# Covering points with rotating polygons

Carlos Alegría-Galicia * David Orden † Leonidas Palios ‡ Carlos Seara § Jorge Urrutia ¶


## Abstract

We study the problem of rotating a simple polygon to contain the maximum number of elements from a given point set. We consider variations of this problem where the rotation center is a given point or lies on a line segment, a line, or a polygonal chain.


## 1 Introduction

Given a simple polygon $P$, the *Polygon Placement Problem* consists in finding a function $\tau$ such that a *placement* $\tau(P)$ satisfies a certain property, for $\tau$ combining certain allowed types of movements. The oldest problem of this family we are aware of was studied in the early eighties by Chazelle [5], who given two polygons $P$ and $Q$ explored the problem of finding, if it exists, a placement $\tau(P)$ that contains $Q$ using translation and rotation.

The most recent contribution to these problems is due to Barequet and Goryachev [3]. Among other results, for a point set $S$, a simple polygon $P$, and $\tau$ a composition of translation and rotation, they show how to compute a *maximum cover placement* for $P$, that is, a placement $\tau(P)$ containing the maximum number of points of $S$. For $n$ and $m$ being the sizes of $S$ and $P$ respectively, their algorithm runs in $O(n^3m^3\log(nm))$ time and $O(nm)$ space.

Although translation-only problems have also being considered [1], to the best of our knowledge there are no previous results where $\tau$ is only a rotation[1]. In this paper we thus study the following *Maximum Cover under Rotation (MCR)* problems:

**Problem 1 (Fixed MCR)** *Given a point $r$ in the plane, compute an angle $\theta \in [0, 2\pi)$ such that, after counterclockwise rotating $P$ by $\theta$ around $r$, the number of points of $S$ contained in $P$ is maximized.*

**Problem 2 (Segment Restricted MCR)** *Given a line segment $\ell$, compute a point $r$ on $\ell$ and an angle $\theta \in [0, 2\pi)$ such that, after counterclockwise rotating $P$ by $\theta$ around $r$, the number of points of $S$ contained in $P$ is maximized.*

Applications of polygon placement problems include global localization of mobile robots, pattern matching, and geometric tolerance (see the references in [3]). Rotation-only versions arise in robot localization using a rotating camera [7] or quality control of objects manufactured around a vertical axis.

We show that Problem 1 is 3SUM-hard (an $o(n^{2-\varepsilon})$-time solution for it implies an affirmative answer to the open question of whether an $o(n^{2-\varepsilon})$-time algorithm for 3SUM exists [6]) and present two algorithms to solve it: one requiring $O(nm\log(nm))$ time and $O(nm)$ space, the other taking $O((n+k)\log n + m\log m)$ time and $O(n+m+k)$ space, where $k = O(nm)$ is the number of events. We also describe an algorithm that solves Problem 2 in $O(n^2m^2\log(nm))$ time and $O(n^2m^2)$ space. This algorithm can be easily extended to solve variations of Problem 2 where $r$ lies on a line or a polygonal chain.

## 2 Fixed MCR (Problem 1)

Let $c_p$ be the circle with center $r$ and radius $|\overline{rp}|$, where $p$ is a point in $S$. If instead of rotating $P$ counterclockwise we rotate $S$ in clockwise direction, $c_p$ is the curve described by $p$ during a $2\pi$ rotation around $r$. The endpoints of the circular arcs resulting from intersecting $P$ and $c_p$ mark the rotation angles where $p$ enters (*in-event*) and leaves (*out-event*) the polygon $P$. In the worst case, the number of such events per element of $S$ is $O(m)$, for a total of $O(nm)$ if we consider all the points in $S$. See Figure 1.


*Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, calegria@uxmcc2.iimas.unam.mx.

†Departamento de Física y Matemáticas, Universidad de Alcalá, Spain, david.orden@uah.es. Research supported by MINECO Projects MTM2014-54207 and TIN2014-61627-EXP and by TIGRE5-CM Comunidad de Madrid Project S2013/ICE-2919.

‡Department of Computer Science and Engineering, University of Ioannina, Greece, palios@cs.uoi.gr.

§Departament de Matemàtiques, Universitat Politècnica de Catalunya, Spain, carlos.seara@upc.edu. Research supported by projects Gen. Cat. DGR 2014SGR46, MINECO MTM2015-63791-R.

¶Instituto de Matemáticas, Universidad Nacional Autónoma de México, urrutia@matem.unam.mx. Research supported in part by MTM2006-03909 (Spain) and SEP-CONACYT of México, Proyecto 80268.


[1] Existing results where $\tau$ is a composition of either rotation, translation, and scaling, reduce the search space complexity by only considering placements where a constant number of points from $S$ lie on the boundary of $P$ (see for example references [3] and [4] for algorithms based respectively, on two-point and one-point placements). Rotation-only adaptations of these results would not allow the rotation center to be fixed or restricted to lie on a given curve and therefore, cannot be applied to the problems we deal with in this paper.

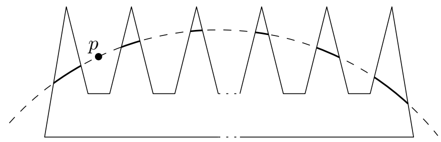



Figure 1: A comb-shaped simple polygon can generate $\Omega(m)$ in- and out-events per point in $S$.

### 2.1 A 3SUM-Hard reduction

We show next that Problem 1 is 3SUM-hard by a reduction from the Segments Containing Points problem that was proved to be 3SUM-hard by Barequet and Har-Peled [2].

**Problem 3 (Segments Containing Points)** *Given a set $A$ of $n$ real numbers and a set $B$ of $m = O(n)$ pairwise-disjoint intervals on the real line, is there a real number $u$ such that $A + u \subseteq B$?*

**Theorem 1** *Fixed MCR is 3SUM-hard*[2].

**Proof.** Let $I$ be an interval of the real line that contains the set $A$ of points and the set $B$ of intervals of an instance of the Segments Containing Points problem. Wrap $I$ on a circle $C$ whose perimeter has length at least twice the length of $I$. This effectively maps the points in $A$ and the intervals in $B$ into a set $A'$ of points and a set $B'$ of intervals on $C$.

Clearly, finding a translation (if it exists) of the elements of $A$ such that $A + u \subseteq B$, is equivalent to finding a rotation of the set of points $A'$ such that all of the elements of $A'$ are mapped to points contained in the intervals of $B'$. To finish our reduction, construct a polygon as shown in Figure 2.

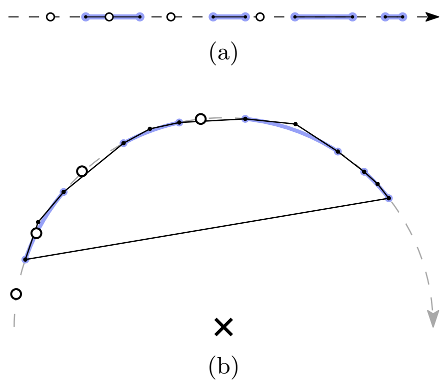



Figure 2: Wrapping $I$ from (a) the real line to (b) a circle $C$. Intervals forming $B$ and $B'$ are highlighted with blue. Elements of $A$ and $A'$ are represented by white points. Additional vertices forming the polygon are the intersection points between the tangents to $C$ at the endpoints of each interval in $B'$.

□

[2]The proof of this theorem is based on the proof of Theorem 4 from Barequet and Har-Peled [2].

### 2.2 An $O(nm\log(nm))$ algorithm.

By Theorem 1 it is unlikely that we could solve Problem 1 in less than quadratic time. We outline now our best solution.

**1. Intersect rotation circles.** Compute the intersection points of $c_p$ and $P$, for every $p$ in $S$.

**2. Compute the sequence of events.** Choose a common reference and translate every intersection point into a rotation angle in $S^1$. Sort all the events by increasing angular order into an event sequence, and determine if they define in- or out-events (see Figure 3). Note that, for each element $p_j$ of $S$, we obtain a sequence of sorted intervals $\mathcal{I}_j = \{I_{j,1}, \ldots, I_{j,i_j}\}$ that determine the rotation angles for which $p_j$ belongs to $P$.

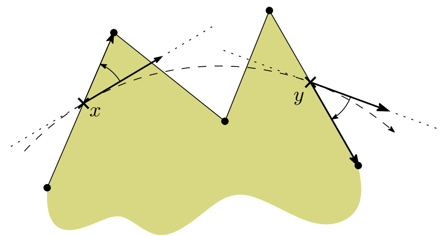



Figure 3: An in-event at $x$ (left turn), and an out-event at $y$ (right turn).

**3. Compute the angle of maximum coverage.** Using standard techniques, we can now perform a sweep on the set obtained by joining all of the intervals in $\mathcal{I}_1 \cup \cdots \cup \mathcal{I}_n$.

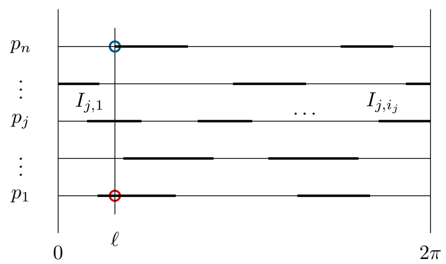



Figure 4: The events sequence and the sweeping line at angle $\theta$. Highlighted with a red circle, the intersection of line $\ell$ with an interval corresponding to $p_1$ ($p_1$ is inside $P$). Highlighted with a blue circle, the intersection of line $\ell$ with one of the endpoints of an interval corresponding to $p_n$ (an in-event).

During the sweeping process, we keep a counter containing the number of points of $S$ in $P$. If an in-event or an out-event occurs, the counter is increased or decreased by one, respectively. At the end of the sweeping process, we report the angular interval(s) where the count is maximized.

Since the complexity of our algorithm is dominated by items 1 and 2, which take $O(nm\log(nm))$ time:

**Theorem 2** *The Fixed MCR problem can be solved in $O(nm \log(nm))$ time and $O(nm)$ space.*

### 2.3 A more efficient algorithm.

Performing a plane sweep using a circular sweepline outwards from the rotation center $r$, it is possible to intersect $P$ and the set of rotation circles in a more efficient way. The idea is to maintain a list of the edges intersecting the sweepline, ordered by appearance while the sweepline is traversed in clockwise direction around $r$. Using the same technique shown in Figure 3, the edges are labeled as defining in- or out-events. The algorithm is outlined next.

**1. Normalize $P$.** In the following steps, we consider $P$ to have no edges intersecting a rotation circle more than once. This can be guaranteed by performing a preprocessing step on $P$: For every edge $e = \overline{ab}$ of $P$, let $p_e$ be the intersection point between the line $\ell$ containing $e$ and the line perpendicular to $\ell$ passing through $r$. If $p_e$ belongs to the relative interior of $e$, subdivide it into the edges $\overline{ap_e}$ and $\overline{p_e b}$. In the worst case, each edge of $P$ gets subdivided in two parts. See Figure 5.

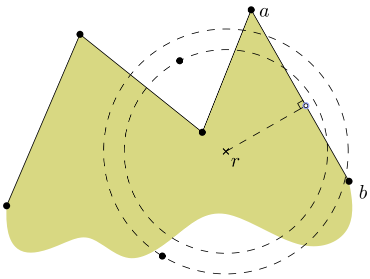



Figure 5: Splitting an edge of $P$.

**2. Process a vertex of $P$.** When the sweepline stops at a vertex of $P$, we update the ordered list of edges intersected by the sweepline.

**3. Compute the intervals sequence for each element of $S$.** When the sweepline reaches a point $p_j$ in $S$, we are ready to compute the sequence $\mathcal{I}_j$ of sorted intervals of $p_j$. It suffices to walk along the ordered list of edges intersected by the sweepline, and compute the corresponding angles clockwise from the ray emanating from $r$ towards $p_j$.

**4. Construct the events sequence.** Since for each point in $S$ we have computed the corresponding sequence of sorted intervals, all we need to do is to merge these (at most $n$) sequences into a complete sequence of events. We do that in a balanced fashion as in the merge sort algorithm.

The normalization process takes $O(m)$ time. Sorting the points in $S$ and the vertices of $P$ by distance from $r$ takes $O(n \log n)$ and $O(m \log m)$ time, respectively. The ordered list of edges intersecting the sweepline is maintained in a balanced binary search tree, so we can process all the vertices of $P$ in $O(m \log m)$ time. On the other hand, processing all the points in $S$ takes $O(k)$ time (recall that $k$ denotes the total number of in- and out-events in a Fixed MCR problem). Finally, merging the $O(n)$ sequences of sorted intervals takes $O(k \log n)$ time from which in $O(k)$ time we obtain a solution. In total, the time complexity of the algorithm is $O(n \log n + m \log m + k \log n)$ time. The space complexity is $O(n + m + k)$. We have thus proved:

**Theorem 3** *The Fixed MCR problem can be solved in $O((n + k) \log n + m \log m)$ time and $O(n + m + k)$ space.*

## 3 Segment Restricted MCR (Problem 2)

Let $\ell = \overline{ab}$ be the line segment restricting the position of the rotation center $r$. Our approach to solve Problem 2 is to characterize, for each $p$ in $S$, the intersection between $P$ and the rotation circle $c_p$ while $r$ moves along $\ell$ from $a$ to $b$. For each edge $e = \overline{uv}$ of $P$, we parametrize the intersection between $c_p$ and $e$ using a function $\omega = f(t)$, for $\omega$ being the clockwise angle shown in Figure 6, and $t$ the $y$-coordinate of $r$. For simplicity, we assume that $a$ lies on the origin $(0, 0)$ and $b$ on the positive $y$-axis.

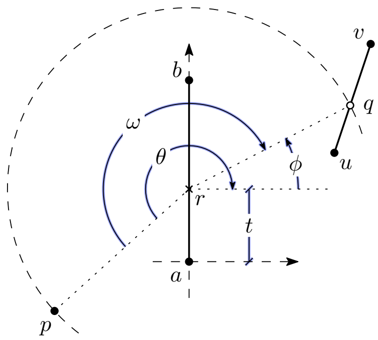



Figure 6: Parametrizing the intersection between $c_p$ and $\overline{uv}$ while $r$ moves along $\overline{ab}$.

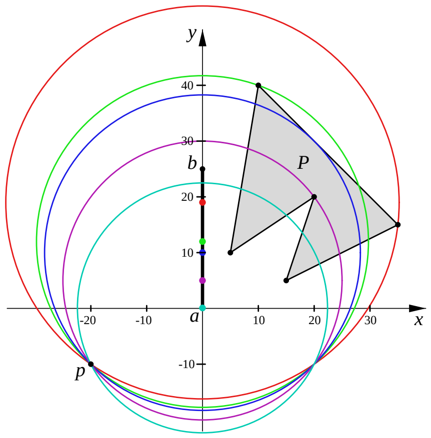
If we consider clockwise and counterclockwise angles being positive and negative respectively, we have from Figure 6 that $\omega = \theta + \phi$. The angle $\theta$ can be easily computed in terms of $t$. By equating the distances from $r$ to $p$ and $q$ and invoking $z = \tan \phi$, we get an equation of the form

$$Az^2 + Bt^2 + Ctz^2 + Dt^2 z + Etz + Ft + Gz + H = 0, \quad (1)$$

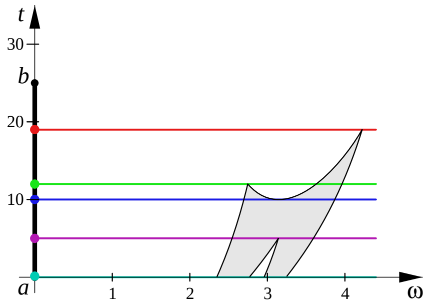
where $A, \ldots, H$ are constants depending on the coordinates of $p$, $u$, and $v$. By resolving Equation (1) for $t$ we obtain

$$t = \frac{f(z) \pm \sqrt{g(z)}}{h(z)}, \quad (2)$$

where $f(z)$, $g(z)$, and $h(z)$ are polynomials of degrees 2, 4, and 1, respectively. The motion of $r$ along $\ell$ thus corresponds to a set of points $(t, \omega)$ for which $p$ belongs to $P$. These points form a set of simple regions in the $t$-$\omega$ plane which are bounded by $O(m)$ curves. Any pair of such regions have disjoint interiors, whereas their boundaries may intersect at most at a common vertex. See Figure 7.




(a)




(b)

Figure 7: (a) A Segment Restricted MCR instance for a point $p$ in $S$ and (b) its corresponding $t$-$\omega$ diagram, where the $\omega$ axis is measured in radians.

By processing all the points in $S$ we end up with a set of $O(nm)$ regions bounded by $O(nm)$ curves in the $t$-$\omega$ plane. From Equation (2) we can show that any two such curves intersect at most a constant number of times, for a total of $O(n^2m^2)$ intersection points in the worst case. Using standard techniques, in $O(n^2m^2 \log(nm))$ time the arrangement of all these regions can be computed, and the dual graph of the resulting arrangement can be traversed looking for the sub-region of maximum depth. Any point in this sub-region determines a position of $r$ and a rotation angle $\omega$ that constitute a solution to the problem. In summary we have:

**Theorem 4** *The Segment Restricted MCR problem can be solved in $O(n^2m^2 \log(nm))$ time and $O(n^2m^2)$ space.*

Note that Problem 2 can also be solved in $O(n^2m^2 \log(nm))$ time even when $r$ is restricted to lie on a line $L$: Compute the Voronoi diagram of $S$ and the vertices of $P$, and apply the algorithm we just described to a segment of $L$ containing all the intersection points of $L$ and the Voronoi edges. Moreover, if we restrict $r$ to lie on a polygonal chain with $s$ segments, we can trivially obtain the optimal placement of $P$ using $O(sn^2m^2 \log(nm))$ time. In both cases the space complexity is $O(n^2m^2)$.

## 4 Concluding remarks

We studied the problem of finding a rotation of a simple polygon that covers the maximum number of points from a given point set. We described algorithms to solve the problem when the rotation center is fixed, or lies on a line segment, a line, or a polygonal chain. Without much effort our algorithms can also be applied when the polygon has holes, and can be easily modified to solve minimization versions of the same problems.

## References

[1] G. Barequet, M. T. Dickerson, and P. Pau. Translating a convex polygon to contain a maximum number of points. *Computational Geometry: Theory and Applications*, Vol. 8(4), (1997), 167–179.

[2] G. Barequet and S. Har-Peled. Polygon containment and translation min-hausdorff-distance between segment sets are 3SUM-hard. *International Journal of Computational Geometry & Applications*, Vol. 11(4), (2001), 465–474.

[3] G. Barequet and A. Goryachev. Offset polygon and annulus placement problems. *Computational Geometry: Theory and Applications*, Vol. 47(3, Part A), (2014), 407–434.

[4] D. Matthew and S. Daniel. Optimal placement of convex polygons to maximize point containment. *Computational Geometry: Theory and Applications*, Vol. 11(1), (1998), 1–16.

[5] B. Chazelle. *Advances in Computing Research*, Vol. 1. Chapter: The polygon containment problem, 1–33, JAI Press, (1983).

[6] A. Gajentaan and M.H. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Computational Geometry: Theory and Applications*, Vol. 5(3), (1995), 165–185.

[7] H. Ishiguro, M. Yamamoto, and S. Tsuji. Omni-directional stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14(2), (1992), 257–262.