International Journal of Computational Geometry & Applications
© World Scientific Publishing Company

A SIMPLE FACTOR-2/3 APPROXIMATION ALGORITHM FOR TWO-CIRCLE POINT LABELING

ALEXANDER WOLFF
Institut für Mathematik und Informatik, Universität Greifswald, Germany
awolff@uni-greifswald.de

MICHAEL THON
Georg-August-Universität Göttingen, Germany
mithon42@gmx.de

and

YINFENG XU*
School of Management, Xi'an Jiaotong University, Xi'an, China
yfxu@xjtu.edu.cn

Received 2 April 2001
Revised 12 September 2001
Communicated by T. Tokuyama, Guest Editor

ABSTRACT

Given a set P of n points in the plane, the two-circle point-labeling problem consists of placing $2n$ uniform, non-intersecting, maximum-size open circles such that each point touches exactly two circles.
It is known that this problem is NP-hard to approximate. In this paper we give a simple algorithm that improves the best previously known approximation factor from $4/(1+\sqrt{33}) \approx 0.5931$ to 2/3. The main steps of our algorithm are as follows. We first compute the Voronoi diagram, then label each point *optimally* within its cell, compute the smallest label diameter over all points and finally shrink all labels to this size. We keep the $O(n \log n)$ time and $O(n)$ space bounds of the previously best algorithm.

Keywords: Computational geometry, cartography, approximation algorithm, multi-label point labeling, Voronoi diagram

1. Introduction

Label placement is one of the key tasks in the process of information visualization. In diagrams, maps, technical or graph drawings, features like points, lines, and polygons must be labeled to convey information. The interest in algorithms that automate this task has increased with the advance in type-setting technology and the

*Research supported by the NSF of China under grant No. 19731001.

amount of information to be visualized. Due to the computational complexity of the label-placement problem, cartographers, graph drawers, and computational geometers have suggested numerous approaches, such as expert systems,[2,9] zero-one integer programming,[26] approximation algorithms,[7,10,21,24] simulated annealing[6] and force-driven algorithms[11] to name only a few. An extensive bibliography about label placement can be found at Ref. [22]. The ACM Computational Geometry Impact Task Force report[5] denotes label placement as an important research area. Manually labeling a map is a tedious task that is estimated to take 50 percent of total map production time.[14]

In this paper we deal with a relatively new variant of the general label placement problem, namely the two-label point-labeling problem. It is motivated by maps used for weather forecasts, where each city must be labeled with two labels that contain for example the city's name or logo and its predicted temperature or rainfall probability.

The two-label point-labeling problem is a variant of the one-label problem that allows sliding. Sliding labels can be attached to the point they label anywhere on their boundary. They were first considered by Hirsch[11] who gave an iterative algorithm that uses repelling forces between labels in order to eventually find a placement without or with only few intersecting labels. Van Kreveld et al.[20] gave a polynomial time approximation scheme and a fast factor-2 approximation algorithm for maximizing the number of points that are labeled by axis-parallel sliding rectangular labels of common height. They also compared several sliding-label models with so-called fixed-position models where only a finite number of label positions per point is considered, usually a small constant like four.[6,10,21] Sliding rectangular labels have also been considered for labeling rectilinear line segments.[13] The problem of labeling points with arbitrarily oriented sliding labels, a generalization that is of interest in the case of graphical, not textual labels, was investigated as well.[7,25]

Point labeling with circular labels, though not as relevant for real-world applications as rectangular labels, is a mathematically interesting problem. The one-label case has already been studied extensively.[7,8,19] For maximizing the label size (i.e. diameter), the best approximation factor[8] now is 1/3.6.

The two- or rather multi-label placement problem was first considered by Kakoulis and Tollis[12] who presented two heuristics for labeling the nodes and edges of a graph drawing with several rectangles. Their aim was to maximize the number of labeled features. The algorithms are based on their earlier work; one is iterative, while the other uses a maximum-cardinality bipartite matching algorithm that matches cliques of pairwise intersecting label positions with the elements of the graph drawing that are to be labeled. They do not give any runtime bounds or approximation factors.

For maximizing the size of square labels, two per point, Zhu and Poon[24] gave the first approximation algorithm. It had an approximation factor of 1/4 that was subsequently improved to 1/3 by Zhu and Qin[25] and to 1/2 by Qin et al..[16]

For the problem that we consider in this paper, namely maximizing the size of circular labels, two per point, again Zhu and Poon[24] gave the first approximation algorithm. They achieved an approximation factor of 1/2. Like all of the following algorithms, their algorithm relies on the fact that there is a region around each input point p such that the region of p does not intersect the region of any other input point. The size of the region—in their case a circle centered at p—makes it possible to place labels whose size is a constant fraction of an upper bound for the maximum label size.

Later Qin et al.[16] improved this result. They gave an approximation algorithm with a factor of $1/(1 + \cos 18°) \approx 0.5125$. They also state that it is NP-hard to decide whether a set of points can be labeled with $2n$ disjoint unit circles, and, even more, that the maximum label size cannot be approximated arbitrarily well unless $\mathcal{P} = \mathcal{NP}$. The regions into which they place the labels are the cells of the Voronoi diagram. They do not compute the Voronoi diagram explicitely, but use certain properties of its dual, the Delaunay triangulation. For estimating the approximation factor of their algorithm they rely on the same upper bound for the maximum label size as Zhu and Poon, namely the minimum (Euclidean) distance of any pair of input points.

Recently Spriggs and Keil[18] presented an algorithm with an approximation factor of $4/(1 + \sqrt{33}) \approx 0.5931$. They use a different upper bound for the maximum label size than the previous two algorithms.

In this paper we give an algorithm that places the labels of each point into its Voronoi cell. However, unlike the algorithm by Qin et al.[16] we do this optimally and compare the label diameter of our algorithm not to an upper bound but directly to the optimal label diameter. This yields an approximation factor of 2/3. At the same time we keep the $O(n \log n)$ time and $O(n)$ space bounds of the previous algorithms, where n is the number of points to be labeled. Both the idea of our new algorithm and the proof of its approximation factor are simpler than those of its predecessor. If a library with some geometric algorithms is available, then the implementation is straight forward as well.

The actual label placement within the Voronoi cells is a special case of a gift wrapping problem, where the gift is a coin (as large as possible) and the wrapping a convex polygonal piece of paper with m edges that can only be folded once along a line. Our problem is special in that it specifies a point on the folding line and thus takes away a degree of freedom. For the problem without this restriction, there is an optimal linear-time algorithm.[4] For our special problem we gave an $O(m \log m)$-time algorithm in a previous version of this paper.[23] That algorithm relies on standard techniques for computing the lower envelope of a set of "well-behaving" functions. In this paper we give a very simple and geometrical algorithm that takes only linear time. We have implemented this algorithm in Java. An interactive demo is available online at *http://www.math-inf.uni-greifswald.de/map-labeling/points/two-circles/*.

This paper is organized as follows. In Section 3 we prove that in each cell of the Voronoi diagram of the given point set P there is enough space for a pair of uniform circular labels whose diameter is 2/3 times the maximum diameter for labeling P with circle pairs. In Section 4 we show how to label points optimally within their Voronoi cells and state our central theorem. Finally in Section 5 we show why our method does not work well on related labeling problems.

Throughout this paper we consider labels to be topologically *open*, i.e. labels may touch each other. We define the *size* of a solution to be the diameter of the uniform circular labels. A solution is *optimal* if no two labels intersect and labels have the largest possible size. We will only consider sets of at least two points since the size of an optimal solution is unbounded otherwise.

2. Previous Work

Zhu and Poon[24] have suggested the first approximation algorithm for the two-circle point-labeling problem. Their algorithm always finds a solution of at least half the optimal size. The algorithm is very simple; it relies on the fact that D_2, the minimum Euclidean distance between any two points of the input point set P, is an upper bound for the optimal label size (i.e. diameter), see Figure 1. On the other hand, given two points p and q in P, the two open circles $C_{p,D_2/2}$ and $C_{q,D_2/2}$ with radius $D_2/2$ centered at p and q do not intersect. Thus if each point is labeled within its circle, no two labels will intersect. Clearly this allows labels of maximum diameter $D_2/2$, i.e. half the upper bound for the optimal label size.

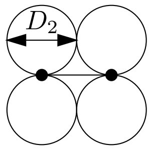

Fig. 1. D_2 is an upper bound for the optimal label size.

The difficulty of the problem immediately comes into play when increasing the label diameter d beyond $D_2/2$, since then the intersection graph of the (open) disks $C_{p,d}$ with radius d centered at points p in P changes abruptly; the maximum degree jumps from 0 to 6.

Later Qin et al.[16] gave an approximation algorithm that overcomes this difficulty and labels all points with circles slightly larger than the threshold of $D_2/2$. Their diameter is $d^\star = D_2/(1 + \cos 18°) \approx 0.5125\, D_2$. Their algorithm also assigns each point a certain region such that no two regions intersect and each point can be labeled within its region. The regions they use are not circles but the cells of the Voronoi diagram[3] of P, a well-known multi-purpose geometrical data structure. Instead of computing the Voronoi diagram explicitely they use the dual of the Voronoi diagram, the Delaunay triangulation $\mathrm{DT}(P)$ to apply a packing argument. $\mathrm{DT}(P)$ is a planar geometric graph with vertex set P and edges for each pair of points that can be placed on the boundary of an open disk that does not contain any other points of P.[3] Qin et al. argue that in $\mathrm{DT}(P)$ each point p can have at most six short edges, where an edge pq is short if the Euclidean distance $d(p, q)$ of p and q is shorter than $2d^\star$. They show that among the lines that go through these short edges there must be a pair of neighboring lines that form an angle α of at least $36°$. They place the circular labels of p with diameter $d^\star$ such that their centers lie on the angular bisector of α. Finally they prove that these labels lie completely within the Voronoi cell $\mathrm{Vor}(p)$ of p. The Voronoi cell of p is the (convex) set of all points in the plane that are closer to p than to any other input point. Thus the Voronoi cells of two different input points are disjoint and the labels of one input point cannot intersect the labels of any other.

Recently Spriggs and Keil[18] have further improved the result of Qin et al.. The algorithm they presented has an approximation factor of $4/(1 + \sqrt{33}) \approx 0.5931$ and works as follows. For each point p Spriggs and Keil determine the orientation θ_p that maximizes the diameter d_p of a region that (a) allows to place labels whose size is a certain fraction λ_{lower} of d_p and (b) guarantees that there is no placement with labels whose size is greater than a fraction λ_{upper} of d_p. Finally they label all points with circles of size $\min_{p \in P} d_p \lambda_{\text{lower}}$; the circle pair of each point p in orientation θ_p. The afore mentioned approximation factor then is $\lambda_{\text{lower}}/\lambda_{\text{upper}}$.

3. The Lower Bound

Our strategy is as follows. We first show that there is a region $Z_{\text{free}}(p)$ around each input point p that does not contain any other input point. The size of $Z_{\text{free}}(p)$ only depends on the optimal label diameter d_{opt}. Let $Z_{\text{label}}(p)$ be the "Voronoi cell" of p that we would get if all points on the boundary of $Z_{\text{free}}(p)$ were input points. We compute the largest label diameter d_{lower} such that two disjoint circular labels of p completely fit into $Z_{\text{label}}(p)$. It turns out that $d_{\text{lower}} = 2/3 \cdot d_{\text{opt}}$. We do not know the orientation of $Z_{\text{label}}(p)$ relative to p, but we know that $Z_{\text{label}}(p)$ lies completely in $\text{Vor}(p)$. Thus our algorithm must go the other way round: we label each point optimally within its Voronoi cell, compute the smallest label diameter over all points and finally shrink all labels to this size. Then we know that each label is contained in the Voronoi cell of its point, and that the labels have at least diameter d_{lower}.

Let $\mathcal{C}_{\text{opt}}$ be a fixed optimal solution of the input point set P. $\mathcal{C}_{\text{opt}}$ can be specified by the label size d_{opt} and an angle $0 \leq \alpha_p < 180°$ for each input point. The angle α_p specifies the position of a line through p that contains the centers of the labels of p (at a distance $d_{\text{opt}}/2$ from p). By convention we measure α_p from the horizontal line (oriented to the right) through p to the line (oriented upwards) through the label centers. In the following we assume that $d_{\text{opt}} = 1$; the input point set can always be scaled so that this is true.

Definition 1 *Let $C_{m,r}$ be an open disk with radius r centered at a point m and let H_{pq} be the open halfplane that contains p and is bounded by the perpendicular bisector of p and q. For each $p \in P$, let the* point-free zone $Z_{\text{free}}(p) = C_{Z_1,\sqrt{3}/2} \cup C_{Z_2,\sqrt{3}/2} \cup C_{p,1}$*, where Z_1 and Z_2 are the centers of the labels L_1 and L_2 of p in a fixed optimal solution $\mathcal{C}_{\text{opt}}$. The* label zone $Z_{\text{label}}(p)$ *is the intersection of all halfplanes H_{pq} with q a point on the boundary of $Z_{\text{free}}(p)$.*

Note that $Z_{\text{free}}(p)$ and $Z_{\text{label}}(p)$ are symmetric to the lines that form angles of α_p and $\alpha_p + 90°$ through p. The size of these areas only depends on d_{opt}, their orientation only on α_p. In Figures 2 and 3 we set $\alpha_p = 0°$.

Lemma 1 *$Z_{\text{free}}(p)$ does not contain any input points except p.*

Proof. Refer to Figure 2. Intuitively speaking, it indicates that the boundary of $Z_{\text{free}}(p)$ is the locus of all potential input points that are as close to p as possible.

First we show that the two disks $C_i := C_{Z_i,\sqrt{3}/2}$ $(i = 1, 2)$ do not contain any input point other than p. Let T_i and B_i be the top- and bottommost points on C_i, respectively. The arc that forms the boundary of $Z_{\text{free}}(p)$ between the points B_1 and T_1 stems from rotating a potential input point q around the label L_1 of p in $\mathcal{C}_{\text{opt}}$ such that the labels of q both touch L_1. This means that the centers of these three labels of diameter 1 form an equilateral triangle whose height $\sqrt{3}/2$ corresponds to the distance of q and Z_1. In other words, if there was a point $q \in P \setminus \{p\}$ with $d(q, Z_1) < \sqrt{3}/2 = \text{radius}(C_1)$, then in $\mathcal{C}_{\text{opt}}$ a label of q would intersect L_1, a contradiction to $\mathcal{C}_{\text{opt}}$ being an optimal solution. Due to symmetry the same holds for C_2.

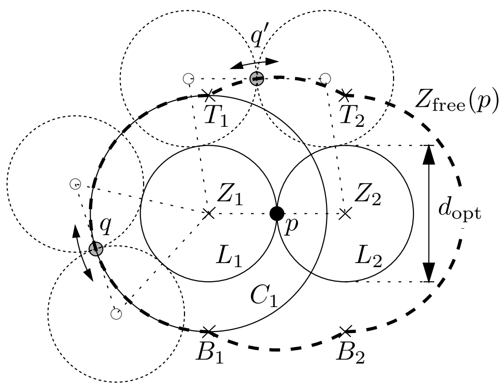

Fig. 2. The point-free zone of p.

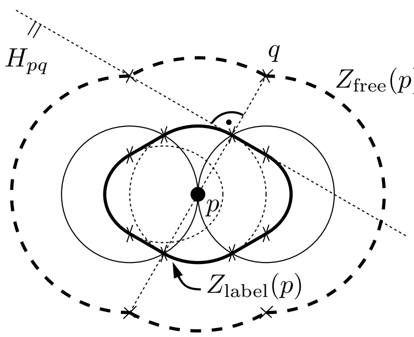

Fig. 3. The label zone of p.

It remains to show that $C_{p,1}$, the third of the three circles that contributes to $Z_{\rm free}(p)$, does not contain any input point other than p. Consider the arc that forms the boundary of $Z_{\rm free}(p)$ between the points T_1 and T_2. This arc is caused by rotating another potential input point q' around p such that each of the labels of q' touches a label of p. Observe that the centers of the four labels of q' and p form a rhombus of edge length 1 when q' moves continously from T_1 to T_2. Since q' and p are the midpoints of two opposite edges of the rhombus, their distance during this movement remains constant. Clearly the distance of the edge midpoints equals the edge length of the rhombus, i.e. 1, which in turn is the radius of $C_{p,1}$. Thus if q' entered the area $C_{p,1} \setminus (C_1 \cup C_2)$, a label of q' would intersect a label of p in $\mathcal{C}_{\rm opt}$: a contradiction. □

Lemma 2 *The Voronoi cell of p contains the label zone $Z_{\rm label}(p)$ of p.*

Proof. The Voronoi cell of p can be be written as $\mathrm{Vor}(p) = \bigcap_{v \in P \setminus \{p\}} H_{pv}$. It contains $Z_{\rm label}(p) = \bigcap_{v' \in \mathrm{boundary}(Z_{\rm free}(p))} H_{pv'}$ since for all input points $v \neq p$ there is a point $v' = \mathrm{boundary}(Z_{\rm free}(p)) \cap \overline{pv}$ such that H_{pv} contains $H_{pv'}$. □

Lemma 3 *For each input point p there are two disjoint circles of diameter $d_{\rm lower} = 2/3 \cdot d_{\rm opt}$ that touch p and lie completely within $Z_{\rm label}(p)$.*

Proof. We do not compute the boundary of $Z_{\rm label}(p)$ (see Figure 3) explicitely but parameterize the radius r of the labels of p such that we get the largest possible labels that do not touch the constraining halfplanes of $Z_{\rm label}(p)$.

Our coordinate system is centered at p and uses four units for $d_{\rm opt}$, see Figure 4. We show that for any point q on the boundary of $Z_{\rm free}(p)$ the halfplane H_{pq} does not intersect the labels of p whose centers we place at $Z_1(-r, 0)$ and $Z_2(r, 0)$. Due to symmetry we may assume that q lies in the first quadrant of our coordinate system. We may further assume that q lies on the circle C_2 since Figure 4 shows that H_{pq} does not add any relevant constraint if q lies on the other circle $C_{p,d_{\rm opt}}$ that contributes to $Z_{\rm free}(p)$. Let q' be the point in the center of the line segment $\overline{pq}$. Then q' lies on the densely dotted circle C_2' with the equation $(x-1)^2 + y^2 = 3$. We parameterize the x-coordinate of q' using the variable $t \in [1, 1+\sqrt{3}]$. The vector $q' = (t, \sqrt{3-(t-1)^2})$ is normal to the boundary h_{pq} of the halfplane H_{pq}. Hence the distance of any point s to h_{pq} is given by the normalized scalar product of $(s - q')$ and q', namely

$$d(s, h_{pq}) = \frac{|(s - q')q'|}{\|q'\|}.$$

For $s = Z_2(r, 0)$, the center of one of the labels of p, we have

$$r = d(Z_2, h_{pq}) = \frac{|rt - 2t - 2|}{\sqrt{2(t+1)}}.$$

Since the enumerator is always less, and the denominator greater than zero, we get

$$r = f(t) = \frac{2(t+1)}{t + \sqrt{2(t+1)}}.$$

The zeros of the derivative f' of f are given by the equation

$$\sqrt{2(t+1)} - 1 - \frac{(t+1)}{\sqrt{2(t+1)}} = 0$$

which yields a minimum of $r = 4/3$ at $t = 1$ for $t \in [1, 1 + \sqrt{3}]$. This value of r corresponds to the largest circle centered at $Z_2(r, 0)$ that lies in all halfplanes H_{pq} for $q \in Z_{\text{free}}(p)$. The diameter of this circle is $2/3 \cdot d_{\text{opt}}$. The case for $Z_1(-r, 0)$ is of course symmetric. □

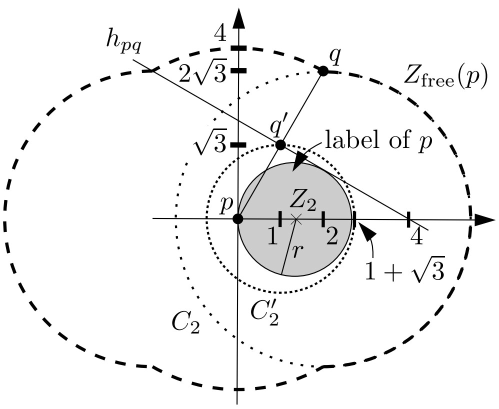

Fig. 4. Estimating the size of the labels of p.

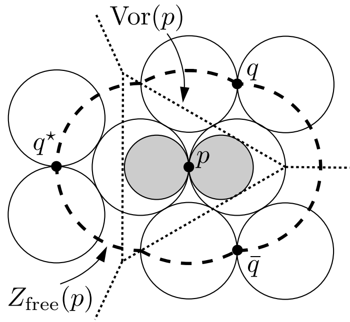

Fig. 5. An example point set showing that our approximation factor is tight.

In Figure 5 we show why the approximation factor of our algorithm is tight. Recall that the point $q = (2, 2\sqrt{3})$ minimizes the size of the largest possible labels of $p = (0, 0)$ in the analysis above (for $t = 1$). Consider the point set $\{p, q, \bar{q}, q^\star\}$, where q, $\bar{q} = (2, -2\sqrt{3})$, and $q^\star = (-2 - 2\sqrt{3}, 0)$ lie on the boundary of $Z_{\text{free}}(p)$. Given these four points, our algorithm labels p with labels (shaded in Figure 5) of diameter 8/3 within the Voronoi cell Vor(p) of p, while the labels in the optimal solution (circles with solid lines) have diameter 4.

4. The Algorithm

In this section we first show how to label a point optimally within its Voronoi cell. We do this in time and space linear in m, the number of edges of the cell. We refer the reader to our interactive online demo mentioned in the introduction. Using the result of the previous section, we give our factor-2/3 approximation algorithm for two-circle point labeling.

We introduce some notation. Let G be a convex polygon, possibly unbounded, $m > 1$ the number of edges of G and p a point in the interior of G. Let μ_p be the function that mirrors each point of the plane at p, and let $G' = G \cap \mu_p(G)$ be the intersection of G and a copy of G point-mirrored at p, see Figure 6. Note that G' contains all sets $M \subseteq G$ that are point-symmetric to p.

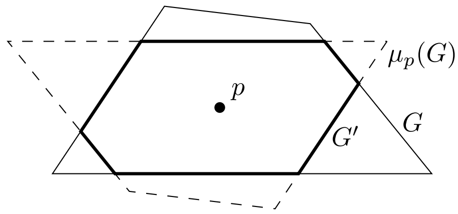

Fig. 6. The polygons G, $\mu_p(G)$ and $G' = G \cap \mu_p(G)$.

The optimal solution of the one-circle labeling problem in G, for short an optimal 1-circle of G, is a maximum-size open circle C_1 that touches p and is contained in G. Similarly, the optimal solution of the two-circle labeling problem in G, for short an optimal 2-circle of G, is a maximum-size open circle C_2 such that C_2 touches p and both C_2 and $\mu_p(C_2)$ are contained in G.

Our strategy is very simple. We reduce the problem of finding an optimal 2-circle in G to that of finding an optimal 1-circle in G'. The polygon G' can be computed in linear time, [15] and we show how to find an optimal 1-circle in G' in linear time with the help of the medial axis. Recall that the medial axis of G is the set of points in G that are centers of circles that lie in G and touch its boundary in at least two points. For the computation of the medial axis we assume that the edges of G are given in cyclic order.

Lemma 4 *The centers of all optimal 1-circles of G lie on the medial axis of G.*

Proof. Let C_1 be an optimal 1-circle of G. If C_1 does not touch the boundary of G in at least two points, then C_1 can be slightly turned around p away from the boundary of G and then stretched by a small amount, contradicting its optimality. $\square$

Lemma 5 *An optimal 1-circle of G can be computed in $O(m)$ time and space.*

Proof. Due to Lemma 4 we can restrict the search for centers of optimal 1-circles to the medial axis of G. Since G is a convex polygon, the medial axis of G can be computed in linear time.[1] It consists of $O(m)$ straight line segments that form a tree whose leaves are the vertices of G. Each (open) line segment s consists of the centers of all circles that lie in G and touch the boundary of G in the same two edges e and f. When constructing the medial axis, each segment can be labeled with these two edges. A candidate for being the center of an optimal 1-circle

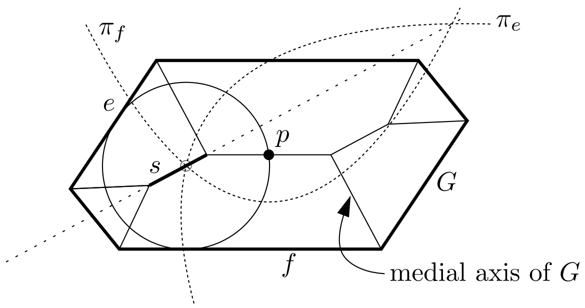

Fig. 7. Constructing candidates for centers of optimal 1-circles of G'.

must not only lie on the medial axis but also on the locus of those points that have equal distance to p and some edge g of G, namely the parabola π_g with focus p and directrix g. Thus the only candidates on s are the intersection points of s and π_e (or π_f), see Figure 7. For each (closed) line segment of the medial axis there are at most two such intersection points, and they can be computed in constant time. Among all $O(m)$ intersection points those with maximum distance to p correspond to optimal 1-circles. □

Lemma 6 *Any optimal 1-circle of G' is an optimal 2-circle of G.*

Proof. G contains G', and G' is point-symmetric to p. Hence if a set M is contained in G', then M and $\mu_p(M)$ lie in G. Therefore each optimal 1-circle of G' is also an optimal 2-circle of G' and a (possibly suboptimal) 2-circle of G. Now let C_2 be an optimal 2-circle of G. According to the definition of optimal 2-circles both C_2 and $\mu_p(C_2)$ lie in G. Since the union of these two circles is a point-symmetric set, it must lie in G'. However, C_2 cannot be larger than an optimal 1-circle of G', which proves the claim. □

Lemma 7 *An optimal 2-circle of G can be determined in $O(m)$ time and space.*

Proof. In order to compute an optimal 2-circle of G, we first compute G' in $O(m)$ time, e.g. by Ref. [15] although for our special case a much simpler algorithm is sufficient. Clearly G' is convex and has $O(m)$ edges. Thus we can compute an optimal 1-circle C in G' in $O(m)$ time according to Lemma 5. Finally Lemma 6 assures that C is also an optimal 2-circle in G. □

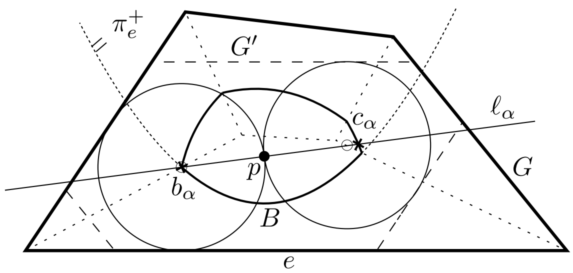

Fig. 8. Computing the boundary B of $\bigcap_e \pi_e^+$.

We have another algorithm for labeling a point with optimal 2-circles in a convex polygon that does not use the medial axis. It is easier to implement but it is harder to see its linear complexity, that is why we only sketch it here. We mention it since it gives a new perspective of the problem. For an edge e of G let π_e^+ be the part of the plane that contains p and is bounded by the parabola π_e with focus p and directrix e, see Figure 8. Then the boundary B of the intersection of all π_e^+ is the trace of the center of a circle that is rotated around p while touching p and being of maximum size within G. The points on B that are furthest from p are thus the centers of optimal 1-circles. Let ℓ_α be a line that goes through p and forms an angle of α with the horizontal. This line intersects B in two points b_α and c_α. Angles α that maximize $r_\alpha = \min\{|pb_\alpha|, |pc_\alpha|\}$ correspond to optimal 2-circles of radius r_α whose centers lie on ℓ_α.

It is not hard to see that given B an optimal 2-circle can be found in linear time. One can compute B incrementally using the given cyclic order of the edges $e_1, \ldots, e_m$ around G. Let $\pi_i = \pi_{e_i}$ and $\Pi_i = \pi_1^+ \cap \ldots \cap \pi_i^+$ and let π_j be the last parabola with an arc A_j on the boundary of Π_i. Observe that two parabolas intersect exactly twice, namely in the centers of the two unique circles that touch p and the directrices of the parabolas. Is not hard to see that each parabola contributes at most one arc to Π_i. When intersecting Π_i with π_{i+1}^+ it is enough to analyze the behavior of the two arcs A_j and $A = \pi_j \cap \pi_{i+1}^+$. From each of the four cases (a) $A \subset A_j$, (b) $A_j \subset A$, (c) $A \cap A_j = \emptyset$, and (d) $A \cap A_j \notin \{\emptyset, A, A_j\}$ it is immediately clear whether or not π_j and π_{i+1} will contribute to Π_{i+1}. This is the key to the linear runtime for computing B.

After this sketch we state our main theorem.

Theorem 1 *A set P of n points in the plane can be labeled with $2n$ non-intersecting circular labels, two per point, of diameter $2/3$ times the optimal label size in $O(n \log n)$ time using linear space.*

Proof. Our algorithm is as follows. First we compute the Voronoi diagram of P. This takes $O(n \log n)$ time and linear space.[3] Then for each input point p we use the algorithm described in the proof of Lemma 7 to compute the largest pair of circles that labels p within the Voronoi cell $\mathrm{Vor}(p)$ of p. Let d_p be the diameter of these circles. Since the complexity of the Voronoi diagram is linear[3] in n, labeling all points takes $O(n)$ time and space in total. Next we set $d_{\mathrm{algo}} = \min_{p \in P} d_p$ and go through all input points once more. We scale the labels of each point p by a factor of d_{algo}/d_p using p as the scaling center. We output these scaled labels, all of which now have diameter d_{algo}. Clearly this algorithm runs in $O(n \log n)$ time and uses linear space.

Its correctness can be seen as follows. Since $d_{\mathrm{algo}}/d_p \leq 1$, the scaled labels lie completely within the original labels. Each of these lies in its label zone which in turn lies in the corresponding Voronoi cell according to Lemma 2. Thus no two of the scaled labels intersect. It remains to show that $d_{\mathrm{algo}} \geq d_{\mathrm{lower}}$.

Lemma 1 guarantees that for each input point p there is an orientation (namely that determined by the centers of the labels of p in $\mathcal{C}_{\mathrm{opt}}$) such that the label zone $Z_{\mathrm{label}}(p)$ lies completely within $\mathrm{Vor}(p)$. We do not know this orientation, but Lemma 3 asserts that there are two disjoint labels for p, both of diameter d_{lower}, that lie within $Z_{\mathrm{label}}(p)$—and thus within $\mathrm{Vor}(p)$. This is true even for a point q that receives the smallest labels before scaling, i.e. $d_q = d_{\mathrm{algo}}$. On the other hand we have $d_q \geq d_{\mathrm{lower}}$ since d_q is the maximum label diameter for a pair of labels for q that lies completely within $\mathrm{Vor}(q)$. □

5. Related Labeling Problems

We hoped that the "Voronoi method" could also be applied to other label-size maximization problems. However, we can only state a few negative results here. It is the two-circle point-labeling problem that seems to go best with the Voronoi diagram. Figures 9 to 11 give the counter examples. In each figure the labels of an optimal labeling have solid outline, a possible outcome of the Voronoi method has shaded labels, and the edges of the Voronoi diagram are marked by dotted lines.

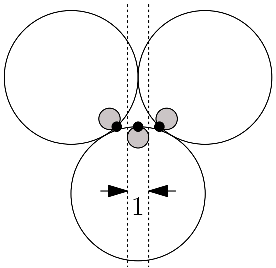

Fig. 9. one-circle labeling

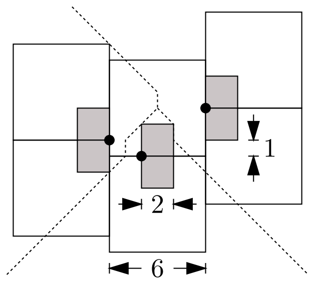

Fig. 10. two-square labeling

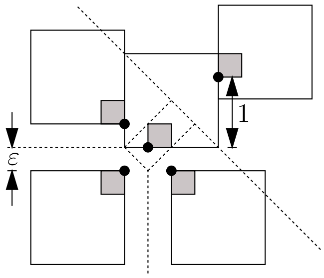

Fig. 11. one-square labeling

First, consider the one-circle point-labeling problem.[8] With the Voronoi method the points $(-1,0)$, $(0,0)$, and $(0,1)$ can only be labeled with labels of diameter 1, since the Voronoi cell of the middle point is a strip of that width, see Figure 9. However, some basic geometry[19] shows that the optimal labels for these three points have diameter $2/(1-\sqrt{2\sqrt{3}-3}) \approx 6.29$. Thus our method would yield an approximation factor of at most 1/6.29 which is considerably worse than that of the currently best approximation algorithm,[8] namely 1/3.6.

Second, we investigate two-square point labeling.[16] For square labels it is not the Euclidean (or L_2-) metric, but the maximum (or L_∞-) metric that helps to bound the maximum label size.[24] Using the Voronoi diagram induced by the maximum norm, the three points $(0,1)$, $(2,0)$, and $(6,3)$ in Figure 10 can be labeled with labels of edge length at most 2, while the maximum label size (i.e. edge length) for three points always equals their L_∞-diameter.[24] This is 6 in our case, and thus our method would yield at most a 1/3-approximation, which is less than the currently best 1/2-approximation.[16] Using the Voronoi diagram induced by the Manhattan (i.e. L_1-) metric would result in labels of the same size, while they would be even slightly smaller in the Euclidean case.

Third, we take a look at one-square point labeling. The currently best known algorithm[17] for this problem has an approximation factor of 1/2. Consider the five points $(0,0)$, $(1,1)$, $(-\varepsilon,+\varepsilon)$, $(-\varepsilon,-\varepsilon)$, and $(+\varepsilon,-\varepsilon)$ for some $\varepsilon > 0$, see Figure 11. The Voronoi cell of $(0,0)$ is a rectangle of height $\sqrt{2}\varepsilon$ whose edges form an angle of 45° with the coordinate axes. While the maximum label size for the five points is $1+\varepsilon$, the largest label that fits into the Voronoi cell of the origin has an edge length of only ε. This is true for the Voronoi diagram induced by any of the L_1-, L_2- and L_∞-metrics, the last of which is depicted in Figure 11. Thus for one-square labeling the Voronoi method does not give any approximation at all.

Acknowledgments

We are extremely indebted to Tycho Strijk, Utrecht University, for pointing out a mistake in an earlier version of the proof of the approximation factor of our algorithm. We also wish to thank Thomas Erlebach, ETH Zürich, for presenting our result at ISAAC'00.

References

1. Alok Aggarwal, Leonida J. Guibas, James Saxe, and Peter W. Shor. A linear-time algorithm for computing the Voronoi diagram of a convex polygon. *Discrete and Computational Geometry*, 4(6):591–604, 1989.
2. John Ahn and Herbert Freeman. AUTONAP—an expert system for automatic map name placement. In *Proc. Int. Symp. Spatial Data Handling (SDH'84)*, pages 544–569, 1984.
3. Franz Aurenhammer. Voronoi diagrams: A survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, September 1991.
4. Prosenjit Bose, Jurek Czyzowicz, Evangelos Kranakis, and Anil Maheshwari. Algorithms for packing two circles in a convex polygon. In Jin Akiyama, Mikio Kano, and Masatsugu Urabe, editors, *Proceedings of the Japanese Conference on Discrete and Computational Geometry (JCDCG'98)*, volume 1763 of *Lecture Notes in Computer Science*, pages 93–103, Tokyo, 9–12 December 1998. Springer-Verlag.
5. Bernard Chazelle and 36 co-authors. The computational geometry impact task force report. In B. Chazelle, J. E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, volume 223, pages 407–463. American Mathematical Society, Providence, 1999.
6. Jon Christensen, Joe Marks, and Stuart Shieber. An empirical study of algorithms for point-feature label placement. *ACM Transactions on Graphics*, 14(3):203–232, 1995.
7. Srinivas Doddi, Madhav V. Marathe, Andy Mirzaian, Bernard M.E. Moret, and Binhai Zhu. Map labeling and its generalizations. In *Proceedings of the 8th ACM-SIAM Symposium on Discrete Algorithms (SODA'97)*, pages 148–157, New Orleans, LA, 4–7 January 1997.
8. Srinivas Doddi, Madhav V. Marathe, and Bernard M.E. Moret. Point set labeling with specified positions. *International Journal of Computational Geometry and Applications*, 12(1–2):29–66, 2002.
9. Jeffrey S. Doerschler and Herbert Freeman. An expert system for dense-map name placement. In *Proc. Auto-Carto 9*, pages 215–224, 1989.
10. Michael Formann and Frank Wagner. A packing problem with applications to lettering of maps. In *Proc. 7th Annu. ACM Sympos. Comput. Geom. (SoCG'91)*, pages 281–288, 1991.
11. Stephen A. Hirsch. An algorithm for automatic name placement around point data. *The American Cartographer*, 9(1):5–17, 1982.
12. Konstantinos G. Kakoulis and Ioannis G. Tollis. On the multiple label placement problem. In *Proc. 10th Canadian Conf. Computational Geometry (CCCG'98)*, pages 66–67, 1998.
13. Sung Kwon Kim, Chan-Su Shin, and Tae-Cheon Yang. Labeling a rectilinear map with sliding labels. *International Journal of Computational Geometry and Applications*, 11(2):167–179, April 2001.

14. Joel L. Morrison. Computer technology and cartographic change. In D.R.F. Taylor, editor, *The Computer in Contemporary Cartography*. J. Hopkins Univ. Press, New York, 1980.

15. Joseph O'Rourke, Chi Bin Chien, Thomas Olson, and David Naddor. New linear algorithm for intersecting convex polygons. *Computer Graphics and Image Processing*, 19(4):384–391, August 1982.

16. Zhongping Qin, Alexander Wolff, Yinfeng Xu, and Binhai Zhu. New algorithms for two-label point labeling. In Mike Paterson, editor, *Proc. 8th Annu. Europ. Symp. on Algorithms (ESA'00)*, volume 1879 of *Lecture Notes in Computer Science*, pages 368–379, Saarbrücken, 5–8 September 2000. Springer-Verlag.

17. Zhongping Qin and Binhai Zhu. A factor-2 approximation for labeling points with maximum sliding labels. In Martti Penttonen and Erik Meineche Schmidt, editors, *Proc. 8th Scandinavian Workshop on Algorithm Theory (SWAT'02)*, volume 2368 of *Lecture Notes in Computer Science*, pages 100–109, Turku, 3–5 July 2002. Springer-Verlag.

18. Michael J. Spriggs and J. Mark Keil. A new bound for map labeling with uniform circle pairs. *Information Processing Letters*, 81(1):47–53, 2002.

19. Tycho Strijk and Alexander Wolff. Labeling points with circles. *International Journal of Computational Geometry and Applications*, 11(2):181–195, April 2001.

20. Marc van Kreveld, Tycho Strijk, and Alexander Wolff. Point labeling with sliding labels. *Computational Geometry: Theory and Applications*, 13:21–47, 1999.

21. Frank Wagner and Alexander Wolff. A practical map labeling algorithm. *Computational Geometry: Theory and Applications*, 7:387–404, 1997.

22. Alexander Wolff and Tycho Strijk. The Map-Labeling Bibliography. `www.math-inf.uni-greifswald.de/map-labeling/bibliography`, 1996.

23. Alexander Wolff, Michael Thon, and Yinfeng Xu. A better lower bound for two-circle point labeling. In D.T. Lee, editor, *Proc. 11th Annual International Symposium on Algorithms and Computation (ISAAC'00)*, volume 1969 of *Lecture Notes in Computer Science*, pages 422–431, Taipei, 18–20 December 2000. Institute of Information Science, Academia Sinica, Springer-Verlag.

24. Binhai Zhu and Chung Keung Poon. Efficient approximation algorithms for two-label point labeling. *International Journal of Computational Geometry and Applications*, 11(4):455–464, August 2001.

25. Binhai Zhu and Zhongping Qin. New approximation algorithms for map labeling with sliding labels. *Journal of Combinatorial Optimization*, 6(1):99–110, 2002.

26. Steven Zoraster. The solution of large 0-1 integer programming problems encountered in automated cartography. *Operations Research*, 38(5):752–759, 1990.