Portfolio

English exam

MCS Ph. D. Programme

Boris Zolotov

ABSTRACT 1

Edge-vertex resolution of a planar drawing is an important measure that reflects a minimal separation between a vertex and an edge, allowing the vertices to be objects of non-zero size rather than points.

The authors present an algorithm that for any planar 3-connected graph computes its drawing with the edge-vertex resolution equal to 1/2 (that is, a disk-link drawing).

What is more, the drawing is computed in time linear in the number of vertices of the graph, and the vertices of the drawing are located on a linear-size grid (which is proved to be optimal).

The algorithm is based on the earlier result of Chrobak and Kant who had a similar problem but did not obey the requirement for the drawing to be disc-link.

ABSTRACT 2

Unit Disc Cover (UDC) problem asks for the minimal number of unit disks covering a given set of points and the placements of these circles.

The problem is proved to be NP-hard; however, there exist plenty of polynomial and pseudopolynomial approximation algorithms solving it.

The authors implement several known algorithms for UDC and compare their efficiency. They also present a new 7-approximation algorithm running in linear time on average and two heuristics that help speed it up.

The algorithm devised by the authors covers the query set with circles aligned to the sqrt(2)-sized grid. Then it is proved that the algorithm outputs not more than seven times the number of circles contained in the optimal answer.

The authors conclude that the algorithm by Liu and Lu (2014) is optimal for practical applications, yet their algorithm overperforms it in certain cases.

ABSTRACT 3

The authors present an algorithm that enumerates and classifies all edge-to-edge gluings of unit squares that correspond to convex polyhedra.

The authors show that the number of such gluings of n squares is polynomial in n, and the algorithm runs in time polynomial in n (pseudopolynomial if n is considered the only input).

Their technique can be applied in several similar settings, including gluings of regular hexagons and triangles.

1. Penumbra of aggravations — countless annoying little problems related to a big problem

2. Coherent argument — a well-organized reasoning without any self-contradictions

3. Substantive contribution — an important piece of study, essential for the field

4. Diffuse activity — the skill in which is spread between researchers by itself

5. Credible resources — ones to which respect is paid

6. Prowess in a discipline — expertise

7. Skills inculcated — taught thoroughly

8. By-product approach — developing something not as a main result of a process

9. Doctoral instruction — the process of writing thesis and receiving knowledge under supervision

10. Humanities and social sciences — гуманитарные науки

11. Set out to write a book — decide to do something huge

12. Grapple with — deal or understand a difficult problem

13. Shelf-bending thesis — one that is never published

14. Apprentice researcher — working for low payment for a more skilled person

15. Dubious kudos — praise that is not certainly there

16. Taught Ph. D

17. Classical Ph. D

18. Professional doctorate

19. Leveled planar graph — a graph drawn on the plane whose vertices are on several parallel lines

20. Layered pathwidth — the maximum size of the intersection between a layer and a bag in a decomposition of a graph

20. Outerplanar graph — a planar graph drawn in a way such that all the vertices are in the outer (unbounded) face

21. Squaregraphs — 4-faces, each vertex either in outer face or has degree 4+

22. Breadth first layering — each vertex is marked by its distance to the source vertex

23. Specify an ordering — assign consequent natural (positive integer) numbers to objects

24. Net — a simple non-overlapping unfolding of a polyhedron

25. Unit-speed parametrization — parametrization by the distance to the origin

26. Osculating circle — the one with the same curvature

27. Unit normal vector — perpendicular, of length 1

28. Principal curvatures — the highest and the lowest cuvatures of a surface at a point

29. A flaw of sorts — same things having different properties

30. Subtended — bounded, contracted

31. Intrinsic — defined from an internal view

32. Geodesic — measured inside surface

33. Elliptic, hyperbolic points — with respect to curvature

34. Angle deficit, Gaussian curvature — the difference between 2*pi and the facial angle at a point

35. Meager evidence — sparse, insufficient

36. Lament — to complain

37. Paucity — sparcity, poverty

38. Star unfolding, source unfolding — methods of unfolding a polyhedron that (supposedly) produce a non-overlapping net

39. Overlap penetration — the ratio of a penetration to the diameter

40. Follow suit — to act conformly

41. Convex hull — a minimal convex polygon containing a given set of points

42. Vertex-edge resolution — a minimal separation between a vertex and a non-incident edge in a drawing

43. Disk-link drawing — a drawing with vertex-edge resolution of at least 1/2

44. Gluing — a pair of a set of polygons and rules specifying how to glue their boundaries

45. Polyhedron — a union of convex polyhedra, which are intersections of half-spaces

46. UDC Problem — a problem of covering a given set of points with the minimal number of unit disks

47. Erdős–Rényi graph — a randomly generated graph with sufficient connectivity

48. Prisoner's dilemma — a noncooperative game about betraying or keeping silent

49. Hawk-dove game — a noncooperative game about showing off or retreating

50. MAXQ — a method of reinforcement learning that takes the next action depending on the output of max-nodes and q-nodes

51. Schnyder labeling — an orientation on a maximal planar graph where each vertex has 3 outcoming edges

52. Canonical ordering of a planar graph — an ordering that helps draw the graph: each next vertex is in the outer face connected with a countinuous segment of previous vertices

53. LR-partition — a partition of all the back-edges with respect to a DFS-ordering into two sets. It helps test planarity of a graph

54. n-connected graph — a graph that remains connected after removal of n-1 vertices

55. Closed rectangle-of-influence drawing — a drawing such that no vertex lies in the axis-parallel rectangle defined by two ends of every edge

56. Bold drawing — vertices are drawn as disks of radius r and edges as rectangles of width w

57. Planar embedding — an equivalence class of topologically-equivalent planar drawings

58. Straight-line drawing — a drawing where all the arcs are straight-line segments

59. Grid drawing — straight-line drawing where vertices have integer coordinates

60. Point density — ratio of |P| to the area of its bounding box

61. Track layout — partition of a graph into tracks, which are independent sets, and edges between them do not cross

62. Tree decomposition — an assignment of bags such that endpoints of any edge share a bag, and each vertex is in a continuous subtree of bags

63. Path decomposition — a tree decomposition but the tree is a path

64. Treewidth — minimal size of a maximal bag -1

65. Minor — a graph obtained by deleting edges / vertices and contracting edges

66. Euler characteristic — V-E+F

67. Euler genus — 1 - chi/2

69. General position — assumptions about data that hold almost often and help evade degenerate cases

70. Hull signature — an ordered list of vertices of the convex hull of a set

71. Dynamic nearest neighbor structure — a structure supporting insertion of a site and the query of the nearest site to a given point

72. Voronoi circle — a circle passing through three adjacent Voronoi sites

73. Sweep line — a line that is moved through the plane used to construct several geometric structures

74. Combinatorial change — a change in the graph of a structure, not only in the location of its vertices

75. Affirm — show that one has obtained a message and has no objections

76. Visibility center — a point such that the maximum geodesic distance from it to _see_ any point in the set is minimized

77. Book embedding — a drawing of a graph on several half-planes glued along their boundary

78. Abuse of notation — using the same denotion for objects of different generality

79. Winding number — the number of counterclockwise rotations of a curve around a point

80. Young diagram — a way to demonstrate integer partitions

81. Bipartite graph — a graph colored in black and white in a way that each edge has endpoints of distinct colors

82. Spanning tree — a tree obtained by deletion of several edges from a graph, but on the same set of vertices

83. Building greedily — taking a maximal possible set on each step

84. Minor-closed class — a class such that if G is in it, every minor of G is in it

85. Dual graph — a graph whose vertices are faces of G and whose edges are edges incident to these faces

86. Geodesic center — a point that minimizes the maximum geodesic distance from it to any point of the set

87. Chord — a segment inside a polygon intersecting its boundary only at endpoints

88. Geodesically convex subset — a subset such that if two points lie in it, so does the geodesic shortest path between them

89. Pareto optimality — a situation where no individual or preference criterion can be better off without making at least one individual or preference criterion worse off

90. Nash equilibrium — each player is assumed to know the equilibrium strategies of the other players, and no one has anything to gain by changing only one's own strategy

91. Oracle — a thing that is somehow able to solve a complicated problem in constant time

92. Winning strategy — a sequence of actions that depends on a position in a game that makes a player win the game

93. DFS — graph traversal that first goes as deep as possible along the edges

94. BFS — graph traversal that considers all the vertice's neighbors immediately after this vertex

95. Manifold — a topological space with continuous base that is locally homeomorphic to Euclidean space

96. Base — a set of sets such that any open set of the topology can be represented as a union of the sets in the base

97. Curry—Howard isomorphism — a mapping between habitable types and provable statements

98. Unit-disk graph — a graph for a set of vertices in the plane that only has edges of unit length

99. Series-parallel graph — a graph with two terminal vertices that is formed by application of composition or parallel composition

100. Tree-apex graph — a graph that is a tree with a vertex added that is connected to each leaf

101. Apex graph — a graph that can be made planar by removing a single vertex

102. Dent — an unplanned piece of curvature on a metal sheet left behind by a miscalculated impact

103. Primary root — a generator of the multiplication group of residuies modulo p (which is, in fact, cyclic)

104. Quadratic residue — a residue congruent to the square of an integer number modulo n

105. Euclidean algorithm — an algorithm to find the greatest common divisor of a pair of numbers or polynomials that does not rely on the prime decomposition

106. Determinant — a number assigned to a matrix that behaves in a certain way under addition or permutation of rows

107. Permutation — a bijection between a finite set and itself

108. Sign — a homomorphism from a group to {-1,1}

109. Homomorphism — a linear map, i. e. a map preserving the structure

110. Arrangement — the subdivision of the plane into cells, edges and vertices induced by a given set of lines

111. Heavy path decomposition — a decomposition of a tree into paths that is used to implement link-cut tree

112. Splay tree — a tree where each vertex is raised to the root before any operation is carried out on it

113. B-tree — a balanced search tree with a varying number of descendants of each vertex

114. AVL-tree — the default balanced binary search tree that demands rotations at certain vertices in order to stay balanced

115. Range searching — asking to return all pieces of data in a given interval.

116. Output-sensitive algorithm — an algorithm such that one has to take into consideration its output to estimate its running time.

117. Kd-tree — a tree used for range searching in which adjacent nodes split the data according to the dimensions of the space that are alternating

118. Amortised — evaluated on average during large number of consecutive queries

119. Sublinear — estimated by o(n)

120. Median — a number in a set such that exactly a half of the set's numbers are smaller than it

121. Recurrence — a way to introduce a function such that the value f(n) is calculated using values on smaller arguments

122. Canonical subset of v — the subset of points stored in the leaves of the subtree rooted at a node v

123. Fractional cascading — a technique to improve the running time of a range searching tree

124. Squaring search — a way to find an upper bound for the answer by considering 2^2^k at each step. This way logarithms of the query double at each step.

125. Kuratowski's problem — a problem of determining how many different sets one can obtain using Cl and Int operations

126. Composite-number space — a space in which pairs of points are sorted lexicographically

127. Point location query — given a map and a query point specified by its coordinates, find the region of the map containing it

128. Planar subdivision — a structure of cells, edges and vertices covering the plane.

129. Vertical slab — a stripe inside which there are no vertices

130. Trapesoidal map — a subdivision whose cells are trapezoids with vertical bases or triangles

131. Incremental algorithm — an algorithm constructing a structure adding one site after another, rather than considering all sites at once. It is usually slower than an offline algorithm.

132. Topology — a collection of sets that are to be called open. It satisfies certain properties: any union of open sets is an open set.

133. Advisory board — a group of people making decisions.

134. Voronoi assignment model — the model where every point is assigned to the nearest site

135. Cartesian coordinates — a system assigning a pair of numbers to each point in the plane

136. Metric — a symmetric function that is zero on coinciding elements and satisfies the triangle inequality

137. Metric space — a set equipped with a metric

138. Triangulation — a decomposition of an object into a set of triangles or simplices

139. Face — a polygon of the highest possible dimension that belongs to a graph / to a surface

140. Edge — an intersection of two faces

142. Incident — two objects of different priorities next to each other

143. Adjacent — two objects of equal priorities next to each other

144. Quadrilateral — a polygon with four sides

145. Tight bound — a bound that is reachable and thus can not be improved

146. Iterative — a process that is repeated several times

148. Nearest neighbor — a site such that the distance from it to the query point is minimal

149. Residue — a number between 0 and n-1 that is congruent to the given number modulo n

150. Power series — a formal sum of powers of a variable with various coefficients

151. Congruence — any sensible equivalence relation

152. Convex domain — a domain such that any segment connecting two its points lies inside it

153. Connected space — a space that cannot be represented as a union of two open sets

154. Path connected space — a space where there exists a path between any two points

155. Simply connected space — a space where every loop can be contracted

156. A loop — a path from a point to itself

157. Primality test — an algorithm testing if a given number is prime

158. Turing machine — a model with a finite number of states, an infinite tape and a set of rules describing transition between states

159. Cellular automaton — a set of cells where the state of a cell changes at each step in accordance with the states of its neighbors

160. Markov's algorithm — a Turing-complete model comprised only of replacement rules

161. Winning position — a position from which a turn can be taken into a losing position

163. Losing position — a position from which any turn leads to a winning position

164. Voronoi diagram — a subdivision of the plane according to the nearest neighbor queries.

165. Flarb — a combinatorial operation of adding a new face to a graph.

166. Tree — a connected graph with no cycles.

167. Pendant vertex — a vertex of degree 1.

168. Euler cycle — a cycle that visits each edge of a graph exactly once.

169. Hamiltonian cycle — a cycle that visits each vertex of a graph exactly once.

170. Search — a problem of outputting the piece of data assigned to a given key

171. Sorting — a problem of placing several numbers or keys in the ascending order.

172. RAM machine — a model that can access any memory cell in constant time

173. Forest — a possibly disconnected graph with no cycles.

174. Dual graph — a graph whose vertices are faces of G and whose edges are edges incident to these faces

175. Language — a set of strings that one usually wants to be able to distinguish

176. Computational hierarchy — a separation of problems into levels according to the number of quantifiers preceding a computable problem

177. Motion planning — a problem of assigning a strategy to several moving robots such that no collisions happen and total traveled distance is minimal

178. Geodesic center — a point of the set such that maximal geodesic distance from it to a point of the set is minimised

179. Noncooperative game — a game where players take their turns simultaneously, and the prize is split immediately afterwards

180. Network — a graph with a large number of vertices and a set of rules that apply to its connectivity

181. PCB — a printed contact board that is used for connecting several electronic devices. It is comprised of several sheets on which tracks are laid

182. Delaunay triangulation — a triangulation in the plane such that the circumcircle of any triangle does not contain any vertices but of this triangle

183. Euclidean shortest path problem — a problem of connecting two points in a Euclidean space (with polyhedral obstacles) by a shortest path

184. Mesh — a subdivision of an area into cells of simple shapes

185. Ray tracing problem — given a set of objects in space, produce a data structure that efficiently outputs which object a query ray intersects first

186. Flipping — an operation replacing one diagonal of a quadrilateral in a triangulation with the other diagonal

187. Divide-and-conquer algorithm — an algorithm whose initial step is splitting the query set into two roughly equal parts

188. Farthest-first traversal — a sequence of points where the first point is selected arbitrarily and each successive point is as far as possible from the set of previously-selected points

189. Prefix — a part of the sequence from the beginning to a certain point

190. Suffix — a part of the sequence from a certain point to the end

191. Delaunay refinement — adding vertices to a triangulation in a way such that the triangulation obeys certain quality requirements afterwards.

192. Pigeonhole principle — a statement concerning not being able to fit n+1 pigeons into n holes

193. Osculating circle — the circle passing through the point of a curve that has the same curvature as the curve at this point

194. Source vertex — a vertex in an oriented graph with no incoming edges

195. Sink vertex — a vertex in an oriented graph with no outcoming edges

196. Triangle strip — a sequence of triangles where each next triangle shares an edge with the previous one

197. Subdivision of a graph — a graph obtained from the given one by adding vertices of degree 2 in the interior of the edges

198. Regular graph — a graph in which all the vertices have the same degree

199. Flow on a graph — an assignment of a number and an orientation to each edge of the graph such that the sum of the numbers at each vertex is zero

200. Hashing — assigning certain numbers to strings in such a way that it becomes easy to check if two long strings differ

Algorithms for Incremental Voronoi Diagrams

ZOLOTOV Boris

June 3, 2022

1 Introduction

Computational geometry studies computations on discrete geometric objects such as arrangements, diagrams, foldings and drawings. In particular one of its main interests is to design algorithms to construct such objects and data structures that store them efficiently.

The most crucial and defining concept applied in all those constructions is *distance:* whatever is studied, there is always some underlying metric that describes the object in question and defines its properties. It can be just the Euclidean metric, or some polyhedral metric, or any other metric that corresponds to a surface or a folding of a polygon.

We will be considering three different types of distance metrics. The Euclidean metric in $\mathbb{R}^2$ gives rise to the ordinary Voronoi diagrams, and we are interested in algorithms that allow for fast updates of a Voronoi diagram that is stored explicitly. We also consider polyhedral metrics of the spaces induced by gluing n congruent regular polygons edge to edge, and we propose a way to classify all such spaces in time polynomial in n. Finally, PCB-s are of our interest, which are unions of disjoint planes each equipped with a certain metric.

2 Literature Review

Even though there are many well-known algorithms that construct the Voronoi diagram for a given family of sites, the problem of making a Voronoi diagram dynamic, i. e. implementing changes to it when a new point site is inserted, has only drawn attention in the implicit case: maintaining implicit Voronoi diagram can be done in very little time. Maintaining a Voronoi diagram explicitly has not yet been considered.

The worst one can expect is that when a new site is inserted, the number of updates in a Voronoi diagram (i. e. vertices and edges that change) is linear. This can happen for every insertion if we consider an embedded diagram and store the coordinates of all the vertices.

The situation improves if we consider *the graph* of the Voronoi diagram that is subject to combinatorial changes: no coordinates matter, it is just deletion or addition of edges that is performed. In Allen et al. it was proved that when a new site is inserted to a Voronoi diagram, only $O(\sqrt{n})$ combinatorial changes happen to the graph of the diagram. This opened a possibility to find a sublinear algorithm that finds and implements combinatorial changes in an explicitly stored graph of the Voronoi diagram.

There is a known algorithm that is sublinear in n (which is the number of sites) that does exactly that. The algorithm has running time of $\tilde{O}(n^{3/4})$, which still does not achieve the $O(\sqrt{n})$ bound on the number of changes, that is why the quest for a better algorithm is still open.

Another type of metrics and distances that can be often seen in computational geometry is polyhedral metrics and geodesic distances. A question concerning them that has been standing for a long time already is the Alexandrov's problem of finding a convex polyhedron corresponding to a given polyhedral metric. There is almost no hope of solving this problem exactly for an arbitrary polyhedral metric, that is why several special cases are considered in literature.

2.1 Voronoi diagrams

We begin with standard definitions related to Voronoi diagrams and their basic properties. A detailed treatment of Voronoi diagrams and their applications can be found in. Let $S := \{s_1, s_2, \ldots, s_N\}$ be a set of N distinct points in $\mathbb{R}^2$; these points are called *sites*. Let $\mathrm{dist}(\cdot, \cdot)$ denote the Euclidean distance between two points in $\mathbb{R}^2$. We assume that the sites in S are in *general position*, that is, no four sites lie on a common circle.

Definition 1. The *Voronoi diagram* of S is the subdivision of $\mathbb{R}^2$ into N cells, called *Voronoi cells*, one cell for each site in S, such that a point q lies in the Voronoi cell of a site s_i if and only if $\mathrm{dist}(q, s_i) < \mathrm{dist}(q, s_j)$ for each $s_j \in S$ with $j \neq i$.

Let f_i denote the Voronoi cell of a site s_i. Edges of the Voronoi diagram, called *Voronoi edges,* are portions of bisectors between two sites which are the common boundary of the corresponding Voronoi cells. *Voronoi vertices* are points where at least three Voronoi cells meet. The *Voronoi circle* of a Voronoi vertex v is the circle passing through the sites whose cells are incident to v. Vertex v is the center of its Voronoi circle.

Since the sites are in the general position, each Voronoi vertex has degree three. Each Voronoi edge is either a segment or a ray and the graph of the Voronoi diagram formed by its edges and vertices is planar and connected.

2.1.1 Combinatorial Changes to the Voronoi Diagram and the Flarb Operation

We now overview the definitions and results from Allen et al. that we need to present our approach. In order to prove the $\Theta(N^{\frac{1}{2}})$ bound on the number of combinatorial changes caused by insertion of a site, a graph operation called *flarb* is introduced.

Let G be a planar 3-regular graph embedded in $\mathbb{R}^2$ without edge crossings (edges are not necessarily straight-line). Let $\mathcal{C}$ be a simple closed Jordan curve in $\mathbb{R}^2$.

Definition 2. Curve $\mathcal{C}$ is called *flarbable* for G if:

- the graph induced by vertices inside the interior of $\mathcal{C}$ is connected,
- $\mathcal{C}$ intersects each edge of G either at a single point or not at all,
- $\mathcal{C}$ passes through no vertex of G, and
- the intersection of $\mathcal{C}$ with each face of G is path-connected.

Given a graph G and a curve $\mathcal{C}$ flarbable for G, the *flarb* operation is, informally, removing part of G that is inside $\mathcal{C}$ and replacing it with $\mathcal{C}$. Formally, the flarb operation for G and $\mathcal{C}$ is defined as follows:

- For each edge $e_i \in G$ that intersects $\mathcal{C}$ let u_i be its vertex lying inside $\mathcal{C}$ and v_i its vertex outside $\mathcal{C}$. Create a new vertex $w_i = \mathcal{C} \cap e_i$ and connect it to v_i along e_i.
- Connect consecutive vertices w_i along $\mathcal{C}$.
- Delete all the vertices and edges inside $\mathcal{C}$.

Let $\mathcal{G}(G, \mathcal{C})$ denote the graph obtained by applying the flarb operation to graph G and curve $\mathcal{C}$.

Lemma 1. *The following holds for graph $\mathcal{G}(G, \mathcal{C})$:*

(a) $\mathcal{G}(G, \mathcal{C})$ has at most two more vertices than G does;
(b) $\mathcal{G}(G, \mathcal{C})$ is a 3-regular planar graph;
(c) $\mathcal{G}(G, \mathcal{C})$ has at most one more face than G does.

Proof. Items (a) and (b) have already been proved in literature. To prove (c) note that there is one new face bounded by the cycle added along $\mathcal{C}$ while performing the flarb. All the other faces of G are either deleted, left intact, or cropped by $\mathcal{C}$; these operations obviously do not increase the number of faces. □

Theorem 2. *Let G be a graph of the Voronoi diagram of a set of $N-1$ sites $s_1 \dots s_{N-1}$. For any new site s_N there exists a flarbable curve $\mathcal{C}$ such that the graph of the Voronoi diagram of sites $s_1 \dots s_N$ is $\mathcal{G}(G, \mathcal{C})$.*

2.1.2 Cost of the Flarb

We want to analyze the number of structural changes that a graph undergoes when we apply the flarb operation to it. There are two basic combinatorial operations on graphs:

- *Link* is the addition of an edge between two non-adjacent vertices.
- *Cut* is the removal of an existing edge.

Other combinatorial operations, for example insertion of vertex of degree 2, are assumed to have no cost.

Definition 3. $\text{cost}(G, \mathcal{C})$ is the minimum number of links and cuts needed to transform G into $\mathcal{G}(G, \mathcal{C})$.

Note that sometimes there are less combinatorial changes needed than the number of edges intersected by $\mathcal{C}$. Consider edges e_1, e_2 of G crossed consecutively by $\mathcal{C}$ and edge n adjacent to them that reappears in $\mathcal{G}(G, \mathcal{C})$ as a part n^* of $\mathcal{C}$. Then n^* can be obtained without any links or cuts by lifting n along e_1 and e_2 until it coincides with n^* or (which is the same) shrinking e_1 and e_2 until their endpoints coincide with their intersections with $\mathcal{C}$. We will call it *preserving operation.*

Theorem 3. *For a flarbable curve $\mathcal{C}$, it holds that*

$$\text{cost}(G, \mathcal{C}) \leq 12|\mathcal{S}(G, \mathcal{C})| + 3|\mathcal{B}(G, \mathcal{C})| + O(1).$$

Where

- *$|\mathcal{B}(G, \mathcal{C})|$ is the number of faces of G wholly contained inside $\mathcal{C}$.*
- *$|\mathcal{S}(G, \mathcal{C})|$ is the number of* shrinking *faces – i.e., the faces whose number of edges decreases when flarb operation is applied.*

The following upper bound can be used to evaluate the number of combinatorial changes needed to update the graph of a Voronoi diagram when a new site is inserted.

Theorem 4. *Consider one insertion of a new site to a Voronoi diagram V.*

- *The number of cells of V undergoing combinatorial changes is $O(N^{\frac{1}{2}})$ amortized in a sequence of insertions;*
- *There are a constant number of combinatorial changes per cell;*
- *The cells of V with combinatorial changes form a connected region.*

By *a change in cell* we always mean a combinatorial change, that is a *link* or a *cut.*

2.2 The coin problem

A detailed introduction into amortized analysis, including the definition of potential function and examples of estimates of it is given in [?]. Here we focus on one classical result concerning amortized analysis. It is rather folklore and is given as an exercise in several university courses; we show it here to establish its formal proof, because it is crucial for understanding of what we prove further.

Given n coins stored in n piles (some piles may be empty). A series of operations is executed on these piles. One operation consists of selecting a pile and distributing all its coins equally among other piles, one coin per pile. Piles that receive a coin and piles that do not can be chosen freely. An example of such operations can be seen in Figure **??**. We assume the number of piles is equal to the number of coins so that these operations can be executed on any configuration of coins.

Clearly during one operation at most n coins are distributed. However, during k consecutive operations the average number of coins disrtibuted per operation is $o(n)$. Informally, each operation makes the heights of the piles more uniform, making it impossible to have a large pile to distribute after each of the operations. This observation is formalized by the following theorem:

Theorem 5. *For $k \geq \sqrt{n}$, if k consecutive operations are executed, then the average number of coins distributed per operation is at most $3\sqrt{n}$.*

Proof. Denote piles by $p_1, \ldots, p_n$. We introduce a potential function that describes the configuration of piles and helps estimate the number of coins distributed during an operation executed on a pile:

$$\Phi = \sum_{j=1}^{n} \min\left\{\text{size}(p_j), \sqrt{n}\right\}.$$

Note that Φ is always at most n, since n is the sum of actual sizes of all the piles. Denote by T_i the number of coins distributed during the i-th operation, and by Φ_i the value of the potential after the i-th operation. Therefore, Φ_0 and Φ_k are the values of the potential before and after the execution of all the operations respectively.

We will now estimate $T_i + \Phi_{i-1} - \Phi_i$. To do so, note that the pile that is being distributed (without loss of generality, p_1) decreases in size to zero, and several other piles increase in size by 1. Distributing p_1 makes Φ decrease by at most $\sqrt{n}$, regardless of size(p_1) being greater or less than $\sqrt{n}$, by definition of Φ. Figure **??** illustrates it: a change in size of a pile does not affect the potential if the size of the pile is greater than $\sqrt{n}$.

Several other piles receive one coin, the number of such piles is equal to T_i. However, some of these piles may not contribute to the change in the potential, again because their size is greater than $\sqrt{n}$. Note that the number of such large piles is at most $\sqrt{n}$. Thus, when the coins are being put into piles, there is at least $T_i - \sqrt{n}$ increase in potential. Combining these observations,

$$T_i + \Phi_{i-1} - \Phi_i \;\leq\; T_i + \sqrt{n} - \left(T_i - \sqrt{n}\right) \;=\; 2\sqrt{n}.$$

We can now sum up $T_i + \Phi_{i-1} - \Phi_i$ for each of the consecutive operations. Note that $\Phi_0 - \Phi_k$ is between $-n$ and n.

$$\sum_{i=1}^{k} (T_i + \Phi_{i-1} - \Phi_i) \;=\; \sum_{i=1}^{k} T_i + \Phi_0 - \Phi_k \;\leq\; 2\sqrt{n} \cdot k + n.$$

This means average T_i per operation is at most $2\sqrt{n} + \frac{n}{k} = 3\sqrt{n}$. □

Informally, this theorem means that if during an operation one pile loses many coins, and many distinct piles get at most one coin each, there can only be so many of such operations. We will rely on this observation further on.

2.3 Gluings of squares

Given a collection of 2D polygons, a *gluing* describes a closed surface by specifying how to glue each edge of these polygons onto another edge. We consider only proper gluings, where only segments of equal lengths can be glued together. The following theorem is crucial in that it establishes the connection between gluings and convex polyhedra:

Theorem 6. *If a gluing is homeomorphic to a sphere and the sum of angles at each of its vertices is at most* $360°$ *then there is a single convex polyhedron* P *that can be glued from this net.*

Note that the polygons of the gluing may be folded in order to glue the polyhedron.

There is no known exact algorithm for reconstructing the 3D polyhedron. It is known that the problem of reconstructing the polyhedron can be reduced to a system of partial differential equations. Still this method does not produce the exact answer, and there is no known algorithm for it that works faster than pseudopolynomial in n, n being the number of vertices. Sometimes the coordinates of the polyhedron, even if its general shape is known, can not be expressed as closed formulas.

Enumerating all possible valid gluings is also not an easy task. Demaine et al. showed that for any even n there is a polygon with n vertices that has $2^{\Omega(n)}$ gluings: that is a star with two additional vertices on midpoints of edges half perimeter from one another. So it is also important to estimate the number of gluings of the collection of polygons under consideration.

Complete enumerations of gluings and the resulting polyhedra are only known for very specific cases such as the Latin cross, a single regular convex polygon, and a collection of regular pentagons glued edge-to-edge.

The case when the polygons to be glued together are all congruent regular k-gons, and the gluing is edge-to-edge, was studied recently for $k \geq 6$. Our aim is to study the case of $k = 4$: namely, to *enumerate* all valid gluings of squares and *classify* them up to isomorphism.

2.3.1 Chen—Han algorithm for gluings of squares

It is shown that polyhedra are isomorphic if the lengths of shortest geodesic paths between their vertices of nonzero curvature coincide. Thus, the problem of finding out if two gluings are isomorphic can be reduced to calculating the pairwise geodesic distances between vertices of a gluing. Algorithm we are using for this is the Chen—Han algorithm.

The idea of the algorithm is to project a cone of all possible paths from the source onto the polygons of the gluing. For n faces, this algorithm runs in $O(n^2)$ time. To apply it for arbitrary edge-to-edge gluings of squares, it has to be proven that the running time is preserved. We make use of a Lemma that was proved in the Bachelor's thesis.

Lemma 7. *If* T *is a square of the gluing and* π *is a geodesic shortest path between two vertices of the gluing then the intersection between* π *and* T *is of at most 5 segments.*

This lemma implies the following theorem.

Theorem 8. *The isomorphism between two edge-to-edge gluings of at most n squares can be tested in $O(n^2)$ time.*

2.4 PCB routing

A printed circuit board that consists of several layers can be thought of as a disjoint union of several metric spaces, distance between points of distinct spaces being defined additionally. The main problem is as follows. Given several layers of a printed circuit board *(PCB),* route all the wires so that certain design rules are respected and wires have small length, small number of bends, and small number of transitions between layers. There is of course a general problem of connecting points $a_1, \ldots, a_n$ with $b_1, \ldots, b_n$ pairwisely and optimally, but it is too difficult to solve practically, and probably is **NP**-hard. Most routing algorithms these days rely on empirical data only.

3 Purpose of the study

The purpose of this study is to find answers to the following open problems concerning metric spaces:

1) Find out if there exist algorithms for explicit incremental Voronoi diagrams whose running time is between $O(n^{3/4})$ and $O(n^{1/2})$.

2) Find a suitable variant of Voronoi diagrams for multilayer PCB boards and study their properties.

3) Find an appropriate data structure for maintaining and answering queries about divisions of space by arbitrary surfaces.

4) Estimate the number of convex polyhedra that can be glued using at most n congruent regular triangles, hexagons, or quadrilaterals.

5) Find a polynomial time approximation algorithm for the PCB routing problem.

6) Find weights for the parameters such that the corresponding multicriterion optimisation problem outputs results that are applicable in practice.

4 Methodology

4.1 Voronoi diagrams

One can try to employ the well-known «divide and conquer» strategy to devise an algorithm for the incremental diagram: divide all the sites into two halves by a line and update only the half of the diagram that receives the new site. It can be shown that adding a line as a site to the diagram still allows for a $O(\sqrt{n})$ upper bound on the number of combinatorial changes per insertion. This means that, theoretically, «divide and conquer» is a desirable approach.

One possible direction of research is to look into generalisation of algorithms and techniques for both implicit and explicit diagrams to various non-Euclidean metrics, for example those implied by the PCB problem. There are some hints that the algorithms will still work, but there can be certain underwater rocks in the process of translating the algorithm from one metric to another.

Another generalization to look at is considering not Voronoi cells, but any regions of space whose borders are arbitrary surfaces that intersect rarely enough, for example instance surfaces. One can find out what properties hold for such setups and what algorithms and data structures can be developed to work with them.

4.2 Nets

A net consisting of several regular polygons can be associated with a set of polygons drawn on a corresponding grid, edges of those polygons divided into pairs. The set of polygons should satisfy certain conditions: each of them being convex, the sum of angles of all the polygons at any vertex not exceeding $360°$, the edges in a pair having equal projections on coordinate axes. Then the estimate of the number of valid nets really boils down to the estimate of the number of drawings, which is a simple combinatorial problem.

4.3 PCB-s

The problem of finding a shortest path between two points in this setting can be thought of as multicriterion optimisation problem: we are trying to optimize not only overall length of the path, but also the number of turns and the number of transitions between layers (also called *vias,* those are especially expensive to make).

There are certain limitations that apply to the path that we can construct. First of all, only turns of $45°$ (that leave an internal angle of $135°$) are allowed. Second, there is minimal distance required between two adjacent wires. To some extent, we can think that there is a grid that the wires align to, but we can not use the size of this grid in the asymptotic estimations.

There are of course obstacles possible to the path we are constructing. To start with, one can assume that they admit some simple shape (convex, polygonal or round). To do proper preprocessing, however, one needs to count in dynamic obstacles posed by previously routed wires.

5 Contribution

If those problems are solved, the corresponding results will be the cornerstones of the corresponding fields, since they will be the answers for well posed questions that are important and understandable. They make us closer to finding constructive solutions to the Alexandrov's problem and help plan routes and process geodesic data faster and better.

The problem of PCB routing is tightly connected with several industrial projects that need feasible routing algorithms to work with.

Folding Polygons to Polyhedra

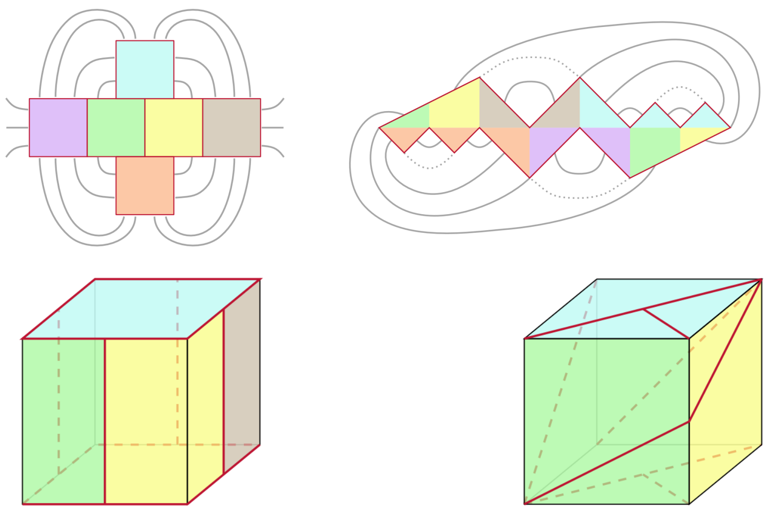
B. Zolotov, Ph. D. English Course

Alexandrov's Theorem

Is always brought up when it comes to gluing things.

If a gluing is homeomorphic to a sphere
and the sum of angles at each of
its vertices is $\leq 360°$, there $\exists!$ convex
polyhedron that corresponds to this gluing.

Alexandrov's Theorem — Examples

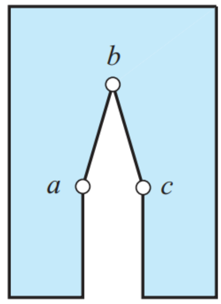


Folding a single polygon

As we can see from the previous example,
a single polygon can be glued to itself,
the same conditions remain.

We will be *only* considering
foldability to *convex* polyhedra.

A not-foldable polygon



Consider where the reflex vertex can be glued?..

Foldability of random polygons

No matter how «a random polygon» is sampled,
we are expecting that it:

- has at least two reflex vertices on average,
- the edge lengths are drawn from a cont. density distribution.

The probability that a random n-gon can be folded into a convex polyhedron approaches 0 as $n \to \infty$.

Perimeter-halving

Any convex polygon can be folded into
a convex polyhedron by *perimeter halving.*

By varying the points half a perimeter apart we get
uncountably many *incongruent* polyhedra.

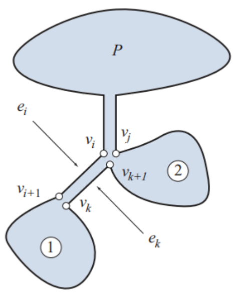
Edge-to-edge folding

Let us consider the subcase when entire edges
must be glued to entire edges.

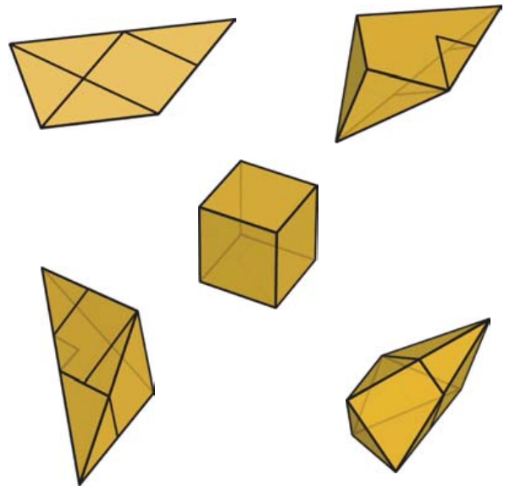
There is a dynamic programming algorithm
to list all the gluings.

Dynamic programming

Idea: find a pair of matching edges, check angles,
reduce to two subproblems.

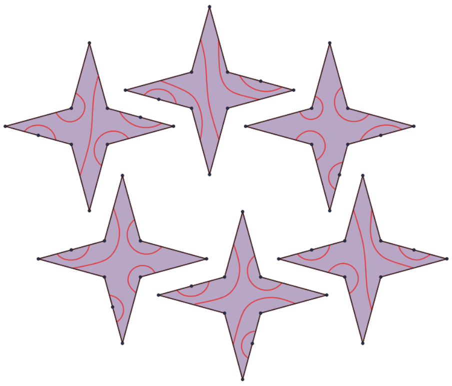


All e2e gluings of the Latin cross



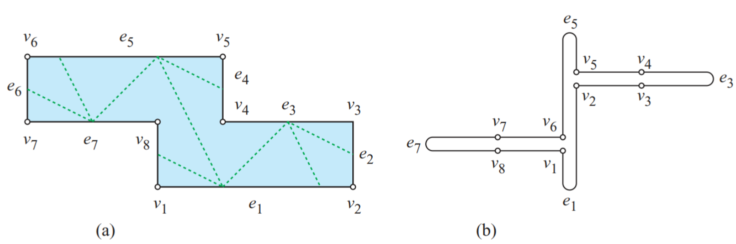
General algorithm

There is also an algorithm that lists *all* the foldings;
it has exponential running time.

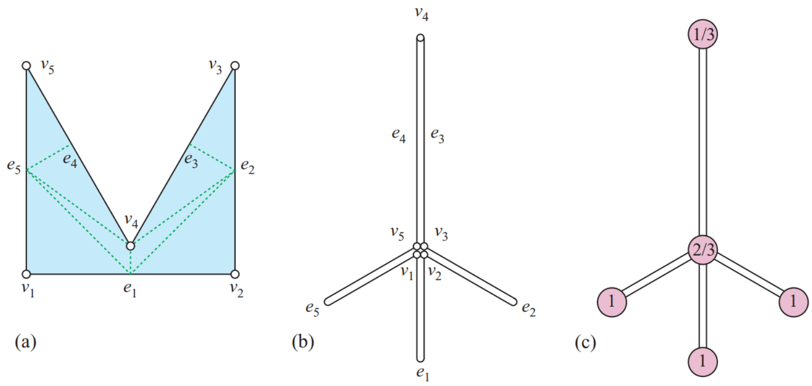


Gluing tree

Shows how the boundary of a folded polygon corresponds to itself. How to find the vertices of the polyhedron by looking at the vertices of the gluing tree?



Analyse the curvature



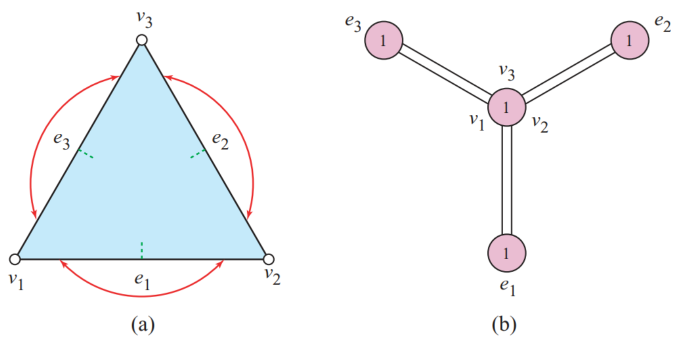
Rolling belt

A path in the gluing tree that:

- Connects two leaves of the gluing tree that are either convex-vertex or fold-point leaves (positive deficit),
- The face angle to each side of the path is, at every point, convex.

Rolling belts: examples

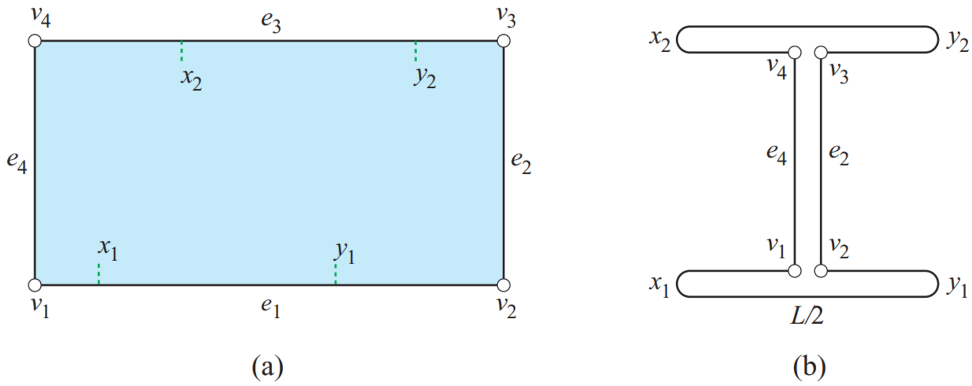
Half of perimeter of any convex polygon or



Gluing tree properties

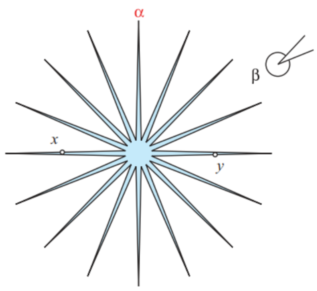
- Each leaf is either a vertex of the polygon, or a fold point.
- At most one nonvertex may be glued at any gluing-tree junction of degree $d \geq 3$.
- A gluing tree may have at most two rolling belts with distinct endpoints.
- The case of four fold-point leaves is possible only under special circumstances.

Rectangle with two rolling belts



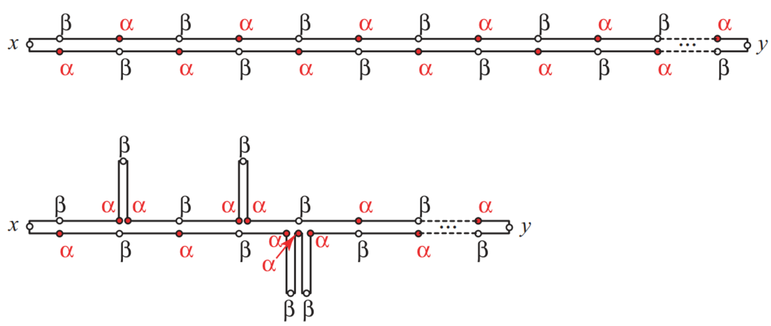
Exponential number of gluing trees

For any even n, there is a polygon P of n vertices that has $2^{\Omega(n)}$ combinatorially distinct edge-to-edge Alexandrov gluings.



Contracted gluing trees

There is the default gluing tree, it can be contracted in an exponential number of ways to form new trees.



Thank you for your attention!

- Not-foldable polygon, perimeter halving
- Algorithms listing foldings
- Gluing trees, rolling belts
- Exponentially many foldings

Folding Polygons to Polyhedra

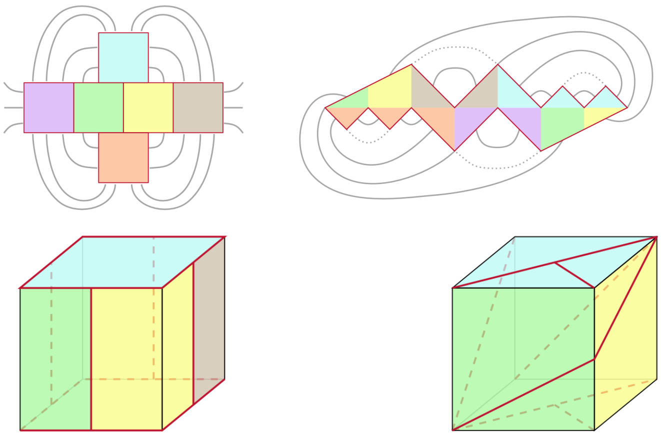
B. Zolotov, Ph. D. English Course

Alexandrov's Theorem

Is always brought up when it comes to gluing things.

If a gluing is homeomorphic to a sphere
and the sum of angles at each of
its vertices is $\leq 360^\circ$, there $\exists!$ convex
polyhedron that corresponds to this gluing.

Alexandrov's Theorem — Examples

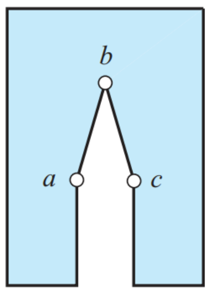


Folding a single polygon

As we can see from the previous example, a single polygon can be glued to itself, the same conditions remain.

We will be *only* considering foldability to *convex* polyhedra.

A not-foldable polygon



Consider where the reflex vertex can be glued?..

Foldability of random polygons

No matter how «a random polygon» is sampled,
we are expecting that it:

- has at least two reflex vertices on average,
- the edge lengths are drawn from a
  cont. density distribution.

The probability that a random *n*-gon can be folded into a convex polyhedron approaches 0 as $n \to \infty$.

Perimeter-halving

Any convex polygon can be folded into
a convex polyhedron by *perimeter halving.*

By varying the points half a perimeter apart we get uncountably many *incongruent* polyhedra.

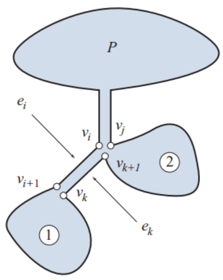
Edge-to-edge folding

Let us consider the subcase when entire edges must be glued to entire edges.

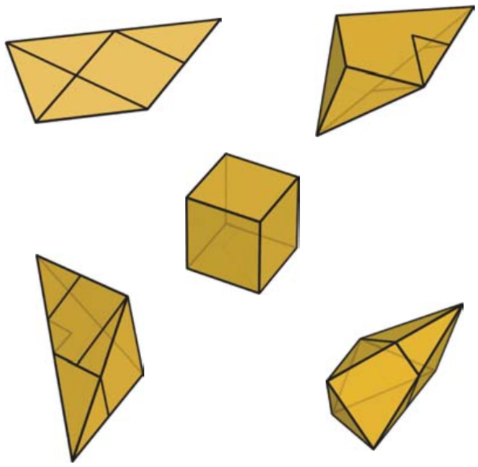
There is a dynamic programming algorithm to list all the gluings.

Dynamic programming

Idea: find a pair of matching edges, check angles, reduce to two subproblems.

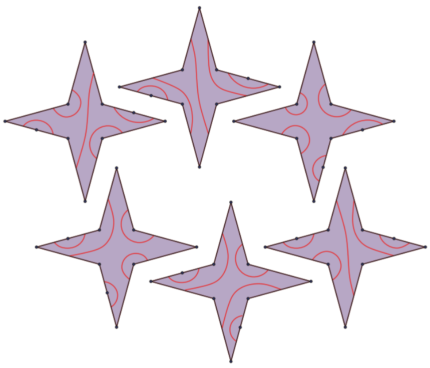


All e2e gluings of the Latin cross



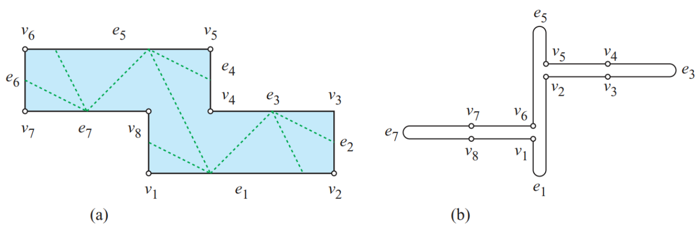
General algorithm

There is also an algorithm that lists *all* the foldings; it has exponential running time.

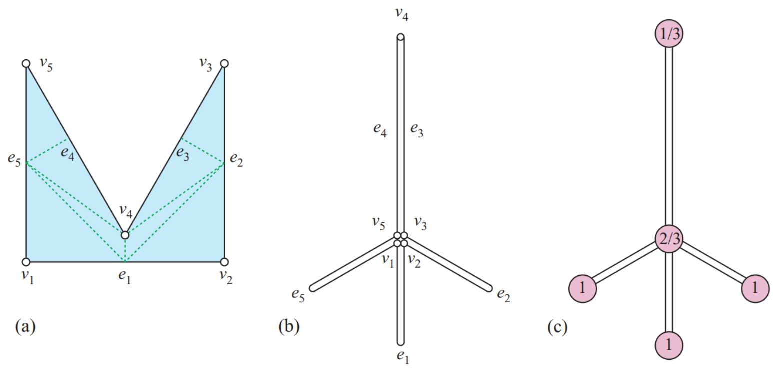


Gluing tree

Shows how the boundary of a folded polygon corresponds to itself. How to find the vertices of the polyhedron by looking at the vertices of the gluing tree?



Analyse the curvature



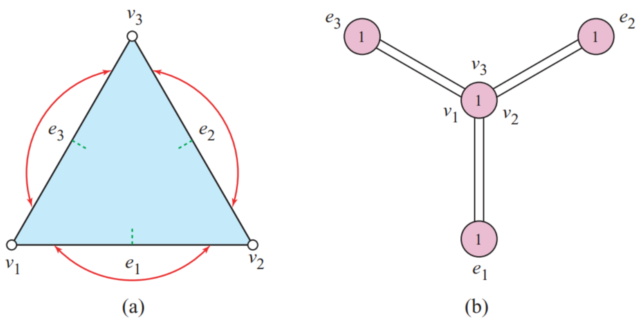
Rolling belt

A path in the gluing tree that:

- Connects two leaves of the gluing tree that are either convex-vertex or fold-point leaves (positive deficit),
- The face angle to each side of the path is, at every point, convex.

Rolling belts: examples

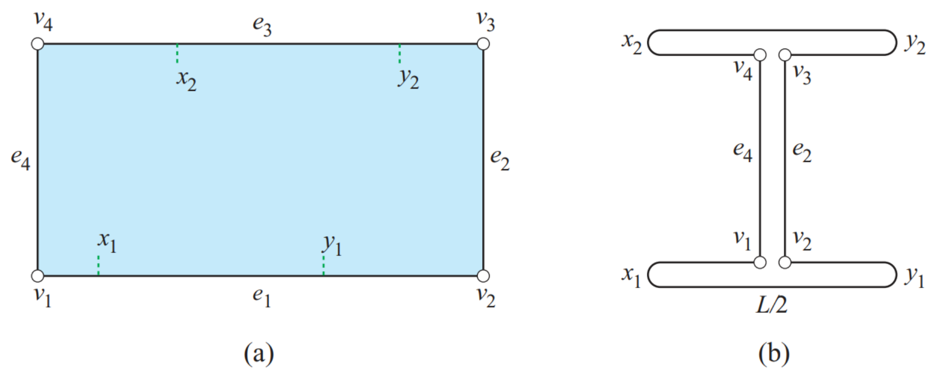
Half of perimeter of any convex polygon or



Gluing tree properties

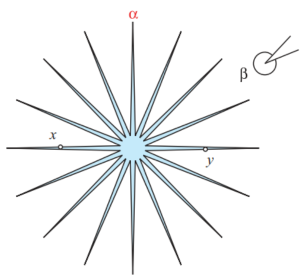
- Each leaf is either a vertex of the polygon, or a fold point.
- At most one nonvertex may be glued at any gluing-tree junction of degree $d \geq 3$.
- A gluing tree may have at most two rolling belts with distinct endpoints.
- The case of four fold-point leaves is possible only under special circumstances.

Rectangle with two rolling belts



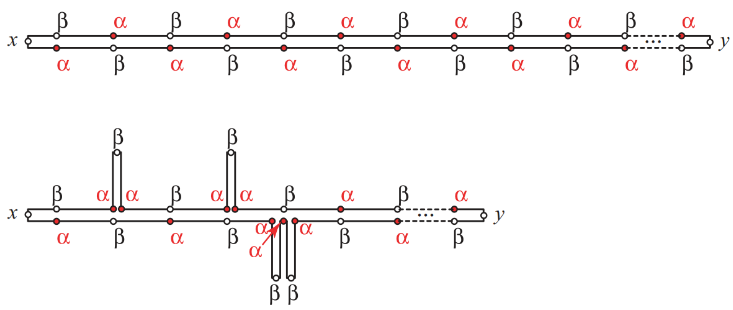
Exponential number of gluing trees

For any even n, there is a polygon P of n vertices that has $2^{\Omega(n)}$ combinatorially distinct edge-to-edge Alexandrov gluings.



Contracted gluing trees

There is the default gluing tree, it can be contracted in an exponential number of ways to form new trees.



Thank you for your attention!

- Not-foldable polygon, perimeter halving
- Algorithms listing foldings
- Gluing trees, rolling belts
- Exponentially many foldings