Computational Geometry
Chapter 5: Polygon Triangulation

Sándor P. Fekete

Algorithms Division
Department of Computer Science
TU Braunschweig

Technische Universität Braunschweig

- Ch 1: Introduction
- Ch 2: Convex hulls
- Ch 3: Closest pairs
- Ch 4: Voronoi diagrams
- Ch 5: Polygon triangulation
- Ch 6: Point triangulation
- Ch 7: Location problems
- Ch 8: Minimum area polygons

Goal

Problem 5.1

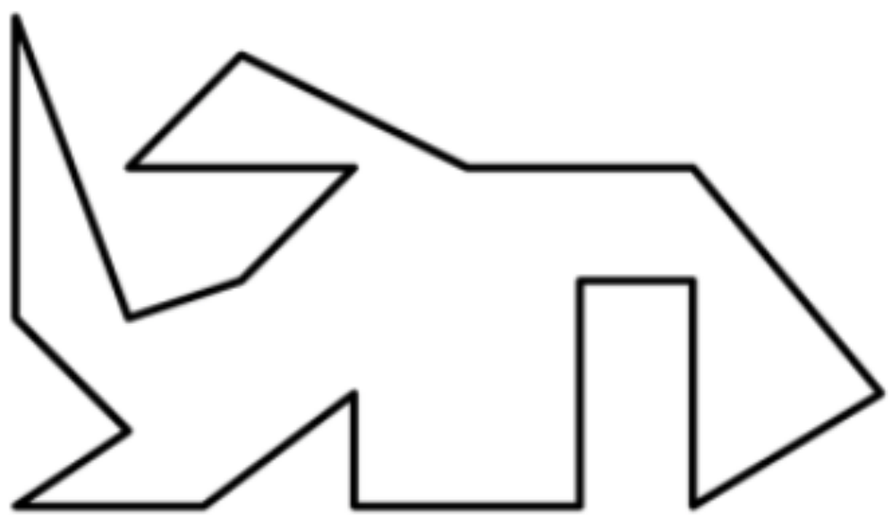
Goal

Problem 5.1

Given: A (simple) polygon P in $\mathbb{R}^2$

Problem 5.1

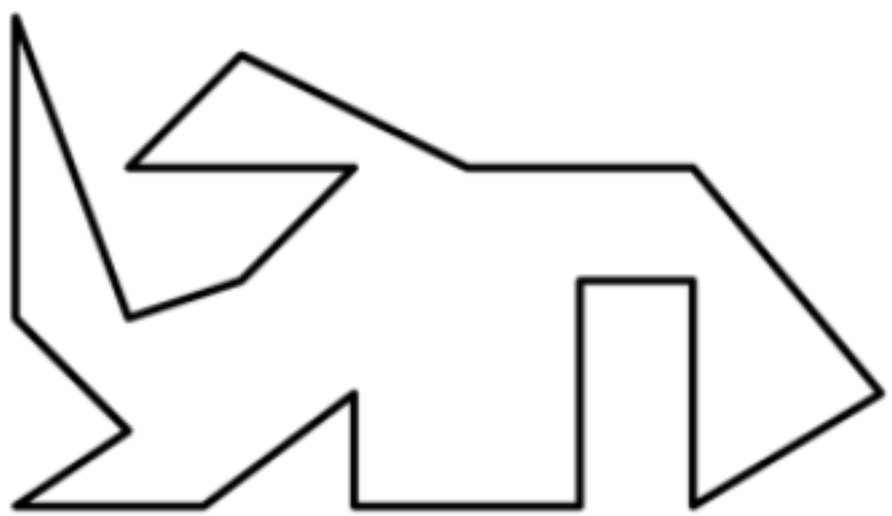
Given: A (simple) polygon P in $\mathbb{R}^2$



Problem 5.1

Given: A (simple) polygon P in $\mathbb{R}^2$

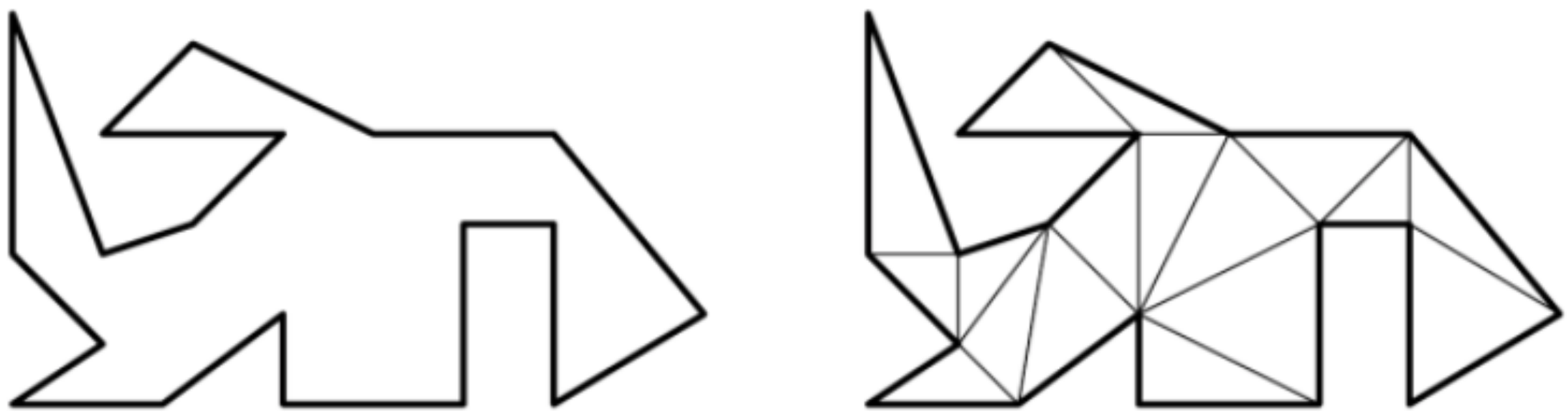
Wanted: Subdivision of P by disjoint vertex connections into triangles



Problem 5.1

Given: A (simple) polygon P in $\mathbb{R}^2$

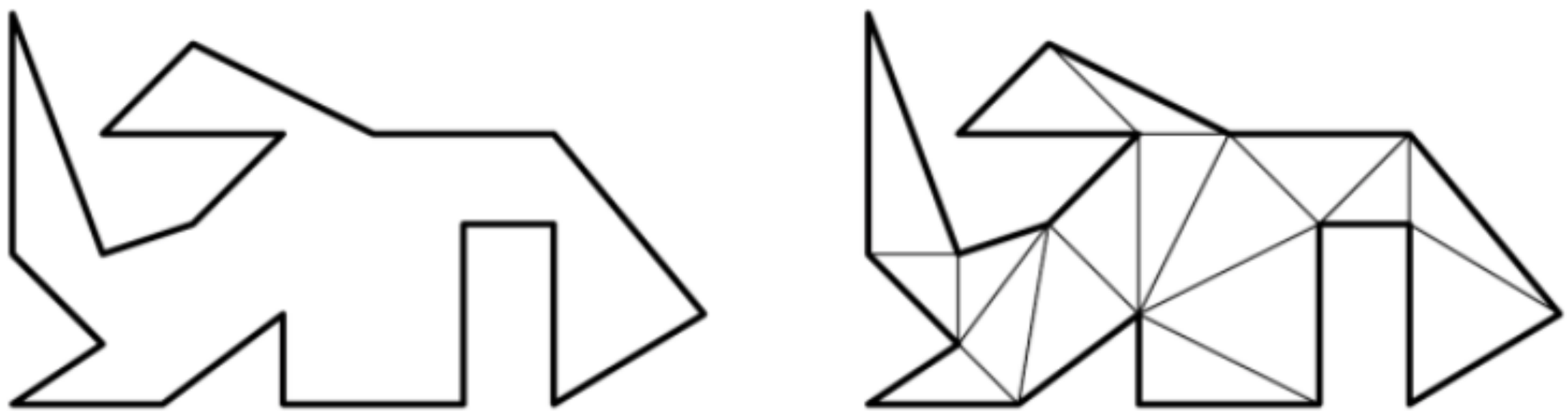
Wanted: Subdivision of P by disjoint vertex connections into triangles



Problem 5.1

Given: A (simple) polygon P in $\mathbb{R}^2$

Wanted: Subdivision of P by disjoint vertex connections into triangles

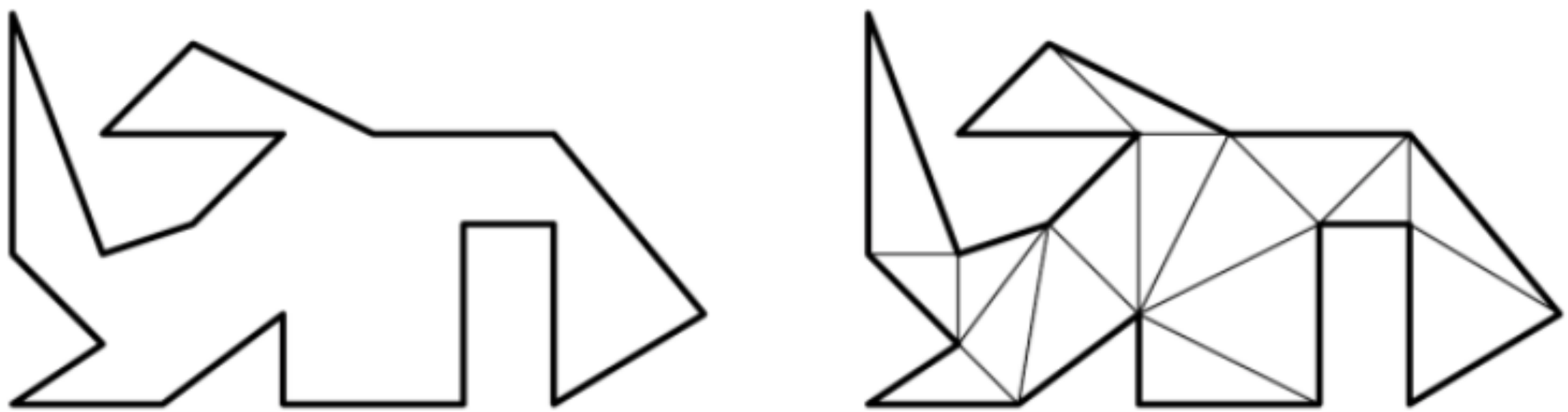


Definition 5.2

Problem 5.1

Given: A (simple) polygon P in $\mathbb{R}^2$

Wanted: Subdivision of P by disjoint vertex connections into triangles



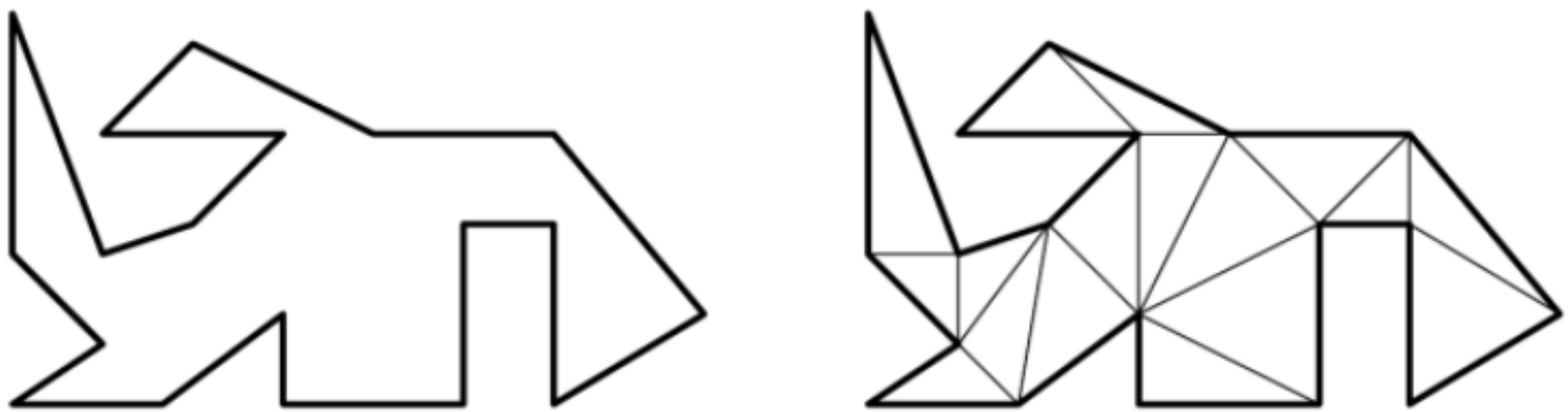
Definition 5.2

- A *diagonal* in a polygon connects two distinct vertices; its interior lies inside the polygon.

Problem 5.1

Given: A (simple) polygon P in $\mathbb{R}^2$

Wanted: Subdivision of P by disjoint vertex connections into triangles

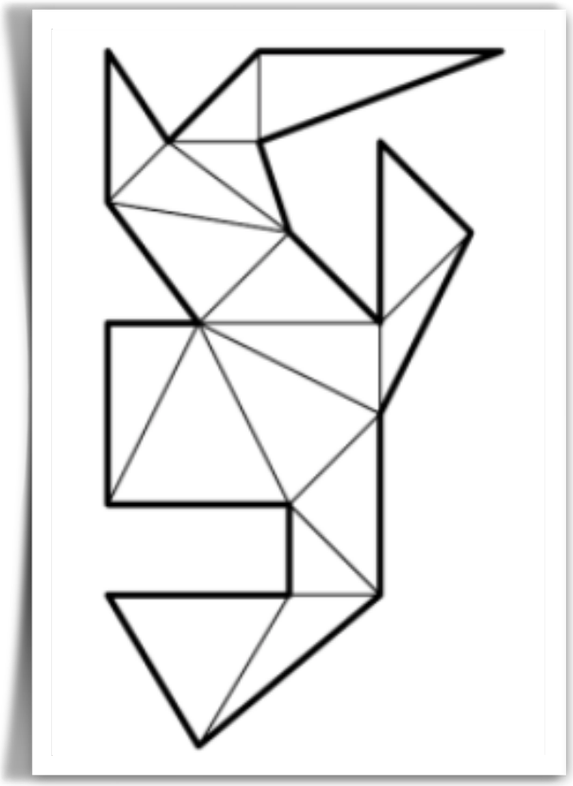


Definition 5.2

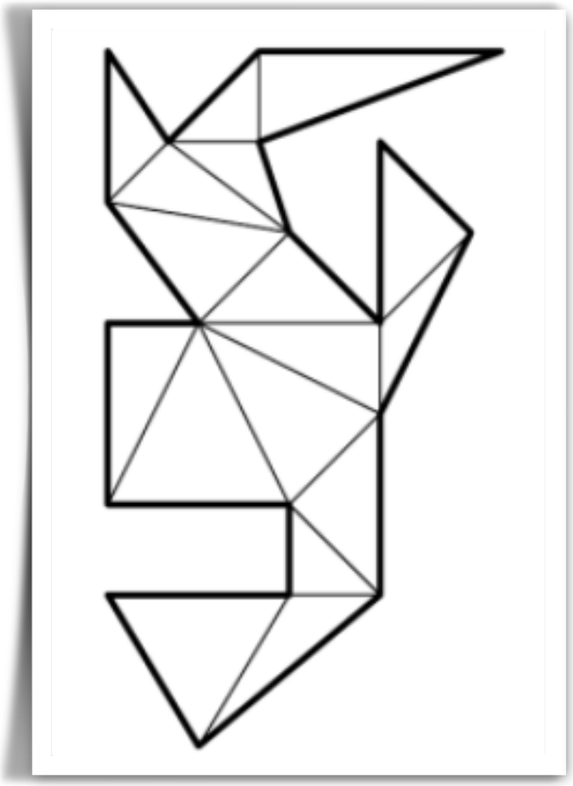
- A *diagonal* in a polygon connects two distinct vertices; its interior lies inside the polygon.
- Two different diagonals in a triangulation may share end points, but nothing else.

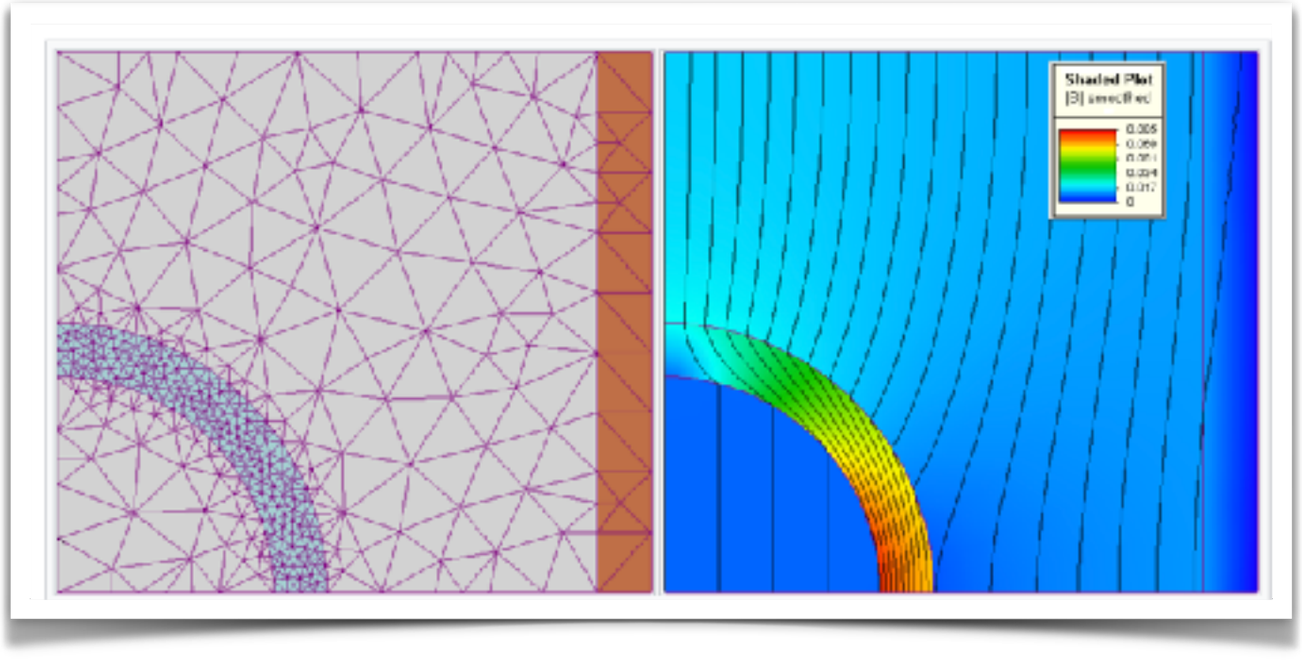
Motivation

Motivation

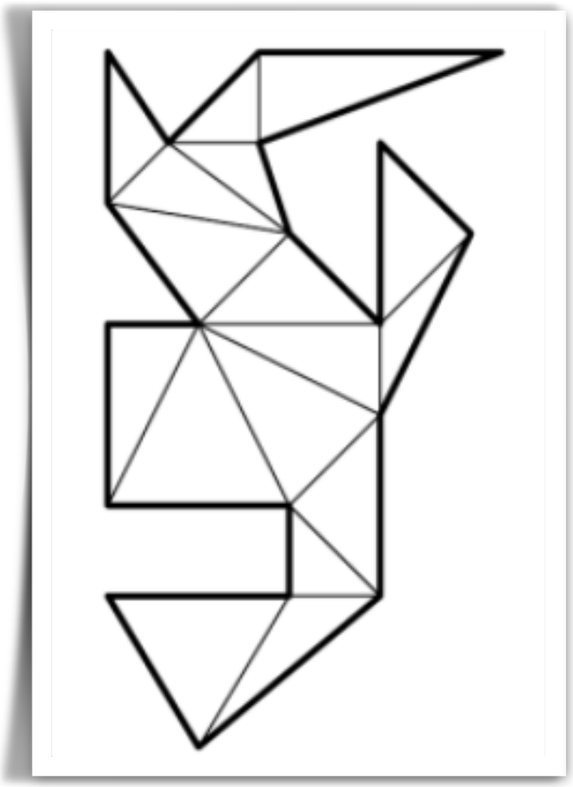


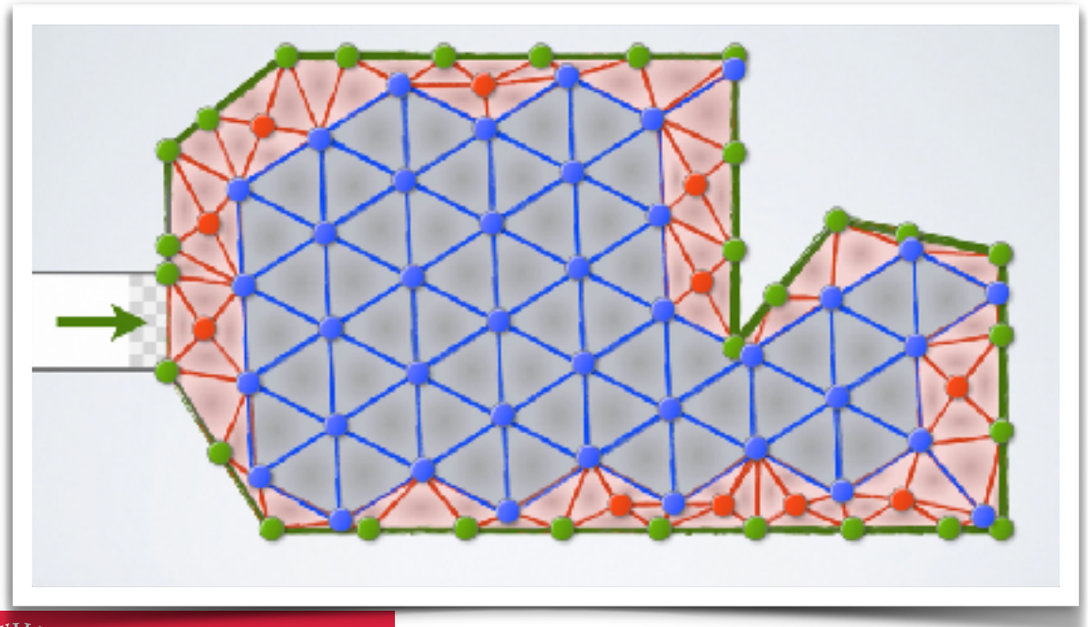
Motivation

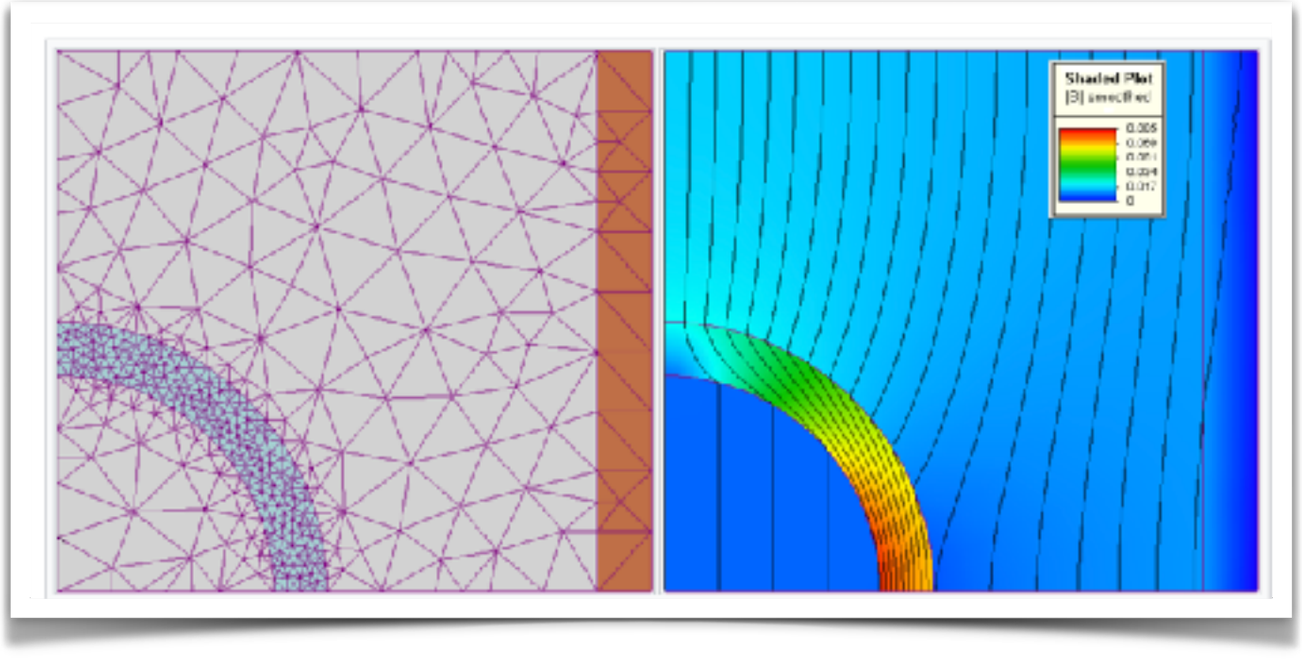




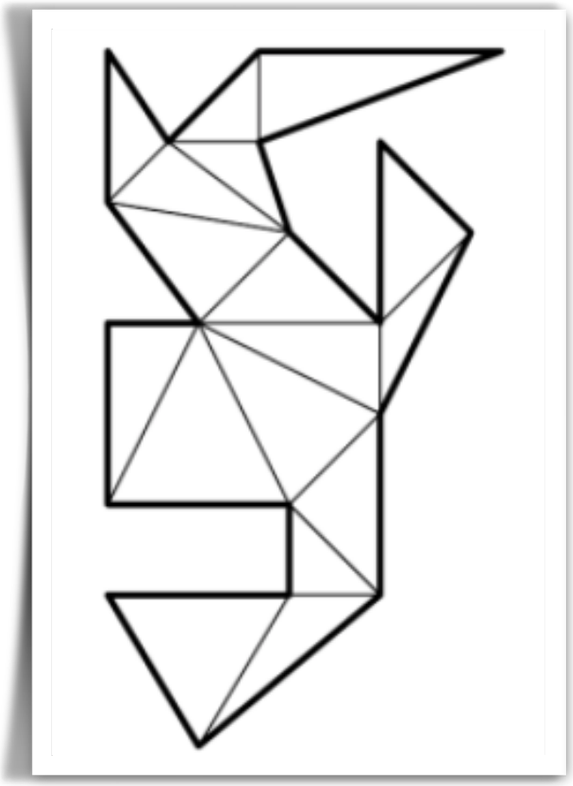
Motivation

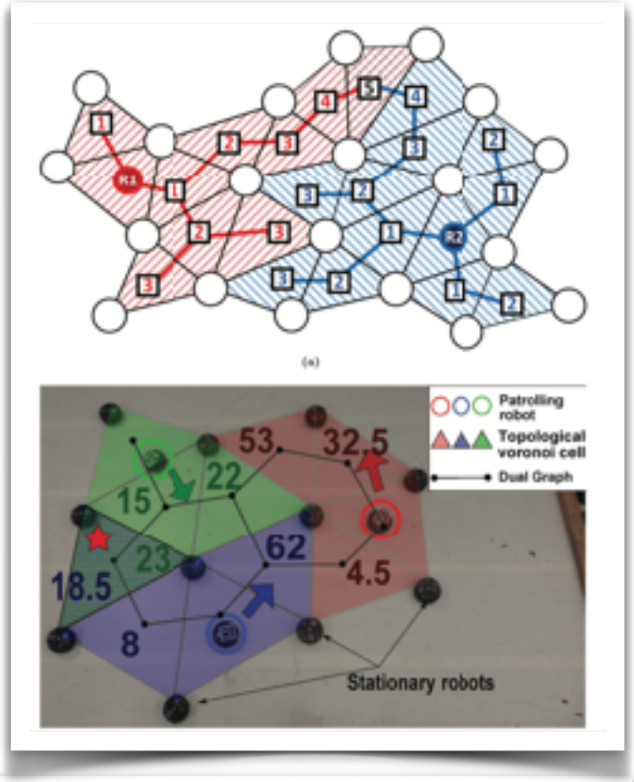


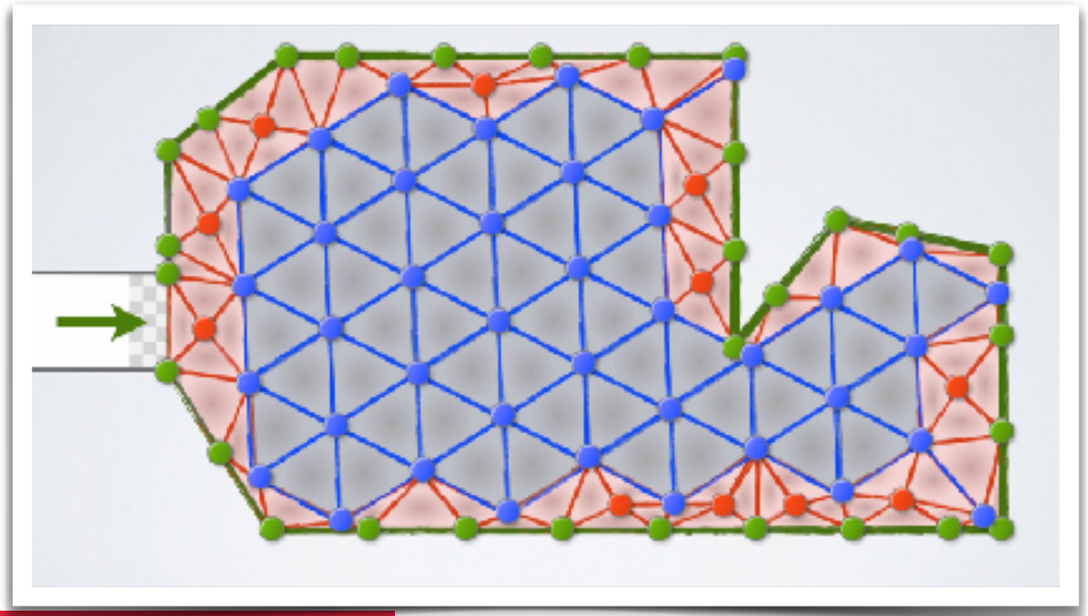


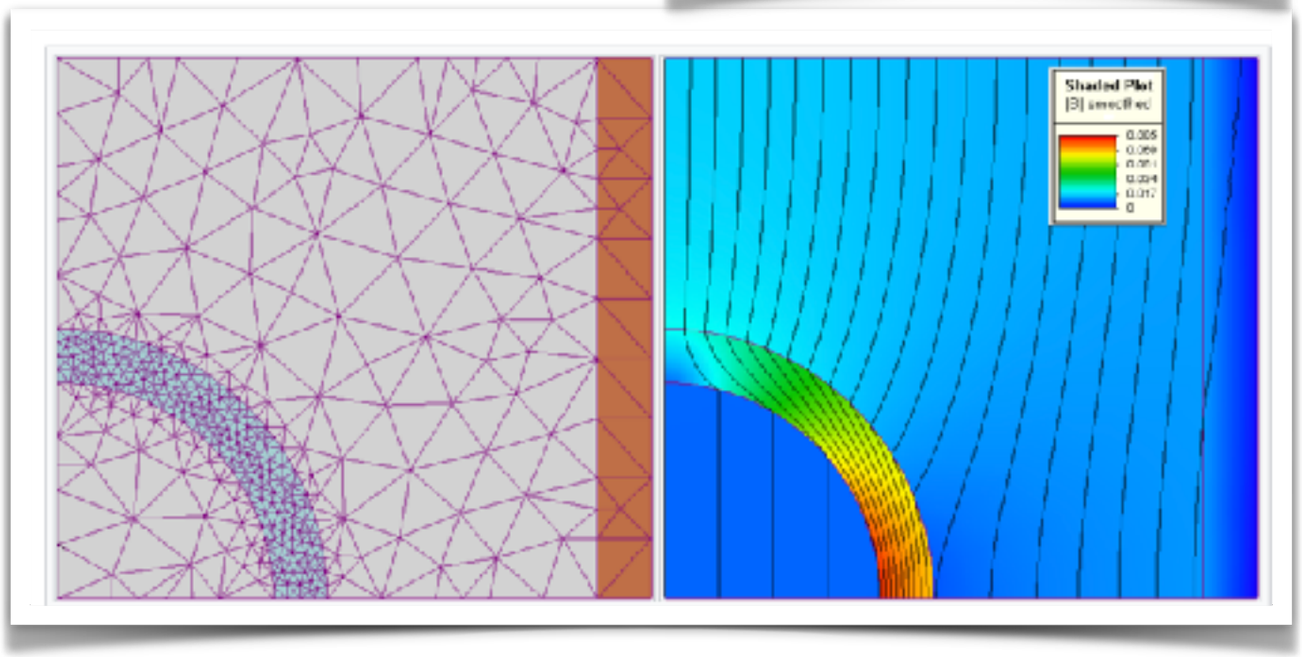


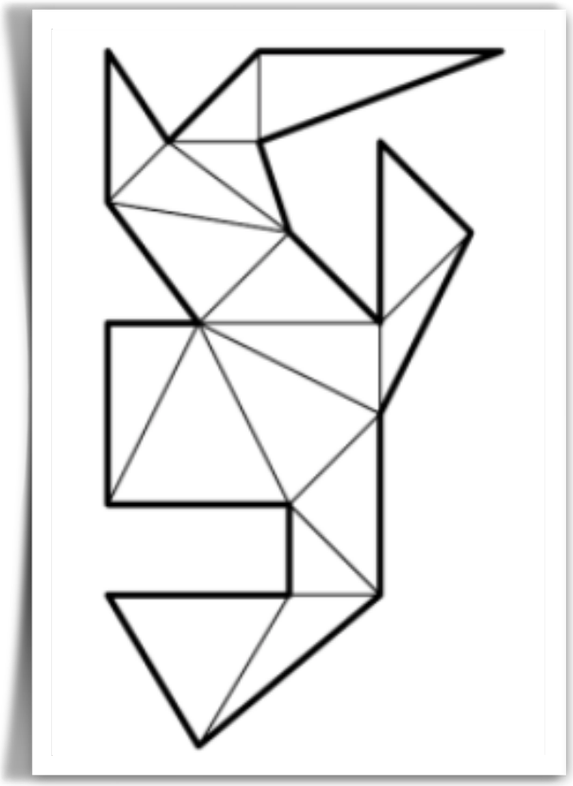
Motivation



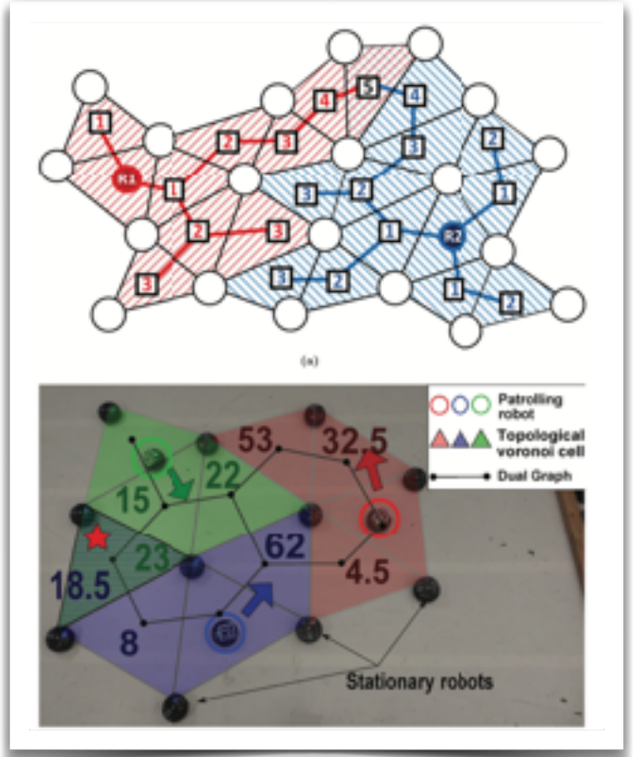

Patrolling robot
Topological voronoi cell
Dual Graph
53
32.5
22
15
62
23
18.5
4.5
8
Stationary robots

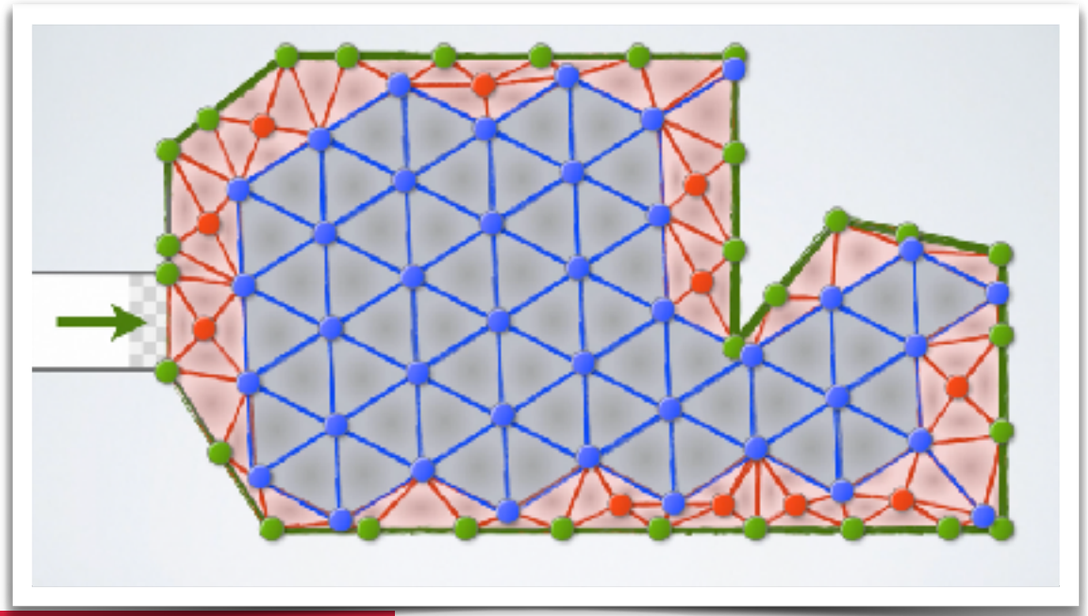


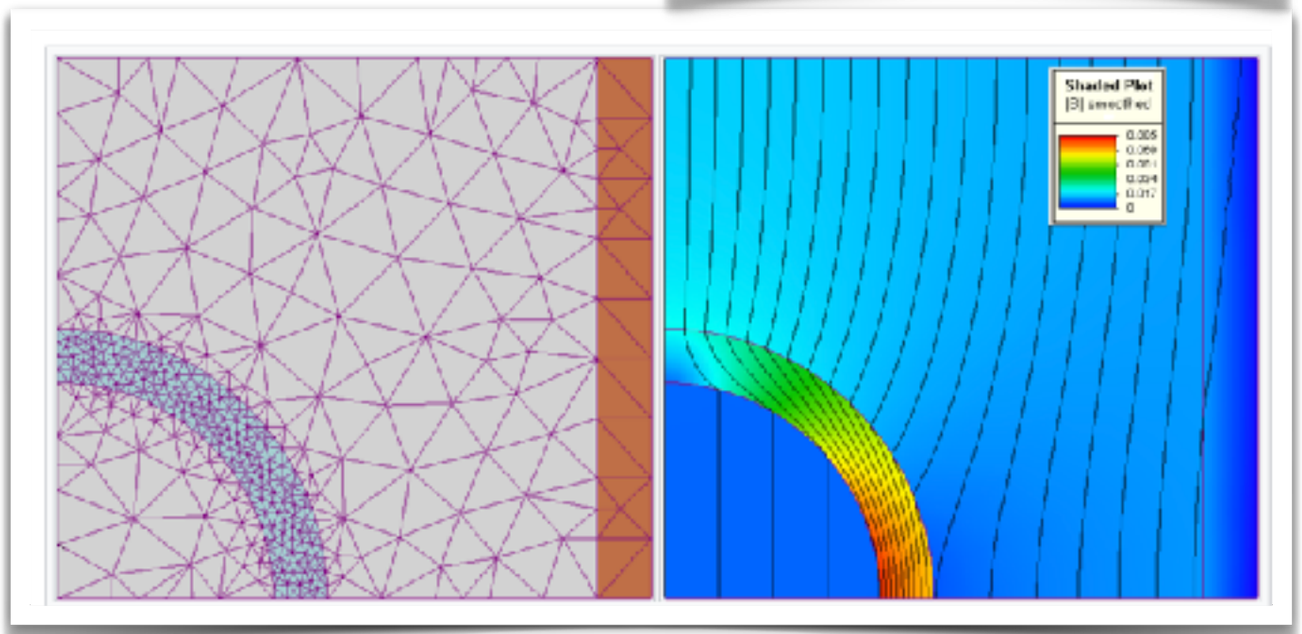

Shaded Plot



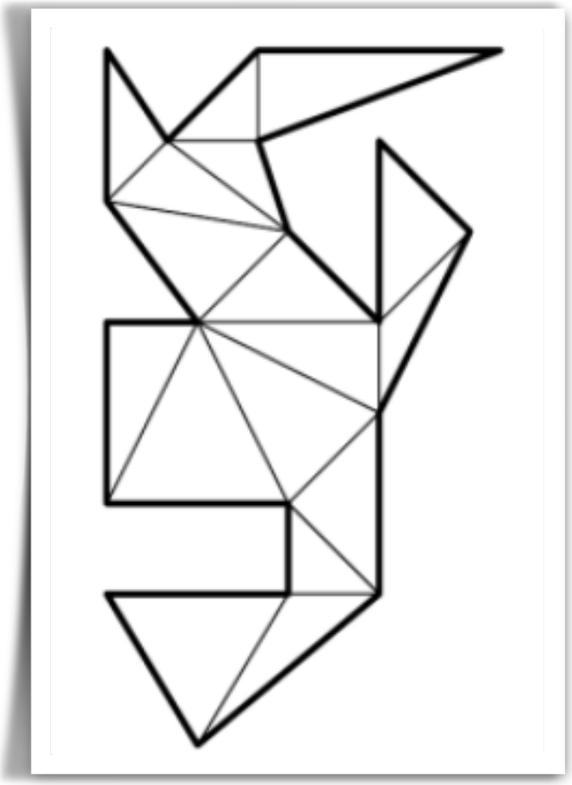
THE ART GALLERY PROBLEM

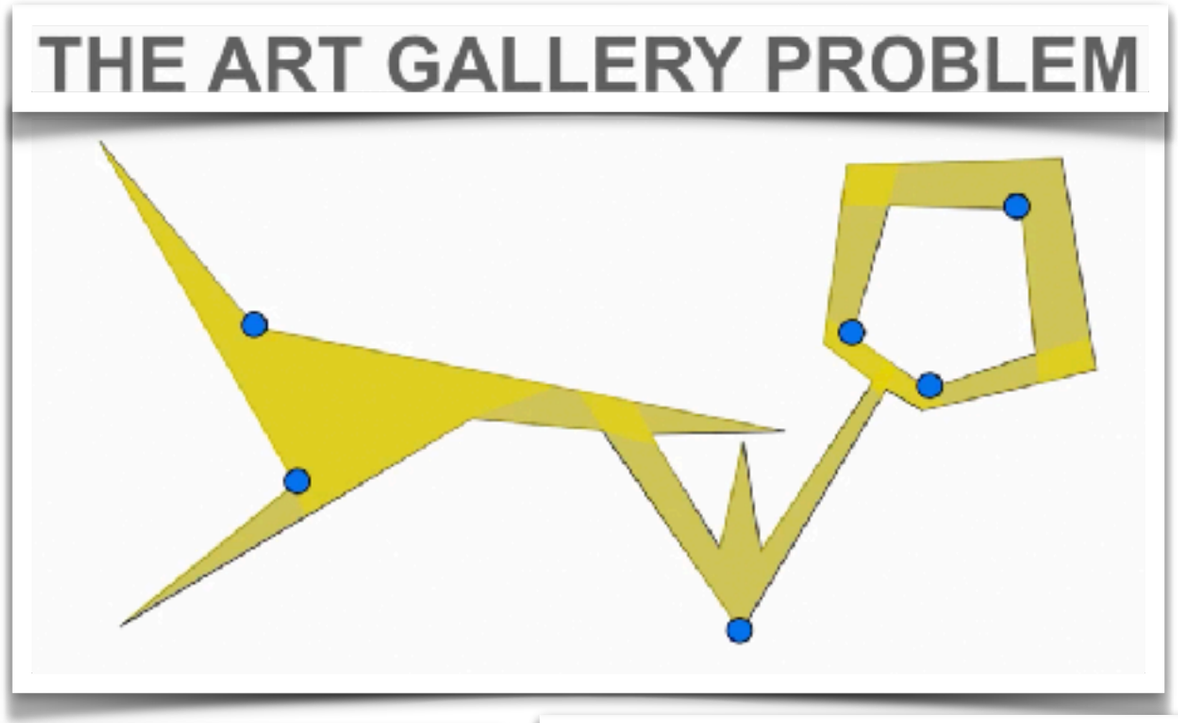


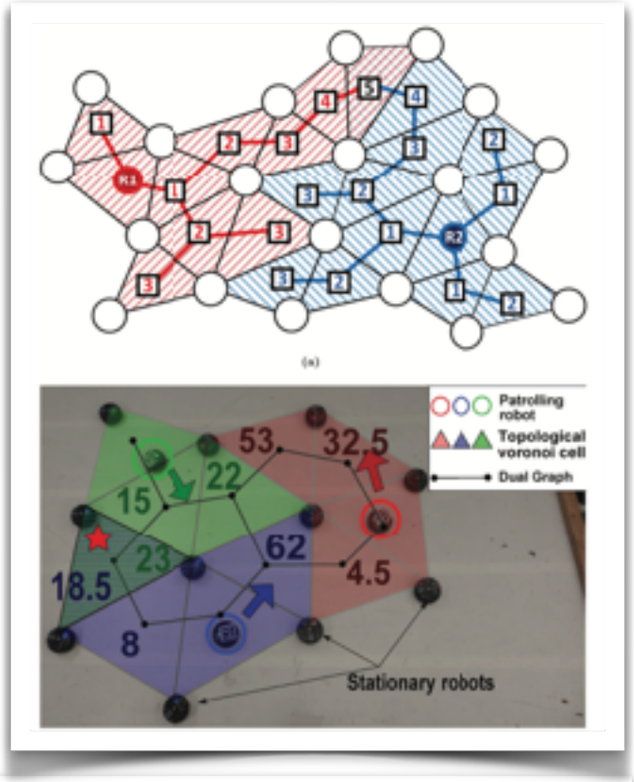


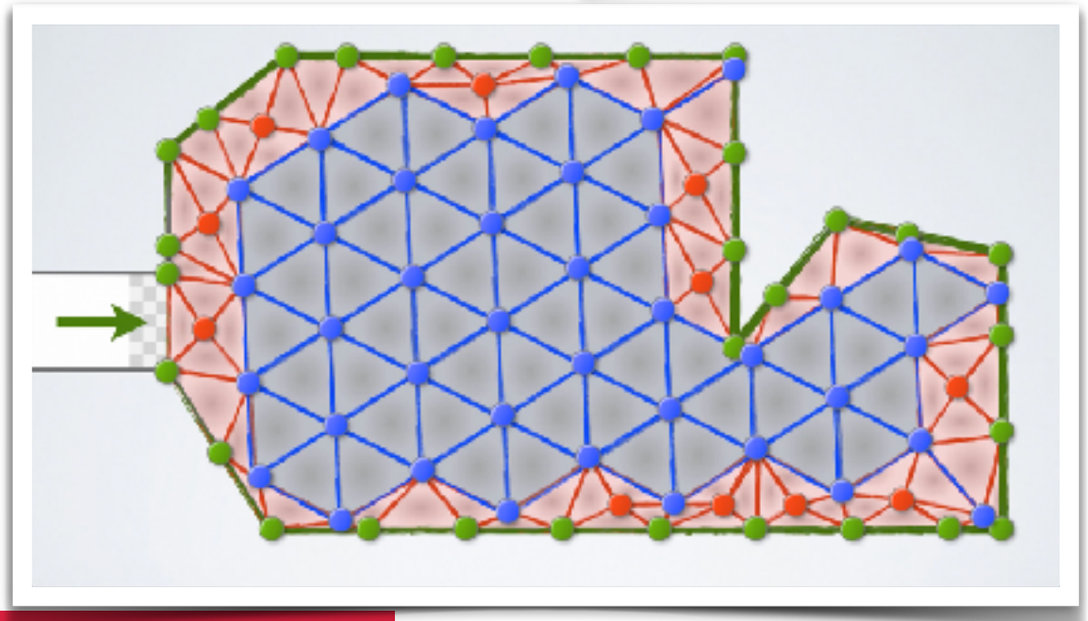


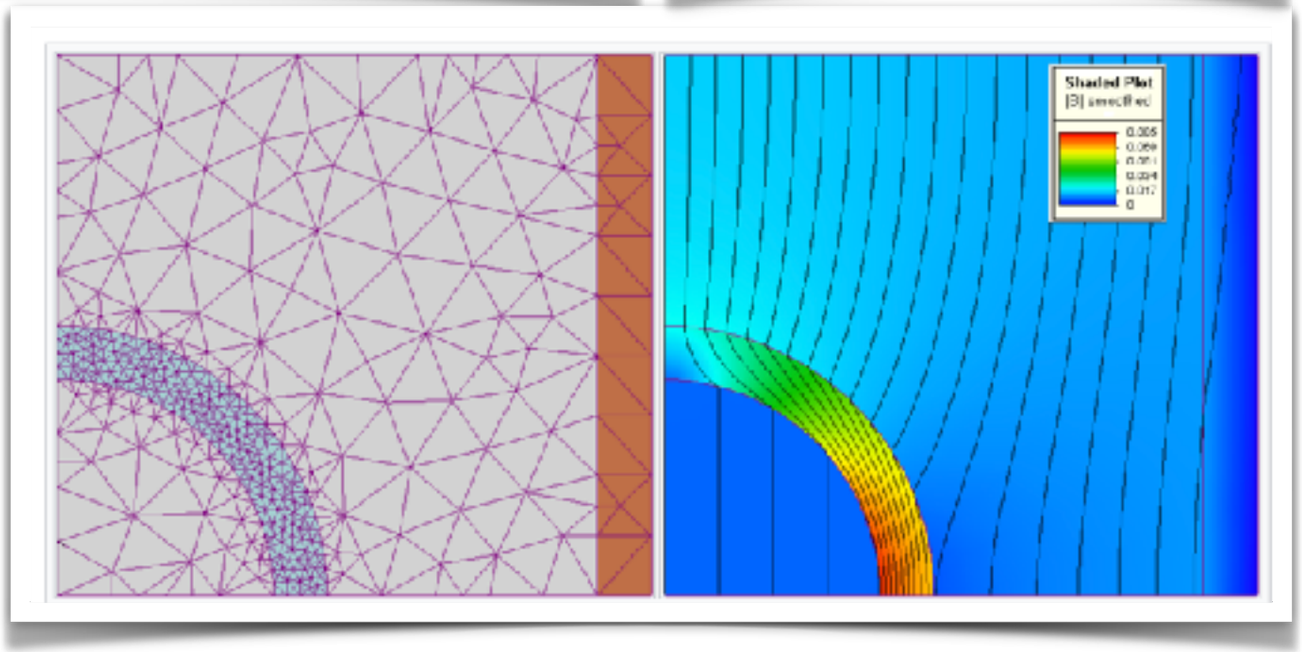
Motivation




THE ART GALLERY PROBLEM


Patrolling robot
Topological voronoi cell
Dual Graph
53
32.5
22
15
62
23
4.5
18.5
8
Stationary robots




Shaded Plot

Theorem 5.9

The dual graph of a triangulation of a simple polygon is a tree.

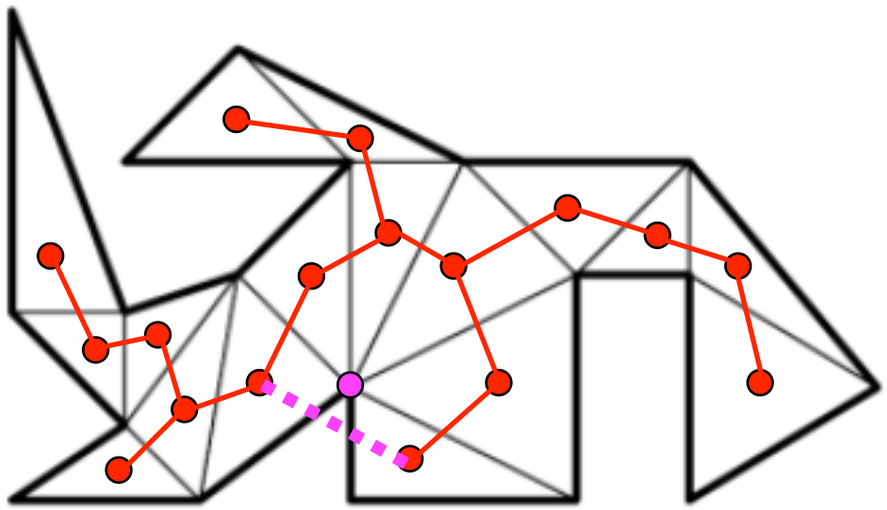
Proof:

The dual graph is trivially connected.

Assume it contains a cycle.

Note that each dual edge corresponds to a path of interior polygon points.

Therefore, a cycle corresponds to a closed path of interior points that surrounds a boundary vertex – a contradiction to a simple polygon having a connected boundary.



Corollary 5.10

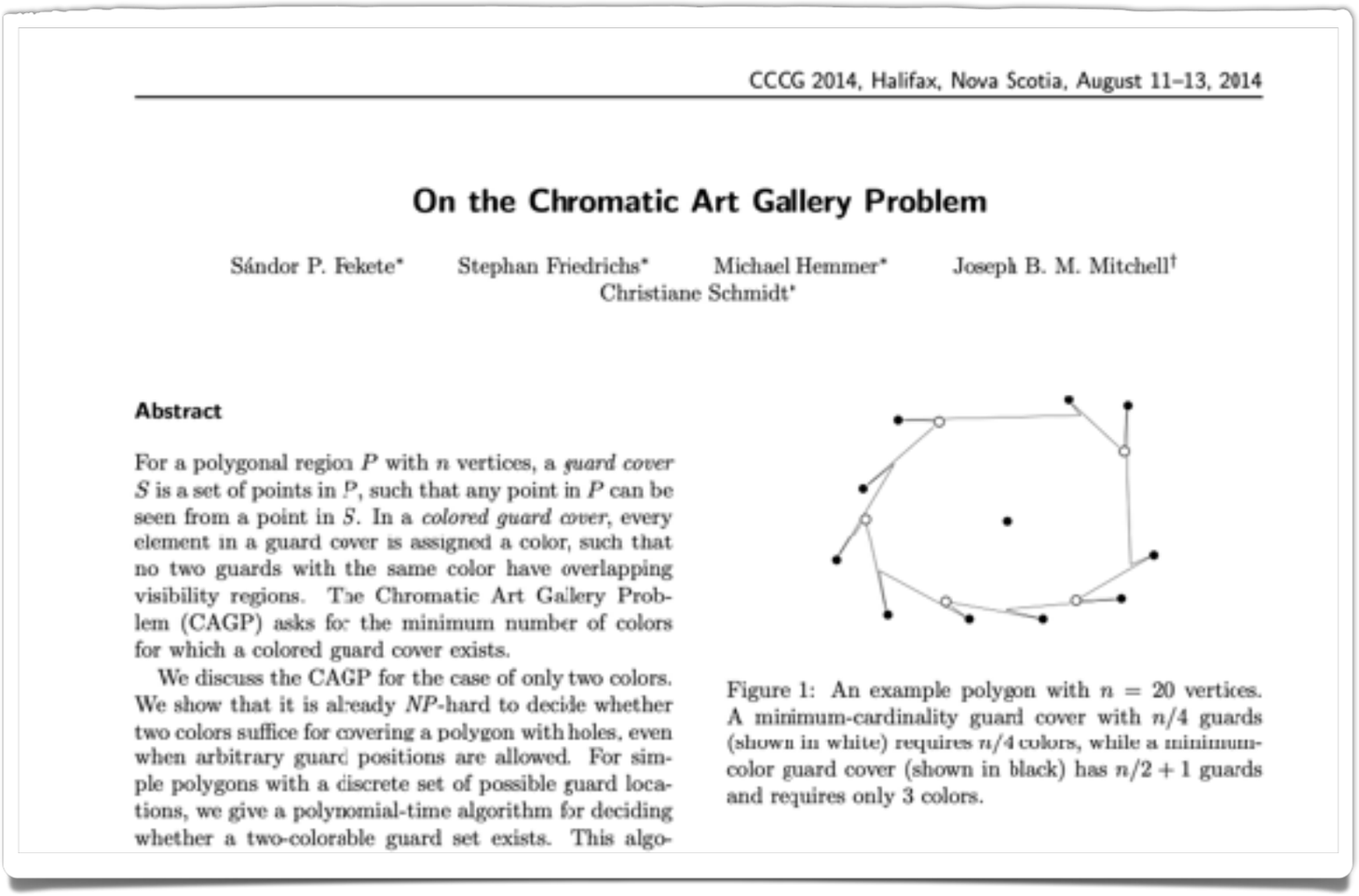
Every triangulation of a simple polygon has two „ears“ - leaves in the dual.

Theorem 5.9

The dual graph of a triangulation of a simple polygon is a tree.

Theorem 5.9

The dual graph of a triangulation of a simple polygon is a tree.



CCCG 2014, Halifax, Nova Scotia, August 11–13, 2014

On the Chromatic Art Gallery Problem

Sándor P. Fekete* Stephan Friedrichs* Michael Hemmer* Joseph B. M. Mitchell† Christiane Schmidt*

Abstract

For a polygonal region P with n vertices, a *guard cover* S is a set of points in P, such that any point in P can be seen from a point in S. In a *colored guard cover*, every element in a guard cover is assigned a color, such that no two guards with the same color have overlapping visibility regions. The Chromatic Art Gallery Problem (CAGP) asks for the minimum number of colors for which a colored guard cover exists.

We discuss the CAGP for the case of only two colors. We show that it is already NP-hard to decide whether two colors suffice for covering a polygon with holes, even when arbitrary guard positions are allowed. For simple polygons with a discrete set of possible guard locations, we give a polynomial-time algorithm for deciding whether a two-colorable guard set exists. This algo-

Figure 1: An example polygon with $n = 20$ vertices. A minimum-cardinality guard cover with $n/4$ guards (shown in white) requires $n/4$ colors, while a minimum-color guard cover (shown in black) has $n/2 + 1$ guards and requires only 3 colors.

Theorem 5.9

The dual graph of a triangulation of a simple polygon is a tree.

CCCG 2014, Halifax, Nova Scotia, August 11–13, 2014

On the Chromatic Art Gallery Problem

Sándor P. Fekete* Stephan Friedrichs* Michael Hemmer* Joseph B. M. Mitchell†
Christiane Schmidt*

Abstract

For a polygonal region P with n vertices, a *guard cover* S is a set of points in P, such that any point in P can be seen from a point in S. In a *colored guard cover*, every element in a guard cover is assigned a color, such that no two guards with the same color have overlapping visibility regions. The Chromatic Art Gallery Problem (CAGP) asks for the minimum number of colors for which a colored guard cover exists.

We discuss the CAGP for the case of only two colors. We show that it is already *NP*-hard to decide whether two colors suffice for covering a polygon with holes, even when arbitrary guard positions are allowed. For simple polygons with a discrete set of possible guard locations, we give a polynomial-time algorithm for deciding whether a two-colorable guard set exists. This algo-

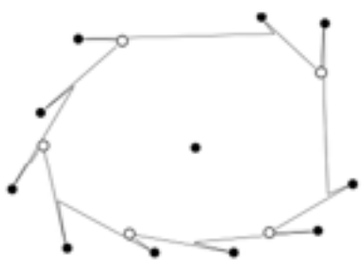


Figure 1: An example polygon with $n = 20$ vertices. A minimum-cardinality guard cover with $n/4$ guards (shown in white) requires $n/4$ colors, while a minimum-color guard cover (shown in black) has $n/2 + 1$ guards and requires only 3 colors.

Theorem 4.1 *For a simple polygon P with n vertices and ℓ discrete guard locations L, it can be decided in polynomial time whether there is a 2-colorable guard set.*

Theorem 5.9

The dual graph of a triangulation of a simple polygon is a tree.

CCCG 2014, Halifax, Nova Scotia, August 11–13, 2014

On the Chromatic Art Gallery Problem

Sándor P. Fekete* Stephan Friedrichs* Michael Hemmer* Joseph B. M. Mitchell†
Christiane Schmidt*

Abstract

For a polygonal region P with n vertices, a *guard cover* S is a set of points in P, such that any point in P can be seen from a point in S. In a *colored guard cover*, every element in a guard cover is assigned a color, such that no two guards with the same color have overlapping visibility regions. The Chromatic Art Gallery Problem (CAGP) asks for the minimum number of colors for which a colored guard cover exists.

We discuss the CAGP for the case of only two colors. We show that it is already *NP*-hard to decide whether two colors suffice for covering a polygon with holes, even when arbitrary guard positions are allowed. For simple polygons with a discrete set of possible guard locations, we give a polynomial-time algorithm for deciding whether a two-colorable guard set exists. This algo-

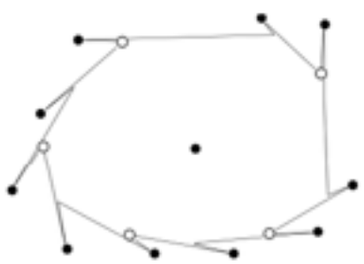


Figure 1: An example polygon with $n = 20$ vertices. A minimum-cardinality guard cover with $n/4$ guards (shown in white) requires $n/4$ colors, while a minimum-color guard cover (shown in black) has $n/2 + 1$ guards and requires only 3 colors.

Theorem 4.1 *For a simple polygon P with n vertices and ℓ discrete guard locations L, it can be decided in polynomial time whether there is a 2-colorable guard set.*

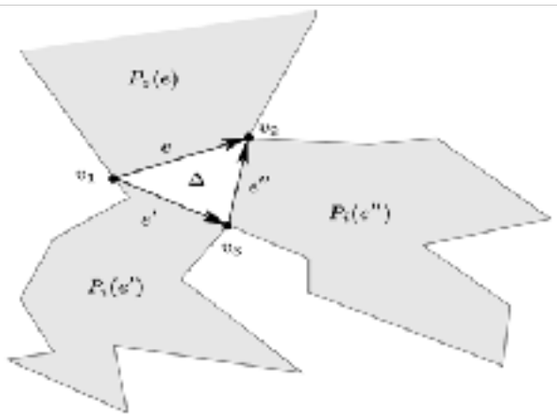


Figure 7: Solving a subproblem (e, Γ) by combining solutions for subproblems (e', Γ') and (e'', Γ'').

Theorem 5.9

The dual graph of a triangulation of a simple polygon is a tree.

CCCG 2014, Halifax, Nova Scotia, August 11–13, 2014

On the Chromatic Art Gallery Problem

Sándor P. Fekete* Stephan Friedrichs* Michael Hemmer* Joseph B. M. Mitchell†
Christiane Schmidt*

Abstract

For a polygonal region P with n vertices, a *guard cover* S is a set of points in P, such that any point in P can be seen from a point in S. In a *colored guard cover*, every element in a guard cover is assigned a color, such that no two guards with the same color have overlapping visibility regions. The Chromatic Art Gallery Problem (CAGP) asks for the minimum number of colors for which a colored guard cover exists.

We discuss the CAGP for the case of only two colors. We show that it is already *NP*-hard to decide whether two colors suffice for covering a polygon with holes, even when arbitrary guard positions are allowed. For simple polygons with a discrete set of possible guard locations, we give a polynomial-time algorithm for deciding whether a two-colorable guard set exists. This algo-

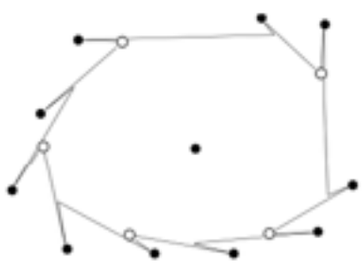


Figure 1: An example polygon with $n = 20$ vertices. A minimum-cardinality guard cover with $n/4$ guards (shown in white) requires $n/4$ colors, while a minimum-color guard cover (shown in black) has $n/2 + 1$ guards and requires only 3 colors.

Theorem 4.1 *For a simple polygon P with n vertices and ℓ discrete guard locations L, it can be decided in polynomial time whether there is a 2-colorable guard set.*

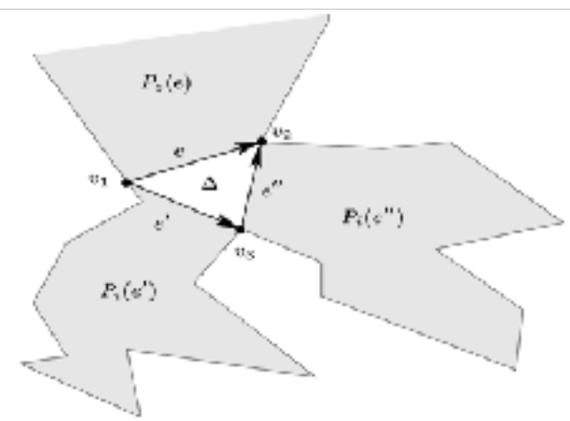


Figure 7: Solving a subproblem (e, Γ) by combining solutions for subproblems (e', Γ') and (e'', Γ'').

Overview

1. Introduction
2. Existence
3. Properties
4. Algorithms: Removing ears
5. Algorithms: Finding diagonals
6. Algorithms: Monotone polygons
7. Algorithms: Monotone decompositions
8. Faster algorithms
9. Application: Art Gallery problems
10. Application: Online triangulation

Algorithm 5.16

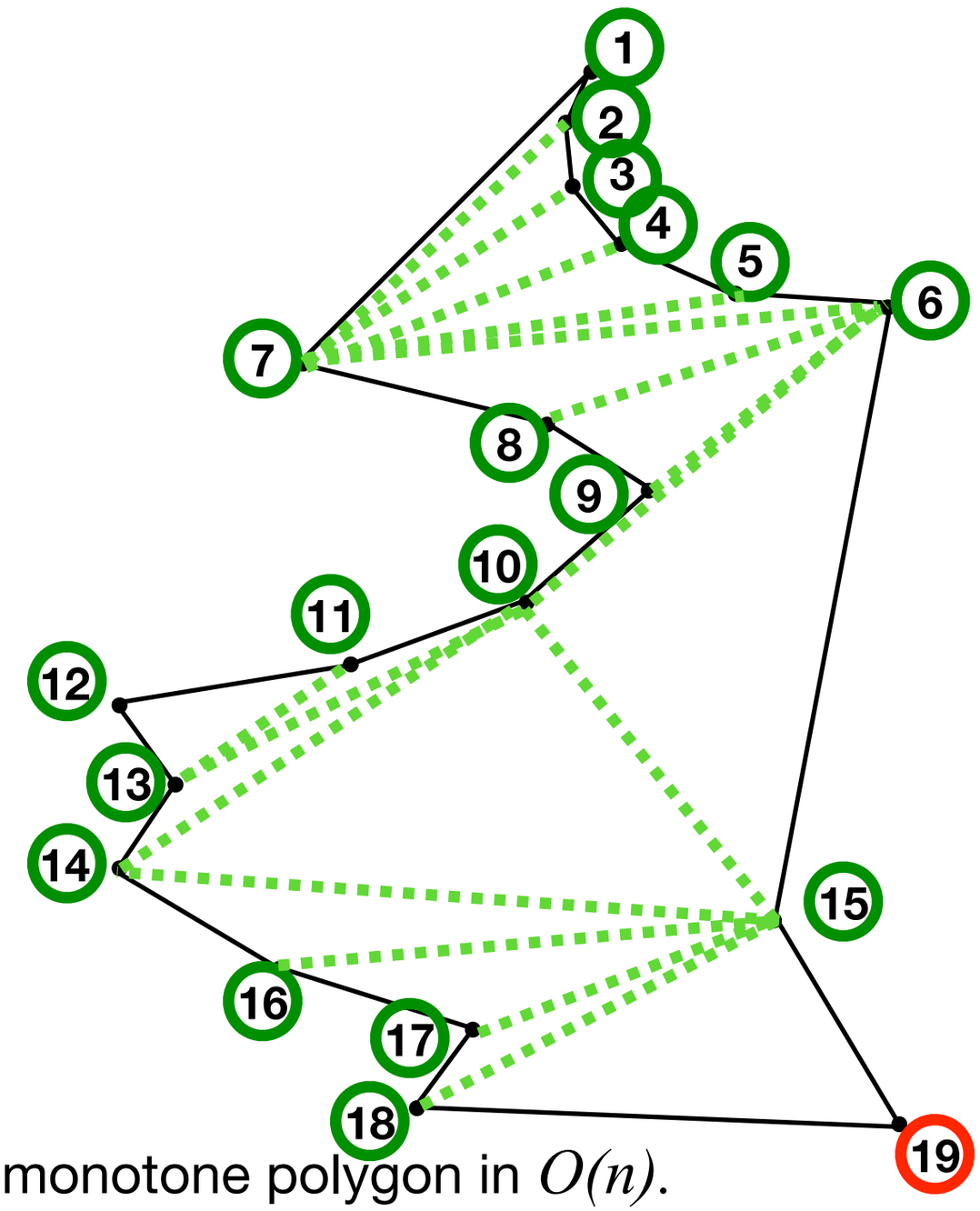
Input: A y-monotone polygon P

Output: A triangulation of P

1. Sort vertices by y-coordinate
2. Maintain queue of vertices still to be processed
   - First two vertices of unfinished subpolygon
   - Further vertices in same chain if predecessor reflex
3. Processing vertices in sorted order:
   3.1 If opposite chain:
   - Connect to all vertices in chain, leave in queue **Chain**

   3.2 If same chain, predecessor reflex:
   - Add to queue **Add**

   3.3 If same chain, predecessor convex:
   - Ear! Connect and update **Ear**



Theorem 5.17 Algorithm 5.16 can triangulate a y-monotone polygon in $O(n)$.

Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]

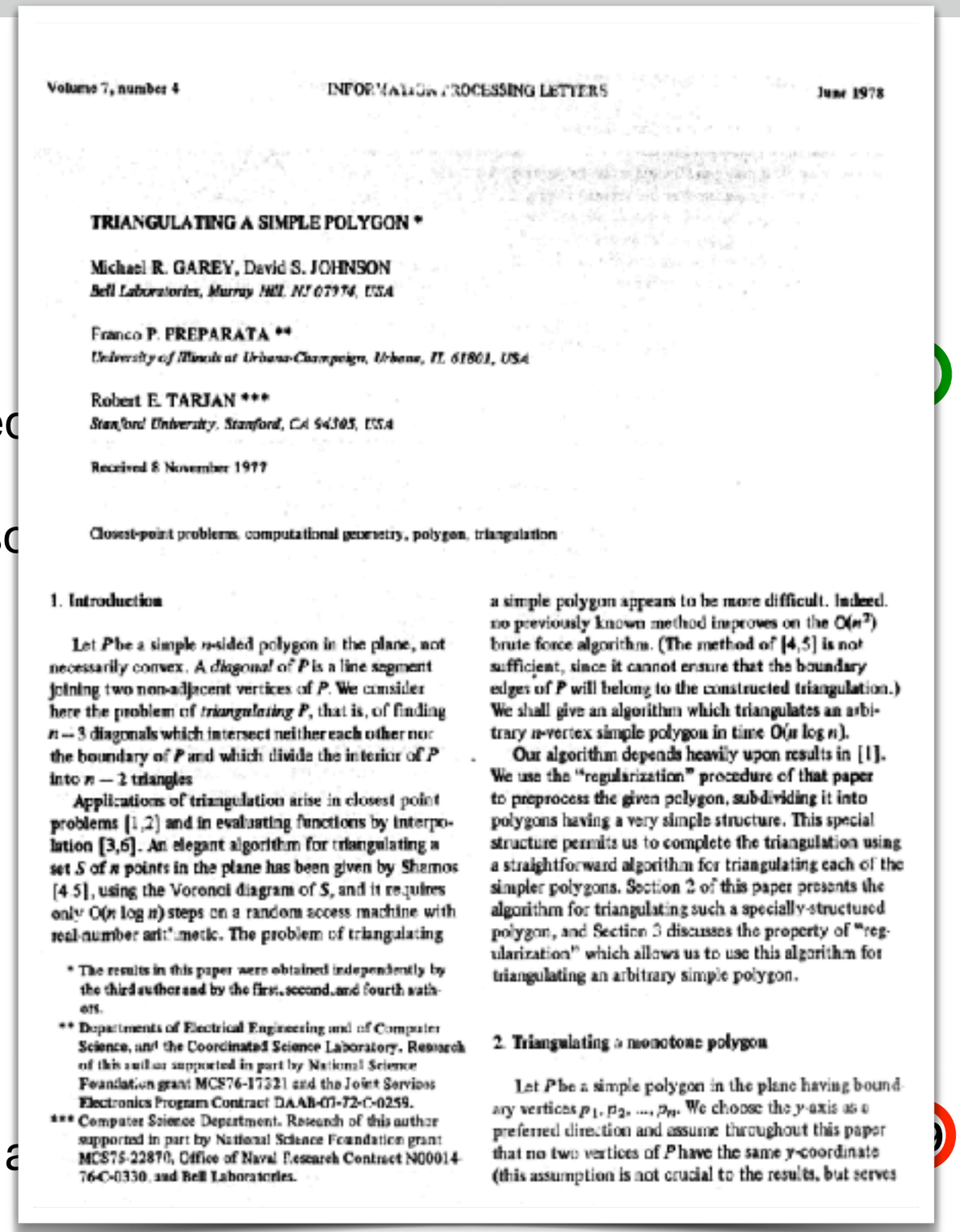
Algorithm 5.16

Input: A y-monotone polygon P

Output: A triangulation of P

1. Sort vertices by y-coordinate
2. Maintain queue of vertices still to be processed
   - First two vertices of unfinished subpolygon
   - Further vertices in same chain if predecessor reflex
3. Processing vertices in sorted order:
   - 3.1 If opposite chain:
     - Connect to all vertices in chain, leave in queue **Chain**
   - 3.2 If same chain, predecessor reflex:
     - Add to queue **Add**
   - 3.3 If same chain, predecessor convex:
     - Ear! Connect and update **Ear**

Theorem 5.17 Algorithm 5.16 can triangulate a



Volume 7, number 4 INFORMATION PROCESSING LETTERS June 1978

TRIANGULATING A SIMPLE POLYGON *

Michael R. GAREY, David S. JOHNSON
Bell Laboratories, Murray Hill, NJ 07974, USA

Franco P. PREPARATA **
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Robert E. TARJAN ***
Stanford University, Stanford, CA 94305, USA

Received 8 November 1977

Closest-point problems, computational geometry, polygon, triangulation

1. Introduction

Let P be a simple n-sided polygon in the plane, not necessarily convex. A *diagonal* of P is a line segment joining two non-adjacent vertices of P. We consider here the problem of *triangulating* P, that is, of finding $n-3$ diagonals which intersect neither each other nor the boundary of P and which divide the interior of P into $n-2$ triangles.

Applications of triangulation arise in closest point problems [1,2] and in evaluating functions by interpolation [3,6]. An elegant algorithm for triangulating a set S of n points in the plane has been given by Shamos [4,5], using the Voronoi diagram of S, and it requires only $O(n \log n)$ steps on a random access machine with real-number arithmetic. The problem of triangulating a simple polygon appears to be more difficult. Indeed, no previously known method improves on the $O(n^2)$ brute force algorithm. (The method of [4,5] is not sufficient, since it cannot ensure that the boundary edges of P will belong to the constructed triangulation.) We shall give an algorithm which triangulates an arbitrary n-vertex simple polygon in time $O(n \log n)$.

Our algorithm depends heavily upon results in [1]. We use the "regularization" procedure of that paper to preprocess the given polygon, subdividing it into polygons having a very simple structure. This special structure permits us to complete the triangulation using a straightforward algorithm for triangulating each of the simpler polygons. Section 2 of this paper presents the algorithm for triangulating such a specially-structured polygon, and Section 3 discusses the property of "regularization" which allows us to use this algorithm for triangulating an arbitrary simple polygon.

* The results in this paper were obtained independently by the third author and by the first, second, and fourth authors.

** Departments of Electrical Engineering and of Computer Science, and the Coordinated Science Laboratory. Research of this author supported in part by National Science Foundation grant MCS76-17321 and the Joint Services Electronics Program Contract DAAB-07-72-C-0259.

*** Computer Science Department. Research of this author supported in part by National Science Foundation grant MCS75-22870, Office of Naval Research Contract N00014-76-C-0330, and Bell Laboratories.

2. Triangulating a monotone polygon

Let P be a simple polygon in the plane having boundary vertices $p_1, p_2, \ldots, p_n$. We choose the y-axis as a preferred direction and assume throughout this paper that no two vertices of P have the same y-coordinate (this assumption is not crucial to the results, but serves

Algorithm 5.16

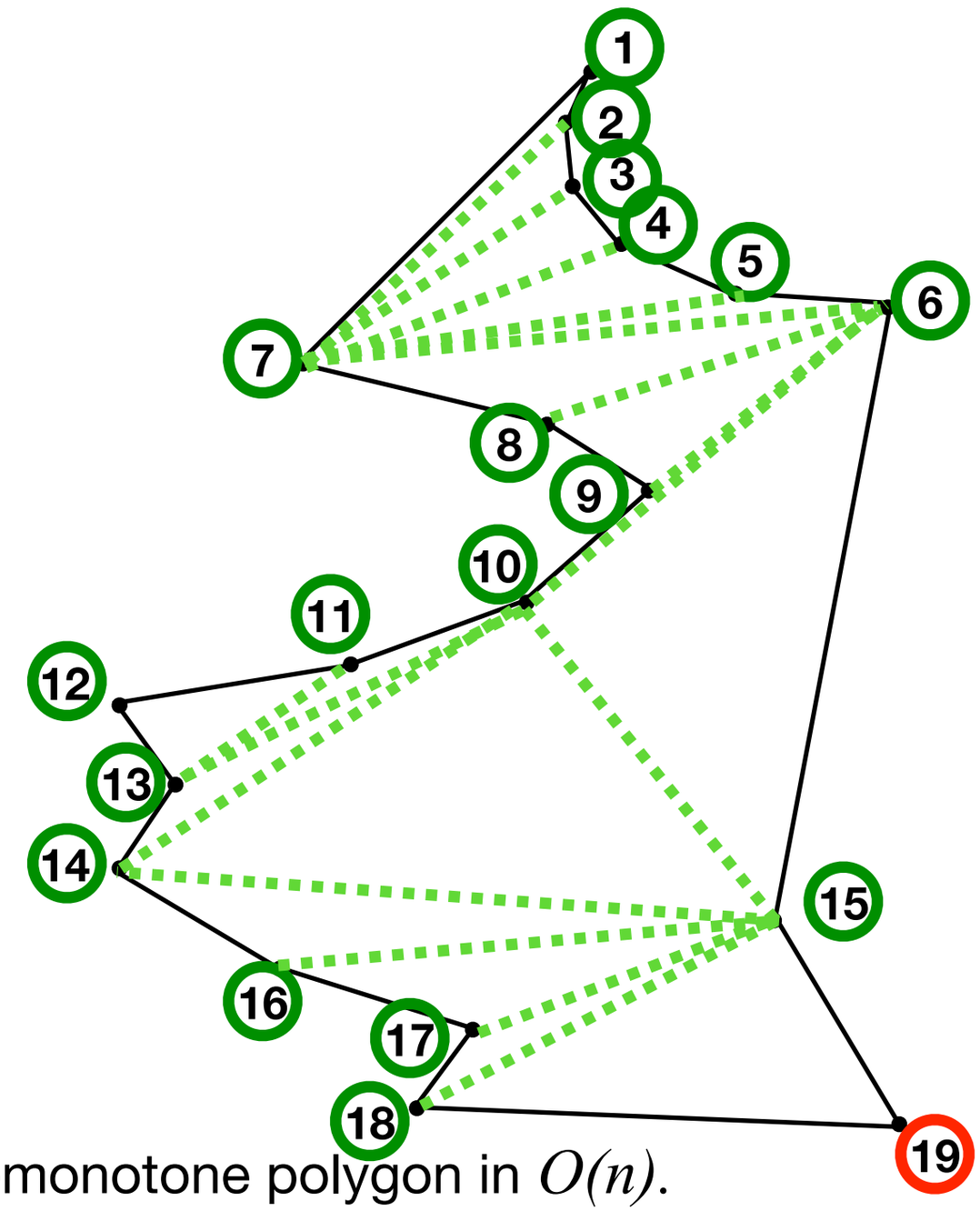
Input: A y-monotone polygon P

Output: A triangulation of P

1. Sort vertices by y-coordinate
2. Maintain queue of vertices still to be processed
   - First two vertices of unfinished subpolygon
   - Further vertices in same chain if predecessor reflex
3. Processing vertices in sorted order:
   3.1 If opposite chain:
   - Connect to all vertices in chain, leave in queue **Chain**

   3.2 If same chain, predecessor reflex:
   - Add to queue **Add**

   3.3 If same chain, predecessor convex:
   - Ear! Connect and update **Ear**



Theorem 5.17 Algorithm 5.16 can triangulate a y-monotone polygon in $O(n)$.

Overview

1. Introduction
2. Existence
3. Properties
4. Algorithms: Removing ears
5. Algorithms: Finding diagonals
6. Algorithms: Monotone polygons
7. Algorithms: Monotone decompositions
8. Faster algorithms
9. Application: Art Gallery problems
10. Application: Online triangulation

Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]

Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]

Volume 7, number 4 INFORMATION PROCESSING LETTERS June 1978

TRIANGULATING A SIMPLE POLYGON *

Michael R. GAREY, David S. JOHNSON
Bell Laboratories, Murray Hill, NJ 07974, USA

Franco P. PREPARATA **
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Robert E. TARJAN ***
Stanford University, Stanford, CA 94305, USA

Received 8 November 1977

Closest-point problems, computational geometry, polygon, triangulation

1. Introduction

Let P be a simple n-sided polygon in the plane, not necessarily convex. A *diagonal* of P is a line segment joining two non-adjacent vertices of P. We consider here the problem of *triangulating* P, that is, of finding $n-3$ diagonals which intersect neither each other nor the boundary of P and which divide the interior of P into $n-2$ triangles.

Applications of triangulation arise in closest point problems [1,2] and in evaluating functions by interpolation [3,6]. An elegant algorithm for triangulating a set S of n points in the plane has been given by Shamos [4,5], using the Voronoi diagram of S, and it requires only $O(n \log n)$ steps on a random access machine with real-number arithmetic. The problem of triangulating a simple polygon appears to be more difficult. Indeed, no previously known method improves on the $O(n^2)$ brute force algorithm. (The method of [4,5] is not sufficient, since it cannot ensure that the boundary edges of P will belong to the constructed triangulation.) We shall give an algorithm which triangulates an arbitrary n-vertex simple polygon in time $O(n \log n)$.

Our algorithm depends heavily upon results in [1]. We use the "regularization" procedure of that paper to preprocess the given polygon, subdividing it into polygons having a very simple structure. This special structure permits us to complete the triangulation using a straightforward algorithm for triangulating each of the simpler polygons. Section 2 of this paper presents the algorithm for triangulating such a specially-structured polygon, and Section 3 discusses the property of "regularization" which allows us to use this algorithm for triangulating an arbitrary simple polygon.

2. Triangulating a monotone polygon

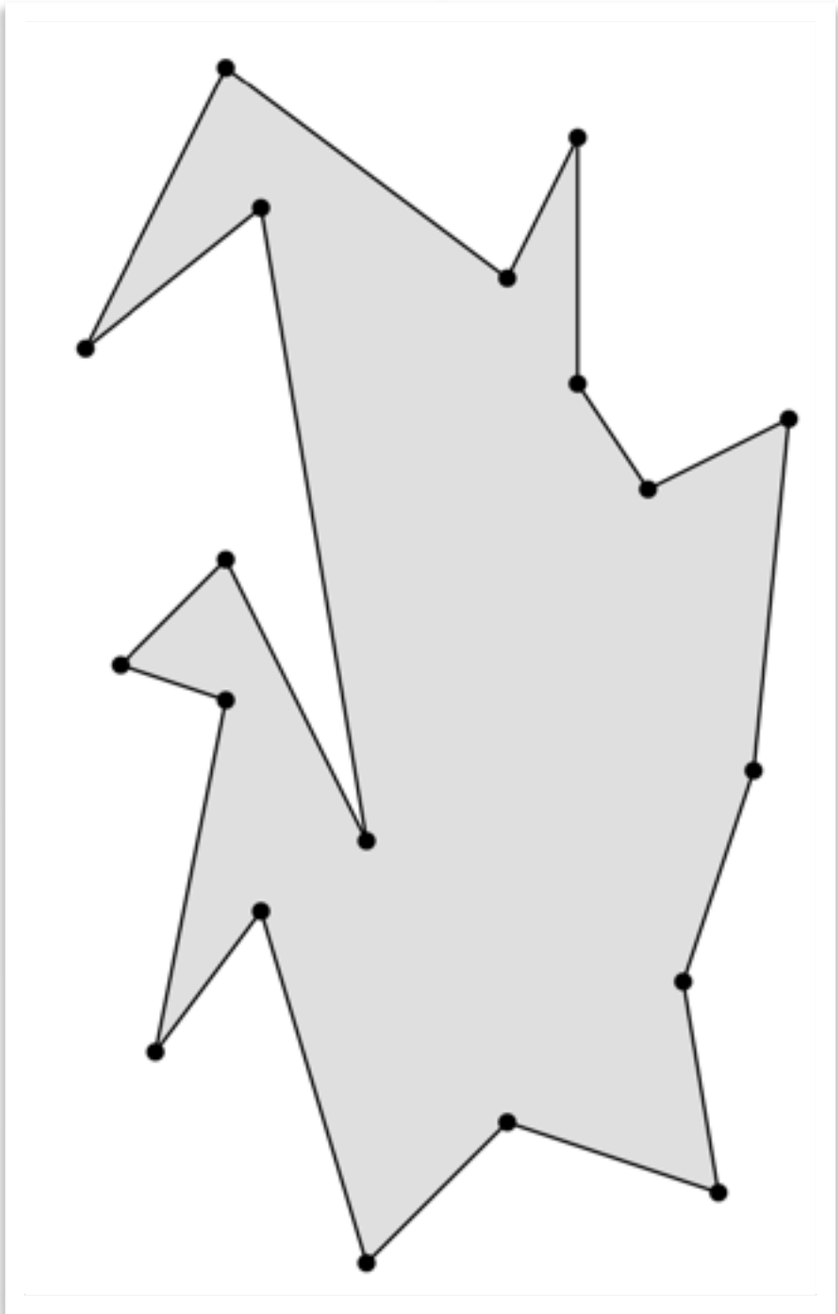
Let P be a simple polygon in the plane having boundary vertices $p_1, p_2, \ldots, p_n$. We choose the y-axis as a preferred direction and assume throughout this paper that no two vertices of P have the same y-coordinate (this assumption is not crucial to the results, but serves

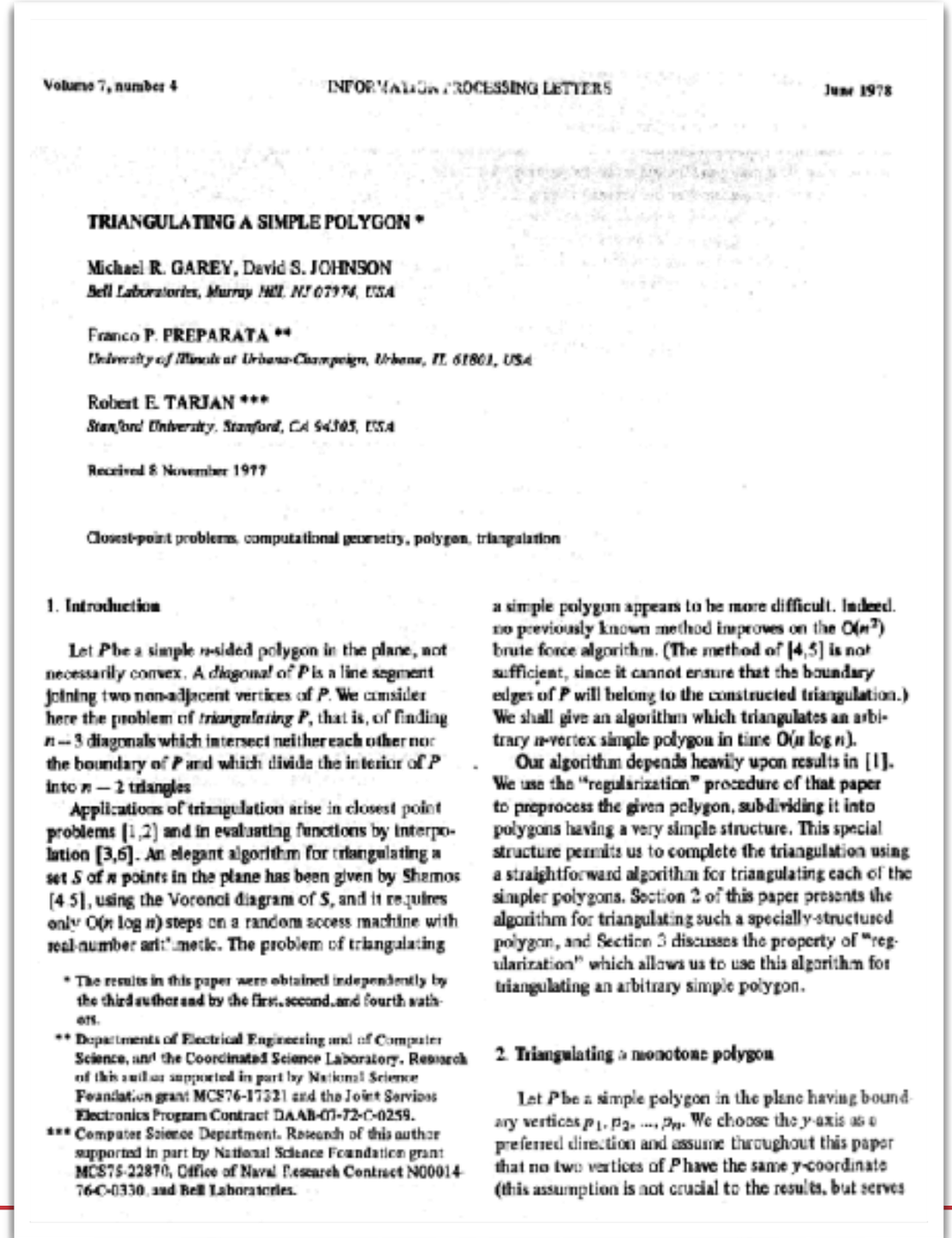
* The results in this paper were obtained independently by the third author and by the first, second, and fourth authors.

** Departments of Electrical Engineering and of Computer Science, and the Coordinated Science Laboratory. Research of this author supported in part by National Science Foundation grant MCS76-17321 and the Joint Services Electronics Program Contract DAAB-07-72-C-0259.

*** Computer Science Department. Research of this author supported in part by National Science Foundation grant MCS75-22870, Office of Naval Research Contract N00014-76-C-0330, and Bell Laboratories.

Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]





Volume 7, number 4 INFORMATION PROCESSING LETTERS June 1978

TRIANGULATING A SIMPLE POLYGON *

Michael R. GAREY, David S. JOHNSON
Bell Laboratories, Murray Hill, NJ 07974, USA

Franco P. PREPARATA **
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Robert E. TARJAN ***
Stanford University, Stanford, CA 94305, USA

Received 8 November 1977

Closest-point problems, computational geometry, polygon, triangulation

1. Introduction

Let P be a simple n-sided polygon in the plane, not necessarily convex. A *diagonal* of P is a line segment joining two non-adjacent vertices of P. We consider here the problem of *triangulating* P, that is, of finding $n-3$ diagonals which intersect neither each other nor the boundary of P and which divide the interior of P into $n-2$ triangles.

Applications of triangulation arise in closest point problems [1,2] and in evaluating functions by interpolation [3,6]. An elegant algorithm for triangulating a set S of n points in the plane has been given by Shamos [4,5], using the Voronoi diagram of S, and it requires only $O(n \log n)$ steps on a random access machine with real-number arithmetic. The problem of triangulating a simple polygon appears to be more difficult. Indeed, no previously known method improves on the $O(n^2)$ brute force algorithm. (The method of [4,5] is not sufficient, since it cannot ensure that the boundary edges of P will belong to the constructed triangulation.) We shall give an algorithm which triangulates an arbitrary n-vertex simple polygon in time $O(n \log n)$.

Our algorithm depends heavily upon results in [1]. We use the "regularization" procedure of that paper to preprocess the given polygon, subdividing it into polygons having a very simple structure. This special structure permits us to complete the triangulation using a straightforward algorithm for triangulating each of the simpler polygons. Section 2 of this paper presents the algorithm for triangulating such a specially-structured polygon, and Section 3 discusses the property of "regularization" which allows us to use this algorithm for triangulating an arbitrary simple polygon.

* The results in this paper were obtained independently by the third author and by the first, second, and fourth authors.

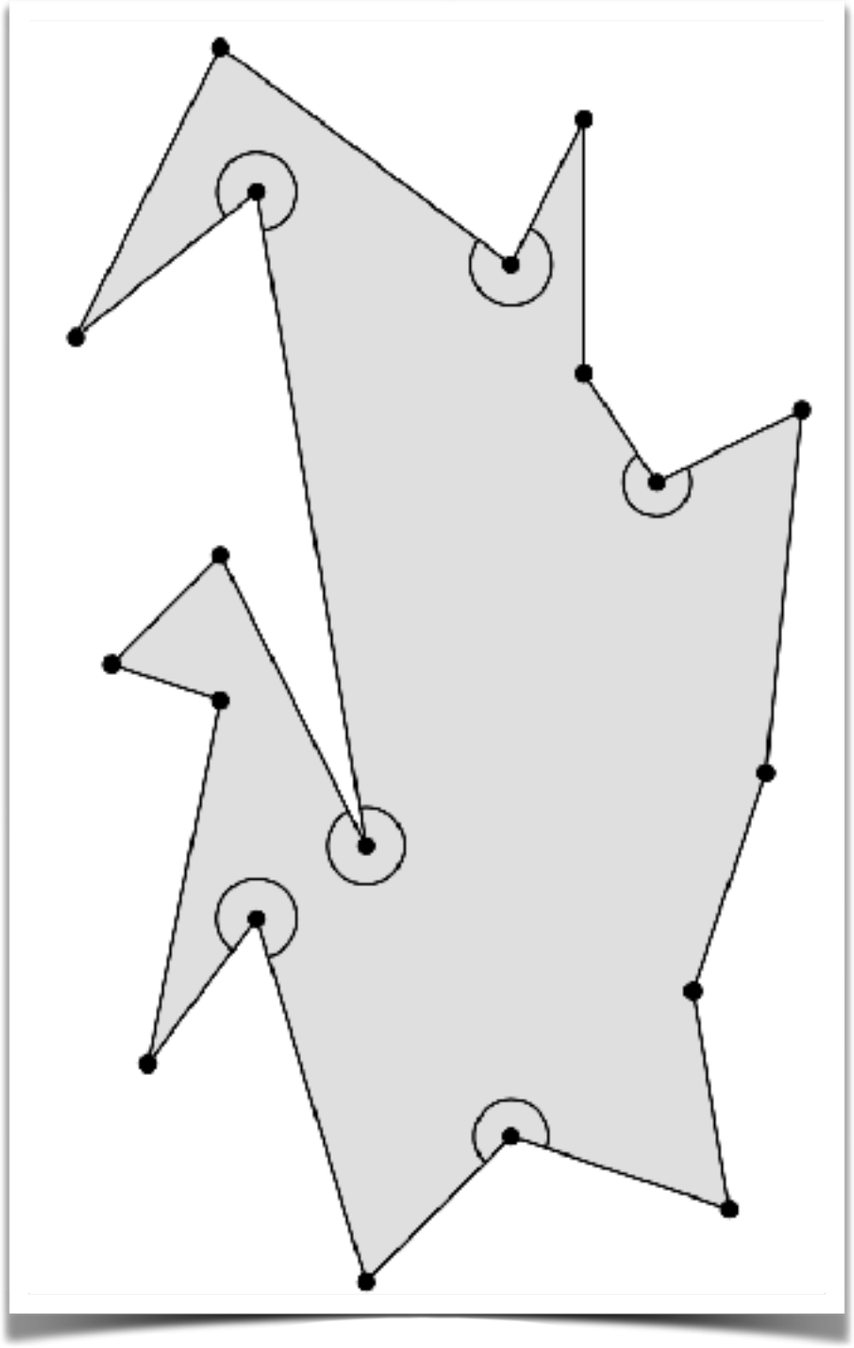
** Departments of Electrical Engineering and of Computer Science, and the Coordinated Science Laboratory. Research of this author supported in part by National Science Foundation grant MCS76-17321 and the Joint Services Electronics Program Contract DAAB-07-72-C-0259.

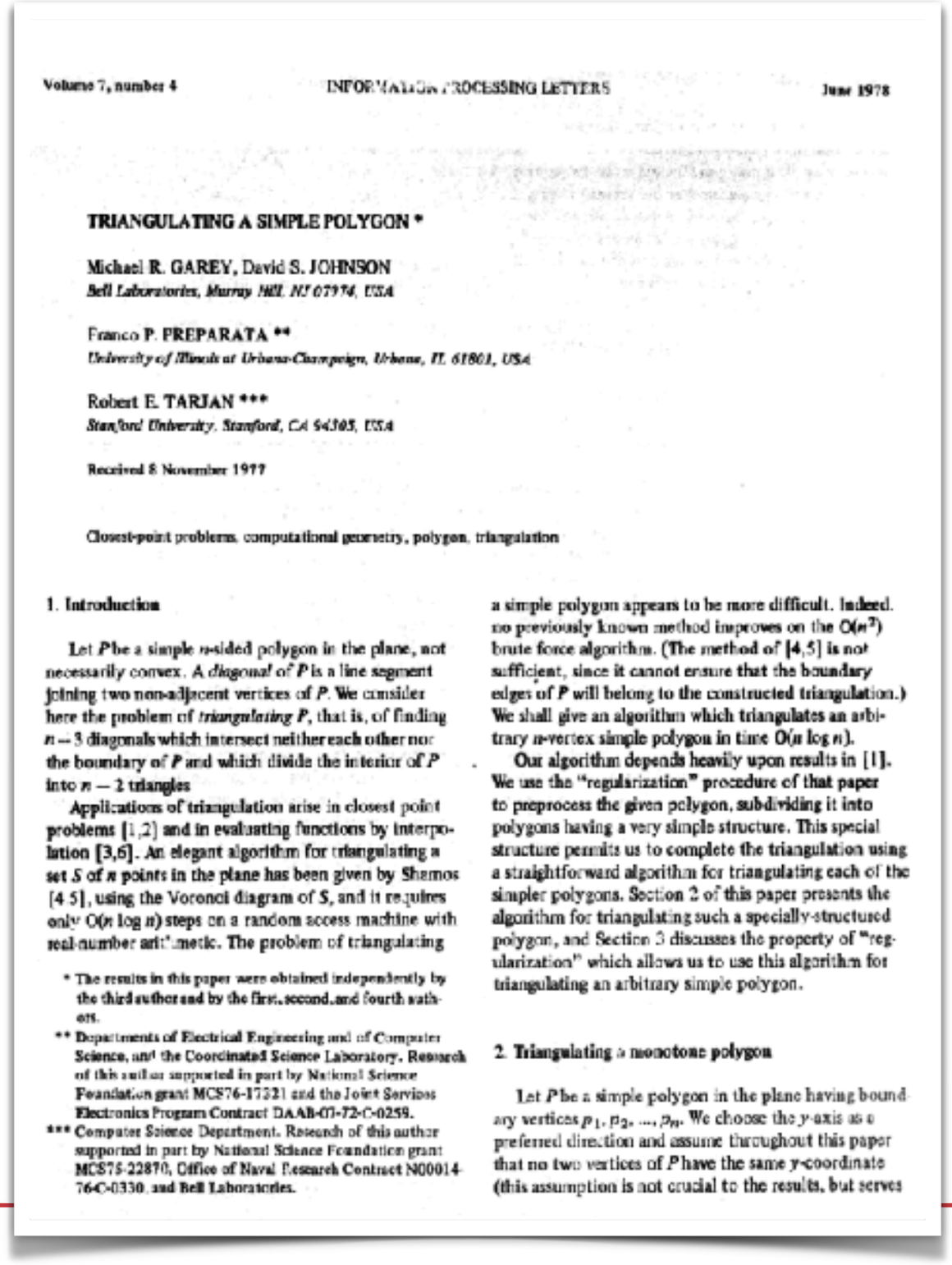
*** Computer Science Department. Research of this author supported in part by National Science Foundation grant MCS75-22870, Office of Naval Research Contract N00014-76-C-0330, and Bell Laboratories.

2. Triangulating a monotone polygon

Let P be a simple polygon in the plane having boundary vertices $p_1, p_2, \ldots, p_n$. We choose the y-axis as a preferred direction and assume throughout this paper that no two vertices of P have the same y-coordinate (this assumption is not crucial to the results, but serves

Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]





Volume 7, number 4 INFORMATION PROCESSING LETTERS June 1978

TRIANGULATING A SIMPLE POLYGON *

Michael R. GAREY, David S. JOHNSON
Bell Laboratories, Murray Hill, NJ 07974, USA

Franco P. PREPARATA **
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Robert E. TARJAN ***
Stanford University, Stanford, CA 94305, USA

Received 8 November 1977

Closest-point problems, computational geometry, polygon, triangulation

1. Introduction

Let P be a simple n-sided polygon in the plane, not necessarily convex. A *diagonal* of P is a line segment joining two non-adjacent vertices of P. We consider here the problem of *triangulating* P, that is, of finding $n-3$ diagonals which intersect neither each other nor the boundary of P and which divide the interior of P into $n-2$ triangles.

Applications of triangulation arise in closest point problems [1,2] and in evaluating functions by interpolation [3,6]. An elegant algorithm for triangulating a set S of n points in the plane has been given by Shamos [4,5], using the Voronoi diagram of S, and it requires only $O(n \log n)$ steps on a random access machine with real-number arithmetic. The problem of triangulating a simple polygon appears to be more difficult. Indeed, no previously known method improves on the $O(n^2)$ brute force algorithm. (The method of [4,5] is not sufficient, since it cannot ensure that the boundary edges of P will belong to the constructed triangulation.) We shall give an algorithm which triangulates an arbitrary n-vertex simple polygon in time $O(n \log n)$.

Our algorithm depends heavily upon results in [1]. We use the "regularization" procedure of that paper to preprocess the given polygon, subdividing it into polygons having a very simple structure. This special structure permits us to complete the triangulation using a straightforward algorithm for triangulating each of the simpler polygons. Section 2 of this paper presents the algorithm for triangulating such a specially-structured polygon, and Section 3 discusses the property of "regularization" which allows us to use this algorithm for triangulating an arbitrary simple polygon.

2. Triangulating a monotone polygon

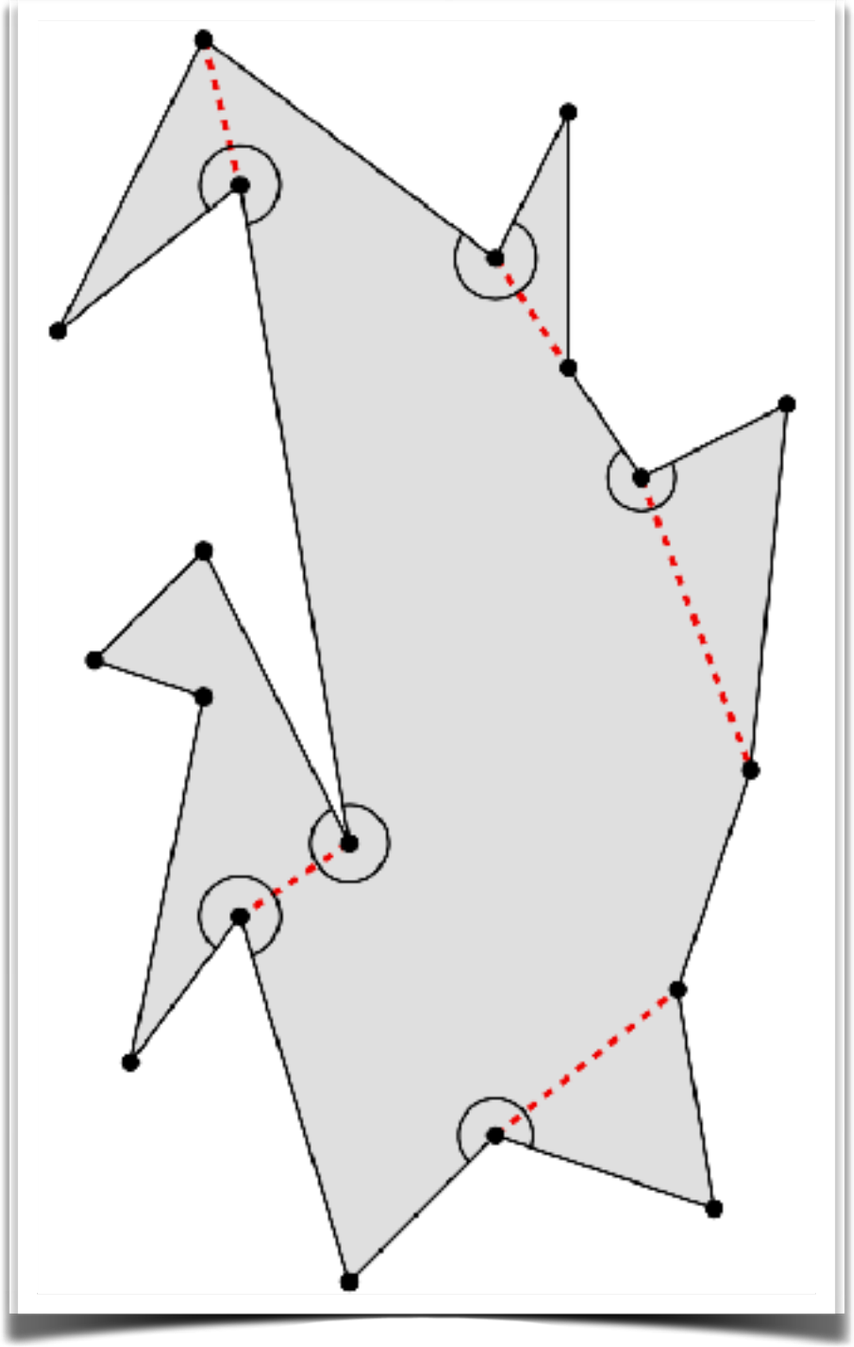
Let P be a simple polygon in the plane having boundary vertices $p_1, p_2, \ldots, p_n$. We choose the y-axis as a preferred direction and assume throughout this paper that no two vertices of P have the same y-coordinate (this assumption is not crucial to the results, but serves

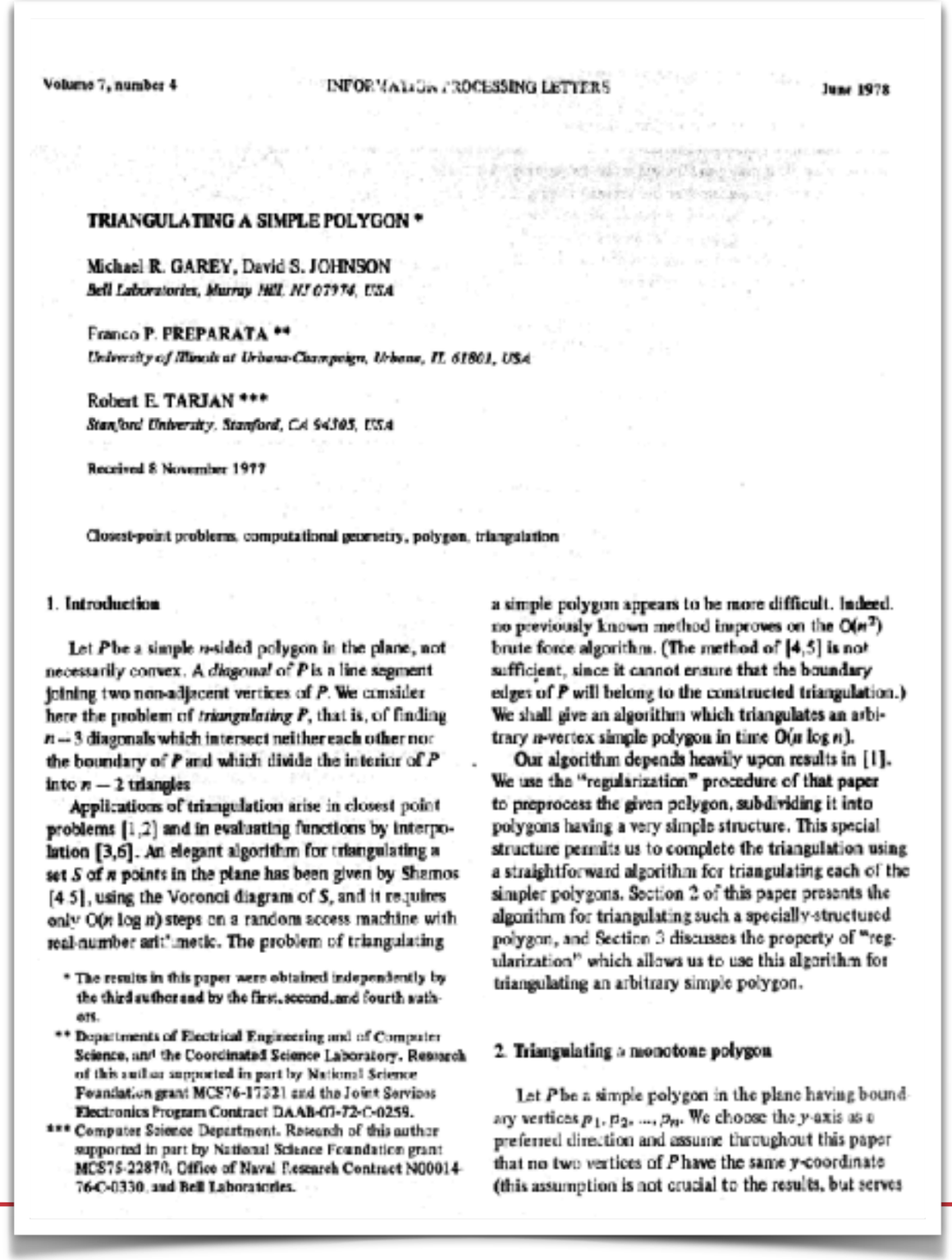
* The results in this paper were obtained independently by the third author and by the first, second, and fourth authors.

** Departments of Electrical Engineering and of Computer Science, and the Coordinated Science Laboratory. Research of this author supported in part by National Science Foundation grant MCS76-17321 and the Joint Services Electronics Program Contract DAAB-07-72-C-0259.

*** Computer Science Department. Research of this author supported in part by National Science Foundation grant MCS75-22870, Office of Naval Research Contract N00014-76-C-0330, and Bell Laboratories.

Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]





Volume 7, number 4 INFORMATION PROCESSING LETTERS June 1978

TRIANGULATING A SIMPLE POLYGON *

Michael R. GAREY, David S. JOHNSON
Bell Laboratories, Murray Hill, NJ 07974, USA

Franco P. PREPARATA **
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Robert E. TARJAN ***
Stanford University, Stanford, CA 94305, USA

Received 8 November 1977

Closest-point problems, computational geometry, polygon, triangulation

1. Introduction

Let P be a simple n-sided polygon in the plane, not necessarily convex. A *diagonal* of P is a line segment joining two non-adjacent vertices of P. We consider here the problem of *triangulating* P, that is, of finding $n-3$ diagonals which intersect neither each other nor the boundary of P and which divide the interior of P into $n-2$ triangles.

Applications of triangulation arise in closest point problems [1,2] and in evaluating functions by interpolation [3,6]. An elegant algorithm for triangulating a set S of n points in the plane has been given by Shamos [4,5], using the Voronoi diagram of S, and it requires only $O(n \log n)$ steps on a random access machine with real-number arithmetic. The problem of triangulating a simple polygon appears to be more difficult. Indeed, no previously known method improves on the $O(n^2)$ brute force algorithm. (The method of [4,5] is not sufficient, since it cannot ensure that the boundary edges of P will belong to the constructed triangulation.) We shall give an algorithm which triangulates an arbitrary n-vertex simple polygon in time $O(n \log n)$.

Our algorithm depends heavily upon results in [1]. We use the "regularization" procedure of that paper to preprocess the given polygon, subdividing it into polygons having a very simple structure. This special structure permits us to complete the triangulation using a straightforward algorithm for triangulating each of the simpler polygons. Section 2 of this paper presents the algorithm for triangulating such a specially-structured polygon, and Section 3 discusses the property of "regularization" which allows us to use this algorithm for triangulating an arbitrary simple polygon.

2. Triangulating a monotone polygon

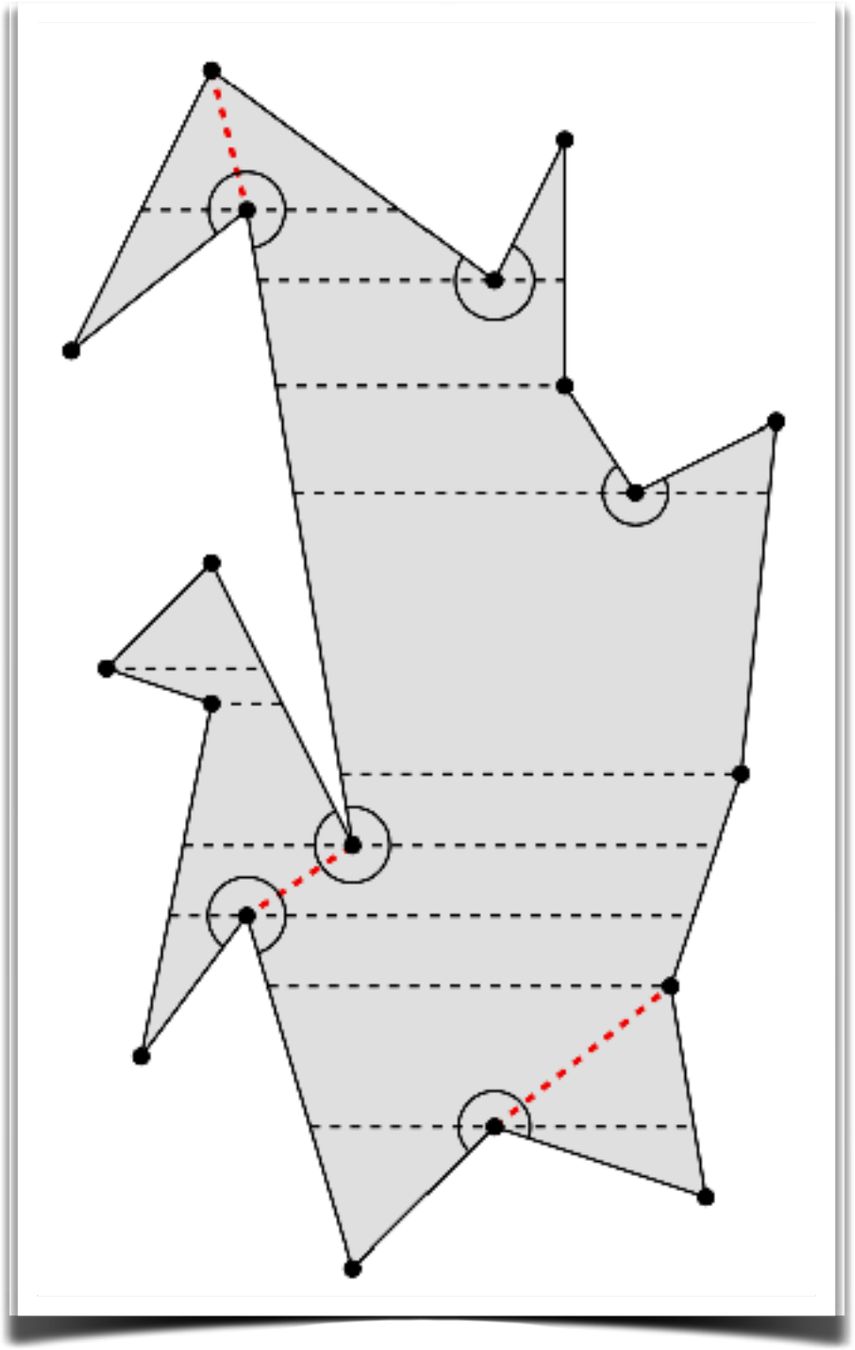
Let P be a simple polygon in the plane having boundary vertices $p_1, p_2, \ldots, p_n$. We choose the y-axis as a preferred direction and assume throughout this paper that no two vertices of P have the same y-coordinate (this assumption is not crucial to the results, but serves

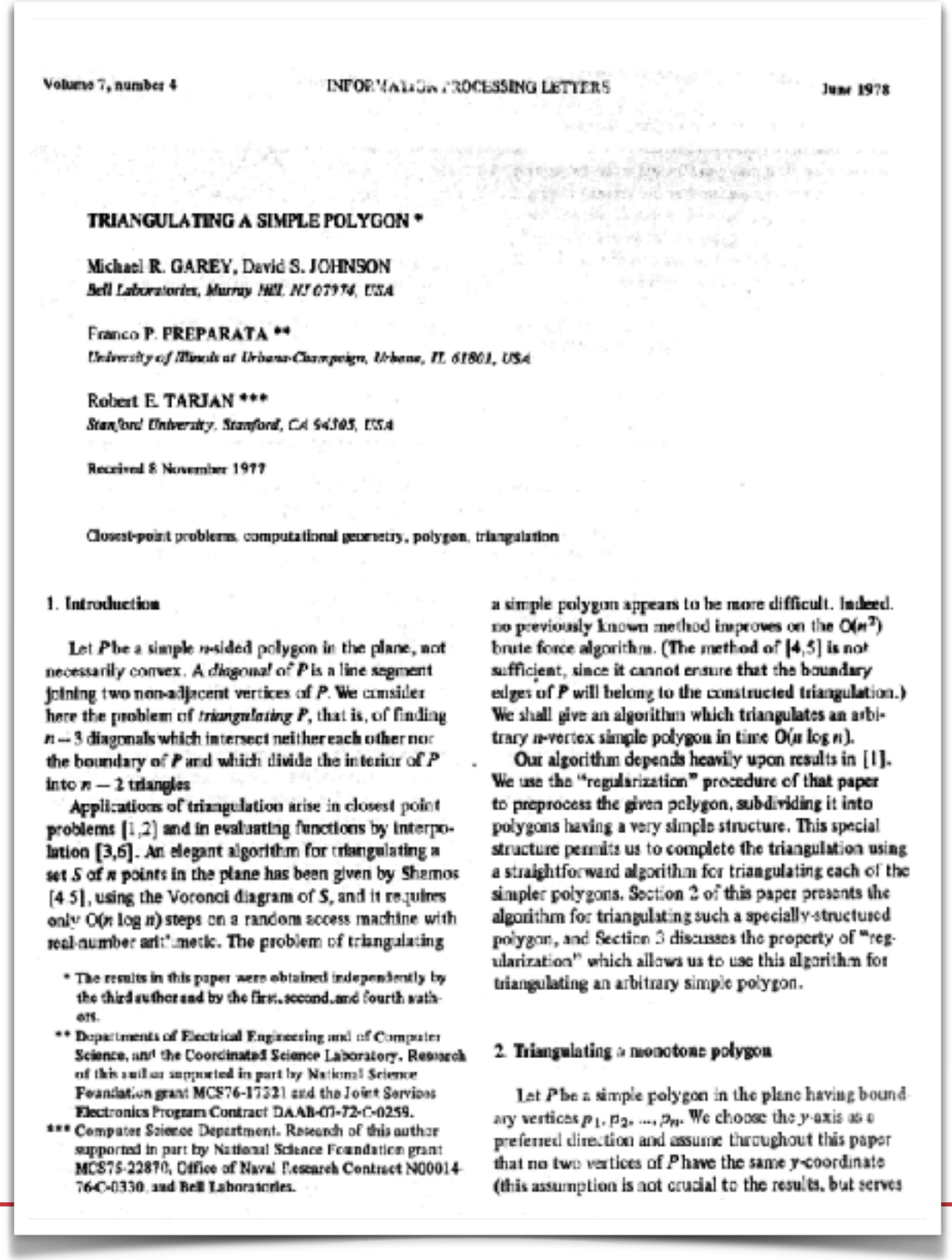
* The results in this paper were obtained independently by the third author and by the first, second, and fourth authors.

** Departments of Electrical Engineering and of Computer Science, and the Coordinated Science Laboratory. Research of this author supported in part by National Science Foundation grant MCS76-17321 and the Joint Services Electronics Program Contract DAAB-07-72-C-0259.

*** Computer Science Department. Research of this author supported in part by National Science Foundation grant MCS75-22870, Office of Naval Research Contract N00014-76-C-0330, and Bell Laboratories.

Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]





Volume 7, number 4 INFORMATION PROCESSING LETTERS June 1978

TRIANGULATING A SIMPLE POLYGON *

Michael R. GAREY, David S. JOHNSON
Bell Laboratories, Murray Hill, NJ 07974, USA

Franco P. PREPARATA **
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Robert E. TARJAN ***
Stanford University, Stanford, CA 94305, USA

Received 8 November 1977

Closest-point problems, computational geometry, polygon, triangulation

1. Introduction

Let P be a simple n-sided polygon in the plane, not necessarily convex. A *diagonal* of P is a line segment joining two non-adjacent vertices of P. We consider here the problem of *triangulating* P, that is, of finding $n-3$ diagonals which intersect neither each other nor the boundary of P and which divide the interior of P into $n-2$ triangles.

Applications of triangulation arise in closest point problems [1,2] and in evaluating functions by interpolation [3,6]. An elegant algorithm for triangulating a set S of n points in the plane has been given by Shamos [4,5], using the Voronoi diagram of S, and it requires only $O(n \log n)$ steps on a random access machine with real-number arithmetic. The problem of triangulating a simple polygon appears to be more difficult. Indeed, no previously known method improves on the $O(n^2)$ brute force algorithm. (The method of [4,5] is not sufficient, since it cannot ensure that the boundary edges of P will belong to the constructed triangulation.) We shall give an algorithm which triangulates an arbitrary n-vertex simple polygon in time $O(n \log n)$.

Our algorithm depends heavily upon results in [1]. We use the "regularization" procedure of that paper to preprocess the given polygon, subdividing it into polygons having a very simple structure. This special structure permits us to complete the triangulation using a straightforward algorithm for triangulating each of the simpler polygons. Section 2 of this paper presents the algorithm for triangulating such a specially-structured polygon, and Section 3 discusses the property of "regularization" which allows us to use this algorithm for triangulating an arbitrary simple polygon.

2. Triangulating a monotone polygon

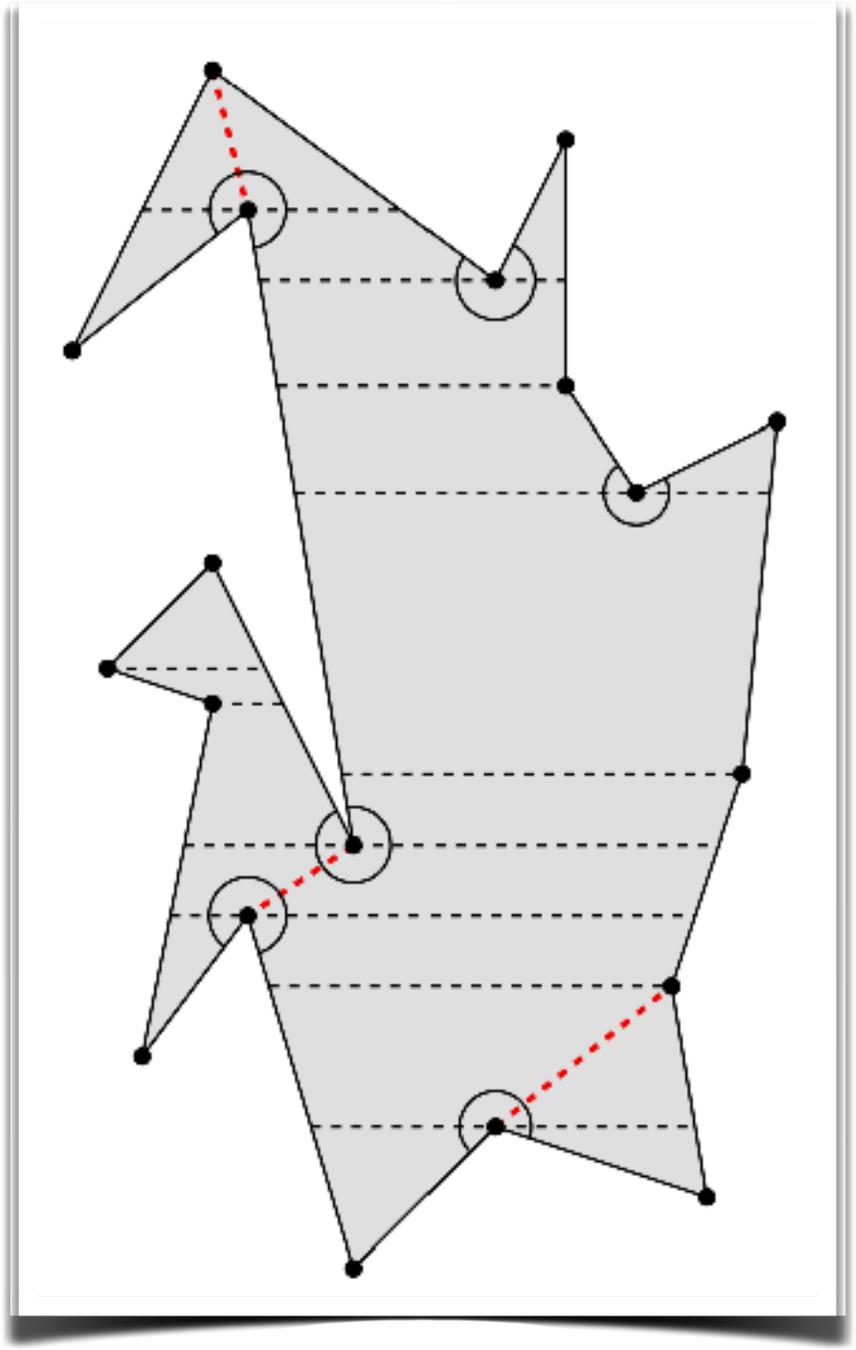
Let P be a simple polygon in the plane having boundary vertices $p_1, p_2, \ldots, p_n$. We choose the y-axis as a preferred direction and assume throughout this paper that no two vertices of P have the same y-coordinate (this assumption is not crucial to the results, but serves

* The results in this paper were obtained independently by the third author and by the first, second, and fourth authors.

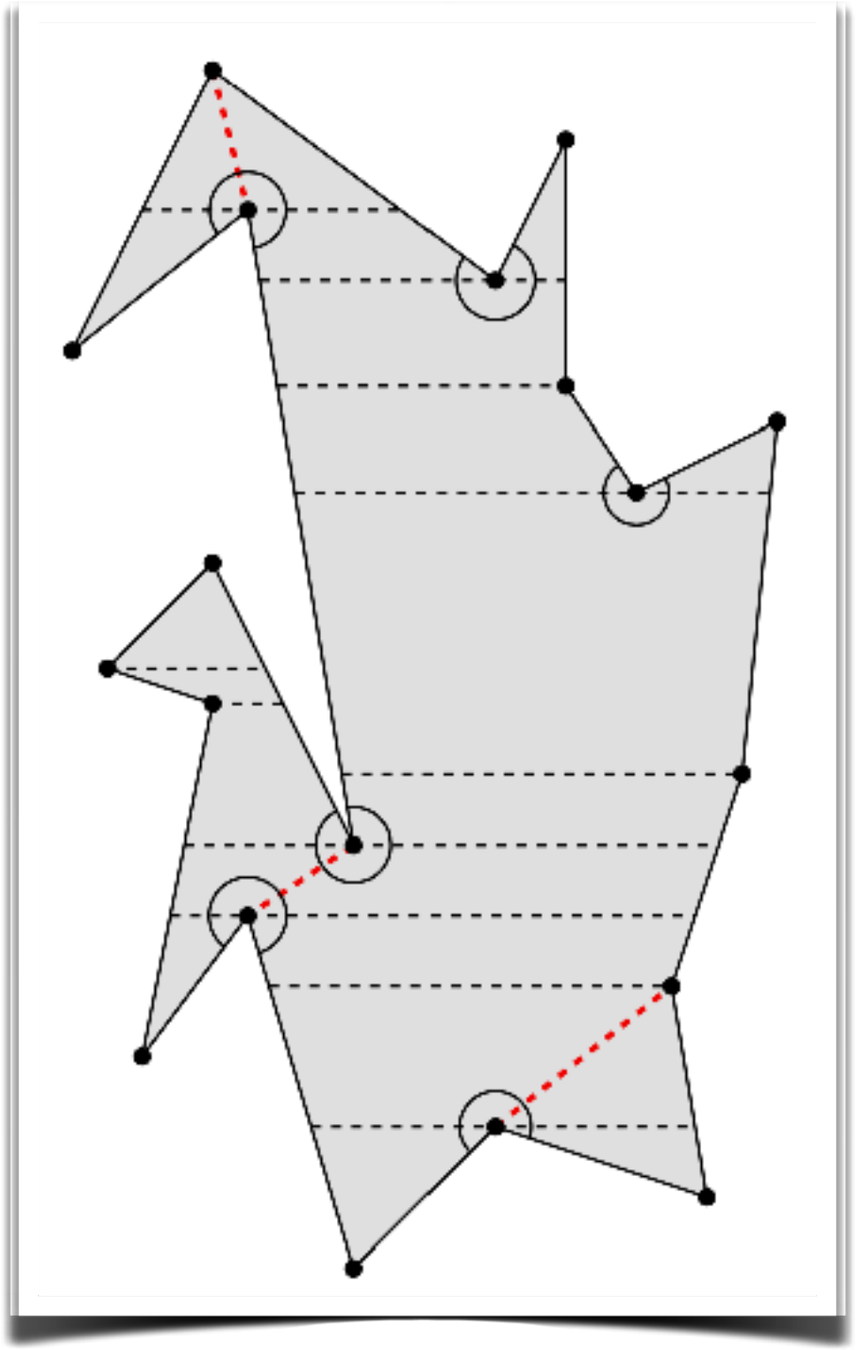
** Departments of Electrical Engineering and of Computer Science, and the Coordinated Science Laboratory. Research of this author supported in part by National Science Foundation grant MCS76-17321 and the Joint Services Electronics Program Contract DAAB-07-72-C-0259.

*** Computer Science Department. Research of this author supported in part by National Science Foundation grant MCS75-22870, Office of Naval Research Contract N00014-76-C-0330, and Bell Laboratories.

Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]



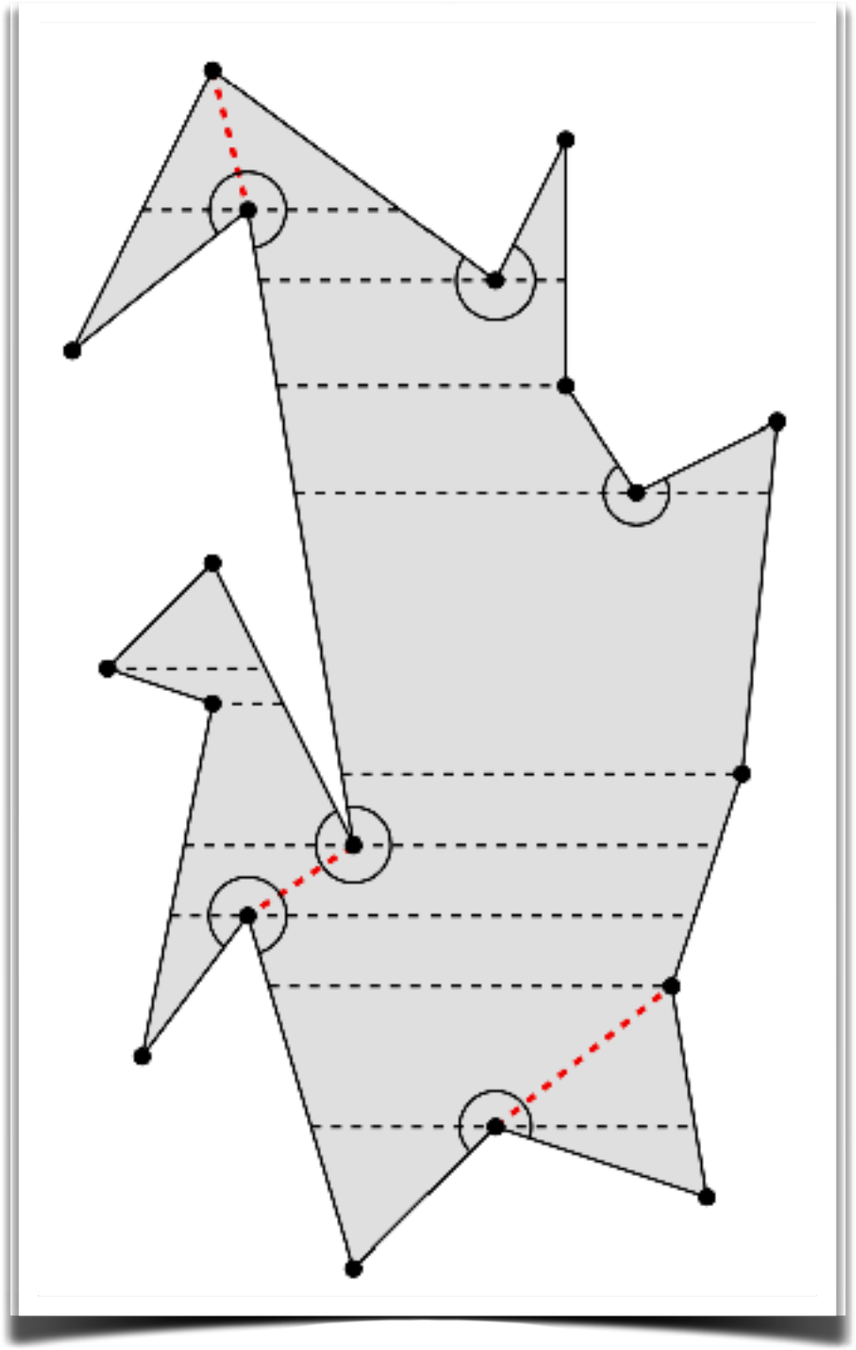
Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]



Algorithm 5.18:

Partitioning into monotone subpolygons

Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]

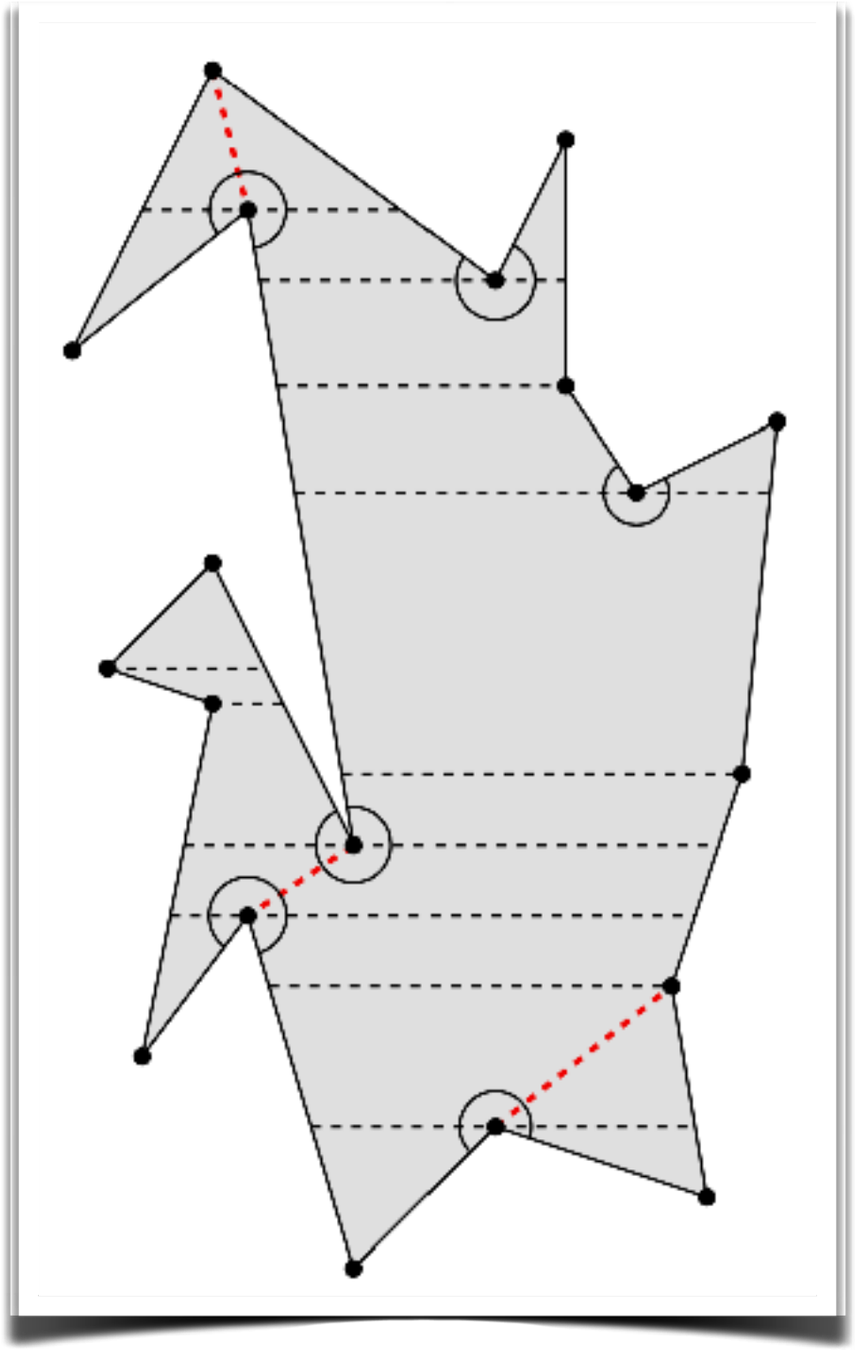


Algorithm 5.18:

Partitioning into monotone subpolygons

Ideas:

Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]



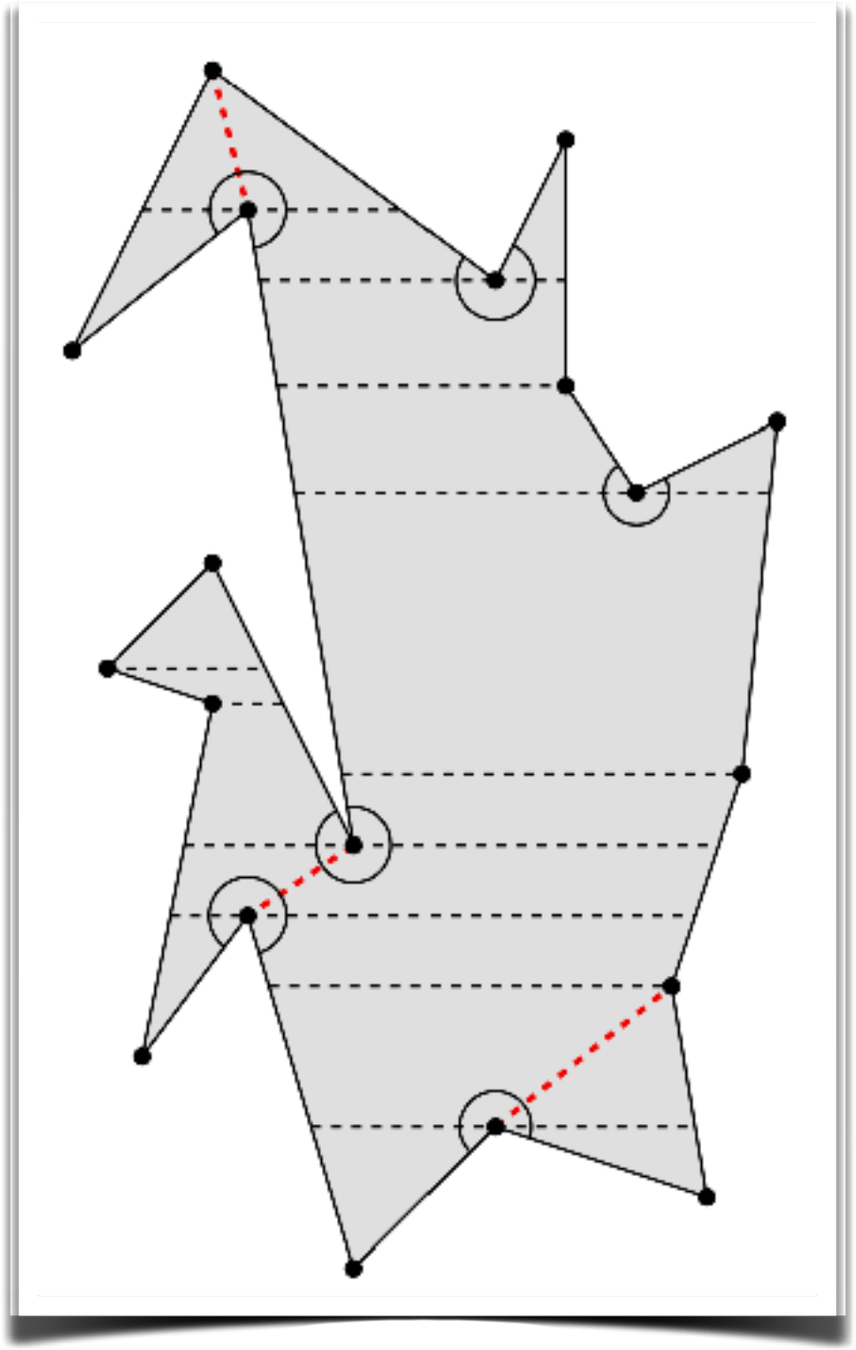
Algorithm 5.18:

Partitioning into monotone subpolygons

Ideas:

- Vertical plane sweep through vertices

Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]



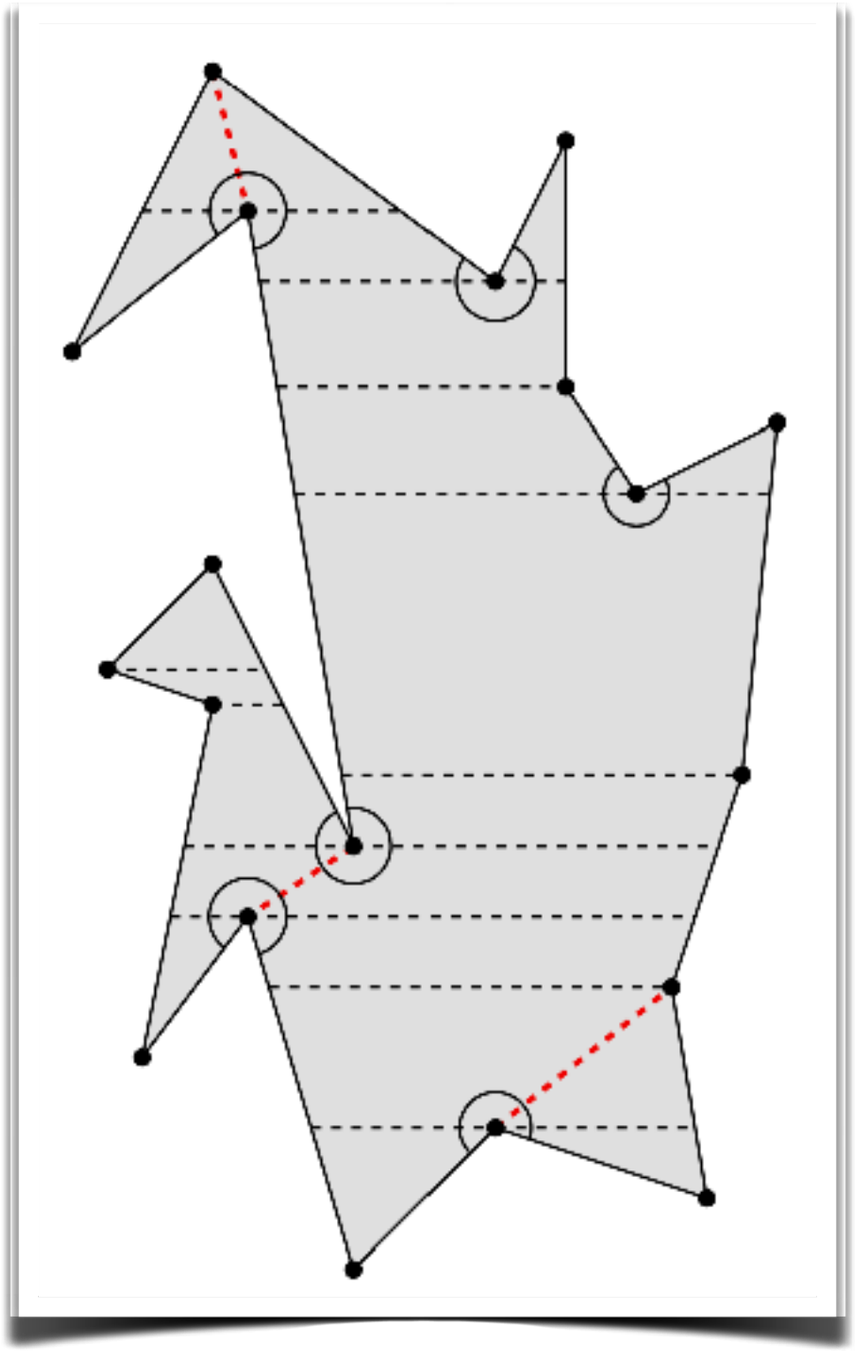
Algorithm 5.18:

Partitioning into monotone subpolygons

Ideas:

- Vertical plane sweep through vertices
- Maintain horizontal sequence of edges

Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]

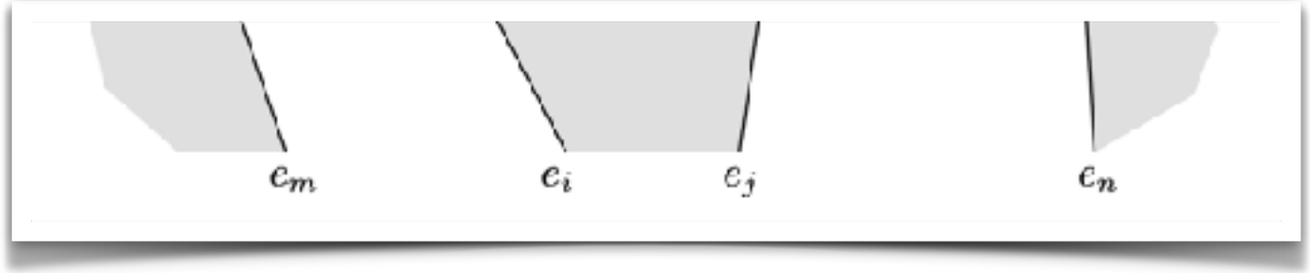


Algorithm 5.18:

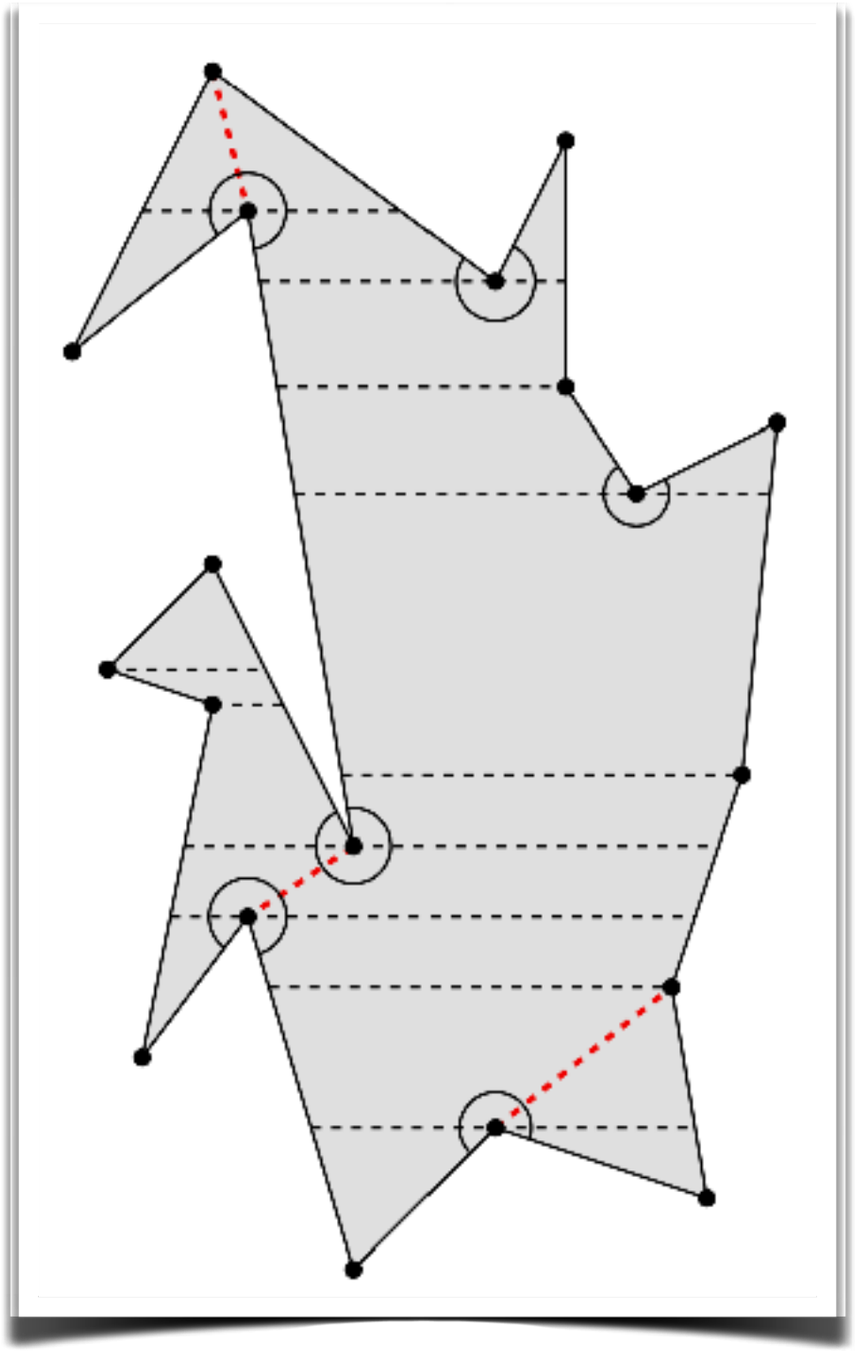
Partitioning into monotone subpolygons

Ideas:

- Vertical plane sweep through vertices
- Maintain horizontal sequence of edges



Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]

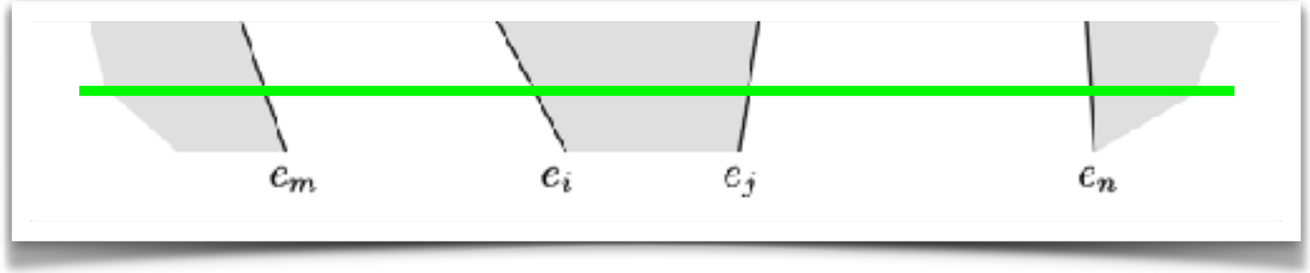


Algorithm 5.18:

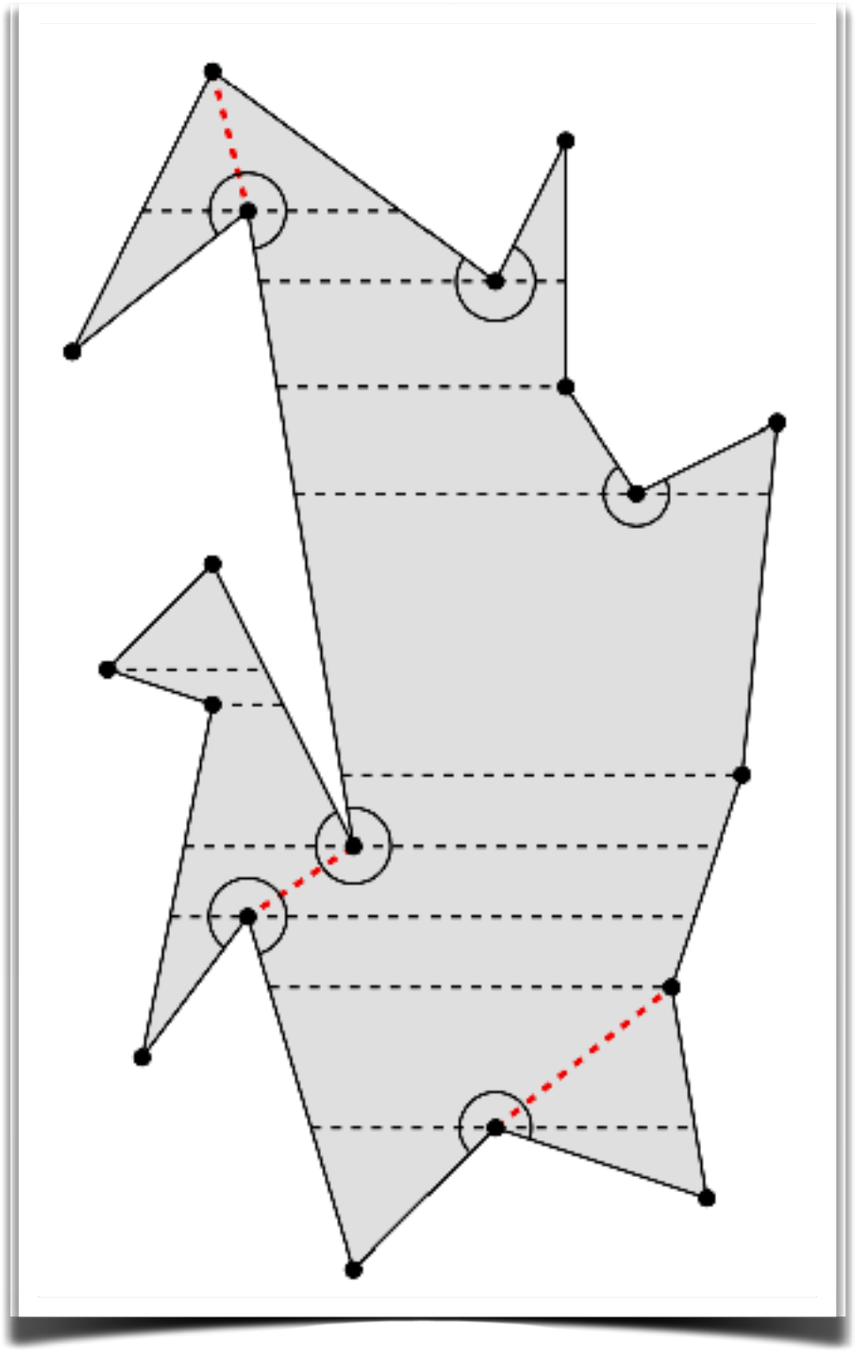
Partitioning into monotone subpolygons

Ideas:

- Vertical plane sweep through vertices
- Maintain horizontal sequence of edges



Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]

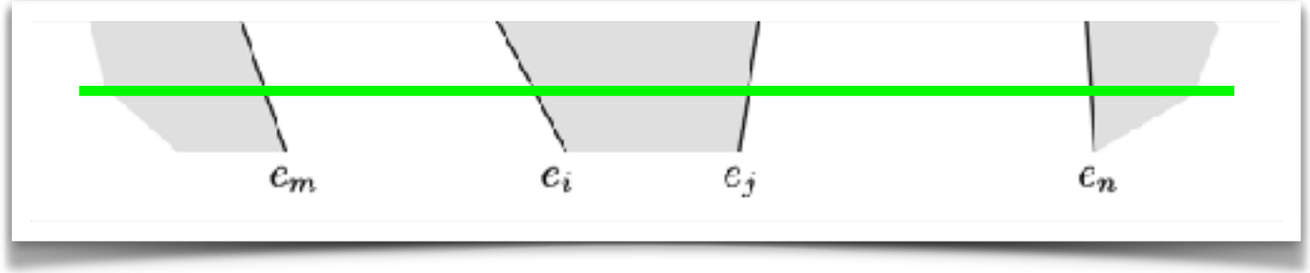


Algorithm 5.18:

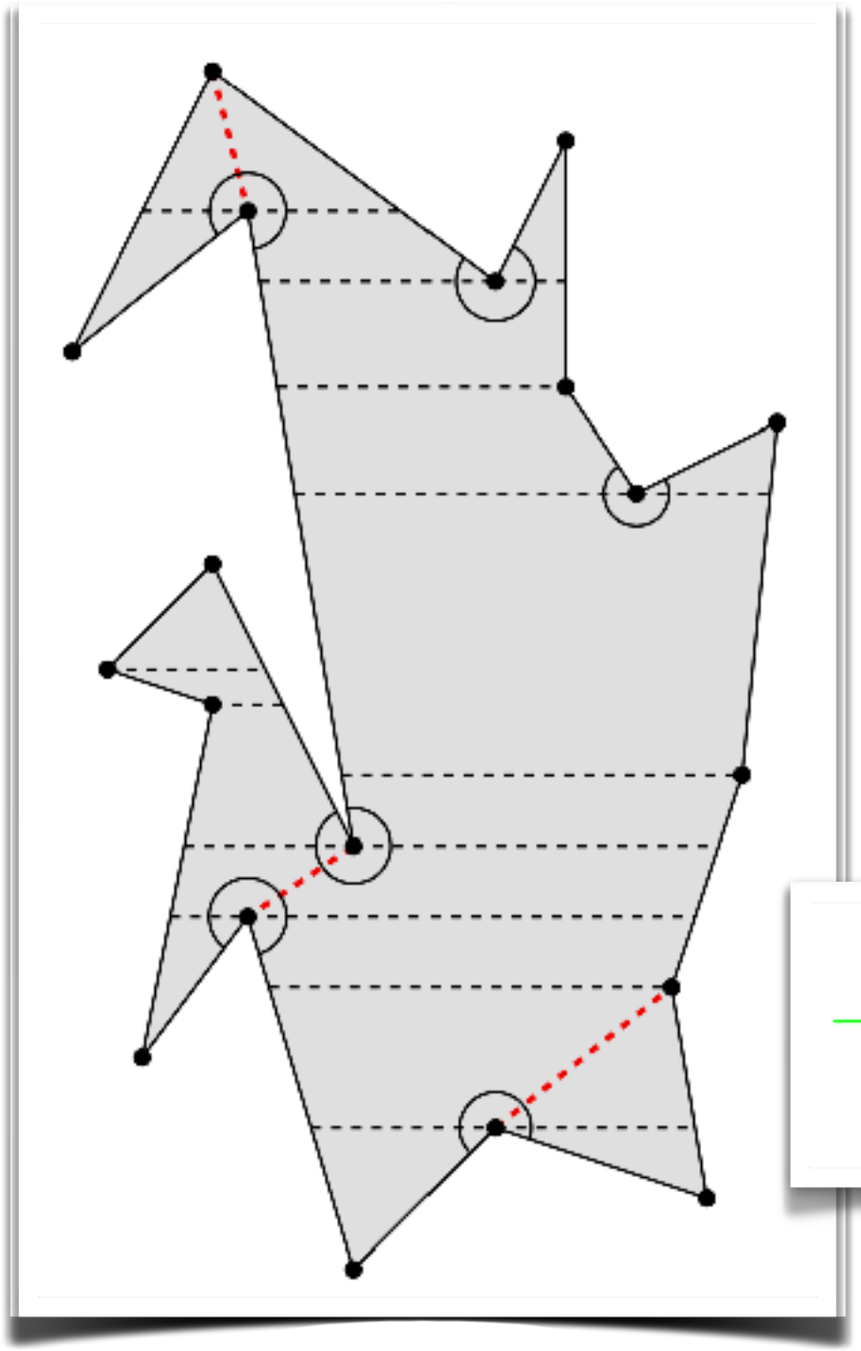
Partitioning into monotone subpolygons

Ideas:

- Vertical plane sweep through vertices
- Maintain horizontal sequence of edges
- Perform updates at events



Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]

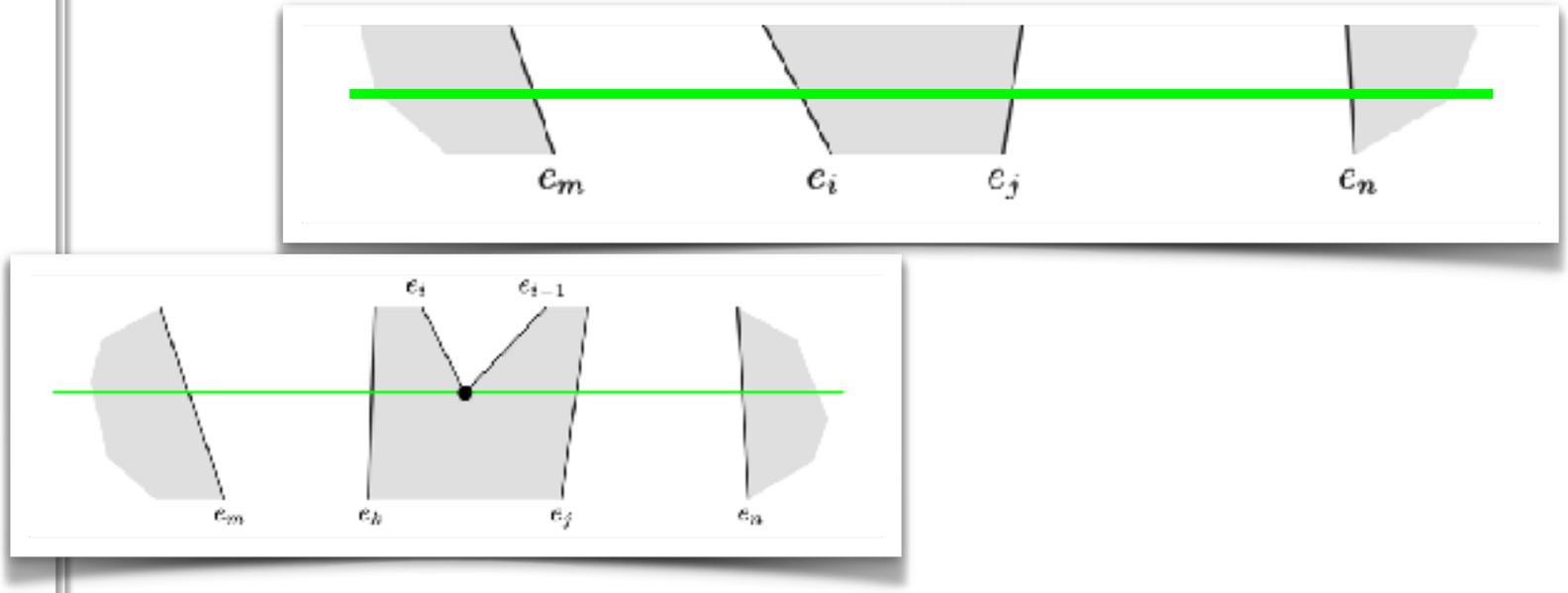


Algorithm 5.18:

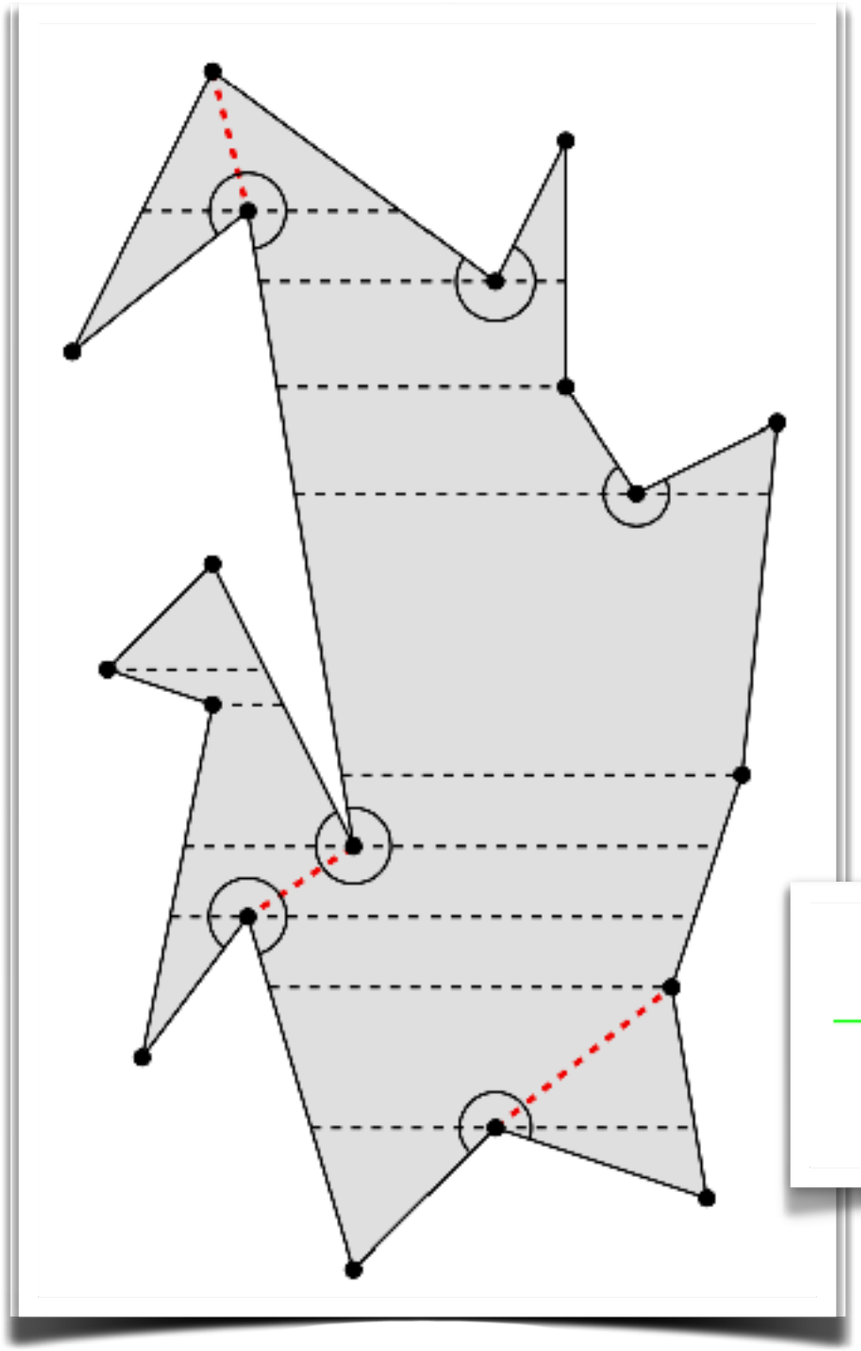
Partitioning into monotone subpolygons

Ideas:

- Vertical plane sweep through vertices
- Maintain horizontal sequence of edges
- Perform updates at events



Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]

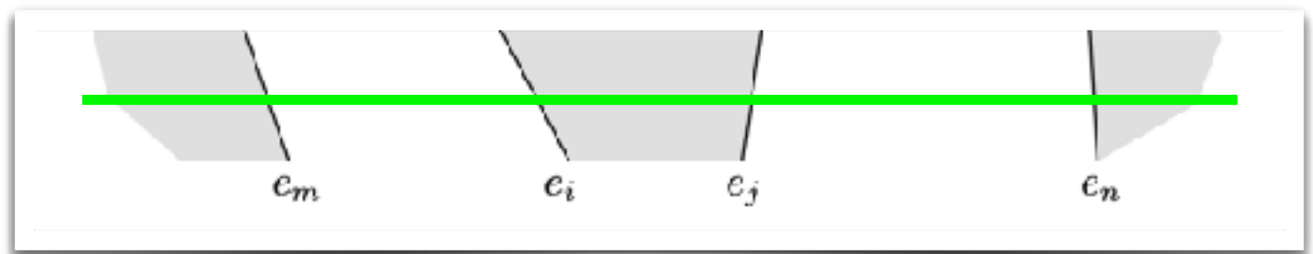


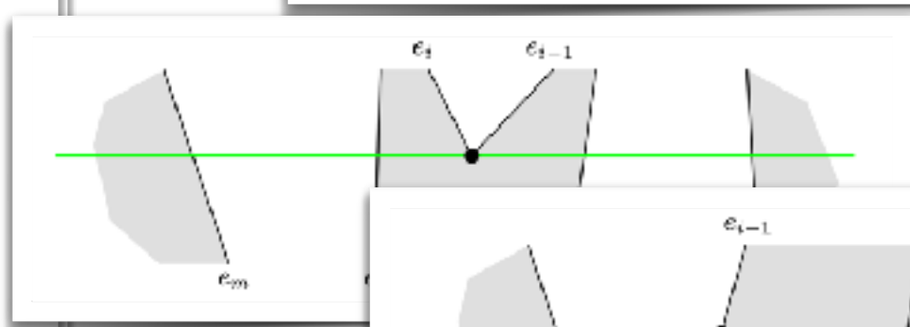
Algorithm 5.18:

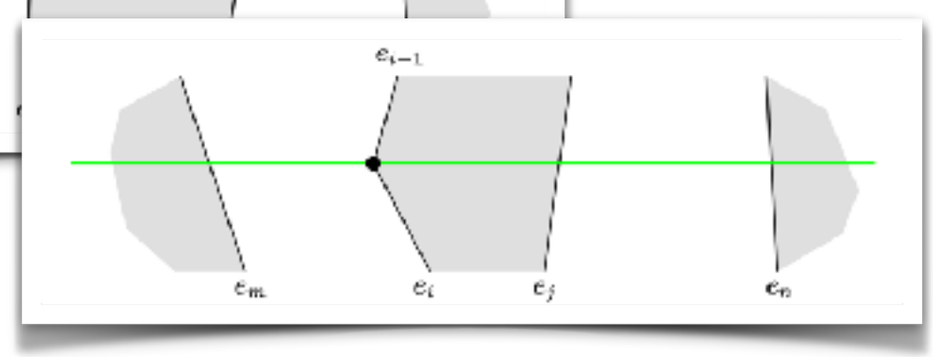
Partitioning into monotone subpolygons

Ideas:

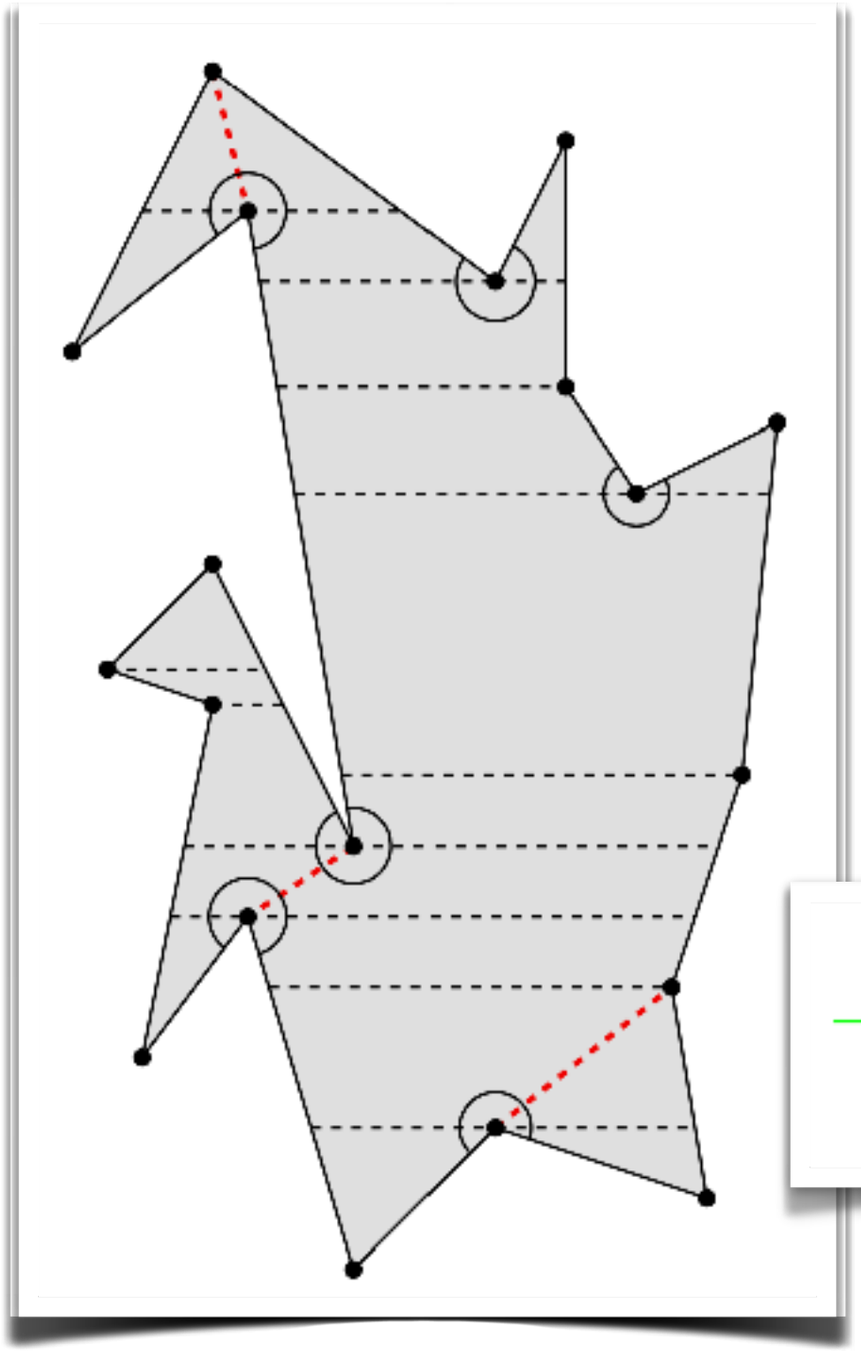
- Vertical plane sweep through vertices
- Maintain horizontal sequence of edges
- Perform updates at events







Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]

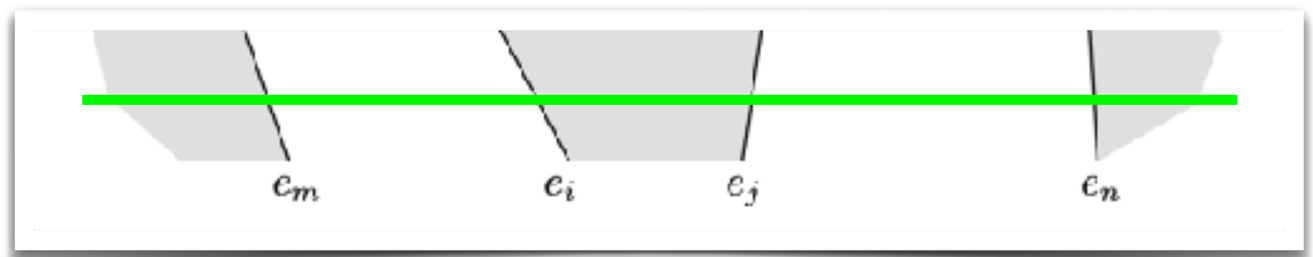


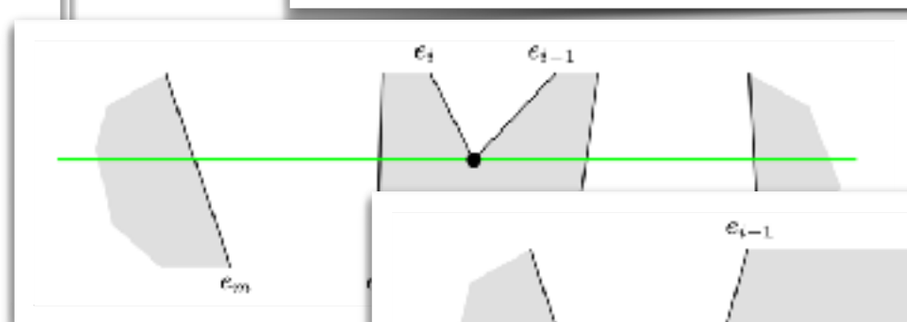
Algorithm 5.18:

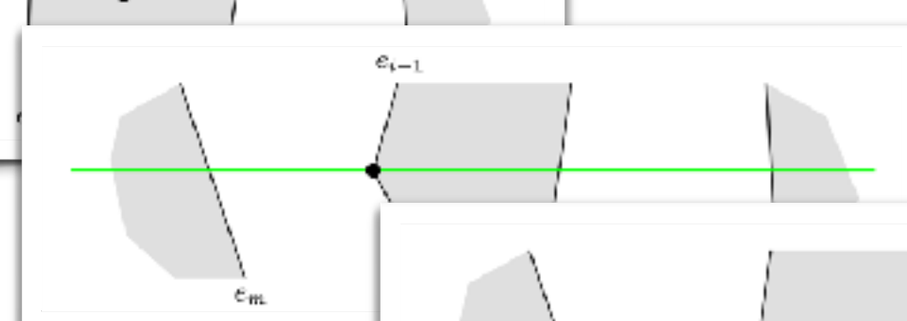
Partitioning into monotone subpolygons

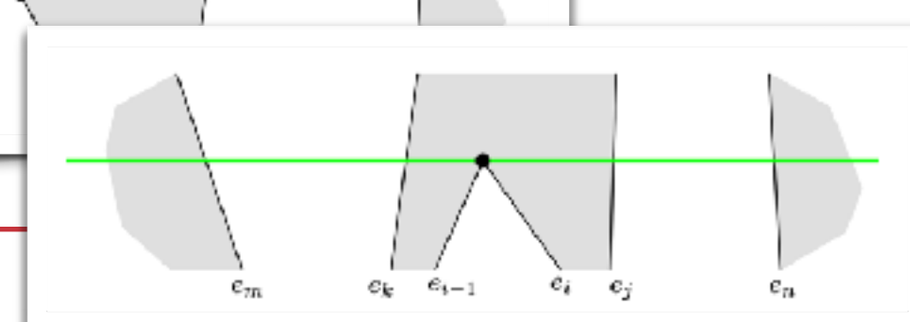
Ideas:

- Vertical plane sweep through vertices
- Maintain horizontal sequence of edges
- Perform updates at events

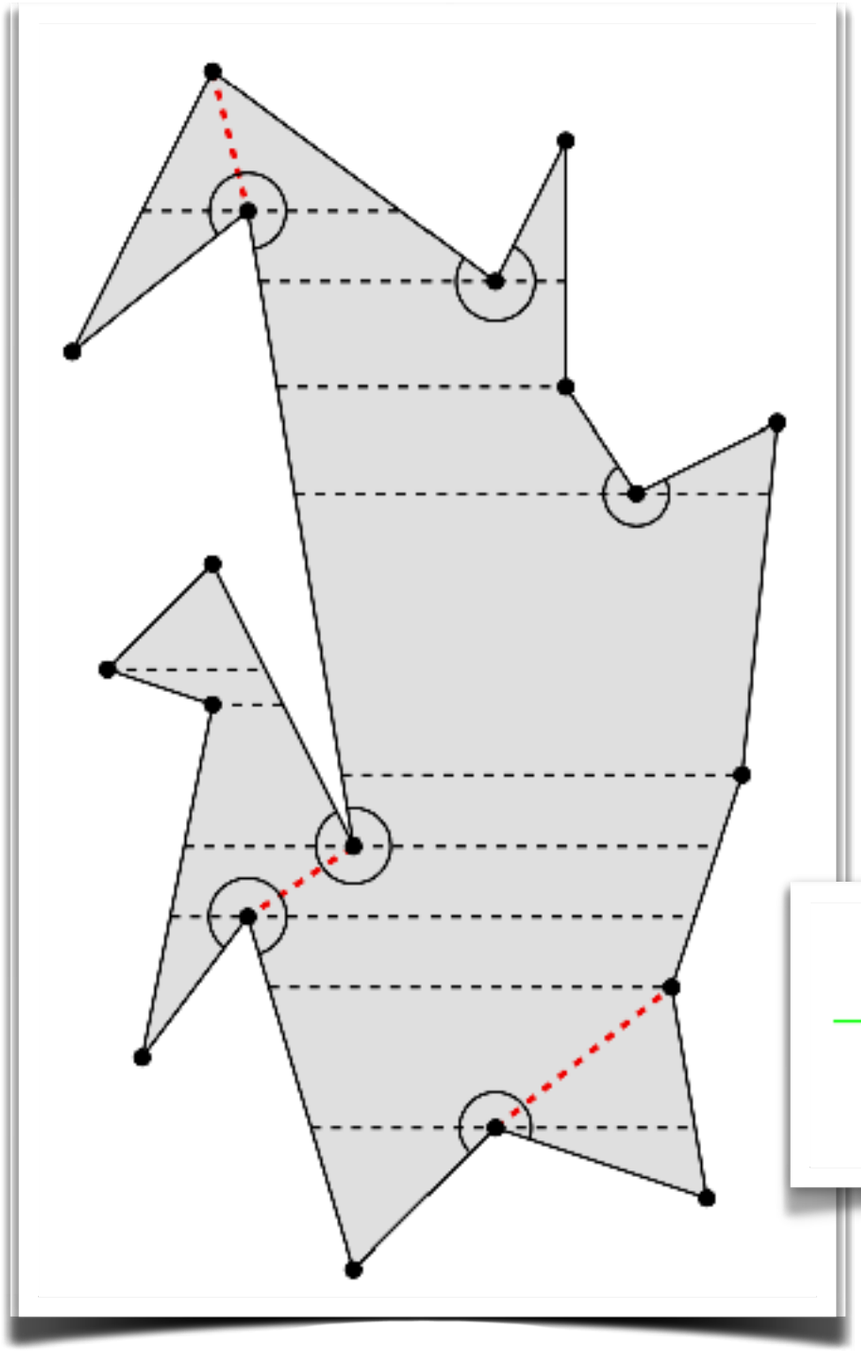








Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]

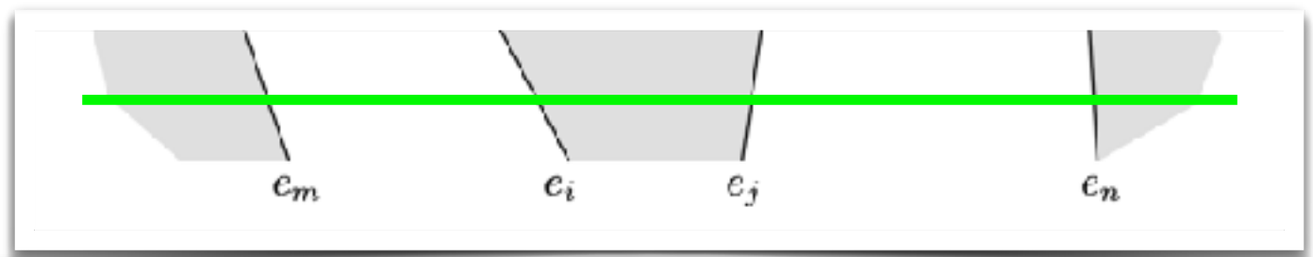


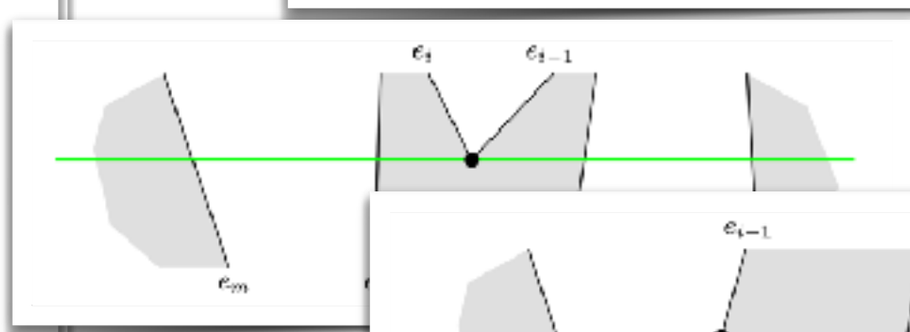
Algorithm 5.18:

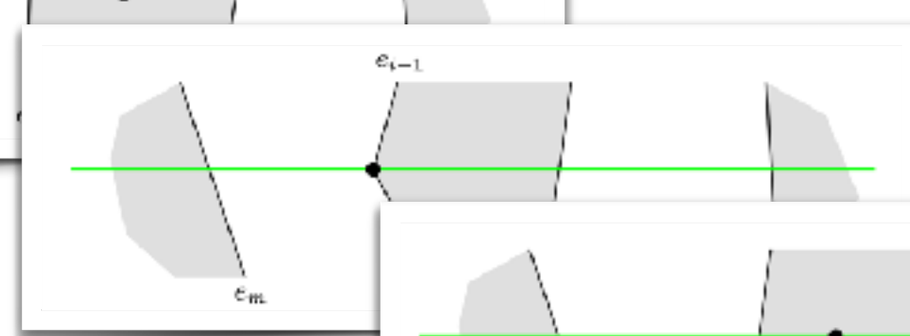
Partitioning into monotone subpolygons

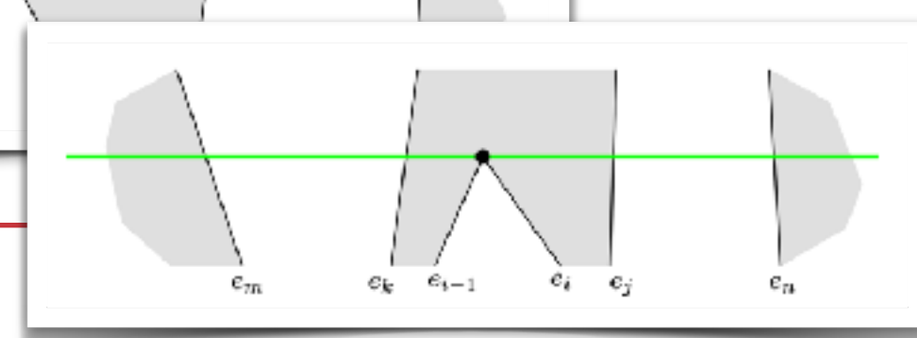
Ideas:

- Vertical plane sweep through vertices
- Maintain horizontal sequence of edges
- Perform updates at events
- Insert or delete in $O(\log n)$

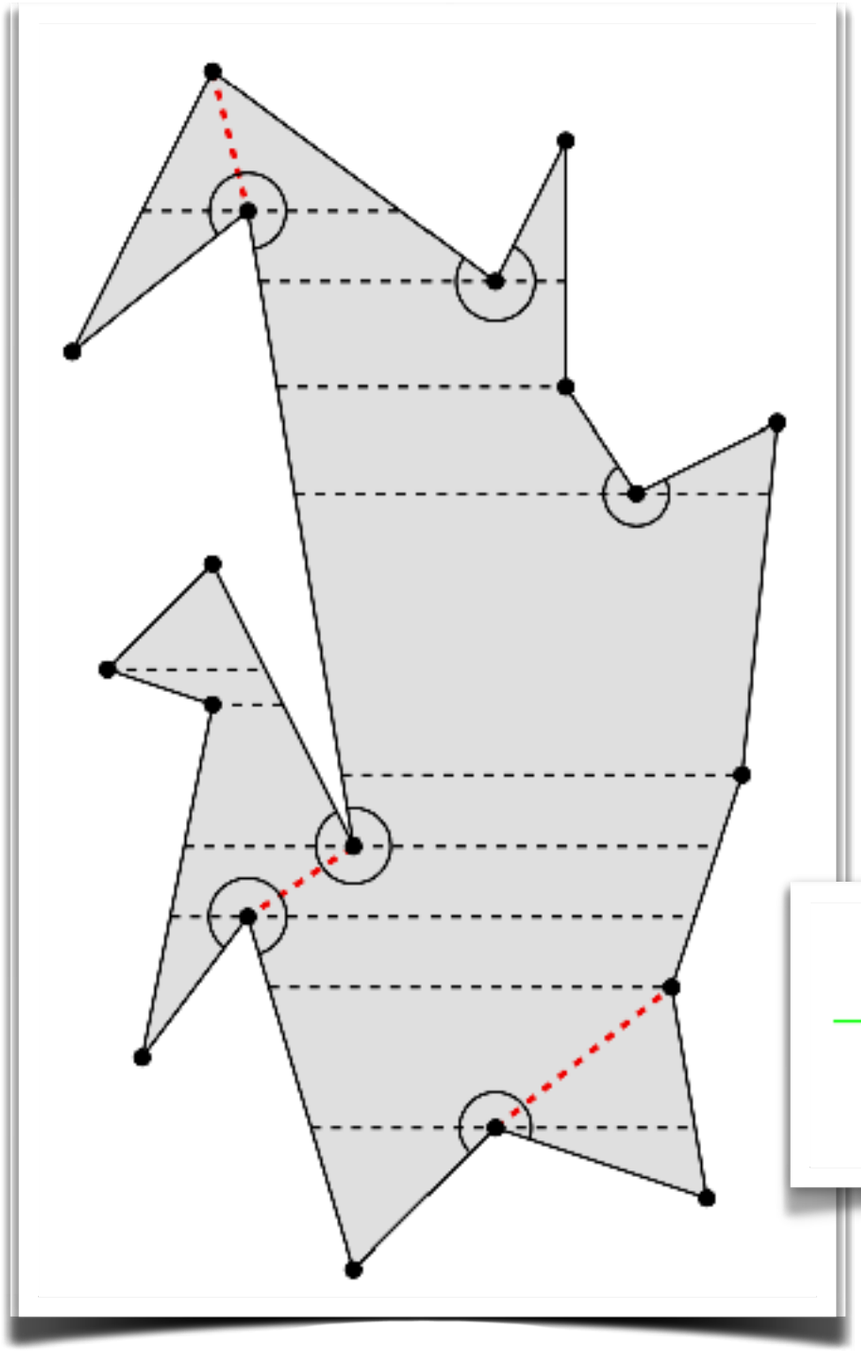








Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]

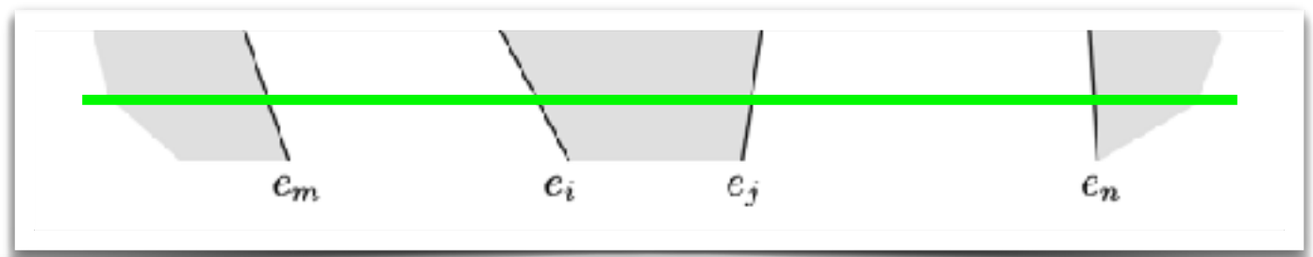


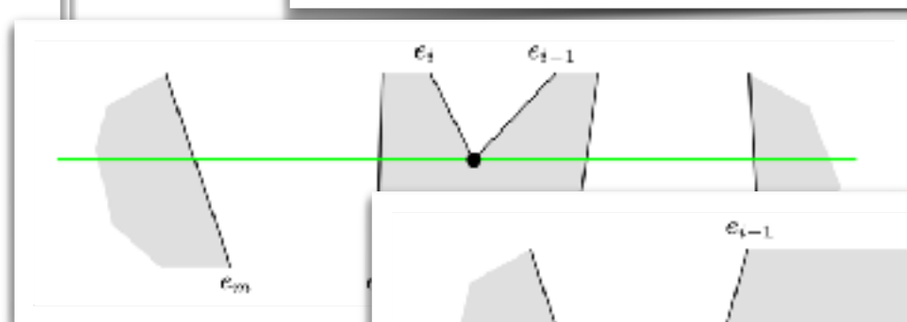
Algorithm 5.18:

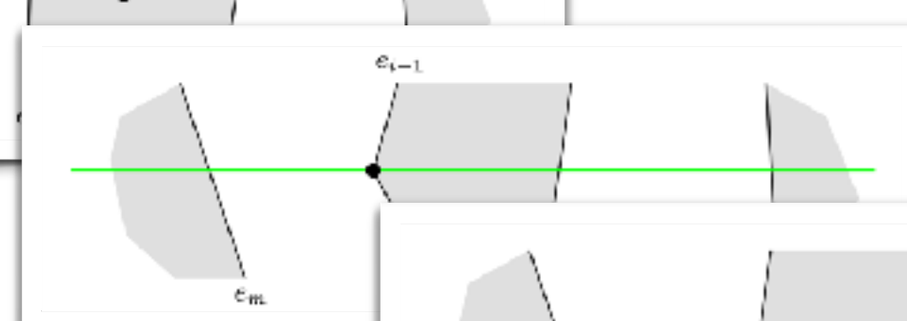
Partitioning into monotone subpolygons

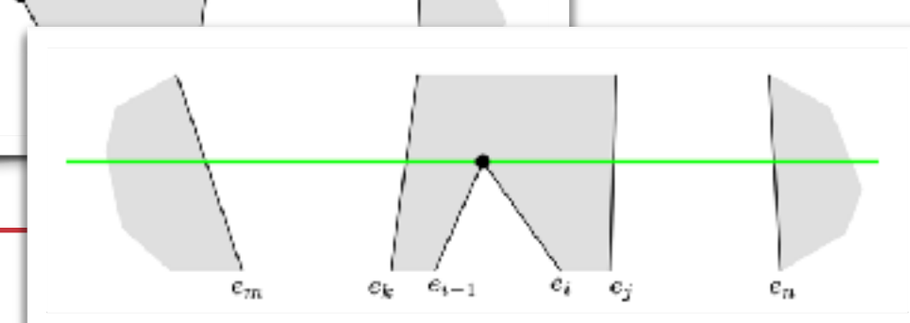
Ideas:

- Vertical plane sweep through vertices
- Maintain horizontal sequence of edges
- Perform updates at events
- Insert or delete in $O(\log n)$
- Connect cusps up or down

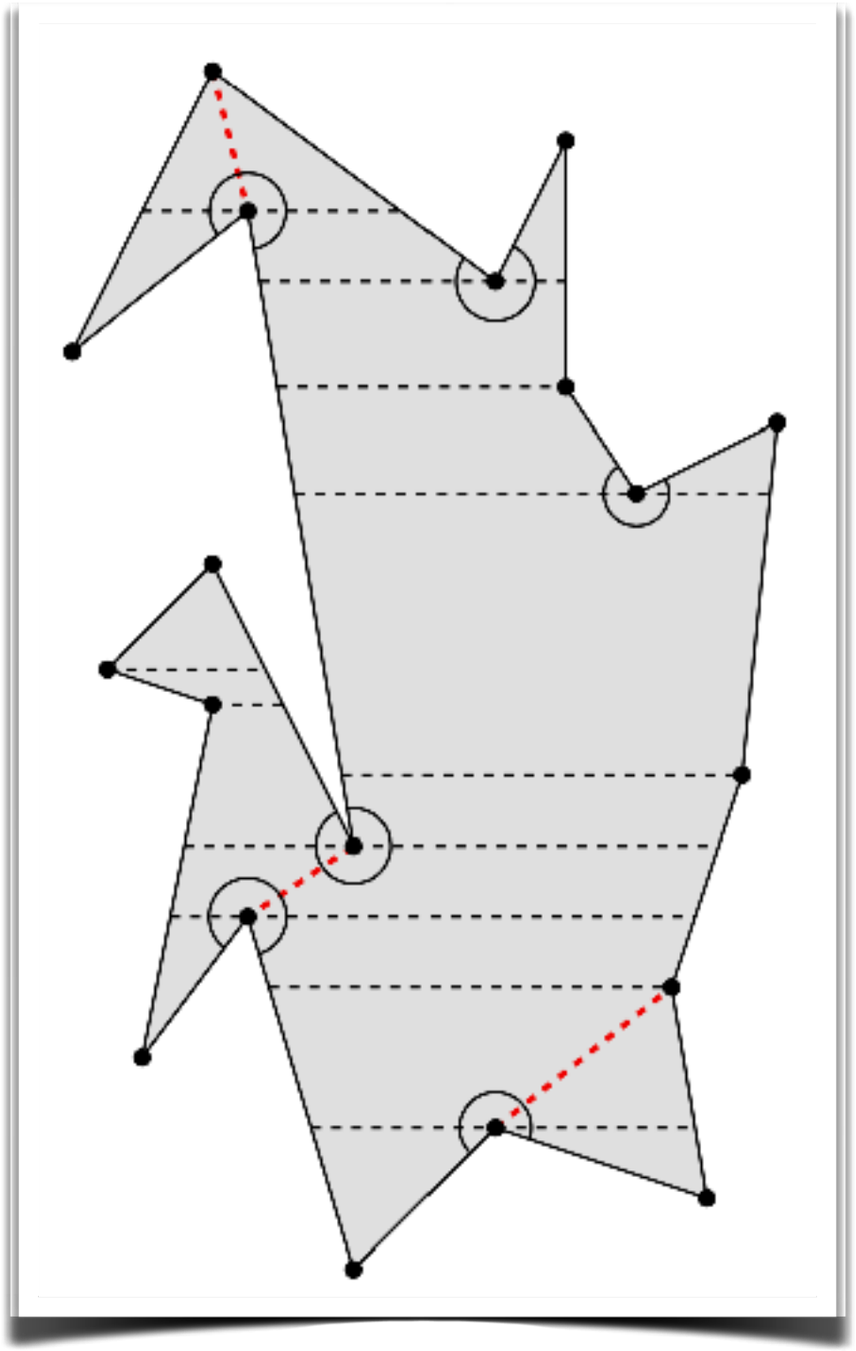








Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]

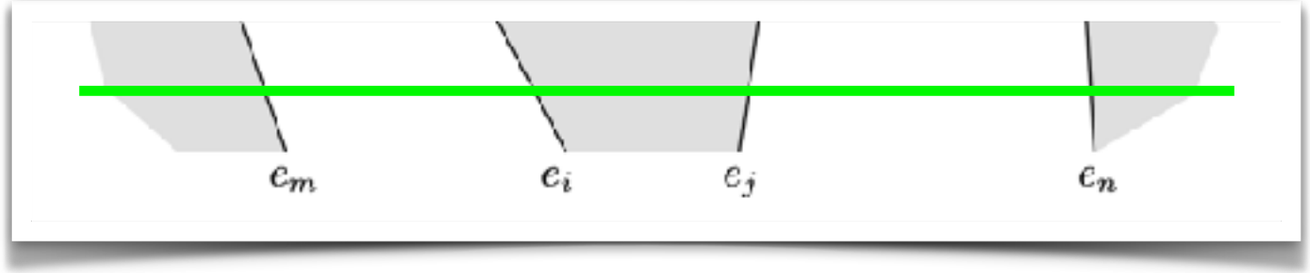


Algorithm 5.18:

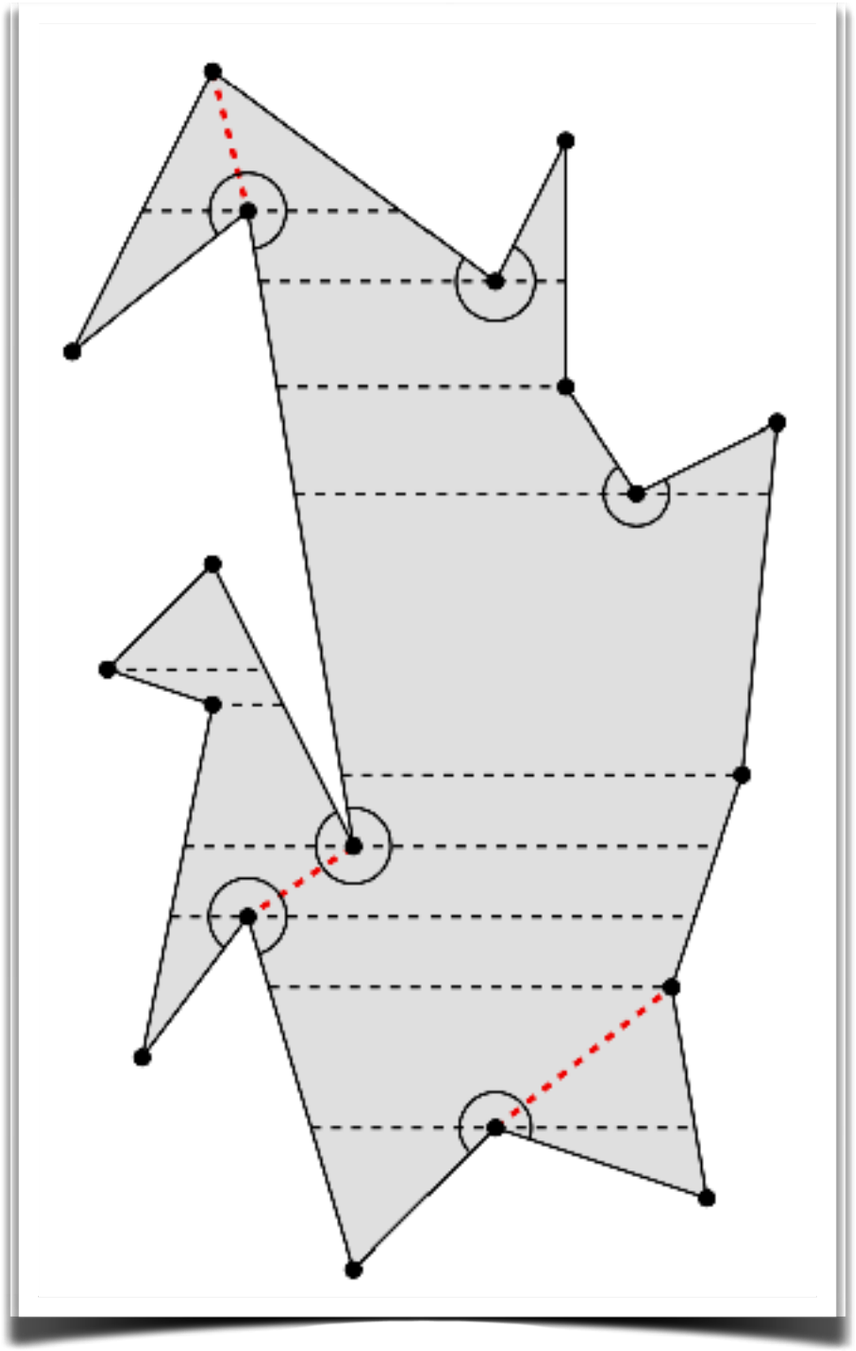
Partitioning into monotone subpolygons

Ideas:

- Vertical plane sweep through vertices
- Maintain horizontal sequence of edges
- Perform updates at events
- Insert or delete in $O(\log n)$
- Connect cusps up or down



Partitioning into Monotone Polygons [Garey, Johnson, Preparata, Tarjan 1978]

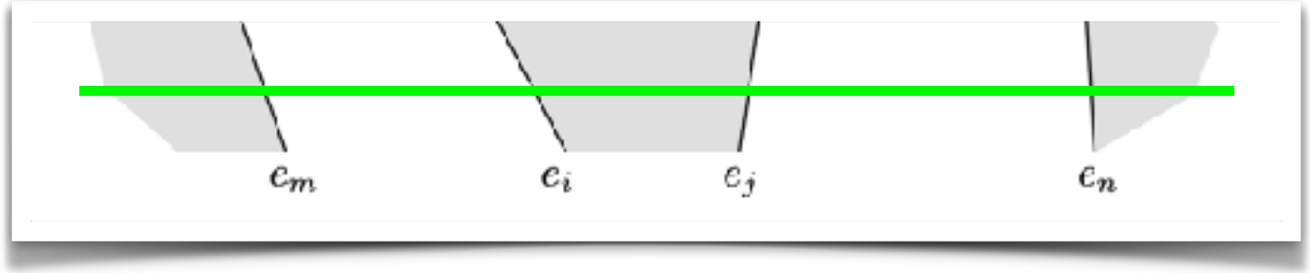


Algorithm 5.18:

Partitioning into monotone subpolygons

Ideas:

- Vertical plane sweep through vertices
- Maintain horizontal sequence of edges
- Perform updates at events
- Insert or delete in $O(\log n)$
- Connect cusps up or down



Theorem 5.19:

Triangulating a simple polygon can be done in $O(n \log n)$.

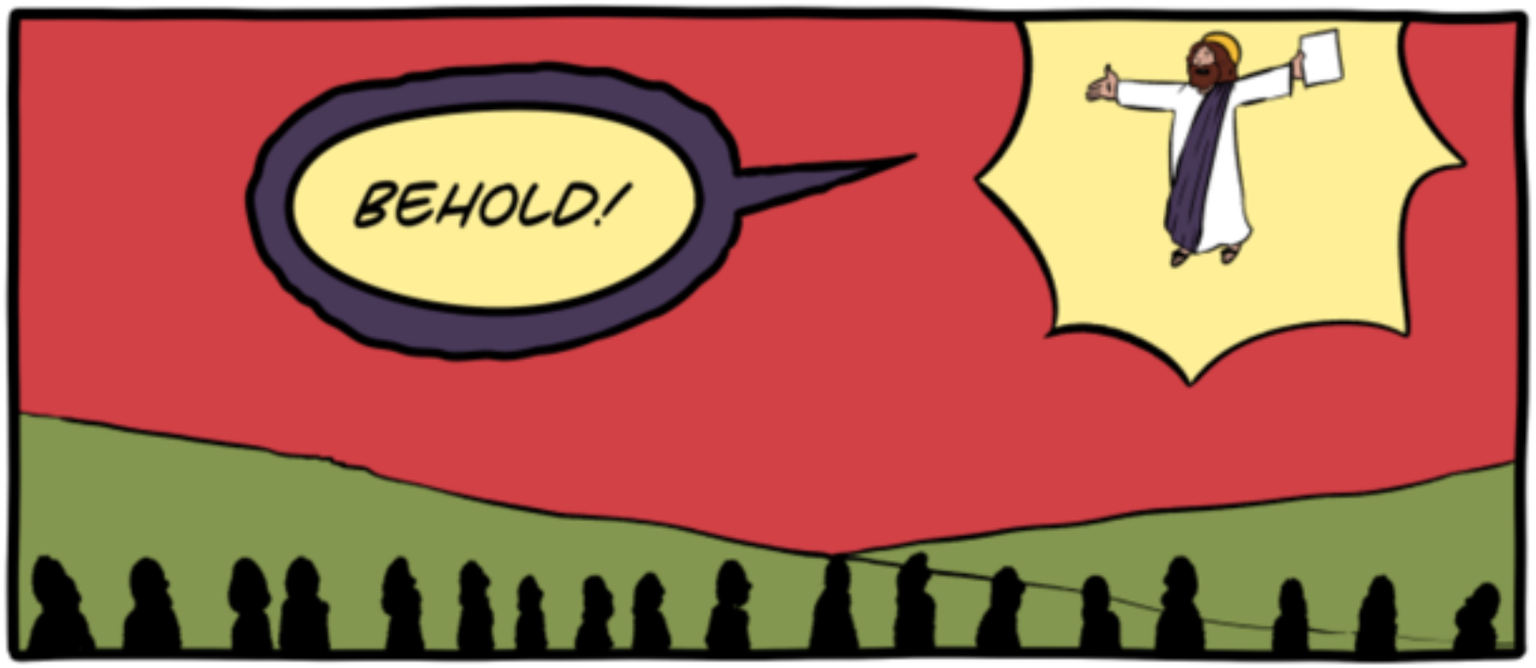
Overview

1. Introduction
2. Existence
3. Properties
4. Algorithms: Removing ears
5. Algorithms: Finding diagonals
6. Algorithms: Monotone polygons
7. Algorithms: Monotone decompositions
8. Faster algorithms
9. Application: Art Gallery problems
10. Application: Online triangulation

Lower bound of $\Omega(n \log n)$ on sorting.

Lower bound of $\Omega(n \log n)$ on sorting.

Lower bound of $\Omega(n \log n)$ on sorting.



Lower bound of $\Omega(n \log n)$ on sorting.

Lower bound of $\Omega(n \log n)$ on sorting.

Lower bound of $\Omega(n \log n)$ on sorting.

Lower bound of $\Omega(n \log n)$ on sorting.

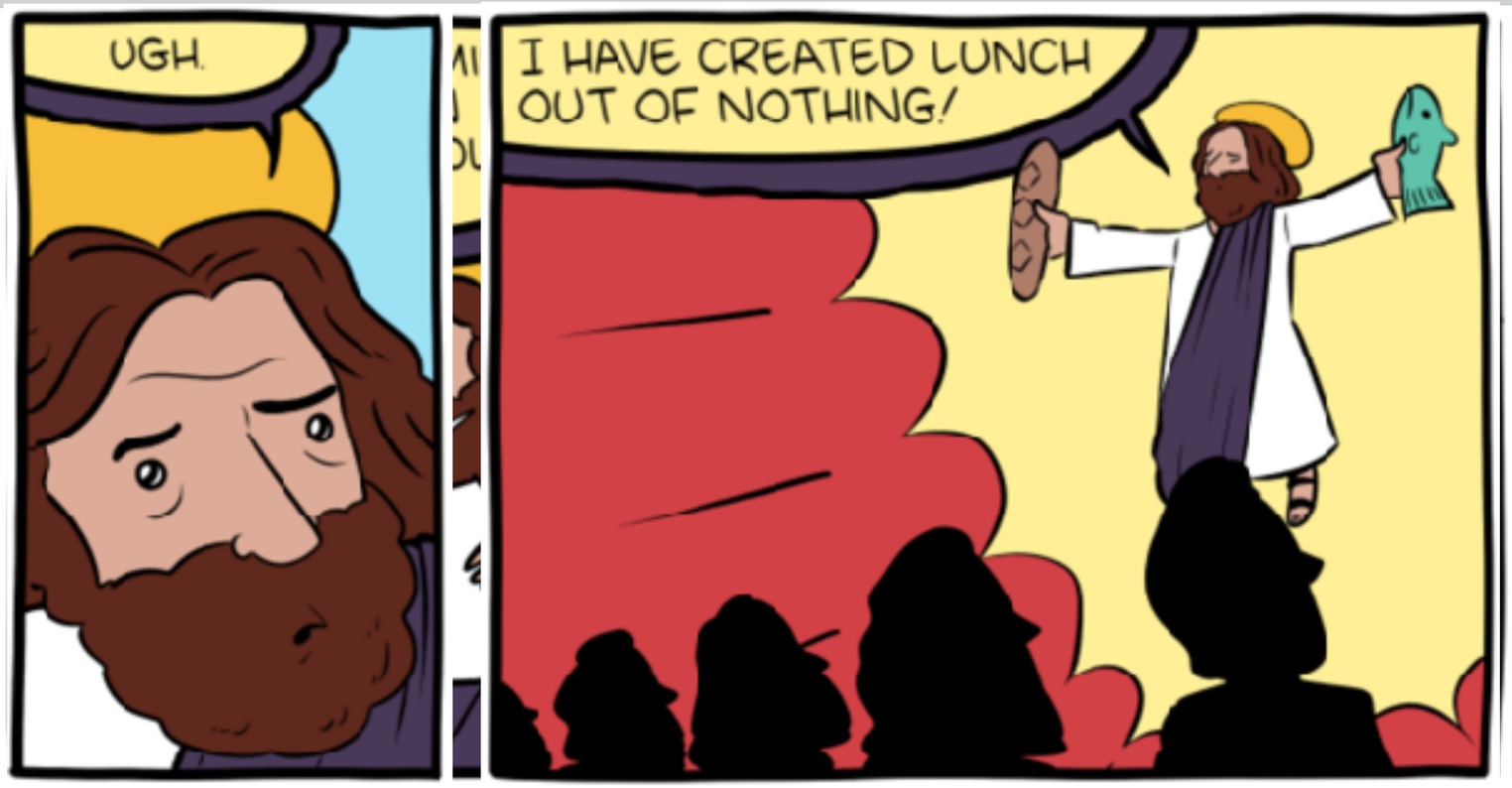
Lower bound of $\Omega(n \log n)$ on sorting.

Lower bound of $\Omega(n \log n)$ on sorting.

Lower bound of $\Omega(n \log n)$ on sorting.

Lower bound of $\Omega(n \log n)$ on sorting.

Lower bound of $\Omega(n \log n)$ on sorting.



Lower bound of $\Omega(n \log n)$ on sorting.

Faster Triangulation

Note:

Note: Lower bound of $\Omega(n \log n)$ only applies to non-simple polygons.

Note: Lower bound of $\Omega(n \log n)$ only applies to non-simple polygons.

Progress:

Note: Lower bound of $\Omega(n \log n)$ only applies to non-simple polygons.

Progress:

Tarjan and van Wyk (1988):

Note: Lower bound of $\Omega(n \log n)$ only applies to non-simple polygons.

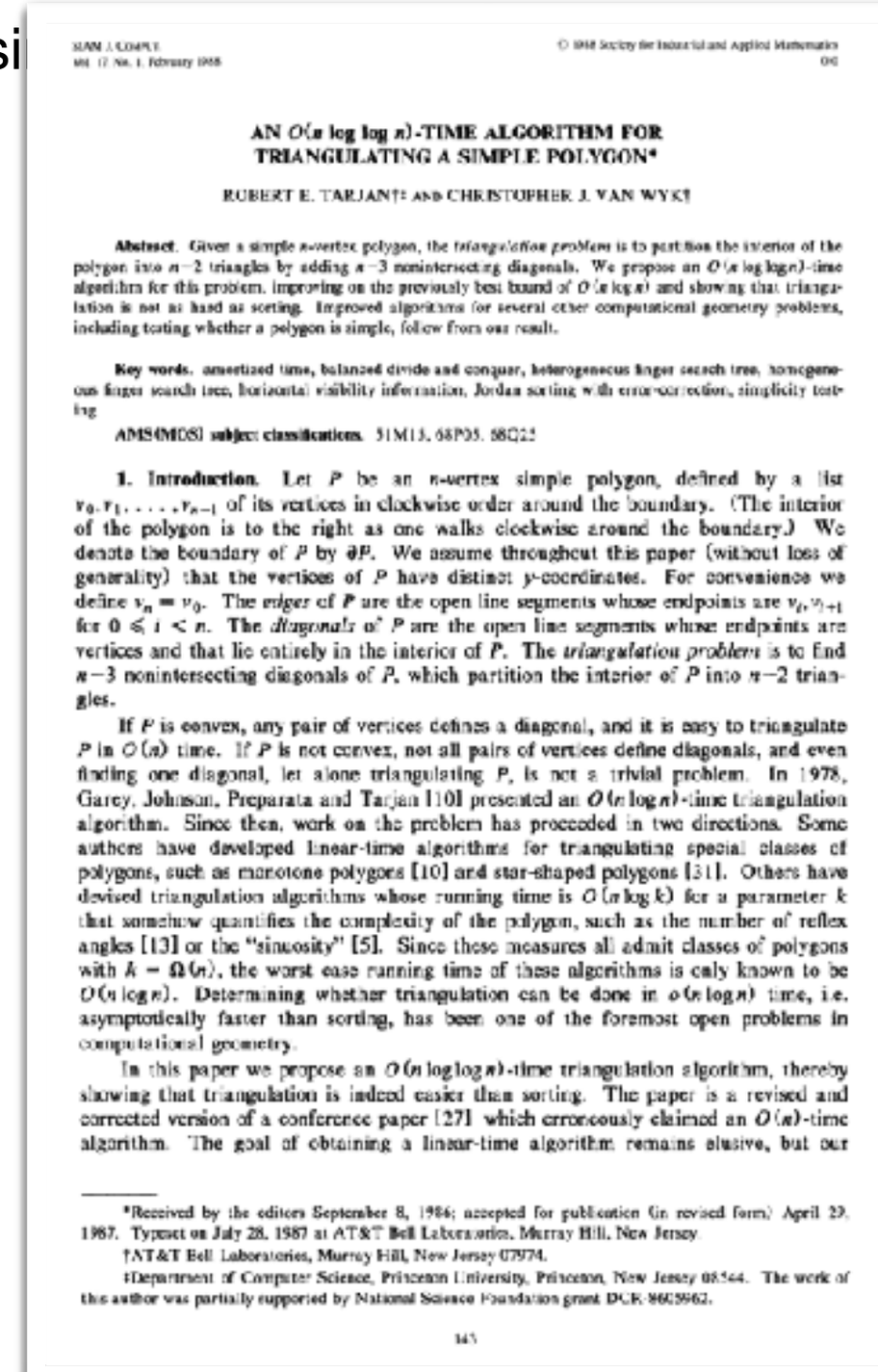
Progress:

Tarjan and van Wyk (1988): $O(n \log \log n)$

Note: Lower bound of $\Omega(n \log n)$ only applies to non-si

Progress:

Tarjan and van Wyk (1988): $O(n \log \log n)$



SIAM J. COMPUT.
Vol. 17, No. 1, February 1988

© 1988 Society for Industrial and Applied Mathematics
006

AN $O(n \log \log n)$-TIME ALGORITHM FOR TRIANGULATING A SIMPLE POLYGON*

ROBERT E. TARJAN† AND CHRISTOPHER J. VAN WYK‡

Abstract. Given a simple n-vertex polygon, the *triangulation problem* is to partition the interior of the polygon into $n-2$ triangles by adding $n-3$ nonintersecting diagonals. We propose an $O(n \log \log n)$-time algorithm for this problem, improving on the previously best bound of $O(n \log n)$ and showing that triangulation is not as hard as sorting. Improved algorithms for several other computational geometry problems, including testing whether a polygon is simple, follow from our result.

Key words. amortized time, balanced divide and conquer, heterogeneous finger search tree, homogeneous finger search tree, horizontal visibility information, Jordan sorting with error-correction, simplicity testing

AMS(MOS) subject classifications. 51M15, 68P05, 68Q25

1. Introduction. Let P be an n-vertex simple polygon, defined by a list $v_0, v_1, \ldots, v_{n-1}$ of its vertices in clockwise order around the boundary. (The interior of the polygon is to the right as one walks clockwise around the boundary.) We denote the boundary of P by ∂P. We assume throughout this paper (without loss of generality) that the vertices of P have distinct y-coordinates. For convenience we define $v_n = v_0$. The *edges* of P are the open line segments whose endpoints are v_i, v_{i+1} for $0 \leq i < n$. The *diagonals* of P are the open line segments whose endpoints are vertices and that lie entirely in the interior of P. The *triangulation problem* is to find $n-3$ nonintersecting diagonals of P, which partition the interior of P into $n-2$ triangles.

If P is convex, any pair of vertices defines a diagonal, and it is easy to triangulate P in $O(n)$ time. If P is not convex, not all pairs of vertices define diagonals, and even finding one diagonal, let alone triangulating P, is not a trivial problem. In 1978, Garey, Johnson, Preparata and Tarjan [10] presented an $O(n \log n)$-time triangulation algorithm. Since then, work on the problem has proceeded in two directions. Some authors have developed linear-time algorithms for triangulating special classes of polygons, such as monotone polygons [10] and star-shaped polygons [31]. Others have devised triangulation algorithms whose running time is $O(n \log k)$ for a parameter k that somehow quantifies the complexity of the polygon, such as the number of reflex angles [13] or the "sinuosity" [5]. Since these measures all admit classes of polygons with $k = \Omega(n)$, the worst case running time of these algorithms is only known to be $O(n \log n)$. Determining whether triangulation can be done in $o(n \log n)$ time, i.e. asymptotically faster than sorting, has been one of the foremost open problems in computational geometry.

In this paper we propose an $O(n \log \log n)$-time triangulation algorithm, thereby showing that triangulation is indeed easier than sorting. The paper is a revised and corrected version of a conference paper [27] which erroneously claimed an $O(n)$-time algorithm. The goal of obtaining a linear-time algorithm remains elusive, but our

*Received by the editors September 8, 1986; accepted for publication (in revised form) April 29, 1987. Typeset on July 28, 1987 at AT&T Bell Laboratories, Murray Hill, New Jersey.

†AT&T Bell Laboratories, Murray Hill, New Jersey 07974.

‡Department of Computer Science, Princeton University, Princeton, New Jersey 08544. The work of this author was partially supported by National Science Foundation grant DCR-8605962.

Note: Lower bound of $\Omega(n \log n)$ only applies to non-simple polygons.

Progress:

Tarjan and van Wyk (1988): $O(n \log \log n)$

Note: Lower bound of $\Omega(n \log n)$ only applies to non-simple polygons.

Progress:

Tarjan and van Wyk (1988): $O(n \log \log n)$

Kirkpatrick, Klawe, Tarjan (1992):

Note: Lower bound of $\Omega(n \log n)$ only applies to non-simple polygons.

Progress:

Tarjan and van Wyk (1988): $O(n \log \log n)$

Kirkpatrick, Klawe, Tarjan (1992): $O(n \log \log n)$

Note: Lower bound of $\Omega(n \log n)$ only applies to non-si

Progress:

Tarjan and van Wyk (1988): $O(n \log \log n)$

Kirkpatrick, Klawe, Tarjan (1992): $O(n \log \log n)$

Discrete Comput Geom 7: 329–346 (1992)

Discrete & Computational Geometry

© 1992 Springer-Verlag New York Inc

Polygon Triangulation in $O(n \log \log n)$ Time with Simple Data Structures*

David G. Kirkpatrick,[1] Maria M. Klawe,[2] and Robert E. Tarjan[2]

[1] Department of Computer Science, University of British Columbia, Vancouver, British Columbia, Canada V6T 1Z2

[2] Department of Computer Science, Princeton University, Princeton, NJ 08540, USA, and NEC Research Institute

Abstract. We give a new $O(n \log \log n)$-time deterministic algorithm for triangulating simple n-vertex polygons, which avoids the use of complicated data structures. In addition, for polygons whose vertices have integer coordinates of polynomially bounded size, the algorithm can be modifed to run in $O(n \log^* n)$ time. The major new techniques employed are the efficient location of horizontal visibility edges that partition the interior of the polygon into regions of approximately equal size, and a linear-time algorithm for obtaining the horizontal visibility partition of a subchain of a polygonal chain, from the horizontal visibility partition of the entire chain. The latter technique has other interesting applications, including a linear-time algorithm to convert a Steiner triangulation of a polygon into a true triangulation.

1. Introduction

Let P be a simple polygon with n vertices. The *diagonals* of P are the open line segments whose endpoints are vertices of the polygon and that lie entirely in the interior of P. A *triangulation* of P is a partition of its interior into $n-2$ triangles by adding $n-3$ nonintersecting diagonals. The problem of triangulating a simple polygon, that is, determining the set of nonintersecting diagonals, has attracted considerable attention in computational geometry literature and elsewhere

* This research was partially supported by the following grants: NSERC 583584, NSERC 580485, ONR-N00014-87-0467, and by DIMACS, an NSF Science and Technology Center (NSF-STC88-09648).

Note: Lower bound of $\Omega(n \log n)$ only applies to non-simple polygons.

Progress:

Tarjan and van Wyk (1988): $O(n \log \log n)$

Kirkpatrick, Klawe, Tarjan (1992): $O(n \log \log n)$

Note: Lower bound of $\Omega(n \log n)$ only applies to non-simple polygons.

Progress:

Tarjan and van Wyk (1988): $O(n \log \log n)$

Kirkpatrick, Klawe, Tarjan (1992): $O(n \log \log n)$

Clarkson, Tarjan, van Wyk (1989):

Note: Lower bound of $\Omega(n \log n)$ only applies to non-simple polygons.

Progress:

Tarjan and van Wyk (1988): $O(n \log \log n)$

Kirkpatrick, Klawe, Tarjan (1992): $O(n \log \log n)$

Clarkson, Tarjan, van Wyk (1989): $O(n \log^* n)$

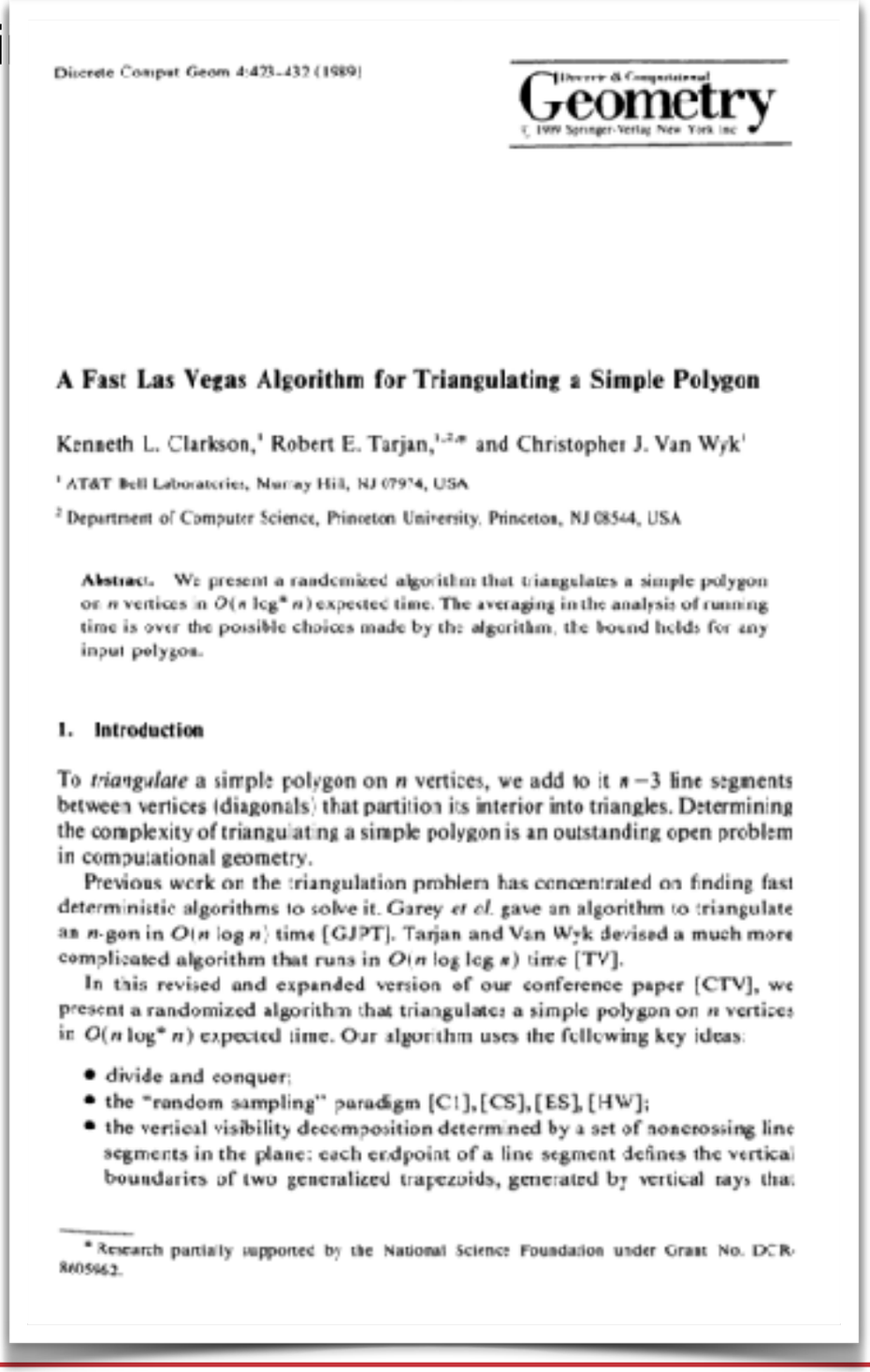
Note: Lower bound of $\Omega(n \log n)$ only applies to non-si

Progress:

Tarjan and van Wyk (1988): $O(n \log \log n)$

Kirkpatrick, Klawe, Tarjan (1992): $O(n \log \log n)$

Clarkson, Tarjan, van Wyk (1989): $O(n \log^* n)$



Discrete Comput Geom 4:423–432 (1989)

Discrete & Computational Geometry
© 1989 Springer-Verlag New York Inc

A Fast Las Vegas Algorithm for Triangulating a Simple Polygon

Kenneth L. Clarkson,[1] Robert E. Tarjan,[1,2,*] and Christopher J. Van Wyk[1]

[1] AT&T Bell Laboratories, Murray Hill, NJ 07974, USA

[2] Department of Computer Science, Princeton University, Princeton, NJ 08544, USA

Abstract. We present a randomized algorithm that triangulates a simple polygon on n vertices in $O(n \log^* n)$ expected time. The averaging in the analysis of running time is over the possible choices made by the algorithm; the bound holds for any input polygon.

1. Introduction

To *triangulate* a simple polygon on n vertices, we add to it $n-3$ line segments between vertices (diagonals) that partition its interior into triangles. Determining the complexity of triangulating a simple polygon is an outstanding open problem in computational geometry.

Previous work on the triangulation problem has concentrated on finding fast deterministic algorithms to solve it. Garey *et al.* gave an algorithm to triangulate an n-gon in $O(n \log n)$ time [GJPT]. Tarjan and Van Wyk devised a much more complicated algorithm that runs in $O(n \log \log n)$ time [TV].

In this revised and expanded version of our conference paper [CTV], we present a randomized algorithm that triangulates a simple polygon on n vertices in $O(n \log^* n)$ expected time. Our algorithm uses the following key ideas:

- divide and conquer;
- the "random sampling" paradigm [C1], [CS], [ES], [HW];
- the vertical visibility decomposition determined by a set of nonerossing line segments in the plane: each endpoint of a line segment defines the vertical boundaries of two generalized trapezoids, generated by vertical rays that

* Research partially supported by the National Science Foundation under Grant No. DCR-8605662.

Note: Lower bound of $\Omega(n \log n)$ only applies to non-simple polygons.

Progress:

Tarjan and van Wyk (1988): $O(n \log \log n)$

Kirkpatrick, Klawe, Tarjan (1992): $O(n \log \log n)$

Clarkson, Tarjan, van Wyk (1989): $O(n \log^* n)$

Note: Lower bound of $\Omega(n \log n)$ only applies to non-simple polygons.

Progress:

Tarjan and van Wyk (1988): $O(n \log \log n)$

Kirkpatrick, Klawe, Tarjan (1992): $O(n \log \log n)$

Clarkson, Tarjan, van Wyk (1989): $O(n \log^* n)$

Chazelle (1991):

Note: Lower bound of $\Omega(n \log n)$ only applies to non-simple polygons.

Progress:

Tarjan and van Wyk (1988): $O(n \log \log n)$

Kirkpatrick, Klawe, Tarjan (1992): $O(n \log \log n)$

Clarkson, Tarjan, van Wyk (1989): $O(n \log^* n)$

Chazelle (1991): $O(n)$

Note: Lower bound of $\Omega(n \log n)$ only applies to non-si

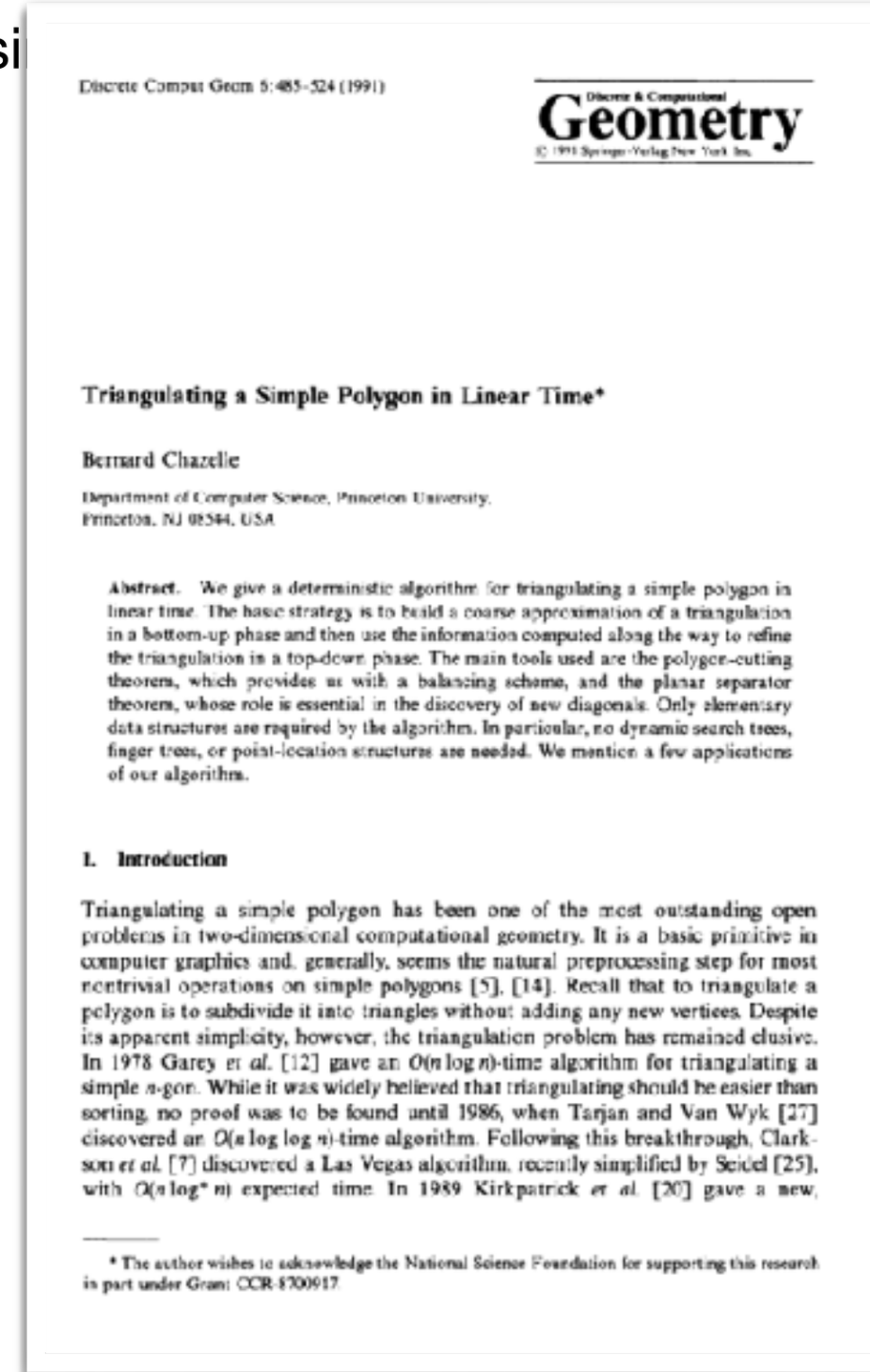
Progress:

Tarjan and van Wyk (1988): $O(n \log \log n)$

Kirkpatrick, Klawe, Tarjan (1992): $O(n \log \log n)$

Clarkson, Tarjan, van Wyk (1989): $O(n \log^* n)$

Chazelle (1991): $O(n)$



Discrete Comput Geom 5:485–524 (1991)

Discrete & Computational Geometry

© 1991 Springer-Verlag New York Inc.

Triangulating a Simple Polygon in Linear Time*

Bernard Chazelle

Department of Computer Science, Princeton University,
Princeton, NJ 08544, USA

Abstract. We give a deterministic algorithm for triangulating a simple polygon in linear time. The basic strategy is to build a coarse approximation of a triangulation in a bottom-up phase and then use the information computed along the way to refine the triangulation in a top-down phase. The main tools used are the polygon-cutting theorem, which provides us with a balancing scheme, and the planar separator theorem, whose role is essential in the discovery of new diagonals. Only elementary data structures are required by the algorithm. In particular, no dynamic search trees, finger trees, or point-location structures are needed. We mention a few applications of our algorithm.

1. Introduction

Triangulating a simple polygon has been one of the most outstanding open problems in two-dimensional computational geometry. It is a basic primitive in computer graphics and, generally, seems the natural preprocessing step for most nontrivial operations on simple polygons [5], [14]. Recall that to triangulate a polygon is to subdivide it into triangles without adding any new vertices. Despite its apparent simplicity, however, the triangulation problem has remained elusive. In 1978 Garey *et al.* [12] gave an $O(n \log n)$-time algorithm for triangulating a simple n-gon. While it was widely believed that triangulating should be easier than sorting, no proof was to be found until 1986, when Tarjan and Van Wyk [27] discovered an $O(n \log \log n)$-time algorithm. Following this breakthrough, Clarkson *et al.* [7] discovered a Las Vegas algorithm, recently simplified by Seidel [25], with $O(n \log^* n)$ expected time. In 1989 Kirkpatrick *et al.* [20] gave a new,

* The author wishes to acknowledge the National Science Foundation for supporting this research in part under Grant CCR-8700917.

Note: Lower bound of $\Omega(n \log n)$ only applies to non-simple polygons.

Progress:

Tarjan and van Wyk (1988): $O(n \log \log n)$

Kirkpatrick, Klawe, Tarjan (1992): $O(n \log \log n)$

Clarkson, Tarjan, van Wyk (1989): $O(n \log^* n)$

Chazelle (1991): $O(n)$

Overview

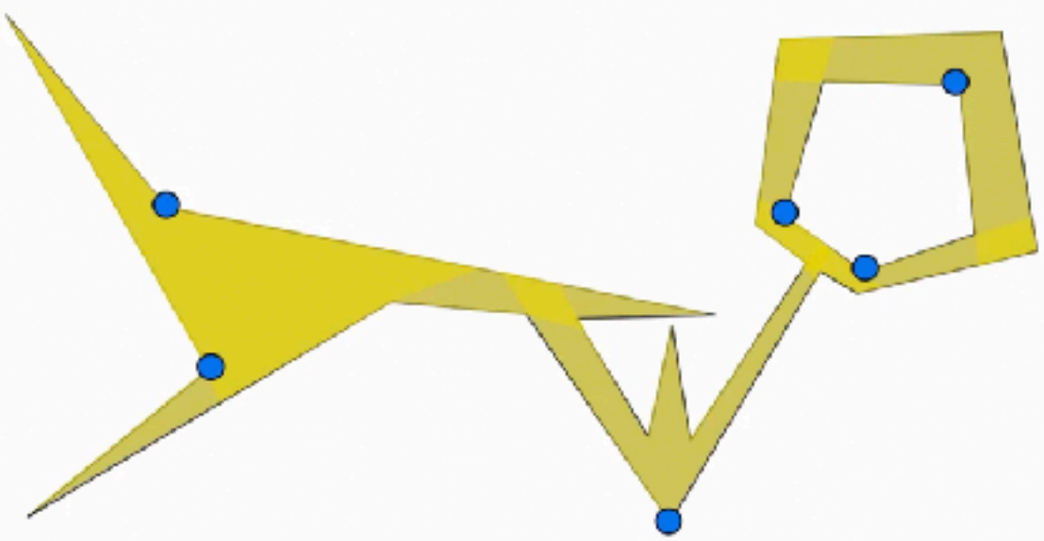
1. Introduction
2. Existence
3. Properties
4. Algorithms: Removing ears
5. Algorithms: Finding diagonals
6. Algorithms: Monotone polygons
7. Algorithms: Monotone decompositions
8. Faster algorithms
9. Application: Art Gallery problems
10. Application: Online triangulation

THE ART GALLERY PROBLEM

Given:
a polygon *P*

Find:
a minimum number of lights

Such that:
P is fully illuminated



JOURNAL OF COMBINATORIAL THEORY, Series B 24, 374 (1978)

Note

A Short Proof of Chvátal's Watchman Theorem

STEVE FISK

Department of Mathematics, Bowdoin College, Brunswick, Maine 04011

Communicated by the Editors

Received October 27, 1977

This note contains a short proof of Chvátal's Watchman Theorem using the existence of a three-coloring of a triangulated polygon.

In 1975 Chvátal [1], proved the following result:

THEOREM. *If S is a polygon with n vertices, then there is a set T of at most $n/3$ points of S such that for any point p of S there is a point q of T with the segment pq lying entirely in S.*

If we think of S as a museum, with paintings on the walls, then the theorem gives a bound on the number of stationary watchmen required to guard every part of the museum. We present a simple proof.

Proof. Triangulate S so that no new vertices are added. Every such triangulation has a coloring with three colors a, b, c. Let T_k be the set of vertices colored k, and assume that $|T_a| \leqslant |T_b| \leqslant |T_c|$. Choosing $T = T_a$ implies $|T| \leqslant n/3$. Finally, every point q of S lies in some triangle of S, and every triangle of S has a point p of T on it. Since triangles are convex, we have $pq \subset S$.

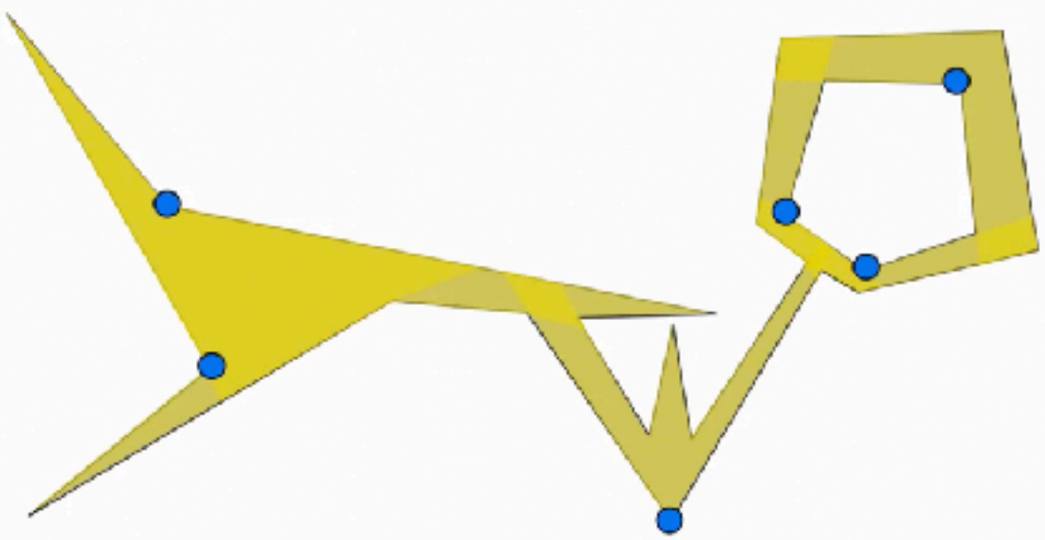
REFERENCE

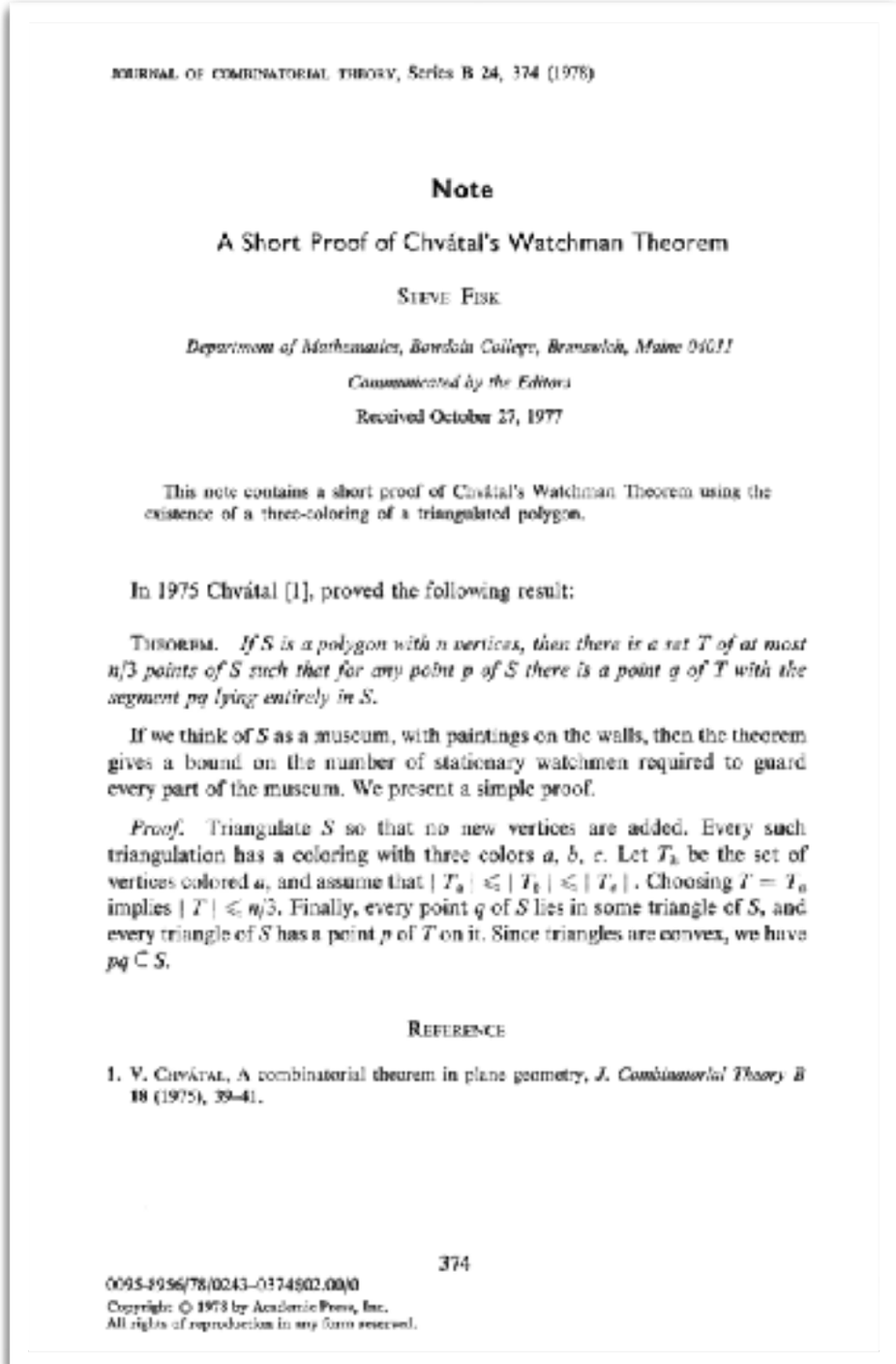
1. V. CHVÁTAL, A combinatorial theorem in plane geometry, *J. Combinatorial Theory B* 18 (1975), 39–41.

0095-8956/78/0243-0374$02.00/0

Copyright © 1978 by Academic Press, Inc.
All rights of reproduction in any form reserved.

ALLERY PROBLEM





JOURNAL OF COMBINATORIAL THEORY, Series B 24, 374 (1978)

Note

A Short Proof of Chvátal's Watchman Theorem

STEVE FISK

Department of Mathematics, Bowdoin College, Brunswick, Maine 04011

Communicated by the Editors

Received October 27, 1977

This note contains a short proof of Chvátal's Watchman Theorem using the existence of a three-coloring of a triangulated polygon.

In 1975 Chvátal [1], proved the following result:

THEOREM. *If S is a polygon with n vertices, then there is a set T of at most $n/3$ points of S such that for any point p of S there is a point q of T with the segment pq lying entirely in S.*

If we think of S as a museum, with paintings on the walls, then the theorem gives a bound on the number of stationary watchmen required to guard every part of the museum. We present a simple proof.

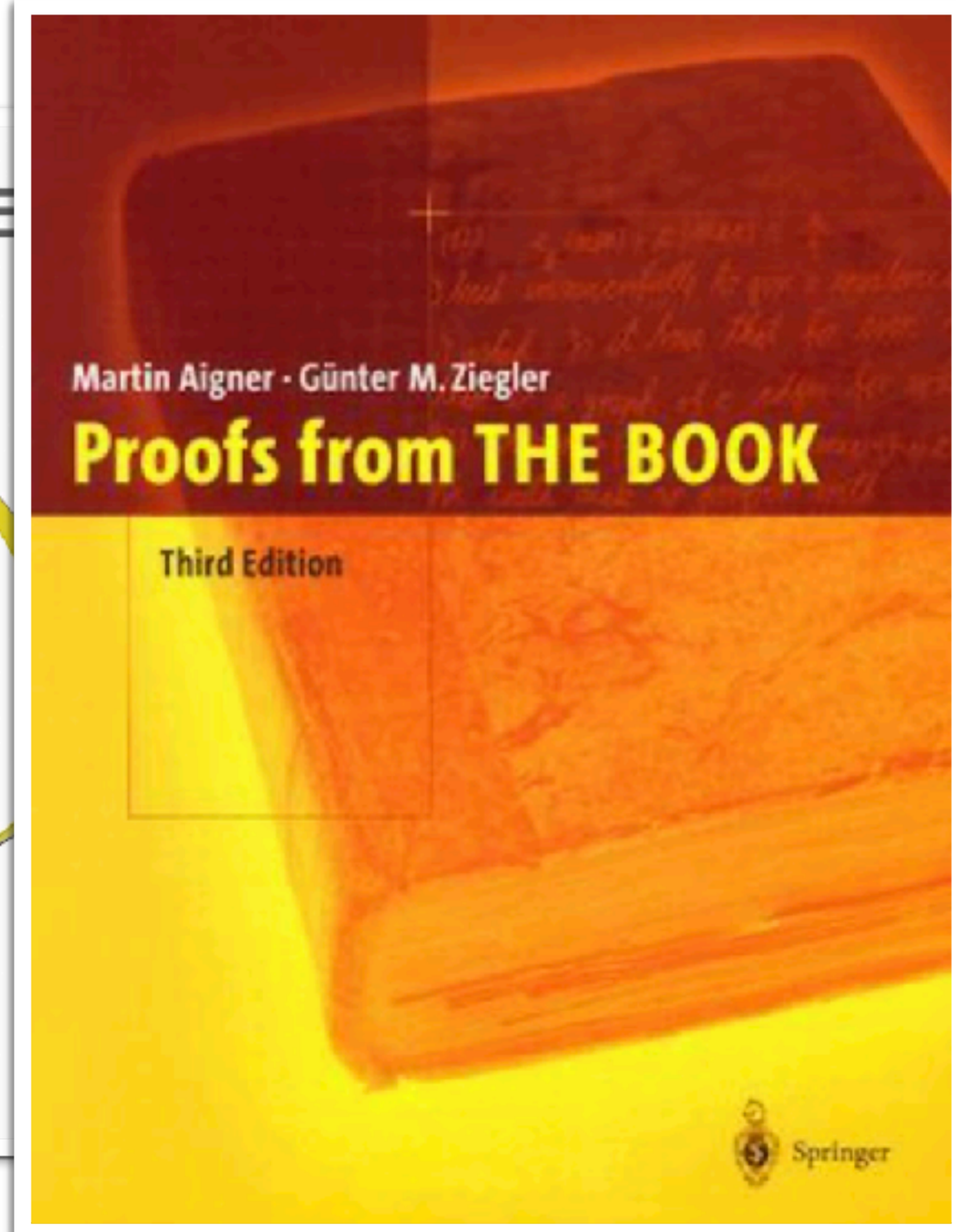
Proof. Triangulate S so that no new vertices are added. Every such triangulation has a coloring with three colors a, b, c. Let T_a be the set of vertices colored a, and assume that $|T_a| \leqslant |T_b| \leqslant |T_c|$. Choosing $T = T_a$ implies $|T| \leqslant n/3$. Finally, every point q of S lies in some triangle of S, and every triangle of S has a point p of T on it. Since triangles are convex, we have $pq \subset S$.

REFERENCE

1. V. CHVÁTAL, A combinatorial theorem in plane geometry, *J. Combinatorial Theory B* 18 (1975), 39–41.

0095-8956/78/0243-0374$02.00/0

Copyright © 1978 by Academic Press, Inc.
All rights of reproduction in any form reserved.



Overview

1. Introduction
2. Existence
3. Properties
4. Algorithms: Removing ears
5. Algorithms: Finding diagonals
6. Algorithms: Monotone polygons
7. Algorithms: Monotone decompositions
8. Faster algorithms
9. Application: Art Gallery problems
10. Application: Online triangulation

Online Triangulation

Triangulating Unknown Environments
using Robot Swarms
Aaron Becker
James McLurkin
SeoungKyou Lee
RICE
Sándor P. Fekete
Alexander Kröller
Christiane Schmidt
Technische
Universität
Braunschweig

Computational Geometry
Chapter 6: Point Triangulation

Sándor P. Fekete

Algorithms Division
Department of Computer Science
TU Braunschweig

Technische Universität Braunschweig

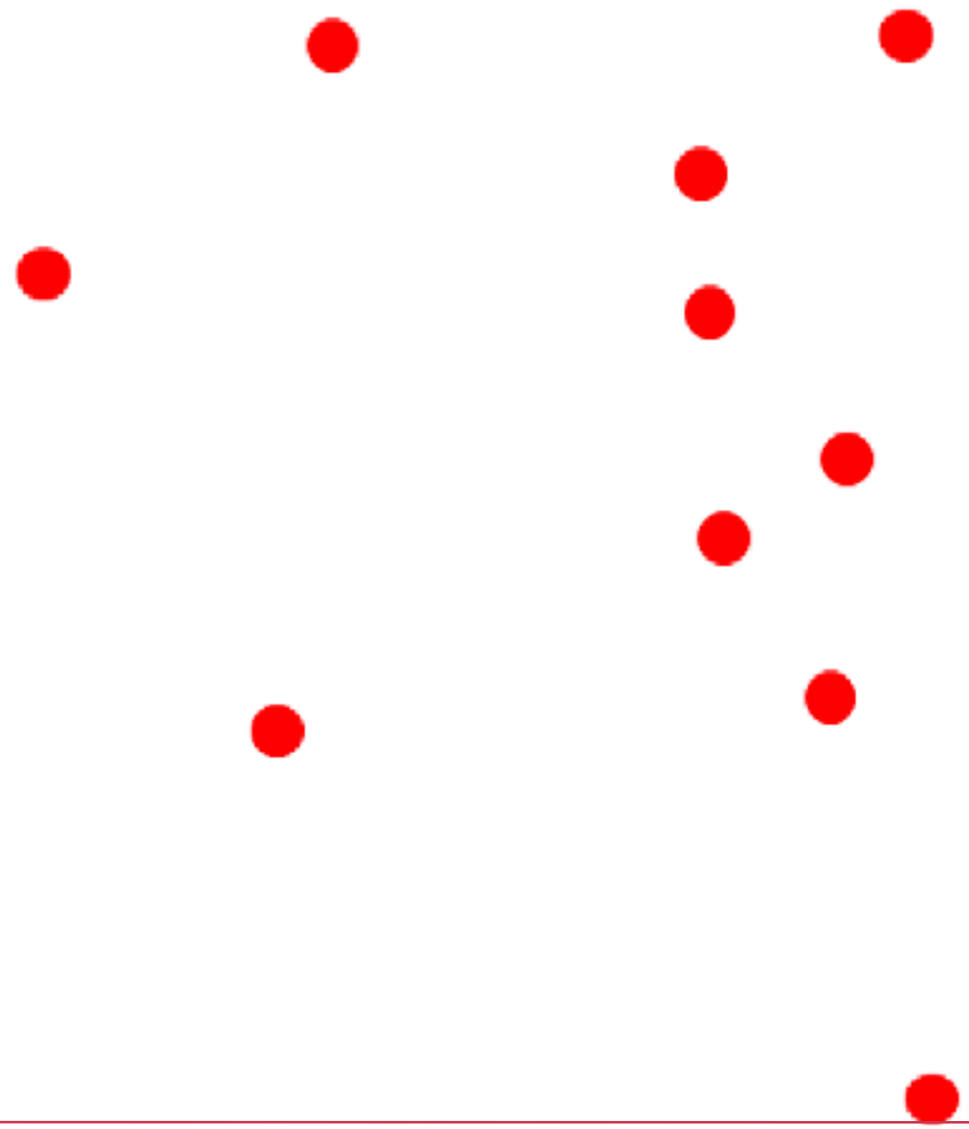
Overview

1. Introduction
2. Minimum angle and Delaunay triangulations
3. Minimum-weight triangulations
4. Min-max edge triangulations
5. Max-min edge triangulations

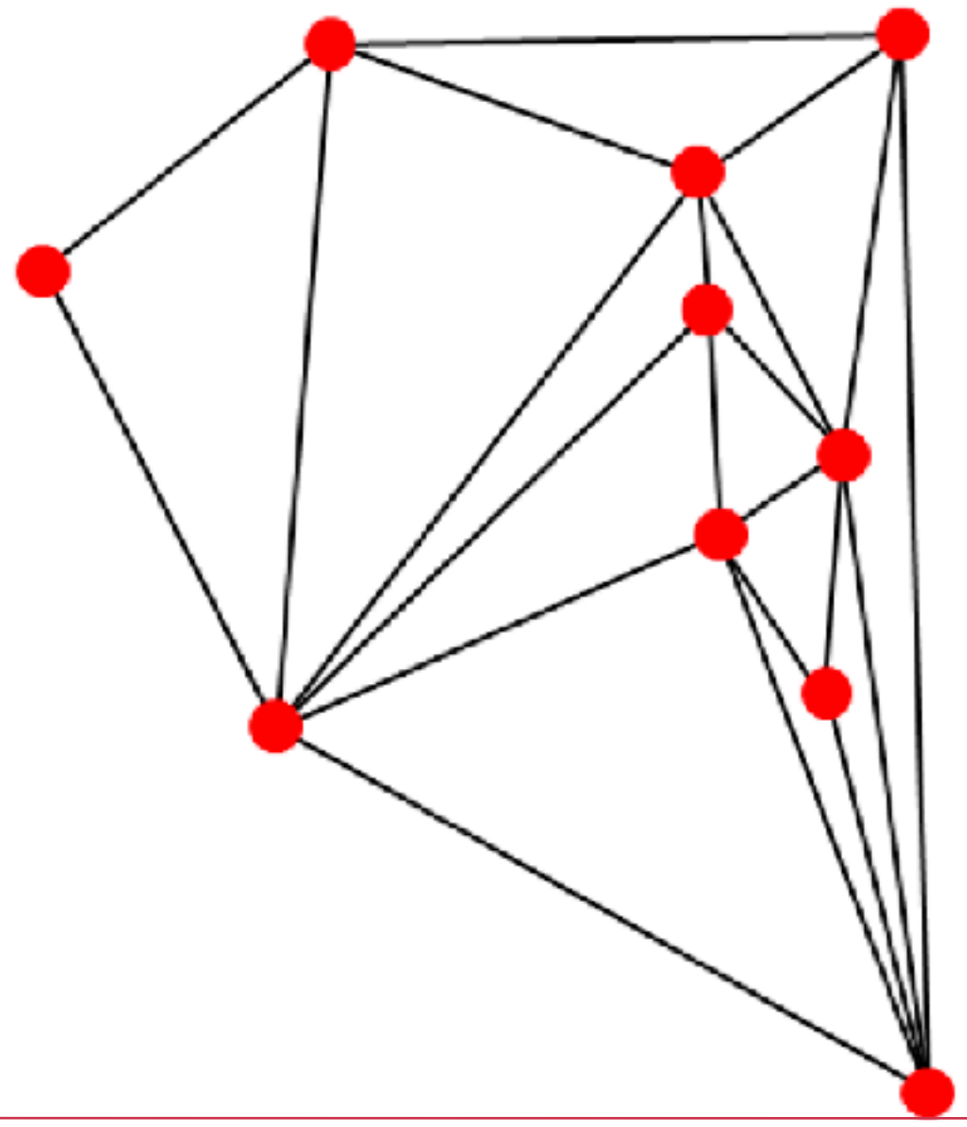
1 Introduction: The Problem

Triangulations

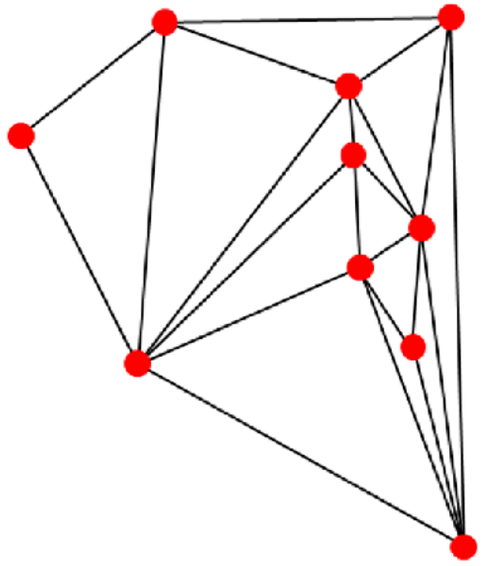
Triangulations



Triangulations

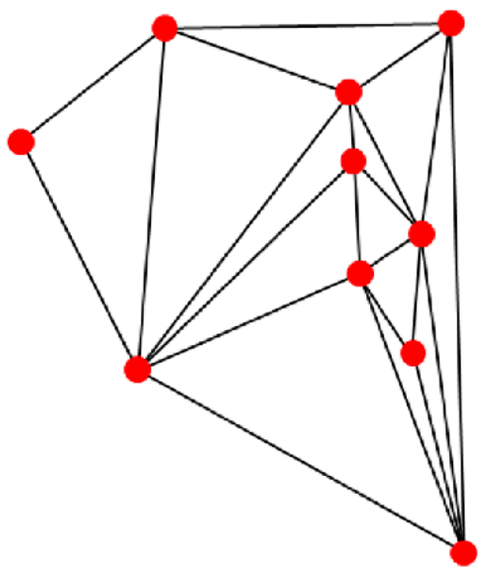


Triangulations



Triangulations

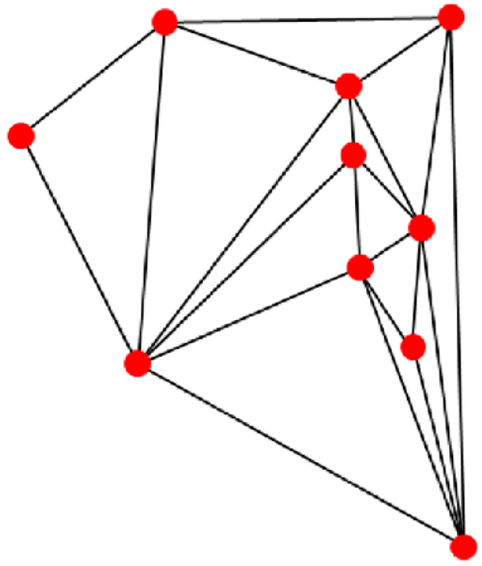
Problem 6.1



Triangulations

Problem 6.1

Given: A set P of points in $\mathbb{R}^2$

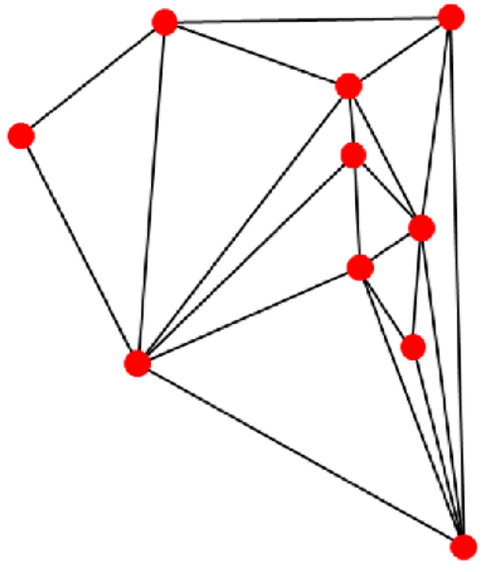


Triangulations

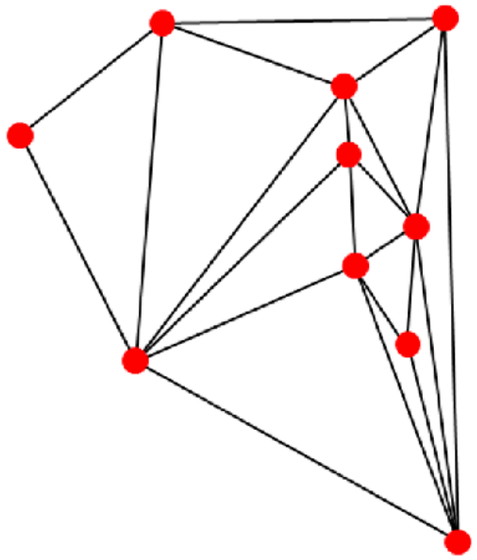
Problem 6.1

Given: A set P of points in $\mathbb{R}^2$

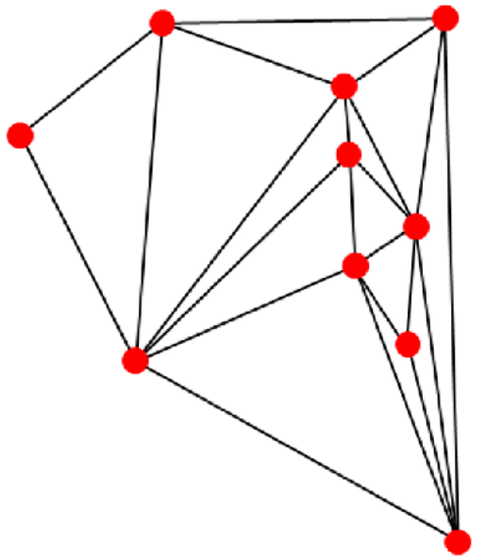
Wanted: Subdivision of $conv(P)$ into triangles with vertices in P and disjoint interior



Objective Functions

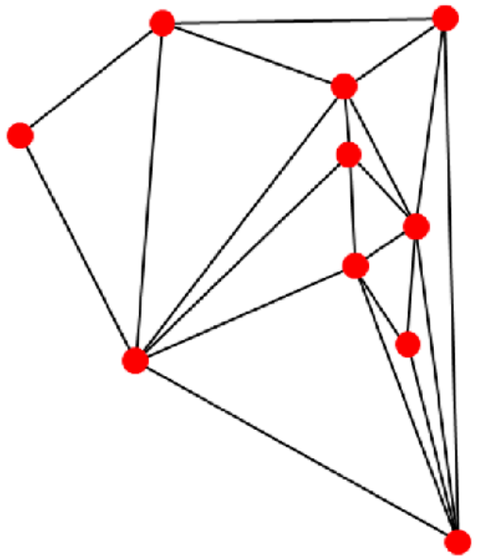


Objective Functions



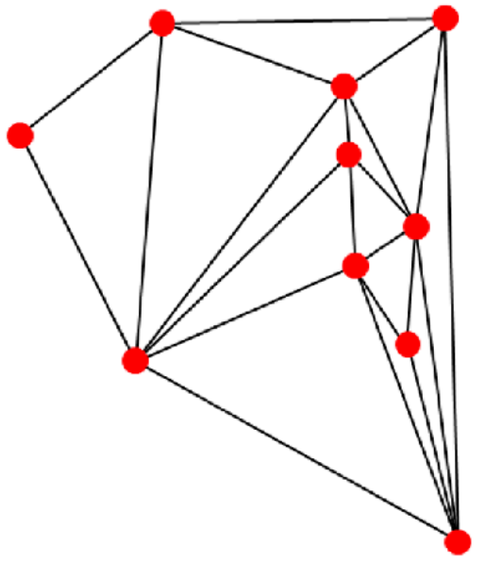
Objective Functions

2. Maximize minimum angle



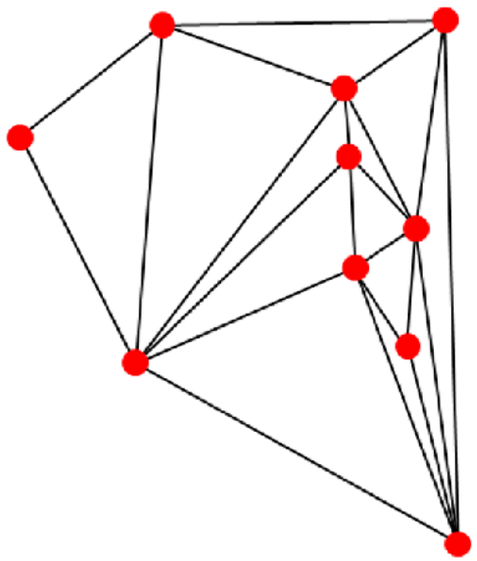
Objective Functions

2. Maximize minimum angle
3. Minimize total edge length



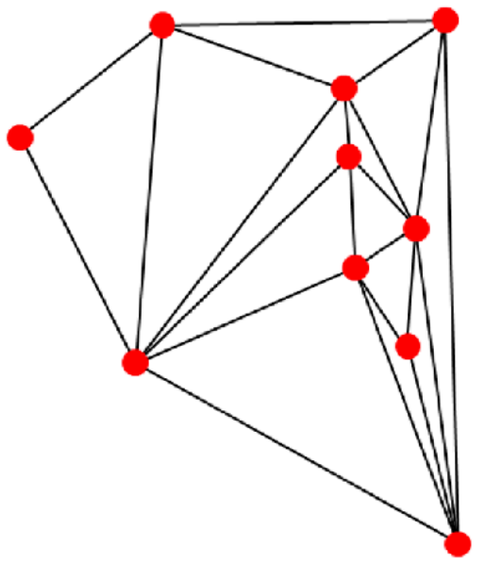
Objective Functions

2. Maximize minimum angle
3. Minimize total edge length
4. Minimize maximum edge length



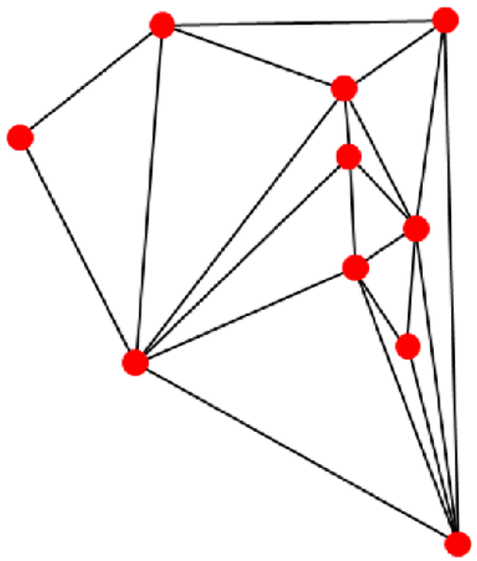
Objective Functions

2. Maximize minimum angle
3. Minimize total edge length
4. Minimize maximum edge length
5. Maximize minimum edge length

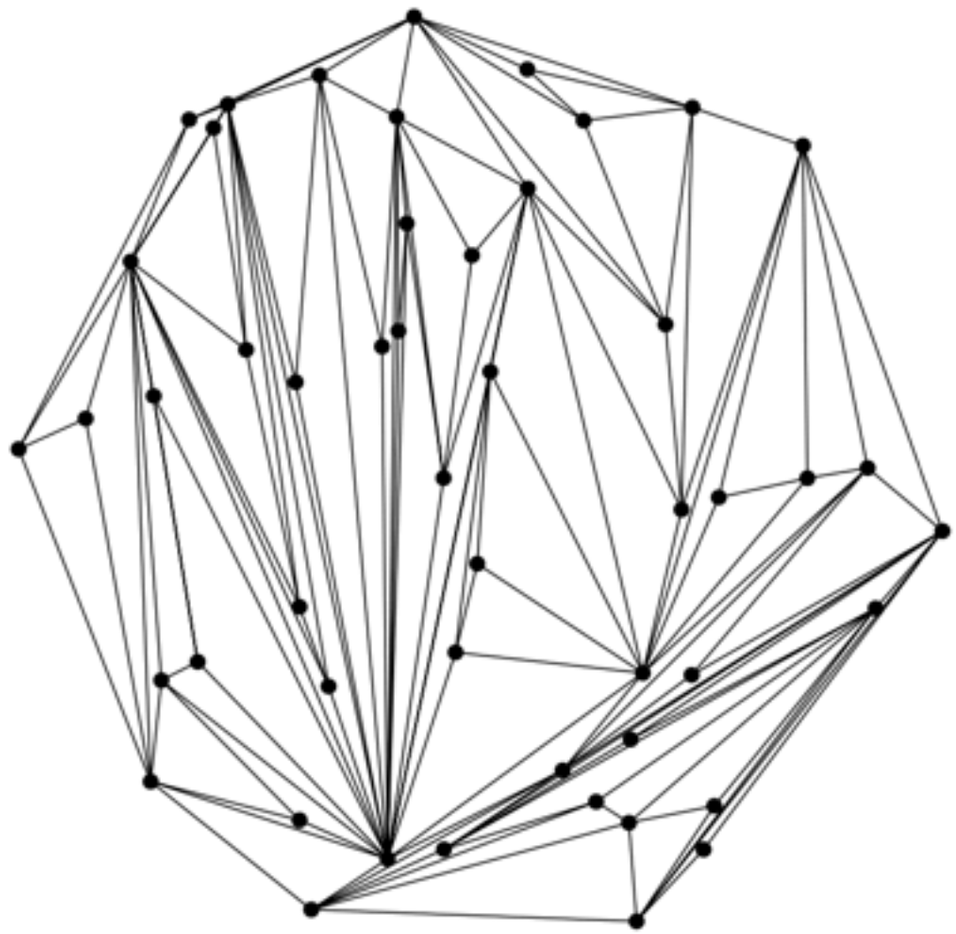


Objective Functions

2. Maximize minimum angle
3. Minimize total edge length
4. Minimize maximum edge length
5. Maximize minimum edge length

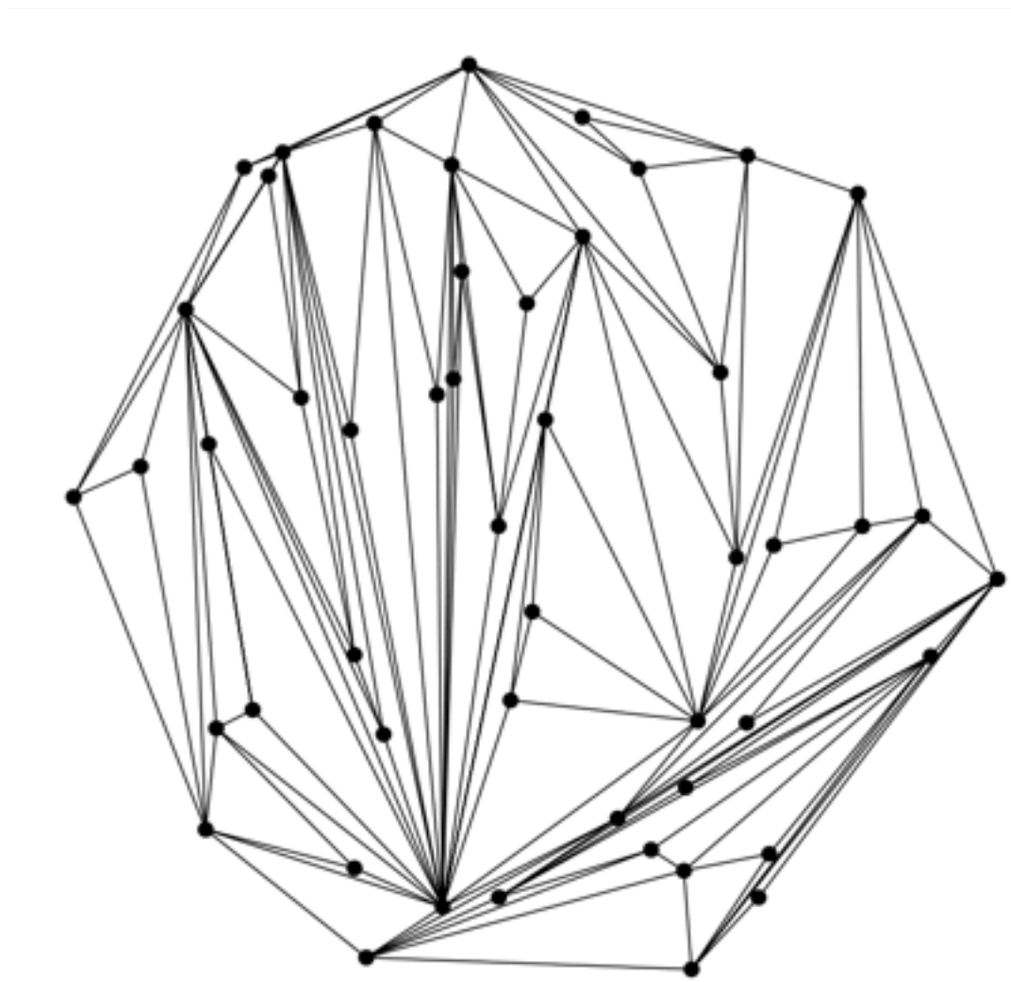


2 Delaunay triangulations

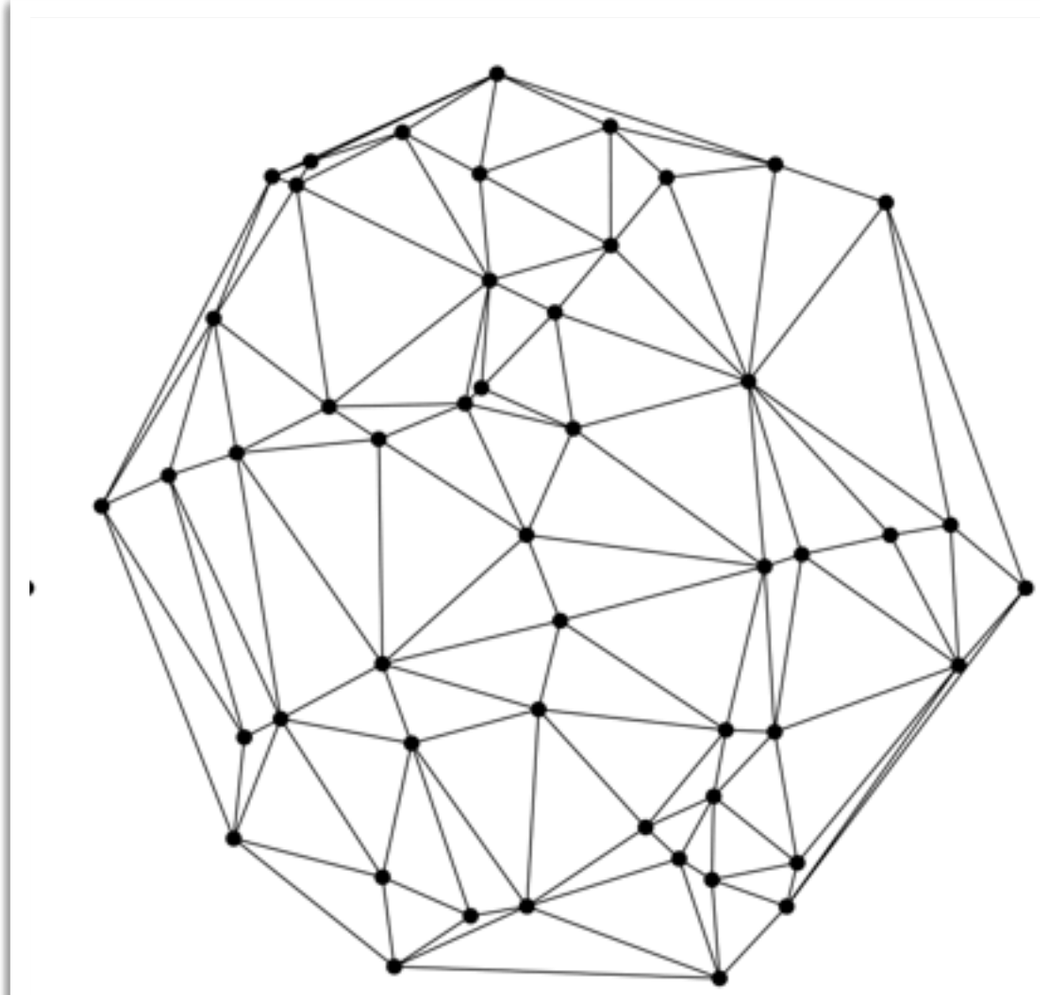


(a) Scan triangulation.

Triangulations



(a) Scan triangulation.



(b) Delaunay triangulation.

Delaunay triangulations [Delaunay, 1934]

Delaunay triangulations [Delaunay, 1934]

ИЗВЕСТИЯ АКАДЕМИИ НАУК СССР. 1934
BULLETIN DE L'ACADÉMIE DES SCIENCES DE L'URSS

Classe des sciences mathématiques et naturelles

Отделение математических и естественных наук

SUR LA SPHÈRE VIDE

À LA MÉMOIRE DE GEORGES VORONOÏ

Par B. DELAUNAY

(*Présenté par* I. Vinogradov, *membre de l'Académie*)

Soit donné un système quelconque de points dans l'espace à n dimensions. Je me propose de considérer une sphère se mouvant entre les points de ce système se rétrécissant et se dilatant à volonté et assujettie à la seule condition d'être «vide», c'est-à-dire de ne pas contenir dans son intérieur des points de ce système. C'est la «méthode de la sphère vide» que j'ai proposé pour la première fois dans une communication faite au Congrès de Toronto.

Je montre dans ce qui suit que le «théorème fondamental», qui est contenu dans le grand mémoire de Voronoï sur les formes quadratiques inseré dans les tomes 134 et 136 du Journal de Crelle, est une conséquence presque immédiate d'un lemme tout à fait général. Je pense que la démonstration plus simple du théorème fondamental de Voronoï, que je propose, contribuerat à répandre parmi les géomètres les resultats excessivement importants obtenus par Voronoï dans le mémoire mentionné du Journal de Crelle, qui était son dernier oeuvre.

§ 1. **Lemme général.** *Soient T des tétraèdres tout à fait arbitraires qui partagent uniformément l'espace à n dimensions étant contigus par des faces entières à $n-1$ dimensions et tels qu'un domaine quelconque limité (c'est-à-dire à diamètre limité) ait des points communs seulement avec un nombre limité de ces tétraèdres, alors la condition nécessaire et suffisante pour qu'aucune sphère circonscrite à un tel tétraèdre ne contienne dans son intérieur aucun sommet d'aucun de ces tétraèdres est que cela ait lieu pour chaque paire de deux de ces tétraèdres contigus par une face à $n-1$ dimensions, c'est-à-dire que dans chaque telle paire le sommet d'un de ces tétraèdres ne soit pas intérieur à la sphère circonscrite à l'autre, et réciproquement.*

Il est évident que cela est nécessaire. Mais cela est aussi suffisant. En effet, soit T_1 un de ces tétraèdres, et A un sommet d'un quelconque de ces tétraèdres. Soit B un point intérieur de T_1. Si le segment de droite AB, en passant d'un

Delaunay triangulations [Delaunay, 1934]

ИЗВЕСТИЯ АКАДЕМИИ НАУК СССР. 1934
BULLETIN DE L'ACADÉMIE DES SCIENCES DE L'URSS

Classe des sciences mathématiques et naturelles

Отделение математических и естественных наук

SUR LA SPHÈRE VIDE

À LA MÉMOIRE DE GEORGES VORONOÏ

Par B. DELAUNAY

(*Presenté par* I. Vinogradov, *membre de l'Académie*)

Soit donné un système quelconque de points dans l'espace à n dimensions. Je me propose de considérer une sphère se mouvant entre les points de ce système se rétrécissant et se dilatant à volonté et assujettie à la seule condition d'être «vide», c'est-à-dire de ne pas contenir dans son intérieur des points de ce système. C'est la «méthode de la sphère vide» que j'ai proposé pour la première fois dans une communication faite au Congrès de Toronto.

Je montre dans ce qui suit que le «theorème fondamental», qui est contenu dans le grand mémoire de Voronoï sur les formes quadratiques inseré dans les tomes 134 et 136 du Journal de Crelle, est une conséquence presque immédiate d'un lemme tout à fait général. Je pense que la démonstration plus simple du théorème fondamental de Voronoï, que je propose, contribuerat à répandre parmi les géomètres les resultats excessivement importants obtenus par Voronoï dans le mémoire mentionné du Journal de Crelle, qui était son dernier oeuvre.

§ 1. **Lemme général.** *Soient T des tétraèdres tout à fait arbitraires qui partagent uniformément l'espace à n dimensions étant contigus par des faces entières à $n-1$ dimensions et tels qu'un domaine quelconque limité (c'est-à-dire à diamètre limité) ait des points communs seulement avec un nombre limité de ces tétraèdres, alors la condition nécessaire et suffisante pour qu'aucune sphère circonscrite à un tel tétraèdre ne contienne dans son intérieur aucun sommet d'aucun de ces tétraèdres est que cela ait lieu pour chaque paire de deux de ces tétraèdres contigus par une face à $n-1$ dimensions, c'est-à-dire que dans chaque telle paire le sommet d'un de ces tétraèdres ne soit pas intérieur à la sphère circonscrite à l'autre, et réciproquement.*

Il est évident que cela est nécessaire. Mais cela est aussi suffisant. En effet, soit T_1 un de ces tétraèdres, et A un sommet d'un quelconque de ces tétraèdres. Soit B un point intérieur de T_1. Si le segment de droite AB, en passant d'un

Soit donné un système quelconque de points dans l'espace à n dimensions.

Delaunay triangulations [Delaunay, 1934]

ИЗВЕСТИЯ АКАДЕМИИ НАУК СССР. 1934
BULLETIN DE L'ACADÉMIE DES SCIENCES DE L'URSS

Classe des sciences mathématiques et naturelles

Отделение математических и естественных наук

SUR LA SPHÈRE VIDE

A LA MÉMOIRE DE GEORGES VORONOÏ

Par B. DELAUNAY

(*Presenté par* I. Vinogradov, *membre de l'Académie*)

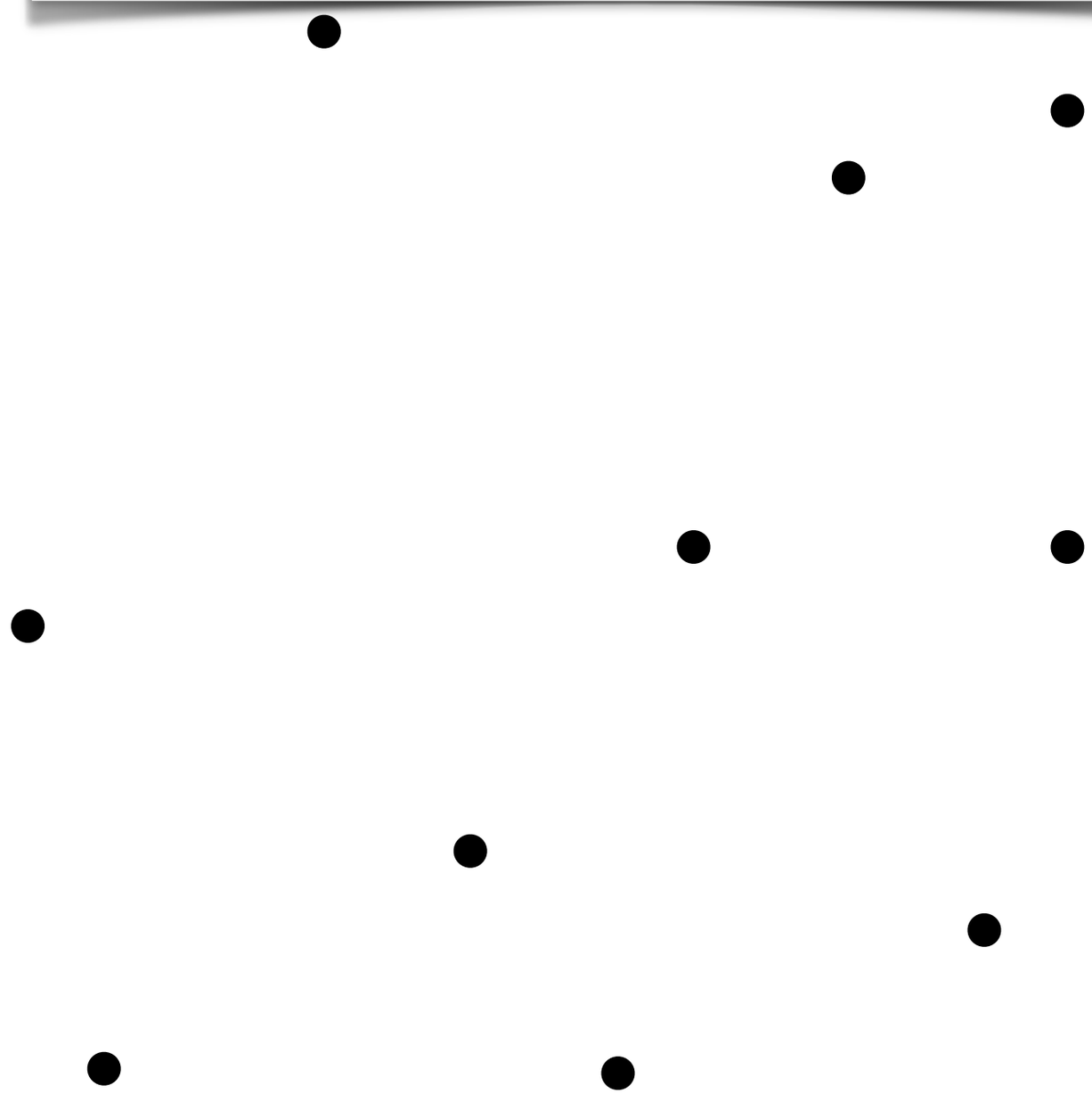
Soit donné un système quelconque de points dans l'espace à n dimensions. Je me propose de considérer une sphère se mouvant entre les points de ce système se rétrécissant et se dilatant à volonté et assujettie à la seule condition d'être «vide», c'est-à-dire de ne pas contenir dans son intérieur des points de ce système. C'est la «méthode de la sphère vide» que j'ai proposé pour la première fois dans une communication faite au Congrès de Toronto.

Je montre dans ce qui suit que le «theorème fondamental», qui est contenu dans le grand mémoire de Voronoï sur les formes quadratiques inseré dans les tomes 134 et 136 du Journal de Crelle, est une conséquence presque immédiate d'un lemme tout à fait général. Je pense que la démonstration plus simple du théorème fondamental de Voronoï, que je propose, contribuerat à répandre parmi les géomètres les resultats excessivement importants obtenus par Voronoï dans le mémoire mentionné du Journal de Crelle, qui était son dernier oeuvre.

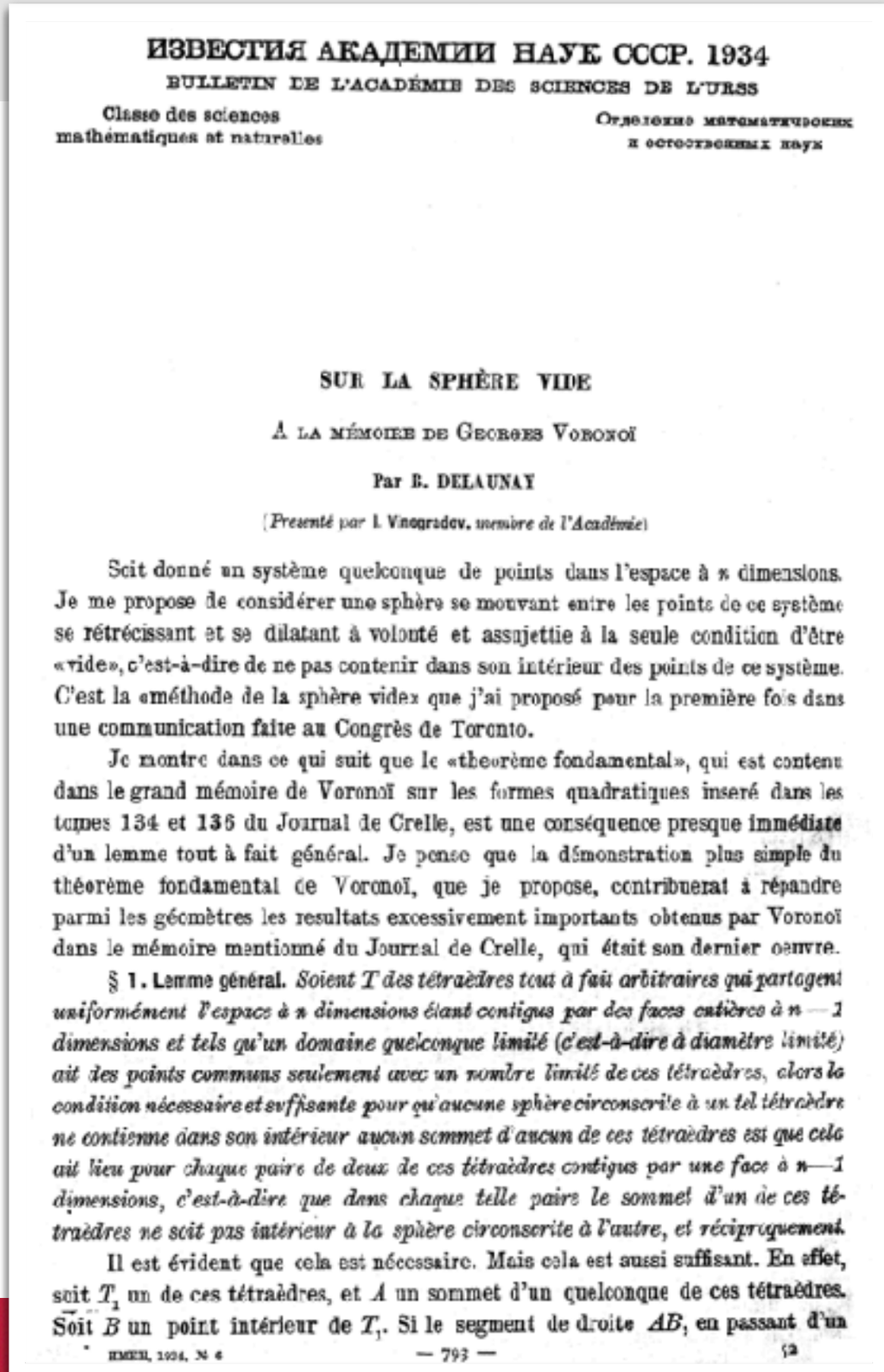
§ 1. Lemme général. *Soient T des tétraèdres tout à fait arbitraires qui partagent uniformément l'espace à n dimensions étant contigus par des faces entières à $n-1$ dimensions et tels qu'un domaine quelconque limité (c'est-à-dire à diamètre limité) ait des points communs seulement avec un nombre limité de ces tétraèdres, alors la condition nécessaire et suffisante pour qu'aucune sphère circonscrite à un tel tétraèdre ne contienne dans son intérieur aucun sommet d'aucun de ces tétraèdres est que cela ait lieu pour chaque paire de deux de ces tétraèdres contigus par une face à $n-1$ dimensions, c'est-à-dire que dans chaque telle paire le sommet d'un de ces tétraèdres ne soit pas intérieur à la sphère circonscrite à l'autre, et réciproquement.*

Il est évident que cela est nécessaire. Mais cela est aussi suffisant. En effet, soit T_1 un de ces tétraèdres, et A un sommet d'un quelconque de ces tétraèdres. Soit B un point intérieur de T_1. Si le segment de droite AB, en passant d'un

Soit donné un système quelconque de points dans l'espace à n dimensions.



Delaunay triangulations [Delaunay, 1934]



ИЗВЕСТИЯ АКАДЕМИИ НАУК СССР. 1934
BULLETIN DE L'ACADÉMIE DES SCIENCES DE L'URSS

Classe des sciences mathématiques et naturelles

Отделение математических и естественных наук

SUR LA SPHÈRE VIDE

A LA MÉMOIRE DE GEORGES VORONOÏ

Par B. DELAUNAY

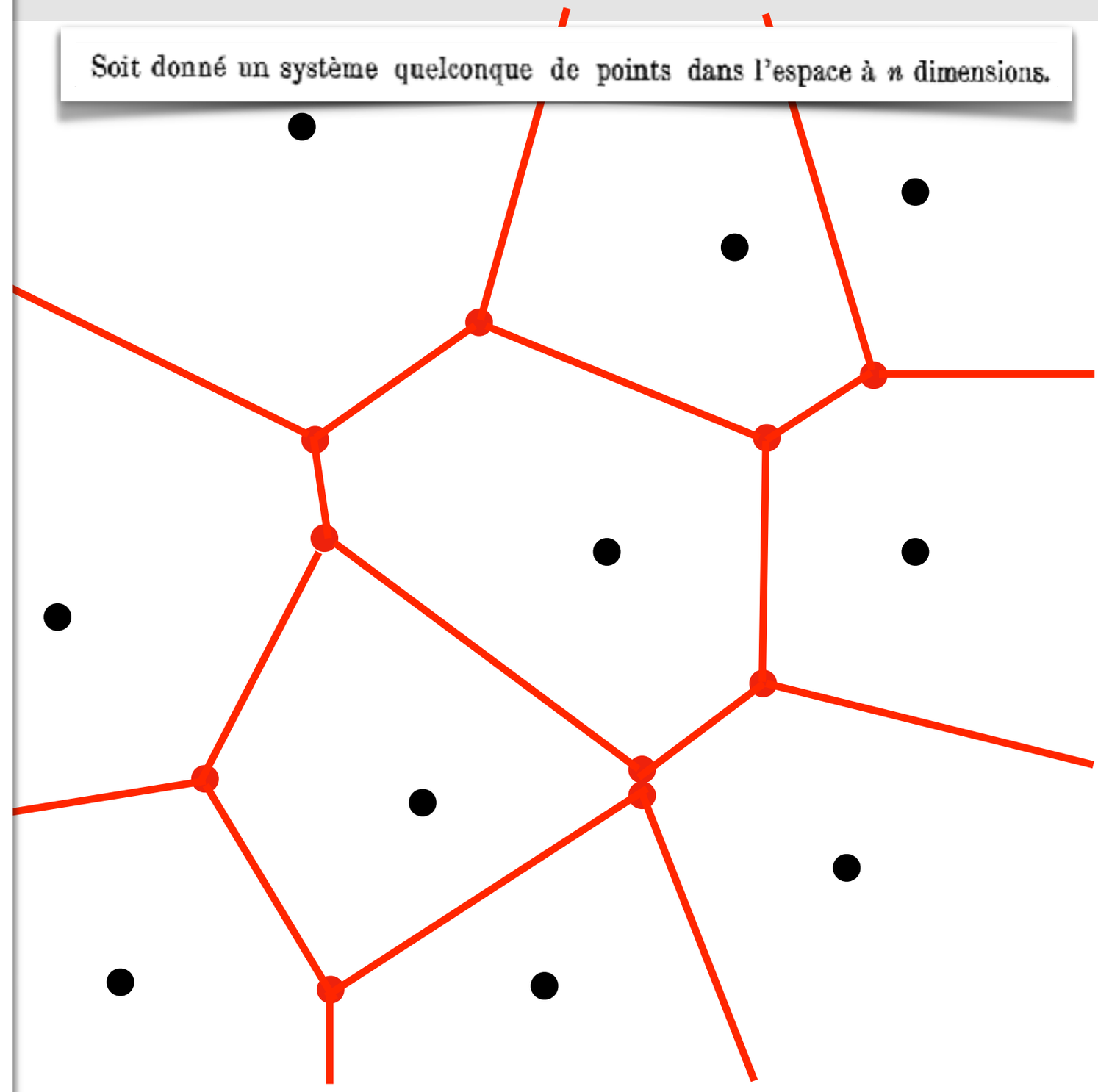
(*Présenté par* I. Vinogradov, *membre de l'Académie*)

Soit donné un système quelconque de points dans l'espace à n dimensions. Je me propose de considérer une sphère se mouvant entre les points de ce système se rétrécissant et se dilatant à volonté et assujettie à la seule condition d'être «vide», c'est-à-dire de ne pas contenir dans son intérieur des points de ce système. C'est la «méthode de la sphère vide» que j'ai proposé pour la première fois dans une communication faite au Congrès de Toronto.

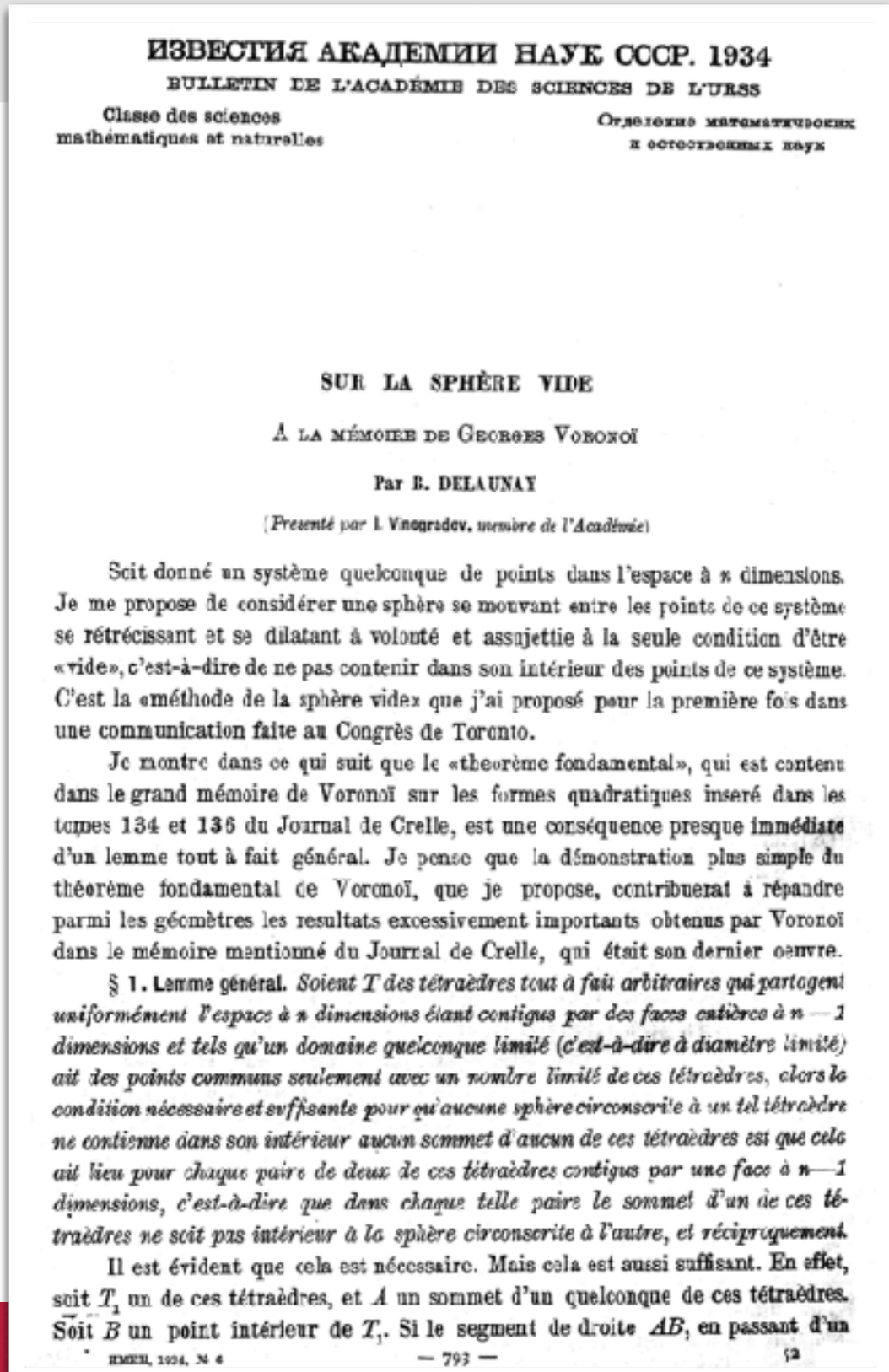
Je montre dans ce qui suit que le «théorème fondamental», qui est contenu dans le grand mémoire de Voronoï sur les formes quadratiques inseré dans les tomes 134 et 136 du Journal de Crelle, est une conséquence presque immédiate d'un lemme tout à fait général. Je pense que la démonstration plus simple du théorème fondamental de Voronoï, que je propose, contribuerat à répandre parmi les géomètres les résultats excessivement importants obtenus par Voronoï dans le mémoire mentionné du Journal de Crelle, qui était son dernier oeuvre.

§ 1. **Lemme général.** *Soient T des tétraèdres tout à fait arbitraires qui partagent uniformément l'espace à n dimensions étant contigus par des faces entières à $n-1$ dimensions et tels qu'un domaine quelconque limité (c'est-à-dire à diamètre limité) ait des points communs seulement avec un nombre limité de ces tétraèdres, alors la condition nécessaire et suffisante pour qu'aucune sphère circonscrite à un tel tétraèdre ne contienne dans son intérieur aucun sommet d'aucun de ces tétraèdres est que cela ait lieu pour chaque paire de deux de ces tétraèdres contigus par une face à $n-1$ dimensions, c'est-à-dire que dans chaque telle paire le sommet d'un de ces tétraèdres ne soit pas intérieur à la sphère circonscrite à l'autre, et réciproquement.*

Il est évident que cela est nécessaire. Mais cela est aussi suffisant. En effet, soit T_1 un de ces tétraèdres, et A un sommet d'un quelconque de ces tétraèdres. Soit B un point intérieur de T_1. Si le segment de droite AB, en passant d'un



Delaunay triangulations [Delaunay, 1934]



ИЗВЕСТИЯ АКАДЕМИИ НАУК СССР. 1934

BULLETIN DE L'ACADÉMIE DES SCIENCES DE L'URSS

Classe des sciences mathématiques et naturelles

Отделение математических и естественных наук

SUR LA SPHÈRE VIDE

A LA MÉMOIRE DE GEORGES VORONOÏ

Par B. DELAUNAY

(*Présenté par* I. Vinogradov, *membre de l'Académie*)

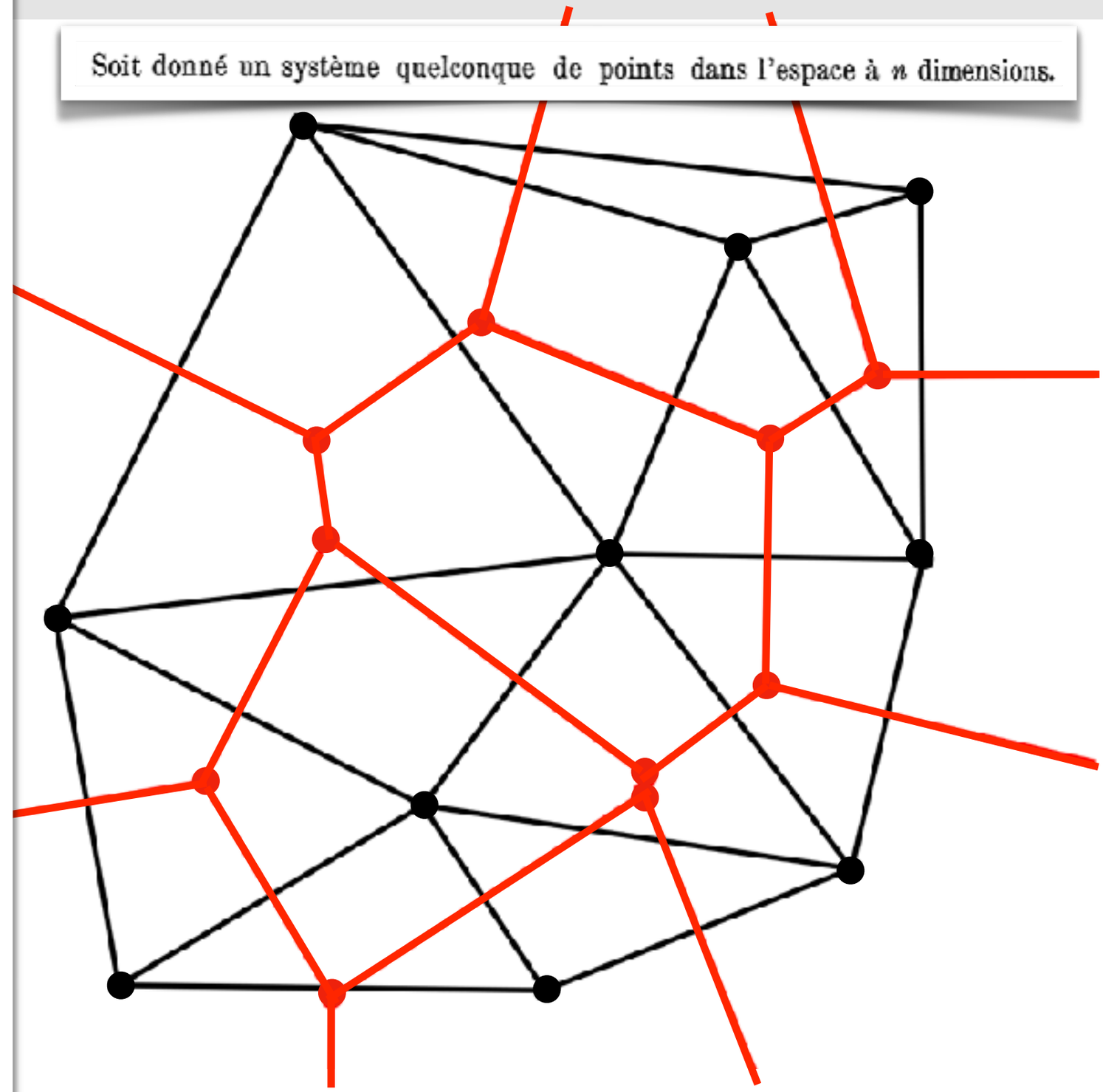
Soit donné un système quelconque de points dans l'espace à n dimensions. Je me propose de considérer une sphère se mouvant entre les points de ce système se rétrécissant et se dilatant à volonté et assujettie à la seule condition d'être «vide», c'est-à-dire de ne pas contenir dans son intérieur des points de ce système. C'est la «méthode de la sphère vide» que j'ai proposé pour la première fois dans une communication faite au Congrès de Toronto.

Je montre dans ce qui suit que le «theorème fondamental», qui est contenu dans le grand mémoire de Voronoï sur les formes quadratiques inseré dans les tomes 134 et 136 du Journal de Crelle, est une conséquence presque immédiate d'un lemme tout à fait général. Je pense que la démonstration plus simple du théorème fondamental de Voronoï, que je propose, contribuerai à répandre parmi les géomètres les resultats excessivement importants obtenus par Voronoï dans le mémoire mentionné du Journal de Crelle, qui était son dernier oeuvre.

§ 1. Lemme général. *Soient T des tétraèdres tout à fait arbitraires qui partagent uniformément l'espace à n dimensions étant contigus par des faces entières à $n-1$ dimensions et tels qu'un domaine quelconque limité (c'est-à-dire à diamètre limité) ait des points communs seulement avec un nombre limité de ces tétraèdres, alors la condition nécessaire et suffisante pour qu'aucune sphère circonscrite à un tel tétraèdre ne contienne dans son intérieur aucun sommet d'aucun de ces tétraèdres est que cela ait lieu pour chaque paire de deux de ces tétraèdres contigus par une face à $n-1$ dimensions, c'est-à-dire que dans chaque telle paire le sommet d'un de ces tétraèdres ne soit pas intérieur à la sphère circonscrite à l'autre, et réciproquement.*

Il est évident que cela est nécessaire. Mais cela est aussi suffisant. En effet, soit T_1 un de ces tétraèdres, et A un sommet d'un quelconque de ces tétraèdres. Soit B un point intérieur de T_1. Si le segment de droite AB, en passant d'un

ИМЕН, 1934, № 6 — 793 — 52



Delaunay triangulation maximizes minimum angle [Sibson 1978]

Delaunay triangulation maximizes minimum angle [Sibson 1978]

Computational Geometry
Lecture Notes HS 2013

Bernd Gärtner <gaertner@inf.ethz.ch>
Michael Hoffmann <hoffmann@inf.ethz.ch>

Delaunay triangulation maximizes minimum angle [Sibson 1978]

Computational Geometry
Lecture Notes HS 2013

Bernd Gärtner <gaertner@inf.ethz.ch>
Michael Hoffmann <hoffmann@inf.ethz.ch>

Definition 6.8 *A triangulation of a finite point set* $P \subset \mathbb{R}^2$ *is called a* **Delaunay triangulation**, *if the circumcircle of every triangle is empty, that is, there is no point from* P *in its interior.*

Delaunay triangulation maximizes minimum angle [Sibson 1978]

Computational Geometry
Lecture Notes HS 2013

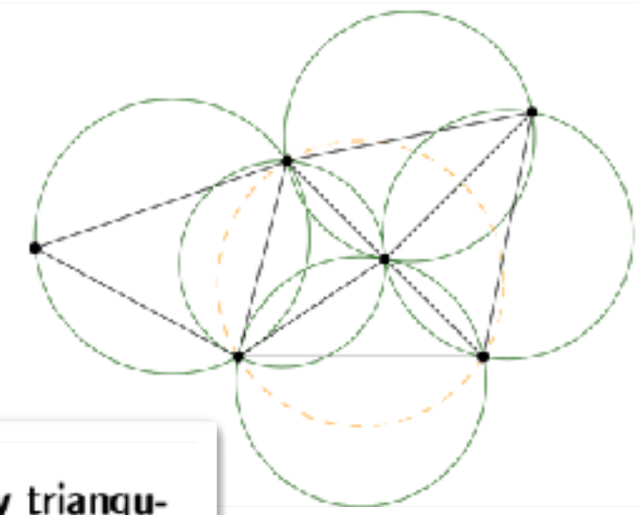
Bernd Gärtner <gaertner@inf.ethz.ch>
Michael Hoffmann <hoffmann@inf.ethz.ch>

Definition 6.8 *A triangulation of a finite point set* $P \subset \mathbb{R}^2$ *is called a* **Delaunay triangulation**, *if the circumcircle of every triangle is empty, that is, there is no point from* P *in its interior.*

Delaunay triangulation maximizes minimum angle [Sibson 1978]

Computational Geometry
Lecture Notes HS 2013

Bernd Gärtner <gaertner@inf.ethz.ch>
Michael Hoffmann <hoffmann@inf.ethz.ch>



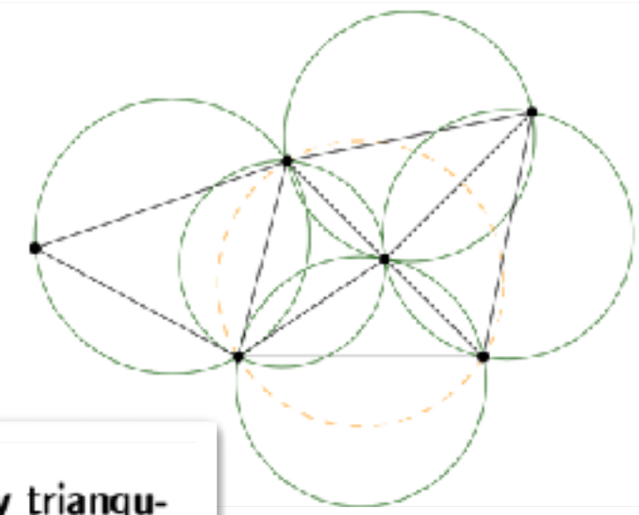
Definition 6.8 *A triangulation of a finite point set* $P \subset \mathbb{R}^2$ *is called a* **Delaunay triangulation**, *if the circumcircle of every triangle is empty, that is, there is no point from* P *in its interior.*

Theorem 6.18 *Let* $P \subseteq \mathbb{R}^2$ *be a finite set of points in general position (not all collinear and no four cocircular). Let* $\mathcal{D}^*$ *be the unique Delaunay triangulation of* P*, and let* $\mathcal{T}$ *be any triangulation of* P*. Then* $A(\mathcal{T}) \leqslant A(\mathcal{D}^*)$.

Delaunay triangulation maximizes minimum angle [Sibson 1978]

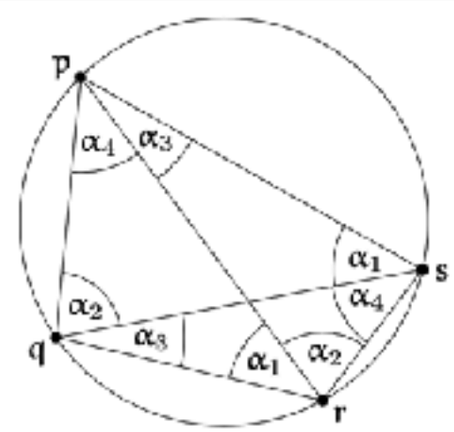
Computational Geometry
Lecture Notes HS 2013

Bernd Gärtner <gaertner@inf.ethz.ch>
Michael Hoffmann <hoffmann@inf.ethz.ch>



Definition 6.8 *A triangulation of a finite point set* $P \subset \mathbb{R}^2$ *is called a* **Delaunay triangulation**, *if the circumcircle of every triangle is empty, that is, there is no point from* P *in its interior.*

Theorem 6.18 *Let* $P \subseteq \mathbb{R}^2$ *be a finite set of points in general position (not all collinear and no four cocircular). Let* $\mathcal{D}^*$ *be the unique Delaunay triangulation of* P*, and let* $\mathcal{T}$ *be any triangulation of* P*. Then* $A(\mathcal{T}) \leqslant A(\mathcal{D}^*)$.

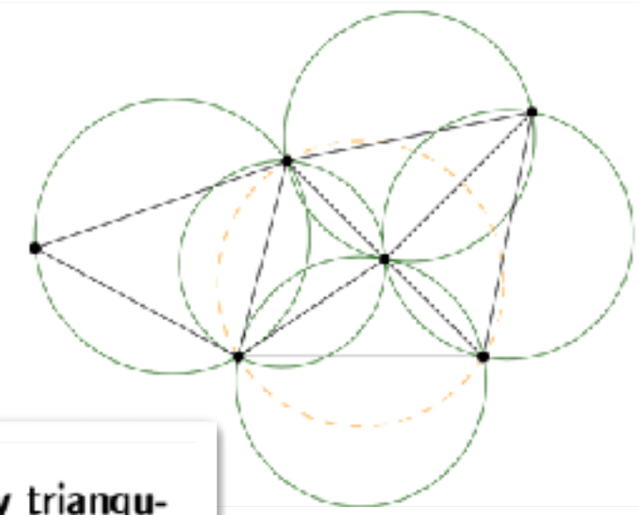


(a) Four cocircular points and the induced eight angles.

Delaunay triangulation maximizes minimum angle [Sibson 1978]

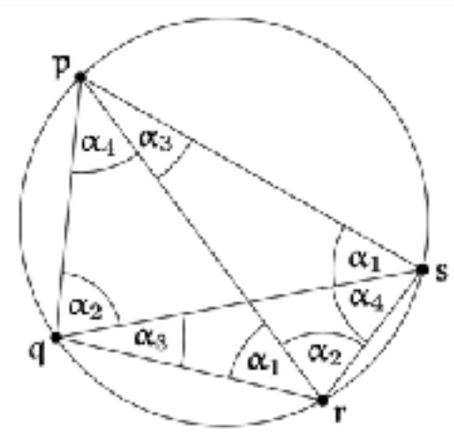
Computational Geometry
Lecture Notes HS 2013

Bernd Gärtner <gaertner@inf.ethz.ch>
Michael Hoffmann <hoffmann@inf.ethz.ch>

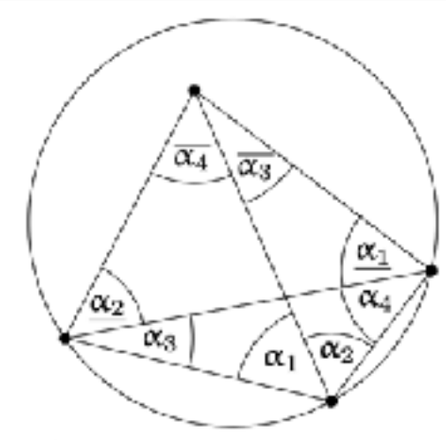


Definition 6.8 *A triangulation of a finite point set* $P \subset \mathbb{R}^2$ *is called a* **Delaunay triangulation**, *if the circumcircle of every triangle is empty, that is, there is no point from* P *in its interior.*

Theorem 6.18 *Let* $P \subseteq \mathbb{R}^2$ *be a finite set of points in general position (not all collinear and no four cocircular). Let* $\mathcal{D}^*$ *be the unique Delaunay triangulation of* P*, and let* $\mathcal{T}$ *be any triangulation of* P*. Then* $A(\mathcal{T}) \leqslant A(\mathcal{D}^*)$.



(a) Four cocircular points and the induced eight angles.



(b) The situation before a flip.

Delaunay triangulation maximizes minimum angle [Sibson 1978]

Locally equiangular triangulations

R. Sibson

School of Mathematics, University of Bath, Claverton Down, Bath, Avon BA2 7AY

Lawson (1972) has given a criterion of local equiangularity of a triangulation of the convex hull of a finite set of distinct points in the plane. In this note it is shown that (with suitable modifications to deal with degeneracy) there is only one such triangulation, and that it is the Delaunay triangulation (Rogers, 1964).

(Received March 1977)

The initial step in the construction of interpolating functions in two dimensions is the construction of the triangular grid on which the interpolation is based. Usually this grid is not given as part of the data in the problem; all that is available is a finite set of distinct points (*data sites*) at each of which the function and any relevant derivatives are evaluated. Triangulating a set of data sites means selecting a triangulation of the convex hull of the set, the vertices of that triangulation being all and only the data sites. This triangulation may then need further refinement, with the addition of extra vertices, to give the regions on which the interpolating function takes a simple form. This problem of further subdivision, which will not concern us here, is discussed by Powell and Sabin (1977).

A triangulation is regarded as 'good' for the purposes of interpolation if its triangles are nearly equiangular. When the data sites are placed almost as part of a regular triangular lattice, there is little doubt about which triangulation to choose and no difficulty over constructing it. But in practice this even spacing may well fail to hold. With arbitrarily placed points a close approach to equiangularity is seldom possible, and a criterion is needed for assessing the acceptability of a triangulation. Lawson (1972) has suggested such a criterion, which he calls the *max-min angle criterion*. This criterion requires that the diagonal of every convex quadrilateral occurring in the triangulation should be well chosen, in the sense that it should make the two resultant triangles as nearly equiangular as possible. A formal statement of the criterion is as follows:

> If two triangles in the triangulation share a common edge, they define a quadrilateral with that common edge as diagonal. If that quadrilateral is strictly convex (that is, each vertex is an extremal point of it) then replacement of the chosen diagonal by the alternative one must not increase the minimum of the six angles in the two triangles making up the quadrilateral, and this must hold for all such strictly convex quadrilaterals.

We shall call such triangulations *locally equiangular*.

It is the purpose of this note to show that there is only one locally equiangular triangulation of the convex hull of a finite set of distinct data sites, and to identify that triangulation as the Delaunay triangulation, the dual of the Dirichlet/Voronoi/Thiessen tessellation. These constructs are described in detail by Rogers (1964); the diversity of names is a consequence of their independent development in various different applications. The Dirichlet tessellation of a finite set of distinct data sites is obtained by associating with each data site a *tile* consisting of that part of the plane strictly closer to its generating data site than to any other. Clearly the tile of the data site P is the intersection of the open half-planes containing P and bounded by the perpendicular bisectors of lines PQ for all other data sites Q. Thus tiles are open convex polygonal regions. The tiles of data sites lying on the boundary of the convex hull of the set extend to infinity; other tiles are bounded. Only a few perpendicular bisectors are actually effective in delimiting each tile. Two tiles which share a boundary segment are said to be *contiguous*, as are their generating data sites. Normally tiles meet in threes; thus if contiguous data sites are joined by edges, a triangulation of the convex hull results. This is the Delaunay triangulation. Each point where three tiles meet is the circumcentre of the corresponding Delaunay triangle. **Fig. 1**, taken from Green and Sibson (1978), illustrates the constructs. Occasionally a degeneracy may occur: four or more tiles may meet at a common point. The generating data sites of such multiple points must be concyclic, with the multiple point as centre, and the Delaunay triangle becomes a cyclic polygon. It is here and here only that any ambiguity can arise. We shall call the Delaunay construct a *pretriangulation*, and shall speak of *triangulations* which are *completions*

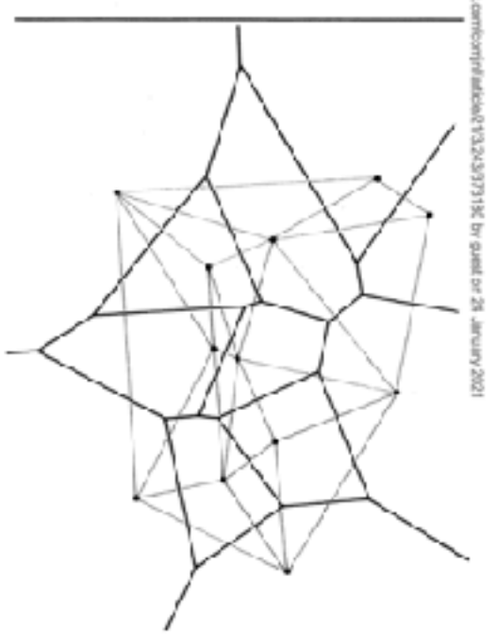
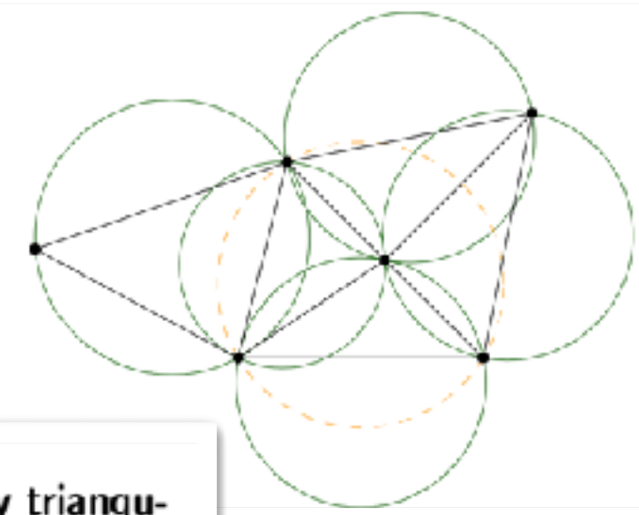


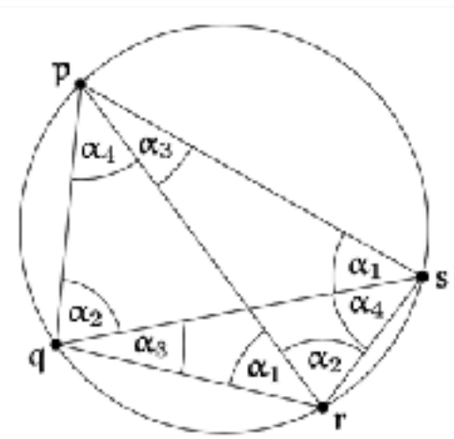
Fig. 1 The Dirichlet tessellation (bold lines) and Delaunay triangulation (fine lines) for a small-scale configuration

Downloaded from https://academic.oup.com/comjnl/article/21/2/243/373136 by guest on 26 January 2021

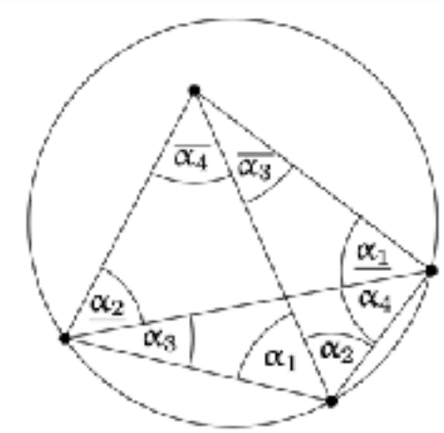


set $P \subset \mathbb{R}^2$ *is called a* Delaunay triangu-
empty, that is, there is no point from

ts in general position (not all collinear
e Delaunay triangulation of P, *and let*
$\mathcal{D}^*$).



(a) Four cocircular points and the induced eight angles.



(b) The situation before a flip.

Minimizing maximum angle [Edelsbrunner, Tan, Waupotitsch 1992]

Minimizing maximum angle [Edelsbrunner, Tan, Waupotitsch 1992]

AN $O(n^2 \log n)$ TIME ALGORITHM FOR THE MINMAX ANGLE TRIANGULATION *

HERBERT EDELSBRUNNER[†], TIOW SENG TAN[†] AND ROMAN WAUPOTITSCH[†]

Abstract. We show that a triangulation of a set of n points in the plane that minimizes the maximum angle can be computed in time $O(n^2 \log n)$ and space $O(n)$. The algorithm is fairly easy to implement and is based on the edge-insertion scheme that iteratively improves an arbitrary initial triangulation. It can be extended to the case where edges are prescribed, and, within the same time- and space-bounds, it can lexicographically minimize the sorted angle vector if the point set is in general position. Experimental results on the efficiency of the algorithm and the quality of the triangulations obtained are included.

Key words. Computational geometry, two dimensions, triangulations, minmax angle criterion, iterative improvement, edge-insertion

AMS(MOS) subject classifications. 68C05, 65M50

Appear in: *SIAM J. Scientific & Statistical Computing*, **13** (4), 994–1008, (1992)

1. Introduction. Let S be a finite set of points in the Euclidean plane. A *triangulation* of S is a maximally connected straight line plane graph whose vertices are the points of S. By maximality, each face is a triangle except for the exterior face which is the complement of the convex hull of S. Occasionally, we will call a triangulation of a finite point set a *general triangulation* in order to distinguish it from a *constrained triangulation* which is a triangulation of a finite point set where some edges are prescribed. A special case of a constrained triangulation is the so-called *polygon triangulation* where S is the set of vertices of a simple polygon and the edges of the polygon are prescribed. In this paper only the triangles *inside* the polygon will be of interest.

For a given set of n points there are, in general, exponentially many triangulations. Among them one can choose those that satisfy certain requirements or optimize certain objective functions. Different properties are desirable for different applications in areas such as finite element analysis [1, 3, 23], computational geometry [21], and surface approximation [12, 18]. The following are some important types of triangulations that optimize certain objective functions.

(i) The *Delaunay triangulation* has the property that the circumcircle of any triangle does not enclose any vertex [5].

(ii) The *constrained Delaunay triangulation* has the same property except that visibility constraints depending on the enforced edges are introduced [13].

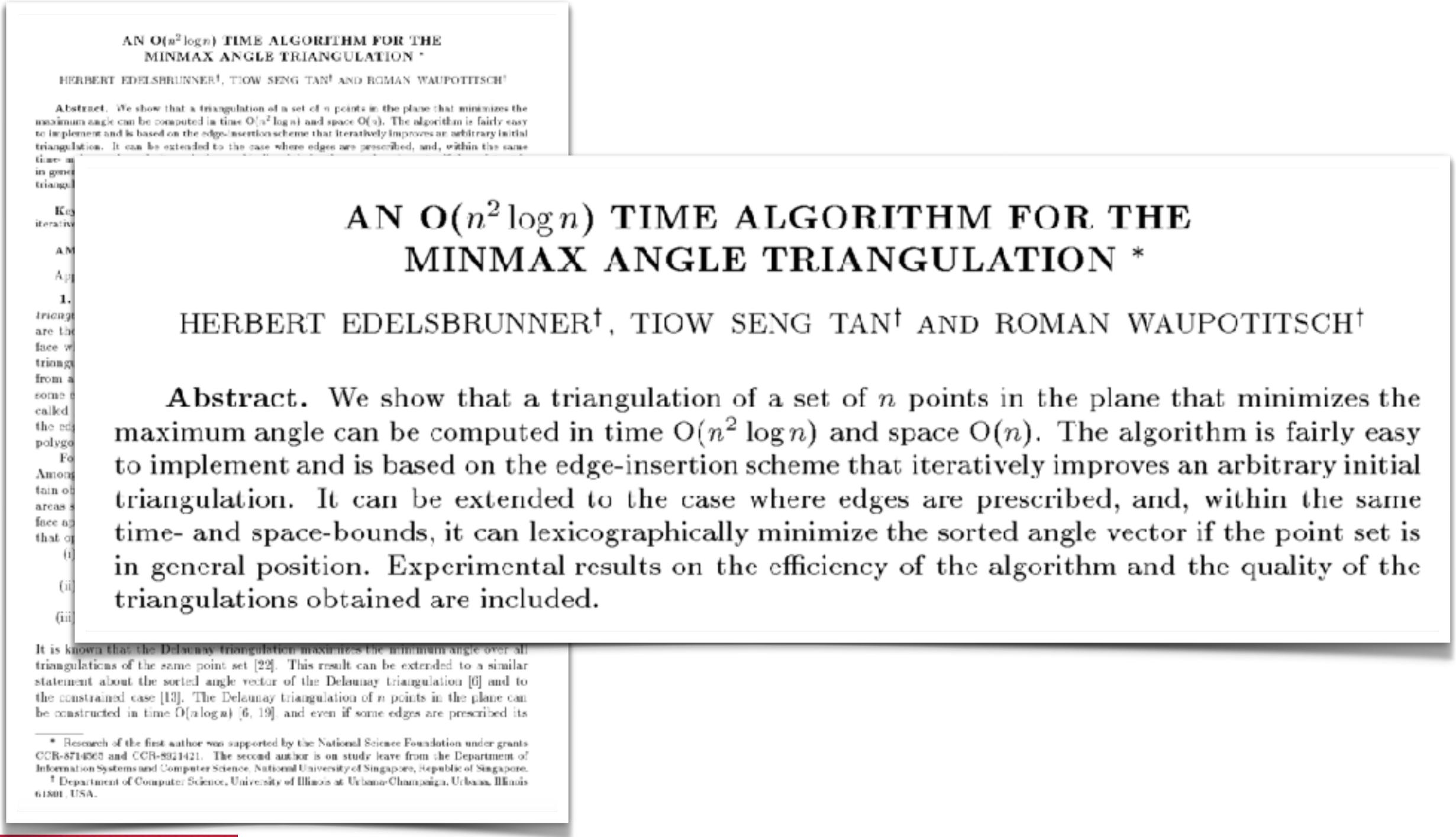
(iii) The *minimum weight triangulation* minimizes the total edge length over all possible triangulations of the same set of points and prescribed edges [11, 17].

It is known that the Delaunay triangulation maximizes the minimum angle over all triangulations of the same point set [22]. This result can be extended to a similar statement about the sorted angle vector of the Delaunay triangulation [6] and to the constrained case [13]. The Delaunay triangulation of n points in the plane can be constructed in time $O(n \log n)$ [6, 19], and even if some edges are prescribed its

* Research of the first author was supported by the National Science Foundation under grants CCR-8714565 and CCR-8921421. The second author is on study leave from the Department of Information Systems and Computer Science, National University of Singapore, Republic of Singapore.

† Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA.

Minimizing maximum angle [Edelsbrunner, Tan, Waupotitsch 1992]

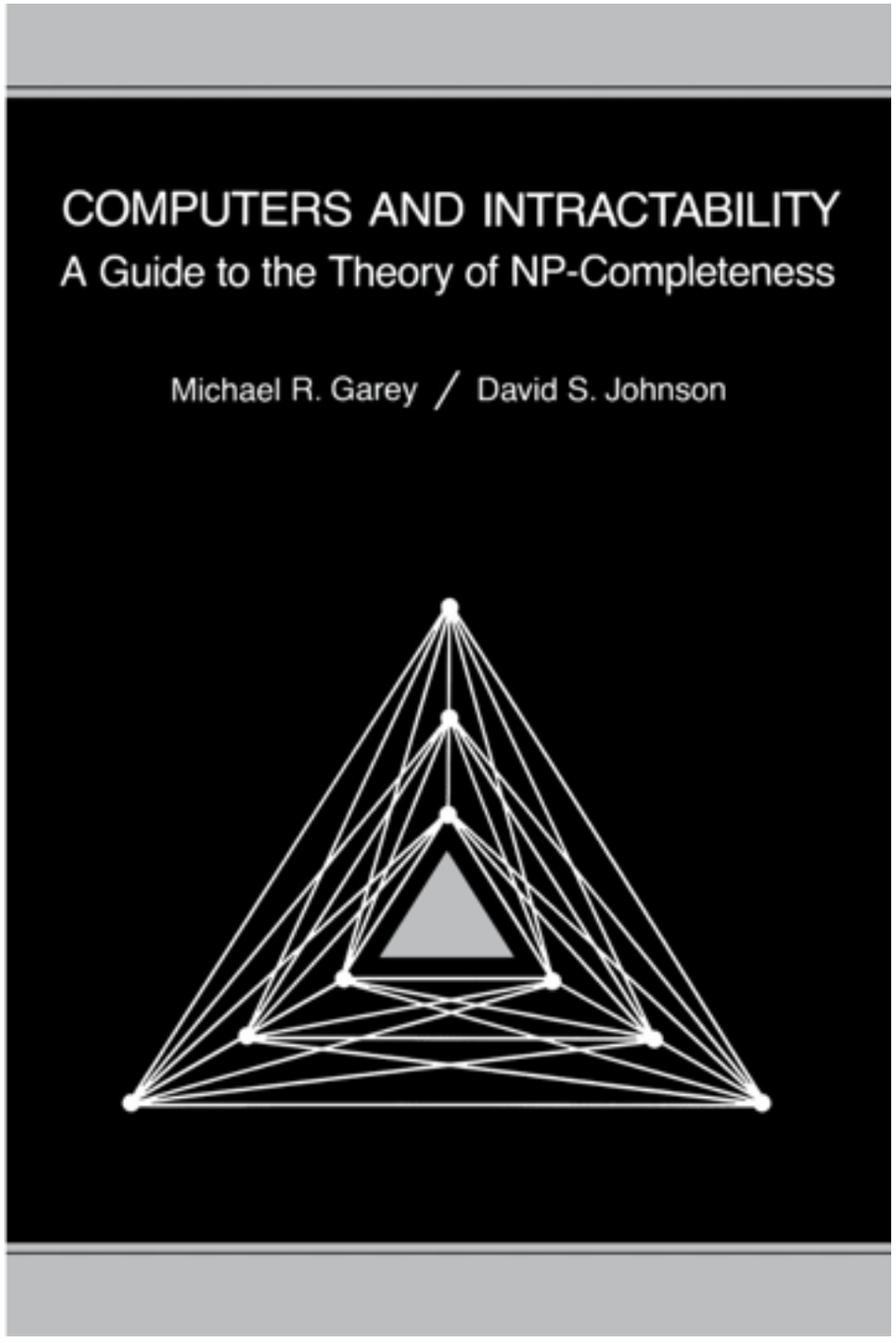


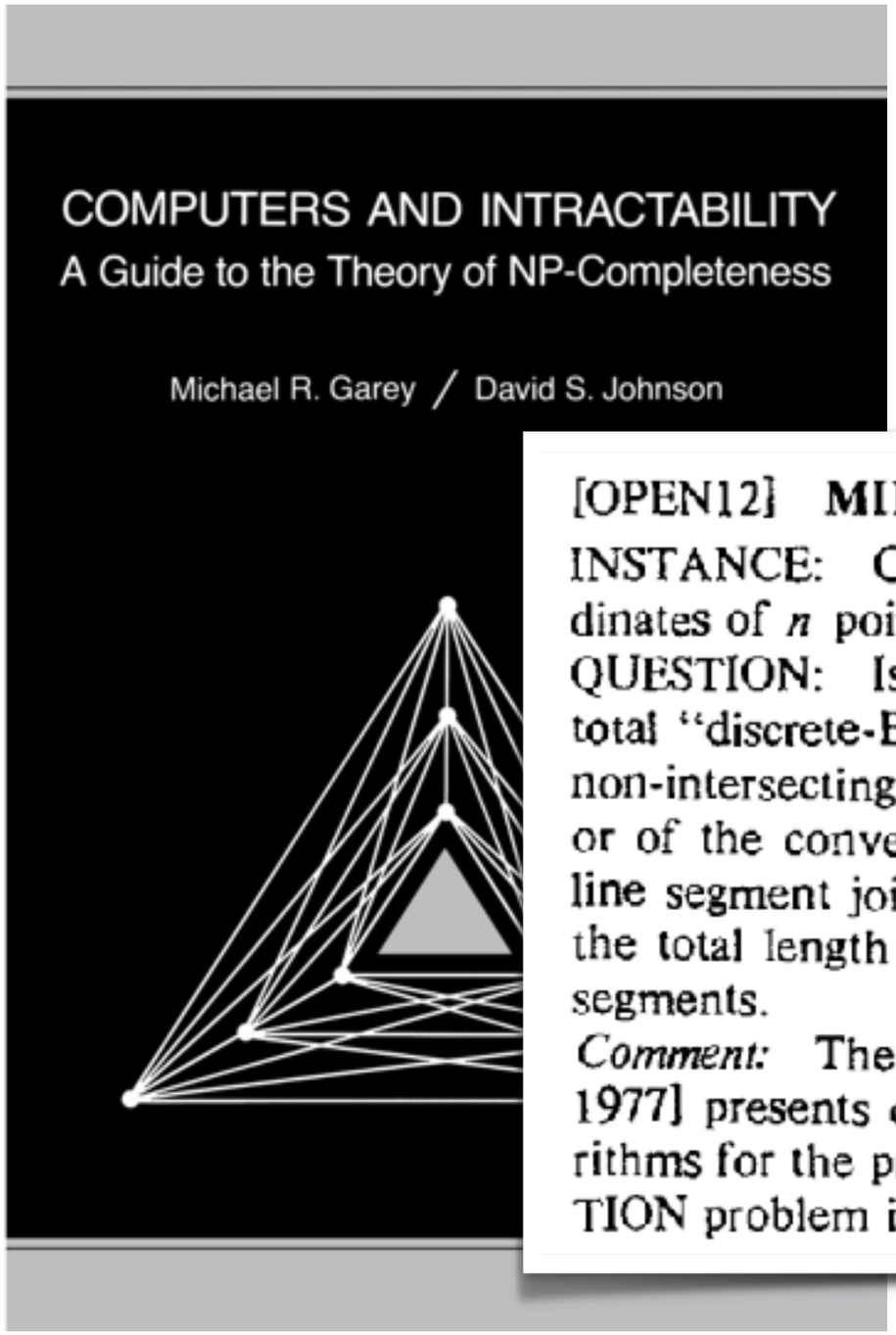
AN O($n^2 \log n$) TIME ALGORITHM FOR THE MINMAX ANGLE TRIANGULATION *

HERBERT EDELSBRUNNER†, TIOW SENG TAN† AND ROMAN WAUPOTITSCH†

Abstract. We show that a triangulation of a set of n points in the plane that minimizes the maximum angle can be computed in time $O(n^2 \log n)$ and space $O(n)$. The algorithm is fairly easy to implement and is based on the edge-insertion scheme that iteratively improves an arbitrary initial triangulation. It can be extended to the case where edges are prescribed, and, within the same time- and space-bounds, it can lexicographically minimize the sorted angle vector if the point set is in general position. Experimental results on the efficiency of the algorithm and the quality of the triangulations obtained are included.

3 Minimum-weight triangulations


COMPUTERS AND INTRACTABILITY
A Guide to the Theory of NP-Completeness
Michael R. Garey / David S. Johnson



[OPEN12] **MINIMUM LENGTH TRIANGULATION**

INSTANCE: Collection $C = \{(a_i, b_i): 1 \leqslant i \leqslant n\}$ of pairs of integers, giving the coordinates of n points in the plane, and a positive integer B.

QUESTION: Is there a triangulation of the set of points represented by C that has total "discrete-Euclidean" length B or less? Here a triangulation is a collection of non-intersecting line segments, each joining two points in C, that divides the interior of the convex hull into triangular regions. The discrete-Euclidean length of a line segment joining (a_i, b_i) and (a_j, b_j) is given by $[((a_i - a_j)^2 + (b_i - b_j)^2)^{1/2}]$, and the total length of a triangulation is the sum of the lengths of its constituent line segments.

Comment: The analogous problem for the rectilinear metric is also open. [Lloyd, 1977] presents counterexamples to a number of conjectured polynomial time algorithms for the problem and proves that the related CONSTRAINED TRIANGULATION problem is NP-complete.

MWT is NP-hard [Mulzer and Rote, 2008]

MWT is NP-hard [Mulzer and Rote, 2008]

Minimum-Weight Triangulation is NP-hard

WOLFGANG MULZER
Princeton University
and
GÜNTER ROTE
Freie Universität Berlin

A triangulation of a planar point set S is a maximal plane straight-line graph with vertex set S. In the *minimum-weight triangulation* (MWT) problem, we are looking for a triangulation of a given point set that minimizes the sum of the edge lengths. We prove that the decision version of this problem is NP-hard, using a reduction from PLANAR 1-IN-3-SAT. The correct scoring of the gadgets is established with computer assistance, using dynamic programming on polygonal faces as well as the β-skeleton heuristic to certify that certain edges belong to the minimum-weight triangulation.

Categories and Subject Descriptors: F.2.2 [**Nonnumerical Algorithms and Problems**]: Geometrical problems and computations; G.2.2 [**Graph Theory**]: Graph algorithms

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Optimal triangulations, PLANAR 1-IN-3-SAT

1. INTRODUCTION

Given a set S of points in the Euclidean plane, a *triangulation* T of S is a maximal plane straight-line graph with vertex set S. The *weight* of T is defined as the total Euclidean length of all edges in T. A triangulation that achieves the minimum weight is called a *minimum-weight triangulation* (MWT) of S.

The problem of computing a triangulation for a given planar point set arises naturally in many applications such as stock cutting, finite element analysis, terrain modeling, and numerical approximation. The *minimum-weight* triangulation has attracted the attention of many researchers, mainly due to its natural definition of optimality, and not so much because it would be important for the mentioned applications. We show that computing a minimum-weight triangulation is NP-hard. Note that it is not known whether the MWT problem is in NP, because it is an open problem whether sums of radicals (namely Euclidean distances) can be compared in polynomial time [Blömer 1991]. This problem is common to many geometric optimization problems, like, perhaps most famously, the Euclidean Traveling Salesperson Problem. To get a variant of the problem that is in NP, one can take the weight of an edge e as the rounded value $\lceil \|e\|_2 \rceil$. Our proof also shows that this variant is NP-complete.

Our proof uses a polynomial time reduction from POSITIVE PLANAR 1-IN-3-SAT, a variant of the well-known PLANAR 3-SAT problem, which is a standard tool for showing NP-hardness of geometric problems.

1.1 Optimal Triangulations

Usually, a planar point set has (exponentially) many different triangulations, and many applications call for triangulations with certain good properties (see Figure 1). Optimal triangulations under various optimality criteria were extensively surveyed

Journal of the ACM, Vol. V, No. N, Month 20YY, Pages 1–31.

MWT is NP-hard [Mulzer and Rote, 2008]

Minimum-Weight Triangulation is NP-hard

WOLFGANG MULZER
Princeton University
and
GÜNTER ROTE
Freie Universität Berlin

A triangulation of a planar point set S is a maximal plane straight-line graph with vertex set S. In the *minimum-weight triangulation* (MWT) problem, we are looking for a triangulation of a given point set that minimizes the sum of the edge lengths. We prove that the decision version of this problem is NP-hard, using a reduction from PLANAR 1-IN-3-SAT. The correct working of the gadgets is established with computer assistance, using dynamic programming on polygonal faces as well as the β-skeleton heuristic to verify that certain edges belong to the minimum-weight triangulation.

Categories and Subject Descriptors: F.2.2 [**Nonnumerical Algorithms and Problems**]: Geometrical problems and computations; G.2.2 [**Graph Theory**]: Graph algorithms

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Optimal triangulations, PLANAR 1-IN-3-SAT

1. INTRODUCTION

Given a set S of points in the Euclidean plane, a *triangulation* T of S is a maximal plane straight-line graph with vertex set S. The *weight* of T is defined as the total Euclidean length of all edges in T. A triangulation that achieves the minimum weight is called a *minimum-weight triangulation* (MWT) of S.

The problem of computing a triangulation for a given planar point set arises naturally in many applications such as stock cutting, finite element analysis, terrain modeling, and numerical approximation. The *minimum-weight* triangulation has attracted the attention of many researchers, mainly due to its natural definition of optimality, and not so much because it would be important for the mentioned applications. We show that computing a minimum-weight triangulation is NP-hard. Note that it is not known whether the MWT problem is in NP, because it is an open problem whether sums of radicals (namely Euclidean distances) can be compared in polynomial time [Blömer 1991]. This problem is common to many geometric optimization problems, like, perhaps most famously, the Euclidean Traveling Salesperson Problem. To get a variant of the problem that is in NP, one can take the weight of an edge e as the rounded value $\lceil \|e\|_2 \rceil$. Our proof also shows that this variant is NP-complete.

Our proof uses a polynomial time reduction from POSITIVE PLANAR 1-IN-3-SAT, a variant of the well-known PLANAR 3-SAT problem, which is a standard tool for showing NP-hardness of geometric problems.

1.1 Optimal Triangulations

Usually, a planar point set has (exponentially) many different triangulations, and many applications call for triangulations with certain good properties (see Figure 1). Optimal triangulations under various optimality criteria were extensively surveyed

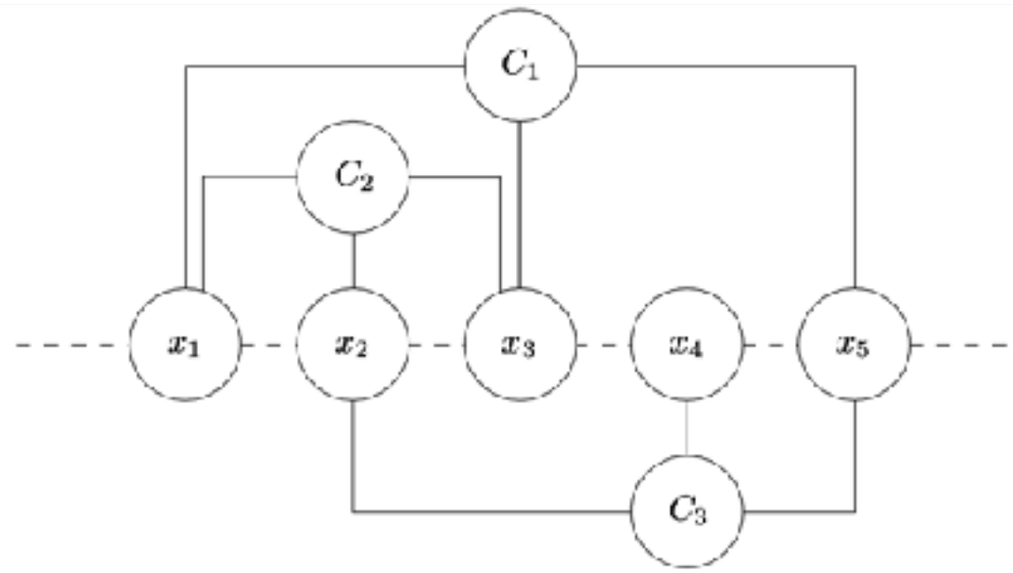


Fig. 5. A rectilinear embedding of graph that is associated with the Boolean formula $(x_1 \vee \neg x_3 \vee x_5) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee \neg x_5)$.

MWT is NP-hard [Mulzer and Rote, 2008]

Minimum-Weight Triangulation is NP-hard

WOLFGANG MULZER
Princeton University
and
GÜNTER ROTE
Freie Universität Berlin

A triangulation of a planar point set S is a maximal plane straight-line graph with vertex set S. In the *minimum-weight triangulation* (MWT) problem, we are looking for a triangulation of a given point set that minimizes the sum of the edge lengths. We prove that the decision version of this problem is NP-hard, using a reduction from PLANAR 1-IN-3-SAT. The correct working of the gadgets is established with computer assistance, using dynamic programming on polygonal faces as well as the β-skeleton heuristic to certify that certain edges belong to the minimum-weight triangulation.

Categories and Subject Descriptors: F.2.2 [**Nonnumerical Algorithms and Problems**]: Geometrical problems and computations; G.2.2 [**Graph Theory**]: Graph algorithms

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Optimal triangulations, PLANAR 1-IN-3-SAT

1. INTRODUCTION

Given a set S of points in the Euclidean plane, a *triangulation* T of S is a maximal plane straight-line graph with vertex set S. The *weight* of T is defined as the total Euclidean length of all edges in T. A triangulation that achieves the minimum weight is called a *minimum-weight triangulation* (*MWT*) of S.

The problem of computing a triangulation for a given planar point set arises naturally in many applications such as stock cutting, finite element analysis, terrain modeling, and numerical approximation. The *minimum-weight* triangulation has attracted the attention of many researchers, mainly due to its natural definition of optimality, and not so much because it would be important for the mentioned applications. We show that computing a minimum-weight triangulation is NP-hard. Note that it is not known whether the MWT problem is in NP, because it is an open problem whether sums of radicals (namely, Euclidean distances) can be compared in polynomial time [Blömer 1991]. This problem is common to many geometric optimization problems, like, perhaps most famously, the Euclidean Traveling Salesperson Problem. To get a variant of the problem that is in NP, one can take the weight of an edge e as the rounded value $\lfloor \|e\|_2 \rfloor$. Our proof also shows that this variant is NP-complete.

Our proof uses a polynomial time reduction from POSITIVE PLANAR 1-IN-3-SAT, a variant of the well-known PLANAR 3-SAT problem, which is a standard tool for showing NP-hardness of geometric problems.

1.1 Optimal Triangulations

Usually, a planar point set has (exponentially) many different triangulations, and many applications call for triangulations with certain good properties (see Figure 1). Optimal triangulations under various optimality criteria were extensively surveyed

Journal of the ACM, Vol. V, No. N, Month 20YY, Pages 1–31.

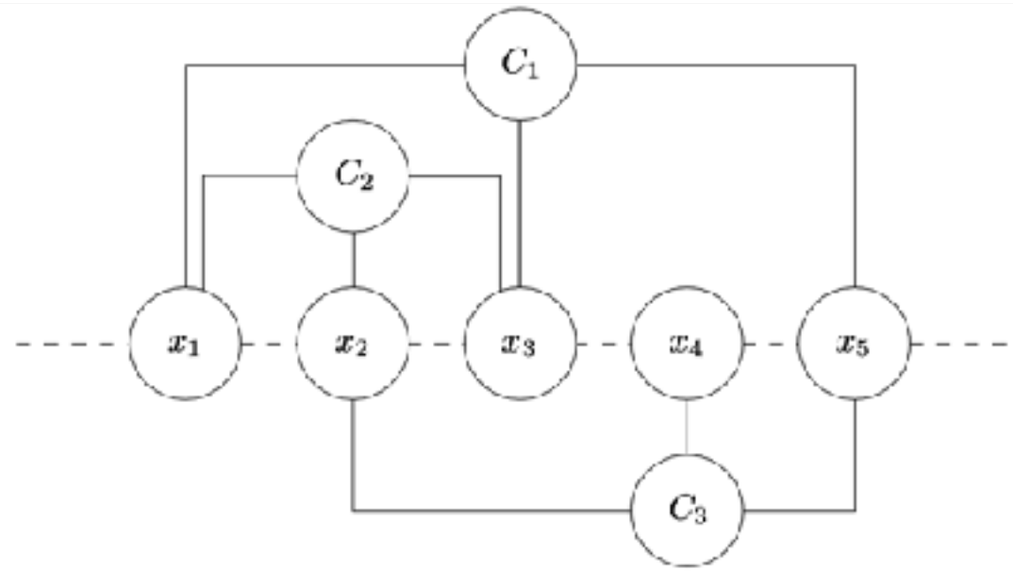
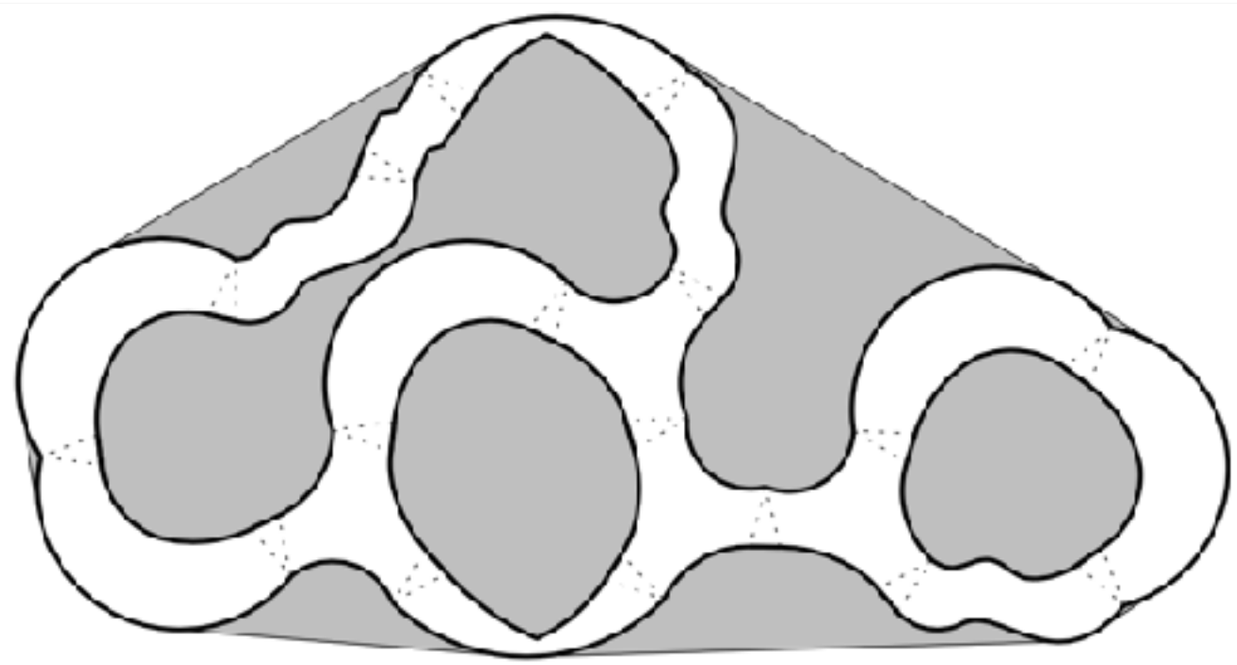


Fig. 5. A rectilinear embedding of graph that is associated with the Boolean formula $(x_1 \vee \neg x_3 \vee x_5) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee \neg x_5)$.



MWT is NP-hard [Mulzer and Rote, 2008]

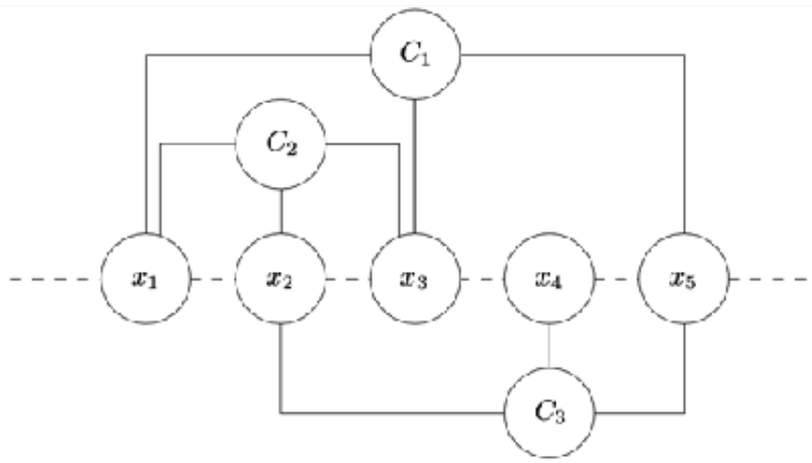


Fig. 5. A rectilinear embedding of graph that is associated with the Boolean formula $(x_1 \vee \neg x_3 \vee x_5) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee \neg x_5)$.

MWT is NP-hard [Mulzer and Rote, 2008]

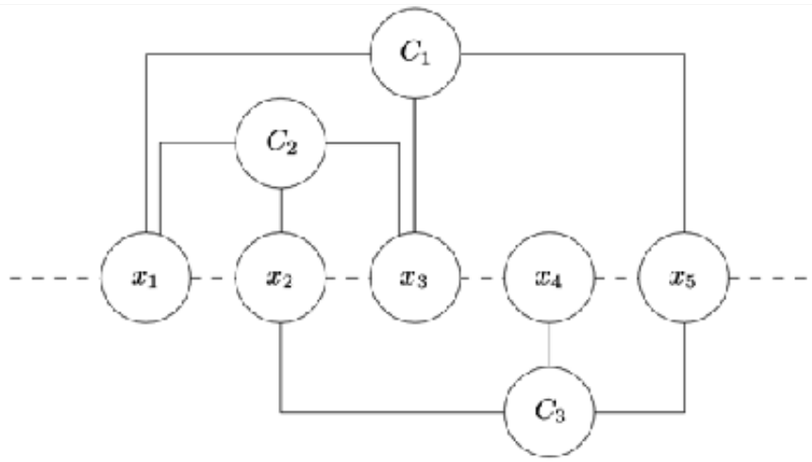
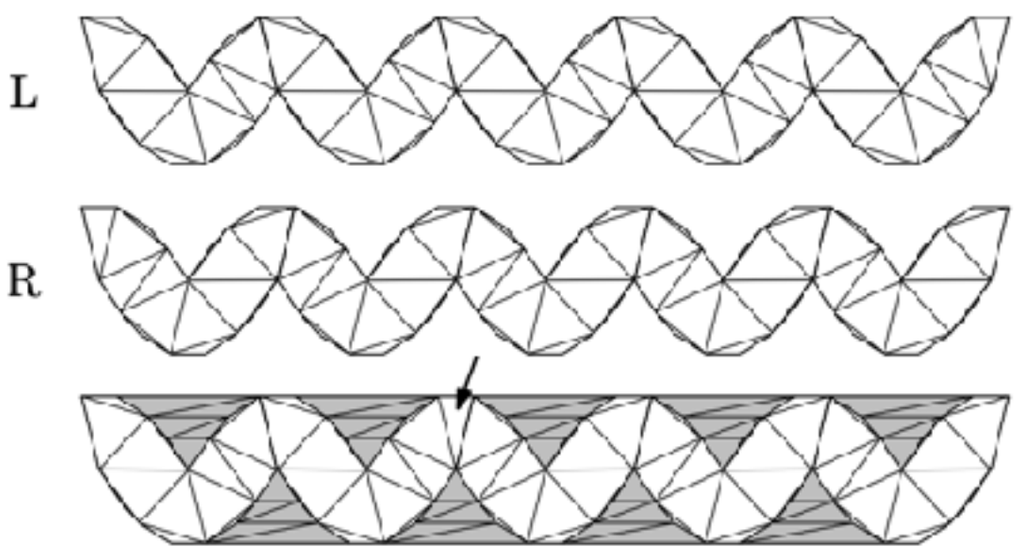


Fig. 5. A rectilinear embedding of graph that is associated with the Boolean formula $(x_1 \vee \neg x_3 \vee x_5) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee \neg x_5)$.



MWT is NP-hard [Mulzer and Rote, 2008]

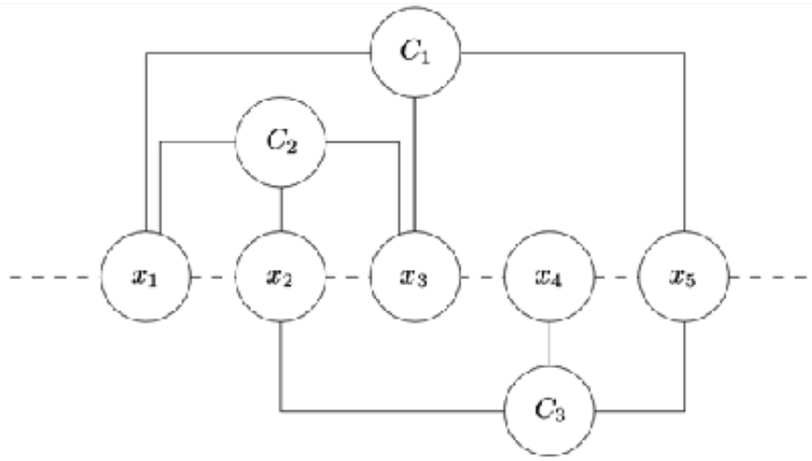
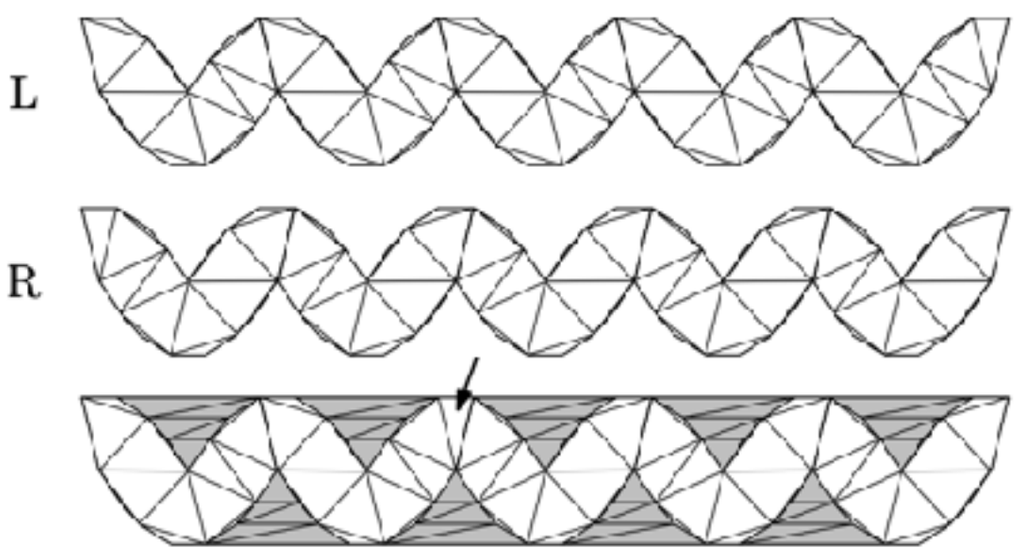


Fig. 5. A rectilinear embedding of graph that is associated with the Boolean formula $(x_1 \vee \neg x_3 \vee x_5) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee \neg x_5)$.



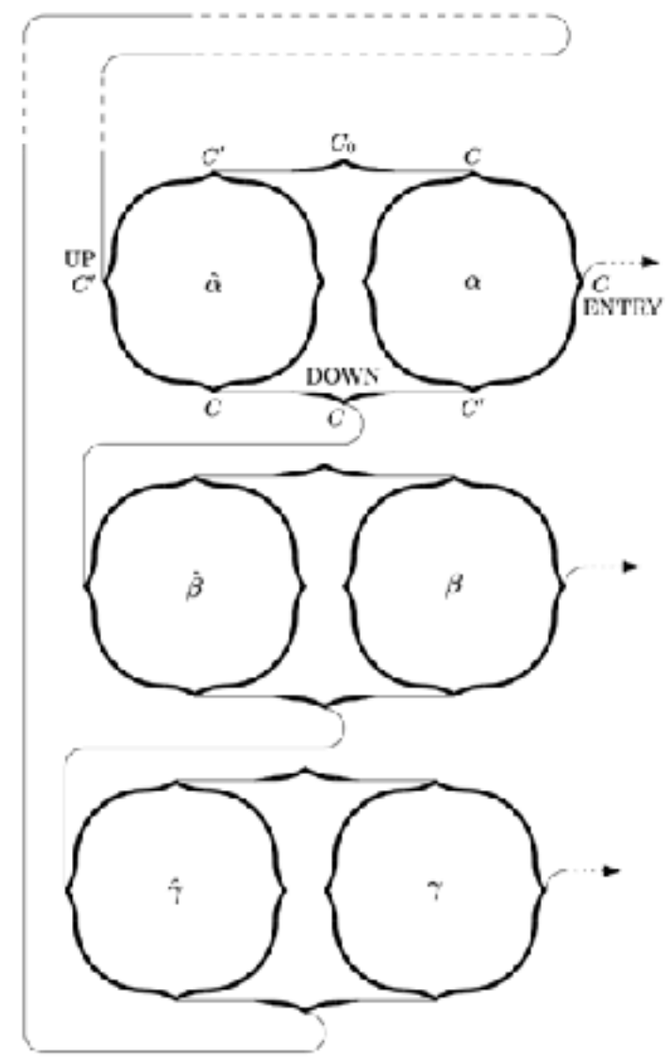
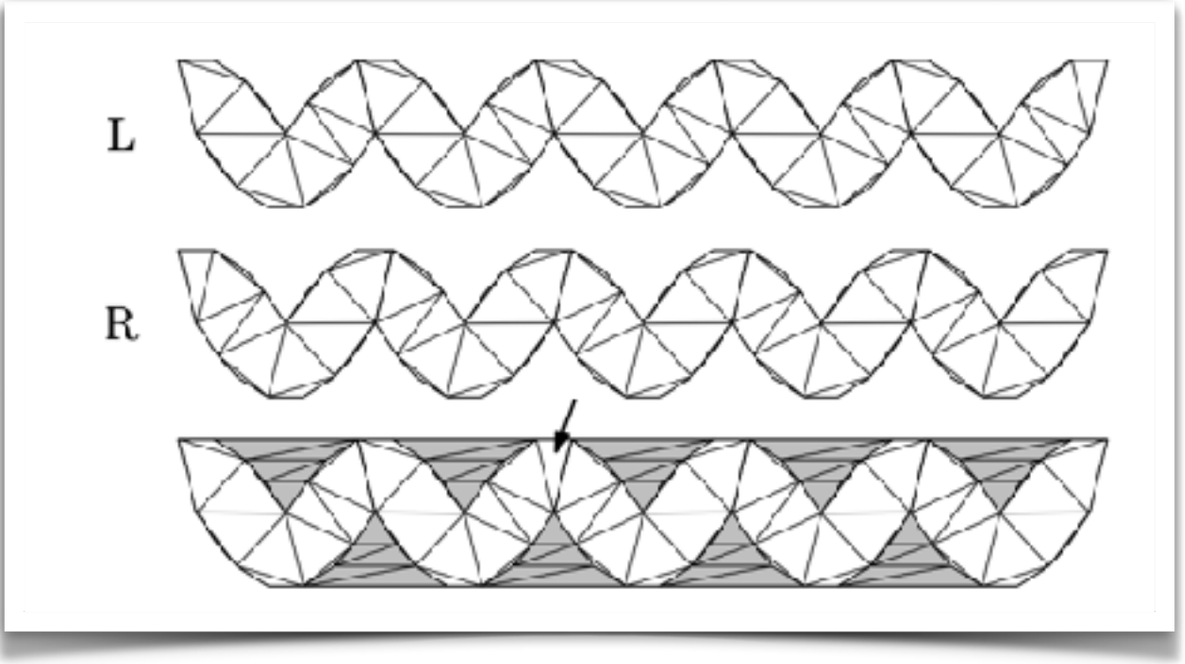
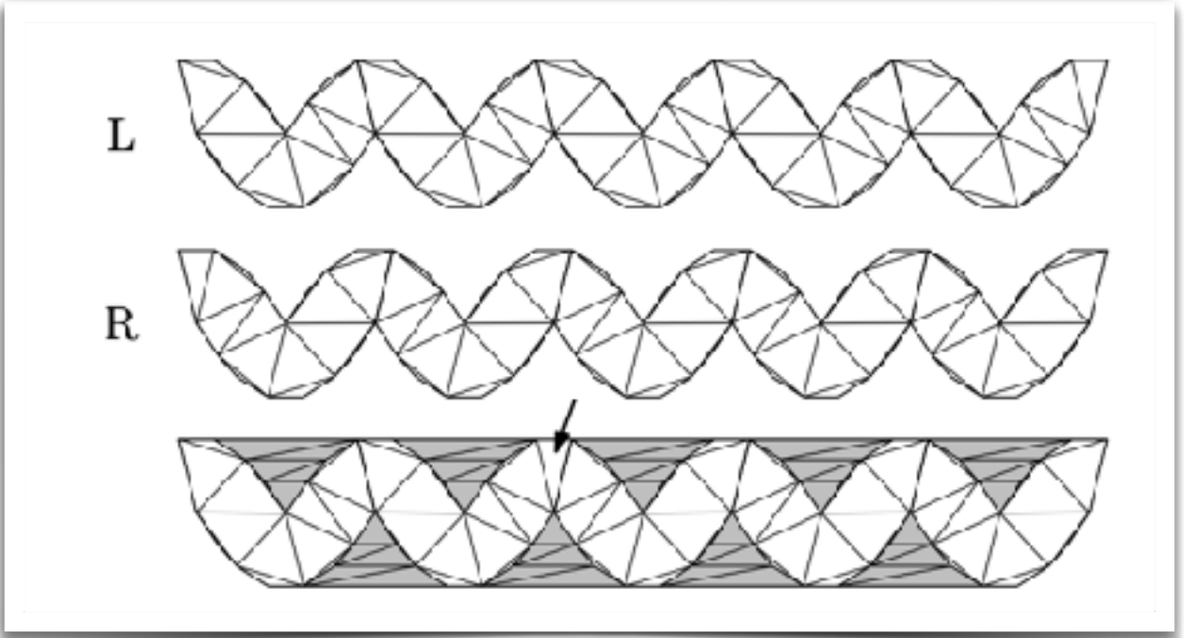


Fig. 19. Schematic view of the clause gadget. The dashed parts of the wires are long enough for sufficiently many copies of the wire-piece and the extended wire-piece to ensure that the wire can reach from the lowest C connection to the left C' connection of $\hat{\alpha}$ (Lemma 6.1).

MWT is NP-hard [Mulzer and Rote, 2008]



MWT is NP-hard [Mulzer and Rote, 2008]



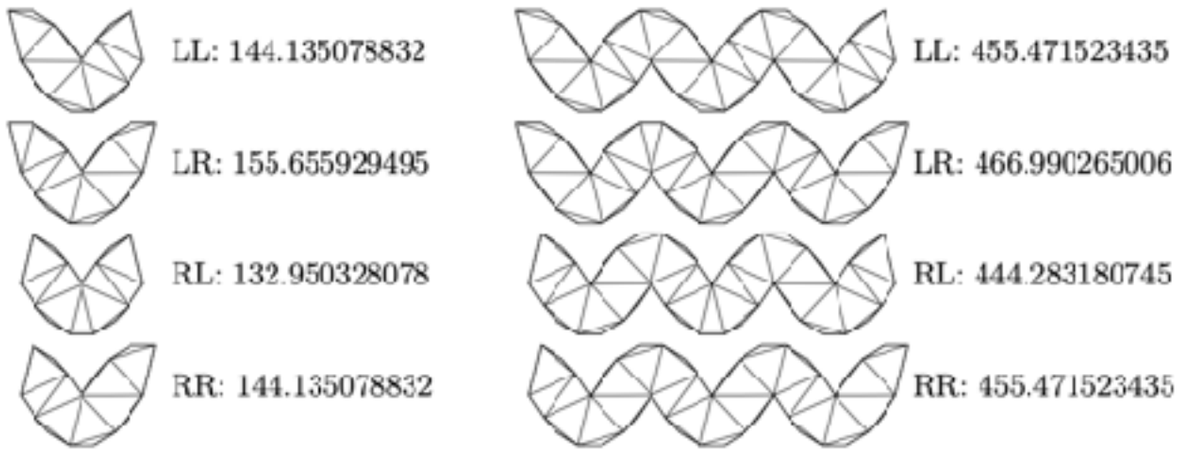
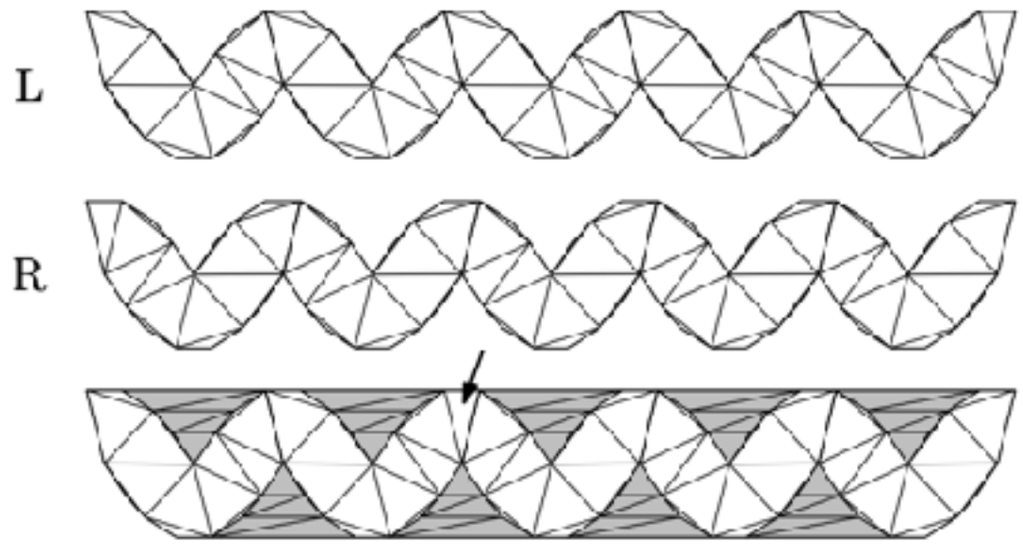


Fig. 17. Optimal solutions for all cases for the wire-piece and the extended wire-piece

MWT is NP-hard [Mulzer and Rote, 2008]



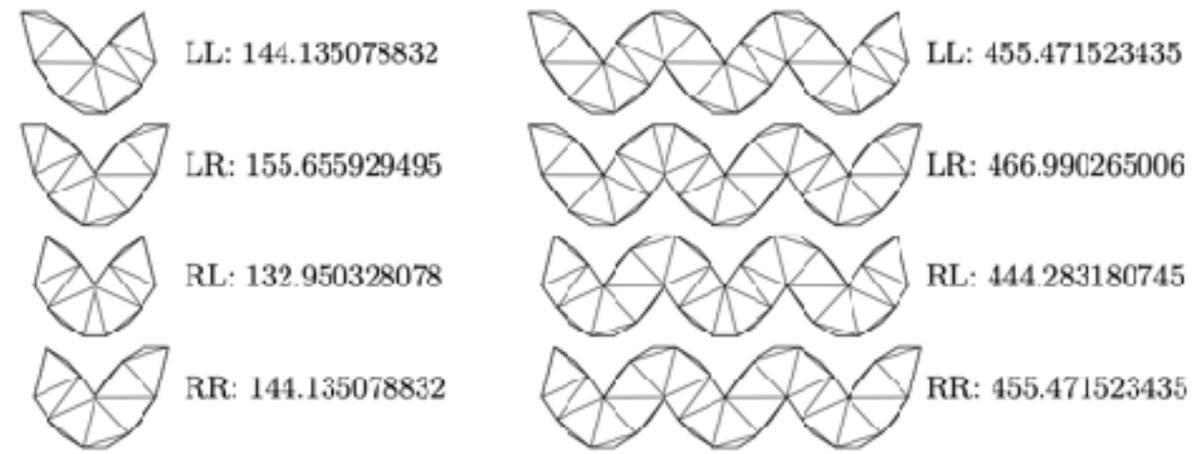
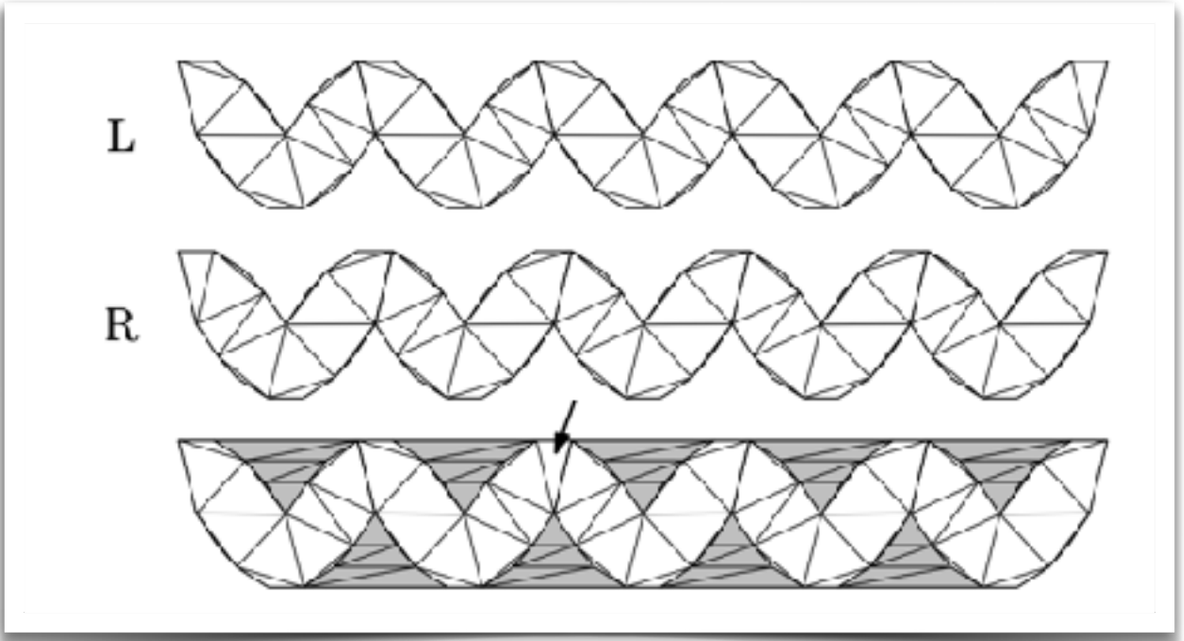


Fig. 17. Optimal solutions for all cases for the wire-piece and the extended wire-piece

pattern	multiplicity	internal cost c	reduced internal cost $\bar{c}$	relative reduced cost $\hat{c}$
LL	1	455.471 523 435	455.471 523 435	0.000 000 000
LR	2	466.990 265 006	455.679 921 006	0.208 397 570
RL	1	444.283 180 745	455.593 524 745	0.122 001 310
RR	1	455.471 523 435	455.471 523 435	0.000 000 000

Table III. Analysis of the extended wire-piece

MWT is NP-hard [Mulzer and Rote, 2008]



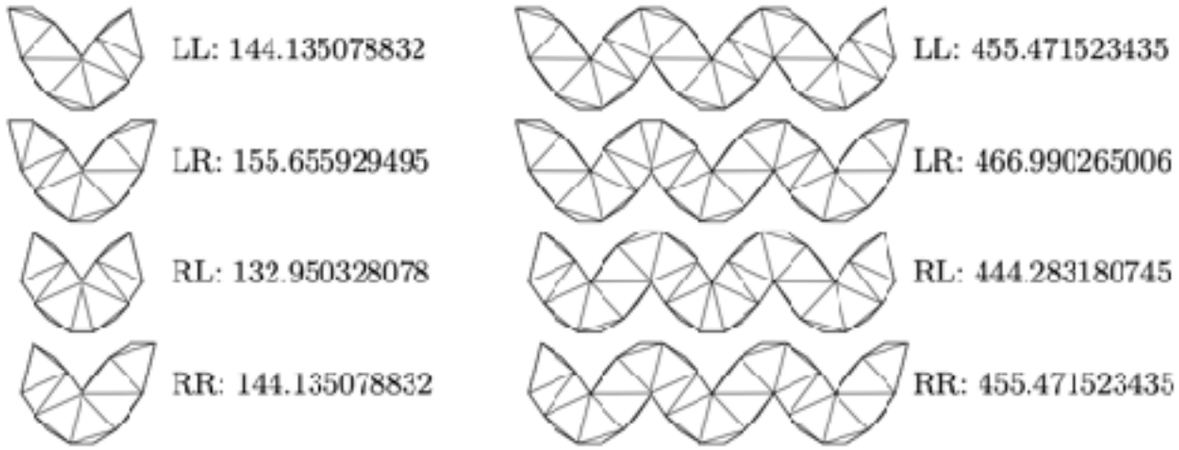


Fig. 17. Optimal solutions for all cases for the wire-piece and the extended wire-piece

pattern	multiplicity	internal cost c	reduced internal cost $\bar{c}$	relative reduced cost $\hat{c}$
LL	1	455.471 523 435	455.471 523 435	0.000 000 000
LR	2	466.990 265 006	455.679 921 006	0.208 397 570
RL	1	444.283 180 745	455.593 524 745	0.122 001 310
RR	1	455.471 523 435	455.471 523 435	0.000 000 000

Table III. Analysis of the extended wire-piece

LEMMA 6.1. *Consider two small vertical terminal triangles at the same height with a horizontal distance $d > 230\,000$ that is a multiple of 0.01. Then the two triangles can be connected by a sequence of wire-pieces and extended wire-pieces.*

An analogous statement holds for vertical connections.

PROOF. If the distance between two terminal triangles is $d = 0.01 \cdot z$ for some integer $z \geq 3 \cdot 2740 \cdot 2739 = 22\,514\,580$, they can be connected by concatenating $y := z \bmod 2740$ extended wire-pieces with $\lfloor z/2740 \rfloor - 3y$ wire-pieces. □

MWT is NP-hard [Mulzer and Rote, 2008]

MWT is NP-hard [Mulzer and Rote, 2008]

	wire-piece	extended wire-piece	thickening adapter	thinning adapter
LL	0.000 000 000	0.000 000 000	0.000 051 402 . . . $= \varepsilon_1$	0.000 000 000
LR	0.210 506 663	0.208 397 570	0.018 887 246	0.018 887 246
RL	0.125 593 246	0.122 001 310	0.014 627 250	0.014 627 250
RR	0.000 000 000	0.000 000 000	0.000 000 000	0.000 051 402 . . . $= \varepsilon_1$

	C_0 connection	left bend	right bend	thick left bend
LL	0.000 000 000	0.000 000 000	0.000 000 000	0.000 000 000
LR	0.044 001 701	0.086 460 895	0.210 506 663	0.891 261 046
RL	0.020 571 757	0.125 593 246	0.125 593 246	0.020 571 757
RR	0.000 000 000	0.000 000 000	0.000 000 000	0.000 000 000

Table IV. The relative reduced costs $\tilde{c}$ for the pieces with two terminals

C		relative reduced cost	C'	
LLl		0.003 861 076 . . . $= \delta_1$		RRr
LLr		0.000 001 575 . . . $= \varepsilon_2$		RRl
LRl		0.029 994 716		LRr
LRr		0.026 135 215		LRl
RLl		0.020 571 757		RLr
RLr		0.020 573 332		RLl
RRl		0.000 000 000 = 0		LLr
RRr		0.004 630 878 . . . $= \delta_2$		LLl

MWT is NP-hard [Mulzer and Rote, 2008]

	wire-piece	extended wire-piece	thickening adapter	thinning adapter
LL	0.000 000 000	0.000 000 000	0.000 051 402 . . . $= \varepsilon_1$	0.000 000 000
LR	0.210 506 663	0.208 397 570	0.018 887 246	0.018 887 246
RL	0.125 593 246	0.122 001 310	0.014 627 250	0.014 627 250
RR	0.000 000 000	0.000 000 000	0.000 000 000	0.000 051 402 . . . $= \varepsilon_1$

	C_0 connection	left bend	right bend	thick left bend
LL	0.000 000 000	0.000 000 000	0.000 000 000	0.000 000 000
LR	0.044 001 701	0.086 460 895	0.210 506 663	0.891 261 046
RL	0.020 571 757	0.125 593 246	0.125 593 246	0.020 571 757
RR	0.000 000 000	0.000 000 000	0.000 000 000	0.000 000 000

Table IV. The relative reduced costs $\tilde{c}$ for the pieces with two terminals

	C	relative reduced cost	C'	
LLl		0.003 861 076 . . . $= \delta_1$		RRr
LLr		0.000 001 575 . . . $= \varepsilon_2$		RRl
LRl		0.029 994 716		LRr
LRr		0.026 135 215		LRl
RLl		0.020 571 757		RLr
RLr		0.020 573 332		RLl
RRl		0.000 000 000 = 0		LLr
RRr		0.004 630 878 . . . $= \delta_2$		LLl

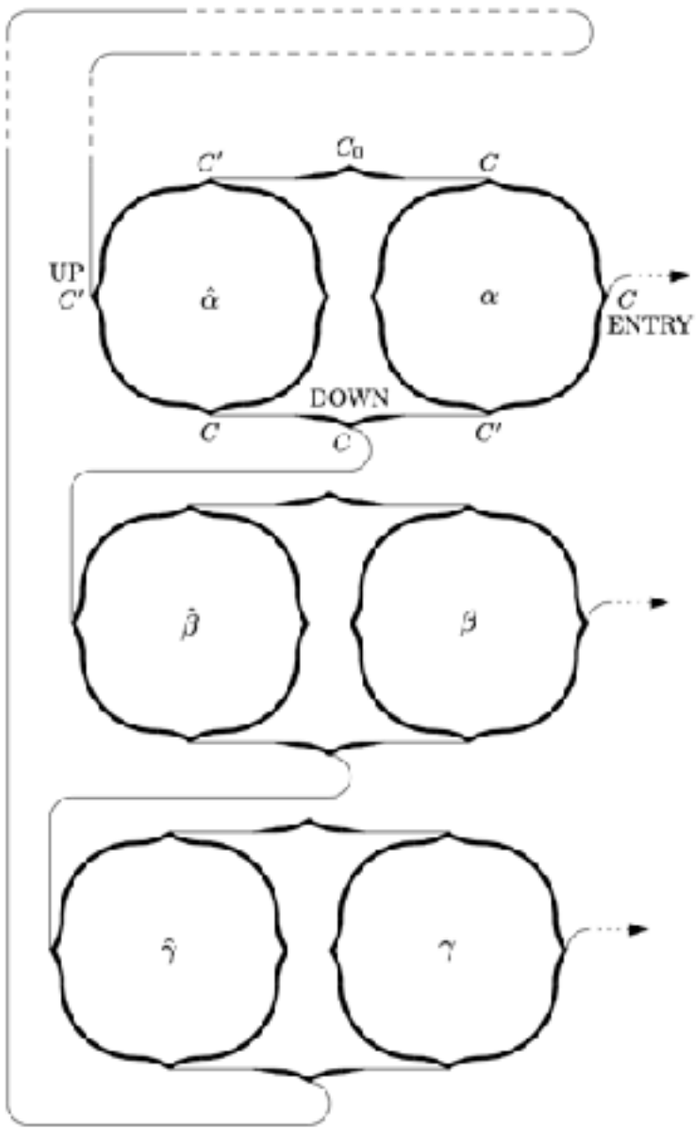


Fig. 19. Schematic view of the clause gadget. The dashed parts of the wires are long enough for sufficiently many copies of the wire-piece and the extended wire-piece to ensure that the wire can reach from the lowest C connection to the left C' connection of $\hat{\alpha}$ (Lemma 6.1).

MWT is practically solvable to optimality [Haas, 2018]

MWT is practically solvable to optimality [Haas, 2018]

Solving Large-Scale Minimum-Weight Triangulation Instances to Provable Optimality

Andreas Haas[1]

1 Department of Computer Science, TU Braunschweig, 38106 Braunschweig, Germany. haas@ibr.cs.tu-bs.de

Abstract

We consider practical methods for the problem of finding a minimum-weight triangulation (MWT) of a planar point set, a classic problem of computational geometry with many applications. While Mulzer and Rote proved in 2008 that computing an MWT is NP-hard, Beirouti and Snoeyink showed in 1998 that computing provably optimal solutions for MWT instances of up to 80,000 uniformly distributed points is possible, making use of clever heuristics that are based on geometric insights. We show that these techniques can be refined and extended to instances of much bigger size and different type, based on an array of modifications and parallelizations in combination with more efficient geometric encodings and data structures. As a result, we are able to solve MWT instances with up to 30,000,000 uniformly distributed points in less than 4 minutes to provable optimality. Moreover, we can compute optimal solutions for a vast array of other benchmark instances that are not uniformly distributed, including normally distributed instances (up to 30,000,000 points), all point sets in the TSPLIB (up to 85,900 points), and VLSI instances with up to 744,710 points. This demonstrates that from a practical point of view, MWT instances can be handled quite well, despite their theoretical difficulty.

Digital Object Identifier 10.4230/LIPIcs.SoCG.2018.26

1 Introduction

Triangulating a set of points in the plane is a classic problem in computational geometry: given a planar point set S, find a maximal set of non-crossing line segments connecting the points in S. Triangulations have many real-world applications, for example in terrain modeling, finite element mesh generation and visualization. In general, a point set has exponentially many possible triangulations and a natural question is to ask for a triangulation that is optimal with respect to some optimality criterion. Well known and commonly used is the Delaunay triangulation, which optimizes several criteria at the same time: it maximizes the minimum angle and minimizes both the maximum circumcircle and the maximum smallest enclosing circle of all triangles. Another natural optimality criterion, and the one we are considering in this paper is minimizing the total weight of the resulting triangulation.

The minimum weight triangulation (MWT) is listed as one of the open problems in the famous book from 1979 by Garey and Johnson on NP-completeness [10]. The complexity status remained open for 27 years until Mulzer and Rote [15] finally resolved the question and showed NP-hardness of the MWT problem.

Independently, Gilbert [11] and Klincsek [17] showed that, when restricting it to simple polygons, the MWT problem can be solved in $O(n^3)$-time with dynamic programming. The dynamic programming approach can be generalized to polygons with k inner points. Hoffmann and Okamoto [16] showed how to obtain the MWT of such a point set in $O(6^k n^3 \log n)$-time. Their algorithm is based on a polygon decomposition through x-monotone paths. Grantson et al. [12] improved the algorithm to $O(n^4 4^k k)$ and showed another decomposition strategy based on cutting out triangles [13] which runs in $O(n^3 3^k k)$-time.

© Andreas Haas;
licensed under Creative Commons License CC-BY
34th Symposium on Computational Geometry (SoCG 2018).
Editors: Bettina Speckmann and Csaba Tóth; Article No. 26; pp. 26:1–26:27
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

MWT is practically solvable to optimality [Haas, 2018]

more efficient geometric encodings and data structures. As a result, we are able to solve MWT instances with up to 30,000,000 uniformly distributed points in less than 4 minutes to provable optimality. Moreover, we can compute optimal solutions for a vast array of other benchmark instances that are *not* uniformly distributed, including normally distributed instances (up to 30,000,000 points), all point sets in the TSPLIB (up to 85,900 points), and VLSI instances with up to 744,710 points. This demonstrates that from a practical point of view, MWT instances can be handled quite well, despite their theoretical difficulty.

1 Introduction

Triangulating a set of points in the plane is a classic problem in computational geometry: given a planar point set S, find a maximal set of non-crossing line segments connecting the points in S. Triangulations have many real-world applications, for example in terrain modeling, finite element mesh generation and visualization. In general, a point set has exponentially many possible triangulations and a natural question is to ask for a triangulation that is optimal with respect to some optimality criterion. Well known and commonly used is the Delaunay triangulation, which optimizes several criteria at the same time: it maximizes the minimum angle and minimizes both the maximum circumcircle and the maximum smallest enclosing circle of all triangles. Another natural optimality criterion, and the one we are considering in this paper is minimizing the total weight of the resulting triangulation.

The minimum weight triangulation (MWT) is listed as one of the open problems in the famous book from 1979 by Garey and Johnson on NP-completeness [10]. The complexity status remained open for 27 years until Mulzer and Rote [18] finally resolved the question and showed NP-hardness of the MWT problem.

Independently, Gilbert [11] and Klincsek [17] showed that, when restricting it to simple polygons, the MWT problem can be solved in $O(n^3)$-time with dynamic programming. The dynamic programming approach can be generalized to polygons with k inner points. Hoffmann and Okamoto [16] showed how to obtain the MWT of such a point set in $O(6^k n^3 \log n)$-time. Their algorithm is based on a polygon decomposition through x-monotone paths. Grantson et al. [12] improved the algorithm to $O(n^4 4^k k)$ and showed another decomposition strategy based on cutting out triangles [13] which runs in $O(n^3 4^k k)$-time.

© Andreas Haas,
licensed under Creative Commons License CC-BY
34th Symposium on Computational Geometry (SoCG 2018)
Editors: Bettina Speckmann and Csaba Tóth; Article No. 29; pp. 29:1–29:27
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

MWT is practically solvable to optimality [Haas, 2018]

Solving Large-Scale Minimum-Weight Triangulation Instances to Provable Optimality

Andreas Haas[1]

1 Department of Computer Science, TU Braunschweig, 38106 Braunschweig, Germany. haas@ibr.cs.tu-bs.de

Abstract

We consider practical methods for the problem of finding a minimum-weight triangulation (MWT) of a planar point set, a classic problem of computational geometry with many applications. While Mulzer and Rote proved in 2008 that computing an MWT is NP-hard, Beirouti and Snoeyink showed in 1998 that computing provably optimal solutions for MWT instances of up to 80,000 uniformly distributed points is possible, making use of clever heuristics that are based on geometric insights. We show that these techniques can be refined and extended to instances of much bigger size and different type, based on an array of modifications and parallelizations in combination with more efficient geometric encodings and data structures. As a result, we are able to solve MWT instances with up to 30,000,000 uniformly distributed points in less than 4 minutes to provable optimality. Moreover, we can compute optimal solutions for a vast array of other benchmark instances that are not uniformly distributed, including normally distributed instances (up to 30,000,000 points), all point sets in the TSPLIB (up to 85,900 points), and VLSI instances with up to 744,710 points. This demonstrates that from a practical point of view, MWT instances can be handled quite well, despite their theoretical difficulty.

Digital Object Identifier 10.4230/LIPIcs.SoCG.2018.26

1 Introduction

Triangulating a set of points in the plane is a classic problem in computational geometry: given a planar point set S, find a maximal set of non-crossing line segments connecting the points in S. Triangulations have many real-world applications, for example in terrain modeling, finite element mesh generation and visualization. In general, a point set has exponentially many possible triangulations and a natural question is to ask for a triangulation that is optimal with respect to some optimality criterion. Well known and commonly used is the Delaunay triangulation, which optimizes several criteria at the same time: it maximizes the minimum angle and minimizes both the maximum circumcircle and the maximum smallest enclosing circle of all triangles. Another natural optimality criterion, and the one we are considering in this paper is minimizing the total weight of the resulting triangulation.

The minimum weight triangulation (MWT) is listed as one of the open problems in the famous book from 1979 by Garey and Johnson on NP-completeness [10]. The complexity status remained open for 27 years until Mulzer and Rote [15] finally resolved the question and showed NP-hardness of the MWT problem.

Independently, Gilbert [11] and Klincsek [17] showed that, when restricting it to simple polygons, the MWT problem can be solved in $O(n^3)$-time with dynamic programming. The dynamic programming approach can be generalized to polygons with k inner points. Hoffmann and Okamoto [16] showed how to obtain the MWT of such a point set in $O(6^k n^3 \log n)$-time. Their algorithm is based on a polygon decomposition through x-monotone paths. Grantson et al. [12] improved the algorithm to $O(n^4 4^k k)$ and showed another decomposition strategy based on cutting out triangles [13] which runs in $O(n^3 3^k k)$-time.

© Andreas Haas;
licensed under Creative Commons License CC-BY
34th Symposium on Computational Geometry (SoCG 2018).
Editors: Bettina Speckmann and Csaba Tóth; Article No. 26; pp. 26:1–26:27
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

MWT is practically solvable to optimality [Haas, 2018]

Solving Large-Scale Minimum-Weight Triangulation Instances to Provable Optimality

Andreas Haas[1]

1 Department of Computer Science, TU Braunschweig, 38106 Braunschweig, Germany. haas@ifor.cs.tu-bs.de

Abstract

We consider practical methods for the problem of finding a minimum-weight triangulation (MWT) of a planar point set, a classic problem of computational geometry with many applications. While Mulzer and Rote proved in 2006 that computing an MWT is NP-hard, Beirouti and Snoeyink showed in 1998 that computing provably optimal solutions for MWT instances of up to 80,000 uniformly distributed points is possible, making used of clever heuristics that are based on geometric insights. We show that these techniques can be refined and extended to instances of much bigger size and different type, based on an array of modifications and parallelizations in combination with more efficient geometric encodings and data structures. As a result, we are able to solve MWT instances with up to 10,000,000 uniformly distributed points in less than 4 minutes to provable optimality. Moreover, we can compute optimal solutions for a vast array of other benchmark instances that are not uniformly distributed, including normally distributed instances (up to 30,000,000 points), all point sets in the TSPLIB (up to 85,900 points), and VLSI instances with up to 744,710 points. This demonstrates that from a practical point of view, MWT instances can be handled quite well, despite their theoretical difficulty.

Digital Object Identifier 10.4230/LIPIcs.SoCG.2018.26

1 Introduction

Triangulating a set of points in the plane is a classic problem in computational geometry: given a planar point set S, find a maximal set of non-crossing line segments connecting the points in S. Triangulations have many real-world applications, for example in terrain modeling, finite element mesh generation and visualization. In general, a point set has exponentially many possible triangulations and a natural question is to ask for a triangulation that is optimal with respect to some optimality criterion. Well known and commonly used is the Delaunay triangulation, which optimizes several criteria at the same time: it maximizes the minimum angle and minimizes both the maximum circumcircle and the maximum smallest enclosing circle of all triangles. Another natural optimality criterion, and the one we are considering in this paper is minimizing the total weight of the resulting triangulation.

The minimum weight triangulation (MWT) is listed as one of the open problems in the famous book from 1979 by Garey and Johnson on NP-completeness [10]. The complexity status remained open for 27 years until Mulzer and Rote [15] finally resolved the question and showed NP-hardness of the MWT problem.

Independently, Gilbert [11] and Klincsek [17] showed that, when restricting it to simple polygons, the MWT problem can be solved in $O(n^3)$-time with dynamic programming. The dynamic programming approach can be generalized to polygons with k inner points. Heffernan and Okamoto [16] showed how to obtain the MWT of such a point set in $O(6^k n^3 \log n)$-time. Their algorithm is based on a polygon decomposition through x-monotone paths. Grantson et al. [12] improved the algorithm to $O(n^4 4^k k)$ and showed another decomposition strategy based on cutting out triangles [13] which runs in $O(n^3 3^k k)$-time.

© Andreas Haas;
licensed under Creative Commons License CC-BY
34th Symposium on Computational Geometry (SoCG 2018).
Editors: Bettina Speckmann and Csaba Tóth; Article No. 26; pp. 26:1–26:27
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

3.1 Diamond property

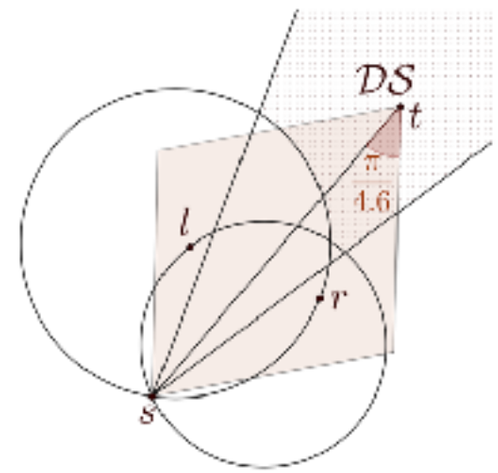


Figure 1 Points l and r induce a region DS such that all edges $e = st$ with $t \in DS$ fail the diamond test. DS is called a dead sector (dotted area).

MWT is practically solvable to optimality [Haas, 2018]

Solving Large-Scale Minimum-Weight Triangulation Instances to Provable Optimality

Andreas Haas[1]

1 Department of Computer Science, TU Braunschweig, 38106 Braunschweig, Germany. haas@ibr.cs.tu-bs.de

Abstract

We consider practical methods for the problem of finding a minimum-weight triangulation (MWT) of a planar point set, a classic problem of computational geometry with many applications. While Mulzer and Rote proved in 2006 that computing an MWT is NP-hard, Beirouti and Snoeyink showed in 1998 that computing provably optimal solutions for MWT instances of up to 80,000 uniformly distributed points is possible, making used of clever heuristics that are based on geometric insights. We show that these techniques can be refined and extended to instances of much bigger size and different type, based on an array of modifications and parallelizations in combination with more efficient geometric encodings and data structures. As a result, we are able to solve MWT instances with up to 10,000,000 uniformly distributed points in less than 4 minutes to provable optimality. Moreover, we can compute optimal solutions for a vast array of other benchmark instances that are not uniformly distributed, including normally distributed instances (up to 30,000,000 points), all point sets in the TSPLIB (up to 85,900 points), and VLSI instances with up to 744,710 points. This demonstrates that from a practical point of view, MWT instances can be handled quite well, despite their theoretical difficulty.

Digital Object Identifier 10.4230/LIPIcs.SoCG.2018.26

1 Introduction

Triangulating a set of points in the plane is a classic problem in computational geometry: given a planar point set S, find a maximal set of non-crossing line segments connecting the points in S. Triangulations have many real-world applications, for example in terrain modeling, finite element mesh generation and visualization. In general, a point set has exponentially many possible triangulations and a natural question is to ask for a triangulation that is optimal with respect to some optimality criterion. Well known and commonly used is the Delaunay triangulation, which optimizes several criteria at the same time: it maximizes the minimum angle and minimizes both the maximum circumcircle and the maximum smallest enclosing circle of all triangles. Another natural optimality criterion, and the one we are considering in this paper is minimizing the total weight of the resulting triangulation.

The minimum weight triangulation (MWT) is listed as one of the open problems in the famous book from 1979 by Garey and Johnson on NP-completeness [10]. The complexity status remained open for 27 years until Mulzer and Rote [18] finally resolved the question and showed NP-hardness of the MWT problem.

Independently, Gilbert [11] and Klincsek [17] showed that, when restricting it to simple polygons, the MWT problem can be solved in $O(n^3)$-time with dynamic programming. The dynamic programming approach can be generalized to polygons with k inner points. Hoffmann and Okamoto [16] showed how to obtain the MWT of such a point set in $O(6^k n^3 \log n)$-time. Their algorithm is based on a polygon decomposition through x-monotone paths. Grantson et al. [12] improved the algorithm to $O(n^4 4^k k)$ and showed another decomposition strategy based on cutting out triangles [13] which runs in $O(n^3 3^k k)$-time.

© Andreas Haas;
licensed under Creative Commons License CC-BY
34th Symposium on Computational Geometry (SoCG 2018).
Editors: Bettina Speckmann and Csaba Tóth; Article No. 26; pp. 26:1–26:27
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

3.1 Diamond property

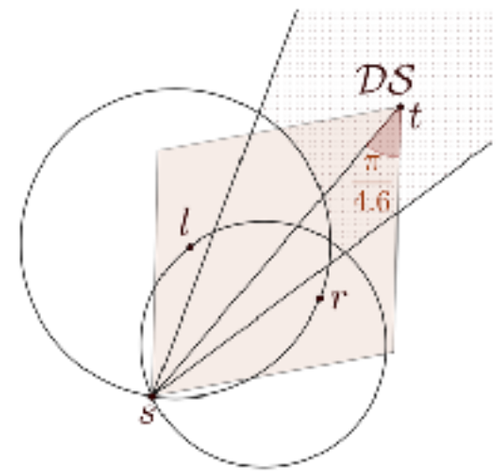
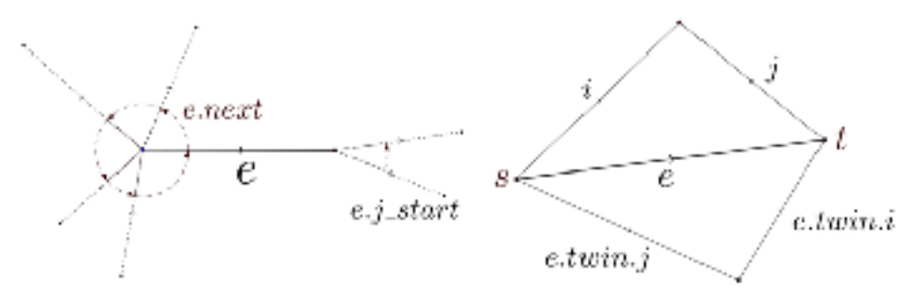


Figure 1 Points l and r induce a region DS such that all edges $e = st$ with $t \in DS$ fail the diamond test. DS is called a dead sector (dotted area).

3.2 LMT-skeleton

The LMT-skeleton was proposed by Dickerson et al. [8, 7], it is a subset of the minimum weight triangulation. The key observation is that MWT(S) is a locally minimal triangulation, i.e., no edge in MWT(S) can be flipped to reduce the total weight.

The LMT-skeleton algorithm eliminates all edges that have no certificate, i.e., for each edge e all pairs of empty triangles bordering e are examined until a certificate is found or no pairs are left. Eliminating e can invalidate previously found certificates. The remaining edges that are not intersected by any other remaining edge form the LMT-skeleton; they must be in all locally minimal triangulations.



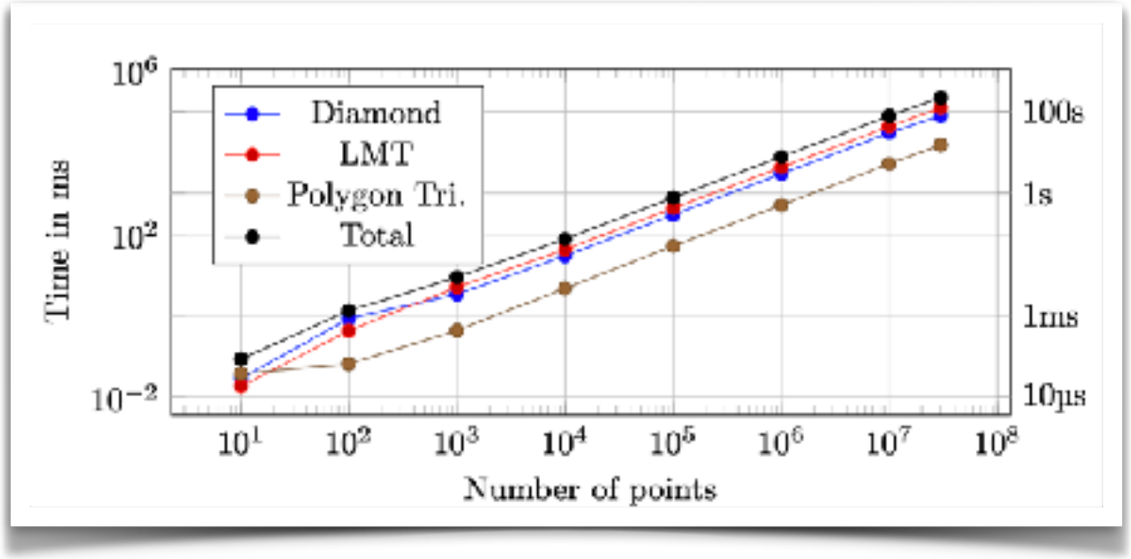
(a) Half-edge e.

(b) i and j point to the current certificate edges.

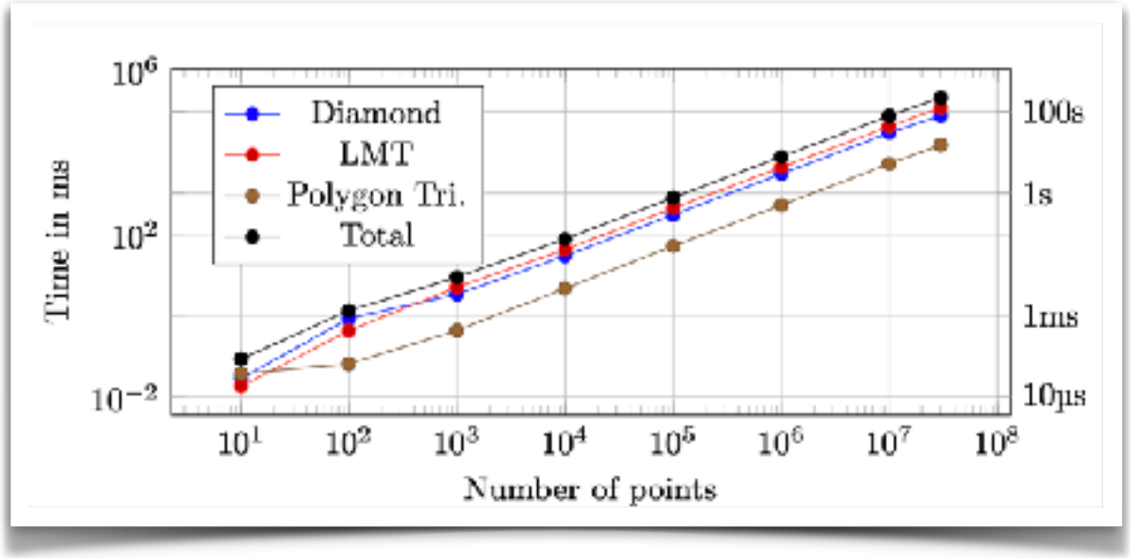
Figure 2 Representation of half-edge e.

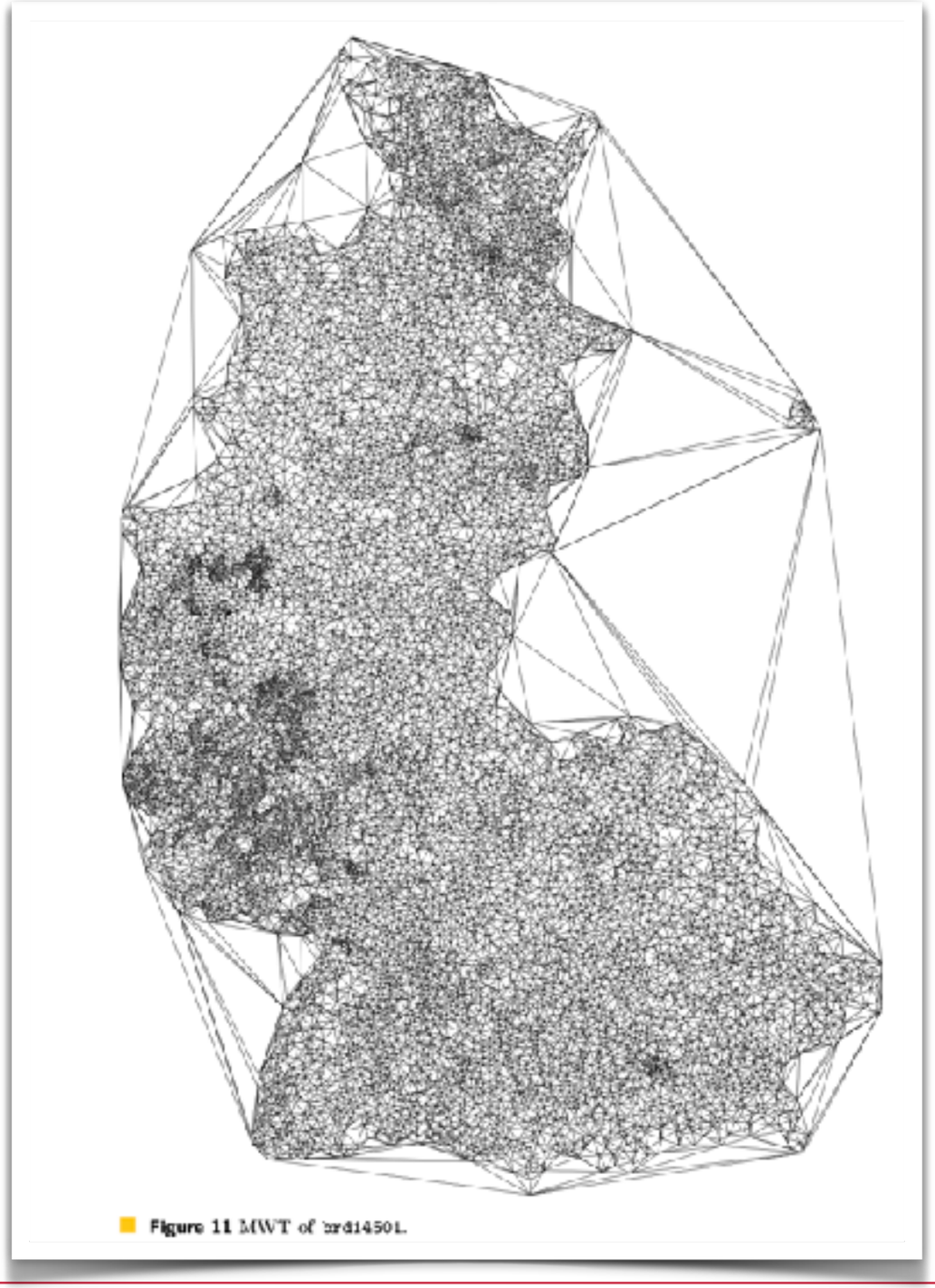
MWT is practically solvable to optimality [Haas, 2018]

MWT is practically solvable to optimality [Haas, 2018]

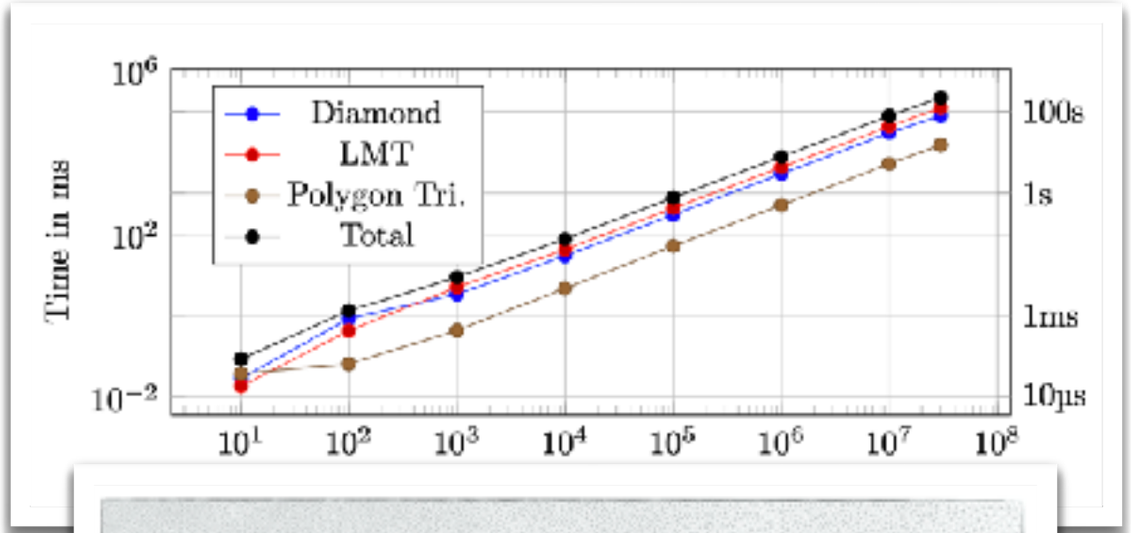


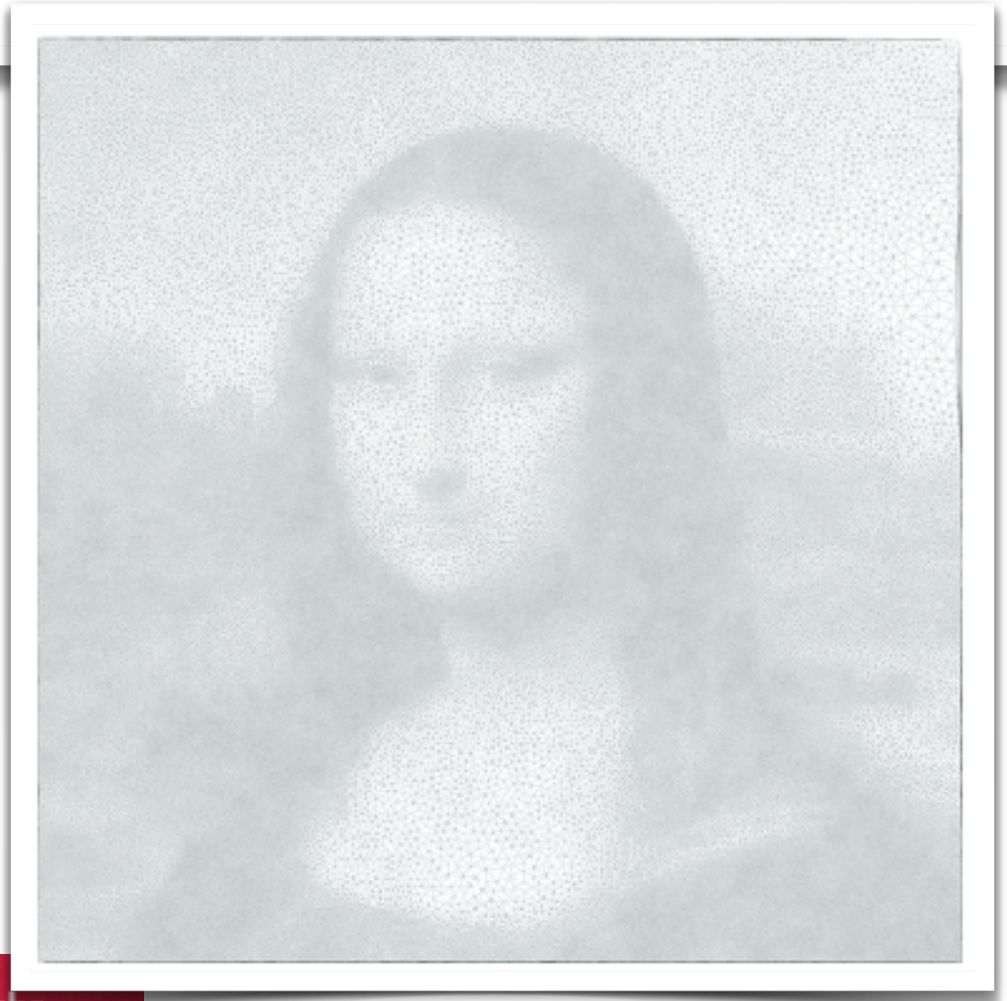
MWT is practically solvable to optimality [Haas, 2018]

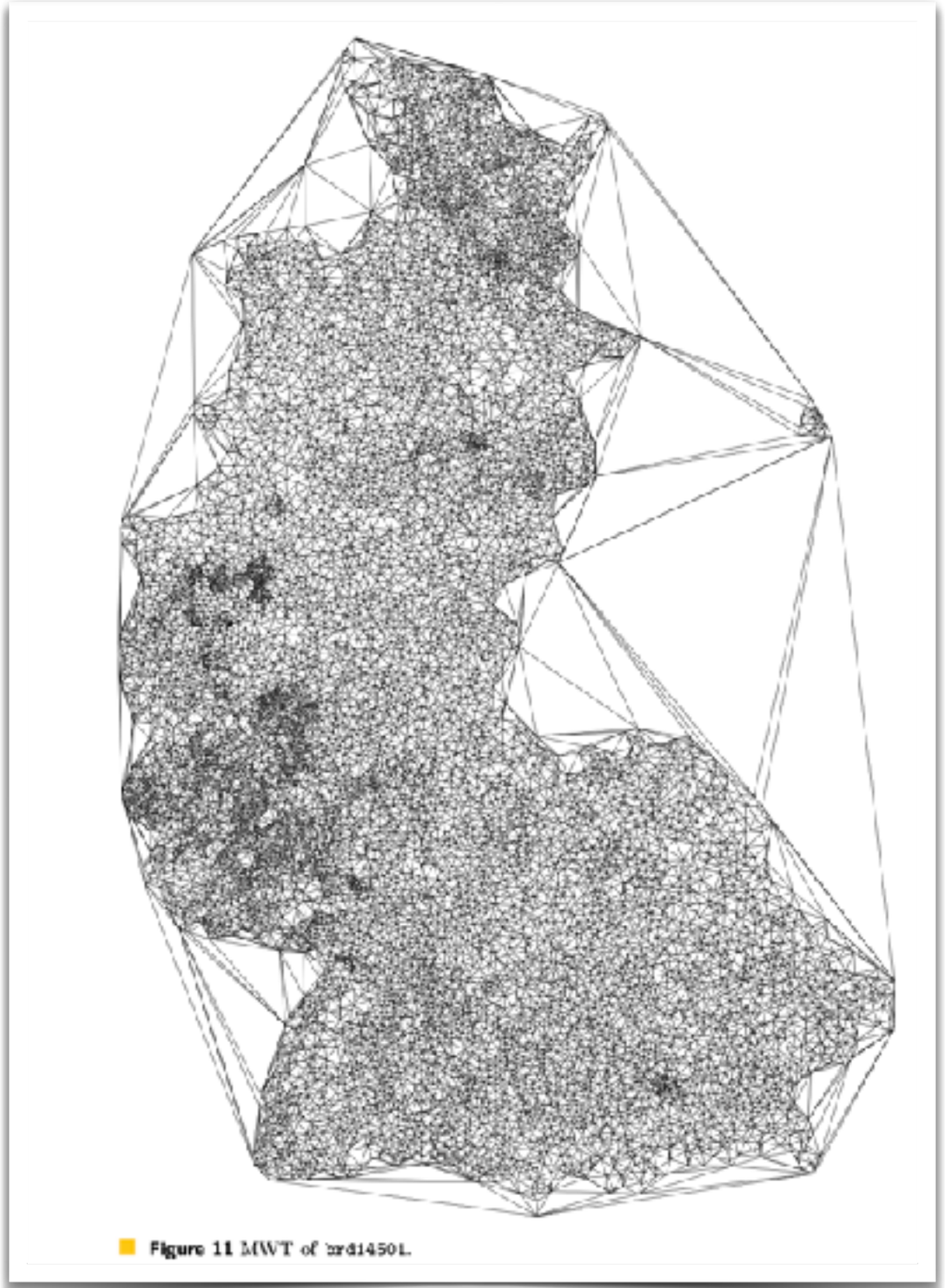




MWT is practically solvable to optimality [Haas, 2018]







4 Min-max triangulations

Minimizing maximum angle [Edelsbrunner, Tan 1993]

Minimizing maximum angle [Edelsbrunner, Tan 1993]

SIAM Journal on Computing, **22** (**3**), 527–551, (1993)

Minimizing maximum angle [Edelsbrunner, Tan 1993]

SIAM Journal on Computing, **22** (**3**), 527–551, (1993)

A QUADRATIC TIME ALGORITHM FOR THE MINMAX LENGTH TRIANGULATION

HERBERT EDELSBRUNNER* AND TIOW SENG TAN†

Abstract. We show that a triangulation of a set of n points in the plane that minimizes the maximum edge length can be computed in time $O(n^2)$. The algorithm is reasonably easy to implement and is based on the theorem that there is a triangulation with minmax edge length that contains the relative neighborhood graph of the points as a subgraph. With minor modifications the algorithm works for arbitrary normed metrics.

Key words. Computational geometry, point sets, triangulations, two dimensions, minmax edge length, normed metrics

5 Max-min triangulations

A Long-Standing Open Problem

A Long-Standing Open Problem

SIAM Journal on Computing, **22 (3)**, 527–551, (1993)

A Long-Standing Open Problem

SIAM Journal on Computing, **22 (3)**, 527–551, (1993)

A QUADRATIC TIME ALGORITHM FOR THE MINMAX LENGTH TRIANGULATION

HERBERT EDELSBRUNNER* AND TIOW SENG TAN†

Abstract. We show that a triangulation of a set of n points in the plane that minimizes the maximum edge length can be computed in time $O(n^2)$. The algorithm is reasonably easy to implement and is based on the theorem that there is a triangulation with minmax edge length that contains the relative neighborhood graph of the points as a subgraph. With minor modifications the algorithm works for arbitrary normed metrics.

Key words. Computational geometry, point sets, triangulations, two dimensions, minmax edge length, normed metrics

A Long-Standing Open Problem

SIAM Journal on Computing, **22 (3)**, 527–551, (1993)

A QUADRATIC TIME ALGORITHM FOR THE MINMAX LENGTH TRIANGULATION

HERBERT EDELSBRUNNER* AND TIOW SENG TAN†

Abstract. We show that a triangulation of a set of n points in the plane that minimizes the maximum edge length can be computed in time $O(n^2)$. The algorithm is reasonably easy to implement and is based on the theorem that there is a triangulation with minmax edge length that contains the relative neighborhood graph of the points as a subgraph. With minor modifications the algorithm works for arbitrary normed metrics.

Key words. Computational geometry, point sets, triangulations, two dimensions, minmax edge length, normed metrics

The results of this paper are an out-growth of our general efforts to understand triangulations that optimize length criteria. There are, however, still many related problems whose complexities remain open. These include the problem of minimizing the entire vector of edge-lengths, the minimum length triangulation problem, and the maxmin length triangulation problem.

A Long-Standing Open Problem

SIAM Journal on Computing, **22 (3)**, 527–551, (1993)

A QUADRATIC TIME ALGORITHM FOR THE MINMAX LENGTH TRIANGULATION

HERBERT EDELSBRUNNER* AND TIOW SENG TAN†

Abstract. We show that a triangulation of a set of n points in the plane that minimizes the maximum edge length can be computed in time $O(n^2)$. The algorithm is reasonably easy to implement and is based on the theorem that there is a triangulation with minmax edge length that contains the relative neighborhood graph of the points as a subgraph. With minor modifications the algorithm works for arbitrary normed metrics.

Key words. Computational geometry, point sets, triangulations, two dimensions, minmax edge length, normed metrics

The results of this paper are an out-growth of our general efforts to understand triangulations that optimize length criteria. There are, however, still many related problems whose complexities remain open. These include the problem of minimizing the entire vector of edge-lengths, the minimum length triangulation problem, and the maxmin length triangulation problem.

A Long-Standing Open Problem

SIAM Journal on Computing, **22 (3)**, 527–551, (1993)

A QUADRATIC TIME ALGORITHM FOR THE MINMAX LENGTH TRIANGULATION

HERBERT EDELSBRUNNER* AND TIOW SENG TAN†

Abstract. We show that a triangulation of a set of n points in the plane that minimizes the maximum edge length can be computed in time $O(n^2)$. The algorithm is reasonably easy to implement and is based on the theorem that there is a triangulation with minmax edge length that contains the relative neighborhood graph of the points as a subgraph. With minor modifications the algorithm works for arbitrary normed metrics.

Key words. Computational geometry, point sets, triangulations, two dimensions, minmax edge length, normed metrics

The results of this paper are an out-growth of our general efforts to understand triangulations that optimize length criteria. There are, however, still many related problems whose complexities remain open. These include the problem of minimizing the entire vector of edge-lengths, the minimum length triangulation problem, and the maxmin length triangulation problem.

A Long-Standing Open Problem

SIAM Journal on Computing, **22 (3)**, 527–551, (1993)

A QUADRATIC TIME ALGORITHM FOR THE MINMAX LENGTH TRIANGULATION

HERBERT EDELSBRUNNER* AND TIOW SENG TAN†

Abstract. We show that a triangulation of a set of n points in the plane that minimizes the maximum edge length can be computed in time $O(n^2)$. The algorithm is reasonably easy to implement and is based on the theorem that there is a triangulation with minmax edge length that contains the relative neighborhood graph of the points as a subgraph. With minor modifications the algorithm works for arbitrary normed metrics.

Key words. Computational geometry, point sets, triangulations, two dimensions, minmax edge length, normed metrics

The results of this paper are an out-growth of our general efforts to understand triangulations that optimize length criteria. There are, however, still many related problems whose complexities remain open. These include the problem of minimizing the entire vector of edge-lengths, the minimum length triangulation problem, and the maxmin length triangulation problem.

Open Problem

Journal of Computational Geometry jocg.org

COMPUTING MAXMIN EDGE LENGTH TRIANGULATIONS*

Sándor P. Fekete,† *Winfried Hellmann,*† *Michael Hemmer,*† *Arne Schmidt,*† *and Julian Troegel*†

ABSTRACT. In 1991, Edelsbrunner and Tan gave an $O(n^2)$ algorithm for finding the MinMax Edge Length Triangulation of a set of points in the plane, but stated the complexity of finding a MaxMin Edge Length Triangulation (MELT) as a natural open problem. We resolve this long-standing problem by showing that computing a MELT is NP-complete. Moreover, we prove that (unless P=NP), there is no polynomial-time approximation algorithm that can approximate MELT within any polynomial factor.

While this may be taken as conclusive evidence from a *theoretical* point of view that the problem is hopelessly intractable, it still makes sense to consider powerful optimization methods, such as integer programming (IP), in order to obtain provably optimal solutions for intances of non-trivial size. A straightforward IP based on pairwise disjointness of the $\Theta(n^2)$ segments between the n points has $\Theta(n^4)$ constraints, making this IP hopelessly intractable from a *practical* point of view, even for relatively small n. The main algorithm engineering twist of this paper is to demonstrate how the combination of geometric insights with refined methods of combinatorial optimization can still help to put together an exact method capable of computing optimal MELT solutions for planar point sets up to $n = 200$. Our key idea is to exploit specific geometric properties in combination with more compact IP formulations, such that we are able to drastically reduce the number of constraints. On the practical side, we combine two of the most powerful software packages for the individual components: CGAL for carrying out the geometric computations, and CPLEX for solving the IPs. In addition, we discuss specific analytic aspects of the speedup for random point sets.

1 Introduction

Triangulating a set of points is one of the basic problems of Computational Geometry: given a set P of n points in the plane, connect them by a maximal set of non-crossing line segments. This implies that all bounded faces of the resulting planar arrangement are triangles, while the exterior face is the complement of the convex hull of P.

Triangulations are computed and used in a large variety of contexts, e.g., in mesh generation, but also as a stepping stone for other tasks. While it is not hard to compute some triangulation, most of these tasks require triangulations with special properties that

*A preliminary extended abstract appears in the Proceedings of ALENEX 2015 [16].

†Department of Computer Science, TU Braunschweig, 38106 Braunschweig, Germany. {s.fekete, w.hellmann, m.hemmer, arne.schmidt, j.troegel}@tu-bs.de.

ing, **22 (3)**, 527–551, (1993)

LGORITHM FOR THE TRIANGULATION

R* AND TIOW SENG TAN†

a set of n points in the plane that minimizes me $O(n^2)$. The algorithm is reasonably easy to is a triangulation with minmax edge length that nts as a subgraph. With minor modifications the

ets, triangulations, two dimensions, minmax edge

vth of our general efforts to understand There are, however, still many related hese include the problem of minimizing m length triangulation problem, and the

Open Problem

Journal of Computational Geometry jocg.org

COMPUTING MAXMIN EDGE LENGTH TRIANGULATIONS*

Sándor P. Fekete,† *Winfried Hellmann,*† *Michael Hemmer,*† *Arne Schmidt,*† *and Julian Troegel* †

ABSTRACT. In 1991, Edelsbrunner and Tan gave an $O(n^2)$ algorithm for finding the MinMax

COMPUTING MAXMIN EDGE LENGTH TRIANGULATIONS*

Sándor P. Fekete,† *Winfried Hellmann,*† *Michael Hemmer,*† *Arne Schmidt,*† *and Julian Troegel* †

ing, **22 (3)**, 527–551, (1993)

LGORITHM FOR THE

TRIANGULATION

Our key idea is to exploit specific geometric properties in combination with more compact IP formulations, such that we are able to drastically reduce the number of constraints. On the practical side, we combine two of the most powerful software packages for the individual components: CGAL for carrying out the geometric computati

the IPs. In addition, we discuss specific analytic aspects of t

sets.

JoCG 9(1), 1–26, 2018

1 Introduction

Triangulating a set of points is one of the basic problems of Computational Geometry: given a set P of n points in the plane, connect them by a maximal set of non-crossing line segments. This implies that all bounded faces of the resulting planar arrangement are triangles, while the exterior face is the complement of the convex hull of P.

Triangulations are computed and used in a large variety of contexts, e.g., in mesh generation, but also as a stepping stone for other tasks. While it is not hard to compute some triangulation, most of these tasks require triangulations with special properties that

*A preliminary extended abstract appears in the Proceedings of ALENEX 2015 [16].

†Department of Computer Science, TU Braunschweig, 38106 Braunschweig, Germany. {s.fekete, w.hellmann, m.hemmer, arne.schmidt, j.troegel}@tu-bs.de.

CC BY

vth of our general efforts to understand

There are, however, still many related

These include the problem of minimizing

m length triangulation problem, and the

5.1: Complexity

An Open Problem Resolved

An Open Problem Resolved

THEOREM 3.1. *It is NP-hard to decide whether a set P of n points in the plane has a triangulation with smallest edge of length at least L, for some positive number L.*

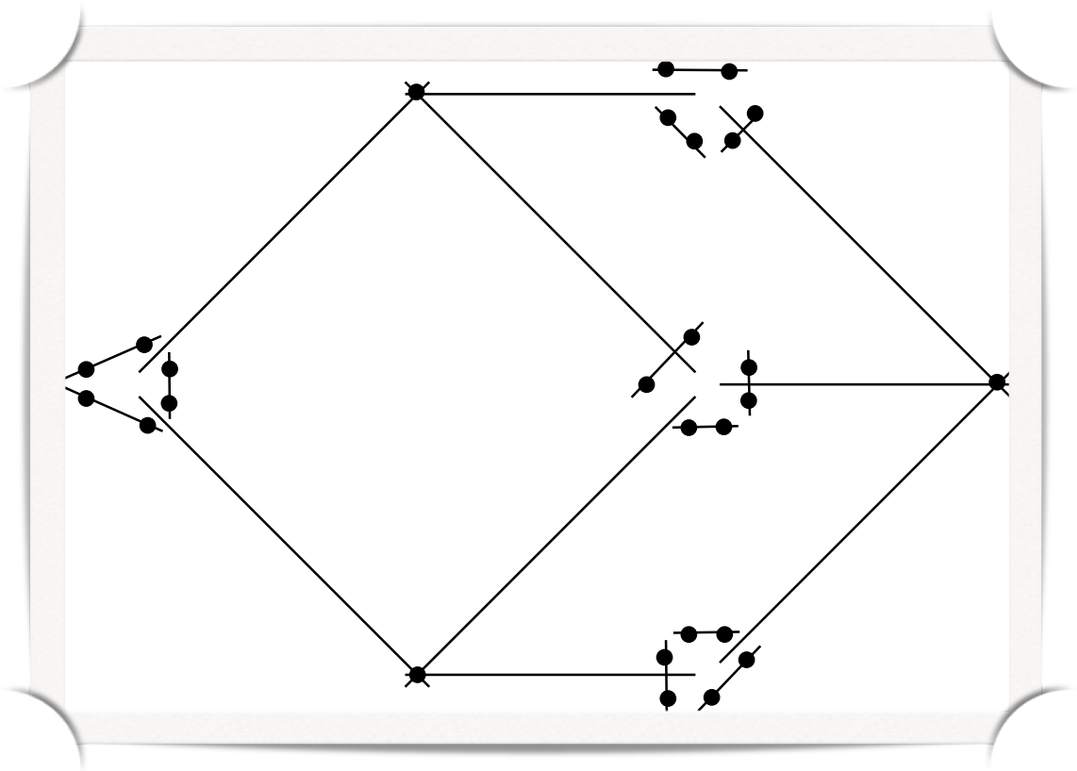
An Auxiliary Problem

An Auxiliary Problem

COVERING BY DISJOINT SEGMENTS

An Auxiliary Problem

COVERING BY DISJOINT SEGMENTS

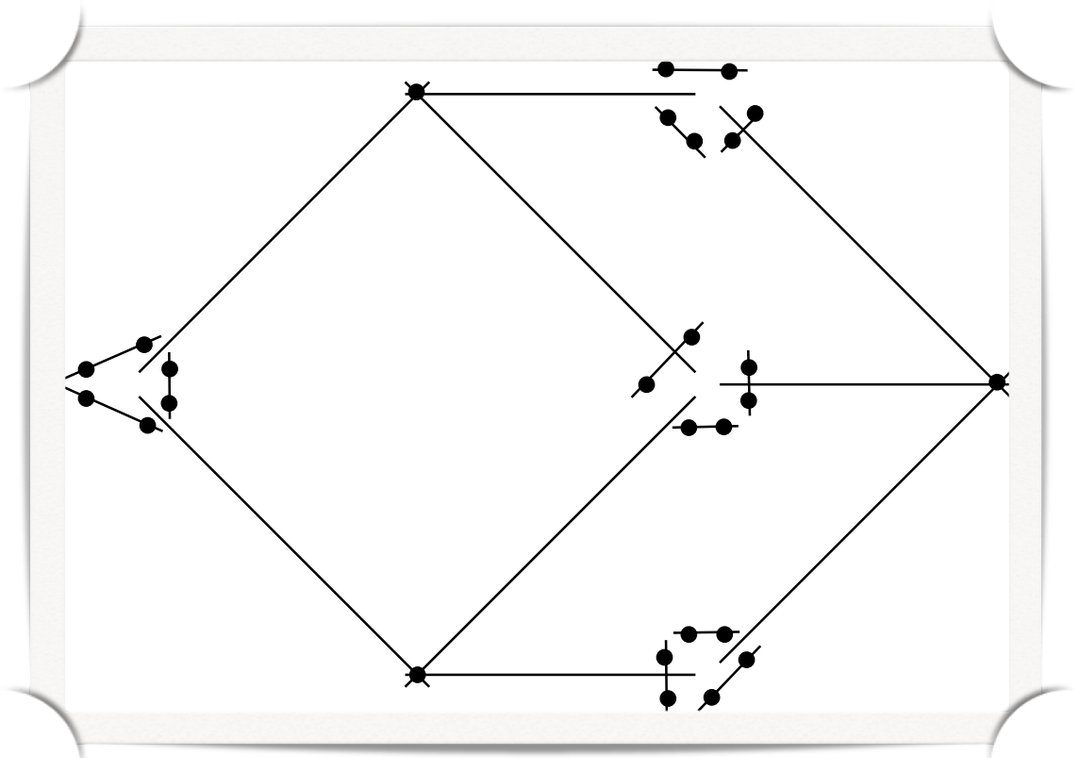


An Auxiliary Problem

COVERING BY DISJOINT SEGMENTS

Given: *A specified set S of line segments ("stabbers") in the Euclidean plane, and a subset T of their intersection points ("targets").*

Wanted: *A non-intersecting subset of the stabbers that covers all targets.*

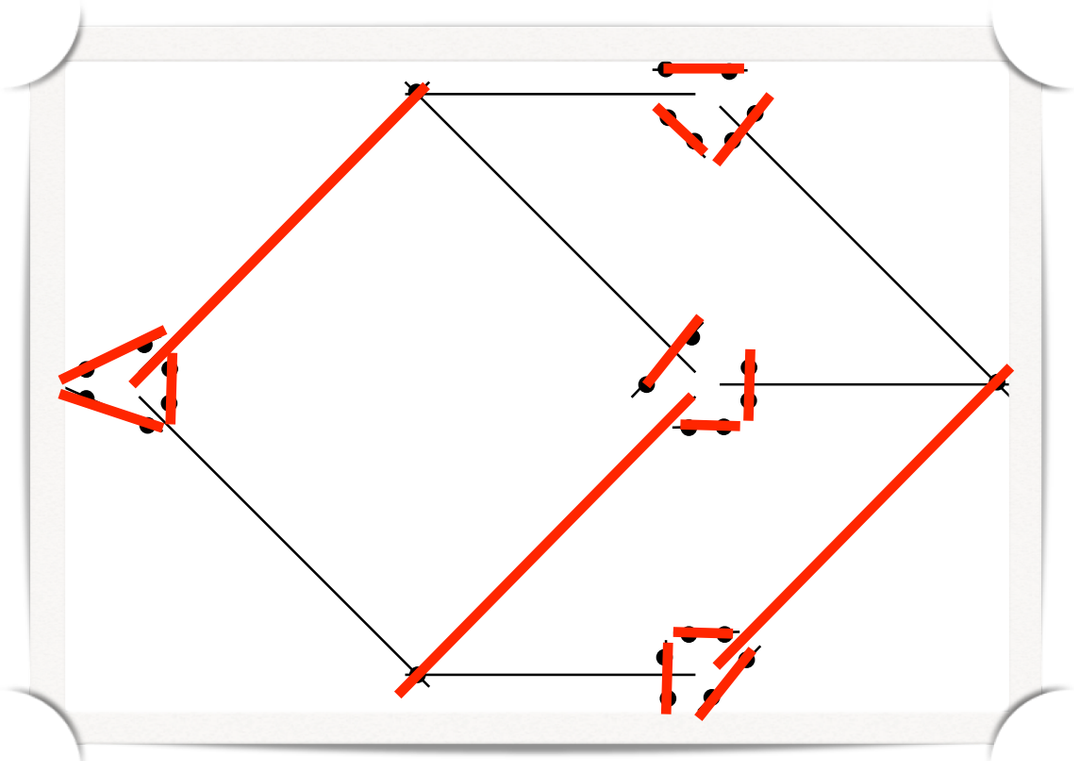


An Auxiliary Problem

COVERING BY DISJOINT SEGMENTS

Given: *A specified set S of line segments ("stabbers") in the Euclidean plane, and a subset T of their intersection points ("targets").*

Wanted: *A non-intersecting subset of the stabbers that covers all targets.*



Proof of NP-Hardness

Proof of NP-Hardness

Idea: Use reduction from ***Planar 3SAT***

Proof of NP-Hardness

Idea: Use reduction from ***Planar 3SAT***

$$\boxed{(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_3 \vee \overline{x_4}) \wedge (\overline{x_1} \vee x_2 \vee x_4)}$$

Proof of NP-Hardness

Idea: Use reduction from ***Planar 3SAT***

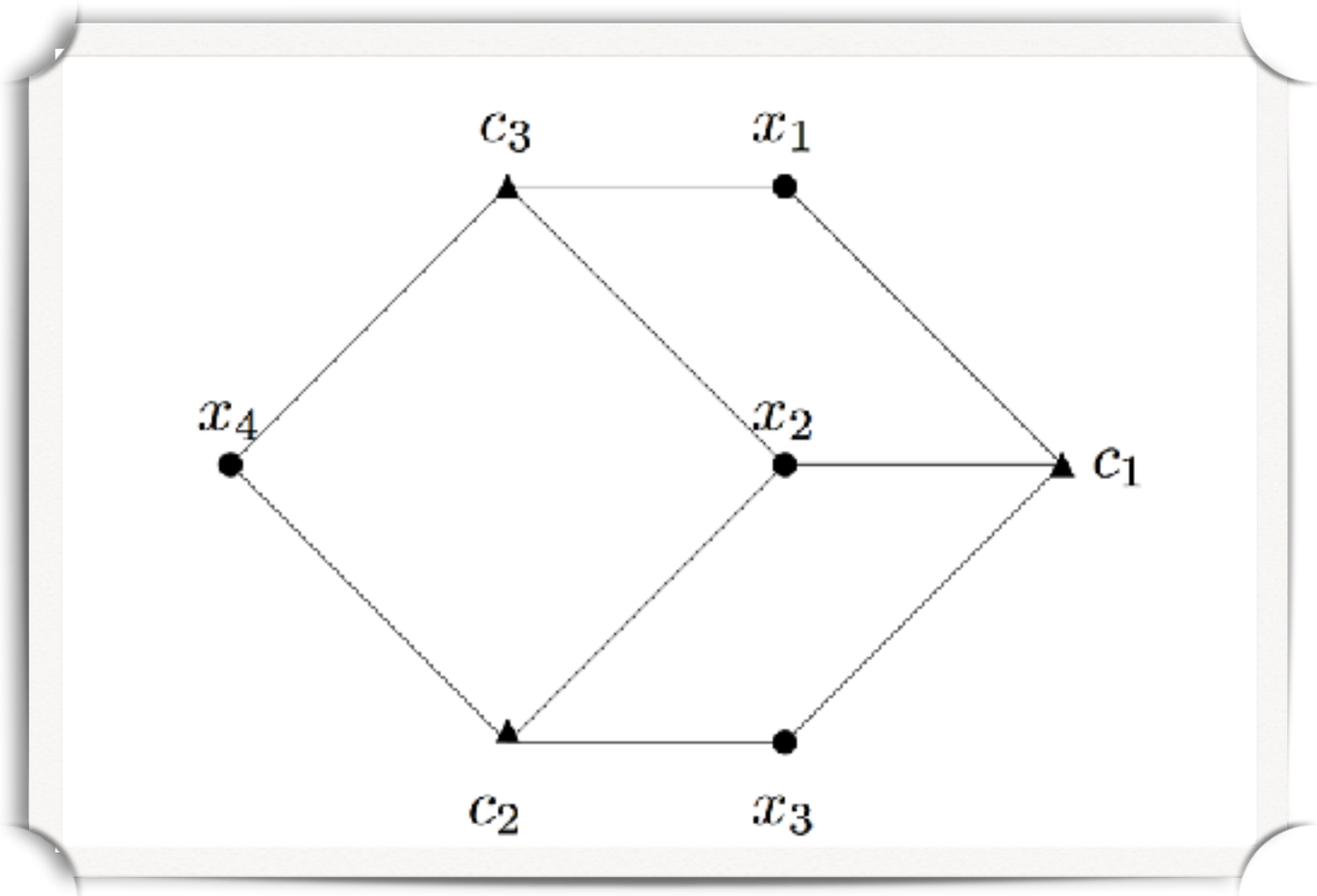
$$\boxed{(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_3 \vee \overline{x_4}) \wedge (\overline{x_1} \vee x_2 \vee x_4)}$$

$$\Downarrow$$

Proof of NP-Hardness

Idea: Use reduction from ***Planar 3SAT***

$$(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_3 \vee \overline{x_4}) \wedge (\overline{x_1} \vee x_2 \vee x_4)$$



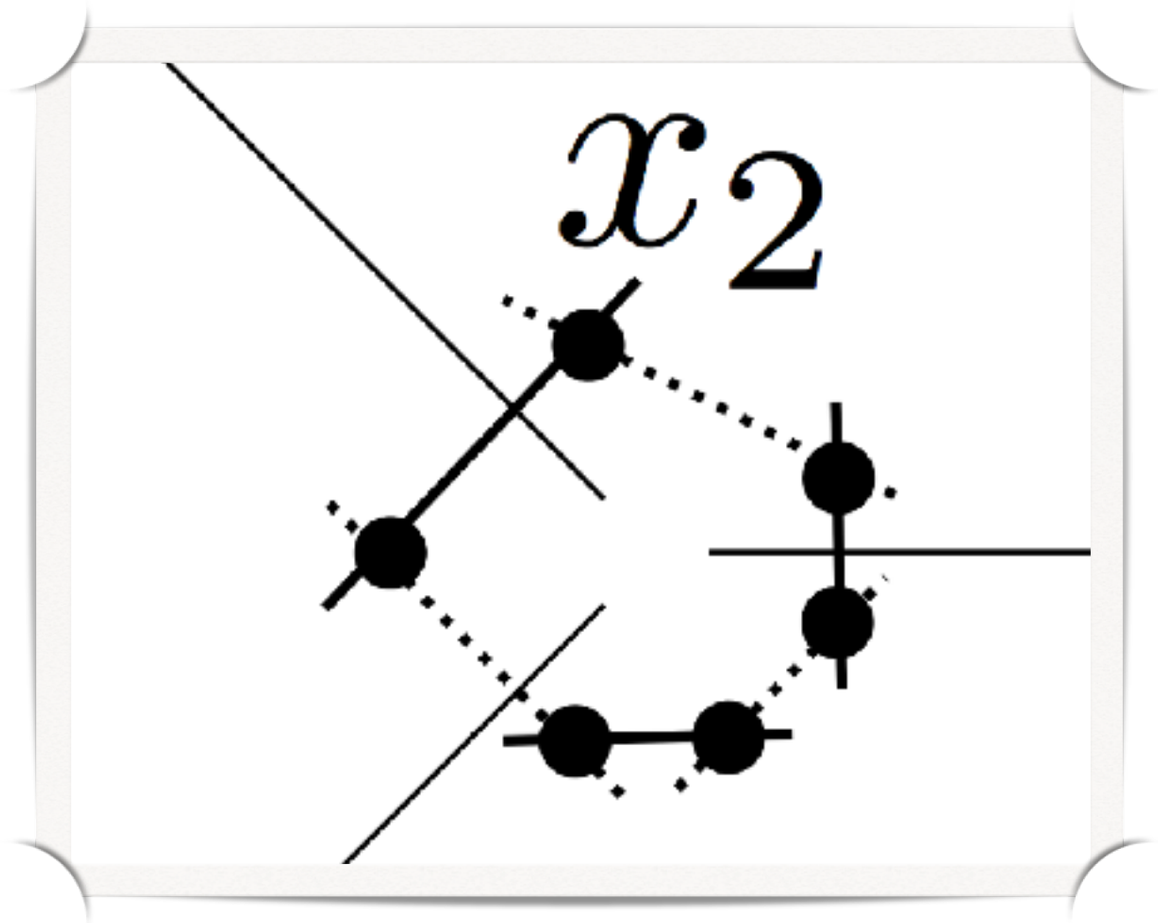
Proof of NP-Hardness

Proof of NP-Hardness

Component 1: Variables

Proof of NP-Hardness

Component 1: Variables



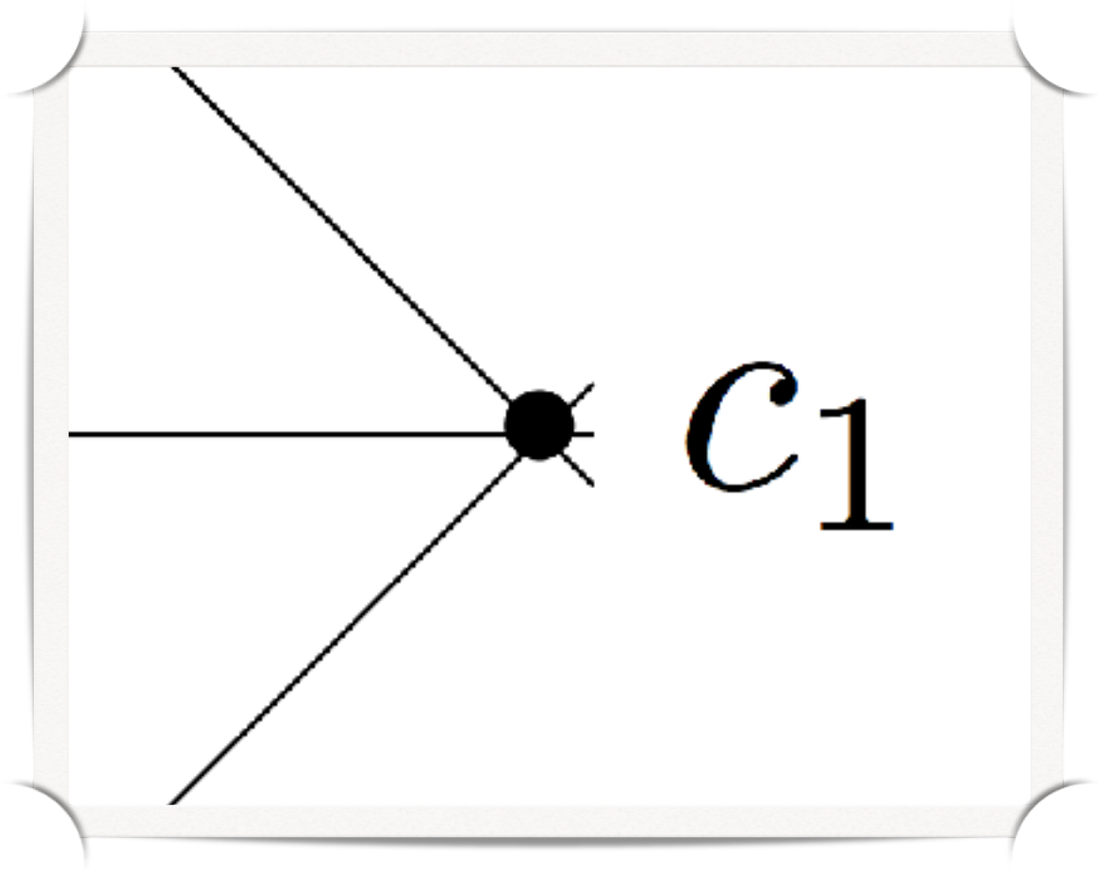
Proof of NP-Hardness

Proof of NP-Hardness

Component 2: Clauses

Proof of NP-Hardness

Component 2: Clauses



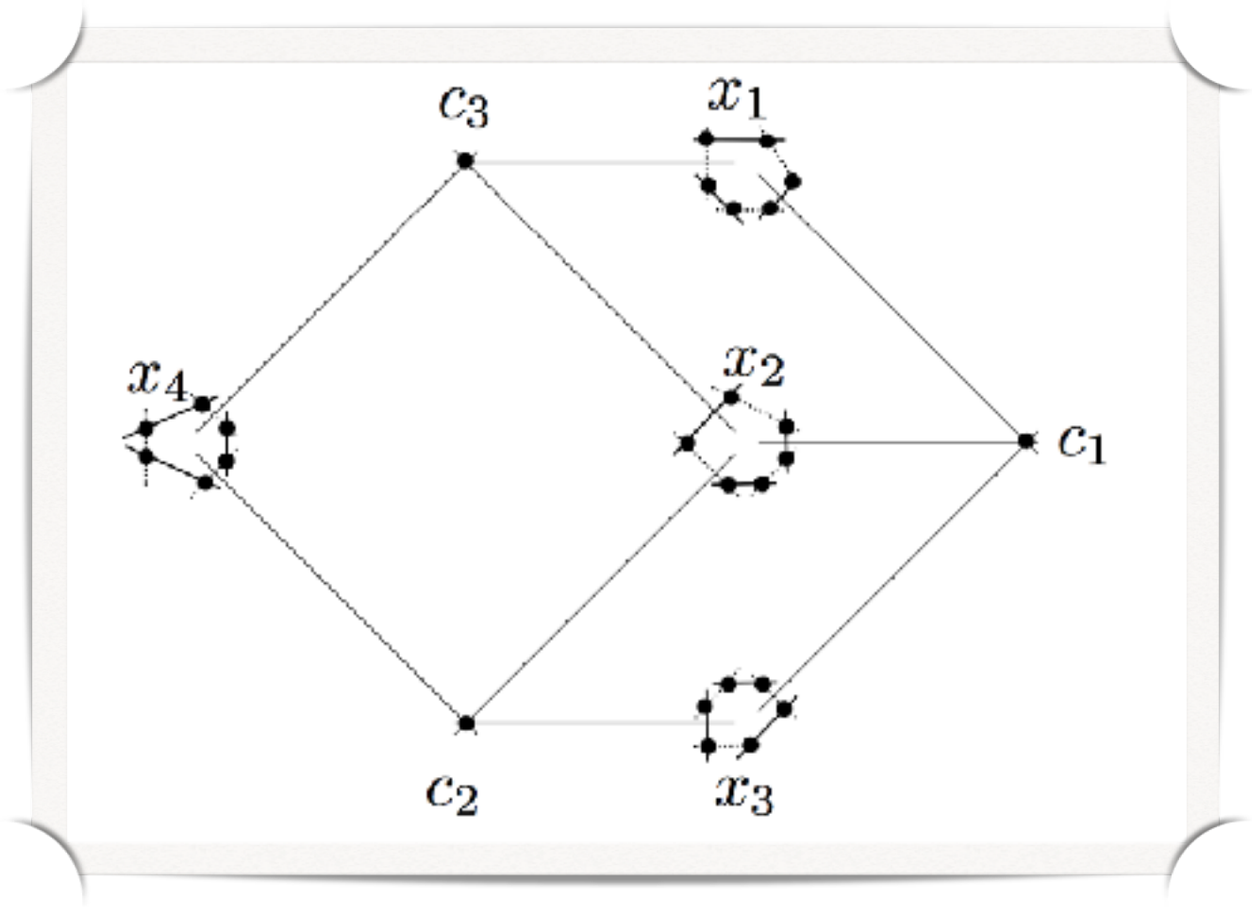
Proof of NP-Hardness

Proof of NP-Hardness

Overall:

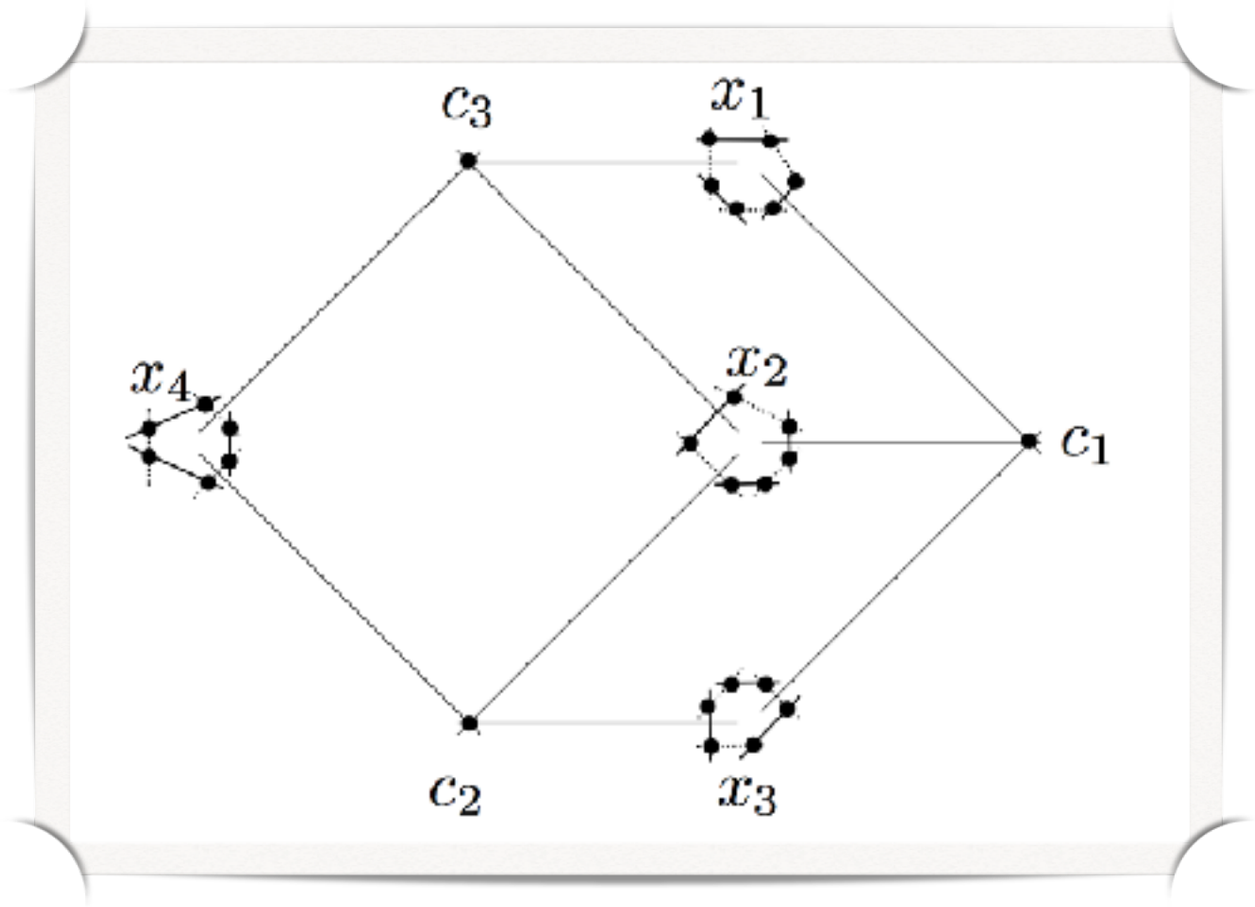
Proof of NP-Hardness

Overall:



Proof of NP-Hardness

Overall:



LEMMA 3.2. *The problem CDS is NP-complete.*

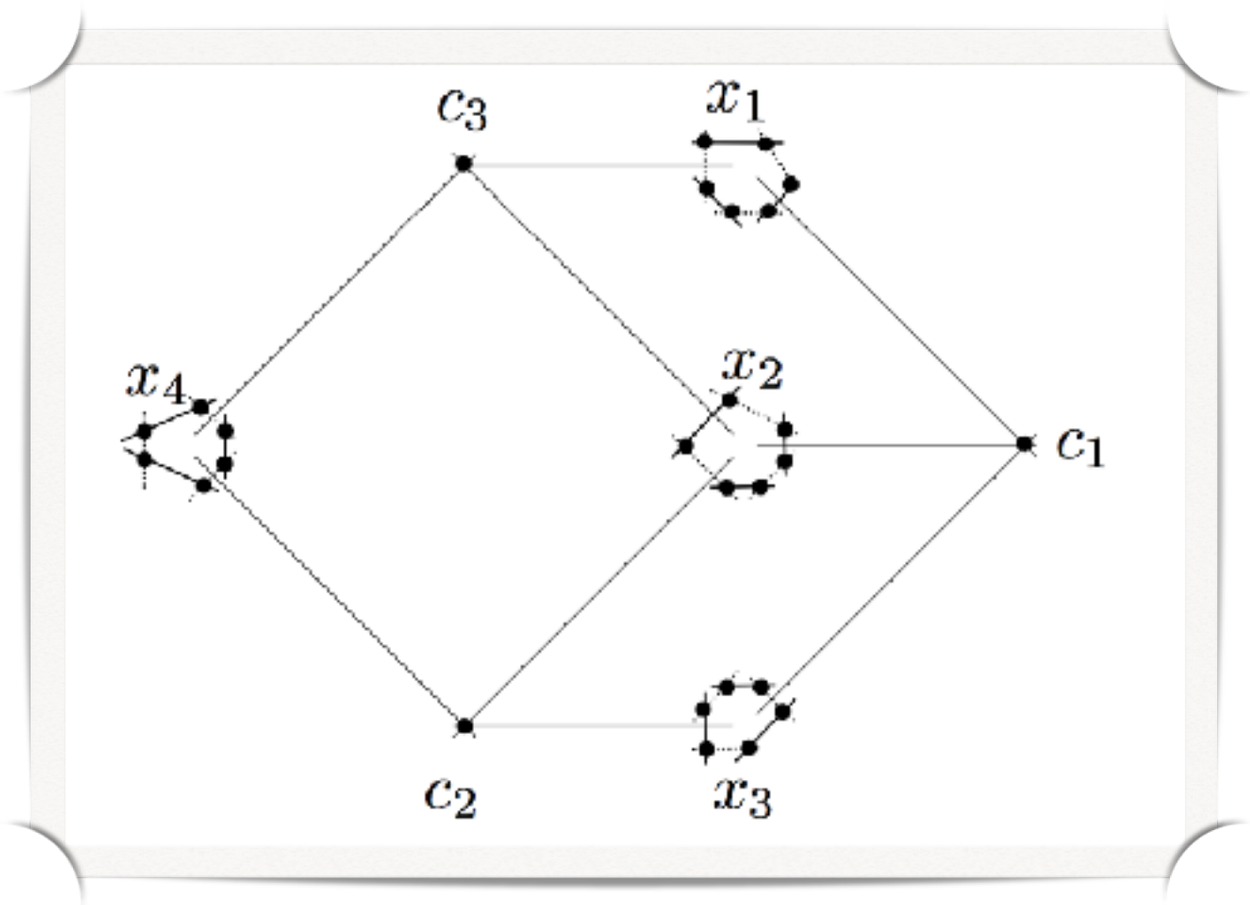
NP-Hardness of MaxMin Triangulations

NP-Hardness of MaxMin Triangulations

Idea: Replace points to be covered by point pairs to be separated.

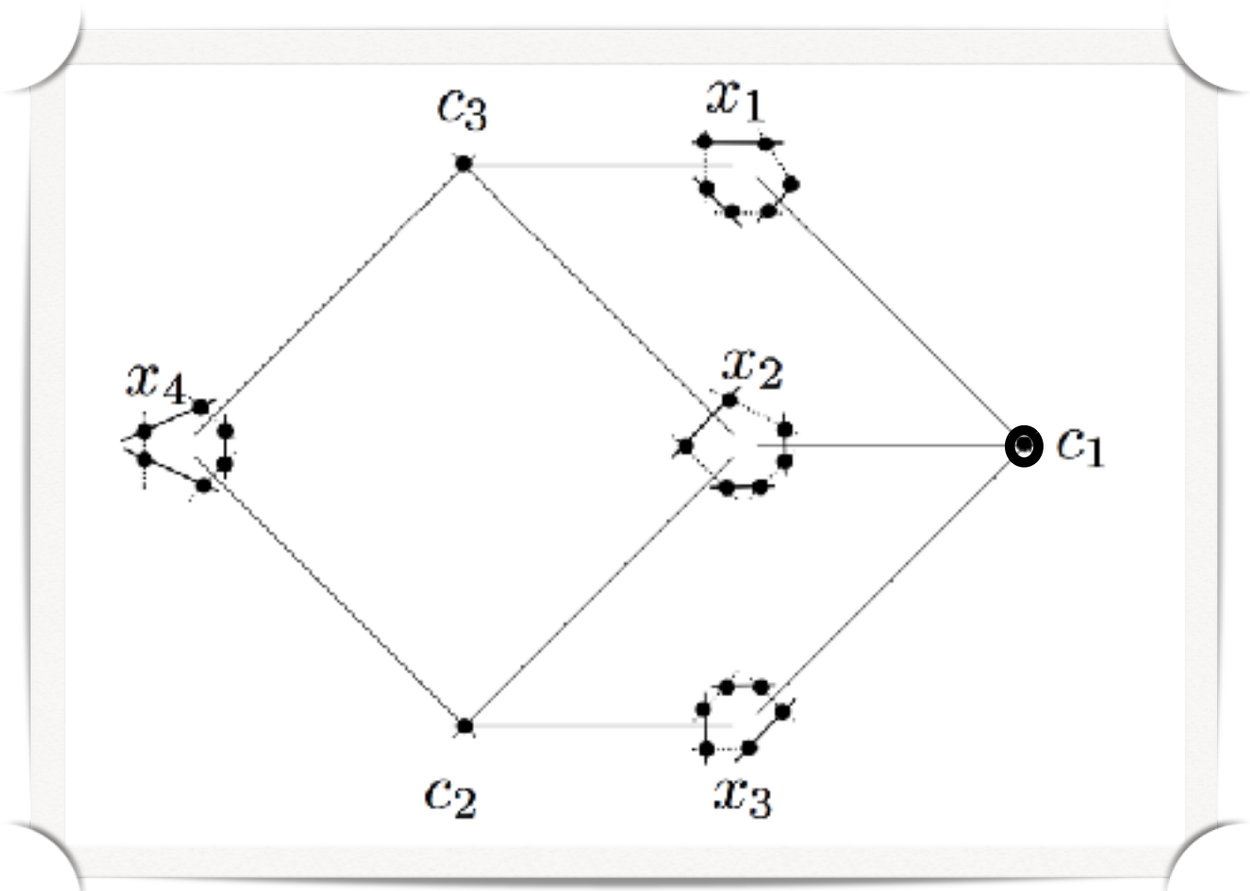
NP-Hardness of MaxMin Triangulations

Idea: Replace points to be covered by point pairs to be separated.



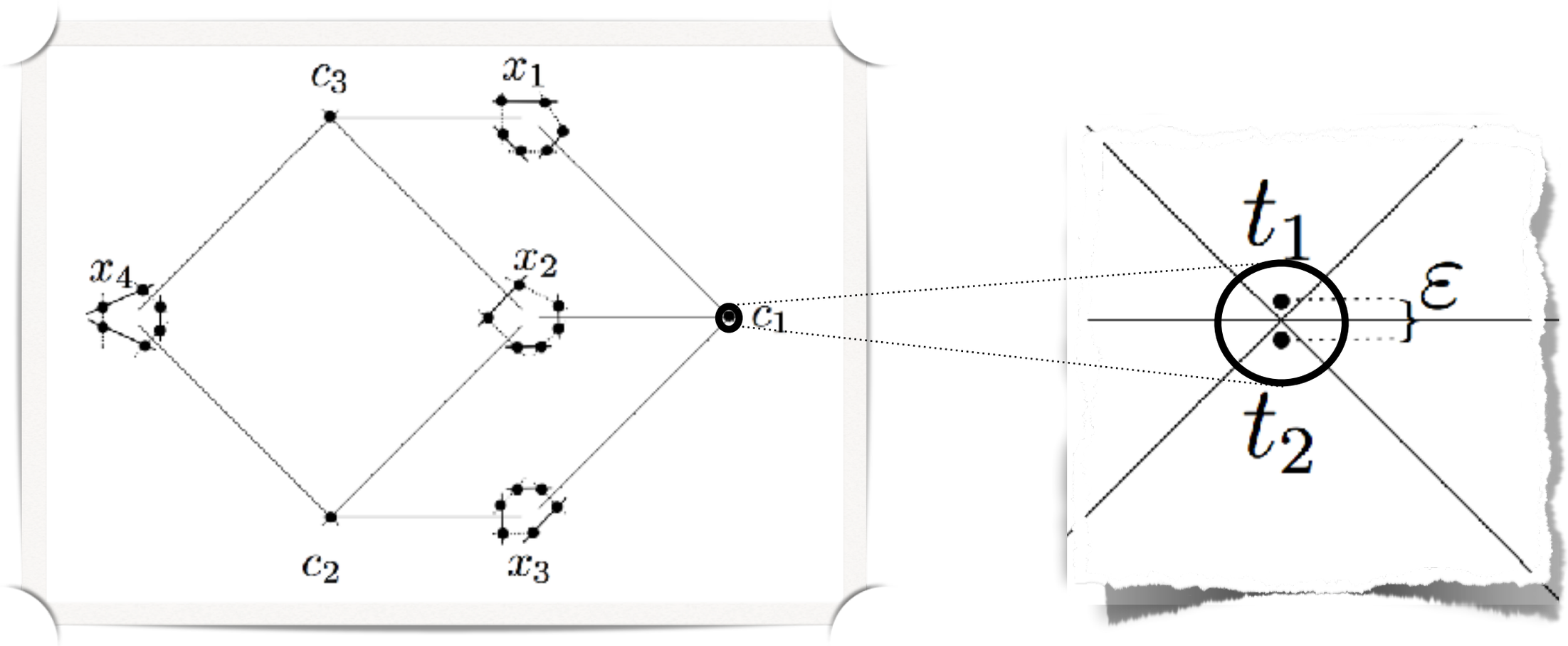
NP-Hardness of MaxMin Triangulations

Idea: Replace points to be covered by point pairs to be separated.



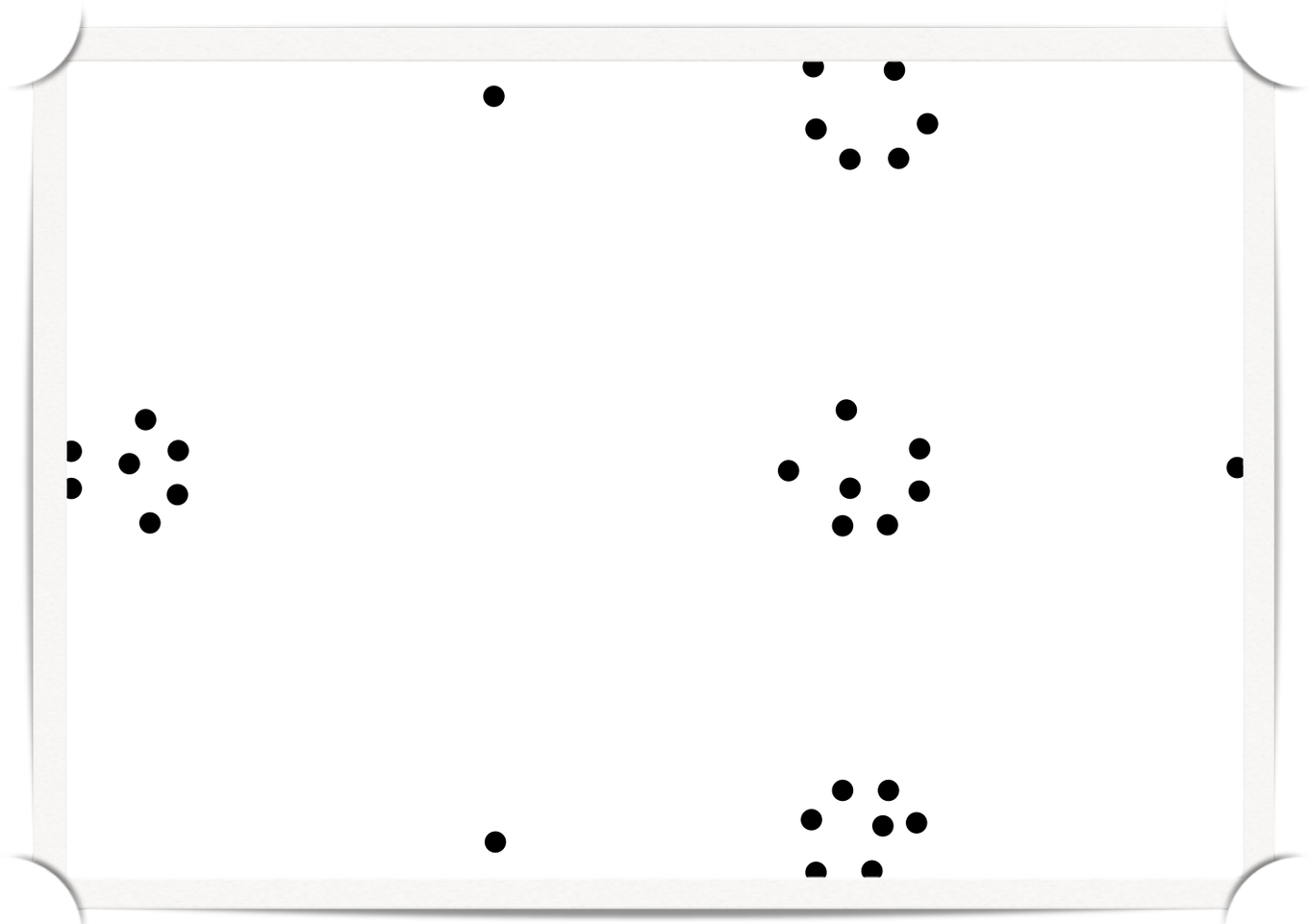
NP-Hardness of MaxMin Triangulations

Idea: Replace points to be covered by point pairs to be separated.

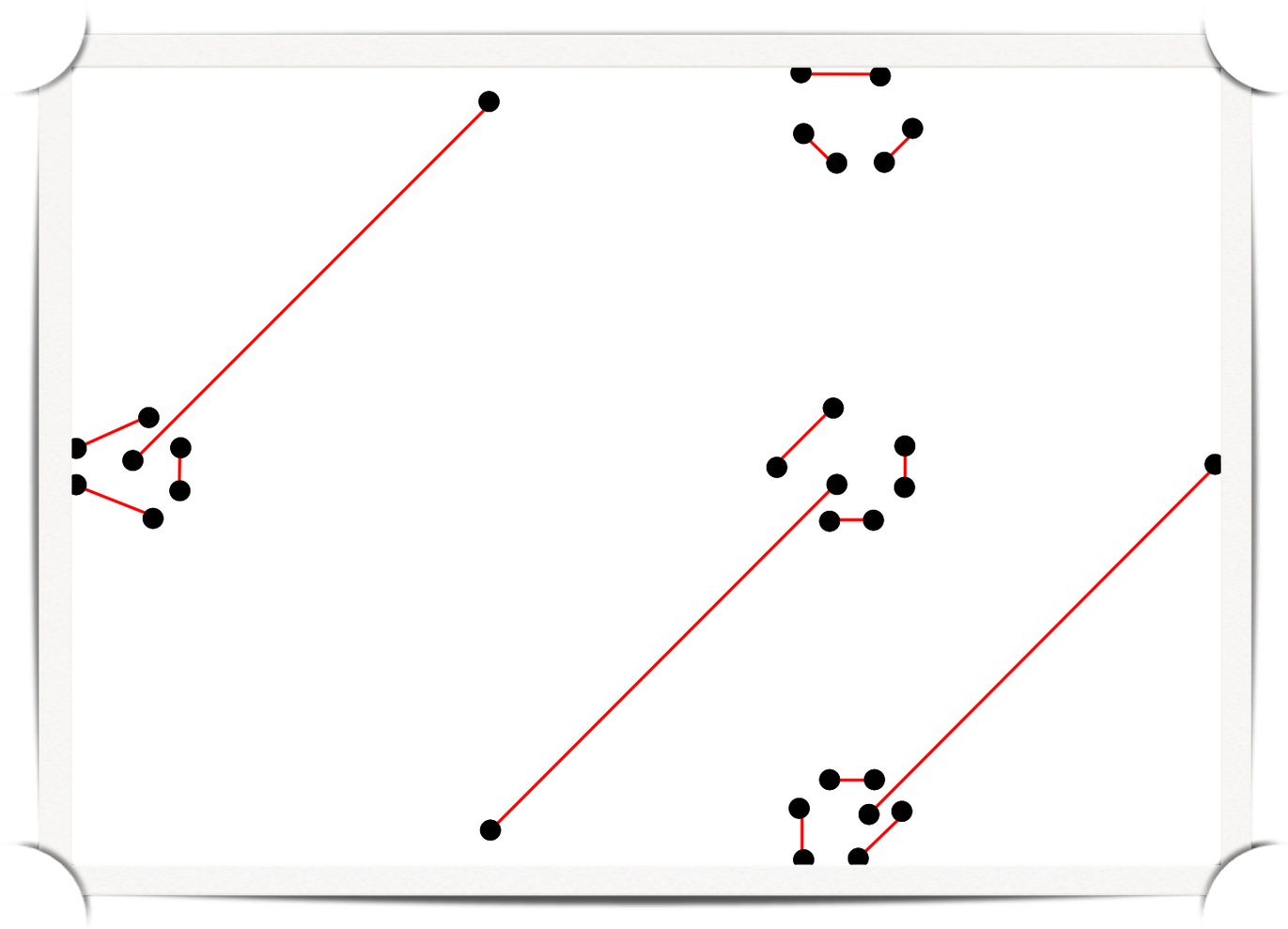


NP-Hardness of MaxMin Triangulations

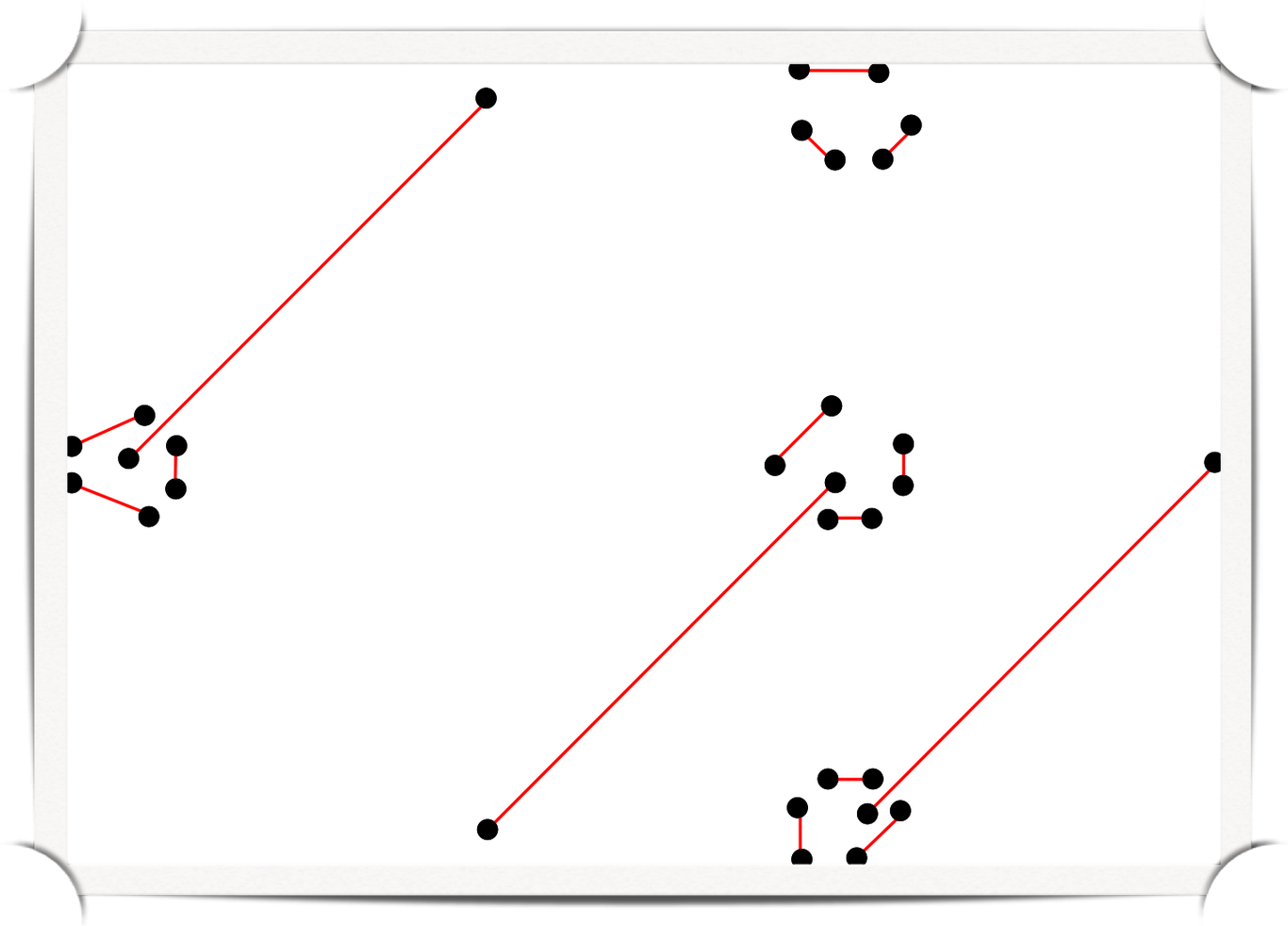
NP-Hardness of MaxMin Triangulations



NP-Hardness of MaxMin Triangulations

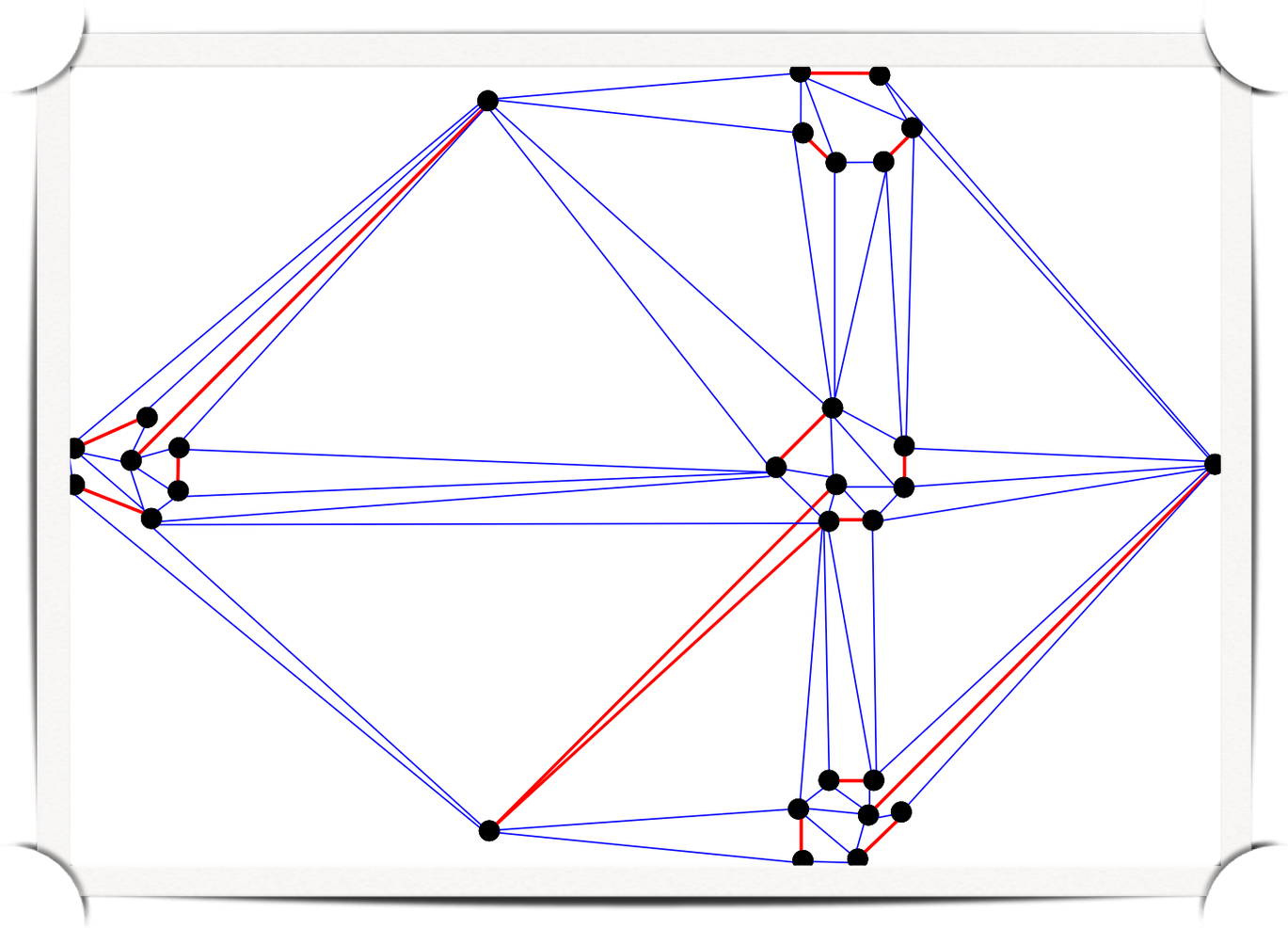


NP-Hardness of MaxMin Triangulations



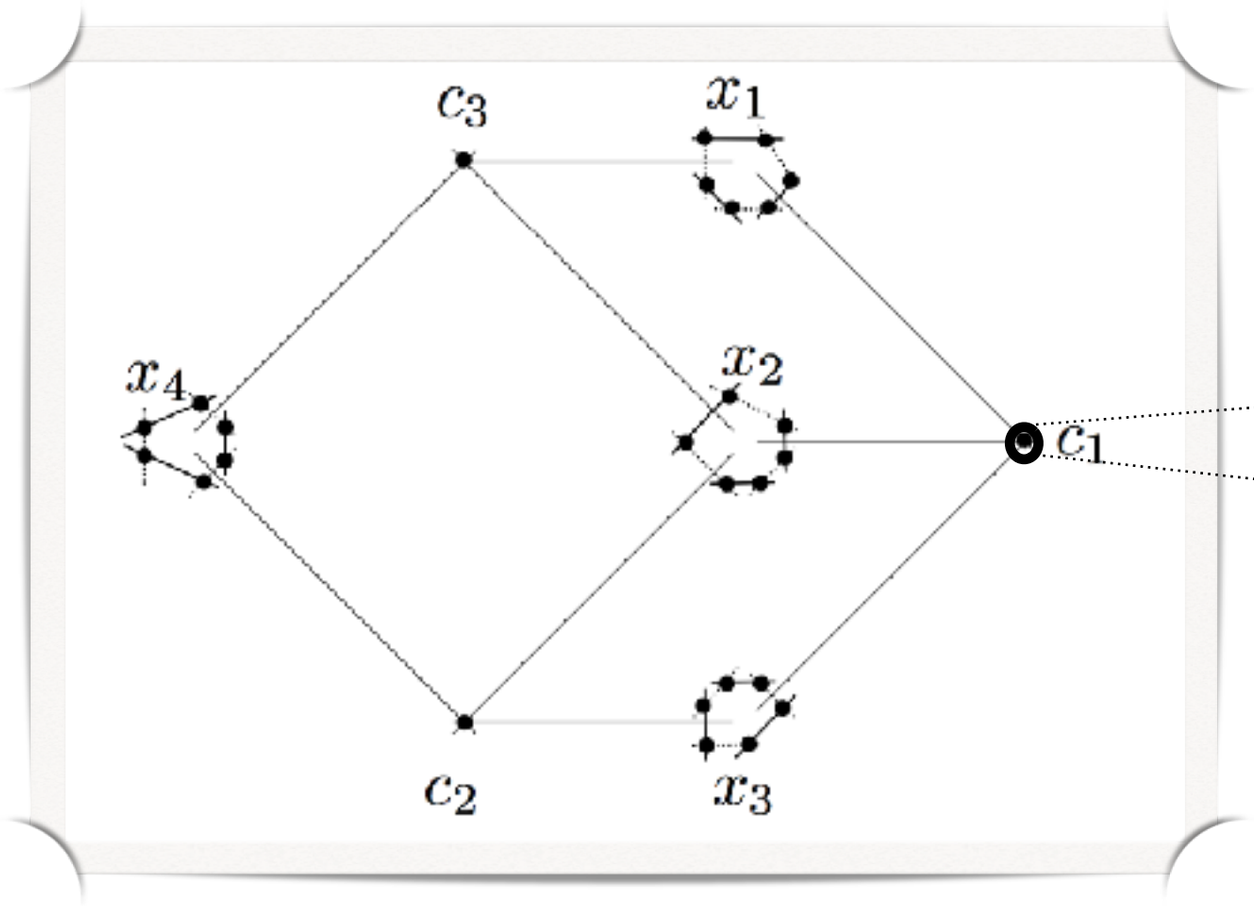
LEMMA 3.3. *Let P be a set of points in the plane, and let $p_i, p_j \in P$. A triangulation Δ contains the edge $\overline{p_i p_j}$, iff there is no edge in Δ that separates $\overline{p_i p_j}$.*

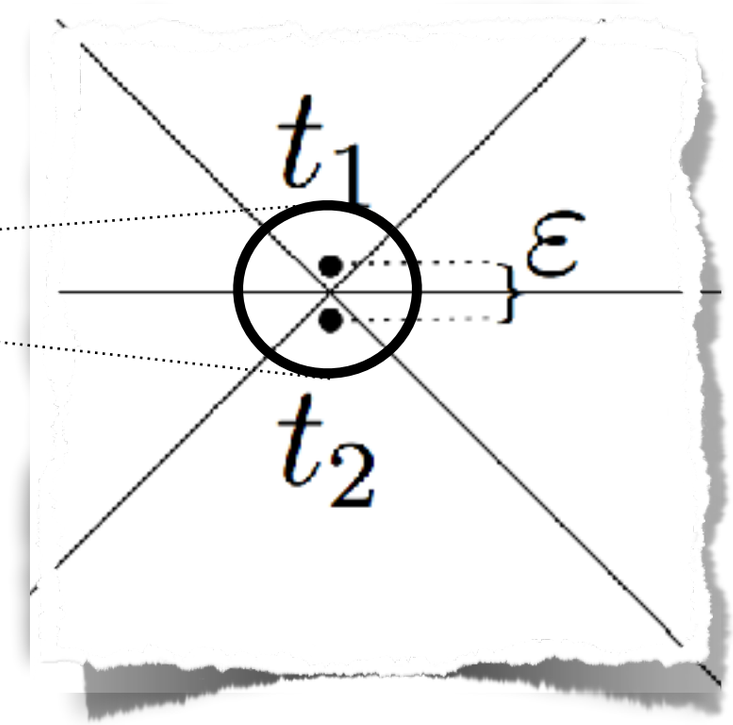
NP-Hardness of MaxMin Triangulations



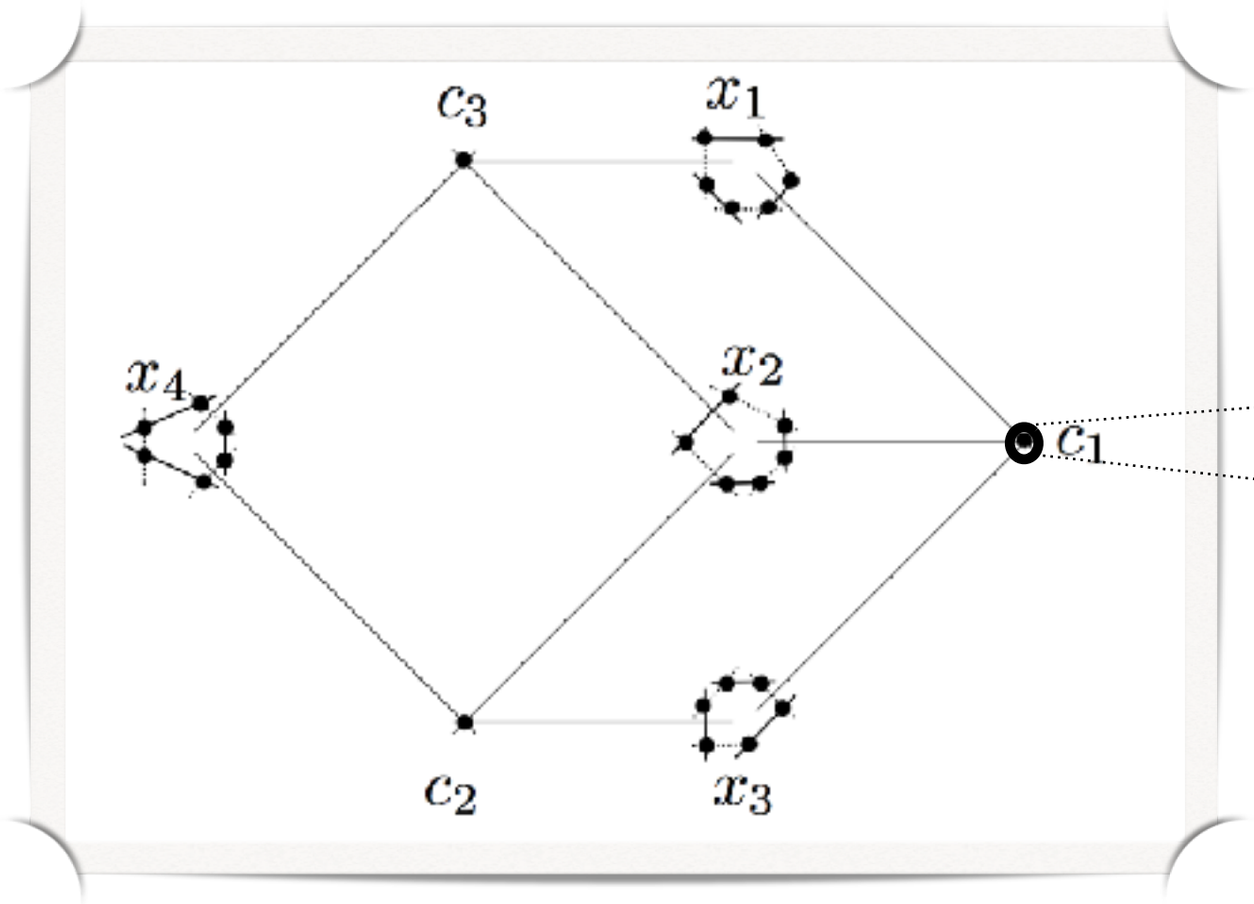
LEMMA 3.3. *Let P be a set of points in the plane, and let $p_i, p_j \in P$. A triangulation Δ contains the edge $\overline{p_i p_j}$, iff there is no edge in Δ that separates $\overline{p_i p_j}$.*

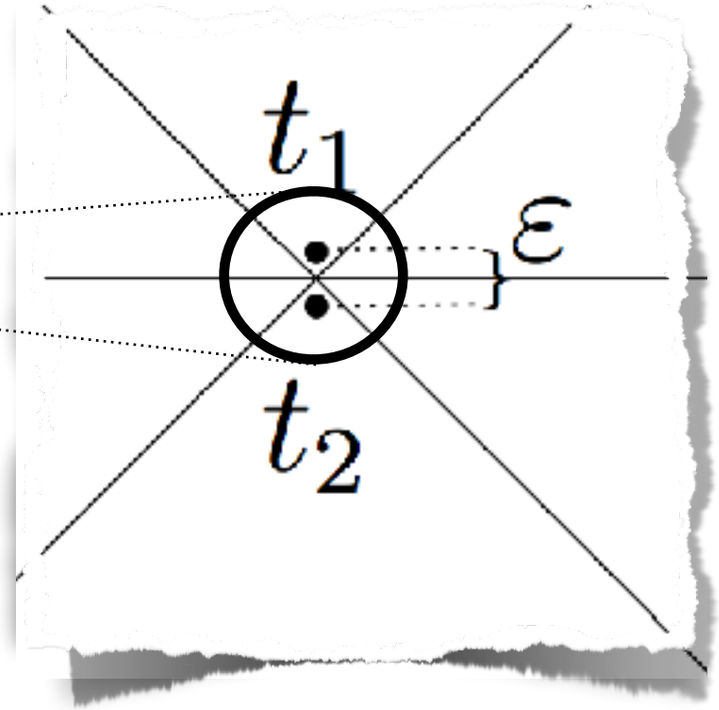
NP-Hardness of MaxMin Triangulations





NP-Hardness of MaxMin Triangulations





COROLLARY 3.4. *Let $p(x)$ be some polynomial. Then the existence of a polynomial-time algorithm that yields a $p(n)$-approximation for MELT implies P=NP.*

5.2: IP Models

Overall Goals

Overall Goals

★ Solve non-trivial instances to provable optimality!

Overall Goals

★ Solve non-trivial instances to provable optimality!

1. I

Overall Goals

★ Solve non-trivial instances to provable optimality!

1. Model as Integer Linear Program (IP)

Overall Goals

★ Solve non-trivial instances to provable optimality!

1. Model as Integer Linear Program (IP)
2. [illegible]
   2.1. [illegible]
3. Solve IP
   3.2. [illegible]

Overall Goals

★ Solve non-trivial instances to provable optimality!

1. Model as Integer Linear Program (IP)
2. [illegible]
   2.1. Turn geometry into IP-formulation
   2.2. Solve IP
   2.3. [illegible]

Overall Goals

★ Solve non-trivial instances to provable optimality!

1. Model as Integer Linear Program (IP)
2. [illegible]
   2.1. Turn geometry into IP-formulation
   2.2. Solve IP
   2.3. Consider solution, reformulate IP for better solution

Overall Goals

★ Solve non-trivial instances to provable optimality!

1. Model as Integer Linear Program (IP)
2. Iterate:
   2.1. Turn geometry into IP-formulation
   2.2. Solve IP
   2.3. Consider solution, reformulate IP for better solution

Overall Goals

★ Solve non-trivial instances to provable optimality!

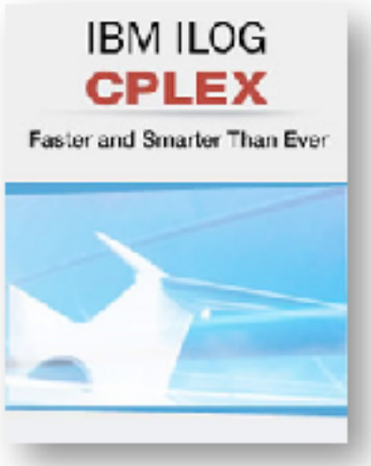
A. Iterate 1.+2. for better results!
  1. Model as Integer Linear Program (IP)
  2. Iterate:
    2.1. Turn geometry into IP-formulation
    2.2. Solve IP
    2.3. Consider solution, reformulate IP for better solution

Overall Goals

★ Solve non-trivial instances to provable optimality!

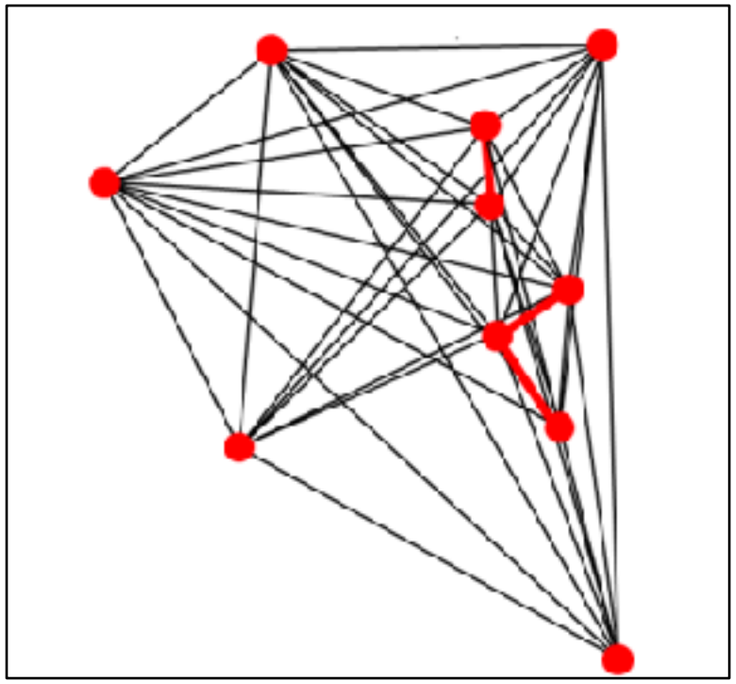
A. Iterate 1.+2. for better results!
  1. Model as Integer Linear Program (IP)
  2. Iterate:
    2.1. Turn geometry into IP-formulation
    2.2. Solve IP
    2.3. Consider solution, reformulate IP for better solution

CGAL **meets**

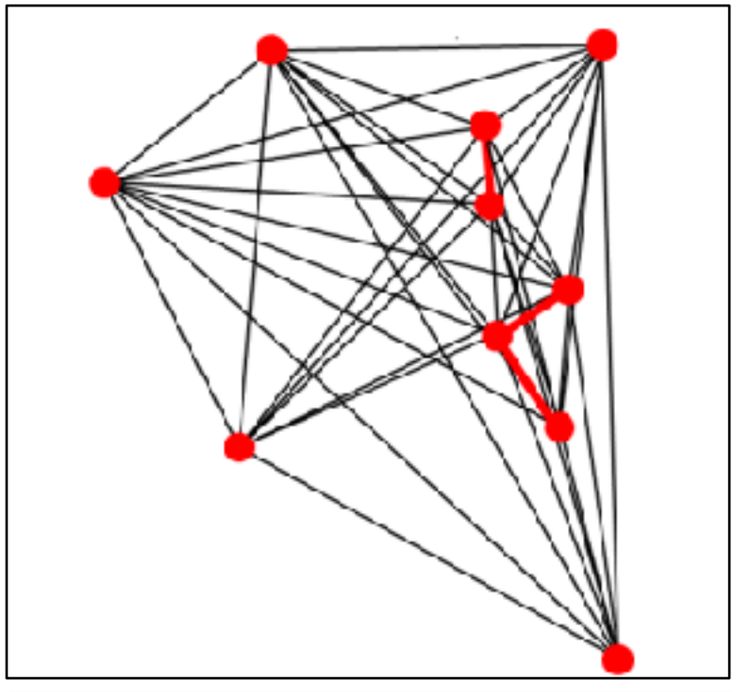


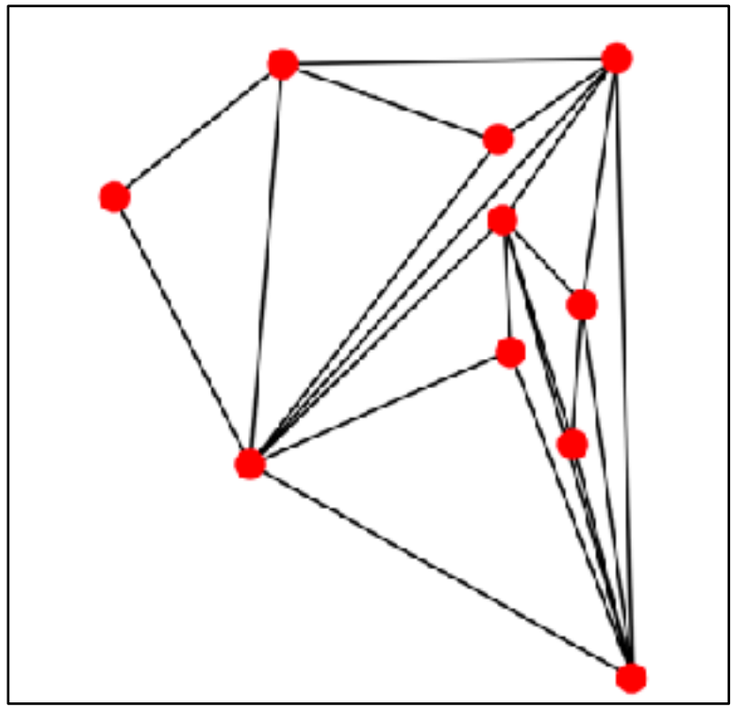
Triangulation by Independent Set

Triangulation by Independent Set

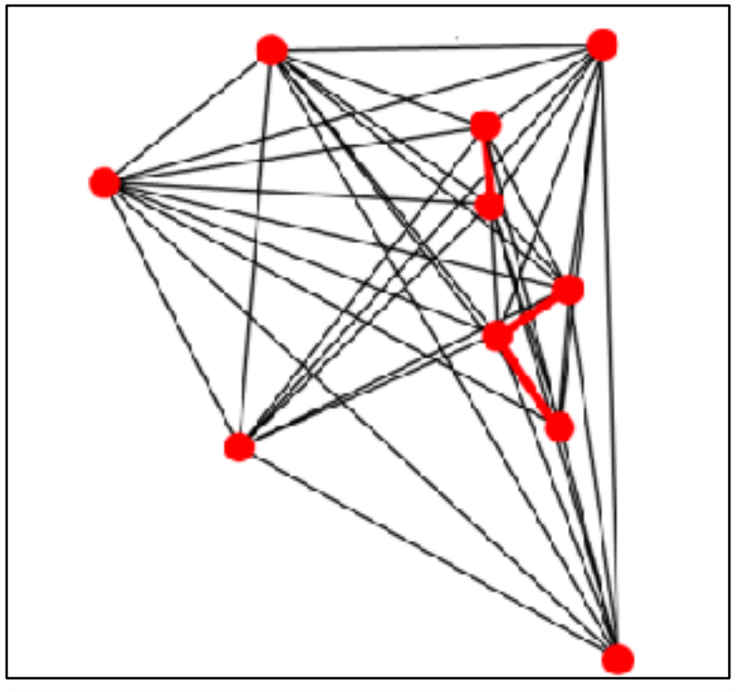


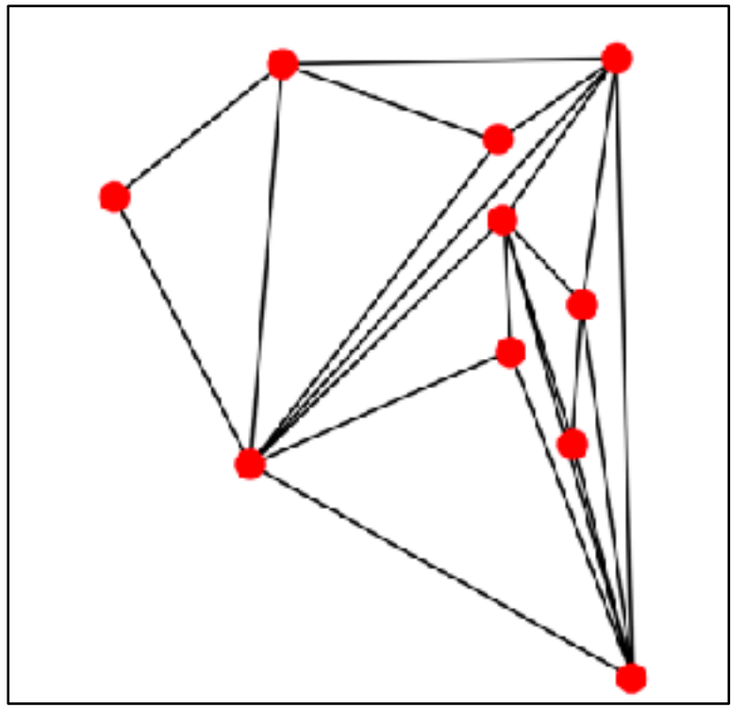
Triangulation by Independent Set





Triangulation by Independent Set





INTEGER PROGRAM 4.1.

(4.1)

$$\max \quad 0$$

(4.2)

$$s.t. \quad x_{e_i} + x_{e_j} \leq 1 \qquad \forall \{e_i, e_j\} \in X_{\geq b}$$

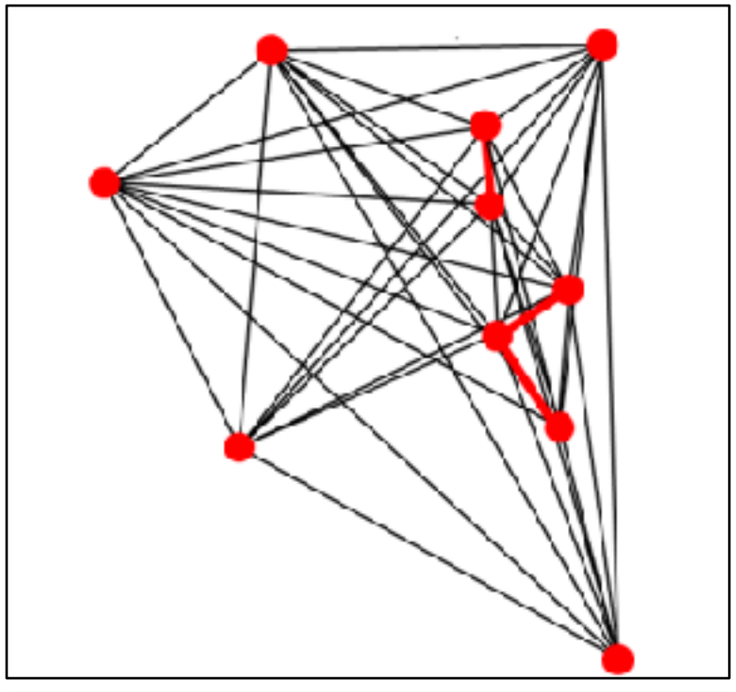
(4.3)

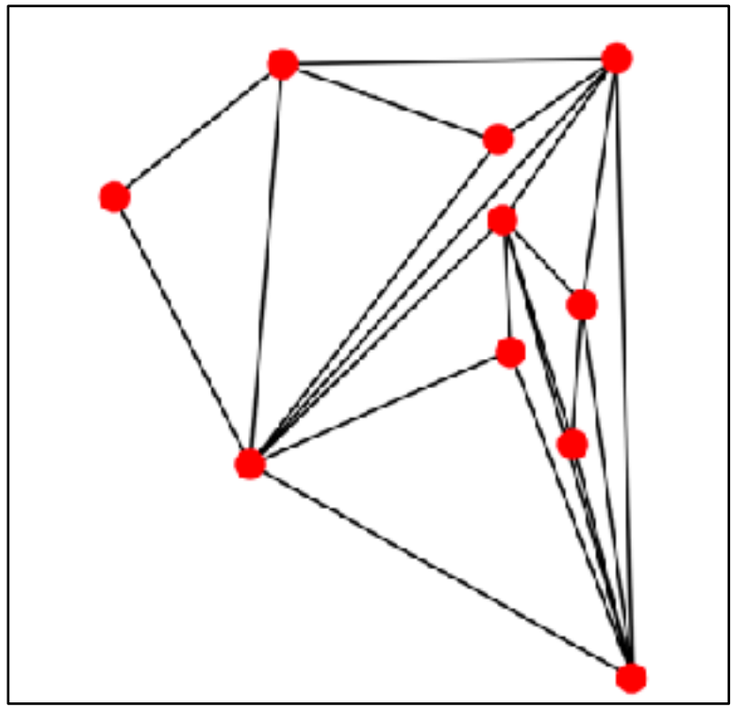
$$\sum_{e_i \in E} x_{e_i} = 3n - 3 - n_c \qquad \forall e_i \in E_{\geq b}$$

(4.4)

$$x_e \in \{0, 1\} \qquad \forall e \in E_{\geq b}$$

Triangulation by Independent Set





INTEGER PROGRAM 4.1.

(4.1)

$$\max \quad 0$$

(4.2)

$$s.t. \quad x_{e_i} + x_{e_j} \leq 1 \qquad \forall \{e_i, e_j\} \in X_{\geq b}$$

(4.3)

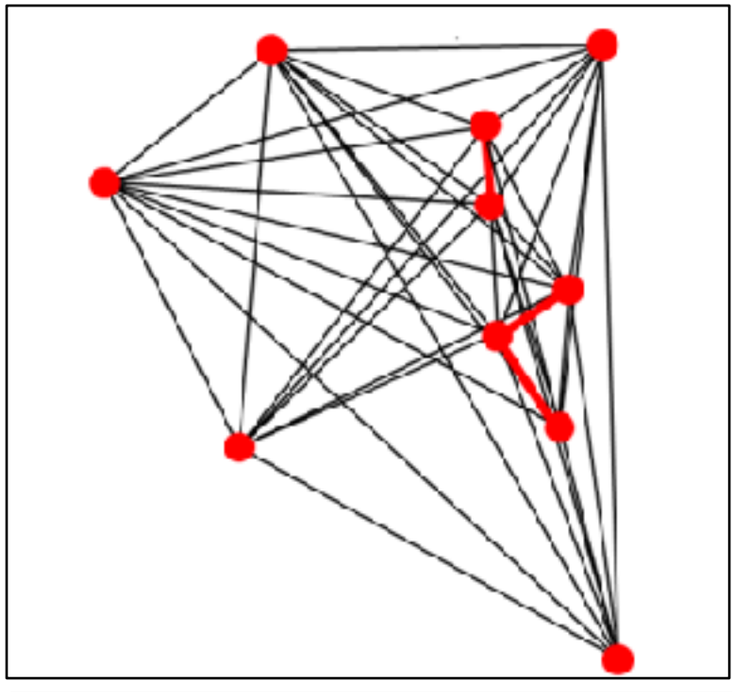
$$\sum_{e_i \in E} x_{e_i} = 3n - 3 - n_c \qquad \forall e_i \in E_{\geq b}$$

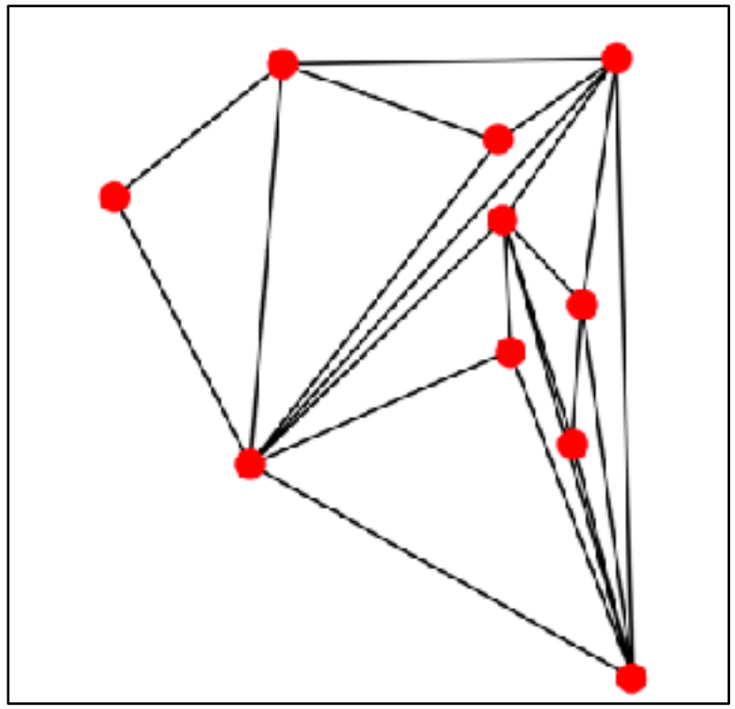
(4.4)

$$x_e \in \{0, 1\} \qquad \forall e \in E_{\geq b}$$

THEOREM 4.2. *We can solve MELT by finding a maximum index b for which IP 4.1 is feasible.*

Triangulation by Independent Set





INTEGER PROGRAM 4.1.

(4.1)

$$\max \quad 0$$

(4.2)

$$s.t. \quad x_{e_i} + x_{e_j} \leq 1 \qquad \forall \{e_i, e_j\} \in X_{\geq b}$$

(4.3)

$$\sum_{e_i \in E} x_{e_i} = 3n - 3 - n_c \qquad \forall e_i \in E_{\geq b}$$

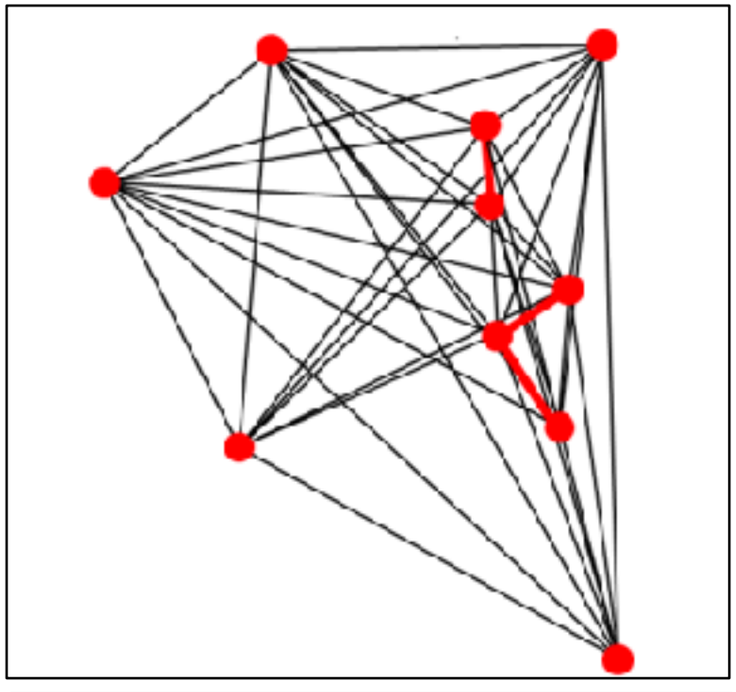
(4.4)

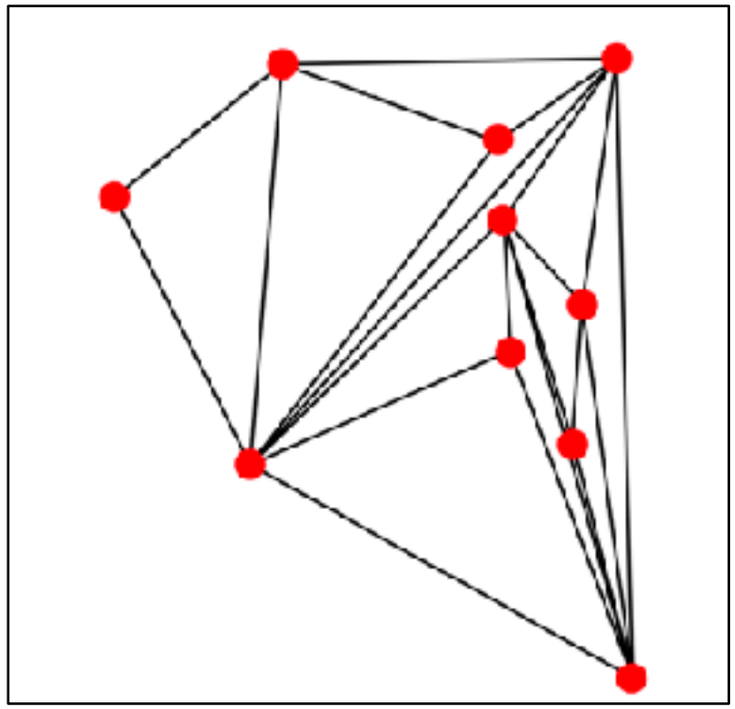
$$x_e \in \{0, 1\} \qquad \forall e \in E_{\geq b}$$

THEOREM 4.2. *We can solve MELT by finding a maximum index b for which IP 4.1 is feasible.*

László Lovász, Katalin Vesztergombi, Uli Wagner, and Emo Welzl, *Convex quadrilaterals and k-sets*, Contemporary Mathematics Series, 342, AMS 2004, 2004, pp. 139–148.

Triangulation by Independent Set





INTEGER PROGRAM 4.1.

(4.1)

$$\max \quad 0$$

(4.2)

$$s.t. \quad x_{e_i} + x_{e_j} \leq 1 \qquad \forall \{e_i, e_j\} \in X_{\geq b}$$

(4.3)

$$\sum_{e_i \in E} x_{e_i} = 3n - 3 - n_c \qquad \forall e_i \in E_{\geq b}$$

(4.4)

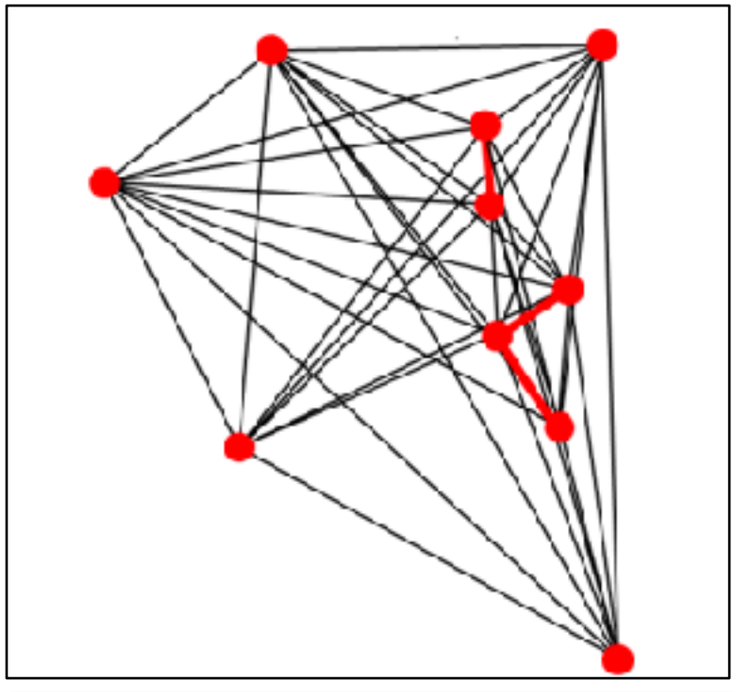
$$x_e \in \{0, 1\} \qquad \forall e \in E_{\geq b}$$

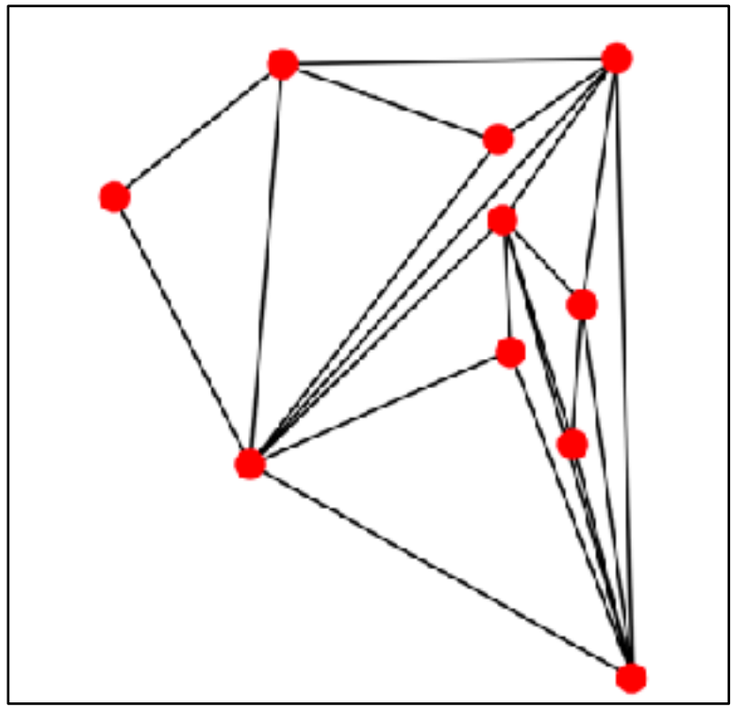
THEOREM 4.2. *We can solve MELT by finding a maximum index b for which IP 4.1 is feasible.*

László Lovász, Katalin Vesztergombi, Uli Wagner, and Emo Welzl, *Convex quadrilaterals and k-sets*, Contemporary Mathematics Series, 342, AMS 2004, 2004, pp. 139–148.

Best-case IP size:

Triangulation by Independent Set





INTEGER PROGRAM 4.1.

(4.1)

$$\max \quad 0$$

(4.2)

$$s.t. \quad x_{e_i} + x_{e_j} \leq 1 \qquad \forall \{e_i, e_j\} \in X_{\geq b}$$

(4.3)

$$\sum_{e_i \in E} x_{e_i} = 3n - 3 - n_c \qquad \forall e_i \in E_{\geq b}$$

(4.4)

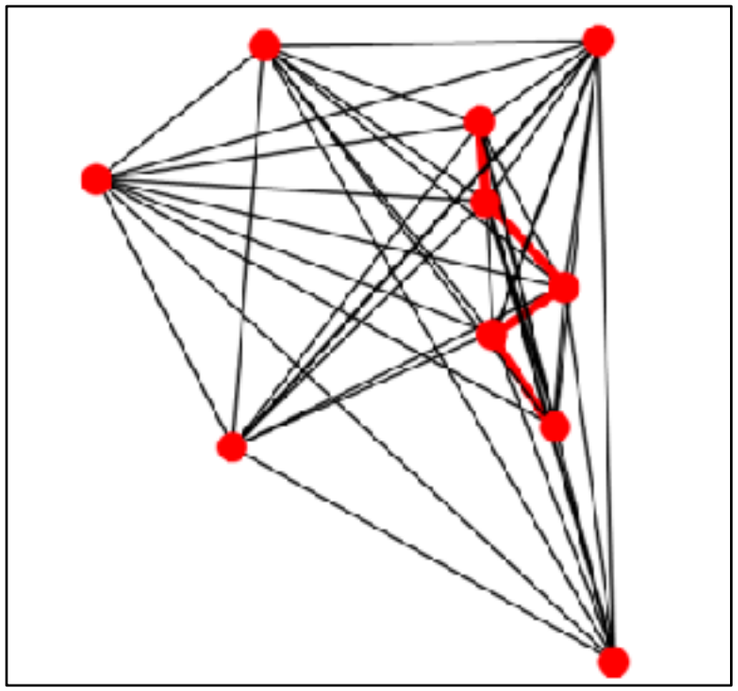
$$x_e \in \{0, 1\} \qquad \forall e \in E_{\geq b}$$

THEOREM 4.2. *We can solve MELT by finding a maximum index b for which IP 4.1 is feasible.*

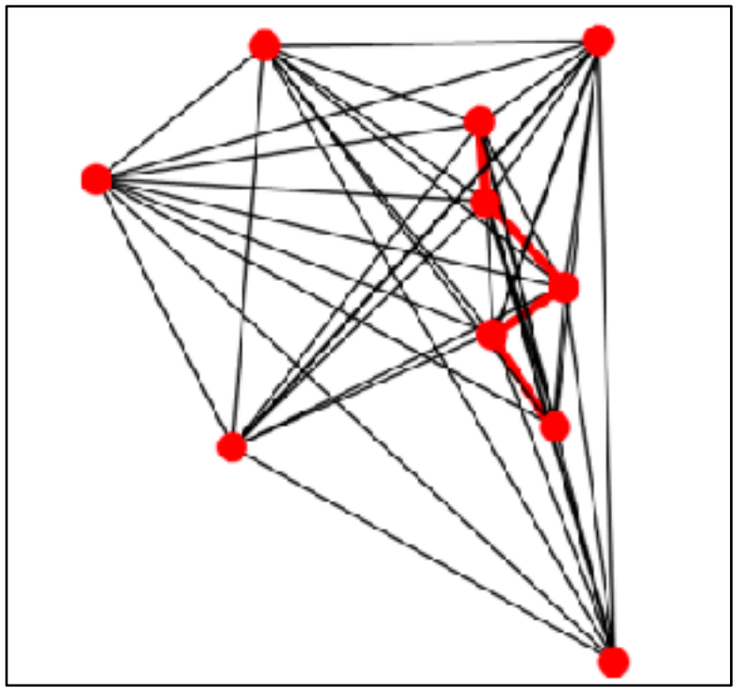
László Lovász, Katalin Vesztergombi, Uli Wagner, and Emo Welzl, *Convex quadrilaterals and k-sets*, Contemporary Mathematics Series, 342, AMS 2004, 2004, pp. 139–148.

Best-case IP size: $\Omega(n^4)$

Triangulation by Hitting Set

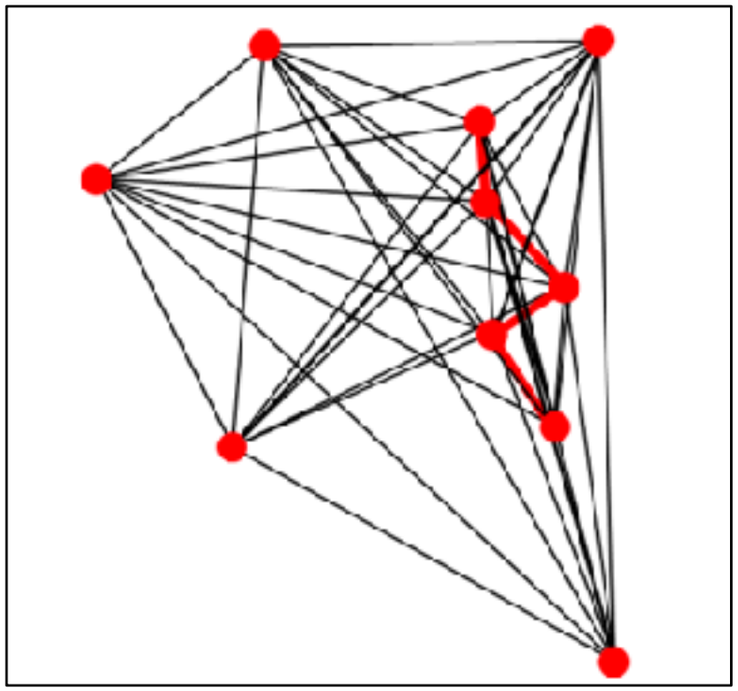


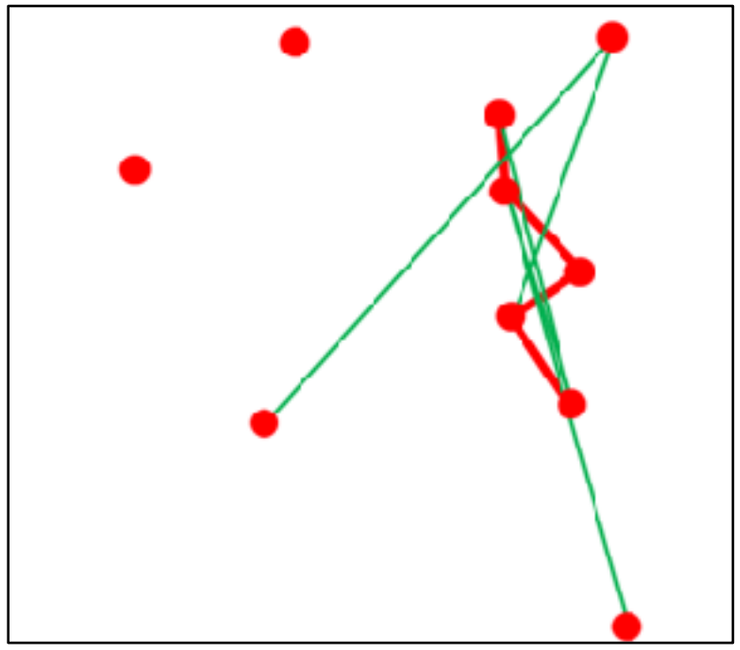
Triangulation by Hitting Set



LEMMA 3.3. *Let P be a set of points in the plane, and let $p_i, p_j \in P$. A triangulation Δ contains the edge $\overline{p_i p_j}$, iff there is no edge in Δ that separates $\overline{p_i p_j}$.*

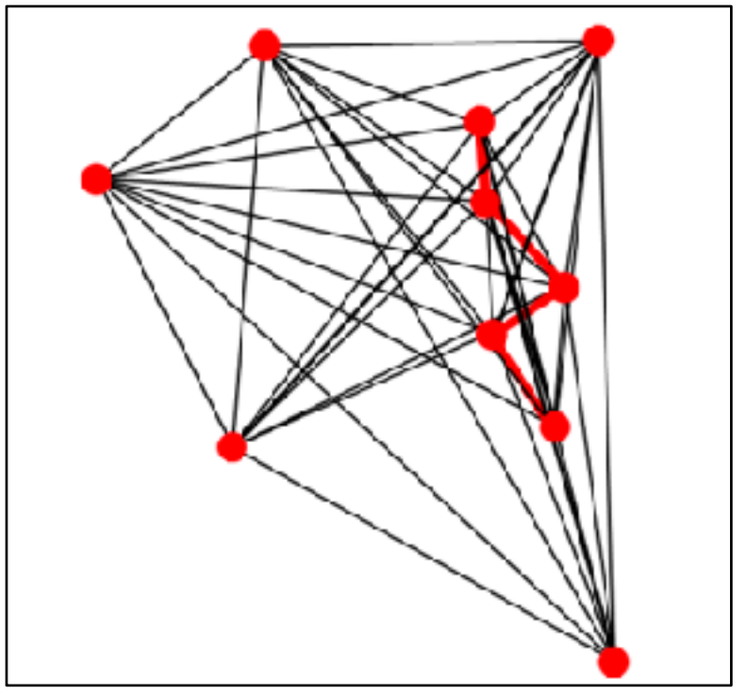
Triangulation by Hitting Set

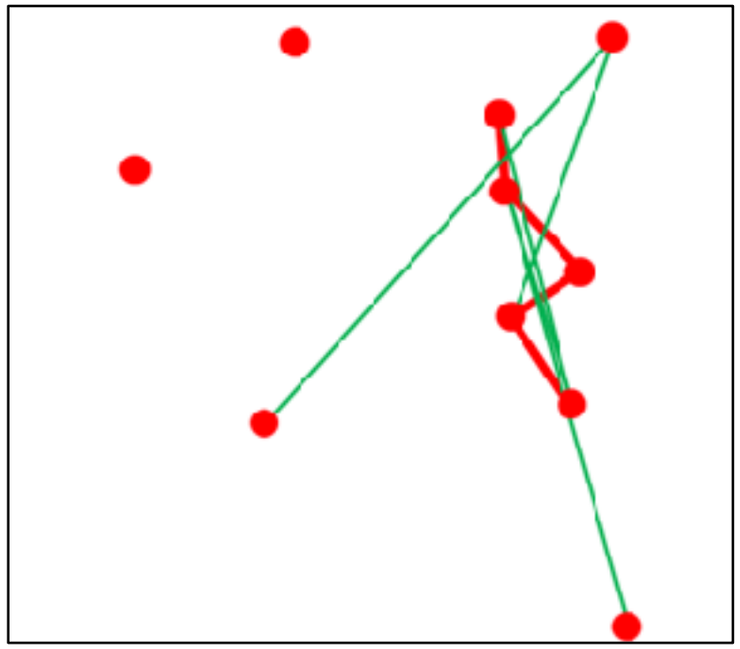




LEMMA 3.3. *Let P be a set of points in the plane, and let $p_i, p_j \in P$. A triangulation Δ contains the edge $\overline{p_i p_j}$, iff there is no edge in Δ that separates $\overline{p_i p_j}$.*

Triangulation by Hitting Set





LEMMA 3.3. *Let P be a set of points in the plane, and let $p_i, p_j \in P$. A triangulation Δ contains the edge $\overline{p_i p_j}$, iff there is no edge in Δ that separates $\overline{p_i p_j}$.*

INTEGER PROGRAM 4.3.

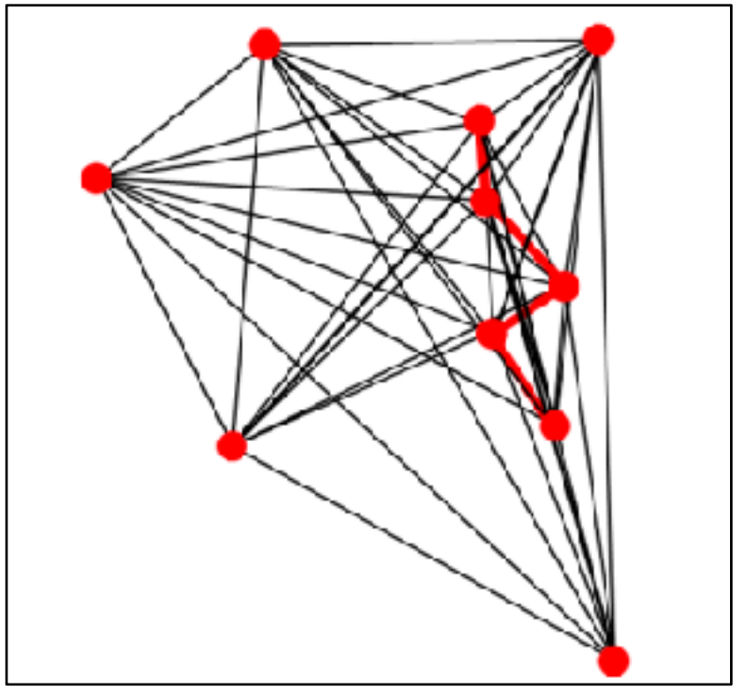
$$\max \quad 0 \tag{4.5}$$

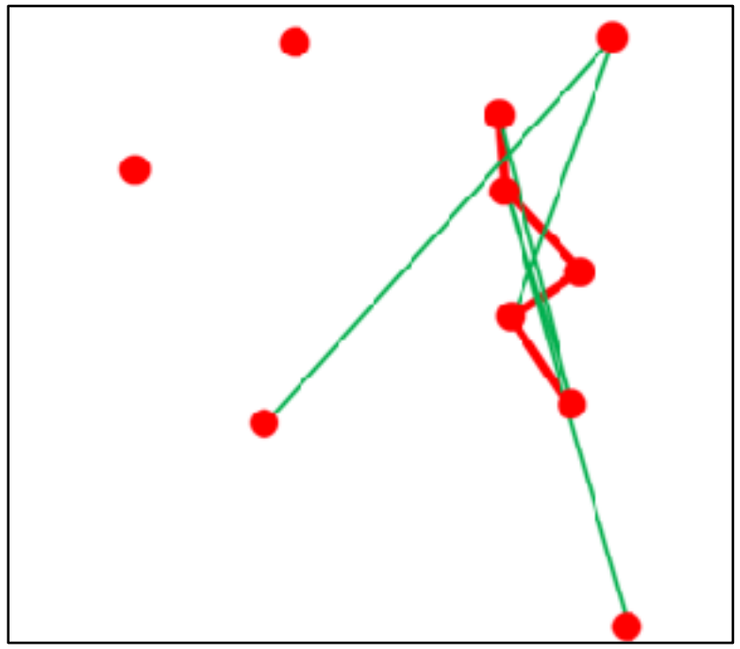
$$\text{s.t.} \quad x_{e_i} + x_{e_j} \leq 1 \qquad \forall \{e_i, e_j\} \in X^*_{\geq b} \tag{4.6}$$

$$\sum_{e_i \in E^*_{\geq b}(e_j)} x_{e_i} \geq 1 \qquad \forall e_j \in E_{<b} \tag{4.7}$$

$$x_{e_i} \in \{0,1\} \qquad \forall e_i \in E^*_{\geq b} \tag{4.8}$$

Triangulation by Hitting Set





LEMMA 3.3. *Let P be a set of points in the plane, and let $p_i, p_j \in P$. A triangulation Δ contains the edge $\overline{p_i p_j}$, iff there is no edge in Δ that separates $\overline{p_i p_j}$.*

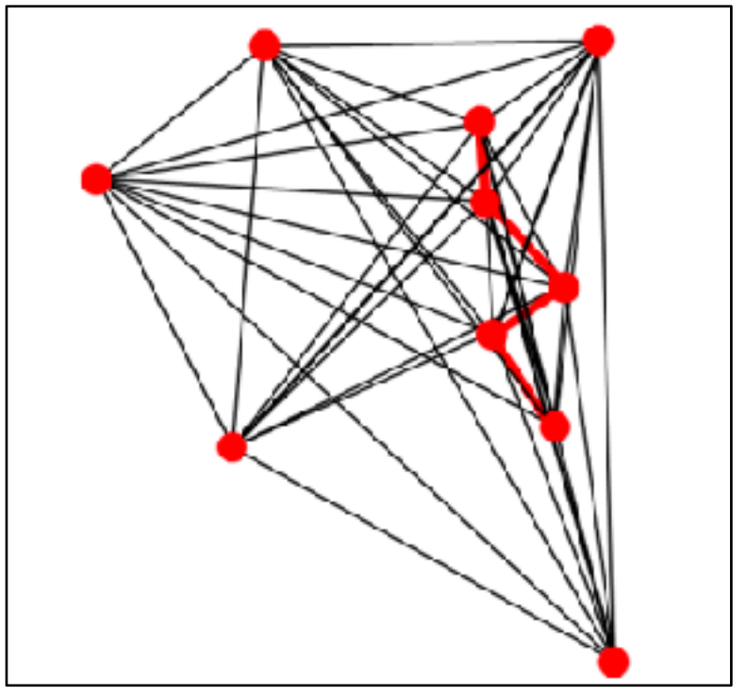
INTEGER PROGRAM 4.3.

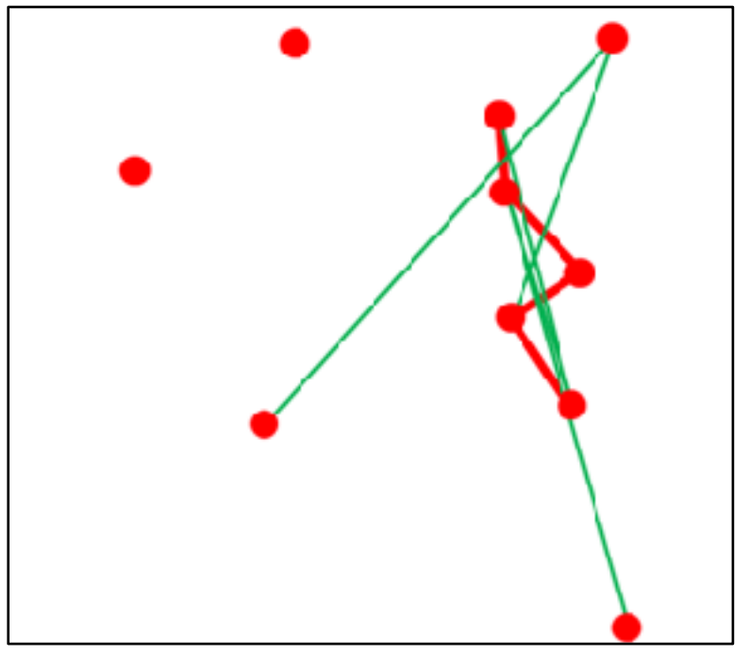
$$\max \quad 0 \tag{4.5}$$

$$\begin{aligned}
&(4.6) && s.t. && x_{e_i} + x_{e_j} && \leq 1 && \forall \{e_i, e_j\} \in X^*_{\geq b} \\
&(4.7) && && \sum_{e_i \in E^*_{\geq b}(e_j)} x_{e_i} && \geq 1 && \forall e_j \in E_{<b} \\
&(4.8) && && x_{e_i} && \in \{0,1\} && \forall e_i \in E^*_{\geq b}
\end{aligned}$$

THEOREM 4.4. *We can solve MELT by finding a maximum value b for which IP 4.3 is feasible.*

Triangulation by Hitting Set





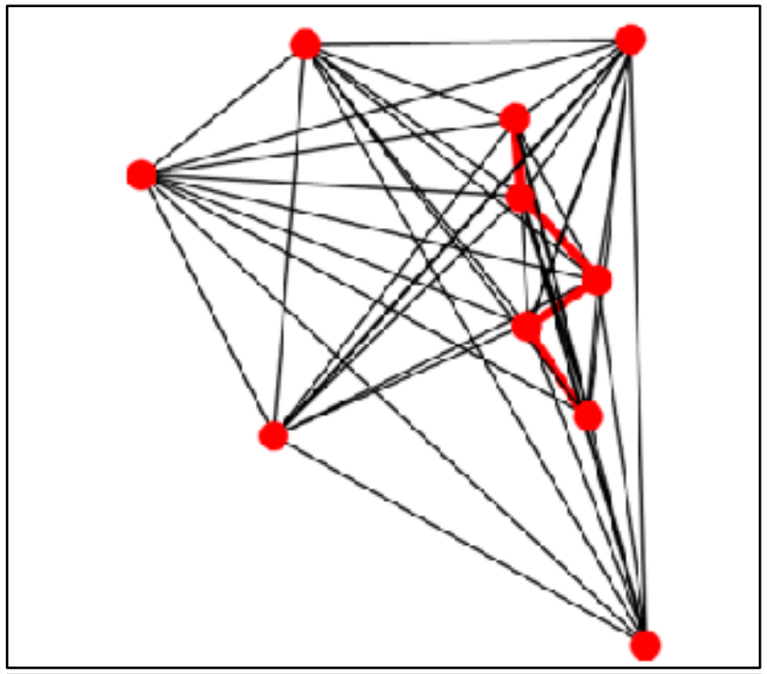
LEMMA 3.3. *Let P be a set of points in the plane, and let $p_i, p_j \in P$. A triangulation Δ contains the edge $\overline{p_i p_j}$, iff there is no edge in Δ that separates $\overline{p_i p_j}$.*

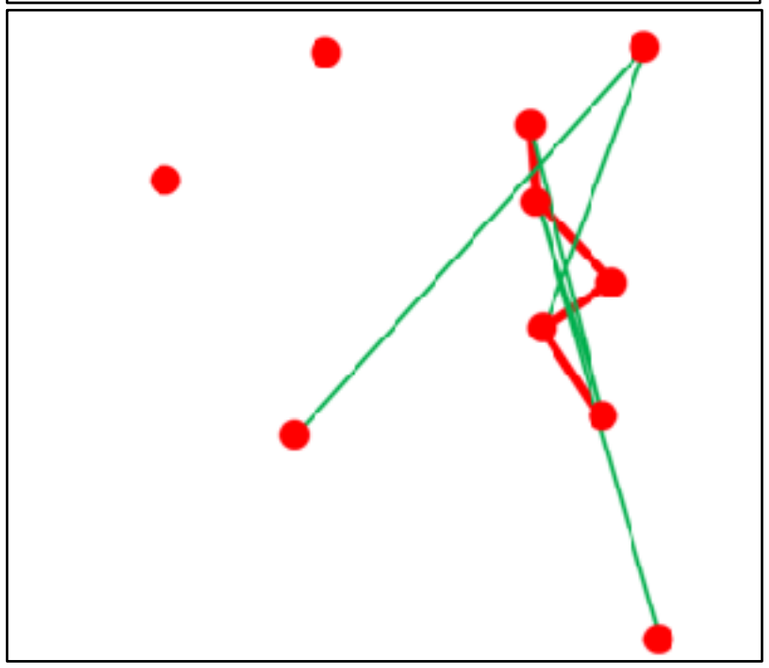
INTEGER PROGRAM 4.3.

$$
\begin{array}{llcll}
(4.5) & \max & 0 & & \\
(4.6) & s.t. & x_{e_i} + x_{e_j} & \leq & 1 \qquad \forall \{e_i, e_j\} \in X^*_{\geq b} \\
(4.7) & & \sum_{e_i \in E^*_{\geq b}(e_j)} x_{e_i} & \geq & 1 \qquad \forall e_j \in E_{<b} \\
(4.8) & & x_{e_i} & \in & \{0,1\} \qquad \forall e_i \in E^*_{\geq b}
\end{array}
$$

THEOREM 4.4. *We can solve MELT by finding a maximum value b for which IP 4.3 is feasible.*

Triangulation by Hitting Set





LEMMA 3.3. *Let P be a set of points in the plane, and let $p_i, p_j \in P$. A triangulation Δ contains the edge $\overline{p_i p_j}$, iff there is no edge in Δ that separates $\overline{p_i p_j}$.*

INTEGER PROGRAM 4.5.

(4.9) $\max \quad 0$

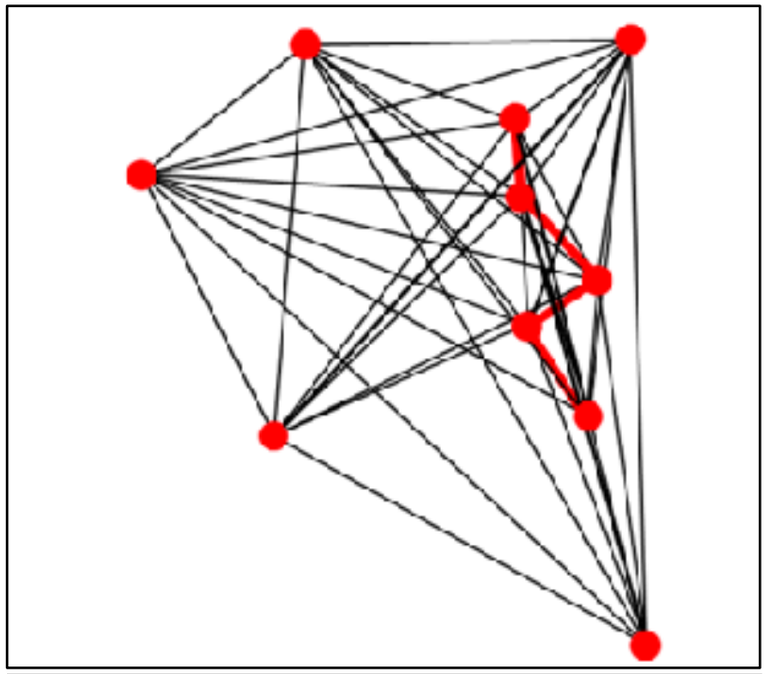
(4.10) $$s.t. \quad k_e \cdot x_e + \sum_{e_i \in E^*_{\geq b}(e)} x_{e_i} \leq k_e \qquad \forall e \in E^*_{\geq b}$$

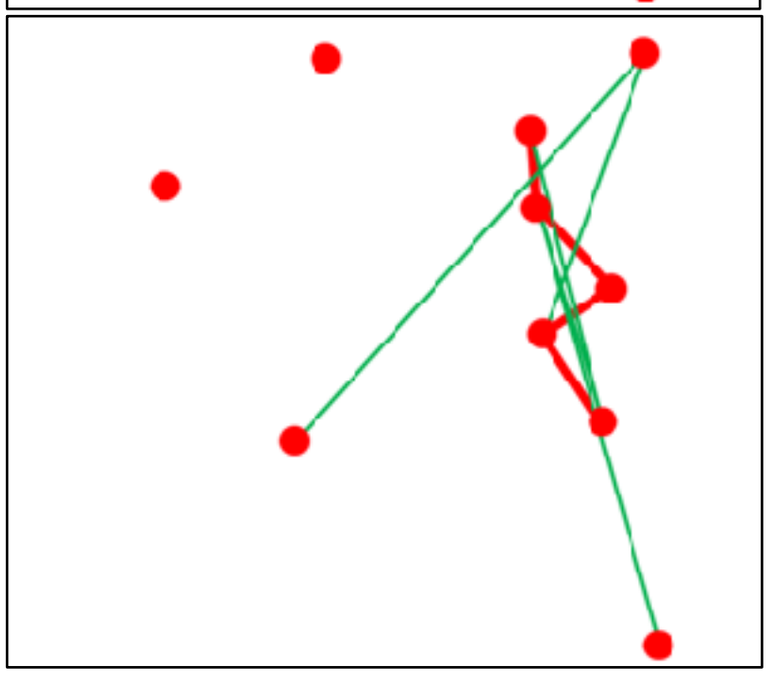
(4.11) $$\sum_{e_i \in E^*_{\geq b}(e)} x_{e_i} \geq 1 \qquad \forall e \in E_{<a}$$

(4.12) $$x_{e_i} \in \{0,1\} \qquad \forall e_i \in E^*_{\geq b}$$

THEOREM 4.6. *We can solve MELT by finding a maximum index b for which IP 4.5 is feasible.*

Finding the Critical Index





LEMMA 3.3. *Let P be a set of points in the plane, and let $p_i, p_j \in P$. A triangulation Δ contains the edge $\overline{p_i p_j}$, iff there is no edge in Δ that separates $\overline{p_i p_j}$.*

INTEGER PROGRAM 4.7.

$$\max \quad b \tag{4.13}$$

$$\text{s.t.} \quad x_{e_i} \cdot i + (1 - x_{e_i}) \cdot c \geq b \qquad \forall i \in [a, c[\tag{4.14}$$

$$k_e \cdot x_e + \sum_{e_i \in E^*_{\geq a}(e)} x_{e_i} \leq k_e \qquad \forall e \in E^*_{\geq a} \tag{4.15}$$

$$x_e + \sum_{e_i \in E^*_{\geq a}(e)} x_{e_i} \geq 1 \qquad \forall e \in E_{[a,c[} \tag{4.16}$$

$$\sum_{e_i \in E^*_{\geq L}(e)} x_{e_i} \geq 1 \qquad \forall e \in E_{<a} \tag{4.17}$$

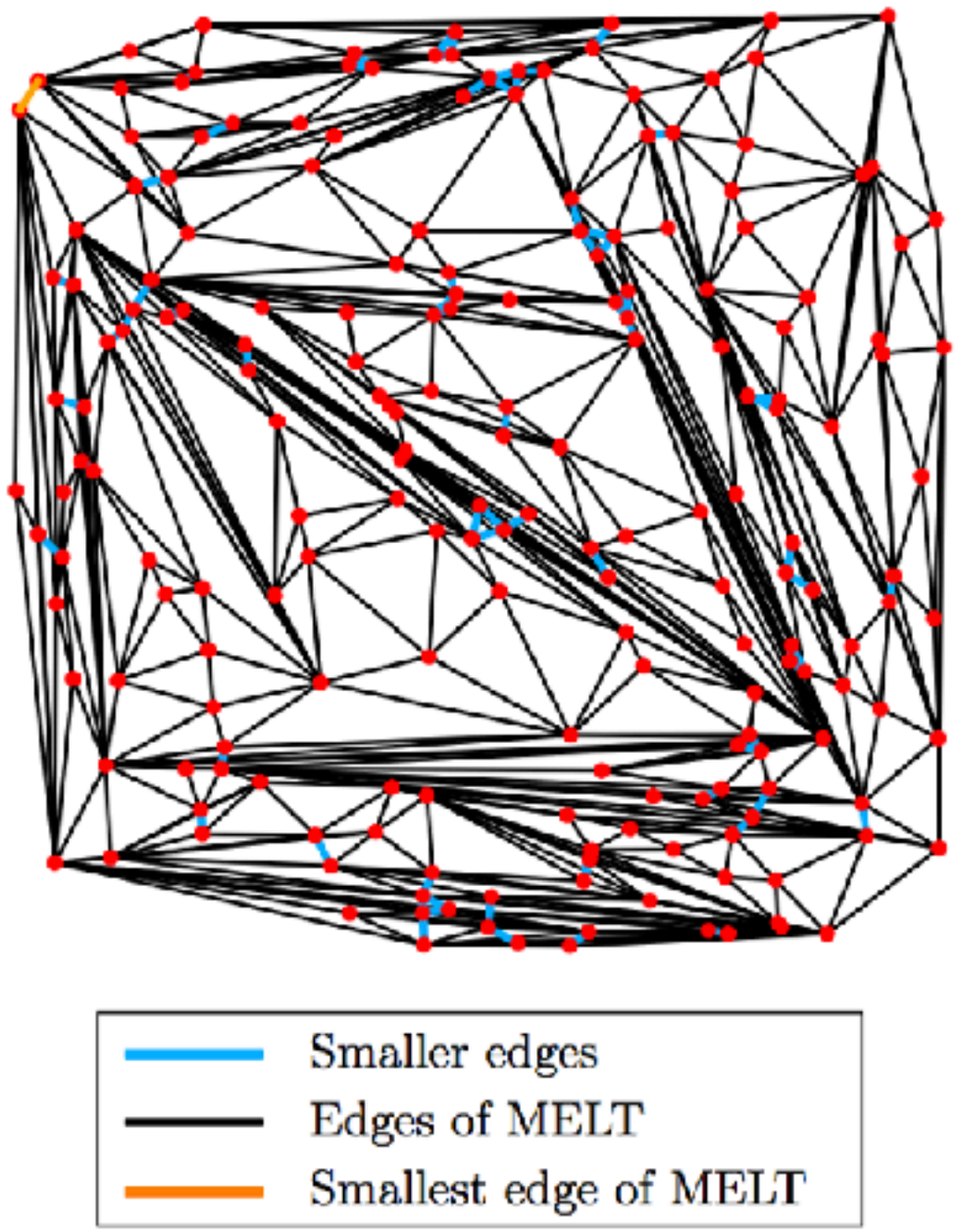
$$x_{e_i} \in \{0, 1\} \qquad \forall e_i \in E^*_{\geq a} \tag{4.18}$$

THEOREM 4.8. *If we know that the critical index $b \in [a, c[$, we can solve MELT by solving IP 4.7.*

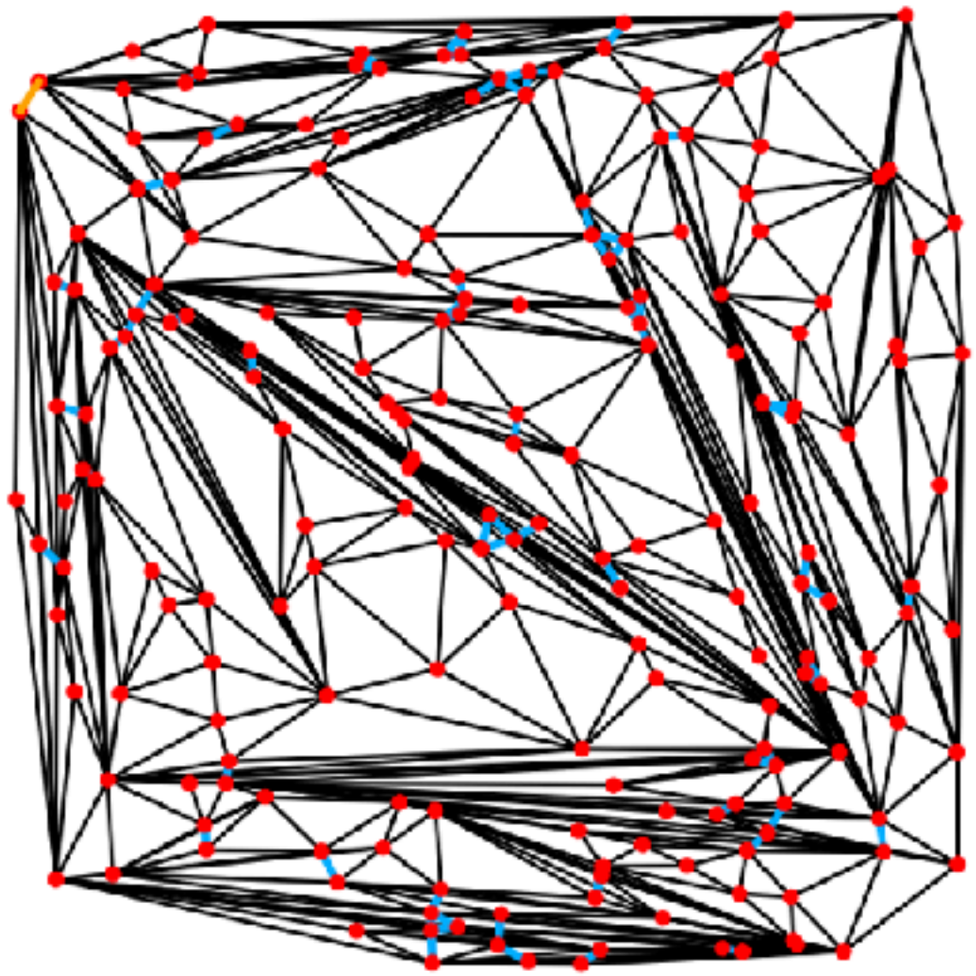
5.3: Experimental Results

Looking at the Critical Index

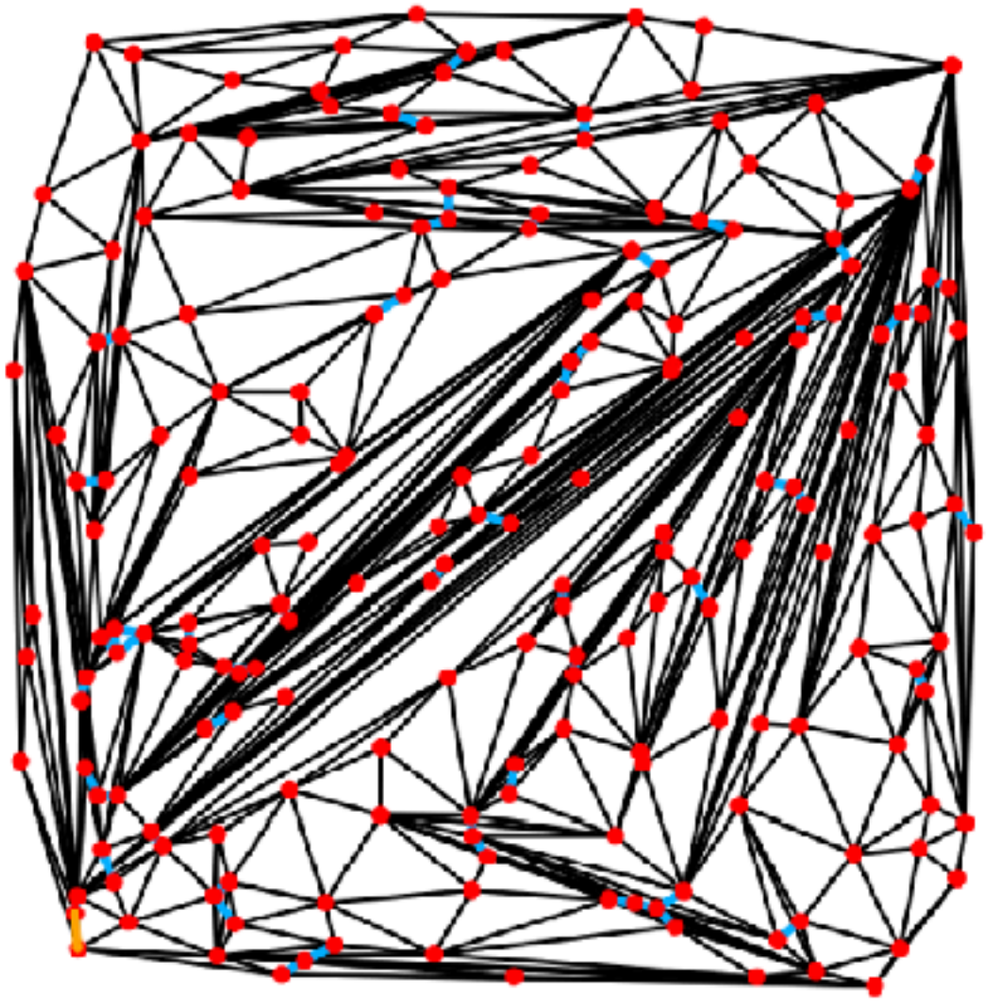
Looking at the Critical Index



Looking at the Critical Index



Smaller edges
Edges of MELT
Smallest edge of MELT



Smaller edges
Edges of MELT
Smallest edge of MELT

Upper Bounds in Comparison

Upper Bounds in Comparison

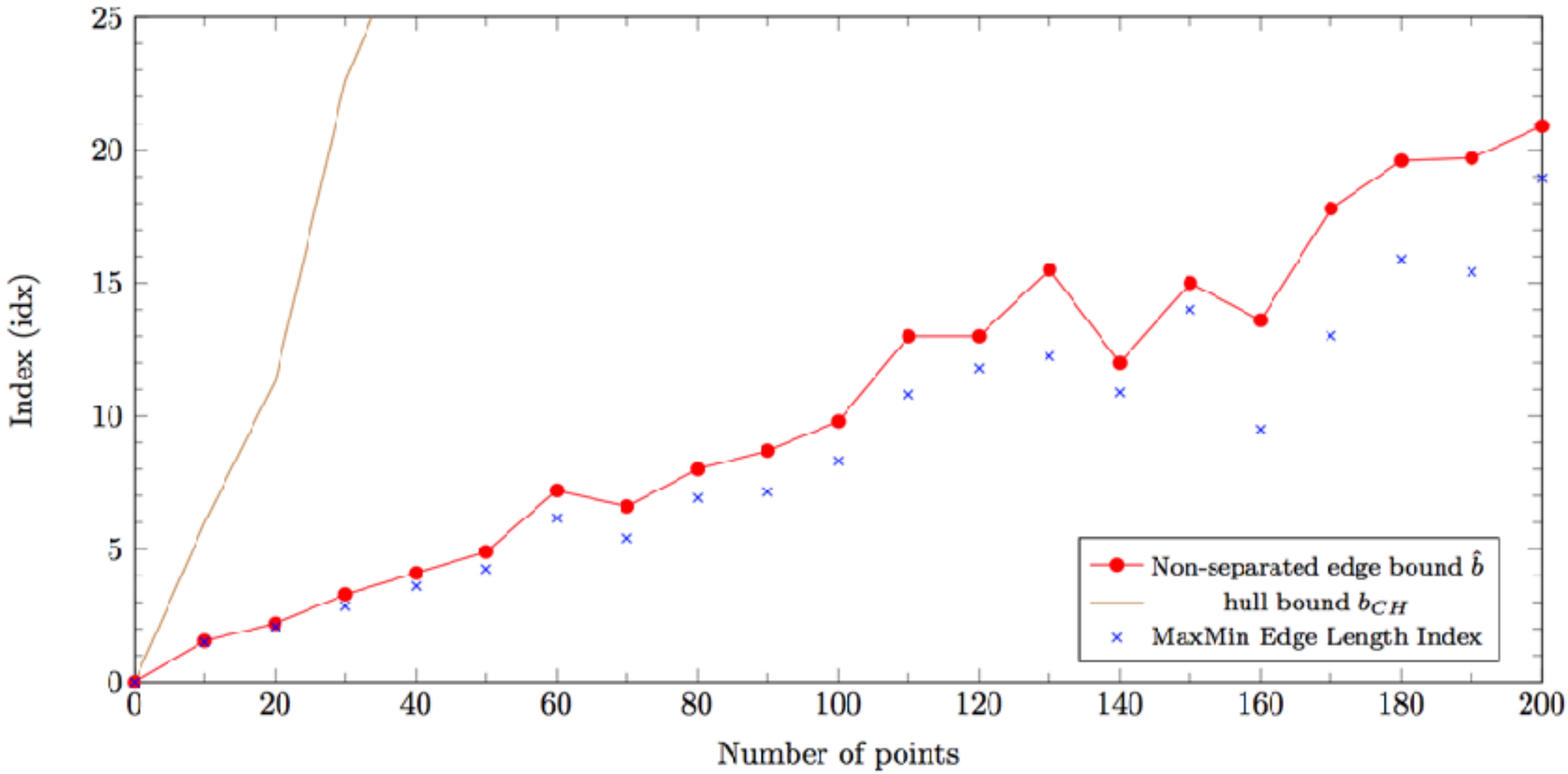


Figure 12: Average values for hull bound, non-separated edge bound, and optimal index for random instances.

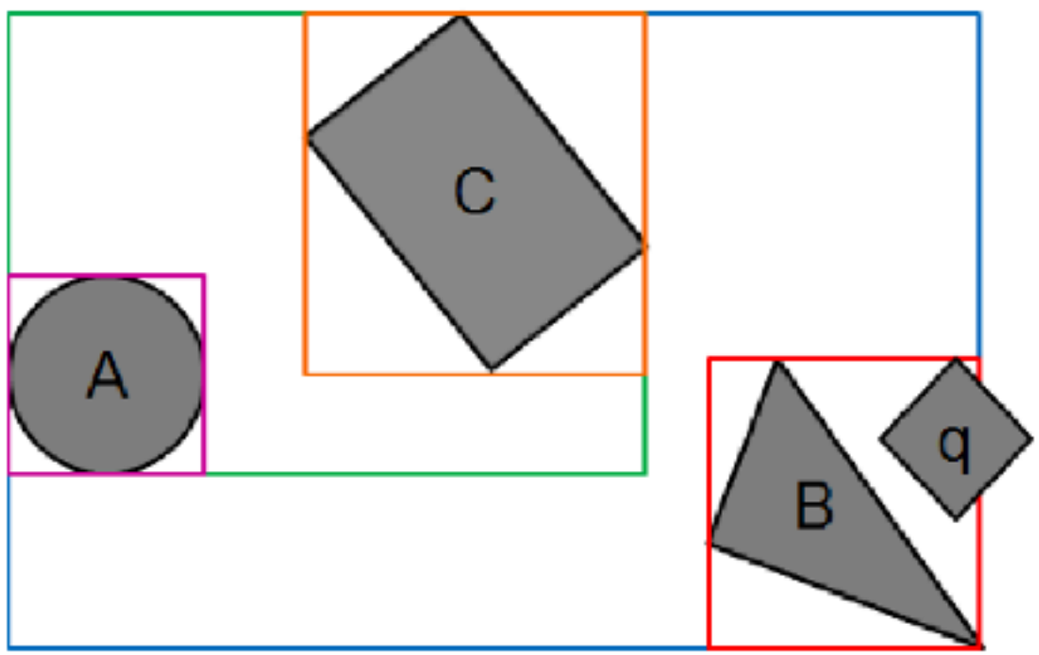
Checking Intersections: AABB Trees

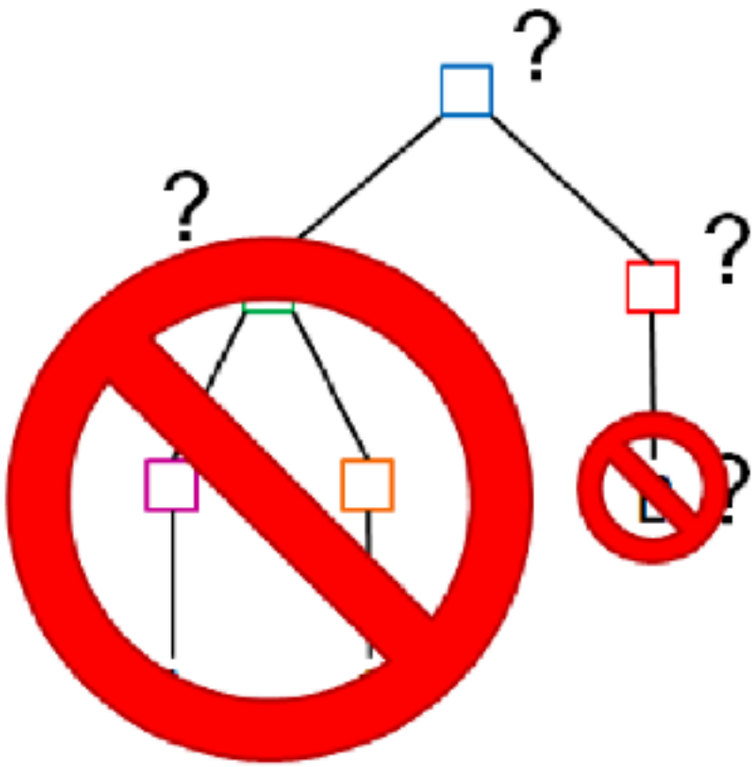
Checking Intersections: AABB Trees

Axis-aligned bounding box tree:
Data structure for speeding up intersection queries

Checking Intersections: AABB Trees

Axis-aligned bounding box tree:
Data structure for speeding up intersection queries





Using AABB Trees

Using AABB Trees

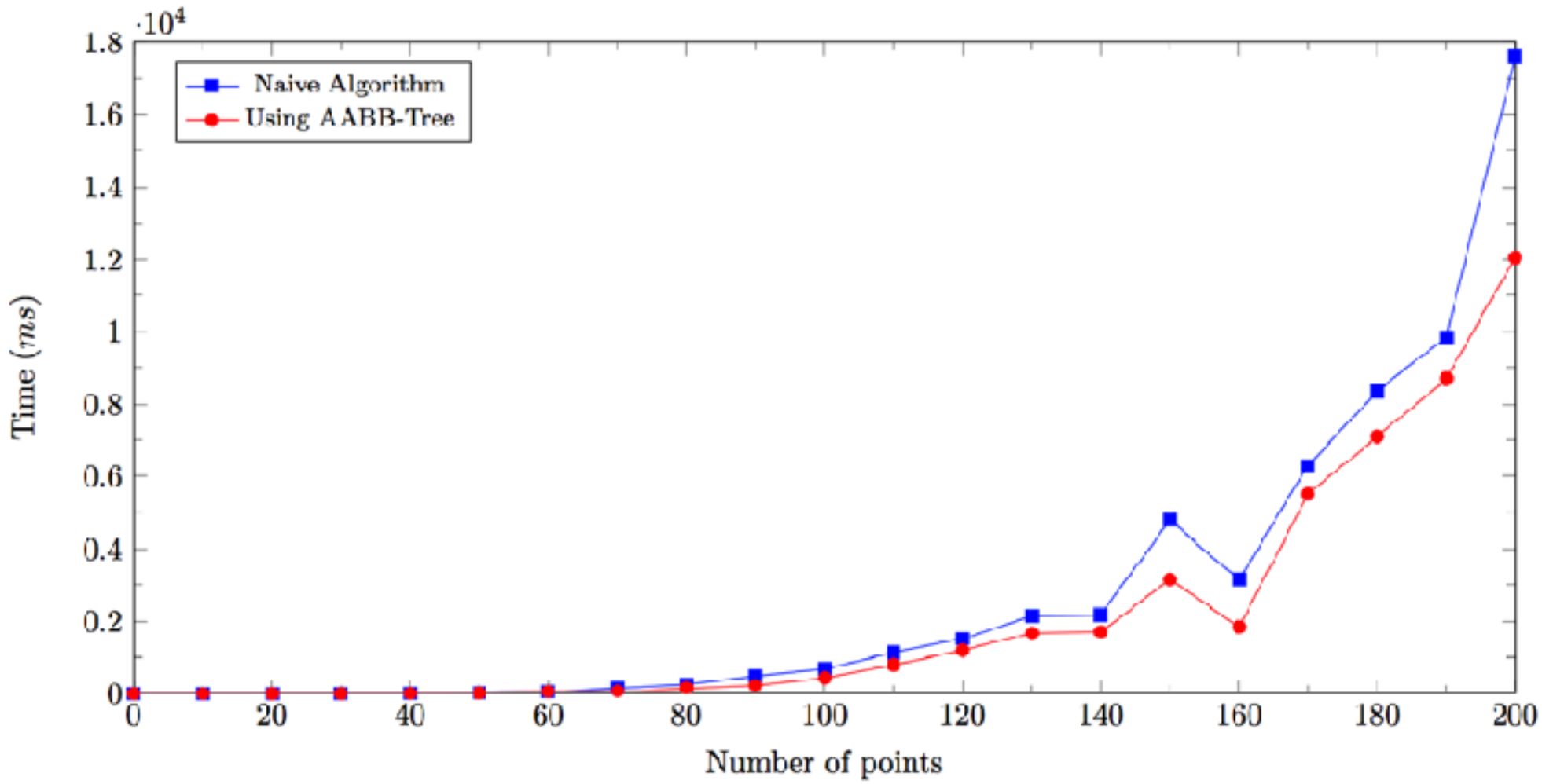


Figure 11: Average time for finding intersections in random instances.

Comparing IP 4.3 and IP 4.5 (1)

Comparing IP 4.3 and IP 4.5 (1)

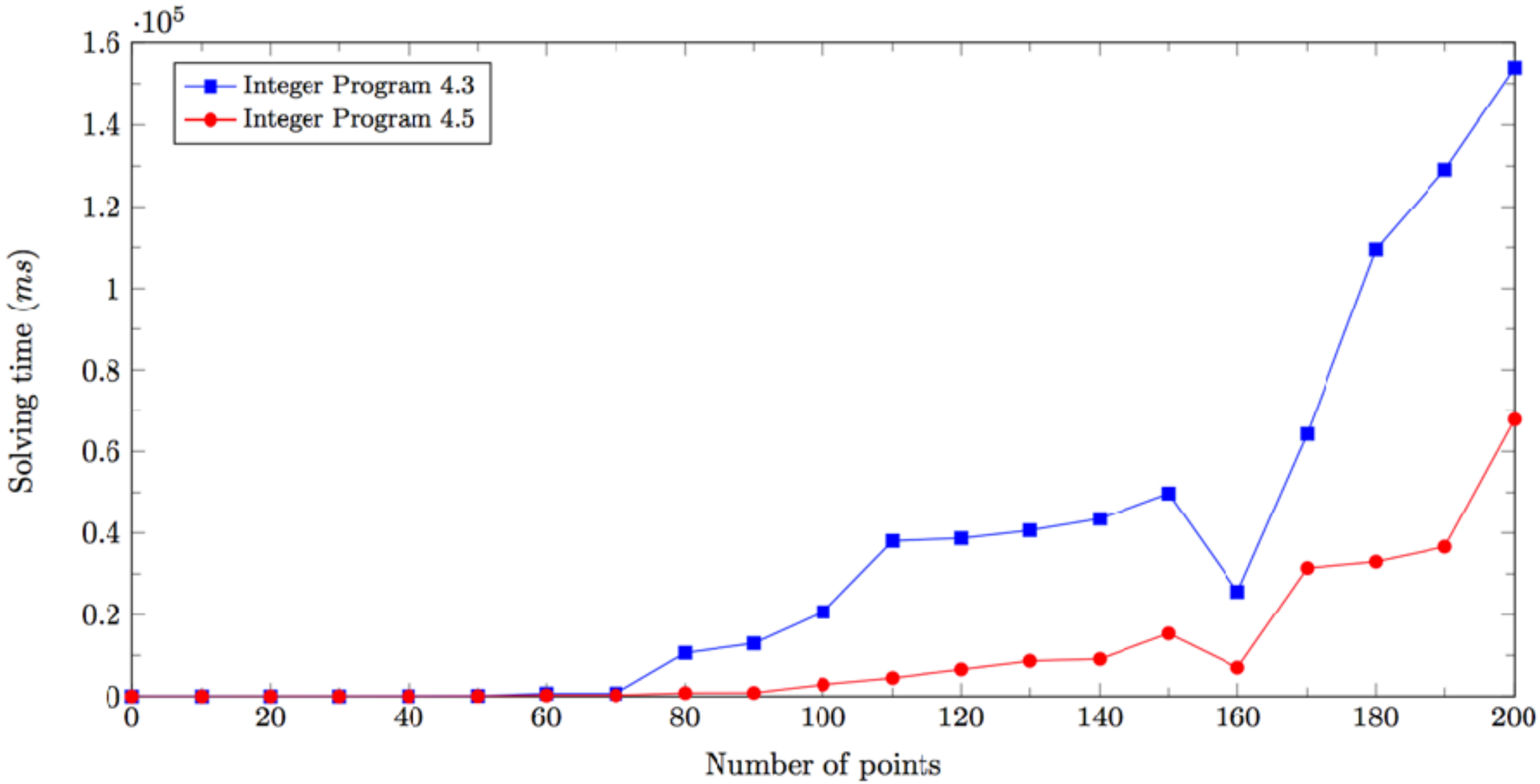


Figure 6: Average solution time of instances: solving time of the IPs.

Comparing IP 4.3 and IP 4.5 (2)

Comparing IP 4.3 and IP 4.5 (2)

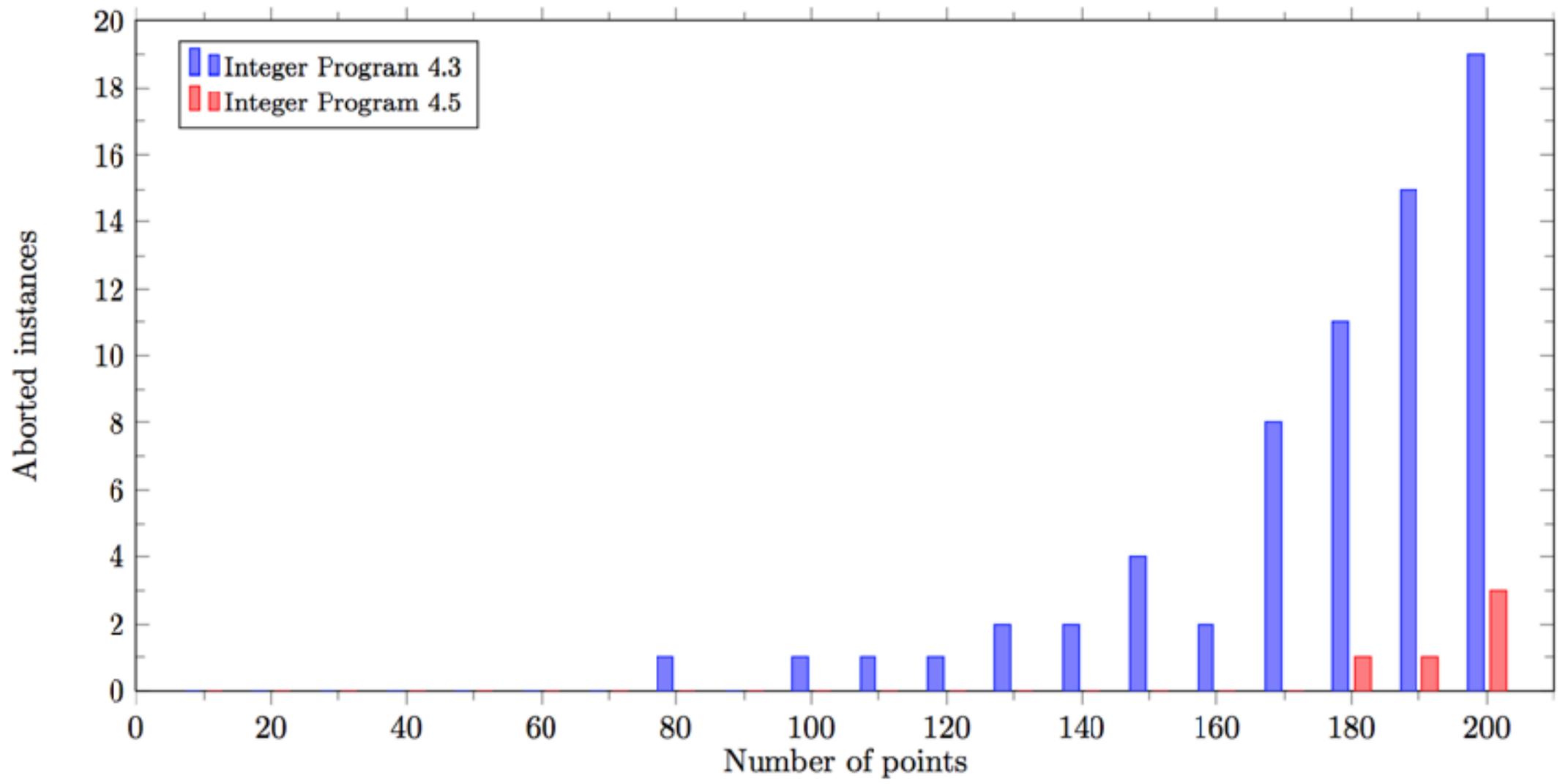


Figure 7: Aborted instances of the IPs, out of 100 instances.

Putting it together

Putting it together

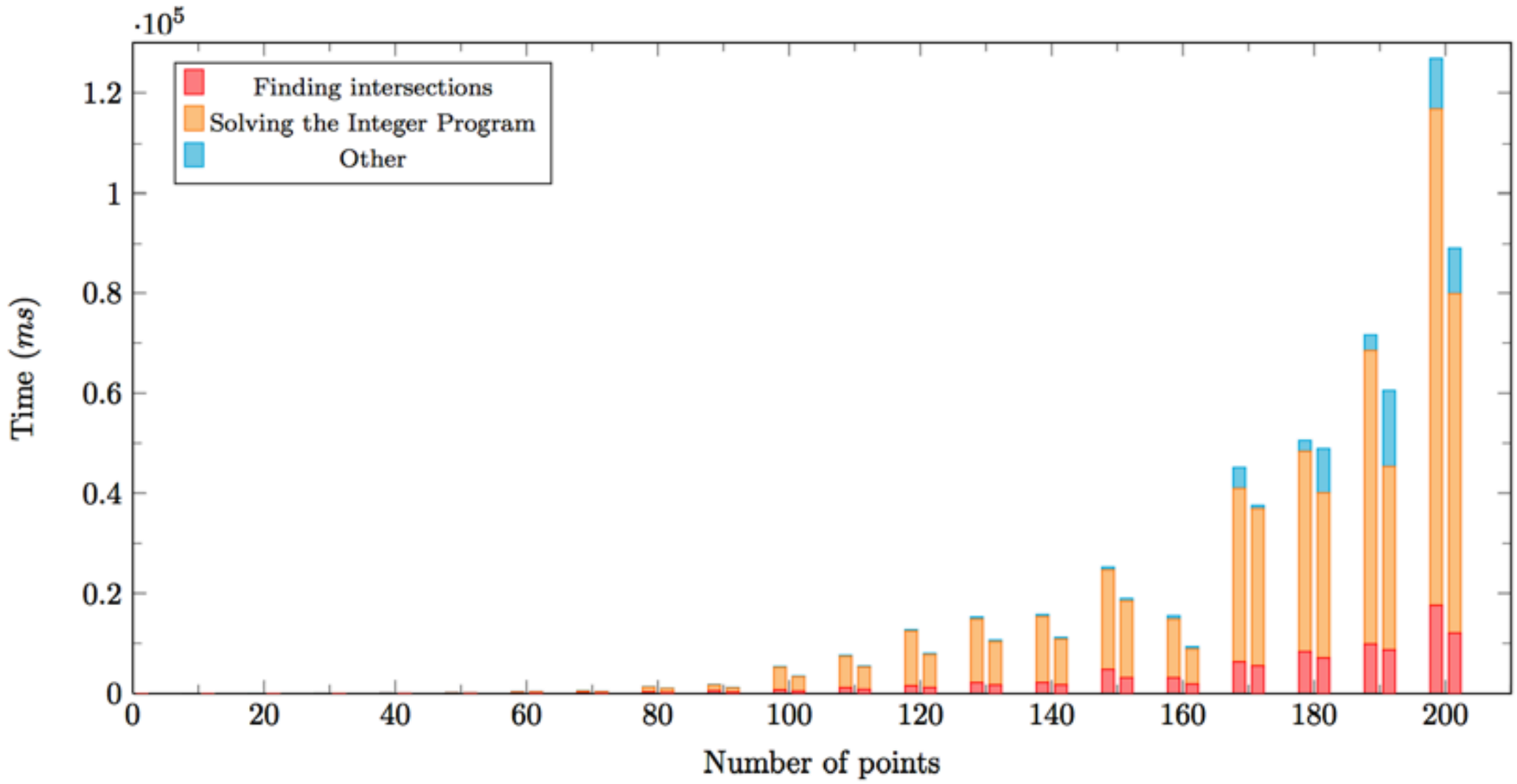
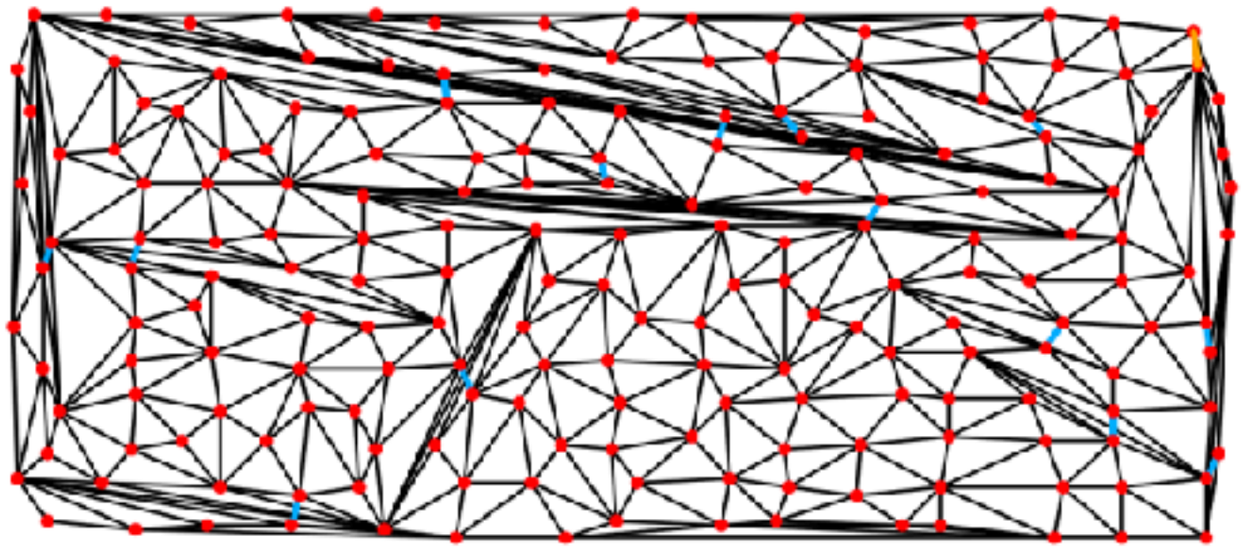


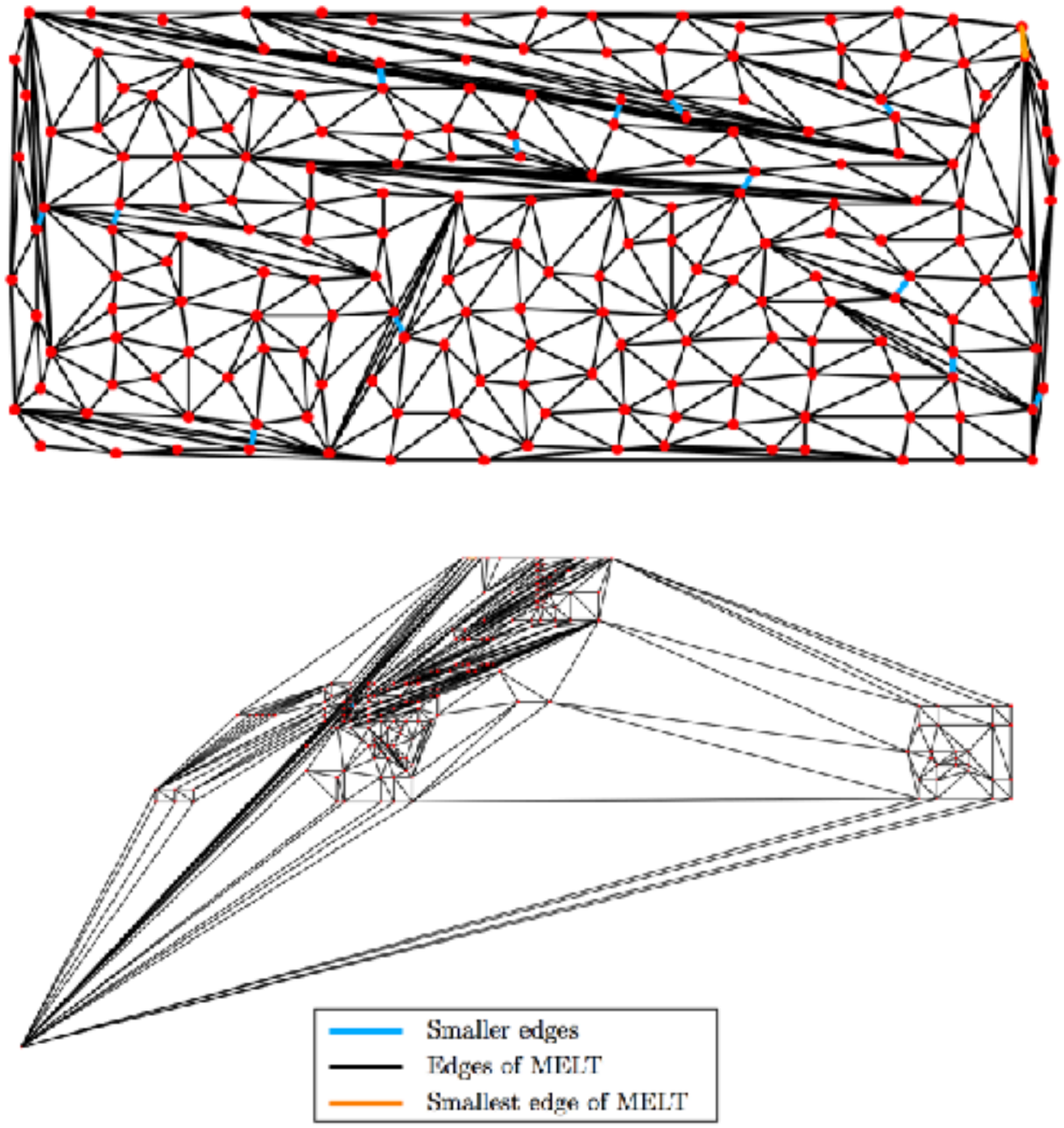
Figure 9: Average time (for random 100 random instances for each size) of the program components.
Left bars: Normal approach, based on IP 4.7
Right bars: Doubling approach, with AABB-Tree and IP 4.7

TSPLIB Instances

TSPLIB Instances



TSPLIB Instances



TSPLIB Instances (2)

TSPLIB Instances (2)

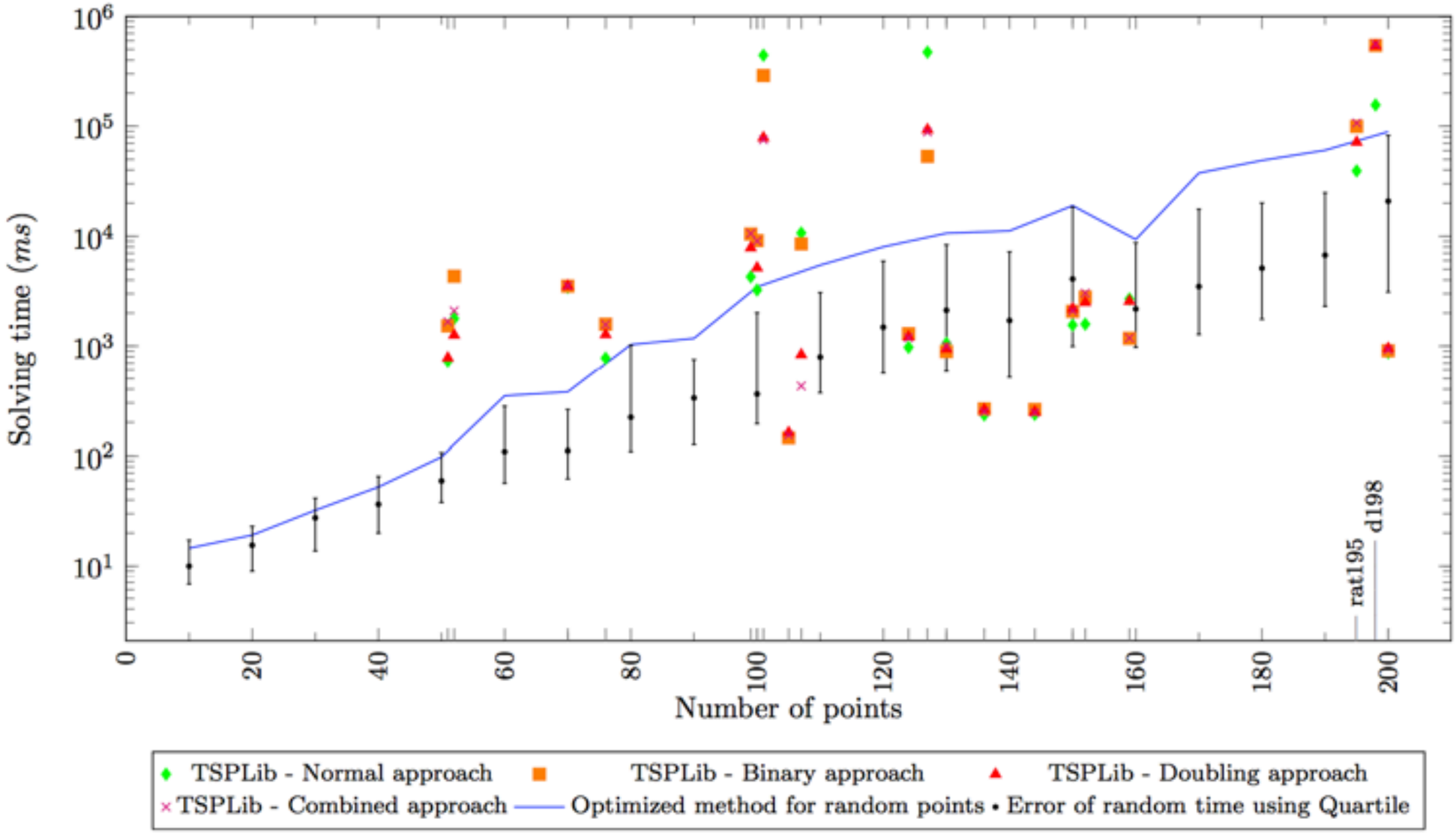


Figure 14: Comparison of TSPLIB instances with average times for random instances, using the doubling approach. (Error bars indicate percentiles 25, 50, 75.)

An Open Problem

An Open Problem

Conjecture: The length of the longest non-separated edge for n random points in a unit square grows with $O(1/n)$

An Open Problem

Conjecture: The length of the longest non-separated edge for n random points in a unit square grows with $O(1/n)$

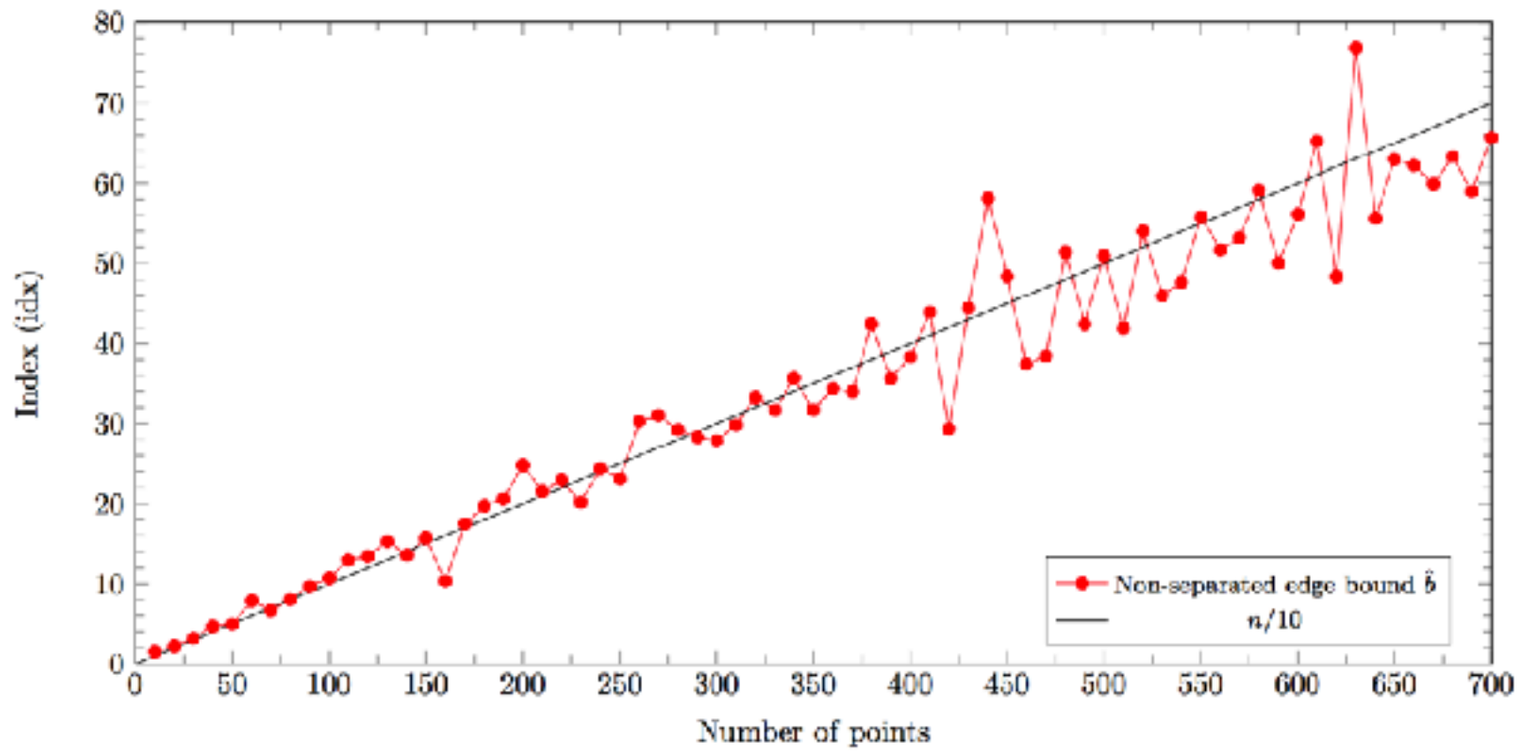


Figure 13: Average indices (for 100 random instances for each size) of the non-separated edge bound $\hat{b}$ for random sets with up to 700 points. (Error bars indicate one standard deviation.)

An Open Problem

Conjecture: The length of the longest non-separated edge for n random points in a unit square grows with $O(1/n)$

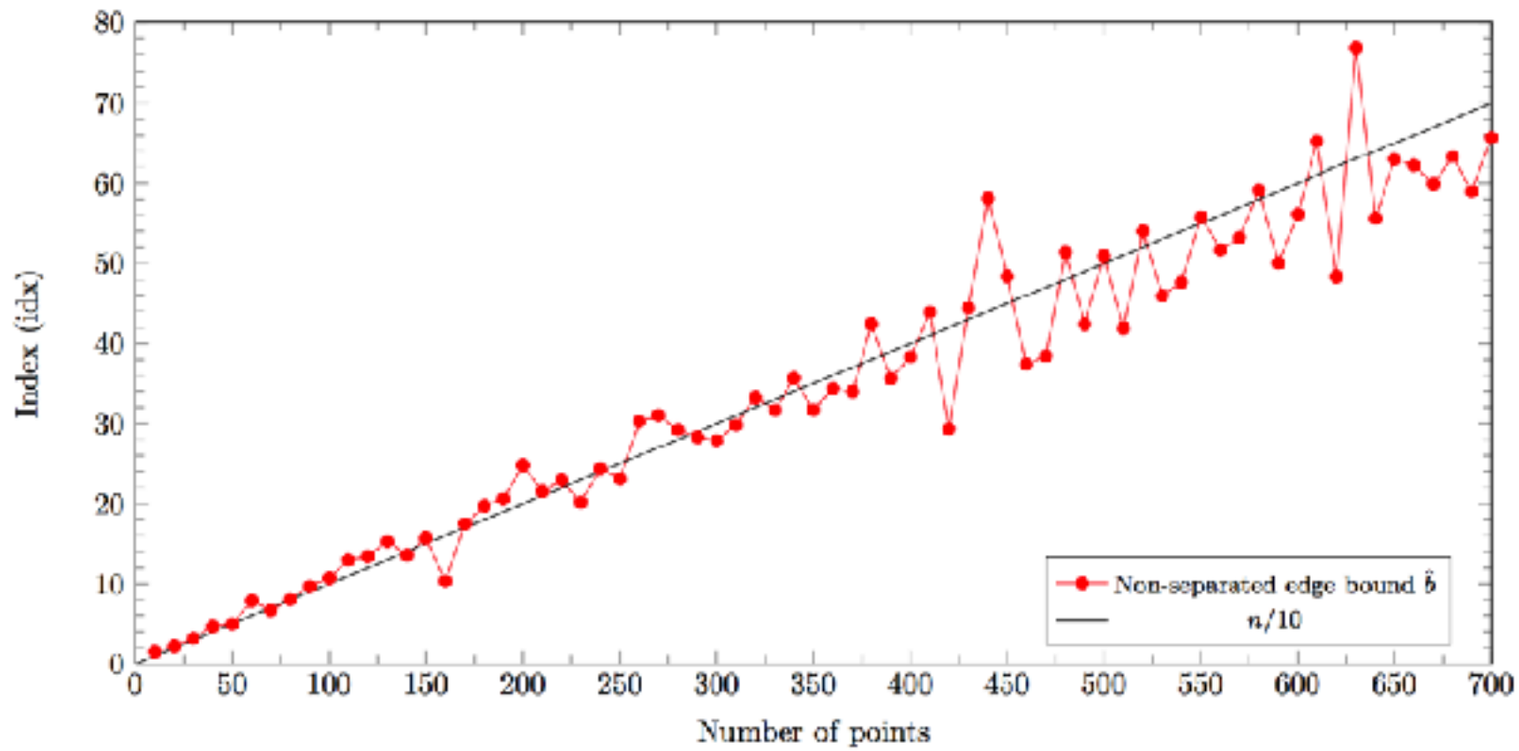
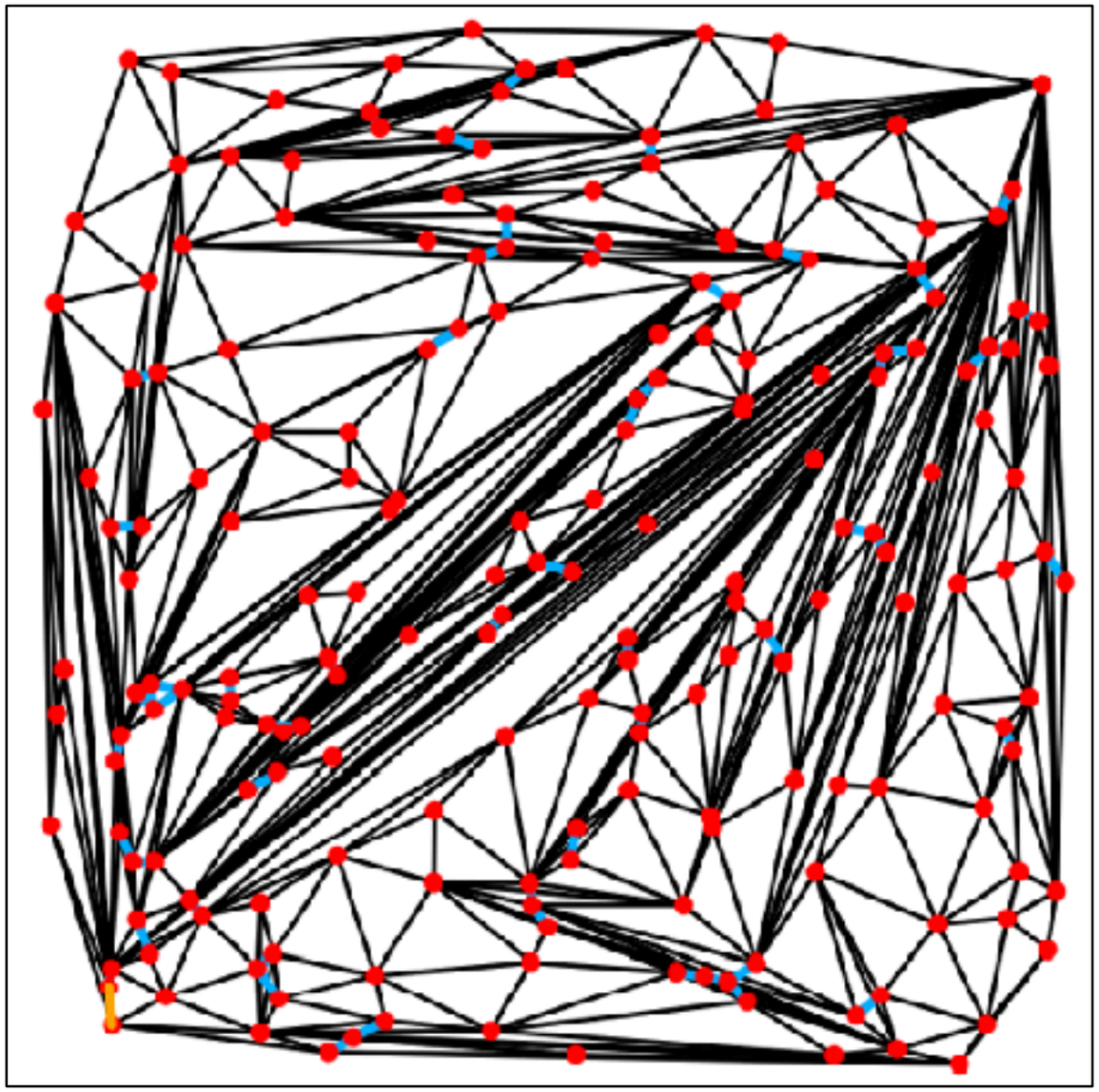


Figure 13: Average indices (for 100 random instances for each size) of the non-separated edge bound $\hat{b}$ for random sets with up to 700 points. (Error bars indicate one standard deviation.)

Consequence: Because the length of the shortest edge for uniform random points in a unit square grows with m^2, the same $\Theta(1/m)$ would hold for the solution of MELT.

Thank you!



Thank you for today!