SIAM J. COMPUT.
Vol. 22, No. 4, pp. 778–793, August 1993

COUNTING CIRCULAR ARC INTERSECTIONS*

PANKAJ K. AGARWAL,† MARCO PELLEGRINI,‡ AND MICHA SHARIR§

Abstract. In this paper efficient algorithms for counting intersections in a collection of circles or circular arcs are presented. An algorithm for counting intersections in a collection of n circles is presented whose running time is $O(n^{3/2+\epsilon})$, for any $\epsilon > 0$ is presented. Using this algorithm as a subroutine, it is shown that the intersections in a set of n circular arcs can also be counted in time $O(n^{3/2+\epsilon})$. If all arcs have the same radius, the running time can be improved to $O(n^{4/3+\epsilon})$, for any $\epsilon > 0$.

Key words. arrangements, point location, random sampling, partition tree

AMS subject classifications. 68P05, 68Q25, 68Q40, 68U05

1. Introduction. Intersection problems are among the fundamental topics in computational geometry. In 1979, Bentley and Ottmann, in their famous paper on the line sweep technique, showed that all K intersections in a collection of n Jordan arcs in $\mathbb{R}^2$ can be reported in time $O((n + K) \log n)$ [BO] (under reasonable assumptions concerning the shape of these arcs and on the model of computation). Their algorithm is significantly faster than the naive quadratic algorithm for small values of K, but is worse than the naive approach if $K = \Theta(n^2)$. Since then much effort has been invested to remove the $\log n$ factor from K. Recently Chazelle and Edelsbrunner [CE] presented an $O(n \log n + K)$ time algorithm to report all intersections in a collection of line segments. However, for general arcs, the Bentley–Ottmann algorithm is still the best known deterministic algorithm. If we allow randomization, algorithms with improved running time of $O(n \log n + K)$ can be obtained (see Clarkson and Shor [CS2] and Mulmuley [Mu]).

All of the above algorithms are quite efficient if we want to *report* the intersections explicitly. In some applications, however, we are only interested in *counting* the total number of intersections (not in finding the actual intersection points). In that case we prefer an algorithm whose running time does not depend on K, because K can be as large as $\Theta(n^2)$. Ideally we would like to have an algorithm that counts the number of intersections in time $O(n \log n)$, but developing such an algorithm seems to be quite hard. For line segments, Chazelle [Cha] proposed an $O(n^{3/2} \log n)$ time algorithm. Later Guibas, Overmars, and Sharir [GOS] gave a randomized algorithm whose expected running time was $O(n^{4/3+\epsilon})$, for any $\epsilon > 0$. Their algorithm can be made deterministic and somewhat improved using recent partitioning algorithms of Matoušek [Mat1] and Agarwal [Ag1]. The best known running time, at present, for counting segment intersections is $O(n^{4/3} \log^{1/3} n)$ (see [Ag2], [Chb]), and the corresponding algorithms are deterministic.

The "bichromatic" version of the segment intersection counting problem has also been studied. Here we are given a set of "red" segments and another set of "blue" segments, and we want to count the number of red-blue intersections. This problem can also be solved in time

*Received by the editors January 28, 1991; accepted for publication (in revised form) May 20, 1992. Part of this work was done while the first author was visiting the Center for Discrete Machematics and Theoretical Computer Science (DIMACS), Rutgers University. Work by the first author has been supported by DIMACS, a National Science Foundation Science and Technology Center, under grant NSF-STC88-09648. Work by the third author has been supported by Office of Naval Research grant N00014–90–J–1284, National Science Foundation grant CCR-89-01484, by grants from the U.S.–Israeli Binational Science Foundation, the Fund for Basic Research administered by the Israeli Academy of Sciences, and the German–Israeli Foundation for Scientific Research and Development.

†Computer Science Department, Duke University, Durham, North Carolina 27706.

‡Computer Science Department, King's College, London, United Kingdom.

§School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel and Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York, New York 10012.

$O(n^{4/3}\log^{1/3} n)$ using a variant of the Guibas, Overmars, and Sharir algorithm (see [Ag2]). For the special case, where no two red segments and no two blue segments intersect, an optimal $O(n \log n)$ algorithm has been developed by Chazelle et al. [CEGS2].

The intersection counting problem seems to be much harder for general arcs, because unlike segments, two arcs may intersect at more than one point, which makes their intersection patterns more involved than those of segments. Using Chazelle and Sharir's generalized point location technique in semi-algebraic varieties [CS1], the arc intersection problem can be solved in slightly subquadratic time (roughly $O(n^{1.98})$). However, we are not aware of any substantially subquadratic algorithm for the general case. The recent algorithm of Agarwal et al. [AASS] can be applied to obtain an $O(n^{4/3}\log^{2/3} n)$ time algorithm for counting intersections in a set of circles with the same radius, but it does not extend to circles of arbitrary radii.

In this paper we consider the intersection-counting problem for a collection of arbitrary circles and of circular arcs. Our main results are:

(i) An algorithm for counting intersections in a collection of n arbitrary circles whose running time is $O(n^{3/2+\epsilon})$, for any $\epsilon > 0$.[1]

(ii) An algorithm for counting intersections in a collection of n circular arcs, whose running time is $O(n^{3/2+\epsilon})$.

(iii) If all arcs are of circles of the same radius, the running time of both algorithms can be improved to $O(n^{4/3+\epsilon})$.

The remainder of the paper is organized as follows. In §2 we present our circle intersection algorithm. Section 3 deals with counting intersections between a set of circles and a set of circular arcs. In §4 we describe our main algorithm, that is, counting intersections in a set of circular arcs. We conclude with some open problems in §5.

2. Counting circle intersections. In this section we present a divide-and-conquer algorithm for the following problem: "Given a collection $\mathcal{C} = \{C_1, \ldots, C_n\}$ of n circles, count the number of pairs of intersecting circles in $\mathcal{C}$." Let $\mathcal{C}_1 = \{C_1, \ldots, C_{\lceil n/2\rceil}\}$ and $\mathcal{C}_2 = \{C_{\lceil n/2\rceil+1}, \ldots, C_n\}$. We first count the number of pairs of intersecting circles in $\mathcal{C}_1$ and in $\mathcal{C}_2$ recursively, and then we count the number of intersecting pairs $(C_i, C_j) \in \mathcal{C}_1 \times \mathcal{C}_2$. Thus, it is sufficient to solve the "bichromatic version" of the above problem, that is,

Given a collection $\mathcal{C}$ of m "red" circles and another collection $\mathcal{C}'$ of n "blue" circles in the plane, count $\mathcal{I}(\mathcal{C}, \mathcal{C}')$, the number of intersecting "red-blue" pairs of circles.

Note that, assuming nondegenerate configurations, the number of red-blue intersection points is twice that count.

LEMMA 2.1. *Given a pair of circles C, C' with centers p, p' and radii ρ, ρ', respectively, C intersects C' if and only if*

$$|\rho - \rho'| \le d(p, p') \le \rho + \rho'. \tag{1}$$

Proof. The proof follows from elementary geometry. □

To compute $\mathcal{I}(\mathcal{C}, \mathcal{C}')$, we define two geometric transforms φ and Ψ. Let C be a circle with center (a, b) and radius r. The first transform, φ, maps C to the point

$$\varphi(C) = (a, b, r) \tag{2}$$

[1]Throughout this paper, ϵ denotes an arbitrarily small positive constant. The meaning of such a bound is that for any $\epsilon > 0$ the algorithm can be fine-tuned so that its running time is within that bound, where the constant of proportionality depends on ϵ and usually tends to ∞ as $\epsilon \downarrow 0$.

in $\mathbb{R}^3$, and the second transform, Ψ, maps C to the region

$$\Psi(C) = \{ (x, y, z) \mid z \geq 0 \text{ and } (z+r)^2 \geq (x-a)^2 + (y-b)^2 \geq (z-r)^2 \} \tag{3}$$

in $\mathbb{R}^3$. The boundary of $\Psi(C)$ consists of two surfaces:

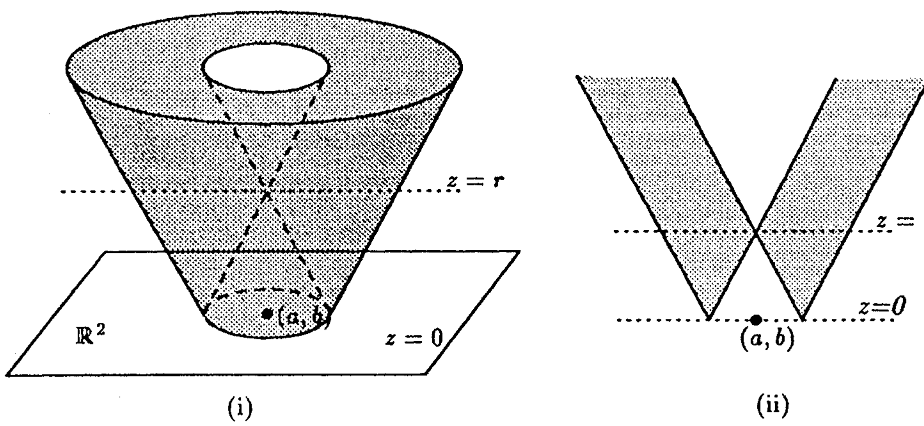

FIG. 1. (i) *Shaded region denotes* $\Psi(C)$; (ii) *cross-section of* $\Psi(C)$ *at a vertical plane through* (a, b).

(i) *The outer surface*, denoted $\psi_O(C)$, is the truncated cone

$$(z+r)^2 = (x-a)^2 + (y-b)^2, \qquad z \geq 0.$$

(ii) *The inner surface*, denoted $\psi_I(C)$, is the truncated double cone

$$(z-r)^2 = (x-a)^2 + (y-b)^2, \qquad z \geq 0.$$

A point $p = (a, b)$ can be considered a circle of radius zero, so $\psi_I(p) = \psi_O(p)$ is the cone $z^2 = (x-a)^2 + (y-b)^2$. An immediate consequence of the previous lemma is the following lemma.

LEMMA 2.2. *A circle C intersects another circle C' if and only if $\varphi(C) \in \Psi(C')$.*

It therefore suffices to count for each circle $C \in \mathcal{C}$ the number of circles $C' \in \mathcal{C}'$ for which $\varphi(C) \in \Psi(C')$. We describe two algorithms for counting these quantities. The first algorithm is quite simple and works well when $m \geq n^3$, while the second algorithm, though more complicated, is efficient for all ranges of m and n.

2.1. A simple algorithm. Our first algorithm for computing $\mathcal{I}(\mathcal{C}, \mathcal{C}')$ works as follows. Let

$$\mathcal{H} = \{ \psi_O(C') \mid C' \in \mathcal{C}' \} \cup \{ \psi_I(C') \mid C' \in \mathcal{C}' \}. \tag{4}$$

Fix a surface $H \in \mathcal{H}$. Intersect it with all other surfaces of $\mathcal{H}$. It is easily checked that each intersection curve is the intersection of H with a plane, and is thus a conic planar curve. Moreover, each pair of such curves intersect each other in at most two points (these points are the points of intersection of H with the line which is the intersection of the two corresponding planes; since H is a quadric, there are at most two such points).

Let Γ denote the resulting collection of curves on H. We next form the two-dimensional arrangement $\mathcal{A}(\Gamma)$ of these curves on H. This can be done deterministically, e.g., in time $O(n^2 \log n)$ using a sweeping technique, similar to that of Bentley and Ottmann [BO]. For each circle $C' \in \mathcal{C}'$ and each face $f \in \mathcal{A}(\Gamma)$, f lies either fully inside or fully outside $\Psi(C')$. Consequently, after $\mathcal{A}(\Gamma)$ is computed, we can calculate, for each face f of this arrangement, the number τ_f of regions $\Psi(C')$ that contain it. Since this number changes only by one as we cross from one face to an adjacent one, a simple traversal of the arrangement will produce all the quantities τ_f in time $O(n^2)$. We repeat this step for each surface $H \in \mathcal{H}$ in overall time $O(n^3 \log n)$.

Next, preprocess $\mathcal{A}(\mathcal{H})$ for spatial point location in time $O(n^{3+\epsilon})$ using the algorithm of Chazelle et al. [CEGS1].[2] The processing is done so that the output to a query is the surface that lies directly below it. The time to answer a query is $O(\log n)$. Note that the method of [CEGS1] does not require an explicit construction of the arrangement of $\mathcal{H}$, especially if the output to queries is to have this restricted form.

We now take each circle $C \in \mathcal{C}$ and locate its image $\varphi(C)$ in $\mathcal{A}(\mathcal{H})$, obtaining the surface H lying directly below $\varphi(C)$ and the vertical projection $\omega(C)$ of $\varphi(C)$ on that surface. We now locate the face f of the corresponding arrangement on H that contains $\omega(C)$, and add either τ_f or $\tau_f + 1$ to a global count, depending on whether the region $\Psi(C')$ bounded by H lies below H or above it near the face f. The sum of these quantities, over all $C \in \mathcal{C}$, is the desired $\mathcal{I}(\mathcal{C}, \mathcal{C}')$. The time required by this step is $O(m \log n)$, so the total running time of the algorithm is $O(n^{3+\epsilon} + m \log n)$.

The running time can be improved using a standard "batching" technique; that is, partition $\mathcal{C}'$ into $t = \lceil \frac{n}{m^{1/3}} \rceil$ sets $\mathcal{C}'_1, \ldots, \mathcal{C}'_t$, each containing at most $m^{1/3}$ circles. For each i, we compute $\mathcal{I}(\mathcal{C}, \mathcal{C}'_i)$ separately, and then add these quantities to determine $\mathcal{I}(\mathcal{C}, \mathcal{C}')$. The overall running time is therefore

$$\left\lceil \frac{n}{m^{1/3}} \right\rceil \times O(m^{1+\epsilon}) = O(m^{2/3+\epsilon} n + m^{1+\epsilon}).$$

Hence, we obtain the following result.

THEOREM 2.3. *Given a collection $\mathcal{C}$ of m "red" circles and a collection $\mathcal{C}'$ of n "blue" circles, the number of intersecting red-blue pairs can be counted in time $O(m^{2/3+\epsilon} n + m^{1+\epsilon})$, for any $\epsilon > 0$.*

Remark 2.4.

1. By flipping the role of red and blue circles, we can obtain another algorithm whose running time is $O(mn^{2/3+\epsilon} + n^{1+\epsilon})$.

2. Note that if $n < m^{1/3}$ (or $m < n^{1/3}$), then the above algorithm runs in time $O(m^{1+\epsilon})$ (respectively, $O(n^{1+\epsilon})$), which is almost optimal.

COROLLARY 2.5. *Given a collection of n circles, the number of intersecting pairs of circles can be counted in time $O(n^{5/3+\epsilon})$, for any $\epsilon > 0$.*

2.2. An improved algorithm. We now present an algorithm for computing $\mathcal{I}(\mathcal{C}, \mathcal{C}')$, which is significantly faster than the previous one, especially when m and n are of the same order of magnitude.

If $n \geq m^3$, $\mathcal{I}(\mathcal{C}, \mathcal{C}')$ is computed using the previous algorithm, in time $O(n^{1+\epsilon})$, so assume that $n < m^3$. Let $\mathcal{H}$ be the set of surfaces defined in (4). Let r be some sufficiently large constant. We compute a $\frac{1}{2r}$-net $\mathcal{R} \subseteq \Gamma$, of size $O(r \log r)$, in time $O(n)$ [Mat2].[3] We decompose $\mathcal{A}(\mathcal{R})$ into $M = O(r^3 \beta(r) \log^3 r)$ simple cells (of constant complexity), using the algorithm of Chazelle et al. [CEGS1] (see also [CEGSW, §6]); $\beta(r)$ is an extremely slow

[2] Although the algorithm described in the original paper is randomized, it can be made deterministic without affecting its asymptotic performance, using a recent result of Matousek for deterministic construction of ϵ-nets [Mat2].

[3] Specializing from the general concept, we call a subset $\mathcal{R} \subseteq \Gamma$ of a set of n (algebraic) surfaces a $\frac{1}{r}$-net, $r < n$, if every (open) cell of constant complexity, of the form obtained in the stratification algorithm of [CEGS1], which does not intersect any surface of $\mathcal{R}$, intersects at most n/r surfaces of Γ; see [HW] for a more formal definition. Haussler and Welzl [HW] showed that a random subset of Γ of size $O(r \log r)$ is a $\frac{1}{r}$-net with high probability. Later Matousek [Mat2] gave an $O(nr^{O(1)})$-time deterministic algorithm for computing a $\frac{1}{r}$-net of size $O(r \log r)$.

growing function, which depends on the inverse Ackermann function $\alpha(r)$. Let $\mathcal{A}^*(\mathcal{R})$ denote the resulting subdivision. Since $\mathcal{A}^*(\mathcal{R})$ is a $\frac{1}{2r}$-net of $\mathcal{H}$, each cell $\tau \in \mathcal{A}^*(\mathcal{R})$ of dimension ≥ 1 intersects at most $2n/2r = n/r$ surfaces of $\mathcal{H}$.

Our approach is to partition the points $\varphi(C)$ among the cells $\tau \in \mathcal{A}^*(\mathcal{R})$ and to distribute the surfaces ψ_I, ψ_O among those cells that they intersect. This gives us a collection of subproblems, one for each cell τ, whose combined solutions give us the desired count.

In more detail, we regard $\mathcal{A}^*(\mathcal{R})$ as a collection of pairwise disjoint relatively open cells of dimensions 3, 2, 1, or 0. For each cell $\tau \in \mathcal{A}^*(\mathcal{R})$, we define $\mathcal{C}'_\tau$ as the set of circles C' for which $\psi_I(C')$ or $\psi_O(C')$ intersects τ. Similarly, we define $\mathcal{C}_\tau$ as the set of circles C for which $\varphi(C)$ lies in τ. Let $m_\tau = |\mathcal{C}_\tau|$, $n_\tau = |\mathcal{C}'_\tau|$, and let λ_τ be the number of circles C' for which $\tau \subseteq \Psi(C')$. Every τ of dimension ≥ 1 intersects at most n/r surfaces of $\mathcal{H}$, thus $n_\tau \leq \frac{n}{r}$. It is easy to see that

$$\mathcal{I}(\mathcal{C}, \mathcal{C}') = \sum_{\tau \in \mathcal{A}^*(\mathcal{R})} \mathcal{I}(\mathcal{C}_\tau, \mathcal{C}'_\tau) + m_\tau \lambda_\tau .$$

Since each cell τ has constant complexity, $\mathcal{C}_\tau$, $\mathcal{C}'_\tau$ and λ_τ can be calculated in $O(m+n)$ time per cell, so the total time required to compute these quantities is $O((m+n)r^3\beta(r)\log^3 r)$ $= O(m+n)$, as r is constant. Next, the quantities $\mathcal{I}(\mathcal{C}_\tau, \mathcal{C}'_\tau)$ are computed as follows. If τ is a three-dimensional cell, we compute $\mathcal{I}(\mathcal{C}_\tau, \mathcal{C}'_\tau)$ recursively. If τ is a two-dimensional cell, we also proceed recursively, except that we have a two-dimensional problem at hand. This can still be solved using an analogous two-dimensional partitioning scheme. If τ is an edge of $\mathcal{A}^*(\mathcal{R})$, we can compute $\mathcal{I}(\mathcal{C}_\tau, \mathcal{C}'_\tau)$, in time $O((m_\tau + n_\tau)\log n_\tau)$, by computing the intersection points of τ and $\partial\Psi(C')$, for $C' \in \mathcal{C}'_\tau$, sorting them along τ, and then locating the points $\varphi(C)$ along τ, for $C \in \mathcal{C}_\tau$. Finally, if τ is a vertex, then we have $\mathcal{I}(\mathcal{C}_\tau, \mathcal{C}'_\tau) = m_\tau \cdot n_\tau$ (note that in this case $m_\tau = 0$ or 1). The maximum total running time $T(m,n)$ to compute $\mathcal{I}(\mathcal{C}, \mathcal{C}')$ therefore satisfies:

$$(5) \qquad T(m,n) \leq \begin{cases} a_1(m+n)\log n + \displaystyle\sum_{\substack{\tau \in \mathcal{A}^*(\mathcal{R}) \\ \dim \tau \geq 2}} T(m_\tau, n_\tau) & \text{if} \quad n < m^3 \\ a_2 n^{1+\epsilon'} & \text{if} \quad n \geq m^3 , \end{cases}$$

where a_1, a_2 are some constants (a_2 depending on ϵ'), $\sum_\tau m_\tau \leq m$, and $n_\tau \leq \frac{n}{r}$ for every τ. It is easily checked that these equations also hold when we recurse on a two-dimensional problem. In this case the second equation holds already when $n \geq m^2$.

LEMMA 2.6. *For every $\epsilon > 0$, there exist constants $A, B, D > 0$ depending on ϵ such that*

$$(6) \qquad T(m,n) \leq Am^{3/4+\epsilon}n^{3/4} + Bn^{1+\epsilon/3} + Dm\log^2 n.$$

Proof. For $n \geq m^3$, (6) is obviously true provided we choose $B \geq a_2$ and $\epsilon \geq 3\epsilon'$. For $n < m^3$, we prove the lemma by induction on n. If A and B are chosen appropriately, (6) holds trivially for small values of n. Assume that the claim is true for all $n' < n$. If $r > 2$, then by inductive hypothesis, (5) can be written as follows (here we bound the number of cells in $\mathcal{A}^*(\mathcal{R})$ by $c_1 r^3\beta(r)\log^3 r$ for an appropriate constant $c_1 > 0$):

$$T(m,n) \leq \sum_{\substack{\tau \in \mathcal{A}^*(\mathcal{R}) \\ \dim \tau \geq 2}} \left(Am_\tau^{3/4+\epsilon}n_\tau^{3/4} + Bn_\tau^{1+\epsilon/3} + Dm_\tau \log^2 n_\tau \right) + a_1(m+n)\log n$$

$$
\begin{aligned}
&\le A\left(\frac{n}{r}\right)^{3/4}\cdot\left(\sum_{\substack{\tau\in\mathcal{A}^*(\mathcal{R})\\ \dim\tau\ge 2}} m_\tau^{3/4+\epsilon}\right) + Bc_1r^3\beta(r)\log^3 r\left(\frac{n}{r}\right)^{1+\epsilon/3}\\
&\quad + a_1(m+n)\log n + D\log n\log\frac{n}{r}\cdot\sum_{\substack{\tau\in\mathcal{A}^*(\mathcal{R})\\ \dim\tau\ge 2}} m_\tau\\
&\le Am^{3/4+\epsilon}n^{3/4}\left(\frac{1}{r}\right)^{3/4}\left(c_1r^3\beta(r)\log^3 r\right)^{1/4-\epsilon} + Bn^{1+\epsilon/3} + Dm\log^2 n\\
&\quad + Bn^{1+\epsilon/3}\left(c_1r^{2-\epsilon/3}\beta(r)\log^3 r - 1 + \frac{a_1}{B}\frac{\log n}{n^{\epsilon/3}}\right) + (a_1 - D\log r)m\log n
\end{aligned}
$$

where the first term of the last inequality follows from Hölder's inequality. If $D > a_1$, then $a_1 - D\log r < 0$ (as $r > 2$). Since $n < m^3$, we have $n^{1+\epsilon/3} \le m^{3/4+\epsilon}n^{3/4}$. We therefore obtain

$$
\begin{aligned}
T(m,n) \quad \le \quad & Am^{3/4+\epsilon}n^{3/4}\left(\frac{(c_1\beta(r)\log^3 r)^{1/4-\epsilon}}{r^{3\epsilon}}\right) + Bn^{1+\epsilon/3} + Dm\log^2 n\\
& + Em^{3/4+\epsilon}n^{3/4},
\end{aligned}
$$

where

$$
E = B\left(c_1r^{2-\epsilon/3}\beta(r)\log^3 r - 1 + \frac{a_1}{B}\frac{\log n}{n^{\epsilon}}\right).
$$

If we choose r and A sufficiently large so that

$$
\frac{(c_1\beta(r)\log^3 r)^{1/4-\epsilon}}{r^{3\epsilon}} + \frac{E}{A} \le 1,
$$

the running time becomes

$$
T(m,n) \le Am^{3/4+\epsilon}n^{3/4} + Bn^{1+\epsilon/3} + Dm\log^2 n. \qquad \square
$$

THEOREM 2.7. *Given a collection $\mathcal{C}$ of m "red" circles and a collection $\mathcal{C}'$ of n "blue" circles, the number of intersecting red-blue pairs of circles can be counted in time $O(m^{3/4+\epsilon}n^{3/4} + n^{1+\epsilon} + m\log^2 n)$, for any $\epsilon > 0$.*

Returning to the original problem of counting circle intersections, the above theorem implies that the merge step of our divide-and-conquer algorithm can be performed in time $O(n^{3/2+\epsilon})$. Hence, we obtain the main result of this section.

THEOREM 2.8. *Given a collection of n circles in the plane, the number of intersecting pairs can be counted in time $O(n^{3/2+\epsilon})$, for any $\epsilon > 0$.*

3. Counting intersections between circles and arcs. In this section, we study the problem of counting the number of intersection points between a collection $\mathcal{C}$ of n circles and another collection Γ of m circular arcs. For the sake of simplicity we assume that all intersections are transversal and do not lie at the endpoints of the arcs.

Let α, β denote the endpoints of an arc γ, and let c denote the center of the circle containing γ. The lines supporting the segments αc and βc partition the plane into four wedges (quadrants), and γ clearly lies completely in one of the wedges; we will denote this wedge by $\omega(\gamma)$, and denote by $\bar{\omega}(\gamma)$ the wedge lying opposite to $\omega(\gamma)$; see Fig. 3 for an

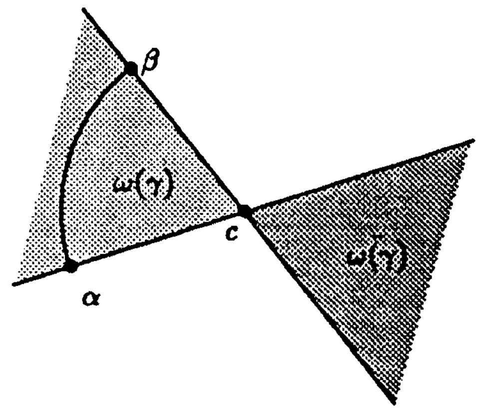

FIG. 2. *Wedges $\omega(\gamma)$ and $\bar{\omega}(\gamma)$.*

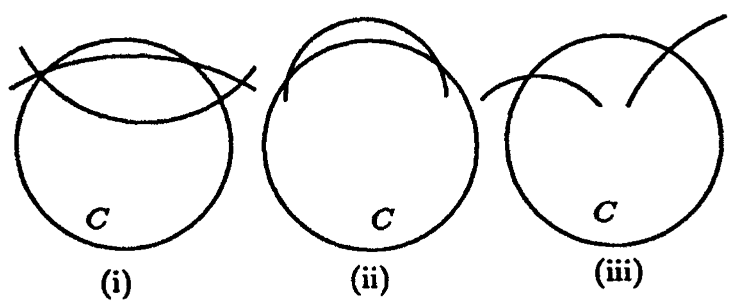

FIG. 3. *Various circle-arc intersection patterns.*

example. For a set $\mathcal{C}$ of circles, let $E(\mathcal{C})$ (respectively, $I(\mathcal{C})$) denote the common exterior (respectively, interior) of circles in $\mathcal{C}$.

LEMMA 3.1. *A circular arc γ intersects a circle C at one point if and only if one of the endpoints of γ lies in the interior of C and the other endpoint lies in the exterior of C.*

Proof. The proof follows immediately from the Jordan curve theorem and the fact that two circles intersect in at most two points. □

LEMMA 3.2. *A circular arc γ, which does not span more than a semicircle, intersects C at two points if and only if C intersects the circle containing γ, and one of the following two conditions is satisfied:*

(i) *the endpoints of γ lie outside C and the center of C lies in $\omega(\gamma)$ (see Fig.* 3(i)) *or*

(ii) *the endpoints of γ lie inside C and the center of C lies in $\bar{\omega}(\gamma)$ (see Fig.* 3(ii)).

Proof. Assume that γ and C intersect at two points, say z_1, z_2. Let ℓ be the perpendicular bisector of z_1z_2. Let C^* be the circle containing γ, and let c, r (respectively, c^*, r^*) denote the center and radius of C (respectively, C^*). Clearly, both c and c^* lie on ℓ. Since $z_1, z_2 \in \omega(\gamma)$, we can easily show that ℓ is fully contained in $\omega(\gamma) \cup \bar{\omega}(\gamma)$, so either $c \in \omega(\gamma)$ or $c \in \bar{\omega}(\gamma)$. See Fig. 4 an illustration.

If we fix z_1, z_2, and let c slide along ℓ, we obtain a one-parameter family of circles C all passing through z_1 and z_2. The line through z_1 and z_2 divides the plane into two halfplanes; we denote by H_1 the halfplane containing c^* and by H_2 the complementary halfplane. It is

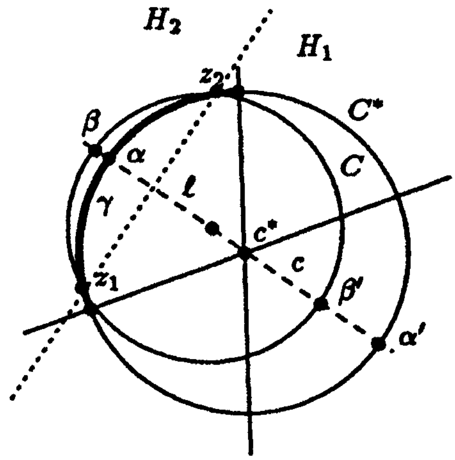

FIG. 4. *Illustration of Lemma* 3.2.

easily checked that, as c moves along ℓ towards $\omega(\gamma)$, $C \cap H_1$ keeps shrinking while $C \cap H_2$ keeps expanding. Since the endpoints of γ lie in H_1, and are incident to C when $c = c^*$, they clearly lie outside C if and only if $c \in \omega(\gamma)$ and inside C if and only if $c \in \bar{\omega}(\gamma)$. This proves the "only if" part of the lemma.

The "if" part is proved in a similar manner. We let z_1, z_2 denote the two points of intersection of C and C^*, and define ℓ, H_1, and H_2 as above. Any of the conditions (i), (ii) implies that ℓ is contained in $\omega(\gamma) \cup \bar{\omega}(\gamma)$. It then suffices to show that both endpoints of γ lie in H_1. If condition (i) occurs then, by the observation made above, we have $C \cap H_2 \supseteq C^* \cap H_2$, so both endpoints of γ must lie in H_1. A similar argument applies for condition (ii). This completes the proof of the lemma. □

In view of Lemmas 3.1 and 3.2, we can divide the pairs of intersecting arcs and circles, $(\gamma, C) \in \Gamma \times \mathcal{C}$, into the following three categories:

I. Both endpoints of γ lie in the exterior of C, the center of C lies in $\omega(\gamma)$, and C intersects the circle containing γ; let $\mathcal{I}_1(\Gamma, \mathcal{C})$ denote the number of such pairs.

II. Both endpoints of γ lie in the interior of C, the center of C lies in $\bar{\omega}(\gamma)$, and C intersects the circle containing γ; let $\mathcal{I}_2(\Gamma, \mathcal{C})$ denote the number of such pairs.

III. Exactly one of the endpoints of γ lies in the exterior of C; let $\mathcal{I}_3(\Gamma, \mathcal{C})$ denote the number of such pairs.

Although all three types of intersecting pairs can be counted by a single algorithm, we prefer to count each of them by a separate procedure, for the sake of clarity.

3.1. Counting $\mathcal{I}_1(\Gamma, \mathcal{C})$. We will view condition I as a conjunction of five constraints, four of which constrain the locations of endpoints of arcs and the locations of centers of circles, and the fifth one requires the two circles to intersect. We will construct a four-level structure, based on the decomposition scheme of Chazelle, Sharir, and Welzl [CSW], which decomposes Γ and $\mathcal{C}$ into a family $\mathcal{F}$ of canonical pairs

$$\mathcal{F} = \{(\Gamma_1, \mathcal{C}_1), \ldots, (\Gamma_k, \mathcal{C}_k)\},$$

such that

(i) $\Gamma_i \subseteq \Gamma$ and $\mathcal{C}_i \subseteq \mathcal{C}$,

(ii) for every pair $(\gamma, C) \in (\Gamma_i, \mathcal{C}_i)$, the endpoints of γ lie in the exterior of C and the center of C lies in $\omega(\gamma)$, and

(iii) for every pair $(\gamma, C) \in \Gamma \times \mathcal{C}$ such that the endpoints of γ lie in the exterior of C and the center of C lies in $\omega(\gamma)$, there is a unique i such that $\gamma \in \Gamma_i$ and $C \in \mathcal{C}_i$.

Each of the four constraints will be satisfied at a different level of the structure. The first-level structure decomposes Γ and $\mathcal{C}$ into a family of canonical pairs $(\Gamma_\triangle, \mathcal{C}_\triangle)$, $\Gamma_\triangle \subseteq \Gamma$, $\mathcal{C}_\triangle \subseteq \mathcal{C}$, so that the counterclockwise endpoints of all arcs in $\Gamma_\triangle$ lie in $E(\mathcal{C}_\triangle)$. The second-level structure then takes each of these canonical pairs and further decomposes them into a collection of canonical pairs $(\Gamma_\tau, \mathcal{C}_\tau)$, so that the clockwise endpoints of all arcs in Γ_τ also lie in $E(\mathcal{C}_\tau)$. Thus the first two levels together ensure that the endpoints of all arcs in Γ_τ lie in $E(\mathcal{C}_\tau)$.

Next, the third and fourth levels decompose each $(\Gamma_\tau, \mathcal{C}_\tau)$ into a family of canonical subsets $(\Gamma_\xi, \mathcal{C}_\xi)$, so that the centers of circles in $\mathcal{C}_\xi$ lie in $\bigcap_{\gamma\in\Gamma_\xi} \omega(\gamma)$. It is easily checked that the set of all fourth-level canonical pairs gives the decomposition $\mathcal{F}$. So, for each fourth-level canonical pair, we compute the number of intersecting pairs of circles using the algorithm described in §2.1, and add up the resulting counts to obtain $\mathcal{I}_1(\Gamma, \mathcal{C})$.

In order to describe the algorithm in detail, we have to define some geometric transforms.

For a circle C of radius r, centered at (a, b), let $\pi(C)$ denote the plane in $\mathbb{R}^3$

$$\pi(C) : z = 2ax + 2by + (r^2 - a^2 - b^2)\,. \tag{7}$$

For a point $p = (\alpha, \beta)$, let $\varpi(p)$ denote the point in $\mathbb{R}^3$

$$\varpi(p) = (\alpha, \beta, \alpha^2 + \beta^2)\,. \tag{8}$$

We will use $\pi^*(C)$ to denote the point dual to the plane $\pi(C)$,

$$\pi^*(C) = (2a, 2b, r^2 - a^2 - b^2)\,, \tag{9}$$

and $\varpi^*(p)$ to denote the plane dual to the point $\varpi(p)$,

$$\varpi^*(C) : z = -\alpha x - \beta y + \alpha^2 + \beta^2\,. \tag{10}$$

It is easily seen that p lies in the exterior of C if and only if $\varpi(p)$ lies above the plane $\pi(C)$, which is the same as saying that the point $\pi^*(C)$ lies below the plane $\varpi^*(p)$.

As in §2, we describe two algorithms. The first algorithm works efficiently when $m^3 \leq n$, and the second algorithm, which uses the first algorithm as a subroutine, works well for all ranges of m and n.

3.2. First algorithm. Let r be some sufficiently large fixed constant. We map the counterclockwise endpoints of arcs in Γ to a collection of planes in $\mathbb{R}^3$

$$\{\varpi^*(\alpha) \mid \alpha \text{ is a counterclockwise endpoint of an arc in } \Gamma\}\,,$$

and decompose the space into a set Ξ of $O(r^3)$ simplices, each of which intersects at most m/r planes. Ξ can be computed in $O(m)$ time using the algorithm of Matoušek [Mat2]. We associate with each simplex $\triangle$ a subset $\Gamma_\triangle \subseteq \Gamma$ of arcs and a subset $\mathcal{C}_\triangle \subseteq \mathcal{C}$ of circles. An arc γ, whose counterclockwise endpoint is α, is in $\Gamma_\triangle$ if $\varpi^*(\alpha)$ intersects the interior of $\triangle$, and a circle $C \in \mathcal{C}_\triangle$ if $\pi^*(C) \in \triangle$. Let $A_\triangle \subseteq \Gamma$ denote the set of arcs corresponding to the planes that lie above $\triangle$. It follows from the above discussion that the counterclockwise endpoints of all arcs in $A_\triangle$ lie in $E(\mathcal{C}_\triangle)$, so we output $(A_\triangle, \mathcal{C}_\triangle)$ as one of the first-level canonical pairs.

We recursively decompose each $(\Gamma_\triangle, \mathcal{C}_\triangle)$. The recursion stops when the number of arcs or circles fall below some fixed constant. In this case, we decompose them by a brute-force method.

Next, we decompose each first-level canonical pair further using the same partitioning techniques, except that we now map the clockwise endpoints (instead of the counterclockwise endpoints) of arcs to planes. Let $(\Gamma_\tau, \mathcal{C}_\tau)$ be a second-level canonical pair. The endpoints of all arcs in Γ_τ lie in $E(\mathcal{C}_\tau)$.

Let $\ell_{CCW}(\gamma)$ (respectively, $\ell_{CW}(\gamma)$) denote the line passing through the center of γ and its counterclockwise (respectively, clockwise) endpoint. We map the arcs of Γ_τ to a set of lines $\{\ell_{CCW}(\gamma) \mid \gamma \in \Gamma_\tau\}$ and decompose the plane in $O(|\Gamma_\tau|)$ time into $O(r^2)$ triangle, each of which intersects at most $|\Gamma_\tau|/r$ lines, again using the technique of [Mat2]. We associate with each triangle ζ a subset of arcs $\Gamma_\zeta \subseteq \Gamma_\tau$ and a subset of circles $\mathcal{C}_\zeta \subseteq \mathcal{C}_\tau$. An arc $\gamma \in \Gamma_\zeta$ if $\ell_{CCW}(\gamma)$ intersects ζ, and a circle $C \in \mathcal{C}_\zeta$ if the center of C lies in ζ. We also associate two other subsets A_ζ and B_ζ of Γ_τ with ζ. An arc γ is in A_ζ (respectively, B_ζ) if both γ and ζ lie below (respectively, above) the line $\ell_{CCW}(\gamma)$. We output $(A_\zeta, \mathcal{C}_\zeta)$ and $(B_\zeta, \mathcal{C}_\zeta)$ as third-level canonical pairs and continue decomposing $(\Gamma_\zeta, \mathcal{C}_\zeta)$ recursively.

Next, for each third-level canonical pair $(\Gamma_\zeta, \mathcal{C}_\zeta)$, we map each arc $\gamma \in \Gamma_\zeta$ to the line $\ell_{CW}(\gamma)$ and apply the same partitioning scheme. For each triangle ξ, Γ_ξ, and $\mathcal{C}_\xi$ are defined in the same way as in the third-level structure. If $\ell_{CCW}(\gamma)$, for each arc $\gamma \in \Gamma_\zeta$, lies below (respectively, above) ξ, we define $A_\xi \subseteq \Gamma_\zeta$ to be the set of arcs γ such that $\ell_{CW}(\gamma)$ lies below (respectively, above) both ξ and γ. We output $(A_\xi, \mathcal{C}_\xi)$ as a fourth-level canonical pair, and continue decomposing $(\Gamma_\xi, \mathcal{C}_\xi)$ recursively.

Each fourth-level canonical pair $(\Gamma_\xi, \mathcal{C}_\xi)$ has the properties that the endpoints of all arcs in Γ_ξ lie in $E(\mathcal{C}_\xi)$ and that centers of all circles in $\mathcal{C}_\xi$ lie in $\bigcap_{\gamma \in \Gamma_\xi} \omega(\gamma)$. By Lemma 3.2, an arc $\gamma \in \Gamma_\xi$ intersects a circle $C \in \mathcal{C}_\xi$ if and only if C intersects the circle containing γ. Therefore, the number of intersecting pairs of arcs in Γ_ξ and of circles in $\mathcal{C}_\xi$ can be counted in time $O(|\mathcal{C}_\xi|^{3+\epsilon} + |\Gamma_\xi| \log |\mathcal{C}_\xi|)$ by the algorithm described in §2.1. For the sake of convenience, we will consider the circle intersection counting procedure as a fifth-level step of the algorithm.

Let $T^{(i)}(a, b)$ denote the maximum time spent in processing a set of a arcs and another set of b circles at level i (i.e., the time spent constructing and processing the structures at levels $\geq i$). It follows from §2.1 that $T^{(5)}(a, b) = O(a^{3+\epsilon} + b \log a)$. For $i \leq 4$ we decompose the problem into κ level-i subproblems, each involving at most a/r arcs and b_j circles, and into κ level-$(i+1)$ problems, each involving at most a arcs and b circles. By construction, $\sum_j b_j = b$, $\kappa = O(r^3)$ for $i = 1, 2$, and $\kappa = O(r^2)$ for $i = 3, 4$. Since we spend $O(a + b)$ time in computing the set of simplices and various subsets of arcs and circles, we obtain the following recurrence:

$$T^{(i)}(a, b) = \begin{cases} O(b \log a + a^{3+\epsilon}) & \text{if } i = 5, \\ \kappa \cdot T^{(i+1)}(a, b) + \displaystyle\sum_{j=1}^{\kappa} T^{(i)}\left(\frac{a}{r}, b_j\right) + & \text{if } i < 5, \\ O(a + b) & \end{cases} \tag{11}$$

where $\kappa = O(r^3)$ and $\sum_{j=1}^{\kappa} b_j = b$. The solution of the above recurrence is easily seen to be

$$T^{(i)}(a, b) = O(a^{3+\epsilon} + b \log^{6-i} a) \quad \text{for } i \leq 5 .$$

Hence we can conclude that, given a set of m circular arcs and another set of n circles, we can count the number of intersection points between them in time $O(n \log^5 m + m^{3+\epsilon})$, for any $\epsilon > 0$.

Remark 3.3. (i) The running time can be improved to $O(n \log m + m^{3+\epsilon})$ by an appropriate modification of the algorithm, but for our purposes the time bound derived above is sufficient.

(ii) If the endpoints of arcs in Γ are already known to lie in $E(\mathcal{C})$, we do not have to construct the first two levels.

3.3. Second algorithm. We now describe another algorithm that works well for all ranges of m and n. In the above algorithm we mapped the endpoints of arcs to planes/lines and circles to points, and constructed a multilevel structure. We follow the same approach, but there are two key differences. The first difference is that we flip the roles of arcs and circles, i.e., we now map circles to planes/lines and endpoints of arcs to points. The second difference is that we stop the recursion when the ratio of the number of circles to the number of points becomes large and solve the problem directly using the first algorithm.

In more detail, we map the circles of $\mathcal{C}$ to a set of planes $\{\pi(C) \mid C \in \mathcal{C}\}$, and partition $\mathbb{R}^3$ into a set Ξ of $O(r^3)$ simplices each of which intersects at most n/r planes [Mat2]. We associate with each simplex $\Delta \in \Xi$ a subset $\Gamma_\Delta \subseteq \Gamma$ and another subset $\mathcal{C}_\Delta \subseteq \mathcal{C}$ of circles. An arc γ, whose counterclockwise endpoint is α, is in Γ_Δ if $\varpi(\alpha) \in \Delta$, and a circle C is in $\mathcal{C}_\Delta$ if $\pi(C)$ intersects the interior of Δ. Let $L_\Delta \subseteq \mathcal{C}$ denote the set of circles C such that $\pi(C)$ lies below Δ. By the above discussion, the counterclockwise endpoints of all arcs in Γ_Δ lie in $E(L_\Delta)$. We output $(\Gamma_\Delta, L_\Delta)$ as one of the first-level canonical pairs.

We recursively construct the first-level structure for $(\Gamma_\Delta, \mathcal{C}_\Delta)$. The recursion stops when $|\Gamma_\Delta|^3 \le |\mathcal{C}_\Delta|$. In this case, we process $(\Gamma_\Delta, \mathcal{C}_\Delta)$ using the first algorithm.

Next, for each first-level canonical pair we apply the same procedure for the clockwise endpoints of arcs. Let $(\Gamma_\tau, \mathcal{C}_\tau)$ be a second-level canonical pair. As earlier, the endpoints of arcs in Γ_τ lie in $E(\mathcal{C}_\tau)$.

In the third level, we dualize the centers of circles of $\mathcal{C}_\tau$ to a set of lines, and partition the plane into $O(r^2)$ triangles so that each triangle intersects at most $|\mathcal{C}_\tau|/r$ lines. We associate with each triangle ζ three subsets $\mathcal{C}_\zeta$, L_ζ, and U_ζ of $\mathcal{C}_\tau$ and a subset Γ_ζ of Γ_τ. An arc $\gamma \in \Gamma_\zeta$ if the point dual to $\ell_{CCW}(\gamma)$ lies in ζ. A circle C is in $\mathcal{C}_\zeta$ (respectively, L_ζ, U_ζ) if the line dual to its center intersects (respectively, lies above, lies below) ζ. Let Γ'_ζ (respectively, Γ''_ζ) be the subset of arcs in Γ_ζ that lie above (respectively, below) the line $\ell_{CCW}(\gamma)$. We output $(L_\zeta, \Gamma''_\zeta)$ and (U_ζ, Γ'_ζ) as third-level canonical pairs. We recursively continue constructing the third-level structure on $(\Gamma_\zeta, \mathcal{C}_\zeta)$. The recursion stops when $|\Gamma_\zeta|^3 \le |\mathcal{C}_\zeta|$. In this case, we invoke the first algorithm.

Next, for each third-level canonical pair $(\Gamma_\zeta, \mathcal{C}_\zeta)$, we apply a similar two-dimensional decomposition to the lines dual to the centers of circles in $\mathcal{C}_\zeta$ and to the points dual to the lines $\ell_{CW}(\gamma)$ for $\gamma \in \Gamma_\zeta$. Each fourth-level canonical pair $(\Gamma_\xi, \mathcal{C}_\xi)$ has the property that the endpoints of arcs in Γ_ξ lie in $E(\mathcal{C}_\xi)$, and that the centers of circles in $\mathcal{C}_\xi$ lie in $\bigcap_{\gamma\in\Gamma_\xi} \omega(\gamma)$. The number of intersection points between $\mathcal{C}_\xi$ and the circles containing the arcs of Γ_ξ can now be computed in time $O(a_\xi^{3/4+\epsilon} b_\xi^{3/4} + a_\xi^{1+\epsilon} + b_\xi^{1+\epsilon})$ using Theorem 2.7.

Let $T^{(i)}(a, b)$ denote the maximum running time of the algorithm at level i, defined as above. Notice that we invoke the first algorithm only when $a^3 \le b$, so the time required by the first algorithm is only $O(b^{1+\epsilon})$. Following the same argument as in the previous subsection, we obtain the following recurrence:

(12)
$$T^{(i)}(a,b) = \begin{cases} O(a^{3/4+\epsilon} b^{3/4} + a^{1+\epsilon} + b^{1+\epsilon}) & \text{for } i = 5\,, \\ O(b^{1+\epsilon}) & \text{for } i < 5, \quad a^3 \le b\,, \\ \kappa \cdot T^{(i+1)}(a,b) + \sum_{j=1}^{\kappa} T^{(i)}\left(a_j, \dfrac{b}{r}\right) + O(a+b) & \text{for } i < 5, \quad a^3 > b\,, \end{cases}$$

where $\kappa = O(r^3)$ and $\sum_j a_j = a$. The solution of the above recurrence is

$$T^{(i)}(a,b) = O(a^{3/4+\epsilon} b^{3/4} + a^{1+\epsilon} + b^{1+\epsilon}) \quad \text{for } i \le 5\,.$$

Hence, $\mathcal{I}_1(\Gamma, \mathcal{C})$ can be counted in time $O(m^{3/4+\epsilon}n^{3/4} + m^{1+\epsilon} + n^{1+\epsilon})$.

3.4. Counting $\mathcal{I}_2(\Gamma, \mathcal{C})$. $\mathcal{I}_2(\Gamma, \mathcal{C})$ is counted using an algorithm very similar to the one we just described. The only difference is that conditions (ii) and (iii) for $\mathcal{F}$ now become:

(ii′) For every pair $(\gamma, C) \in (\Gamma_i, \mathcal{C}_i)$, the endpoints of γ lie in the interior of C and the center of C lies in $\bar{\omega}(\gamma)$, and

(iii′) For every pair $(\gamma, C) \in \Gamma \times \mathcal{C}$, which satisfies the first four constraints of condition II, there is a unique i such that $\gamma \in \Gamma_i$ and $C \in \mathcal{C}_i$.

To this end, we construct a four-level structure similar to that for counting $\mathcal{I}_1(\Gamma, \mathcal{C})$, but we define the subsets of arcs at each level in a somewhat different way, to meet the new kind of constraints. For example, at the first level of the first algorithm, we define $A_\triangle \subseteq \Gamma$ to be the set of arcs corresponding to planes that lie below $\triangle$, and at the third level we define A_ζ (respectively, B_ζ) to be the set of arcs γ such that γ lies above (respectively, below) $\ell_{CCW}(\gamma)$ and $\triangle$ lies below (respectively, above) $\ell_{CCW}(\gamma)$. Similarly, we redefine $L_\triangle, L_\zeta, U_\zeta$ in the second algorithm. Following the same analysis as above, we can show that $\mathcal{I}_2(\Gamma, \mathcal{C})$ can be counted in time $O(m^{3/4+\epsilon}n^{3/4} + m^{1+\epsilon} + n^{1+\epsilon})$.

3.5. Counting $\mathcal{I}_3(\Gamma, \mathcal{C})$. Counting $\mathcal{I}_3(\Gamma, \mathcal{C})$ is relatively simpler, because the conditions on (γ, C') are now defined as a conjunction of only two constraints. Suppose we want to count the number of pairs (γ, C) such that the counterclockwise endpoint of γ lies in the exterior of C and the clockwise endpoint of γ lies in the interior of C; the other case can be handled symmetrically.

The first-level structure is exactly the same as that for counting $\mathcal{I}_1(\Gamma, \mathcal{C})$. Next, for each first-level canonical pair $(\Gamma_\triangle, \mathcal{C}_\triangle)$, we map the clockwise endpoints of arcs in $\Gamma_\triangle$ to a set of planes using (10) and apply the same decomposition scheme, except that for each simplex τ we output a canonical pair $(U_\tau, \mathcal{C}_\tau)$, if the planes corresponding to arcs in U_τ lie below τ. By Lemma 3.1, each $\gamma \in U_\tau$ intersects every circle $C \in \mathcal{C}_\tau$. The total running time is again $O(m^{3/4+\epsilon}n^{3/4} + m^{1+\epsilon} + n^{1+\epsilon})$.

Hence, we can conclude the following theorem.

THEOREM 3.4. *Given a set of m circular arcs and a set of n circles, we can count the number of intersection points between them in time* $O(m^{3/4+\epsilon}n^{3/4} + m^{1+\epsilon} + n^{1+\epsilon})$.

3.6. The case of unit circles and arcs. If all the circles in $\mathcal{C}$ and the circles supporting the arcs of Γ have the same radius, say 1, we can count the number of intersections between $\mathcal{C}$ and Γ more efficiently by modifying the algorithms given above. First, instead of using the algorithm described in §2, we use the algorithm of Agarwal et al. [AASS] to count the number of intersection points between two families of unit circles. Second, it turns out that we only need to apply the decomposition schemes in two dimensions at all levels of the structure (even in the previous algorithms the third and fourth levels apply two-dimensional decomposition schemes; only the first two levels of the structure required three-dimensional schemes). We will describe how to construct the first-level structure for both algorithms; the second-level structure can be handled analogously.

For a point p in the plane, let $D(p)$ denote the unit disk centered at p. We construct the first-level structure of the first algorithm as follows. We map the counterclockwise endpoints of arcs in Γ to a set of unit circles

$$D(\Gamma) = \{D(\alpha) \mid \alpha \text{ is the counterclockwise endpoint of an arc } \gamma \in \Gamma\}.$$

We compute a $\frac{1}{r}$-net $R \subseteq D(\Gamma)$ of size $O(r \log r)$ in time $O(m)$ (see §2 for a definition of a $\frac{1}{r}$-net). We compute the vertical decomposition of the arrangement of R, i.e., draw a vertical line from every vertex of the arrangement and all the points of vertical tangency of every circle in both directions until such a line hits an edge of the arrangement. If there is no such edge, the line is extended to infinity. The vertical decomposition of R partitions the plane into $O(r^2 \log^2 r)$ trapezoidal cells. Since R is a $\frac{1}{r}$-net, each cell of the vertical decomposition intersects at most m/r circles of $D(\Gamma)$. We associate with each cell Δ a subset $\Gamma_\Delta \subseteq \Gamma$ of arcs and another subset $\mathcal{C}_\Delta \subseteq \mathcal{C}$ of circles; an arc γ, whose counterclockwise endpoint is α, is in Γ_Δ if $D(\alpha)$ intersects Δ, and a circle C is in $\mathcal{C}_\Delta$ if the center of C lies in Δ. Let $A_\Delta \subseteq \Gamma$ be the set of arcs corresponding to discs that contain Δ in their exterior. It is easily seen that the counterclockwise endpoints of arcs in A_Δ lie in $E(\mathcal{C}_\Delta)$, so we output $(A_\Delta, \mathcal{C}_\Delta)$ as one of the first-level canonical pairs. We recursively decompose $(\Gamma_\Delta, \mathcal{C}_\Delta)$.

The second-level structure can also be modified similarly. The third and fourth levels remain the same. Finally, for each fourth-level canonical pair, we use the algorithm of [AASS] to count the number of intersection points between two families of circles, so $T^{(5)}(a,b) = O(a^2 + b\log a)$. Furthermore, $\kappa = O(r^2 \log r)$ for all $i \leq 4$, so we have the following recurrence for $i \leq 4$:

$$T^{(i)}(a,b) = O(r^2 \log^2 r) \cdot T^{(i+1)}(a,b) + \sum_{j=1}^{O(r^2 \log^2 r)} T^{(i)}\left(\frac{a}{r}, b_j\right) + O(a+b),$$

where $\sum_j b_j = b$. The solution of the above recurrence is $O(a^{2+\epsilon} + b\log^{6-i} a)$.

At the first-level structure of the second algorithm, we choose a $\frac{1}{r}$-net R' of $\mathcal{C}$ and compute the vertical decomposition of its arrangement. For each cell Δ of the vertical decomposition, we associate the subset $\mathcal{C}_\Delta \subseteq \mathcal{C}$ of circles that intersect Δ and a subset of arcs $\Gamma_\Delta \subseteq \Gamma$ whose counterclockwise endpoints lie in Δ. The set L_Δ is now defined to be the subset of circles that contain Δ in their exterior. We output the pair $(L_\Delta, \mathcal{C}_\Delta)$ and recursively decompose the pair $(\Gamma_\Delta, \mathcal{C}_\Delta)$.

By [AASS], $T^{(5)}(a,b)$ is now $O(a^{2/3+\epsilon}b^{2/3} + a^{1+\epsilon} + b^{1+\epsilon})$. Since $\kappa = O(r^2 \log r)$, the solution of (12) now becomes $O(a^{2/3+\epsilon}b^{2/3} + a^{1+\epsilon} + b^{1+\epsilon})$, which yields the following result.

THEOREM 3.5. *Given a set of m circular arcs, each of radius* 1, *and another set of n unit circles, we can count the number of intersection points between them in time* $O(m^{2/3+\epsilon}n^{2/3} + m^{1+\epsilon} + n^{1+\epsilon})$.

4. Counting arc intersections. In this section we obtain the main result of the paper: We present an algorithm to count the number of intersection points in a collection Γ of n arbitrary circular arcs. As in the previous section, we assume that all arcs in Γ are in general position. The algorithm is based on the following two lemmas.

LEMMA 4.1. *An arc* γ *of a circle* C *intersects an arc* γ' *of another circle* C' *at two points if and only if* $|\gamma \cap C'| = 2$ *and* $|\gamma' \cap C| = 2$ (*see Fig.* 5(i)).

Proof. The "only if" part is trivial. The "if" part follows from the observation that C and C' must intersect at two points, both of which lie in both γ and γ'. □

The case where two given arcs intersect each other in exactly one point is more involved and depends on the pattern of intersections between each arc and the circle containing the other.

LEMMA 4.2. *An arc* γ *of a circle* C *intersects an arc* γ' *of another circle* C' *at exactly one point if and only if one of the following conditions holds:*

(i) $|\gamma \cap C'| = 1$ *and* $|\gamma' \cap C| = 2$.

(ii) $|\gamma \cap C'| = 2$ *and* $|\gamma' \cap C| = 1$ (*see Fig.* 5(ii)).

(iii) $|\gamma \cap C'| = 1$ *and* $|\gamma' \cap C| = 1$, *and the following additional condition holds. Divide* γ *into two equal subarcs, and let* $\gamma_{1/2}$ *be the subarc for which* $|\gamma_{1/2} \cap C'| = 1$. *Define* $\gamma'_{1/2}$ *in an analogous manner. Then both arcs* $\gamma_{1/2}$ *and* $\gamma'_{1/2}$ *fully lie on the same side of the line connecting the centers of the circles* C, C' (*see Fig.* 5(iii)).

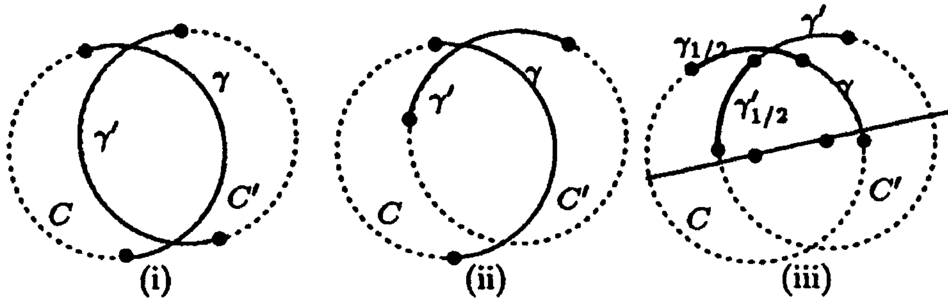

FIG. 5. *Different cases of arcs intersections.*

Proof. To prove the "only if" part of the lemma, suppose $|\gamma \cap \gamma'| = 1$. Then clearly $|\gamma \cap C'| \geq 1$ and $|\gamma' \cap C| \geq 1$. Suppose neither condition (i) nor (ii) holds. Then necessarily $|\gamma \cap C'| = 1$ and $|\gamma' \cap C| = 1$. (If both these numbers were 2, Lemma 4.1 would imply that $|\gamma \cap \gamma'| = 2$, contrary to assumption.) Let γ be an arc that intersects a circle C' in exactly one point. Then it is easily verified that its corresponding halfarc $\gamma_{1/2}$ lies completely on one side of the line connecting the centers of C' and one one side of the circle containing γ, which is the same side containing the point of intersection between γ and C'. This observation easily implies that the second part of condition (iii) is also satisfied.

Consider now the "if" part of the lemma. If γ and γ' satisfy condition (i), then γ' contains the two points of intersection between C and C', and γ contains just one of these points, so clearly these arcs intersect at exactly one point. A symmetric argument applies if condition (ii) holds. Suppose condition (iii) holds, so that $|\gamma \cap C'| = 1$ and $|\gamma' \cap C| = 1$. If γ and γ' do not intersect, then $p = \gamma \cap C'$ and $q = \gamma' \cap C$ are the two distinct points of intersection between C and C', each lying on a different side of the line connecting the circle centers. But then the above observation implies that each of the corresponding halfarcs $\gamma_{1/2}$ and $\gamma'_{1/2}$ fully lie on a different side of this line, contradicting the second part of (iii). This completes the proof of the lemma. □

Lemmas 4.1 and 4.2 suggest the following multilevel structure to count the number of pairs of intersecting arcs in $\Gamma \times \Gamma'$. The preceding lemmas imply that we must count pairs of arcs, (γ, γ'), that satisfy one of the following four conditions (where C is the circle containing γ and C' is the circle containing γ'):

(a) $|\gamma \cap C'| = 2$ and $|\gamma' \cap C| = 2$.
(b) $|\gamma \cap C'| = 1$ and $|\gamma' \cap C| = 2$.
(c) $|\gamma \cap C'| = 2$ and $|\gamma' \cap C| = 1$.
(d) $|\gamma \cap C'| = 1$ and $|\gamma' \cap C| = 1$, and the two halfarcs of condition (iii) of Lemma 4.2 lie on the same side of the line connecting the centers of the circles C, C'.

Finding those pairs of arcs that satisfy one of the conditions (a), (b), or (c) is relatively simple, applying appropriate variants of the machinery presented in the preceding section. Consider for example condition (a). By Lemma 3.2, we can find all these pairs by constructing a nine-level data structure; the first four levels are the same as the first four levels in the data structures of the preceding section, the next four levels are symmetric variants of the first four levels (obtained by interchanging the roles of Γ and Γ'), and the last level tests for intersections between the corresponding circles C, C'. In a similar manner (but using fewer levels) we can find all pairs satisfying condition (b) or (c).

Condition (d) is somewhat more involved. Again we use a multilevel structure. The first two levels enforce, as in the preceding section, the conditions that one endpoint of γ lies inside

C' and one endpoint lies outside C'; the next two levels enforce the symmetric condition for γ' and C. Each resulting canonical pair of subsets (Γ_i, Γ'_i) now has the property that $|\gamma \cap C'| = 1$ and $|\gamma' \cap C| = 1$, for each $\gamma \in \Gamma_i$, $\gamma' \in \Gamma'_i$, with C and C' defined as above.

We next enforce the second part of condition (d). For an arc γ, let $\bar{\omega}(\gamma)$ denote the double wedge formed by $\omega(\gamma) \cup \omega'(\gamma)$. It is easily verified that the second part of (d) is equivalent to requiring that the center of C' lies outside the double-wedge $\bar{\omega}(\gamma_{1/2})$ and the center of C lies outside the double-wedge $\bar{\omega}(\gamma'_{1/2})$. These two subconditions are easy to test for, using standard range counting techniques. That is, we take the collection of centers c' of the circles C' containing the arcs of Γ'_i, and the collection of double wedges that are complements of $\bar{\omega}(\gamma_{1/2})$, $\gamma \in \Gamma_i$, and process them, as in the preceding section, to obtain a canonical collection of pairs of subsets, $(\Gamma_{ij}, \Gamma'_{ij})$, so that, for each such pair, the centers of a circle containing the arc of Γ'_{ij} lie outside every double-wedge $\bar{\omega}(\gamma_{1/2})$, $\gamma \in \Gamma_{ij}$. Finally we apply a symmetric variant of this step to each of these canonical pairs, with the roles of Γ and Γ' being interchanged. The resulting new canonical pairs now fully satisfy condition (d), and the final counting is thus straightforward.

We omit the details of the analysis of the running time of the algorithm, since it is nearly identical to the analysis given in the preceding section. We summarize our results in the following theorem.

THEOREM 4.3. *Given a set Γ of n circular arcs, we can count the number of intersection points in Γ in time $O(n^{3/2+\epsilon})$, for any $\epsilon > 0$.*

If all arcs in Γ have the same radius then, as in §3.6, all levels of the data structure use only two-dimensional decomposition schemes. Thus, following the analysis of the previous section, we can easily conclude the following result.

THEOREM 4.4. *Given a collection Γ of n arcs of the same radius, we can count the number of intersection points in Γ in time $O(n^{4/3+\epsilon})$, for any $\epsilon > 0$.*

5. Conclusion. In this paper we presented efficient algorithms for counting intersections in collections of circles, of circular arcs, and of circles or circular arcs of some fixed radius. Although our algorithms are significantly faster than the best previously known algorithms, we believe that their running time can be further improved, because the best known lower bound for these problems is only $\Omega(n \log n)$. As a first goal, can circular arc intersections be counted in time close to $O(n^{4/3})$, as is the case for collections of segments? We showed that this is the case for circular arcs of the same radius.

Finally, the techniques presented here seem to be quite general. An open problem is to extend them to counting intersections for other types of arcs.

REFERENCES

[Ag1] P. K. AGARWAL, *Partitioning arrangements of lines: I. An efficient deterministic algorithm*, Discrete Comput. Geom., 5 (1990), pp. 449–483.

[Ag2] ———, *Partitioning arrangements of lines: II. Applications*, Discrete Comput. Geom., 5 (1990), pp. 533–573.

[AASS] P. K. AGARWAL, B. ARONOV, M. SHARIR, AND S. SURI, *Selecting distances in the plane*, Proc. 6th ACM Symposium on Computational Geometry, 1990, pp. 321–331; Algorithmica, to appear.

[BO] J. L. BENTLEY AND and T. OTTMANN, *Algorithms for reporting and counting geometric intersections*, IEEE Trans. Comput., C–28 (1979), pp. 643–647.

[Cha] B. CHAZELLE, *Reporting and counting segment intersections*, J. Comput. Systems Sci., 32 (1986), pp. 156–182.

[Chb] ———, *An optimal convex hull algorithm and new results on cuttings*, Proc. 32nd Annual IEEE Symposium on Foundations of Computer Science, 1991, pp. 29–38.

[CE] B. CHAZELLE AND H. EDELSBRUNNER, *An optimal algorithm for intersecting line segments in the plane*, J. Assoc. Comput. Mach., 39 (1992), pp. 1–54.

[CEGS1] B. CHAZELLE, H. EDELSBRUNNER, L. GUIBAS, AND M. SHARIR, *A singly-exponential stratification scheme for real semi-algebraic varieties and its applications*, Proc. 16th International Colloquium on Automata, Languages and Programming, 1989, pp. 179–193; Theoret. Comput. Sci, 84 (1991), pp. 77–105.

[CEGS2] ———, *Algorithms for bichromatic line segment problems and polyhedral terrains*, Algorithmica, to appear.

[CS1] B. CHAZELLE AND M. SHARIR, *An algorithm for generalized point location and its application*, J. Symb. Comput., 10 (1990), pp. 281–309.

[CSW] B. CHAZELLE, M. SHARIR, AND E. WELZL, *Quasi-optimal upper bounds for simplex range searching and new zone theorems*, Algorithmica, 8 (1992), pp. 407–430.

[Cl] K. CLARKSON, *New applications of random sampling in computational geometry*, Discrete Comput. Geom., 2 (1987), pp. 195–222.

[CEGSW] K. CLARKSON, H. EDELSBRUNNER, L. GUIBAS, M. SHARIR, AND E. WELZL, *Combinatorial complexity bounds for arrangements of curves and spheres*, Discrete Comput. Geom., 5 (1990), pp. 99–160.

[CS2] K. CLARKSON AND P. SHOR, *Applications of random sampling in computational geometry* II, Discrete Comput. Geom., 4 (1989), pp. 387–421.

[Ed] H. EDELSBRUNNER, *Algorithms in Combinatorial Geometry*, Springer-Verlag, Heidelberg, 1987.

[EGS] H. EDELSBRUNNER, L. J. GUIBAS, AND J. STOLFI, *Optimal point location in a monotone subdivision*, SIAM J. Comput., 15 (1986), pp. 317–340.

[GOS] L. GUIBAS, M. OVERMARS, AND M. SHARIR, *Counting and reporting intersections in arrangements of line segments*, Tech. Report 434, Dept. Computer Science, New York University, New York, NY, March 1989.

[HW] D. HAUSSLER AND E. WELZL, *ϵ-nets and simplex range queries*, Discrete Comput. Geom., 2 (1987), pp. 127–151.

[Mat1] J. MATOUŠEK, *Construction of ϵ-nets*, Discrete Comput. Geom., 5 (1990), pp. 427–448.

[Mat2] ———, *Approximations and optimal geometric divide-and-conquer*, Proc. 23rd ACM Symposium on Theory of Computing, 1991.

[Mu] K. MULMULEY, *A fast planar partition algorithm*, I, J. Symb. Comput., 10 (1990), pp. 253–280.