europ. Symp. on Algorithms ESA-00* in LNCS **1879**, Springer Verlag (2000), 368–379.

35. `http://www.math-inf.uni-greifswald.de/map-labeling/bibliography/.`