Computing a planar widest empty α-siphon in $o(n^3)$ time

Boaz Ben-Moshe* Binay K. Bhattacharya† Sandip Das‡ Daya R Gaur§ and Qiaosheng Shi¶

Abstract

Given a set of n points P in the Euclidean plane, we consider the problem of locating a 1-corner polygonal chain X such that $\min_{p \in P} d(p, X)$ is maximized. The polygonal chain has the added property that its interior angle is α and it partitions P. This corresponds to computing an α-siphon that does not contain any point of P in its interior and whose radius is maximal. In this note we present an algorithm that solves the problem in $o(n^3)$ time. The previous best running time for this problem was $O(n^3 \log^2 n)$ time [3].

1 Introduction

The facility location problem is a classical operations research problem where the task is to position a set of objects (the facilities) in an underlying space such that some distance measure between the facilities and the given points (the demand points) is optimized. In this paper we address an *obnoxious* (*undesirable*) facility location problem where the distance of the nearest point to the facility is maximized. There is a vast body of literature on location theory contributed by the researchers of operations research and computer science community. The website http://www.ent.ohiou.edu/∼thale/thlocation.html (last updated March, 2004) contains about 3400 references on facility location problems. Some applications of the obnoxious route problem include urban, industrial and military task planning when the transportation of some kind of obnoxious material is addressed. On the other hand, the applications of these problems go well beyond the field of location science. For instance the problem of computing a connecting path avoiding collisions is one of the most important tasks in robotics (e.g. [7]).

Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set of points in the Euclidean plane. An empty corridor through P is an open region partitioning P (i.e. intersecting the convex hull of P), which is bounded by two parallel lines such that no points of P lie in the region. A corridor is represented by $R(l, r)$ where l is the axis line of the corridor and r is the radius of the corridor. $R(l, r)$ is defined to be the Minkowski sum of l and a disk of radius r centered at the origin. Houle [9] proposed an $O(n^2)$ solution to compute a widest empty corridor through P. Other variations of the corridor problem were subsequently studied. Bereg et al. [3] considered a corridor, called a siphon, where its axis is a 1-corner polygonal chain consisting of two half-line links. They proposed an $O(n^3 \log^2 n)$ time and $O(n \log n)$ space solution to the widest empty α-siphon problem where the axis of the α-siphon has the interior angle α. When α is not specified, the widest empty siphon problem is solvable in $O(n^4 \log n)$ time requiring $O(n)$ space [5]. The corridor problem is called the boomerang problem [4] when the endpoints of the 1-corner polygonal chain are anchored. An optimal $O(n \log n)$ solution to the boomerang problem was proposed in [2]. Other generalizations have been considered in the literature. Chen [7] considered an L-shaped corridor which is the concatenation of two perpendicular links where each link is composed of an open unbounded trapezoid. Follert et al. [8] considered the corridor whose axis is a ray. Nandy and Bhattacharya [11] investigated the problem of computing a largest-area empty rectangle.

In this paper we consider the widest empty α-siphon problem for a given α. We show for the first time that this problem can be solved in $o(n^3)$ time. We apply the parametric search technique of Megiddo [10] in the design of the algorithm. We show that the feasibility problem of determining if there exists an empty α-siphon of radius r can be solved in $O(n^2 \log^3 n)$ time using $O(n^2 \log^2 n)$ space. Later we show that this sequential algorithm for the feasibility problem is parallelizable. The sequential algorithm for the feasibility problem is described in section 2. Section 3 discusses the parallel implementation of the sequential algorithm. The conclusions are discussed in section 4.

*Department of Computer Science, College of Judea and Samaria, 44837 Ariel, Israel, benmo@yosh.ac.il

†School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada, V5A 1S6; binay@cs.sfu.ca

‡Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Calcutta, India-700 035; sandipdas@isical.ac.in

§Department of Computer Science, University of Lethbridge, 4401 University Drive, Lethbridge, Alberta, Canada, T1K 3M4; gaur@cs.uleth.ca

¶School of Computing Science, Simon Fraser University Burnaby, B.C., Canada, V5A 1S6; qshi1@sfu.ca

2 Feasibility problem

We consider the problem of computing of the widest empty α-siphon through P, for a fixed α. Suppose that $w(v)$ represents the axis of α-siphon where $w(v)$ is a 1-corner polygonal chain consisting of two half-lines starting at v and makes an angle α at v. The corresponding α-siphon is the Minkowski sum of $w(v)$ and a disk of radius r centered at the origin (see Fig. 1).

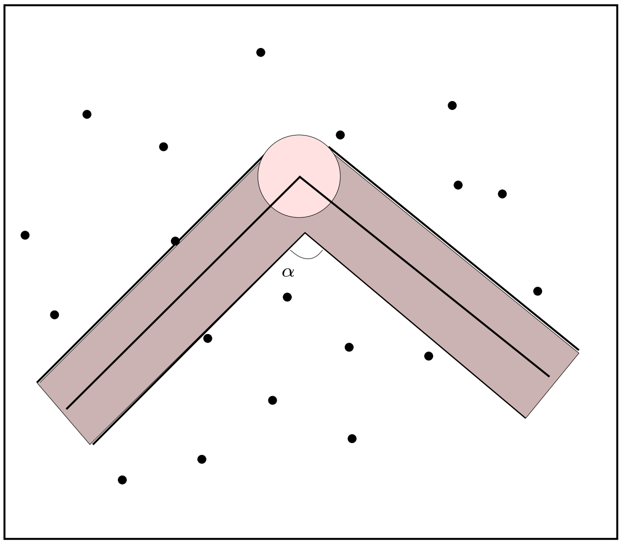

Figure 1: Empty α-siphon through P

Please refer to Fig. 2 for the terminologies introduced here. The α-siphon of radius r is denoted by $R(w(v), r)$ where the two boundaries are called inner and outer boundaries. The inner boundary consists of *cw* and *ccw* half-line links, and the outer boundary consists of a circular segment and two half-line links, *cw* and *ccw*. The axis $w(v)$ also consists of two half-line links, called *cw* and *ccw* links.

The feasibility problem can be formalized as follows. Given a set P of n points in the plane, a fixed value α, $0 \leq \alpha \leq \pi$, and a positive real value r, does there exist an α-siphon with radius r such that no points of P lie in its interior and it partitions P? This is equivalent to determining the existence of a 1-polygonal chain $w(v)$ with interior angle α such that $w(v)$ partitions P and axis $w(v)$ does not intersect the interiors of r-radii disks with the centers at $p_i, i = 1, 2, \ldots, n$ (see Fig. 3).

We first prove the following lemma which characterizes the α-siphon we are looking for.

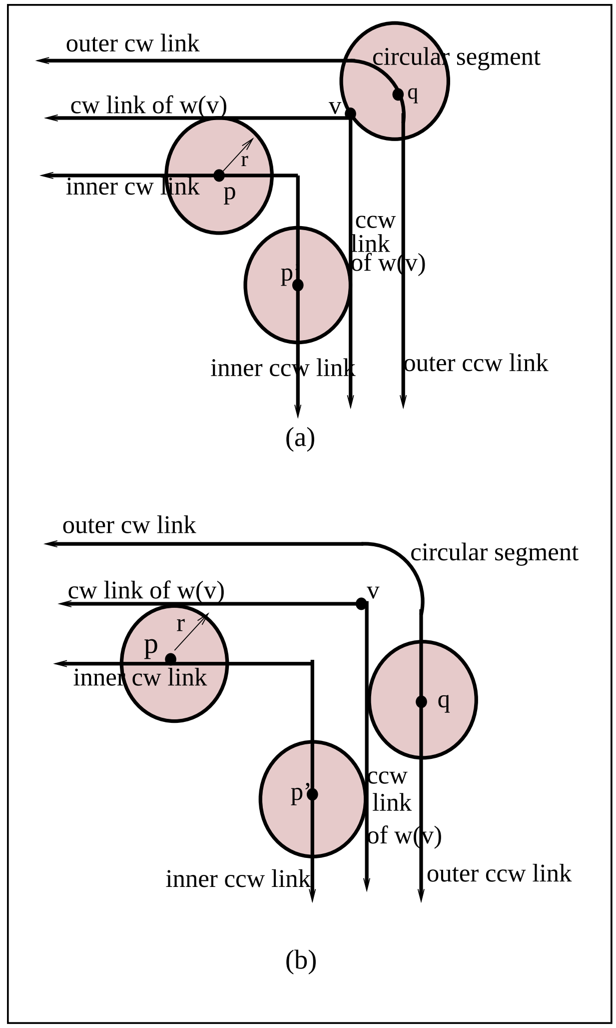

Figure 2: Identifying the various terms of $R(w(v), r)$.

Lemma 1 *The answer to the feasibility problem is "yes" if and only if there exists an empty α-siphon of width r with the following properties. (a) The outer boundary of the α-siphon contains a data point. The point could lie on either the cw link, ccw link or circular segment. (b) Both the cw and ccw links of the interior boundary of the α-siphon contain a data point. It is possible for the same data point to lie on both the links.*

Proof. Suppose we are given a valid empty α-siphon $R(w(v), r)$. Without any loss of generality we assume that the *ccw* axis of the siphon is oriented vertically downwards. Since $R(w(v), r)$ partitions P, we translate $R(w(v), r)$ downwards till it hits a data point, say a. Suppose the inner *cw* link of $R(w(v), r)$ contains the point a. We now translate $R(w(v), r)$ in the direction of *cw* link till another point hits the inner *ccw* link. Let b be the point that lies on the inner *ccw* link of $R(w(v), r)$. We now rotate $R(w(v), r)$ around a and b, keeping the angle of the siphon same, till the boundary hits another point c. If c lies on the inner *cw* link of $R(w(v), r)$, the rotation is continued around c and b. If c lies on the inner *ccw* link of $R(w(v), r)$, the rotation is continued around a and c. Eventually, the outer boundary of $R(w(v), r)$ will encounter a data point since $R(w(v), r)$

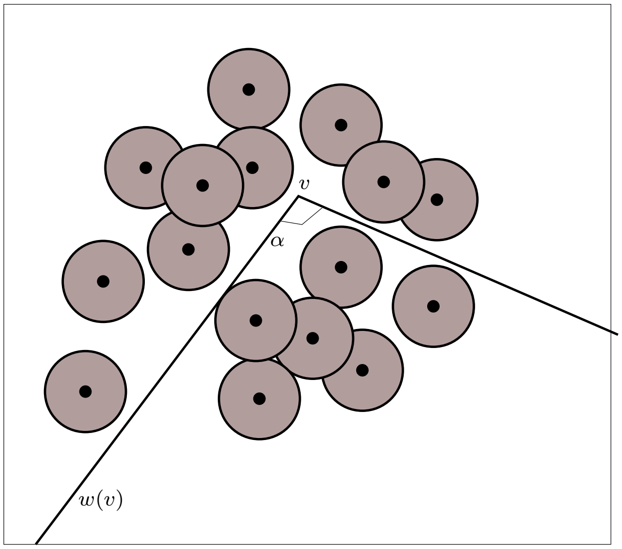

Figure 3: Another way of looking at the α-siphon problem

partitions P. The other part of the proof is straightforward. □

In the following section we will search for an α-siphon $R(w(v), r)$ satisfying the criteria of Lemma 1. For the sake of simplicity, we first assume that $\alpha = \frac{\pi}{2}$. Any other angle can be similarly tackled. We assume that the angle of the *cw* link of $w(v)$ lies between 0 and $\frac{\pi}{2}$, and therefore, the angle of the *ccw* link of $w(v)$ lies between $\frac{\pi}{2}$ and π. We also assume that if the data point on the outer boundary does not lie on the circular segment, it lies on the *ccw* link. Clearly, all the remaining cases are very similar, and therefore can be similarly solved.

The two cases mentioned in Lemma 1(a) are considered separately in the following subsections.

2.1 Data point lying on the circular segment

Let $C_r(P)$ denote the set of disks of radius r with centers located at the points of P. It is clear from the above discussions that we are interested in finding a 1-corner polygonal chain $w(v)$ such that its *cw* link touches a disk of $C_r(P)$, its *ccw* link touches a disk of $C_r(P)$ and its corner point v lies on the boundary of a disk of $C_r(P)$ (see Fig. 2(a)). We also need to make sure that $w(v)$ partitions P and the open region $R(w(v), r)$ is empty. The disk on the boundary of which v lies is called the *starting* disk for both the *cw* and *ccw* links. Here we always assume the case that when a link touches a disk, the center of the disk lies below the link. The other case can be similarly treated.

Our solution methodology consists of two parts. In the first part we implicitly enumerate all the candidate *cw* and *ccw* links, and in the second part we combine the links to construct the axis $w(v)$ of a valid siphon, if it exists. A candidate *cw* link is a half-line f whose starting point lies on the boundary of a disk of $C_r(P)$ and is tangent to another disk of $C_r(P)$, and the open region $R(f, r)$ is empty. We also make sure that the angles of the candidate *cw* and *ccw* links lie in $[0, \frac{\pi}{2}]$ and $[\frac{\pi}{2}, \pi]$ respectively. We now describe the method to enumerate the candidate *cw* links only.

Let p be any point of P. Consider the disk $c_r(p)$ of radius r which is centered at p.

Definition: (Upper-tangent visibility) A point x is upper-tangent visible to disk $c_r(p)$ with respect to $C_r(P)$ if there exists a point x' on $c_r(p)$ such that the line through x and x' is tangent to the disk $c_r(p)$ above p and the line segment xx' does not intersect the interior of any disk in $C_r(P)$.

In Fig. 4, x is upper-tangent visible to $c_r(p)$. x' is called the upper-tangent point of x. However, y is not an upper-tangent visible point since p lies above the line through y and y'. z is also not an upper-tangent visible point since the line segment zz' intersects the interior of another disk.

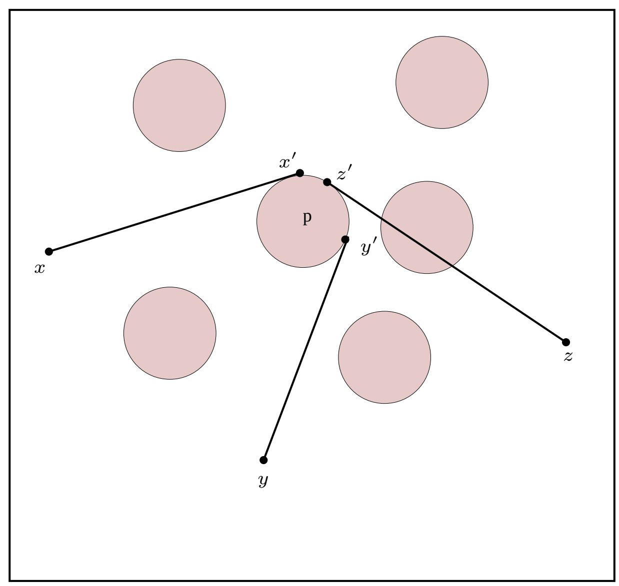

Figure 4: Upper tangent definition

We define $V(P, p, r)$ as the locus of points in the plane which are upper-tangent visible to $c_r(p)$ with respect to $C_r(P)$. $V(P, p, r)$ is called the upper-tangent visibility diagram of $c_r(p)$ (Fig. 5). An useful property of $V(P, p, r)$ is that every point p_i has at most one arc in $V(P, p, r)$ since the disks have the same radius.

Lemma 2 *$V(P, p, r)$ can be constructed in $O(n \log n)$ time.*

Proof. We place n disks of radius r at all points p_i of P. Every disk realizes an arc on the boundary of $c_r(p)$ when projected along the upper-tangent lines as shown

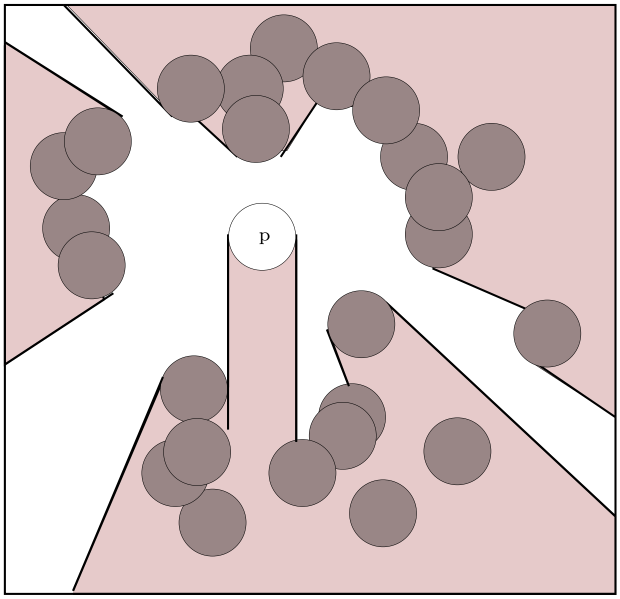

Figure 5: Upper-tangent visibility diagram of $c_r(p)$

in Fig. 6. The disks of $C_r(P)$ are considered in decreasing order of the distances of their centers to the center of $c_r(p)$. Let $V(P_i, p, r)$ denote the upper tangent visibility diagram of $P_i = \{p_1, p_2, \ldots, p_i\}$. When $c_r(p_{i+1})$ is considered, we update $V(P_i, p, r)$ to obtain $V(P_{i+1}, p, r)$. This is done by inserting the projection arc of $c_r(p_{i+1})$ to the set of arcs inserted already. The process is very similar to the one discussed in [3]. The updating step takes $O(1)$ amortized time. □

Once $V(P, p, r)$ is known, all the candidate *cw* and *ccw* links, touching $c_r(p)$ and having angles in the ranges $[0, \frac{\pi}{2}]$ and $[\frac{\pi}{2}, \pi]$ respectively, can be easily determined. We denote a candidate *cw* link by $cw(p, q, \theta)$ where the *cw* link touches the disk $c_r(p)$, starts from the disk $c_r(q)$ and has angle θ. We use the notation $cw(p, q, \theta_1, \theta_2)$ to denote all the cw links of the type $cw(p, q, \theta)$ where $\theta \in [\theta_1, \theta_2]$. In order to be consistent, we will use the notation $cw(p, q, \theta, \theta)$ to denote $cw(p, q, \theta)$. In Fig. 7, $cw(p, a, \theta_4, \theta_4)$ is a candidate *cw* link. All the candidate *cw* links touching $c_r(p)$ with angles in $[\theta_1, \theta_4]$ are represented by $cw(p, a, \theta_1, \theta_2)$, $cw(p, b, \theta_2, \theta_3)$ and $cw(p, c, \theta_3, \theta_4)$ (see Fig. 7). Similar representations are used for the *ccw* links also. Note that a candidate *cw* or *ccw* link may not have a starting disk. In Fig. 7, the *ccw* link $ccw(p, -, \phi, \phi)$ does not have a starting disk. These links can be ignored when we are looking for a siphon where there exists a data point on the circular segment boundary. However, for the case when the outer *cw*/*ccw* link has a data point, these links cannot be ignored. We will assume from now on that every candidate *cw*/*ccw* link has a starting disk. The modifications are minor for the general case. Let $L_{cw}(p)$ and $L_{ccw}(p)$ denote the list of all the candidate *cw* and *ccw* links touching $c_r(p)$ respectively. Let $L_{cw}(P) = \cup_{i=1}^{i=n} L_{cw}(p_i)$ and $L_{ccw}(P) = \cup_{i=1}^{i=n} L_{ccw}(p_i)$. The following lemma easily follows.

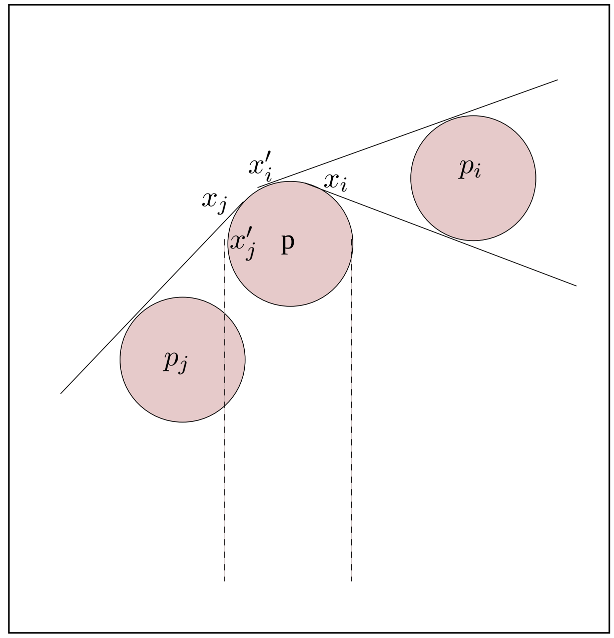

Figure 6: $[x_i, x'_i]$ is the upper-tangent projection of $c_r(p_i)$. The upper-tangent projection of $c_r(p_j)$ is $[x_j, x'_j]$.

Lemma 3 *$L_{cw}(P)$ and $L_{ccw}(P)$ sets can be computed in $O(n^2 \log n)$ time.*

In the second part, we need to find an angle ω, if it exists, such that there exist touching disks $c_r(p)$ and $c_r(p')$, and a point v on a starting disk $c_r(q)$ where the candidate *cw* link, $cw(p, q, \omega, \omega)$, and *ccw* link, $ccw(p', q, \omega + \frac{\pi}{2}, \omega + \frac{\pi}{2})$ start from the same point v (see Fig. 2(a)). It is easy to see that the resulting $\frac{\pi}{2}$-siphon is empty and partitions P. The following lemma facilitates the search process considerably.

Lemma 4 *For any $\theta \in [0, \frac{\pi}{2}]$ and a starting disk $c_r(q)$, there exists at most one candidate cw link of angle θ in $L_{cw}(P)$ with the same starting disk. The same is true for the $L_{ccw}(P)$ list.*

Proof. Suppose, if possible, there exist two candidate *cw* links $cw(p_1, q, \theta, \theta)$ and $cw(p_2, q, \theta, \theta)$. Clearly, the perpendicular the distance between the two links is less than $2r$. Since $c_r(p_1)$ and $c_r(p_2)$ touch the links from below, either p_2 lies in the interior of the region $R(cw(p_1, q, \theta, \theta), r)$ or p_1 lies in the interior of the region $R(cw(p_2, q, \theta, \theta), r)$. Thus we have a contradiction. □

As a consequence of Lemma 4, we need to match each element of $L_{cw}(P)$, say $cw(p, q, \theta_i, \theta_{i'})$ with an element of L_{ccw}, say $ccw(p', q, \phi_j, \phi_{j'})$, if it exists, where

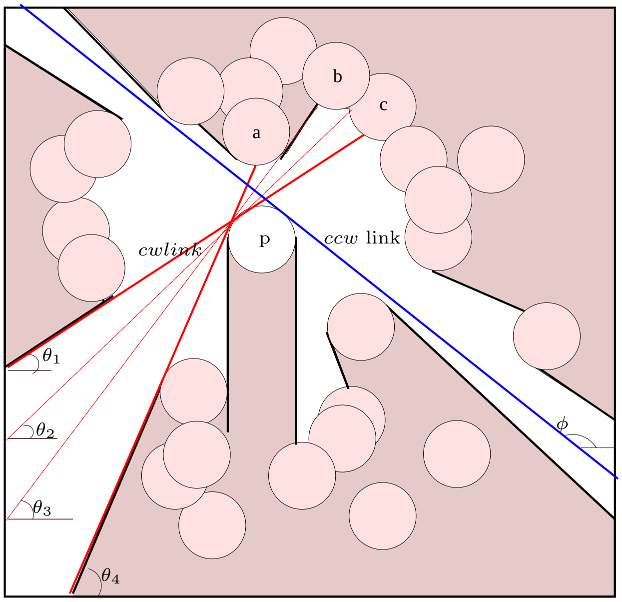

Figure 7: All the candidate cw links, touching $c_r(p)$, with the angles in $[\theta_1, \theta_4]]$ are represented by tuples $cw(p, a, \theta_1, \theta_2)$, $cw(p, b, \theta_2, \theta_3)$ and $cw(p, c, \theta_3, \theta_4)$.

$[\theta_i + \frac{\pi}{2}, \theta_{i'} + \frac{\pi}{2}]$ overlaps with $[\phi_j, \phi_{j'}]$. These matched pairs are easily identifiable once the endpoints of the angle ranges of the elements of $L_{cw}(P)$ and $L_{ccw}(P)$ are sorted. There are $O(n^2)$ matched pairs in the worst case and these can be reported in $O(n^2 \log n)$ time. Each matched pair, $cw(p, q, \theta_i, \theta'_i)$ and $ccw(p', q, \phi_j, \phi'_j)$, is then examined to determine if there exists a point v on $c_r(q)$ such that $cw(p, q, \psi)$ and $ccw(p', q, \psi + \frac{\pi}{2})$ are valid links for some ψ. Since each matched pair can be examined in $O(1)$ time, the total cost for the second step is $O(n^2 \log n)$.

Theorem 5 *The feasibility test to determine if there exists an α-siphon of radius r where the circular segment of the outer boundary contains a data point can be performed in $O(n^2 \log n)$ time. The storage space requirement is $O(n^2)$.*

2.2 Data point lying on the outer ccw link

In this case we are looking for a candidate cw link and a candidate ccw link such that they orthogonally intersect and the resulting $\frac{\pi}{2}$-siphon partitions P (see Fig. 2(b)). We use the $L_{cw}(P)$ and $L_{ccw}(P)$ lists described in section 2.1. The $L_{ccw}(P)$ list can be simplified since we are interested in candidate ccw links which are critical support lines of a pair of disks of $C_r(P)$. A line is a critical support line of two disks $c_r(p')$ and $c_r(\overline{p}')$ if it touches both the disks and the disks lie on different sides of the line. We therefore examine each critical support line (there are $O(n^2)$ such lines) to see if a candidate ccw link can be realized. If it is so, it is added to $L_{ccw}(P)$. We associate the disk, say $c_r(p')$, that touches the ccw link from below as the touching disk. Each stored element is of the form $ccw(p', q', \theta, \theta)$ where θ is the angle of the critical support line. All the $O(n^2)$ critical support lines can be examined easily in $O(n^2 \log n)$ time once the visibility diagrams $V(P, p_i, r), i = 1, 2, \ldots, n$, are known.

Consider a candidate ccw link $ccw(p', q', \phi, \phi)$ of $L_{ccw}(P)$. Also consider the candidate cw links represented by $cw(p, q, \theta, \theta')$ where $\phi - \frac{\pi}{2} \in [\theta, \theta']$. The cw link $cw(p, q, \phi - \frac{\pi}{2}, \phi - \frac{\pi}{2})$ orthogonally intersects $ccw(p'q', \phi, \phi)$ if the two half-lines intersect. This leads to the following lemma which is stated without a proof (see Fig. 8).

Lemma 6 *If a ccw link $ccw(p', q', \phi, \phi)$ orthogonally intersects a cw link of $cw(p, q, \theta, \theta')$, either $q = q'$ or $ccw(p', q', \phi, \phi)$ intersects all the cw links of $cw(p, q, \theta, \theta')$.*

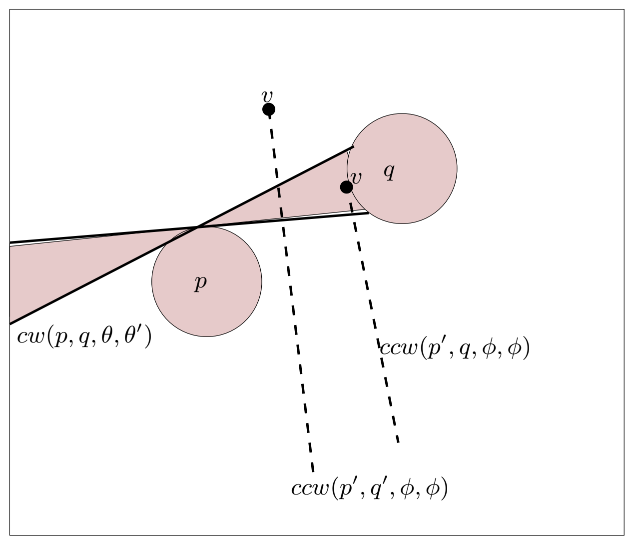

Figure 8: $ccw(p', q', \phi, \phi)$ intersects a cw link of $cw(p, q, \theta, \theta')$ in two different ways depending on the location of the starting point v of the ccw link. v cannot lie in the interior of the shaded region determined by the candidate cw links $cw(p, q, \theta, \theta')$.

Let $x_1, x_2, \ldots, x_s$ be the sorted list of distinct endpoints of the angle ranges of the candidate cw links of $L_{cw}(P)$. We call the interval $[x_i, x_{i+1}]$ an elementary interval. We build a segment tree of the angle ranges whose leaves correspond to elementary intervals. Each internal node u maintains a list of the elements of $L_{cw}(P)$ whose corresponding angle ranges contain $Int(u)$ but does not contain $Int(parent(u))$ where $Int(u)$ is the union of elementary intervals of the leaves in the subtree of u. The segment tree can be built in $O(n^2 \log n)$ time requiring $O(n^2 \log n)$ space in the worst case. A visible description of a segment tree is given in Fig. 9.

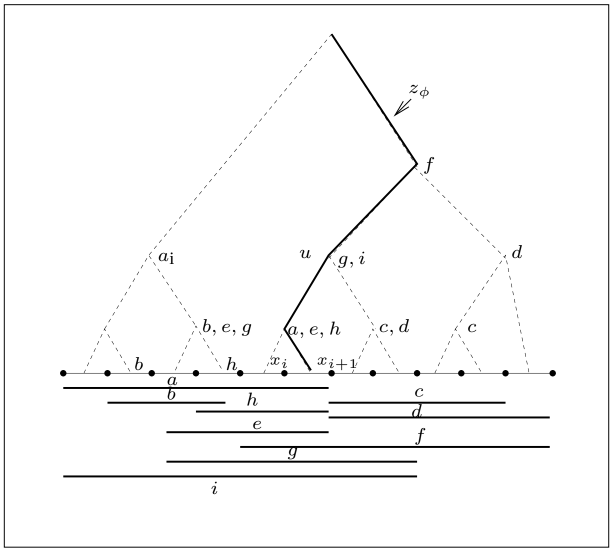

Figure 9: Segment tree structure

The feasibility test is now briefly described. We consider each *ccw* link $ccw(p', q', \phi, \phi)$ of $L_{ccw}(P)$ and do the following. We first find the elementary interval (i.e. the leaf node), say $[x_i, x_{i+1}]$, that contains the angle $\phi - \frac{\pi}{2}$. We then determine all the elements of $L_{cw}(P)$, denoted by $A(\phi)$, stored along the path z_ϕ of the segment tree from the root node to the leaf node representing $[x_1, x_{i+1}]$. The angle ranges of all the elements of $A(\phi)$ contain $[x_i, x_{i+1}]$, and therefore contain $\phi - \frac{\pi}{2}$. Next we determine if there exists any element of $A(\phi)$ of the type $cw(p, q', \theta, \theta')$ for any p. According to Lemma 4, there could be at most one such element. If such an element $a(\phi)$ exists, we check if there exists any *cw* link of the type $a(\phi)$ that orthogonally intersects the *ccw* link $ccw(p', q', \phi, \phi)$. This can be easily done in $O(\log^2 n)$ time by processing each internal node along the path z_ϕ. We are assuming that the elements stored at each internal node are kept in sorted order by their starting disks indices. Let us now consider the case when there is no such intersection as described above. Consider the list of elements of $L_{cw}(P)$ maintained by an internal node $u \in z_\phi$ which is denoted by $A_u(x_{i+1})$. Note that all the angle ranges of the elements of $A_u(x_{i+1})$ contain the angle x_{i+1}. Now, due to lemma 6, we only need to find out whether the *ccw* link $ccw(p', q', \phi, \phi)$ orthogonally intersects a *cw* link of $A_u(x_{i+1})$. This is done by finding out whether $ccw(p', q', \phi, \phi)$ intersects any one of the *cw* links $cw(p_j, q_j, x_u, x_u)$ of $A_u(x_{i+1})$, $j = 1, 2, \ldots, s_u$ where $s_u = |A_u(x_{i+1})|$. Here x_u is the largest angle of the angle range $Int(u)$. Since all the valid *cw* links $cw(p_j, q_j, x_u, x_u)$ of $A_u(x_{i+1})$, for arbitrary p_j and q_j, are parallel, we can preprocess them and answer the query whether the *ccw* link $ccw(p', q', \phi, \phi)$, a half-line, intersects any of the parallel *cw* links of $A_u(x_{i+1})$ in $O(\log^2 n)$ time. This is done by constructing the convex hull of the parallel half-lines $cw(p_j, q_j, x_u, x_u)$, $j = 1, 2, \ldots, s_u$, $s_u = |A_u(x_{i+1})|$, using the divide-and-conquer algorithm where the dividing direction is given by x_u [12]. The preprocessing time and the storage space needed are $O(s_u \log s_u)$ and $O(s_u)$ respectively. Once the element of $A_u(x_{i+1})$, whose representative *cw* link is intersected by $ccw(p', q', \phi, \phi)$, is identified, we can find the correct *cw* link which orthogonally intersects the *ccw* link $ccw(p', q', \phi, \phi)$. In order to guarantee the partition of P by the resulting $\frac{\pi}{2}$-siphon, we look for the intersection of a candidate *cw* link of $A_u(x_{i+1})$ by the line segment $v'p'$ where v' is the starting point of the *ccw* link $ccw(p', q', \phi, \phi)$. Such an intersection, if it exists, will guarantee that the resulting axis of a $\frac{\pi}{2}$-siphon partitions P. The query time in this case is also $O(\log^2 s_u)$. It is possible that the *cw* link of the resulting axis of the $\frac{\pi}{2}$-siphon may not touch any disk. However, we can translate the siphon along the *ccw* link to satisfy all the properties mentioned in Lemma 1. Since the worst case sizes of $L_{cw}(P)$ and $L_{ccw}(P)$ are $O(n^2)$ and every path of the segment tree has $O(\log n)$ internal nodes, we have the following theorem.

Theorem 7 *The feasibility test to determine if there exists an α-siphon of radius r where either the outer cw link or the outer ccw link contains a data point can be performed in $O(n^2 \log^3 n)$ time. The storage space requirement is $O(n^2 \log n)$.*

Combining the theorems 5 and 7, we obtain the following result.

Theorem 8 *The feasibility test for the α-siphon problem can be solved in $O(n^2 \log^3 n)$ time using $O(n^2 \log n)$ space.*

According to Bereg et al. [3], the optimal radius of widest α-siphon is determined by at most three critical points. Therefore, all the critical radii can be enumerated in $O(n^3)$ time. Given the $O(n^2 \log^3 n)$ algorithm for the feasibility test, we can determine the widest empty α-siphon in $O(n^3)$ time. However, in the next section, we briefly describe a parallel version of the $O(n^2 \log^3 n)$ algorithm. This will result in an algorithm that solves the widest empty α-siphon problem in $o(n^3)$ time.

3 Parallel implementation

We apply the parametric search technique [10] to compute the optimal siphon of radius r^*. We need a parallel algorithm to solve the feasibility problem. For that we need to provide parallel implementations of the following steps:

(a) Computing $L_{cw}(P)$ and $L_{ccw}(P)$.

(b) Enumerating the matched pairs as described in section 2.1.

(c) Constructing the segment tree as described in section 2.2.

Each of the upper-tangent visibility diagrams $V(P, p_i, r), i = 1, 2, \ldots, n$, can be constructed using the divide and conquer approach. The corresponding parallel implementation using $O(\log^2 n)$ parallel steps and $O(n)$ processors is straightforward. A similar approach was described in Bereg et al [3]. Thus the lists $L_{cw}(P)$ and $L_{ccw}(P)$ can be determined in $O(\log^2 n)$ parallel steps using $O(n^2)$ processors.

For item (b) mentioned above, we first compute two segment trees in parallel using the angle ranges of the elements for each of $L_{cw}(P)$ and $L_{ccw}(P)$. Each leaf node corresponds to an elementary interval of angles. The segment tree can be constructed in $O(\log n)$ time with $O(n^2 \log n)$ processors using an algorithm by Aggarwal et al. [1]. The elements stored in each internal node u are preprocessed in order to perform he following operation: Given an arbitrary q, determine, if it exists, the element whose starting disk is q. The operation is easy to implement. We keep the list, stored at u, sorted by the starting disks indices. This can be done in $O(\log n)$ parallel steps using $O(n^2 \log n)$ processors. For every starting angle θ of the angle range of the candidate ccw links $ccw(p, q, \theta, \theta')$, we can determine the matched cw links $cw(p', q, \phi, \phi')$ where $\theta - \frac{\pi}{2} \in [\phi, \phi']$, if exists, in $O(\log^2 n)$ time. This is done by first finding the elementary interval containing the angle $\theta - \frac{\pi}{2}$ and then traverse the segment tree of the list $L_{cw}(P)$ from the root to the leaf node representing the elementary interval. Similarly, every starting angle θ of the angle range of the candidate cw links $cw(p, q, \theta, \theta')$, we can determine the matched ccw links $ccw(p', q, \phi, \phi')$ where $\theta + \frac{\pi}{2} \in [\phi, \phi']$, if exists, in $O(\log^2 n)$ time.

For the operation (c) mentioned above we again use the segment tree of $L_{cw}(p)$. The elements stored at each internal node u are preprocessed for performing the following operation: Given an arbitrary ray, determine if there exists an element of $L_{cw}(P)$ that is intersected by the ray. The tree data structure of [12] to construct the convex hull can be computed in $O(\log s_u)$ parallel steps using $O(s_u)$ processors, where s_u is the number of elements stored at u. We use the largest angle of the angle range represented by $Int(u)$ as the dividing direction. Therefore, all the internal nodes of the segment tree can be processed in parallel using $O(\log n)$ steps and $O(n^2 \log n)$ processors. We now query the segment tree for each of the $O(n^2)$ elements of the $L_{ccw}(P)$ list. As observed in the sequential algorithm, each query takes $O(\log^3 n)$ time to answer. Therefore, the steps of section 2.2 can be implemented using $O(\log^3 n)$ parallel time and $O(n^2 \log n)$ processors.

We can now claim the following result.

Theorem 9 *The feasibility test algorithm can be implemented in parallel using $O(\log^3 n)$ steps and $O(n^2 \log n)$ processors. Therefore the widest empty α-siphon problem can be solved in $O(n^2 \log^7 n)$ time. The storage space requirement is $O(n^2 \log n)$.*

4 Conclusions

In this note we showed that the feasibility problem of determining whether there exists any empty α-siphon of radius r can be solved in $O(n^2 \log^3 n)$ time using $O(n^2 \log n)$ space. Previously no such $o(n^3)$-time algorithm was known for the feasibility problem. We also showed that the proposed solution to the feasibility problem can be parallelized. The parallel version of the algorithm uses $O(n^2 \log n)$ processors and $O(\log^3 n)$ parallel steps using the parallel comparison model of Valiant. Therefore, according to Megiddo [10], the widest α-siphon problem can be solved in $O(n^2 \log^7 n)$ time. The method can be modified to solve the widest empty α-siphon, α arbitrary, problem in $o(n^4)$ time [4]. The technique described in this note can easily be applied with some modifications to design $o(n^3)$ time algorithms to solve the following problems: widest empty L-shaped corridor [7], widest empty non-anchored silo [8], largest-area empty rectangle of fixed aspect ratio [11].

References

[1] A. Aggarwal, B. Chazelle, L. Guibas, C. O'únlaing, and C. Yap, " Parallel computational geometry, *Algorithmica*, 3(3):293-328,1988.

[2] B. Ben-Moshe, B. K. Bhattacharya, and Q. Shi, "Computing the widest empty boomerang", In *Proceedings Canadian Conference on Computational Geometry (CCCG)*, 80-83, 2005.

[3] S. Bereg, J. M. Díaz-Báñez, C. Seara, and I. Ventura, "On finding widest empty curved corridors", *Proc. 20th European Workshop on Computational Geometry - EWCG'04*, 33-36, 2004. (Accepted for publication in Computational Geometry: Theory and Applications, 2007.)

[4] J. M. Díaz-Báñez and F. Hurtado, "Computing obnoxious 1-corner polygonal chains", *Computers & Operations Research*, 33(4):1117-1128, 2006.

[5] J. M. Díaz-Báñez, M.A. López, and J.A. Sellarès, "On finding a widest empty 1-corner corridor", *Information Processing Letters*, 98:199-205, 2006.

[6] J. M. Díaz-Báñez, F. Hurtado, H. Meijer, D. Rappaport, and J. A. Sellarès, “The largest empty annulus problem”, *International Journal of Computational Geometry and Applications*,13(4):317–325, 2003.

[7] S-W Cheng, “Widest empty L-shaped corridor”, *Information Processing Letters*, 58:277-283, 1996.

[8] F. Follert, E. Schömer, J. Sellen, M. Smid and Christian Thiel, “Computing a largest empty anchored cylinder, and related problems”, *International Journal of Computational Geometry and Applications*, 7(6):563–580, 1997.

[9] M.E. Houle, “Algorithms for weak and wide separation of sets”, *Discrete and Applied Mathematics*, 45(2):139-159, 1993.

[10] N. Megiddo, “Applying parallel computation algorithms in the design of serial algorithms”, *Journal of ACM*, 30(4):852-865, 1983.

[11] S. C. Nandy and B. B. Bhattacharya, “On finding an empty staircase polygon of largest area (width) in a planar point set”, *Computational Geometry: Theory and Applications*, 26:143-171, 2003.

[12] M.H. Overmars and J. van Leeuwen, “Maintenance of configurations in the plane”, *Journal of Computer System Science*, 23:166-204, 1981.