アルゴリズム 39－7
（1994. 5. 13）

ある多角形包含問題とX＋Yソーティング問題の関係について

アントニオ エルナンデス バレラ

広島大学

多角形包含問題とは、ある種の移動を許した与えられた多角形Pが、固定された多角形Qの中に置くことができるかどうかを決める問題である。任意の変換のもとで、レクトリニアリーな凸多角形の場合における多角形包含問題と、X＋Yソーティング問題と、連続した数の和のソーティング問題が等価であることを示す。

The relation between a polygon containment problem and the problem of sorting X+Y

Antonio Hernandez Barrera

Hiroshima University

The polygon containment problem is the problem of deciding whether a given polygon P, which is allowed to have some kind of motions, can be placed into another fixed Q. We show that the polygon containment problem in case of rectilinearly convex polygons under translation, the problem of sorting X + Y, and the problem of sorting sums of consecutive numbers are equivalent.

1 Introduction

1.1 An overview of polygon containment problems

The polygon containment problem is the problem of deciding whether a given polygon P, which is allowed to have some kind of motions, can be placed into another fixed Q. It has been studied extensively in the last ten years and it has been attacked using several assumptions about the kind of polygons and motions allowed. Here we briefly mention some of these results.

Suppose P and Q are m-gon and n-gon respectively; P can move while Q remains fixed. If we do not explicitly say anything about them, they will be considered any single polygon in what follows. In 1983 Chazelle ([CH]) proposed an $O(mn^2)$ algorithm for the case where Q is convex and it is possible for P to translate and rotate, and gave an $O(m+n)$ procedure valid if P is restricted to translate. Also assuming only translations appeared in [BFM] an algorithm that runs in $O(mn\ log\ m)$ when both P and Q are rectilinearly convex polygons. The problem considering P convex and under translation was solved in [F] using $O(mn\ log\ mn)$. For both P and Q being non convex and even not connected, Avnaim and Boissonnat proposed in [AB] an $O(m^2n^2\ log\ mn)$ algorithm and showed that the method can be generalized to an $O(m^3n^3\ log\ mn)$ procedure when rotations are also possible.

1.2 What this paper is about

If the size of the output in each specific case is taken into account, it might be said that most of the above-mentioned results are nearly optimal. For example, when both P and Q are rectilinearly convex polygons, it was proved in [BFM] that the boundary of the set of feasible placements of P inside Q is a rectilinearly convex polygonal region that could reach $\Omega(mn)$ edges, while the algorithm used there to obtain that region is $O(mn\ log\ m)$. Except for the case in which Q is convex and P can only translate, we are not aware of either the existence of an optimal in the worst case algorithm for any of the above-stated problems or a proof of the nonexistence of such optimal algorithms.

The results we present here show that the difficulty of the PCP (we will use "PCP" as an abbreviation for "polygon containment problem") under translation when both polygons are rectilinearly convex ones is closely related to the difficulty of sorting sets of numbers of the form $X+Y$. More precisely, we prove that these problems are equivalent.

Sorting $X+Y$, where X and Y are the sets of real numbers $(x_i)_{1\leq i\leq n}$ and $(y_j)_{1\leq j\leq m}$ respectively, consists of sorting the sums $(x_i+y_j)_{1\leq i\leq n,1\leq j\leq m}$. It has been studied before ([HPSS], [FR] and [LA]) but the question of how much computation time is really needed is still open, i.e. an optimal $\theta(mn)$ has not been found.

That equivalence relation means that any algorithm which solves one problem in $O(mn)$ can be transformed into a corresponding algorithm for solving the other one within the same bound. In fact, our proof consists of giving these transformations.

By using that relation and the results in [LA], we also prove here that the PCP under translation in case of rectilinearly convex polygons and the problem of sorting the sums of consecutive numbers

$$\{\sum_{k=i}^{j} a_k / 1 \leq i \leq j \leq n\},$$

$(a_i)_{1 \leq i \leq n}$ any sequence of real numbers, are equivalent when m=n.

2 Preliminaries

2.1 Reductions

When we say in this paper that a problem P *reduces* to another Q, $P \longrightarrow Q$, we mean that an $O(nm)+T(n,m)$ ($n+m$ is the size of all of the problems we will discuss as can be seen in subsection **2.2**) algorithm for solving P exists, where $T(n,m)$ is the time complexity of Q. In other words, the algorithm for P uses an algorithm for Q. P is *equivalent* to Q, $P \longleftrightarrow Q$, if $P \longrightarrow Q$ and $Q \longrightarrow P$.

2.2 General definitions and notations

In what follows, we denote the abscissae and the ordinate of a point p as $p.x$ and $p.y$ respectively.

A polygon P is *rectilinear* if the edges of its boundary are vertical or horizontal. P is *rectilinearly convex* if P is rectilinear and the intersection of every horizontal or vertical line with P is a connected (possibly empty) segment. See fig. 1.

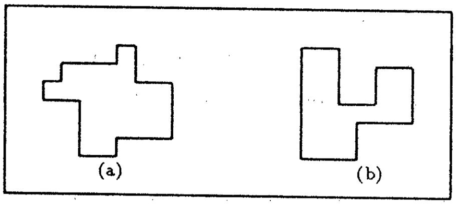

Figure 1: (a) Rectilinearly convex. (b) Non-rectilinearly convex.

Suppose P and Q are m-gon and n-gon respectively. Q is fixed in the coordinate system with origin O_q while P is in the coordinate system with origin O_p which can translate.

Problem P1 Find all placements of O_p in the fixed coordinate system so that P is contained in Q.

It was proved in [BFM] that, when P and Q are rectilinearly convex polygons, the set of placements in which P is contained in Q is a rectilinearly convex polygon with at most nm bounding edges. That set of placements will be denoted here by $H(P,Q)$ and it will be always assumed that it is described by its vertices given in some order.

Let us think of A and B as staircase polygonal lines as in figure 2. Suppose that A (B) refers to the origin O_A (O_B). We will consider that B and its coordinate system are fixed and

A with its coordinate system can translate. We suppose that the bottommost edge of B (the topmost edge of A) is horizontal and its left (right) extreme is in some point in the infinite. On the other hand, the rightmost edge of B (the leftmost edge of A) is vertical and its upper (lower) extreme is also in some point in the infinite.

We assume that B is represented by vertices b_1, b_2, ..., b_n and A by vertices a_1, a_2, ..., a_m as in figure 2.

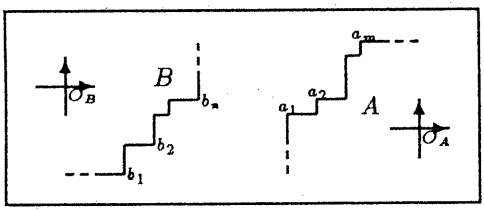

Figure 2: The staircase polygonal lines A and B

Problem 2 Find the vertices of the rectilinear polygonal line that O_A would describe in the fixed coordinate system as A slides along the edges of B.

In other words, we want a description of all the positions that O_A would reach if, beginning in some point in the infinite with the topmost edge of A "touching" the bottommost edge of B, A translates to the right until it is possible to go upward without intersecting B, then A translates in that direction until the distance between some pair of horizontal segments is zero, etc. Notice that in essence, what we want is to solve a polygon containment problem in which both regions are not bounded. So we can call that rectilinear polygonal line $H(A, B)$.

For a certain position O of O_A in the system O_B, the coordinates of the vertex a_j $(1 \leq j \leq m)$ in the fixed coordinate system are $(O.x + a_j.x, O.y + a_j.y)$ that we will denote as a_j^o.

Problem P3 Sort $X + Y$, where X and Y are the sets of real numbers $(x_i)_{1 \leq i \leq n}$ and $(y_j)_{1 \leq j \leq m}$ respectively, i.e. sort the sums $(x_i + y_j)_{1 \leq i \leq n, 1 \leq j \leq m}$.

We will use in some parts of this paper the term $X - Y$ instead of $X + Y$, because it is clearer with respect to the situation we are discussing. Obviously sorting $X + Y$ is equivalent to sorting $X - Y$.

3 The PCP reduces to sort $X + Y$

First we prove $P2 \longrightarrow P3$.

A *contact* between a_j and b_i is the intersection between A and B which arises when a_j is slid along b_i (figure 3). Notice that a contact between two vertices does not always arise as A slides along the edges of B.

Initially suppose O_A is placed in a position O so that there is contact between b_1 and a_m, that is, $a_m^o = b_1$.

Let's consider the sets $X_V = \{b_i.y\}_{1 \leq i \leq n}$ and $Y_V = \{a_j^o.y\}_{1 \leq j \leq m}$. Let's denote the difference $b_i.y - a_j^o.y$ as $(i,j)_V$. So $X_V - Y_V = \{(i,j)_V / 1 \leq i \leq n, 1 \leq j \leq m\}$. For the sake of simplicity we assume now that no two pairs

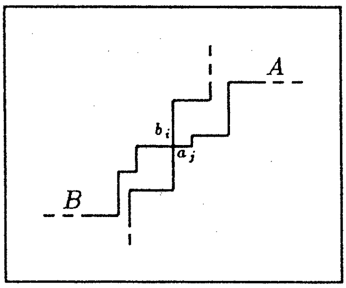

Figure 3: A contact between a_j and b_i

in $X_V - Y_V$ are the same, i.e. there is not $i, j, r, s,\ 1 \le i, r \le n,\ 1 \le j, s \le m$, such that $(i,j)_V = (r,s)_V$.

Loosely speaking, our algorithm computes all contacts, in the order they take place, between vertices in B and vertices in A as the latter translates. Note that, if for a certain location of O_A, p, a_j contacts b_i, and we know that the next contact as A slides first upward and after to the right is b_r and a_s, then the next two positions of O_A are $(p.x, p.y + (b_r.y - a_s^p.y))$ and $(p.x + (b_r.x - a_s^p.x), p.y + (b_r.y - a_s^p.y))$, where the last one is the position for O_A in which b_r contacts a_s. This observation tell us how to obtain all the positions of O_A.

So, the problem is to obtain b_r and a_s. Let $(X_V - Y_V)^*$ be the sorted set $X_V - Y_V$.

Lemma 3.1

If b_i touches a_j $(1 \le i \le n,\ 1 \le j \le m)$ for a certain location p of O_A then, the next contact as A slides upward and to the right is between b_r and a_s if, $(r,s)_V$ is the next pair in $(X_V - Y_V)^*$ following $(i,j)_V$ on the condition that $b_r.x - a_s^p.x > 0$.

Thus, to compute the set of points which O_A can be at, it is only necessary to scan the sequence $(X_V - Y_V)^*$ asking for the pairs which satisfy the condition of the lemma.

It should be noticed that the running time of this algorithm would be dominated by sorting. The rest could be done in $O(nm)$. If we use a standard sorting algorithm an $O(nm\ log\ m)$ upper bound would be obtained, which equals the one showed in [BFM]. Could $X_V - Y_V$ be sorted in less time, taking into consideration the particular structure of this set?. We already said that for this problem of sorting, the question of how much computation time is really needed is still open.. It was proved in [LA] that, for two given sequences of numbers $(x_i)_{1 \le i \le N}$ and $(y_j)_{1 \le j \le N}$, there exists an algorithm to compute the N^2 sums $(x_i + y_j)_{1 \le i,j \le N}$ in $O(N^2)$ comparisons. In fact, such an algorithm was presented there, but unfortunately, its performance was analized just in terms of comparisons and the existence of an algorithm with a similar bound in the case of a more general study remains still unknown.

Thus, we have proved the following theorem:

Theorem 3.2 P2 $\longrightarrow$ P3.

Now we are ready to prove the main result of this section:

Theorem 3.3 P1 $\longrightarrow$ P3.

Proof The algorithm described in [BFM] to compute the set of placements of P so that P is contained in Q consists of three parts:

(1) Divide P and Q each into four quadrant parts. The lower right quadrant for example,

consists of the bottommost edge, the steps up and to the right and the rightmost edge. In a similar way we can describe the lower left, upper right and upper left quadrants.

(2) Determine the placements for matching quadrant parts (with open edges extended to rays).

(3) Intersect the four regions of step (2), giving the set of possible positions of P inside Q.

As it was shown in [BFM], (1) can be done in $O(m+n)$ while (3) requires $O(mn)$. The algorithm designed for the second step took $O(mn \ log \ m)$. Then, the whole algorithm complexity depends on (2). It is straightforward to recognize in the quadrant parts the staircase polygonal lines defined in the problem P2. So P1 $\longrightarrow$ P2. It follows from Theorem 3.2 that P1 $\longrightarrow$ P3. □

4 Sorting $X+Y$ reduces to solve the PCP

In this chapter we prove that $P3 \longrightarrow P1$.

Let's make some assumptions in relation with the problem of sorting $X-Y$. They do not result in any loss of generality. We will consider X and Y two sorted sets of numbers on the real line (it can be done in $O(max(nlogn, mlogm))$), i.e. $x_1 \leq x_2 \leq ... \leq x_n$, $y_1 \leq y_2 \leq ... \leq y_m$. We also assume $y_m = x_1$ (otherwise we can get sorted $X-Y$ by sorting $X - \{y_j - \alpha / y_j \epsilon Y\}$, α such that $y_m - \alpha = x_1$).

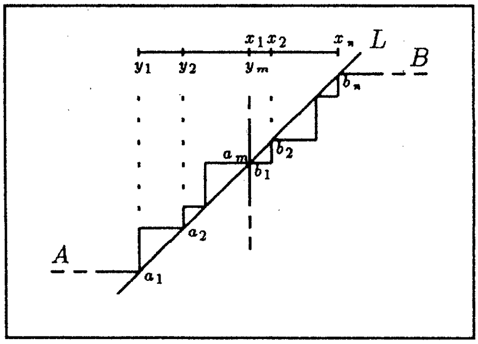

Figure 4: Generation of the polygonal lines used in the proof of theorem 4.1

Theorem 4.1 $P3 \longrightarrow P2$.

Proof Using the set Y, we construct the staircase polygonal line A. Let's generate the vertices a_j, $1 \leq j \leq m$ in such a way that all of them lie on the same line L. Let's consider that the clockwise angle between this line and the abscissae axis, measured from L, is some θ, $0 < \theta < 90°$. We consider that, for every j, $1 \leq j \leq m$, $a_j.x = y_j$. The bottommost edge extends to a ray to the left while the rightmost edge extends to a ray upward. We construct the polygonal line B in a similar way, assuming that every vertex b_i, $1 \leq i \leq n$, lies on L. See figure 4. These constructions can be done in $O(n+m)$.

Let's suppose that an algorithm for determining $H(A,B)$ (the path O_A describes as O_A translates along the edges of B) is known. Let $h_0, h_1, ..., h_k$ be the list of its vertices, where h_0 is a vertex with its ordinate in the infinite. Then we have these lemmas:

Lemma 4.2 $\forall v, 1 \leq v \leq k, h_v.x - h_0.x$ is equal to one, or more, sums $x_i - y_j, 1 \leq i \leq n, 1 \leq j \leq m$.

Lemma 4.3 $\forall i, j,\ 1 \leq i \leq n, 1 \leq j \leq m, \exists v, 1 \leq v \leq k$ such that $x_i - y_j = h_v.x - h_0.x$.

Lemma 4.2 and lemma 4.3 tell us that the elements of $X - Y$ can be got sorted from $H(A, B)$. □

Therefore,

Theorem 4.7 $P3 \longrightarrow P1$

5 Yet another equivalent problem

The results we present in this section derive from those in [LA]. There, the following problem was considered: Let $a_1, a_2, ..., a_n$ be n numbers and let's define for $1 \leq i \leq j \leq n$

$$\sigma(i,j) = \sum_{k=i}^{j} a_k$$

Problem P4 Sort the set of numbers $A^+ = \{\sigma(i,j),\ 1 \leq i \leq j \leq n\}$

Theorem 5.1 If we consider that in the problem of sorting $X + Y$, the size of both sets X and Y is the same, i.e. $n = m$, then $P3 \longrightarrow P4$.

Proof See the explanation in [LA] about how an algorithm to sort A^+ can be adapted to sort $(x_i + y_j)_{1 \leq i,j \leq n}$ □

Theorem 5.2 $P4 \longrightarrow P3$

Proof Suppose we are given n numbers $a_1, a_2, ..., a_n$. Without loss of generality we assume n odd. Let's take $N = \frac{n+1}{2}$. Following a

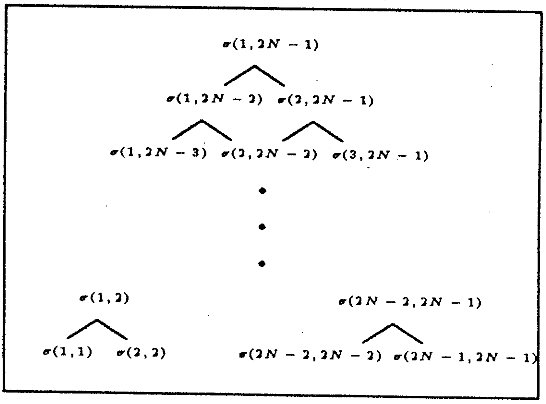

Figure 5: Representation of A^+

similar idea to that in [LA], we will define the $2N$ values $x_1, x_2, ..., x_N,\ y_1, y_2, ..., y_N$ as follows:

$$\begin{aligned} y_N &= \text{any real number} \\ y_{N-i} &= y_{N-i+1} - a_i & 1 \leq i \leq N-1 \\ x_1 &= a_N - y_1 \\ x_i &= a_{N+i-1} + x_{i-1} & 2 \leq i \leq N \end{aligned}$$

The set A^+ can be represented like in figure 5.

Therefore, the set A^+ can be also represented in the way figure 6 shows.

If we take a look at the three regions indicated in the pyramid of figure 9, we can understand that actually, the central one is $X + Y$, the region on the left is the set $\{y_i - y_j / 1 \leq j < i \leq N\}$ and the region on the right is the set $\{x_i - x_j / 1 \leq j < i \leq N\}$.

The last two sets are subsequences of $Y - Y$

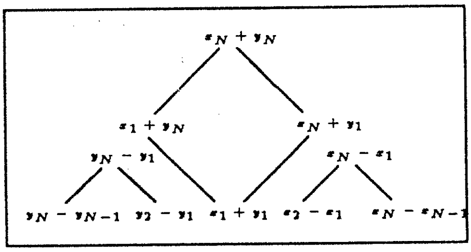

Figure 6: Another way of seeing A^+

and $X - X$ respectively. Hence, an algorithm for sorting $X + Y$ may be used to sort each region. By merging the three sorted sets, set A^+ will be finally sorted. □

6 Conclusions

The main result of this paper is a proof of the equivalence between the polygon containment problem in case of rectilinearly convex polygons under traslation, the problem of sorting $X + Y$ and the problem of sorting sums of consecutive numbers.

7 References

[AB] F. Avnaim, J.D. Boissonnat, "Polygon placement under translation and rotation", *Proceedings of the STACS 1988*, Lectures Notes in Computer Science 294, Springer-Verlag, 1988, pp. 322-333.

[BFM] B.S. Baker, S.J. Fortune, S.R. Mahaney, "Polygon containment under translation", *Journal of Algorithms* 7 (1986), pp. 532-548.

[CH] B.Chazelle, "The polygon containment problem", *Advances of Computing Research*, Vol. 1, JAI Press Inc, Greenwich, Connecticut, 1983, pp. 1-33.

[F] S.J. Fortune, "A fast algorithm for polygon containment by translation", *Proceedings of the 12th ICALP*, Lecture Notes in Computer Science 194, Springer-Verlag, 1985, pp.189-198.

[FR] M.L. Fredman, "How good is the information theory bound in sorting?", *Theoretical Computer Science* 1 (1976), pp. 355-361.

[HPSS] L.H. Harper, T.H. Payne, J.E. Savage, E. Straus, "Sorting X+Y", *Communications of the ACM*, June 1975, Vol. 18, Number 6, pp. 347-349.

[LA] J.L. Lambert, "Sorting the sums $(x_i + y_j)$ in $O(n^2)$ comparisons", *Proceedings of the STACS 1990*, Lectures notes in Computer Science 415, Springer-Verlag, 1990, pp 195-206.