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. In this note we present an algorithm that solves the problem in $o(n^3)$ time and space. The previous best running time for this problem was $O(n^3 \log^2 n)$ time [2].

1 Introduction

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, which is bounded by two parallel lines such that no points of P lie in the region. A corridor of radius r is represented by $R(l, r)$ where l is the axis of the corridor. Bereg et al. [2] proposed 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 [3]. Other variations of the corridor problem have been proposed (see [2] for the references).

We show for the first time that the widest empty α-siphon problem can be solved in $o(n^3)$ time. 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 and extensions 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; research is partially supported by NSERC and MITACS

‡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

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. For the terminologies please refer to Fig. 1. 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.

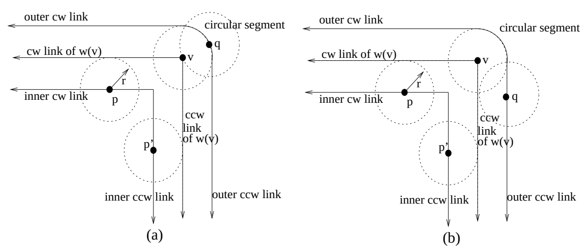

Figure 1: Identifying the various terms of $R(w(v), r)$.

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 the open region $R(w(v), r)$ is empty.

The following lemma characterizes the α-siphon we are looking for.

Lemma 1 *If the answer to the feasibility problem is "yes", there exists an empty α-siphon of radius 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.*

For the sake of simplicity, we first assume that $\alpha = \frac{\pi}{2}$. 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.

2.1 Data point lying on the circular segment

Let $C_r(P)$ denote the set of circles of radius r with centers located at the points of P. We are now interested in finding a 1-corner polygonal chain $w(v)$ such that its *cw* link touches a circle of $C_r(P)$, its *ccw* link touches a circle of $C_r(P)$ and its corner point v lies on the boundary of a circle of $C_r(P)$ (see Fig. 1(a)). We also need to make sure that $w(v)$ partitions P and the open region $R(w(v), r)$ is empty (i.e. $w(v)$ does not intersect the interior of any circle of $C_r(P)$). The circle on the boundary of which v lies is called the *starting* circle for both the *cw* and *ccw* links. We always assume that when a link touches a circle, the center of the circle lies below the link.

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 circle of $C_r(P)$ and is tangent to another circle of $C_r(P)$, and the open region $R(f, r)$ is empty. We also make sure that the angles of the candidate *cw* links lie in $[0, \pi]$ and the angles of the candidate *ccw* links lie in $[\frac{\pi}{2}, \pi]$. We now describe the method to enumerate the candidate *cw* links only.

Let p be any point of P. Consider the circle $c_{2r}(p)$ of radius $2r$ which is centered at p (see Fig. 2). Let l_0 be the horizontal line through p. Let l_0^T be the line which is parallel to l_0 and is tangent to the circle $c_{2r}(p)$ above p. Let $B(l_0)$ be the set of points lying in the open region of the strip $S(l_0)$ where l_0 and l_0^T are the two bounding lines of the strip. Let $m(l_0)$ be the axis line of the strip $S(l_0)$. Let v be the leftmost intersecting point of $m(l_0)$ with the convex hull of the circles of $C_r(B(l_0))$. If $B(l_0)$ is empty then v is set to a point at infinity. We assume without any loss of generality that there always exists at least one point inside the rotating strip. Let v lie on the boundary of circle $c_r(q)$. Now if the half-line along $m(l_0)$, starting at v and avoiding the circles of $C_r(B(l_0))$ also touches the circle $c_r(p)$, then according to our definition the half-line is a candidate *cw* link. This candidate *cw* link is denoted by $h(p, q, 0)$.

Suppose that the half-line $h(p, q, 0)$ is a candidate *cw* link. Suppose now that the strip $S(l_0)$ is rotated counter-clockwise around p by angle θ such that the set $B(l_\theta)$ remains unchanged, i.e. $B(l_\theta) = B(l_0)$ (see Fig. 2). In this case, there exists a starting circle $c_r(q)$ of $C_r(B(l_\theta))$ such that $h(p, q, \theta)$ is a candidate *cw* link. As a matter of fact, for any $\phi \in [0, \theta]$, $B_{l_0} = B_{l_\phi}$, and there exists a starting circle $c_r(q')$ of $C_r(B_{l_0})$ such that

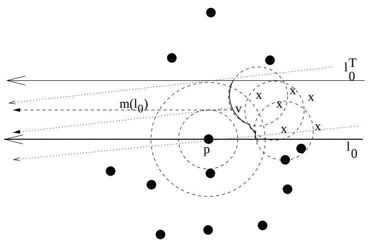

Figure 2: Generating candidate *cw* links touching p

$h(p, q', \phi)$ is a candidate *cw* link.

We now describe our algorithm to determine all the candidate *cw* links, all of which touch the circle $c_r(p)$. Let $S(l_\theta)$ be the strip whose inner (bottom) line has angle θ and goes through p. The sweep starts at $\theta = 0$ and ends when $\theta = \frac{\pi}{2}$. We keep track of the set $B(l_\theta) \subset P$ of points lying in the interior of the strip $S(l_{\theta+\epsilon})$, for arbitrarily small ϵ. We maintain the convex hull of the circles determined by $B(l_\theta)$. Whenever an event takes place, such as when a point is added to or deleted from $B(l_\theta)$, the convex hull of $C_r(B(l_\theta))$ is updated and the half-line $h(p, q, \theta)$ where q is the left most circle of $C_r(B(l_\theta))$ that the axis line $m(l_\theta)$ of $S(l_\theta)$ intersects, is determined. If $h(p, q, \theta)$ touches the circle $c_r(p)$, it is a candidate *cw* link and is stored in a list H_p as a 4-tuple $(p, q, \theta, \mathit{continueflag})$ where the first two items of the tuple indicate the touching and the starting circles respectively, and θ indicates the angle of the half-line. The $\mathit{continueflag}$ is set true if $h(p, *, l_{\theta+\epsilon})$ is still a candidate *cw* link for arbitrarily small ϵ. Consider two consecutive candidate *cw* links (p, q_1, θ_1, b_1) and (p, q_2, θ_2, b_2), where b_1 and b_2 are boolean flags and $\theta_1 < \theta_2$. If b_1 is set false, we can conclude that there is no candidate *cw* link whose angle lies in (θ_1, θ_2). If b_1 is set true, for any $\theta \in [\theta_1, \theta_2]$, there exists a candidate *cw* link whose angle is θ. Suppose now that b_1 is true. If $q_1 = q_2$, all the candidate *cw* links whose angles lie in $[\theta_1, \theta_2]$ have the same starting circle and the same touching circle. We output this interval as $(p, q_1, \theta_1, \theta_2)$. If $q_1 \neq q_2$, the candidate *cw* links whose angles lie in $[\theta_1, \theta_2]$, do not have the same starting circles. We divide the interval $[\theta_1, \theta_2]$ into subintervals such that the starting circle and the touching circle for the candidate *cw* links in each of these subintervals do not change. We output these subintervals along with their starting and touching circles information. Let L_p denote the list of all 4-tuples determined from H_p.

Let $L_{cw} = \cup_{i=1}^{i=n} L_{p_i}$. Clearly L_{cw} can be determined in $O(n^2 \log n)$ time. Each element $(p, q, \theta_1, \theta_2)$ of L_{cw} represents a type of *cw* link where for any angle $\theta \in [\theta_1, \theta_2]$ there exists a candidate *cw* link whose angle is θ and the starting and touching circles are $c_r(p)$ and $c_r(q)$ respectively. In a similar way we can compute the list L_{ccw} of candidate *ccw* links whose angles lie in the range $[\frac{\pi}{2}, \pi]$.

In the second part, we need to find an angle ω such that there exist touching circles $c_r(p)$ and $c_r(p')$, and a point v on a starting circle $c_r(q)$ where the candidate *cw* link, $h(p, q, \omega)$, and *ccw* link, $h(p', q, \omega + \frac{\pi}{2})$ start from the same point v (see Fig. 1(a)). It is easy to see that the resulting $\frac{\pi}{2}$-siphon is empty and also partitions P. The following lemma (proof omitted) facilitates the search process considerably.

Lemma 2 *For any θ and a touching circle $c_r(q)$, there exists at most one element in L_{cw} of the following type $(*, q, *, *)$ where θ lies in the angle range of the element. The same is true for the L_{ccw} list.*

As a consequence of Lemma 2, we need to match each element of L_{cw}, say $(p, q, \theta_i, \theta_{i'})$ with an element of L_{ccw}, say $(p', q, \phi_j, \phi_{j'})$, if it exists, where $[\theta_i + \frac{\pi}{2}, \theta_{i'} + \frac{\pi}{2}]$ overlaps with $[\phi_j, \phi_{j'}]$. There are $O(n^2)$ matched pairs in the worst case and these can be reported in $O(n^2 \log n)$ time in the worst case.

Theorem 3 *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 $h(p, q, \theta)$ and a candidate *ccw* link $h(p', q', \theta + \frac{\pi}{2})$, such that they intersect (orthogonally) and the resulting $\frac{\pi}{2}$-siphon partitions P (see Fig. 1(b)). We use the L_{cw} and L_{ccw} lists described in section 2.1. The L_{ccw} list can be simplified since we are interested in candidate *ccw* links which are critical support lines of a pair of circles of $C_r(P)$. A line is a critical support line of two circles $c_r(p')$ and $c_r(\overline{p}')$ if it touches both the circles and the circles 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}. Each stored *ccw* link is a 3-tuple of the type (p', q', θ) where θ is the angle of a critical support line, and $c_r(p')$ and $c_r(q')$ are the touching and starting circles. All the $O(n^2)$ critical support lines can be examined in $O(n^2 \log n)$ time using the sweep method described in the previous section.

Consider a candidate *ccw* link $h(p', q', \phi)$. Also consider the *cw* links of the type (p, q, θ, θ') where $\phi - \frac{\pi}{2} \in [\theta, \theta']$. The following lemma states what happens when these two links intersect orthogonally.

Lemma 4 *If $h(p', q', \phi)$ orthogonally intersects a cw link of the type (p, q, θ, θ'), either $q = q'$ or $h(p', q', \phi)$ intersects all the cw links of the type (p, q, θ, θ').*

Let $x_1, x_2, \ldots, x_s$ be the list of distinct endpoints of the angle ranges of the elements of L_{cw}, sorted in increasing order. We call the interval $[x_i, x_{i+1}]$ an elementary interval. We build a segment tree of the angle intervals whose leaves correspond to elementary intervals. Each internal node u maintains a list of the elements of L_{cw} whose corresponding angle intervals contain $Int(u)$, that are the union of elementary intervals of the leaves in its subtree. The tree can be built in $O(n^2 \log n)$ time requiring $O(n^2 \log n)$ space in the worst case.

The feasibility test is now briefly described. We consider each *ccw* link (p', q', ϕ) of L_{ccw} 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}, denoted by $A(\phi)$, stored along the path z_ϕ from the root node to the leaf node corresponding to $[x_1, x_{i+1}]$. The angle ranges of all the elements of $A(\phi)$ contain $[x_i, x_{i+1}]$. Next we determine if there exists any element of $A(\phi)$ of the type $(*, q', *, *)$. According to Lemma 2, 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 (p', q', ϕ). This can be easily done in $O(\log^2 n)$ time by visiting each internal node along the path z_ϕ. Suppose that there is no such intersection. Let us consider the list of elements of L_{cw} 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 we need to find out whether the *ccw* link $h(p', q', \phi)$ intersects a *cw* link of the form $h(p, q, \phi - \frac{\pi}{2})$ belonging to an element of the list $A_u(x_{i+1})$. This is done by finding out whether $h(p', q', \phi)$ intersects any one of the *cw* links $(p_j, q_j, *, *))$ of $A_u(x_{i+1})$, $j = 1, 2, \ldots, s_u$ where $s_u = |A_u(x_{i+1})|$. The correctness of this claim follows from Lemma 4. Since all the *cw* links representing the elements of $A_u(x_{i+1})$ are parallel, we can preprocess them and answer the query whether the *ccw* link $h(p', q', \phi)$ intersects any of the representative parallel *cw* links of $A_u(x_{i+1})$ in $O(\log^2 n)$ time. 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 $h(p', q', \phi)$, is identified, we can find the correct *cw* link which orthogonally intersects the *ccw* link $h(p', q', \phi)$. In order to guarantee the partition of P by the resulting $\frac{\pi}{2}$-siphon, we look for the intersection of a representative *cw* link by the line segment $v'p'$ where v' is the starting point of the *ccw* link $h(p', q', \phi)$. 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 $O(\log^2 s_u)$. Since the worst case sizes of L_{cw} and L_{ccw} are $O(n^2)$ and every path of the segment tree has $O(\log n)$ internal nodes, we have the following theorem.

Theorem 5 *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 3 and 5, we obtain the following result.

Theorem 6 *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.*

3 Parallel implementation

The algorithm presented in section 2 is easily parallelizable. We briefly describe the parallel algorithm to compute the L_{cw} list using the parallel comparison model of Valiant. The sweeping process of a strip around p is performed in the following way. For each point $q \in P$, $p \neq q$, we determine the angle range $a_i(q) = [\phi_q, \psi_q]$ such that q will lie inside the strip $S(l_\theta)$ of radius r for any $\theta \in [\phi_q, \psi_q]$. We then build a segment tree of these $O(n)$ intervals in parallel, where each leaf node corresponds to an elementary interval of angles. This can be done in $O(\log n)$ time with $O(n \log n)$ processors using an algorithm by Aggarwal et al. [1]. Then, for each internal node u in the segment tree in parallel, we compute the convex hulls of the circles of radius r with the centers located at the points whose angle ranges are stored at u. This computational step can also be performed in $O(\log n)$ time using $O(n \log n)$ processors [1]. For any given θ, we can now obtain the elements of $B(l_\theta)$ in $O(\log n)$ lists by traversing the segment tree from the root node to the leaf node corresponding to $[x_i, x_{i+1}]$ where $\theta \in [x_i, x_{i+1}]$. Therefore, the leftmost intersection point of the axis $m(l_\theta)$ of the strip $S(l_\theta)$ with the circles of $C_r(B(l_\theta))$ can be determined in $O(\log n)$ parallel steps using $O(n \log n)$ processors. Therefore, the list H_p can be determined in $O(\log n)$ parallel steps using $O(n \log n)$ processors. Once H_p is known, the list L_p can be obtained easily.

Since the rest of the steps of section 2.1 can be easily processed in parallel in $O(\log n)$ steps using $O(n^2)$ processors, the parallel implementation of the algorithm in section 2.1 takes $O(\log n)$ parallel steps using $O(n^2 \log n)$ processors.

In section 2.2, the major step is to build the segment tree with the elements of L_{cw} list. Elements stored in each internal node u are preprocessed for efficient intersection test. Using the divide-and-conquer algorithm in [1], we construct the convex hull in $O(\log s_u)$ parallel steps using $O(s_u)$ processors, where s_u is the number of elements stored at u. 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 need to query the segment tree for each of the at most $O(n^2)$ elements of the L_{ccw} 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 steps using $O(n^2 \log n)$ processors.

Therefore we claim the following result.

Theorem 7 *The feasibility test algorithm can be implemented in parallel using $O(\log^3 n)$ steps and $O(n^2 \log n)$ steps. The widest empty α-siphon problem can therefore be solved in $O(n^2 \log^7 n)$ time [5]. The storage space requirement is $O(n^2 \log n)$.*

4 Conclusions

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. The proposed solution to the feasibility problem can be parallelized and 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 [5], 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, widest empty non-anchored silo, largest-area empty rectangle of fixed aspect ratio (see [2] for the references).

References

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

[2] S. Bereg, J. M. Díaz-Báñez, C. Seara, and I. Ventura. On finding widest empty curved corridors. *Computational geometry: Theory and Applications*, Accepted for Publication, 2007.

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

[4] 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.

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