Open Problems from CCCG 2004

Erik D. Demaine[*] Joseph O'Rourke[†]

The following is a list of the problems presented on August 9, 2004 at the open-problem session of the 16th Canadian Conference on Computational Geometry held in Montréal, Québec, Canada.

Minimum-Bend Graph Drawing in Other Grids
Therese Biedl
U. Waterloo
biedl@uwaterloo.ca

What is the best worst-case guarantee one can make on the number of bends required to embed a graph into a particular grid, in particular, the hexagonal and octagonal grids? A *grid point-drawing* of a graph maps each vertex to a unique point of the grid, and maps each edge to a path in the grid connecting the endpoints; the paths are allowed to intersect at common endpoints and at proper crossings (points at which two or more paths meet but do not bend), but must be edge-disjoint. Such a drawing exists only for graphs whose maximum vertex degree is at most the maximum degree Δ of a point in the grid. The goal of this problem is to obtain a bound of the form "every graph on n vertices of maximum vertex degree Δ has a grid point-drawing with at most $f(n)$ bends" for various grids.

In the usual square grid, 2D points have integral coordinates and an edge connects every two points at Euclidean distance exactly 1, so $\Delta = 4$. Here it is known that every graph with maximum vertex degree at most $\Delta = 4$ has a square-grid point-drawing with at most $2n + 4$ bends [BK98, LMS98].

The "8-way square grid" has the same set of points but connects two points at Euclidean distance either 1 or $\sqrt{2}$—that is, it draws two diagonals within each square cell—so $\Delta = 8$. In this case, Biedl can prove an upper bound of $6n+O(1)$ bounds, and the proof is not too difficult. Is there a better bound?

The three tilings of the plane by identical regular polygons are the square grid, the triangular grid ($\Delta = 6$), and the hexagonal grid ($\Delta = 3$). For the triangular grid, Biedl can prove an upper bound of $4n + O(1)$ bends by a relatively easy proof, and an upper bound of $3.5\,n + O(1)$ bounds by a rather complicated proof. Is there a better bound, or a simpler proof of the latter bound? What about the hexagonal grid?

References

[BK98] Therese Biedl and Goos Kant. A better heuristic for orthogonal graph drawings. *Comput. Geom. Theory Appl.*, 9:159–180, 1998.

[LMS98] Yanpei Liu, Aurora Morgana, and Bruno Simeone. A linear algorithm for 2-bend embeddings of planar graphs in the two-dimensional grid. *Discrete Applied Math.*, 81(1–3):69–91, January 1998.

Cutting and Joining Matrices
John Iacono
Polytechnic U.
iacono@john.poly.edu

How quickly can one maintain a collection of rectangular matrices subject to the following operations?

Make: Create a new 1×1 matrix with entry 0.

Get: Return $A[i, j]$ for specified matrix A and indices i and j.

Set: Set $A[i, j]$ to a specified value x, for specified matrix A and indices i and j.

Horizontal Cut: Cut specified $m \times n$ matrix A below specified row i, resulting in one $i \times n$ piece and one $(m - i) \times n$ piece.

Vertical Cut: Cut specified $m \times n$ matrix A right of specified column j, resulting in one $m \times j$ piece and one $m \times (n - j)$ piece.

Horizontal Join: Join the bottom edge of specified $m_1 \times n$ matrix A_1 to the top edge of specified $m_2 \times n$ matrix A_2, resulting in one $(m_1 + m_2) \times n$ matrix.

Vertical Join: Join the right edge of specified $m \times n_1$ matrix A_1 to the left edge of specified $m \times n_2$ matrix A_2, resulting in one $m \times (n_1 + n_2)$ matrix.

[*]MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar Street, Cambridge, MA 02139, USA, edemaine@mit.edu
[†]Department of Computer Science, Smith College, Northampton, MA 01063, USA. orourke@cs.smith.edu. Supported by NSF Distinguished Teaching Scholars award DUE-0123154.

At the session, an $O(\sqrt{n})$ upper bound was claimed, but such a bound no longer seems easy to obtain (personal communication with Ilya Baran, John Iacono, Stefan Langerman, and others). Is an $O(\sqrt{n})$ bound, a $o(\sqrt{n})$ bound, or even a a polylogarithmic bound, possible?

The analogous problem in 1D is solvable in $O(\log n)$ time per operation, by a splittable concatenable balanced search tree such as red-black trees, and this time bound is the best possible [PD].

References

[PD] Mihai Pătraşcu and Erik D. Demaine. Logarithmic lower bounds in the cell-probe model. *SIAM Journal on Computing*. To appear.

Whitespace Management

Satyan Devadoss
Williams College
satyan.devadoss@williams.edu

Informally, we are given a set of objects on a map, and we seek a method of continuously enlarging the objects over time so that these objects are still visible as the map continuously zooms out. A simple scaling of the objects is not a feasible solution because it can lead to intersections of the objects, making the map illegible. In this case, the objects need to be moved away from each other as we enlarge them in such a way as to preserve *topology*. Over time, the objects will use up the surrounding *whitespace* until they fill the entire map. It is clear that the *shape* of the objects will radically deform over time as they begin to fill the whitespace.

More formally, let $\mathcal{R} = \bigcup R_i$ where the R_i's are pairwise disjoint, pairwise ε-separated, connected regions inside a bounding region $M \subset \mathbb{R}^k$. Define two regions as *adjacent* if their Voronoi regions are adjacent. Construct a homotopy $f_t : \mathcal{R} \to M$, for $t \in [0, 1]$, having the following properties:

1. f_0 is the inclusion map (i.e., the identity map on each R_i) and f_t is a homeomorphism onto its image for all t.
2. f_t is area-ratio preserving for all t.
3. $f_t(R_i)$ are pairwise ε-separated for all t.
4. f_t preserves relative position for all t (i.e., the adjacency graph does not change over t).
5. For all points x on the boundary of region R_i, $f_1(x)$ is at distance exactly ε from $f_1(y)$ for some point y on the boundary of either M or another region R_j, $j \neq i$.

Constrained Higher-Order Delaunay Triangulations

Marc van Kreveld
Utrecht U.
marc@cs.uu.nl

What is the complexity of finding the lowest-order triangulation of a planar point set S that includes a given set of edges? The *order of a triangulation* is the maximum order of any triangle; the *order of a triangle* is the number of points from S inside the circumcircle. Thus a triangulation has order 0 if and only if it is the Delaunay triangulation. This problem captures trying to find the “most Delaunay” triangulation when constrained to include a specified set of edges in the triangulation. This class of Delaunay-like triangulations is introduced in [GHK02] as *higher-order Delaunay triangulations*.

It is also interesting to consider worst-case bounds in terms of the order of the original set of edges. Here the *order of an edge set* is the maximum order of any edge, and the *order of a line segment* is the minimum number of points contained in a circle passing through the two endpoints. van Kreveld can prove that, given a single constraint edge of order k, there is always a constrained triangulation of order at most $2k - 2$, and that this is the best possible such bound in the worst case. However, given more than one constraint edge, the problem is unsolved. van Kreveld can prove that the minimum order of a triangulation constrained by two edges is not always the maximum of the minimum order for each constraint edge individually.

References

[GHK02] Joachim Gudmundsson, Mikael Hammar, and Marc van Kreveld. Higher order Delaunay triangulations. In *Proc. 8th Ann. European Sympos. Algorithms*, 2000, vol. 1879 of Lecture Notes in Computer Science, pp. 232–243.

Monochromatic Division by Parallel Lines

Carlos Seara
U. Politècnica de Catalunya
carlos.seara@upc.edu

Given a set R of red points and a set B of blue points in the plane, what is the complexity of finding the minimum number of parallel lines to separate the plane into monochromatic regions (strips)? The goal is to minimize the worst-case complexity of an algorithm in terms of the total number of points, $n = |R| + |B|$.

A simple $O(n^3 \log n)$-time algorithm considers each of the $\binom{n}{2}$ slopes of interest (parallel to the line

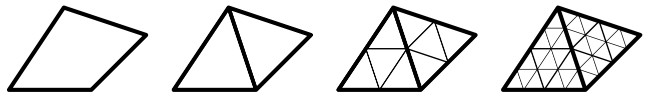

Figure 1: 1-to-4 refinement.

through two input points) and greedily constructs the optimal separating set of lines with that slope in $O(n \log n)$ by sorting the points in the perpendicular direction. The same idea can be improved to an $O(n^2 \log n)$-time algorithm by transforming the problem into the dual according to the standard mapping $(a, b) \leftrightarrow ax + b = 1$. The dual problem is the following: Given a set of nonvertical red lines and blue lines in the plane, find a set of points on a vertical line that stabs the wedge enclosed by every pair of red and blue lines. This problem can be solved in $O(n^2 \log n)$ time by constructing the arrangement of lines and running a line sweep over the arrangement.

Is there an algorithm with running time $o(n^2 \log n)$? What if the minimum number k of separating lines (the size of the output) is known to be small? Is there an $O(nk \log n)$-time algorithm? For $k \leq 2$, an $O(n \log n)$-time algorithm is presented in [HNRS01]. Seara further claims an $O(n \log n)$-time algorithm for $k \leq 4$. What about larger k?

References

[HNRS01] Ferran Hurtado, Marc Noy, Pedro A. Ramos, and Carlos Seara. Separating objects in the plane by wedges and strips. *Discrete Applied Math.* 109(1–2):109–138, Apr. 2001.

Edge-Unfolding via Refinement

Joseph O'Rourke
Smith College
orourke@cs.smith.edu

How many times must a convex polyhedron's surface be refined before it becomes edge-unfoldable? A polyhedron is *edge-unfoldable* if there is a set of edges that can be cut such that the result is connected and can be isometrically flattened into the plane with no overlap. Any successive, regular refinement process is of interest. One particular process, shown in Figure 1, is to first triangulate the surface, and then at each successive refinement level, partition each triangle into four subtriangles by dividing at the midpoint of each edge (*1-to-4 refinement*).

The goal is to obtain a worst-case upper bound, perhaps dependent on the combinatorial complexity n of the polyhedron (the total number of vertices, edges, and faces). Interesting possible answers are 0, 1, $O(1)$, $o(n)$, $O(n)$, $n^{O(1)}$, $2^{O(n)}$, and ∞. An upper bound of 0 (or a lower bound of 1) would solve a much harder problem, edge-unfolding of convex polyhedra or triangulated convex polyhedra, which has been unsolved for hundreds of years [TOPP9]: an upper bound of 0 would mean that no refinement is needed for any convex polyhedron, and a lower bound of 1 would mean that some convex polyhedron needs refinement before it is edge-unfoldable. Obtaining an upper bound of more than 0 is potentially easier than the classic problem, by giving the flexibility of a (limited amount of) refinement.

Without any restriction on the refinement, it is known that adding $O(n^2)$ additional edges can make any convex polyhedron edge-unfoldable, via the standard source unfolding or star unfolding (see [DO05]). Refinement would permit approximating these unfoldings. So the problem can be viewed as asking whether some finite approximation suffices to avoid overlap.

References

[DO05] Erik D. Demaine and J. O'Rourke. A survey of folding and unfolding in computational geometry. In J. E. Goodman, J. Pach, and E. Welzl, editors, *Combinatorial and Computational Geometry.* Cambridge University Press, 2005.

[TOPP9] Erik D. Demaine, Joseph S. B. Mitchell, and Joseph O'Rourke, editors. Edge-unfolding convex polyhedra. Problem 9 of *The Open Problems Project.* http://cs.smith.edu/~orourke/TOPP/P9.html.

Wrapping a Box with a Rectangle

Robert Dawson
Saint Mary's U.
rdawson@husky1.stmarys.ca

Given a rectangle of paper and a box, can the box be wrapped by folding the rectangle? There are only five real inputs: the dimensions of the rectangle and of the box. Characterize when this problem has a solution. There are examples where an "axis-parallel wrapping" (initially placing the rectangle with sides parallel to the box) is impossible, but rotating the rectangle relative to the box enables a wrapping. Another form of the problem from the computational origami literature is as follows: given the dimensions of a rectangle and a box, what is the smallest scaling of the rectangle that can wrap the box?

Although this decision problem captures the essence, of more practical interest is some form of optimization. One suggestion of Dawson is to find the smallest-perimeter rectangle that wraps a box with specified dimensions. (The smallest-area rectangle is ill-defined: the area can be arbitrarily close to the surface area of the polyhedron [DDM00].)

References

[DDM00] Erik D. Demaine, Martin L. Demaine, and Joseph S. B. Mitchell. Folding flat silhouettes and wrapping polyhedral packages: New results in computational origami. *Comput. Geom. Theory Appl.*, 16(1):3–21, 2000.

Moving Coins

David Rappaport
Queen's U.
daver@cs.queensu.ca

What is the minimum possible number of straight-line translations without self-intersection to rearrange n unlabeled equal-radius disks from one configuration to another? Here one disk is moved at a time, by translating along a straight line segment, and throughout each such move, the set of coins must be interior-disjoint. The goal is to find the best worst-case bound in terms of n, over all pairs of configurations of n coins (in which the coins are interior-disjoint).

An easy upper bound is $2n - 1$ [AHORT04]. First, order the coins according to their projection onto a generic axis. Second, move each coin in this order, except the last, far in that direction (and so that the first is substantially farther than the second, and so on). Third, move the last coin to the target position with minimum coordinate along the axis. Fourth, move each of the other coins, in reverse order, to target positions of increasing coordinate along the axis.

On the other hand, a $3n/2$ lower bound is known [AHORT04]. What is the correct bound?

References

[AHORT04] Manuel Abellanas, Ferran Hurtado, Alfredo Garcia Olaverri, David Rappaport, and Javier Tejel. Moving coins. In *Proc. Japan Conf. Discrete Comput. Geom.*, Tokyo, October 2004.