Discrete Comput Geom 29:153–158 (2003) DOI: 10.1007/s00454-002-2829-x



Finding Sets of Points without Empty Convex 6-Gons

Mark Overmars

Department of Information and Computing Sciences, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands markov@cs.uu.nl

Abstract. Erdös asked whether every large enough set of points in general position in the plane contains six points that form a convex 6-gon without any points from the set in its interior. In this note we show how a set of 29 points was found that contains no empty convex 6-gon. To this end a fast incremental algorithm for finding such 6-gons was designed and implemented and a heuristic search approach was used to find promising sets. The experiments also led to two observations that might be useful in proving that large sets always contain an empty convex 6-gon.

1. The Problem

Given a set *V* of *n* points in the plane, no three of which are collinear, we consider subsets $\Phi_k(V)$ of *V* of cardinality *k* that lie in convex position, i.e., they form the vertices of some convex *k*-gon, and there is no point of *V* in the interior of this polygon (the polygon is empty). Let F_k denote the smallest number such that any set *V* of cardinality at least F_k contains some subset $\Phi_k(V)$. Erdös [3] proposed the study of finding bounds on F_k . See [6] for an overview of the state of the art for this problem and related problems. It is trivial to prove that $F_1 = 1$, $F_2 = 2$, $F_3 = 3$ and $F_4 = 5$. The following results are known:

Theorem 1.1 [4]. $F_5 = 10$.

Theorem 1.2 [5]. $F_7 = \infty$. (In other words, for any size *n* there exists a set of *n* points without an empty convex 7-gon.)

Clearly this last result also holds for k > 7. For k = 6 no exact bounds on F_k are known. In [1] Avis and Rappaport give a method to determine whether a given set of

points does contain an empty convex 6-gon. Using this method they succeeded in finding a set of 20 points that does not contain an empty convex 6-gon, showing that $F_6 \ge 21$.

About 10 years ago we devised a new incremental algorithm for checking whether a set contains an empty convex 6-gon. Using an implementation of that algorithm we managed to find a set of 26 points without an empty 6-gon, showing that $F_6 \ge 27$. The results were never properly published [7]. Recently we reimplemented the algorithm and combined it with heuristic search techniques, resulting in a set of 29 points without an empty convex 6-gon. This note discusses this result and the algorithm and implementation used. Because the original note [7] is not publicly available, we briefly repeat the approach here. The experiments also led to two observations that might be helpful in proving that large enough sets always contain an empty convex 6-gon.

2. The Algorithm

Because our implementation creates sets in an incremental way we want to solve the following problem:

Problem. Given a set *V* of *n* points without an empty convex 6-gon, and a point *p*, test whether $V \cup \{p\}$ contains an empty convex 6-gon.

To be able to test many different points an efficient algorithm (and, in particular, efficiently implementable) for this problem was required. To this end an algorithm by Dobkin et al. [2] for finding empty polygons was adapted to our incremental approach. We give a brief sketch of the algorithm. For more details consult [7]. An empty convex 6-gon in $V \cup \{p\}$ must have p as a vertex. Consider the rays r^+ , that runs from p upwards to infinity, and r^- , that runs downwards to infinity. An empty convex 6-gon including p cannot intersect both r^+ and r^- . We treat both cases separately.

We consider the case in which the 6-gon does not intersect r^+ . The other case can be treated similarly. We sort all points of V counterclockwise by angle around p, starting with the direction of r^+ , leading to a sequence p_1, \ldots, p_n . p_1, \ldots, p_n together with p forms a starshaped polygon P with p in the kernel. We want to determine whether this polygon contains an empty convex 6-gon involving p. To this end we compute the visibility graph inside P, ignoring all edges involving p. For this we can use a slight adaptation of the approach by Dobkin et al. [2], which takes $O(n^2)$ time.

Finding an empty convex 6-gon involving p is equivalent to finding a convex chain of four edges in the visibility graph, whose total angle around p is less than π . To this end we walk counterclockwise along the vertices of the polygon. For each edge we determine the "best" convex chain of length 3 and 2 that ends on this edge. By "best" we mean: that starts the latest. This means that the chain makes the smallest angle with p. For an edge e we use C_e^3 to denote the starting point of the best chain of length 3 and C_e^2 to denote the starting point of edge e but we do not actually have to store it. When we reach a point p_i we assume that for all incoming edges the C^3 and C^2 values are known (if they exist).

Note that both the incoming edges and outgoing edges of p_i are sorted by angle (that is the way the visibility graph algorithm produces them). We now determine which

outgoing edges form a convex angle with what incoming edges. We start with the first outgoing edge e, which can be connected to the smallest number of incoming edges, and determine all incoming edges with which it forms a convex angle. Of these we determine the best C^3 , C^2 and C^1 values. If C^3 exists and together with e and the point p forms an empty convex 6-gon we are done. Otherwise, we set C_e^3 to the best C^2 value and C_e^2 to the best C^1 value. We then continue with the next outgoing edge. We do not have to recheck all the incoming edges because we already know the best values of C^3 , C^2 and C^1 among them. We only have to check untreated incoming edges that form a convex angle with the outgoing edge. In this way we proceed until we either find a convex 6-gon or reach the last vertex. It is easy to see that the algorithm takes $O(n^2)$ time.

In the implementation of the method, the construction of the visibility graph and the checking of convex chains are interleaved. So for every edge of the visibility graph that is created, the chain information is immediately computed. As a result, the program can stop as soon as it finds a chain of length 4, saving a lot of work because most of the time an empty convex 6-gon does indeed exist.

The algorithm was implemented in Delphi. A careful look at the algorithm reveals that the only geometric operation required is to determine whether a sequence of three points makes a left turn, a right turn or is straight. This can be implemented by comparing two numbers created using two additions and a multiplication. As point coordinates we use 30 bit positive integers. As a result 62 bits of precision is required to do the calculation exact. Using 64 bit integers for the operation guarantees that the algorithm is robust. The algorithm is also very fast. For example, on a Pentium III 500 MHz the program can check about 15,000 points per second in a set of 29 points.

3. The Result

Just testing random sets of points never resulted in a set of more that 18 points. (A similar observation was made in [1].) Using an incremental approach, adding points in a slowly growing region increased this number to 22. (The reason for slowly growing the region is that any point set without an empty convex 6-gon can be created in an incremental way by starting with the innermost point and each time adding a point outside the convex hull of the current set.)

To obtain larger sets we used a form of random backtracking. Here we first add random points in a growing region. If we cannot add a new point without introducing an empty 6-gon, that is, if 10,000 tries fail, we randomly remove a number of points (four to six) without introducing an empty convex 6-gon, and start again adding points. If the newly obtained set is equal to or larger than the previous set we continue with this new set. Otherwise we reuse the previous set, removing other points.

To make sure that the sets the algorithm finds are maximal (up to precision issues) we compute the arrangement of lines formed by all pairs of points in the set. When we test a point in each cell of this arrangement we can establish for certain that no point can be added. Unfortunately, it is impossible to compute all cells efficiently with the bounded precision we are dealing with. So instead we compute the midpoints of all edges of the arrangement and test points at the four rounded values around these. This will test all cells that are large enough with respect to our choosen precision. As precision seems not



Fig. 1. A set of 29 points without an empty convex 6-gons.

to be the bottleneck (see below) this is a reasonable indication that the set can indeed not be extended anymore.

The final technique we used was to apply Brownian motion. Given a large set we randomly move the points a small random distance in a random direction without introducing an empty convex 6-gon. The idea is that this will open up a "hole" in which an additional point can be placed. Next we test the arrangement as indicated above to see whether a point can be added. This process is then repeated.

After some tuning we just let the program run. After about 4 days it reported a set of 29 points without an empty convex 6-gon, leading to the following result:

Theorem 3.1. $F_6 \ge 30$.

Proof. See Figure 1 for a set of 29 points that does not contain an empty convex 6-gon.

Since this set was found the program has run for months. When using the correct settings, it reports sets of 29 points every few days but it never found a larger set. Our first thought was that this might be related to the bounded resolution of 30 bit integers that we used for input coordinates. To see whether precision played a crucial role, we added an algorithm that tries to reduce the precision needed. This algorithm first scales the point set down, making sure that no three points end up on a line and that the set still has no empty convex 6-gon. Once this fails we again use some Brownian motion, with a tendency to move the points towards the center of the set. Letting this run on some of our examples of 29 points, we managed to get all coordinates in the range 0–1300. So

11 bits of precision turn out to be enough. This led us to believe that precision was not the reason for not finding larger sets. It seems that either the answer is the correct upper bound, or that the chance of finding a 30 point set is very much smaller than the chance of finding a 29 point set.

Anybody who is interested in running the program on their machine can download it from the website

http://www.cs.uu.nl/people/markov/sixgon/

If you want to check or adapt the source, please contact the author.

4. Some Observations

Obviously, the interesting open problem remains whether 29 is the correct bound or whether there is a bound at all. The fact that we did not manage to find a larger set is no proof. Based on our experiments with the program we did though make some observations that might be useful in obtaining a proof.

Looking at the different solutions with 29 points we observed that they all have a similar structure when we consider the convex layers. They all have six convex layers. The outer layer (the convex hull) consists of three points. The next layer consists of four points. Then there are three layers of seven points each, and finally there is a single point in the center. See Fig. 2 for the convex layers in our set of 29 points.

This first led us to the conjecture that a set without empty convex 6-gons can have at most seven points on the convex hull. We added a test for this in the software. Unfortunately the conjecture was wrong. After testing millions of sets a set of 19 points



Fig. 2. The convex layers for the set of 29 points without an empty convex 6-gons.

was discovered that has eight points on the convex hull. No larger convex hull was ever discovered and the set found can be easily extended to a larger set with less points on the convex hull. So it might be promising to try and prove a bound on the size of the convex hull. This would also imply a bound on the size of each convex layer, because peeling off a layer cannot introduce empty convex 6-gons.

Secondly, we conjectured that the number of convex layers was bounded by six. Again we added a test and again it turned out that the conjecture was wrong. After many tests we found a set of 28 points with seven convex layers. However, no set was found with more layers. So it might be promising to try and prove a bound on the number of convex layers. Together with the previous conjecture this would immediately lead to a bound on F_6 .

Acknowledgments

I thank Berta Scholten and Ingrid Vincent who, back in 1988, implemented a first version of the incremental algorithm.

References

- Avis, D., and D. Rappaport, Computing the largest empty convex subset of a set of points, *Proc. First ACM Symp. on Computational Geometry*, Baltimore, MD, 1985, pp. 161–167.
- Dobkin, D.P., H. Edelsbrunner and M.H. Overmars, Searching for empty convex polygons, *Algorithmica*, 5 (1990), 561–571.
- Erdös, P., Combinatorial Problems in Geometry and Number Theory, Proc. Symp. Pure Math., volume 34, American Mathematical Society, Providence, RI, 1979, pp. 149–162.
- 4. Harborth, H., Konvex Funfecke in ebene Punkmengen, Elem. Math. 33 (1978), 116-118.
- 5. Horton, J.D., Sets with no empty convex 7-gons, Canad. Math. Bull. 26 (1983), 482-484.
- Morris, W., and V. Soltan, The Erdös–Szekeres problem on points in convex position a survey, Bull. Amer. Math. Soc. 37 (2000), 437–458.
- Overmars, M.H., B. Scholten and I. Vincent, Sets without empty convex 6-gons, *Bull. EATCS* 7 (1989), 160–168. A copy can be found on the website http://www.cs.uu.nl/people/markov/sixgon/.

Received July 9, 2001, and in revised form February 19, 2002. Online publication November 14, 2002.

158