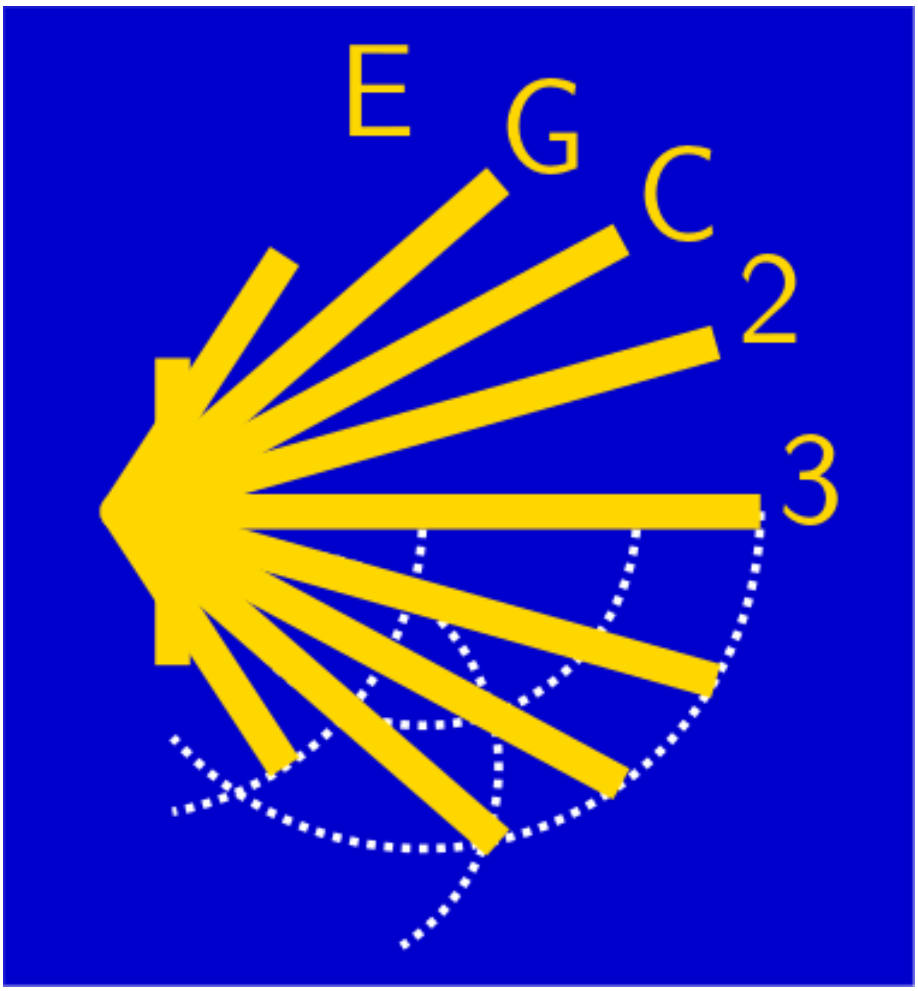
XX Spanish Meeting on Computational Geometry

July 3-5, 2023, Santiago de Compostela, Spain
https://egc23.web.uah.es



Book of Abstracts

Preface

The XX Spanish Meeting on Computational Geometry (formerly Encuentros de Geometría Computacional, EGC 2023) was held on July 3-5 in Santiago de Compostela, Spain. This book contains the abstracts of the invited talks, contributed papers, and contributed talks, accepted for presentation at this meeting.

This series of biennial meetings focuses on current research topics in discrete and computational geometry. Since the seminal edition in 1990, the Encuentros have combined a strong scientific program with a friendly atmosphere. The intended audience ranges from experienced researchers to students facing their debut in the area. The strong collaboration links of the Spanish community with foreign colleagues made advisable, in 2011, to change the language of the meeting to English and have the submissions peer-reviewed by an international program committee.

As in the last editions, authors were able to choose between submitting a 4-page ("paper") or a 1-page ("talk") abstract. We received a total of 30 submissions, consisting of 16 talks and 14 papers. Among them, one was rejected and the other 29 submissions were finally accepted, composing the core of this book.

The current edition of the meeting is the result of the work and dedication of a lot of people. First, our thanks go to the authors for choosing EGC to share and disseminate their work. Secondly, we would like to thank the members of the program committee and the external reviewers for accepting to contribute their expertise to this meeting carefully, constructively, and on time. Thirdly, we are truly thankful to the three excellent invited speakers for accepting our invitation: Ruy Fabila-Monroy, Christiane Schmidt, and Antonio Gómez Tato. Finally, it is very appreciated the support of the Departamento de Matemática, Área de Xeometría e Topoloxía da Facultade de Matemáticas da Universidade de Santiago de Compostela (Spain).

This edition would not have been possible without the tough work of the organizing committee: Enrique Macías - co-chair, David Mosquera, and Alba Sendón from the Universidade de Santiago de Compostela, David Orden - co-chair and Guillermo Esteban from the Universidad de Alcalá, and Carlos Seara - co-chair from the Universitat Politècnica de Catalunya. We also thank Fabian Klute from the Universitat Politècnica de Catalunya for helping producing these proceedings.

We are very grateful for the generous support of our contributors: Universitat Politècnica de Catalunya, BarcelonaTech (UPC), Departament de Matemàtiques (UPC), Universidad de Alcalá (UAH) and Departamento de Física y Matemáticas (UAH).

We like to express our gratitude to our appreciated colleagues who retire from academics around this edition and who contributed so much to the Spanish community on Computational Geometry: Mercè Claverol and Alfredo García.

During the business meeting a single bid was presented for 2025 by Francisco Santos (Universidad de Cantabria) and, as a consequence, the XXI edition of EGC 2025 will take place in Santander (Spain). Looking forward to seeing you all in two years in Santander!

July 2023,

Clemens Huemer, Pablo Pérez-Lantero, and Carlos Seara

Program Committee

Oswin Aichholzer (Graz University of Technology)
Carlos Alegría (Universitá degli Studi Roma Tre)
Jesús Antonio Álvarez López (Universidade de Santiago de Compostela)
Sergey Bereg (University of Texas at Dallas)
Mónica Blanco Gómez (Universidad de Cantabria)
Maike Buchin (Ruhr-University Bochum)
Jean Cardinal (Université libre de Bruxelles)
Frank Duque (Universidad Nacional de Colombia)
David Flores Peñaloza (Universidad Nacional Autónoma de México)
Marta Fort (Universitat de Girona)
María A. Hernández Cifre (Universidad de Murcia)
Clemens Huemer - co-chair (Universitat Politècnica de Catalunya)
Fabian Klute (Universitat Politècnica de Catalunya)
Dolores Lara (Cinvestav)
Enrique Macías (Universidade de Santiago de Compostela)
Joseph S. B. Mitchell (State University of New York at Stony Brook)
Mercè Mora (Universitat Politècnica de Catalunya)
Pablo Pérez Lantero - co-chair (Universidad de Santiago de Chile)
María Luz Puertas (Universidad de Almería)
Carlos Seara (Universitat Politècnica de Catalunya)
Ana Paula Tomás (Universidade do Porto)

External Reviewers

José Luis Álvarez (Universidad Nacional Autónoma de México)
Alma Arévalo Loyola (Carleton University)
Nestaly Marín Nevárez (Universidad Nacional Autónoma de México)
Juan Carlos Catana-Salazar (HP Labs)

Organizing Committee

Guillermo Esteban (Universidad de Alcalá de Henares & Carleton University)
Enrique Macías - co-chair (Universidade de Santiago de Compostela)
David Mosquera (Universidade de Santiago de Compostela)
David Orden - co-chair (Universidad de Alcalá de Henares)
Carlos Seara - co-chair (Universitat Politècnica de Catalunya)
Alba Sendón (Universidade de Santiago de Compostela)

Contents

Invited Talks

Contributed Papers

Planar point sets with integer grids

Ruy Fabila-Monroy[1]

[1]Cinvestav (México)

Let S be a set of n points in general position in the plane" is a mantra in Discrete and Computational Geometry. In this talk we give an overview on various results on finite planar point sets in general position, where it is required that the points have integer coordinates.

k-Transmitters/k-Modems

Christiane Schmidt[1]

[1]Linköping University (Sweden)

For the classical Art Gallery problem numerous variants–varying either the capabilities of the guards or the environment to be guarded–have been studied. One such variant was introduced in 2009 by Aichholzer et al./Fabila-Monroy et al. (a limited version was actually already presented in 1988): k-modems or k-transmitters, which are guards that can see through at most k walls.

In this talk, I will review some of the existing work on the problem, highlight several of the–still–open problems and present both an inapproximability result and a polylog approximation algorithm for a mobile k-transmitter, that is, a k-transmitter watchman.

Parts of this talk are joint work with Bengt J. Nilsson.

Mathematics and biology in the 21st Century

Antonio Gómez Tato[1]

[1]Universidade de Santiago de Compostela (Spain)

The publication in February 2001 of the first draft of the human genome sequences marks the beginning of a new era in biological research. In this new stage of biology, mathematics is more necessary than ever. In this lecture I will give several examples where mathematics is proving its effectiveness in the task of understanding cells and living organisms and will present some of the challenges in biology that mathematics can and should contribute to solve.

Realizations of multiassociahedra via rigidity *

Luis Crespo Ruiz† and Francisco Santos‡

Departamento de Matemáticas, Estadística y Computación, Universidad de Cantabria, 39005 Santander, Spain

Abstract

Let $\Delta_k(n)$ denote the simplicial complex of $(k+1)$-crossing-free subsets of edges in $\binom{[n]}{2}$. Here $k, n \in \mathbb{N}$ and $n \geq 2k+1$. Jonsson (2005) proved that (neglecting the short edges that cannot be part of any $(k+1)$-crossing), $\Delta_k(n)$ is a shellable sphere of dimension $k(n-2k-1)-1$, and conjectured it to be polytopal.

Despite considerable effort, the only values of (k, n) for which the conjecture is known to hold are $n \leq 2k+3$ (Pilaud and Santos, 2012) and $(2, 8)$ (Bokowski and Pilaud, 2009). Using ideas from rigidity theory we realize $\Delta_k(n)$ as a polytope for $(k, n) \in \{(2, 9), (2, 10), (3, 10)\}$. We also realize it as a simplicial fan for all $n \leq 13$ and arbitrary k, except the pairs $(3, 12)$ and $(3, 13)$.

1 The multiassociahedron

Triangulations of the convex n-gon P $(n > 2)$ are the facets of an abstract simplicial complex with vertex set $\binom{[n]}{2}$ and defined by taking as simplices all the non-crossing sets of diagonals. This simplicial complex, ignoring the boundary edges $\{i, i+1\}$, is a polytopal sphere of dimension $n-4$ dual to the *associahedron*. (Here and all throughout the paper, indices for vertices of the n-gon are regarded modulo n). A similar complex can be defined if we forbid crossings of more than a certain number k of edges (assuming $n > 2k+1$), instead of forbidding pairwise crossings.

Definition 1 *Two disjoint pairs $\{i, j\}, \{k, l\} \in \binom{[n]}{2}$, with $i < j$ and $k < l$, of $\binom{[n]}{2}$* cross *if $i < k < j < l$ or $k < i < l < j$. That is, if they cross as diagonals of a convex n-gon. A k-*crossing *is a subset of k elements of $\binom{[n]}{2}$ such that every pair cross. A subset of $\binom{[n]}{2}$ is $(k+1)$-*free *if it doesn't contain any $(k+1)$-crossing. A k-*triangulation *is a maximal $(k+1)$-free set. We call $\Delta_k(n)$ the simplicial complex consisting of $(k+1)$-free sets of diagonals, whose facets are the k-triangulations.*

Diagonals of length at most k (with length measured cyclically) cannot participate in any $(k+1)$-crossing. Thus, it makes sense to define the reduced complex $\overline{\Delta}_k(n)$ obtained from $\Delta_k(n)$ by deleting them. We call $\overline{\Delta}_k(n)$ the *multiassociahedron* or k-associahedron.

It was proved in [14, 9] that every k-triangulation of the n-gon has exactly $k(2n-2k-1)$ diagonals. That is, $\Delta_k(n)$ is pure of dimension $k(2n-2k-1)-1$. Jonsson [11] further proved that the reduced version $\overline{\Delta}_k(n)$ is a shellable sphere of dimension $k(n-2k-1)-1$, and conjectured it to be the normal fan of a polytope. See [15, 16, 19] for additional information.

Conjecture 2 ([11]) *$\overline{\Delta}_k(n)$ is a polytopal sphere for every $n \geq 2k+1$; that is, there is a simplicial polytope of dimension $k(n-2k-1)-1$ with $\binom{n}{2} - kn$ vertices whose lattice of proper faces is isomorphic to $\overline{\Delta}_k(n)$.*

Conjecture 2 is easy to prove for $n \leq 2k+3$ [16]. The only additional case for which Jonsson's conjecture is known to hold is $k = 2$ and $n = 8$ [2]. In some additional cases $\overline{\Delta}_k(n)$ has been realized as a complete simplicial fan, but it is open whether this fan is polytopal. This includes the cases $n \leq 2k+4$ [1], $k = 2$ and $n \leq 13$ [13], and $k = 3$ and $n \leq 11$ [1].

Interest in the polytopality of $\overline{\Delta}_k(n)$ also comes from cluster algebras and Coxeter combinatorics. Let $w \in W$ be an element in a Coxeter group W and let Q be a word of a certain length N. Assume that Q contains as a subword a reduced expression for w. The *subword complex* of Q and w is the simplicial complex with vertex set $[N]$ and with faces the subsets of positions that can be deleted from Q and still contain a reduced expression for w. Knutson and Miller [12, Theorem 3.7 and Question 6.4] proved that every subword complex is either a shellable ball or sphere, and they asked whether all spherical subword complexes are polytopal. It was later proved by Stump [19, Theorem 2.1] that $\overline{\Delta}_k(n)$ is a spherical subword complex for the Coxeter system A_{n-2k-1} and, moreover, it is *universal*: every other spherical subword complex of type A appears as a link in some $\overline{\Delta}_k(n)$ [17, Proposition 5.6]. Hence, Conjecture 2 is equivalent to a positive answer in type A to the question of Knutson and Miller.

2 Realizing a simplicial complex as a polytope

If Δ is a pure simplicial complex with vertex set V of dimension $D-1$ (its facets have size D) realizing it as a

*Supported by grant PID2019-106188GB-I00 funded by MCIN/AEI/10.13039/501100011033, by FPU19/04163 of the Spanish Government, and by project CLaPPo (21.SI03.64658) of Univ. de Cantabria and Banco Santander.

†Email: luis.cresporuiz@unican.es.

‡Email: francisco.santos@unican.es.

polytope is the same as finding a vector configuration $\mathcal{V} = \{v_i\}_{i\in V} \subset \mathbb{R}^D$ on which Δ yields a *complete simplicial fan*, and then proving the fan to be a *regular triangulation* of $\mathcal{V}$. See [8, Section 9.5] for details.

To prove that an embedding is a simplicial fan we use a version of [8, Corollary 4.5.20] which says that in order for a vector configuration $\mathcal{V} \subset \mathbb{R}^D$ to embed Δ as a simplicial fan the following *Interior Cocircuit Property (ICoP)* is necessary and almost sufficient:

(ICoP) For every facet T of Δ the vectors $\{v_{ij} : \{i,j\} \in T\}$ are independent, and for every two adjacent facets T_1 and T_2 the linear dependence among the vectors $\{v_{ij} : \{i,j\} \in T_1 \cup T_2\}$ has the same sign for the two elements in $T_1 \setminus T_2$ and $T_2 \setminus T_1$.

We apply this to the complex $\overline{\Delta}_k(n)$, for which $V \subset \binom{[n]}{2}$ and $D = k(n-2k-1)$. Each facet is a k-triangulation and two facets are adjacent if and only if the k-triangulations differ by a *flip*, defined as follows:

Proposition 3 (Flips [16, Section 5]) *For every edge f of a k-triangulation T with length greater than k, there is a unique edge $e \in \binom{[n]}{2}$ such that*

$$T \triangle \{e,f\} := T \setminus \{f\} \cup \{e\}$$

is another k-triangulation.

Once we have the complete fan, regularity is equivalent to the feasibility of a system of linear inequalities. We check this with a version of [18, Theorem 3.7], which in turn is related to [8, Proposition 5.2.6(i)].

In some proofs we also use the following fact:

Proposition 4 (Short cycles [5, Cor. 2.9]) *All links of dimension 1 in $\overline{\Delta}_k(n)$ are cycles of length ≤ 5.*

3 Rigidity

Let $\mathbf{p} = (p_1, \dots, p_n)$ be a set of n points in $\mathbb{R}^d$, labelled by $[n]$. Their *bar-and-joint rigidity matrix* is the following $\binom{n}{2} \times nd$ matrix:

$$R(\mathbf{p}) := \begin{pmatrix} p_1-p_2 & p_2-p_1 & \dots & 0 \\ p_1-p_3 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ p_1-p_n & 0 & \dots & p_n-p_1 \\ 0 & p_2-p_3 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & p_n-p_{n-1} \end{pmatrix}.$$

The shape of the matrix is as follows: there is a row for each pair $\{i,j\} \in \binom{[n]}{2}$, so rows can be considered labeled by edges in the complete graph K_n. Then, there are n blocks of columns, one for each point p_i and with d columns in each block; in the row of an edge $\{i,j\}$ (or $\{j,i\}$) only the blocks of vertices i and j ae nonzero, and they contain respectively the vectors $p_i - p_j$ and $p_j - p_i$. Put differently, the matrix can be interpreted as a "directed incidence matrix" of the complete graph K_n, except instead of having a single $+1$ and -1 for each edge-vertex incidence we have the d-dimensional vectors $p_i - p_j$ and $p_j - p_i$. For an $E \subset \binom{[n]}{2}$ we denote by $R(\mathbf{p})|_E$ the restriction of $R(\mathbf{p})$ to the rows or elements indexed by E.

Definition 5 *Let $E \subset \binom{[n]}{2}$ be a subset of edges of K_n (equivalently, of rows of $R(\mathbf{p})$). We say that E, or the corresponding subgraph of K_n, is* self-stress-free *or* independent *if the rows of $R(\mathbf{p})|_E$ are linearly independent, and* rigid *or* spanning *if they are linearly spanning (that is, they have the same rank as the whole matrix $R(\mathbf{p})$).*

That is, self-stress-free and rigid graphs are, respectively, the independent and spanning sets in the linear matroid of rows of $R(\mathbf{p})$. We call this matroid the *bar-and-joint rigidity matroid* of $\mathbf{p}$ and denote it $\mathcal{R}(\mathbf{p})$.

The number $k(2n-2k-1) = 2kn - \binom{2k+1}{2}$ of edges in a k-triangulation happens to coincide with the rank of $R(\mathbf{p})$ (or of $\mathcal{R}(\mathbf{p})$) when $\mathbf{p}$ is a set of n points in general position in $\mathbb{R}^{2k}$. This suggest to try to use these matrices to try to embed $\overline{\Delta}_k(n)$ as a simplicial fan. Or, more generally, we can use any of the following two other versions of rigidity, based on matrices of the same shape, size, and rank as $R(\mathbf{p})$, and which fit into the framework of *abstract rigidity matroids of dimension $2k$ on n elements*.

- The *hyperconnectivity* matroid of $\mathbf{p} \subset \mathbb{R}^d$, denoted $\mathcal{H}(\mathbf{p})$, is the matroid of rows of

$$H(\mathbf{p}) := \begin{pmatrix} p_2 & -p_1 & 0 & \dots & 0 & 0 \\ p_3 & 0 & -p_1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ p_n & 0 & 0 & \dots & 0 & -p_1 \\ 0 & p_3 & -p_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & p_n & -p_{n-1} \end{pmatrix}$$

- For points $\mathbf{q} = (q_1, \dots, q_n)$ in $\mathbb{R}^2$ and a parameter $d \in \mathbb{N}$, the d-dimensional *cofactor rigidity* matroid of the points $q_1, \dots, q_n$, which we denote $\mathcal{C}_d(\mathbf{q})$, is the matroid of rows of

$$C_d(\mathbf{q}) := \begin{pmatrix} \mathbf{c}_{12} & -\mathbf{c}_{12} & 0 & \dots & 0 \\ \mathbf{c}_{13} & 0 & -\mathbf{c}_{13} & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{c}_{1n} & 0 & 0 & \dots & -\mathbf{c}_{1n} \\ 0 & \mathbf{c}_{23} & -\mathbf{c}_{23} & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & -\mathbf{c}_{n-1,n} \end{pmatrix},$$

where the vector $\mathbf{c}_{ij} \in \mathbb{R}^d$ associated to $q_i = (x_i, y_i)$ and $q_j = (x_j, y_j)$ is

$$\mathbf{c}_{ij} = \big((x_i - x_j)^{d-1}, (y_i - y_j)(x_i - x_j)^{d-2}, \ldots, (y_i - y_j)^{d-1}\big).$$

In [3] we prove that these three rigidity theories coincide when the points $\mathbf{p}$ or $\mathbf{q}$ are chosen along the moment curve (for bar-and-joint and hyperconnectivity) and the parabola (for cofactor). More precisely:

Theorem 6 ([3]) *Let $t_1 < \cdots < t_n \in \mathbb{R}$ be real parameters. Let*

$$p_i = (1, t_i, \ldots, t_i^{d-1}) \in \mathbb{R}^d,$$
$$p'_i = (t_i, t_i^2, \ldots, t_i^d) \in \mathbb{R}^d,$$
$$q_i = (t_i, t_i^2) \in \mathbb{R}^2.$$

Then, the matrices $H(p_1, \ldots, p_n)$, $R(p'_1, \ldots, p'_n)$ and $C(q_1, \ldots, q_n)$ can be obtained from one another multiplying on the right by a regular matrix and then multiplying its rows by some positive scalars. In particular, the three matrices define the same oriented matroid.

Definition 7 *We call the matrix $H(p_1, \ldots, p_n)$ in the statement of Theorem 6 the* polynomial d-rigidity matrix with parameters $t_1, \ldots, t_n$. *We denote it $P_d(t_1, \ldots, t_n)$, and denote $\mathcal{P}_d(t_1, \ldots, t_n)$ the corresponding matroid.*

Summing up: for any choice of points $\mathbf{p} \in \mathbb{R}^{2k}$ or $\mathbf{q} \in \mathbb{R}^2$ in general position, the rows of the matrices $R(\mathbf{p})$, $H(\mathbf{p})$ or $C_{2k}(\mathbf{q})$ are a real vector configuration $\mathcal{V} \subset \mathbb{R}^{2kn}$ of rank $k(2n - 2k - 1)$. Moreover, if $\mathbf{p}$ is chosen along the moment curve or $\mathbf{q}$ along the parabola the three theories give linearly equivalent embeddings. The question we address is whether using these vectors as rays we get that the reduced k-associahedron $\overline{\Delta}_k(n)$ is a polytopal fan.

An alternative to realize the fan is "bipartizing" the k-triangulations, as follows:

Definition 8 *The* bipartization *of a graph $G = ([n], E)$ is the graph $G' = ([n] \cup [n]', E')$ where $E' = \{(i, n+1-j) : \{i, j\} \in E, i < j\}$. The* (reduced) bipartization *of a k-triangulation is its bipartization restricted to $[n-k-1] \cup [n-k-1]'$.*

Reduced bipartizations of k-triangulations have $2kn - 3k^2 - 2k$ edges, which is exactly the rank of the hyperconnectivity matroid in dimension k restricted to bipartite graphs. So, we can also use as a vector configuration the rows of $H(\mathbf{p})$ for $\mathbf{p} \subset \mathbb{R}^k$ in general position, restricted or not to the moment curve.

Conjecture 9 *1. k-triangulations of the n-gon are bases in the bar-and-joint rigidity matroid of generic points along the moment curve in dimension $2k$.*

2. Bipartized k-triangulations of the n-gon are bases in the bar-and-joint rigidity matroid of generic points along the moment curve in dimension k.

4 Main results

First, as evidence for Conjecture 9 we prove the case $k = 2$:

Theorem 10 ([5, Thm. 1.4]) *2-triangulations are isostatic in dimension 4 for generic positions along the moment curve.*

One may be tempted to change "generic" to "arbitrary" in Conjecture 9, but we show that this stronger conjecture fails in the worst possible way; for every $k \geq 3$ and $n \geq 2k + 3$, the standard positions along the moment curve make some k-triangulation not a basis:

Theorem 11 ([5, Thm. 1.6], [6, Th. 1.13])

1. *The graph $K_9 - \{16, 37, 49\}$ is a 3-triangulation of the n-gon, but it is dependent in the rigidity matroid $\mathcal{C}_6$ for any configuration $\mathbf{q} \subset \mathbb{R}^2$ if the lines through q_1q_6, q_3q_7, and q_4q_9 meet at a point. This occurs, for example, if we take the nine points on the parabola with $t_i = i$.*

2. *The bipartization of the same graph is dependent in $\mathcal{H}_3$ if the cross-ratio between the hyperplanes $(12, 23; 24, 25)$ equals $(2'4', 2'3'; 1'2', 2'5')$, as happens with points along the moment curve with $\mathbf{t} = (1, 3, 4, 5, 7, 1, 3, 4, 5, 7)$.*

In fact, for $n \leq 2k + 3$ we can characterize exactly what positions realize $\overline{\Delta}_k(n)$ as a fan, for cofactor rigidity (and, in particular, for the other two forms of rigidity with positions along the moment curve), and for bipartite rigidity along the moment curve. In the case $n = 2k + 3$ this is governed by the geometry of the star-polygon formed by the k-relevant edges. More precisely, we call "big side" of each relevant edge (that is, edge of $k+1$) in a $(2k+3)$-gon the open half-plane containing $k + 1$ vertices:

Theorem 12 ([5, Thm. 3.14], [6, Thm. 5.6])

1. *For $n = 2k + 2$, any choice of $q_1, \ldots, q_{2k+2} \in \mathbb{R}^2$ in convex position for cofactor rigidity, and any choice of $t_1 < \cdots < t_{k+1}, t'_1 < \cdots < t'_{k+1}$ in the moment curve for bipartite rigidity, realizes $\overline{\Delta}_k(2k + 2)$ as a polytopal fan.*

2. *Let $q_1, q_2, \ldots, q_{2k+3} \in \mathbb{R}^2$ be in convex position. $\overline{\Delta}_k(2k + 3)$ is realized by $C_{2k}(\mathbf{q})$ as a complete fan if and only if the big sides of all relevant edges have a non-empty intersection.*

3. *Let $t_1 < \cdots < t_{k+2}, t'_1 < \cdots < t'_{k+2}$ be parameters for the vertices of $K_{k+2,k+2}$ in the moment curve. $\overline{\Delta}_k(2k+3)$ is realized by $P_k(\mathbf{t})$ as a complete fan if and only if one of the following holds:*

   - *$k = 2$.*
   - *$k = 3$ and the cross-ratio $(1, 3; 4, 5)$ is greater than $(4', 3'; 1', 5')$.*
   - *$k \geq 4$ and the cross-ratio $(i_1, i_2; i_3, k+2)$ is greater than $((k+3-i_1)', (k+2-i_2)'; (k+3-i_3)', (k+2)')$, for any i_1, i_2, i_3 with $2 \leq i_1 < i_2 < i_3 - 1 \leq k$.*

   Here, by cross-ratio between four points, we mean the cross-ratio between their parameters t.

Interestingly, from parts (2) and (3) of this result it is quite easy to show that *no positions* of points along the moment curve realize $\overline{\Delta}_k(n)$, for several values of k and n:

Corollary 13 ([5, Thm. 1.7], [6, Thm. 1.14])

1. *If $k \geq 3, n \geq 2k+6$ then no choice of points $\mathbf{q} \subset \mathbb{R}^2$ in convex position realizes $\overline{\Delta}_k(n)$ as a fan via cofactor rigidity.*

2. *If $k = 3, n \geq 12$, or $k \geq 4, n \geq 2k+4$, then no choice of points $\mathbf{t} \in \mathbb{R}^{2(n-k-1)}$ in the moment curve realizes $\overline{\Delta}_k(n)$ as a fan via cofactor rigidity.*

Observe that this is not a counter-example to Conjecture 9, which is only about linear independence of the vectors generating each facet of the fan, not about the fan itself.

Finally, for every $n \leq 13$ we have experimentally found positions along the moment curve realizing $\overline{\Delta}_k(n)$ as a fan, except in the cases $(n,k) \in \{(3,12),(3,13)\}$ which are forbidden by Corollary 13. For many of them we have also realized the polytope:

Theorem 14 ([5, Lem. 4.13 & 4.14], [6, Thm. 5.10]) *Let $\mathbf{t} = \{1, 2, \ldots, n\}$ be standard positions for the parameters. Then:*

1. *Standard positions realize $\overline{\Delta}_2(n)$ as the normal fan of a polytope for $P_4(\mathbf{t})$ with the original graph if $n \leq 9$, and for $P_2(\mathbf{t})$ with the bipartized graph if $n \leq 8$.*

2. *The non-standard positions $\mathbf{t} = (-2, 1, 2, 3, 4, 5, 6, 7, 9, 20)$ for $P_4(\mathbf{t})$ with the original graph, and the near-lexicographic positions $t_i = t'_i = 2^{(i-1)^2}$ for $P_2(\mathbf{t})$ with the bipartized graph, realize $\overline{\Delta}_2(10)$ as the normal fan of a polytope.*

3. *Standard positions realize $\overline{\Delta}_2(n)$ as a complete fan for all $n \leq 13$ with both forms of rigidity.*

4. *Equispaced positions along the circle with the original graph realize $\overline{\Delta}_k(n)$ as a fan for $(k,n) \in \{(3,10),(3,11),(4,12),(4,13)\}$. The first one is polytopal.*

5. *The positions $\mathbf{t} = (0, 1, 31, 32, 42, 67, 100)$ at both sides with bipartite rigidity realize $\overline{\Delta}_3(11)$ as a fan.*

References

[1] N. Bergeron, C. Ceballos, J.-P. Labbé, Fan realizations of subword complexes and multi-associahedra via Gale duality, *Discrete Comput. Geom.* **54**(1) (2015), 195–231.

[2] J. Bokowski, V. Pilaud, On symmetric realizations of the simplicial complex of 3-crossing-free sets of diagonals of the octagon, Proceedings of the 21st Annual Canadian Conference on Computational Geometry, Vancouver, British Columbia, Canada, August 17–19, 2009.

[3] L. Crespo Ruiz, F. Santos, Bar-and-joint rigidity on the moment curve coincides with cofactor rigidity on a conic, *Combinatorial Theory*, **3**(1) (2023), #15, 13pp.

[4] L. Crespo Ruiz, F. Santos, Multitriangulations and tropical Pfaffians, preprint, 2022, `arXiv:2203.04633`.

[5] L. Crespo Ruiz, F. Santos, Realizations of multiassociahedra via rigidity, preprint, 2022, `arXiv:2212.14265`.

[6] L. Crespo Ruiz, Realizations of multiassociahedra via bipartite rigidity, preprint, 2023, `arXiv:2303.15776`.

[7] P. R. Cromwell, *Polyhedra*, Cambridge University Press, 1997.

[8] J. A. De Loera, J. Rambau, F. Santos, *Triangulations: Structures for Algorithms and Applications*, Springer-Verlag, 2012.

[9] A. Dress, J. H. Koolen and V. Moulton. On line arrangements in the hyperbolic plane. *Eur. J. Comb.*, **23**(5) (2002), 549–557.

[10] B. Grünbaum, G. C. Shephard, *Tilings and Patterns*, W. H. Freeman & Co., 1989.

[11] J. Jonsson, Generalized triangulations and diagonal-free subsets of stack polyominoes, *J. Comb. Theory Ser. A* **112**(1) (2005), 117–142.

[12] A. Knutson and E. Miller. Subword complexes in Coxeter groups. *Adv. Math.*, **184**(1) (2004), 161–176.

[13] T. Manneville, Fan realizations for some 2-associahedra, *Experimental Mathematics*, **27**(4) (2017), 377–394.

[14] T. Nakamigawa. A generalization of diagonal flips in a convex polygon. *Theor. Comput. Sci.* **235**(2) (2000), 271–282.

[15] V. Pilaud, M. Pocchiola, Multitriangulations, Pseudotriangulations and Primitive Sorting Networks. *Discrete Comput. Geom.* **48** (2012), 142–191.

[16] V. Pilaud, F. Santos, Multitriangulations as Complexes of Star Polygons, *Discrete Comput. Geom* **41** (2009), 284–317.

[17] V. Pilaud, F. Santos, The brick polytope of a sorting network. *European J. Combin.* **33**(4) (2012), 632–662.

[18] G. Rote, F. Santos, I. Streinu, Expansive motions and the polytope of pointed pseudo-triangulations, in: *Discrete and Computational Geometry – The Goodman-Pollack Festschrift*, Algorithms and Combinatorics, 25, Springer Verlag, Berlin, 2003, pp. 699–736.

[19] C. Stump. A new perspective on k-triangulations. *J. Comb. Theory A* **118**(6) (2011), 1794–1800.

[20] W. Whiteley, Some Matroids from Discrete Applied Geometry, in: *Matroid Theory*, Contemporary Mathematics, 197, 1996, pp. 171–311.

Covering segments on a line with drones

Sergey Bereg*[1], José-Miguel Díaz-Báñez†[2], Alina Kasiuk‡[2], Miguel-Angel Pérez-Cutiño§[2], and Fabio Rodríguez¶[2]

[1]Department of Computer Science, University of Texas at Dallas, USA
[2]Department of Applied Mathematics, University of Seville, SPAIN

Abstract

Covering a set of segments in a plane with vehicles of limited autonomy is a problem of practical interest. The limited battery endurance imposes periodical visits to a static base station. Typically, two optimization problems are considered: minimize the number of tours, and minimize the total traveled distance. In a general setting, the problems are NP-hard and in this letter, we study the one-dimensional version. For covering segments on a line, we design efficient solutions for both optimization problems. First, we design a Greedy algorithm that is optimal for the first task, and for both tasks when only one segment is considered. Being n and m the number of segments and tours of an optimal solution, respectively, our algorithm runs in $O(m+n)$ time. For the second criterion, our solution is based on Dynamic Programming and runs in $O(n^2)+O(nm)$ time.

1 Introduction

Trajectory optimization through linear segments is of practical interest in the robotics community. Road network patrolling, anomaly detection in solar power plants, power lines inspection and other similar infrastructures with unmanned vehicles are studied in various pieces of research [2, 1, 6]. In this work, we will use the term drone, though this research may be applied to any agent with limited autonomy. The use of drones or Unmanned Aerial Vehicles (UAVs), commonly called drones, has been proposed for the efficient maintenance of infrastructures, in order to reduce potential risks and costs for the distribution companies [7, 3]. The battery life limit of these small-size robots severely restricts the duration of the mission, as it becomes impossible to complete the overall coverage with a single tour. Therefore, considering each tour should start and end at a base station, the problem of minimizing the total cost of travel is hard in general and some heuristics have been considered in the literature [4, 5].

Two optimization problems can be formulated when the objects to be covered are line segments. Given a set of line segments S with any distribution in the plane, a depot or a base location O from where the robots can be launched and recharged and, a real number L, we consider:

- $MinTours$-problem: Finding the minimum number of tours $t_1, \cdots, t_p$ covering S, that is,

$$S \subset \bigcup_{i=1}^{p} t_i.$$

- $MinDistance$-problem: Compute a set of tours that covers S with minimum total length.

A tour t is considered to be *valid* if it starts and ends at O, and the length of t is at most L. The length of a tour is the sum of the Euclidean distance between its consecutive vertices; the length of a set of tours is the sum of the lengths of each of its elements. The NP-hardness of $MinTours$- and $MinDistance$-problems in the plane can be proved by a reduction from the Traveling Salesman Problem (TSP). However, in this paper we show that the one-dimensional case related to both problems, where segments are located through a line, can be solved in polynomial time. We consider several scenarios, designing efficient algorithms capable of finding the optimal solution. The paper is structured as follows: Section 2 formally defines the considered problems; Section 3 describes the optimal solution to the problem of finding the minimum number of tours; and Section 4 focuses on finding the set of tours covering S with minimum total length. In this version, we omit several proofs due to space restrictions.

2 Problem formulation

Let $S = \{s_1, \cdots, s_n\}$ be a set of disjoint segments arranged on a line, O be a point on the plane corresponding to a base station, and L be the maximum distance that a drone with constant battery life can travel. The problem is to find a set of drone routes (*tours*) with lengths no greater than L starting and ending at O so that they jointly traverse all segments

*Email: besp@utdallas.edu
†Email: dbanez@us.es
‡Email: akasiuk@us.es
§Email: migpercut@alum.us.es
¶Email: frodriguex@us.es

in S with minimum total cost. Two objective functions are considered: the number of tours or the total traveled distance (sum of the lengths of the tours).

Since segments in S lie on a line, we define the problems using the following notations. Let $s_i = [a_i, b_i], i = 1, 2, \ldots, n$ be n disjoint intervals on the line $y = 0$ such that $a_1 < a_2 < \cdots < a_n$; a_1, b_n are the edges of S. Let $O = (0, -h)$ be the base station and $L > 0$ be the maximum length of a tour using the full battery. Formally, the problem is to compute a set of tours $T = \{t_1, t_2, \ldots, t_m\}$ covering S so that:

- $1DMinTours$-problem: the number of tours $m = |T|$ is minimized.
- $1DMinDistance$-problem: $\sum_{i=1}^{m} l_i$ is minimized, where l_i is the length of t_i.

For simplicity, we assume that the tours of T are given ordered, that is, from left to right or from right to left. In addition, we consider other important notations and definitions. For an interval $s_i = [a_i, b_i]$, we term a_i, b_i as the left and right point respectively of s_i. This concept is extended for any tour t: the left point of t is the leftmost point of t that lies on the line defined by the intervals covered by t; the right point of t is defined analogously. If a point $x \in s_i$ for some $s_i \in S$, then we assume the relaxation of $x \in S$. This is important to define subsets of S as $S_{p,q} = \{x : x \in S, p \leq x \leq q\}$; then $T_{p,q}$ is the set of tours covering $S_{p,q}$. If p, q are the left and right points respectively of the tour t_i, then we define the portion of S covered by t_i as $S^i_{p,q}$, and $S - t_i$ as the part of S not covered by t_i. Finally, we consider that t_i is a maximal tour if $l_i = L$, we term m as the minimal number of tours to cover S, and T^* is the optimal set of tours covering S. See Figure 1 for a visual explanation of some of the aforementioned definitions.

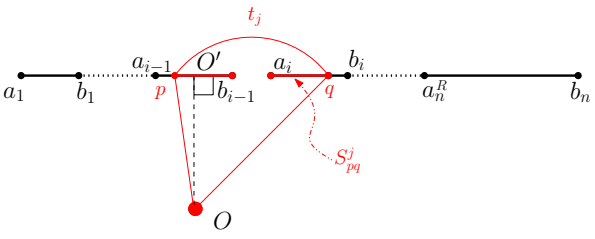


Figure 1: An example of a set S of intervals. A tour t_j (in red) is the path $OpqO$. The portion of S covered by this tour is $S^j_{pq} = [p, b_{i-1}] \cup [a_i, q]$.

3 Minimizing the number of tours

In this section, we show that the 1DMinTours-problem can be solved using the following greedy approach. For a tour $t = OpqO$, let $S - t$ be the closure of part of S not covered by t.

Greedy Strategy (GS): *Let f be the farthest point from O in S. If S can be covered by one tour, perform a minimal length tour t covering S, else perform a maximal tour t covering f and update $S := S - t$.*

In the following, we prove that GS retrieves an optimal solution (it is easy to see that the optimal solution is not necessarily unique).

Theorem 1 *GS computes an optimal solution for minimizing the number of tours.*

Proof. Proof by induction on m, the minimum number of tours.

Base Case: If all segments of S can be covered with one tour, the greedy algorithm computes only one tour using the farthest point from O.

Inductive Step: Suppose that the minimum number of tours covering a set S is at least two, i.e. $m \geq 2$. Let $f \in S$ be the farthest point from O and let t_f be the maximal tour covering f. Assume w.l.o.g. that $f = b_n$ (the proof is analogous if $f = a_1$). Let $T^* = \{t_1, t_2, \ldots, t_m\}$, t_m be an optimal solution such that the tour t_m reaches f. Let S^* be the set of points in S covered by tours $t_1, t_2, \ldots, t_{m-1}$. Let S' be the set of points in S not covered by tour t_f. Since t_f is maximal, $S' \subseteq S^*$. Then S' can be covered by $m - 1$ tours (for example, $t_1, t_2, \ldots, t_{m-1}$). By the induction hypothesis, the greedy algorithm covers S' by at most $m-1$ tours. Therefore the greedy algorithm computes at most m covering tours for S. Since m is the minimum number of tours covering a set S, the number of tours computed by the greedy algorithm is exactly m. □

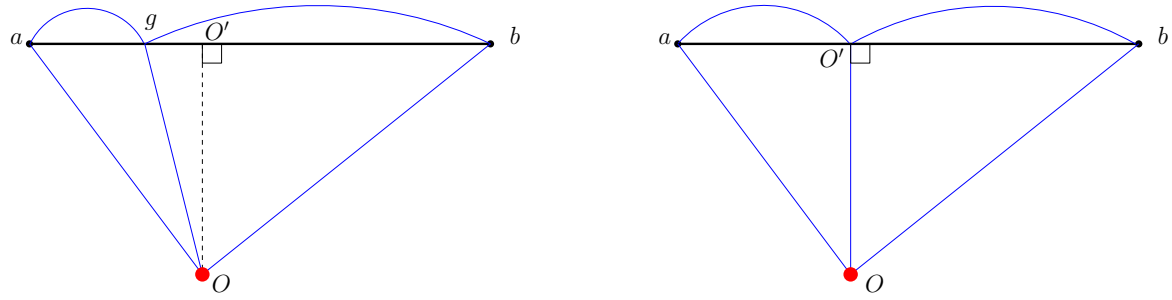
Theorem 2 *The 1DMinTours-problem can be solved in $O(m \log n)$ or $O(m + n)$, where n is the number of segments and m is the minimal number of tours.*

4 Minimizing the total distance

4.1 One segment

First, note that GS is not optimal for minimizing the total distance, even for restricted scenarios where only one segment is considered (i.e. $n = 1$). This is the case of Figure 2, where the solution provided by GS (shown in (a)) is worst than the solution (shown in (b)). We extend GS to optimally solve the 1DMinDistance-problem for only one segment.

Greedy Strategy with Projection Point (GSP): *Go to the untraveled point of S farthest from O, and perform a maximal tour while the projection point O' is not reached. If possible, cover the last part with one tour; otherwise, select the two tours containing the projection point.*



(a) Greedy solution.

(b) Coming to the base on the projection point.

Figure 2: GS is not optimal to cover segment $[a, b]$ when the optimal set of tours for minimizing the total distance includes the projection point O'.

Theorem 3 *GSP is optimal for the 1DMinDistance-problem with only one segment.*

Proof. The optimal solution for minimizing the total distance in one segment has the particularity that the tours that do not include O' have to be of maximum length. Otherwise, for a tour of non-maximum length, we can change its returning point for a point closer to O that is also valid but with a lower distance to the base. On the other hand, in the optimal solution (this is unique), the projection point O' can be covered by one or two tours (since points in S are covered at a maximum of two times). GSP is optimal since it uses the aforementioned characterizations to build the solution. First, GSP extracts the maximum-length tours that do not contain O'. Finally, it checks if the rest of the segment can be covered with one tour (optimal), or if we need two. In the second case, the optimal partition of the segment uses the projection point, as this is the closest distance from O to S. □

Theorem 4 *1DMinDistance-problem for one segment can be solved in $O(m)$ where m is the number of tours of the optimal solution.*

For two o more segments, the greedy approach does not solve the problem. The reason is that each gap between two segments poses a decision problem: covering it with a tour of maximum length (when it is possible) or finishing the tour at the end of a segment and start a new tour from the next one.

4.2 Segments to one side

In this letter, we only show how to solve the scenario where all segments are on one side of the projection point. The general case can be solved by using an extension of this case. Formally, we call 1DMD-one-side-problem to the 1DMinDistance-problem, with an additional restriction: either $0 \leq a_1$ or $a_n \leq 0$. Without loss of generality, we consider the case where $0 \leq a_1$.

Let us built a discrete set using the following approach: For every b_i, we consider the set of points C_i defined by the jumps of the greedy solution starting at b_i and continuing until a gap is reached, or all the segments are covered (Figure 3); each C_i contains at most m points. Let $C = \bigcup C_i, i \in [1 \dots n]$, be the set of candidate points defined with this strategy that contains, at most, nm points.

Lemma 5 *The right point q of any tour t_j in the optimal solution T^* satisfies $q \in C$.*

Proof. Assume $t_m \in T^*$ as the last tour with the rightmost point not in C. Let q be the right point of t_m and $s_i = [a_i, b_i]$ the segment where q lies; then $q \in (a_i, b_i)$. Hence, the tour t_{m+1} with leftmost point q, has a length lower than L because the rightmost point of t_{m+1} is in C. As $a_i < q$, we can increase the length of t_{m+1}, hence reducing the total distance. This contradicts that T^* is optimal. □

Lemma 6 *The left point p of any tour t_j in the optimal solution T^* satisfies that p is the left point of some interval of S, or t_j is maximal.*

Proof. Assume $t_i \in T^*$ as a non-maximum length tour with the left point $p \in (a_i, b_i]$ for some interval in S. Then, we can increase t_i by moving p left towards, reducing the distance from the base station to p. This contradicts the optimality of T^*. Hence tours of non-maximal length lie only at the left point of some interval of S. □

As a consequence of Lemma 6, it is straightforward to notice that the left point of a tour in the optimal solution is the left point of a segment, or is in C. Using this fact and Lemma 5, we design a polynomial algorithm based on dynamic programming. Our algorithm iterates over the sorted points of C, in ascending order. For every point $c_k \in C$, we compute the maximum-length tour starting on it, and its associated left point c'_k. We know that either $c'_k \in C$, or is the left point of some interval of S. Let j_k (j'_k) be the index of the

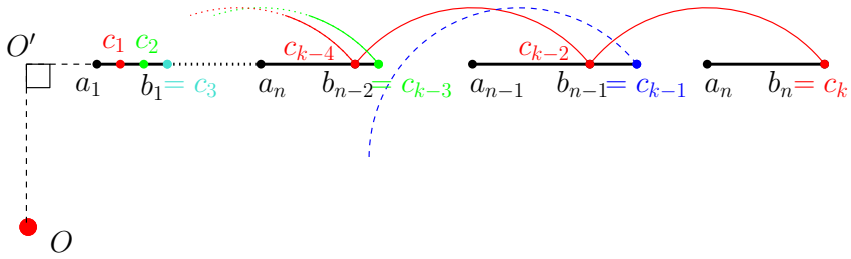


Figure 3: The one side case. Construction of the candidate set C.

segment where c_k (c'_k) is located, and be $\Sigma^*(c_k)$ the optimal cost for S_{a_1,c_k}. The formula for any c_k is as follows:

$$\Sigma^*(c_k) = \begin{cases} \text{len}(a_1, c_k) & \text{if } a_1 = c'_k \\ \min\limits_{j'_k \le j \le j_k} \{\text{len}(a_j, c_k) + \Sigma^*(b_{j-1}))\} & \text{if } c'_k \notin C \\ \min\{L + \Sigma^*(c'_k), \min\limits_{j'_k < j \le j_k} \{\text{len}(a_j, c_k) + \Sigma^*(b_{j-1})\}\}, & \text{otherwise,} \end{cases} \tag{1}$$

where $\text{len}(a_j, c_k)$ is the length of the tour that defines the interval S_{a_j,c_k}; and a_j (b_j) is the left (right) point of any segment contained within the maximum-length tour starting at c_k. A maximum of $n-1$ values of a_j needs to be checked for every c_k; one for every gap. We term the algorithm based on the formula 1 as DPOS (Dynamic Programming on One Side). As a consequence of Lemmas 5 and 6, we have:

Theorem 7 *DPOS is optimal for the 1DMD-one-side-problem.*

Theorem 8 *The 1DMD-one-side problem can be solved in $O(n^2) + O(nm)$, where n is the number of segments and m is the number of tours in the optimal solution.*

Acknowledgments

This work is partially supported by grants PID2020-114154RB-I00 and TED2021-129182B-I00 funded by MCIN/AEI/10.13039/501100011033 and the European Union NextGenerationEU/PRTR.

References

[1] Saurav Agarwal and Srinivas Akella. Line coverage with multiple robots. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3248–3254. IEEE, 2020.

[2] Sergey Bereg, Luis Evaristo Caraballo, and José Miguel Díaz-Báñez. Efficient inspection of underground galleries using k robots with limited energy. In *Iberian Robotics conference*, pages 706–717. Springer, 2017.

[3] Alvaro Calvo, Giuseppe Silano, and Jesús Capitán. Mission planning and execution in heterogeneous teams of aerial robots supporting power line inspection operations. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1644–1649. IEEE, 2022.

[4] Tucker Farrell, Kidus Guye, Rebecca Mitchell, and Guangdong Zhu. A non-intrusive optical approach to characterize heliostats in utility-scale power tower plants: Flight path generation/optimization of unmanned aerial systems. *Solar Energy*, 225:784–801, 2021.

[5] Mahdi Jemmali, Ali Kashif Bashir, Wadii Boulila, Loai Kayed B Melhim, Rutvij H Jhaveri, and Jawad Ahmad. An efficient optimization of battery-drone-based transportation systems for monitoring solar power plant. *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[6] František Nekovář, Jan Faigl, and Martin Saska. Multi-tour set traveling salesman problem in planning power transmission line inspection. *IEEE Robotics and Automation Letters*, 6(4):6196–6203, 2021.

[7] Joon-Young Park, Seok-Tae Kim, Jae-Kyung Lee, Ji-Wan Ham, and Ki-Yong Oh. Method of operating a gis-based autopilot drone to inspect ultrahigh voltage power lines and its field tests. *Journal of Field Robotics*, 37(3):345–361, 2020.

Computing k-Crossing Visibility through k-levels

Frank Duque*[1]

[1]Escuela de Matemáticas, Universidad Nacional de Colombia, Medellín, Colombia

1 Introduction

The notion of visibility has been used extensively in Computational Geometry, in the context of the art gallery problem [13, 11]. The development of wireless network connections has motivated the study of a new kind of visibility, where the line of visibility can cross k obstacles [1].

Let $\mathcal{A}$ be an arrangement of straight lines and segments in $\mathbb{R}^2$ (or planes in $\mathbb{R}^3$). The k-crossing visibility on $\mathcal{A}$ of a point p, denoted by $\mathcal{V}_k(p, \mathcal{A})$, is the set of points q on elements of $\mathcal{A}$ such that the segment pq intersects at most k elements of $\mathcal{A}$. See Figure 1.

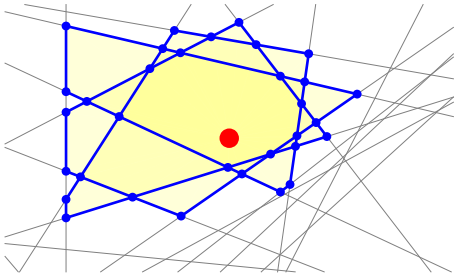


Figure 1: The blue points and segments illustrate the 2-crossing visibility of the red point on an arrangement of lines.

Some early works on k-crossing visibility are [12, 9, 5]. In [4] recently Bahoo et al. introduced an algorithm that computes $\mathcal{V}_k(p, \mathcal{A})$ in $O(kn)$-time, where $\mathcal{A}$ consists of the edges of a polygon.

Theorem 1 (Bahoo et al. [4]) *Given a simple polygon P with n vertices and a query point p in P, the region of P that is k-crossing visible from p, can be computed in $O(kn)$ time.*

In this work we obtain another proof of Theorem 1 and we prove Theorem 2, Theorem 4, Proposition 3 and Proposition 5.

*Email: frduquep@unal.edu.co. Research supported by the Universidad Nacional de Colombia research, grant HERMES-58357.

Theorem 2 *Let $\mathcal{A}$ be an arrangement of n lines in the plane, and let p be a query point. Then $\mathcal{V}_k(p, \mathcal{A})$ can be computed in $O(n \log n + kn)$ time.*

Given an arrangement $\mathcal{A}$ of straight lines, rays and segments in the plane (or planes in $\mathbb{R}^3$), the combinatorial complexity of $\mathcal{A}$, is the total number of vertices and edges (and faces) defined by $\mathcal{A}$.

Proposition 3 *The maximum combinatorial complexity of the k-crossing visibility on arrangements of n straight lines in the plane is $\theta(kn)$.*

Theorem 4 *Let $\mathcal{A}$ be an arrangement of n planes in $\mathbb{R}^3$, and let p be a query point. Then $\mathcal{V}_k(p, \mathcal{A})$ can be computed in $O(n \log n + k^2 n)$ expected time.*

Proposition 5 *The maximum combinatorial complexity of the k-crossing visibility on arrangements of n planes in $\mathbb{R}^3$ is $\theta(k^2 n)$.*

Note that, by Proposition 3 and Proposition 5, Theorem 2 and Theorem 4 are optimal for $k = \Omega(\log n)$ and $k = \Omega(\sqrt{\log n})$, respectively.

Given an arrangement $\mathcal{A}$ of objects in $\mathbb{R}^2$ (or $\mathbb{R}^3$), the $(\leq k)$-level-region of $\mathcal{A}$ is the set of points in $\mathbb{R}^2$ (or $\mathbb{R}^3$) with at most k elements of $\mathcal{A}$ lying above it. In the following we denote by $(\leq k)level(\mathcal{A})$, to the portion of the elements of $\mathcal{A}$ that are in the $(\leq k)$-level-region of $\mathcal{A}$.

Let $\mathcal{T}$ be the transformation

$$\mathcal{T}((x, y)) = (x/y, 1/y) \quad \text{in the } \mathbb{R}^2 \text{ case, or}$$
$$\mathcal{T}((x, y, z)) = (x/z, y/z, 1/z) \quad \text{in the } \mathbb{R}^3 \text{ case.}$$

In this paper we obtain a linear time reduction, of the problem of obtaining $\mathcal{V}_k(p, \mathcal{A})$ to the problem of obtaining $(\leq k)level(\mathcal{A})$, by applying $\mathcal{T}$. This reduction can be easily adapted for obtaining k-crossing visibilities on another arrangements whose $(\leq k)$-level is known.

2 Results in $\mathbb{R}^2$

Let $\mathcal{D}$ be the set of points $(x, y) \in \mathbb{R}^2$ such that $y \neq 0$. Throughout this section, O denotes the point $(0, 0) \in \mathbb{R}^2$ and $\mathcal{T}$ denotes the transformation $\mathcal{T} : \mathcal{D} \to \mathcal{D}$ such that $\mathcal{T}((x, y)) = (x/y, 1/y)$.

In this section, we prove that $\mathcal{T}$ determines a bridge between k-crossing visibility and $(\leq k)$-levels in $\mathbb{R}^2$. Then, we use this result to prove Theorem 1, Theorem 2 and Proposition 3.

2.1 Properties of $\mathcal{T}$

Given $D \subset \mathcal{D}$ we denote by $\mathcal{T}[D]$ the image of D under $\mathcal{T}$. We also denote by $\mathcal{T}[\mathcal{A}]$ the set images of the elements of $\mathcal{A}$ under $\mathcal{T}$. In this section, we first prove several properties of $\mathcal{T}$. Then, we determine $\mathcal{T}[D]$ for different instances of D. Finally, we prove that $\mathcal{V}_k(O, \mathcal{A})$ can be obtained from $(\leq k)$*level* $(\mathcal{T}[\mathcal{A}])$.

Proposition 6 *$\mathcal{T}$ is self-inverse.*

Proof. $\mathcal{T} o \mathcal{T}((x,y)) = \mathcal{T}((x/y, 1/y)) = (x,y)$. □

Proposition 7 *$\mathcal{T}$ sends straight lines to straight lines. More precisely, if L is the straight line in $\mathcal{D}$ with equation $ax+by+c=0$ then $\mathcal{T}[L]$ is the straight line in $\mathcal{D}$ with equation $ax+cy+b=0$.*

Proof. Let L' be the straight line with equation $ax+cy+b=0$. If (x_0,y_0) is in L then $ax_0+by_0+c=0$; thus, as $a\frac{x_0}{y_0}+c\frac{1}{y_0}+b=0$, then $\mathcal{T}(x_0,y_0)$ is in L'. Similarly, if (x_0,y_0) is in L' then $\mathcal{T}^{-1}(x_0,y_0) = \mathcal{T}(x_0,y_0)$ is in L. □

Proposition 8 *$\mathcal{T}$ preserve incidences between points and lines. More precisely the point p is in the straight line L if and only if $\mathcal{T}(p)$ is in the straight line $\mathcal{T}[L]$.*

Proof. Let $p=(x_0,y_0)$ be a point in $\mathcal{D}$ and let $L: ax+by+c=0$ be a straight line in $\mathcal{D}$. This proof follows from the fact that (x_0,y_0) satisfies $ax+by+c=0$ if and only if $\left(\frac{x_0}{y_0}, \frac{1}{y_0}\right)$ satisfies $\mathcal{T}[L]: ax+cy+b=0$. □

Given a line L in $\mathcal{D}$, we denote by L^+ the set of points in L whose second coordinate is greater than zero, and we denote by L^- the set of points in L whose second coordinate is less than zero.

Proposition 9 *Let L be a straight line in $\mathcal{D}$. Then $\mathcal{T}[L^+] = \mathcal{T}[L]^+$ and $\mathcal{T}[L^-] = \mathcal{T}[L]^-$. Moreover, If $p_1, p_2, \ldots, p_k$ are in L^+ (or they are in L^-) ordered by their distance to the x-axis from the closest to the furthest, then $\mathcal{T}(p_1), \mathcal{T}(p_2), \ldots, \mathcal{T}(p_k)$ are in $\mathcal{T}[L^+]$ (or they are in $\mathcal{T}[L^-]$, respectively), ordered by their distance to the x-axis from the furthest to the closest.*

Proof. As $\mathcal{T}$ sends straight lines to straight lines and it does not change the sign of the second coordinate, then $\mathcal{T}[L^+] = \mathcal{T}[L]^+$ and $\mathcal{T}[L^-] = \mathcal{T}[L]^-$. If the second coordinates of p_i and p_j are y_i and y_j, respectively, then the second coordinates of $\mathcal{T}(p_i)$ and $\mathcal{T}(p_j)$ are $1/y_i$ and $1/y_j$, respectively. This proof follows from the fact that $|y_i| < |y_j|$ if and only if $|1/y_i| > |1/y_j|$. □

Let $\mathcal{D}^+$ denote the set of points in $\mathcal{D}$ whose second coordinate is greater than zero, and let $\mathcal{D}^-$ denote the set of points in $\mathcal{D}$ whose second coordinate is less than zero. The proofs of Proposition 10 and Proposition 11 follows from Proposition 9.

Proposition 10 *Let D be a line segment contained in a straight line L, whose endpoints are p and q.*

- *If both p and q are in $\mathcal{D}^+$ ($\mathcal{D}^-$), then $\mathcal{T}[D]$ is the line segment contained in $\mathcal{D}^+$ ($\mathcal{D}^-$) whose endpoints are $\mathcal{T}(p)$ and $\mathcal{T}(q)$.*
- *If p is in the x-axis and q is in $\mathcal{D}^+$ ($\mathcal{D}^-$), then $\mathcal{T}[D]$ is the ray contained in $\mathcal{D}^+$ ($\mathcal{D}^-$), defined by the straight line $\mathcal{T}[L]$ and the point $\mathcal{T}(q)$.*

Given a $D \subset \mathcal{D}$ we denote by $\overline{D}$ the closure of D in $\mathbb{R}^2$.

Proposition 11 *Let D be a no horizontal ray contained in a straight line L, whose endpoint is p.*

- *If both p and D are in $\mathcal{D}^+$ ($\mathcal{D}^-$), then $\mathcal{T}[D]$ is the line segment contained in $\mathcal{D}^+$ ($\mathcal{D}^-$), whose endpoints are $\mathcal{T}(p)$ and the intersection of $\overline{\mathcal{T}[L]}$ with the x-axis.*
- *If p is in the x-axis and D is in $\mathcal{D}^+$ ($\mathcal{D}^-$), then $\mathcal{T}[D]$ is the ray defined by the part of the straight line $\mathcal{T}[L]$ in $\mathcal{D}^+$ ($\mathcal{D}^-$).*

Proposition 12 *Let D be a horizontal ray contained in a straight line L whose endpoint is p. If D is contained in $\mathcal{D}^+$ ($\mathcal{D}^-$), then $\mathcal{T}[D]$ is the horizontal ray in $\mathcal{D}^+$ ($\mathcal{D}^-$), defined by the straight line $\mathcal{T}[L]$ and the point $\mathcal{T}(p)$. If D is contained in $\mathcal{D}^+$, then D and $\mathcal{T}[D]$ have the same direction; in the other case, D and $\mathcal{T}[D]$ have opposite direction.*

Proof. If L has equation $by+c=0$ then $\mathcal{T}[L]$ is the horizontal line with equation $cy+b=0$. □

From Proposition 7, Proposition 10, Proposition 11 and Proposition 12 we conclude that: If D is a straight line, ray or segment contained in $\mathcal{D}^+$ ($\mathcal{D}^-$) then $\mathcal{T}[D]$ is a straight line, ray or segment contained in $\mathcal{D}^+$ ($\mathcal{D}^-$).

Proposition 13 *Let L be a straight line in $\mathcal{D}$. Then $O \in \overline{L}$ if and only if $\mathcal{T}[L]$ is a vertical line.*

Proof. This proof follows from the fact that $\overline{L}$ has equation $ax+by=0$ if and only if $\mathcal{T}[L]$ has equation $ax+b=0$ □

Let $\mathcal{V}_k^+(O, \mathcal{A})$ denote the portions of the elements of $\mathcal{V}_k(O, \mathcal{A})$ in $\mathcal{D}^+$, *i.e.*

$$\mathcal{V}_k^+(O, \mathcal{A}) = \left\{ D \cap \mathcal{D}^+ : D \in \mathcal{V}_k(O, \mathcal{A}) \right\}$$

Similarly, let $\mathcal{V}_k^-(O,\mathcal{A})$ denote the portions of $\mathcal{V}_k(O,\mathcal{A})$ in $\mathcal{D}^-$, *i.e.*

$$\mathcal{V}_k^-(O,\mathcal{A}) = \left\{D \cap \mathcal{D}^- : D \in \mathcal{V}_k(O,\mathcal{A})\right\}$$

Let $(\leq k)level^+(\mathcal{A})$ denote the portion of the elements of $(\leq k)level(\mathcal{A})$ in $\mathcal{D}^+$, *i.e.*

$$(\leq k)level^+(\mathcal{A}) = \left\{D \cap \mathcal{D}^+ : D \in (\leq k)level(\mathcal{A})\right\}$$

The $(\leq k)$-lower-level-region of $\mathcal{A}$ is the set of points in $\mathbb{R}^2$ ($\mathbb{R}^3$) with at most k elements of $\mathcal{A}$ lying below it. Let $(\leq k)level^-(\mathcal{A})$ denote the portion of the elements of $\mathcal{A}$ in both $\mathcal{D}^-$ and the $(\leq k)$-lower-level-region of $\mathcal{A}$.

Lemma 14 *Let $\mathcal{A}$ be an arrangement of straight lines, segments or rays. Then:*

1. $\mathcal{V}_k^+(O,\mathcal{A}) = \mathcal{T}[(\leq k)level^+(\mathcal{T}[\mathcal{A}])]$.
2. $\mathcal{V}_k^-(O,\mathcal{A}) = \mathcal{T}[(\leq k)level^-(\mathcal{T}[\mathcal{A}])]$.

Proof. We prove 1, the proof of 2 is similar.

Let $p \in \mathcal{D}^+$ be such that $p \in D$ for some $D \in \mathcal{A}$, and let L be the line that contains p and O. Then $p \in L^+$, $\mathcal{T}[D] \in \mathcal{T}[\mathcal{A}]$ and $\mathcal{T}(p) \in \mathcal{T}[D]$. As $\mathcal{T}$ preserves incidences, by Proposition 13 and Proposition 9, the line segment between O and p crosses at most k elements of $\mathcal{A}$, if and only if, there are at most k elements of $\mathcal{T}[\mathcal{A}]$ laying above $\mathcal{T}(p)$. □

2.2 Proofs of results in $\mathbb{R}^2$

Theorem 15 (Everett et al. [10]) *Let $\mathcal{A}$ be an arrangement of n lines in the plane. Then $(\leq k)level(\mathcal{A})$ can be computed in $O(n\log n + kn)$ time.*

We use Theorem 15 in order to prove Theorem 2.

Proof. [Proof of Theorem 2] Without loss of generality, we may assume that p is at the origin, otherwise p and the elements of $\mathcal{A}$ can be translated. We also may assume that the x-axis does not contain an element of $\mathcal{A}$ or an intersection between two elements of $\mathcal{A}$, otherwise, the elements of $\mathcal{A}$ can be rotated.

By Proposition 7, $\mathcal{T}[\mathcal{A}]$ is an arrangement of n straight lines. Thus, as the k-crossing visibility of O on $\mathcal{A}$ can be obtained from $\mathcal{V}_k^+(O,\mathcal{A})$ and $\mathcal{V}_k^-(O,\mathcal{A})$, this proof follows from Lemma 14 and Theorem 15. □

Let $\mathcal{A}$ be an arrangement of straight lines, rays and segments. The vertical decomposition (also known as trapezoidal decomposition) of $\mathcal{A}$ is obtained by erecting vertical segments upwards and downwards from each vertex in $\mathcal{A}$ and extend them until they meet another line or all the way to infinity.

Lemma 16 *Let $\mathcal{A}$ be an arrangement of n straight lines, rays and segments. Then $(\leq k)level(\mathcal{A})$ can be obtained from a vertical decomposition of $\mathcal{A}$ in $O(kn)$ time.*

Proof. Suppose that the vertical decomposition of A is known. Then for each vertex, extend a vertical segment upwards until it reaches $k+1$ elements of $\mathcal{A}$ or its way to infinity; such vertex is in $(\leq k)level(\mathcal{A})$ if and only if the vertical segment reaches its way to infinity. □

Proof. [Another proof of Theorem 1] As in the proof of Theorem 2, we may assume that p is at the origin and the x-axis does not contain edges of P. By Proposition 10, $\mathcal{T}[P]$ is an arrangement of at most $2n$ line segments or rays. Thus, as the k-crossing visibility of O on P can be obtained from $\mathcal{V}_k^+(O,P)$ and $\mathcal{V}_k^-(O,P)$, by Lemma 14 and Lemma 16, it is enough to obtain the vertical decomposition of $\mathcal{T}[P]\cap\mathcal{D}^+$ and $\mathcal{T}[P]\cap\mathcal{D}^-$ in linear time; we do this for $\mathcal{T}[P]\cap\mathcal{D}^+$, the other case is similar.

Let $L: y+c=0$ be a horizontal line, high enough that all the endpoints of $\mathcal{T}[P]\cap\mathcal{D}^+$ are below L. Let $L' = \mathcal{T}[L]$ and note that L' is a horizontal line with equation $cy+1=0$. Suppose that the points in P above L' are blue and the others are red. Let P' be the polygon in $\mathcal{D}^+$ obtained from P by scaling vertically its red part, keeping the endpoints on L' fixed.

In [7] Chazelle prove that the vertical decomposition of a polygon can be computed in linear time (see also Amato et al. [3]). Thus, as P' is contained in $\mathcal{D}^+$, by Proposition 10 $\mathcal{T}[P']$ is a polygon, and the vertical decomposition of $\mathcal{T}[P']$ can be computed in linear time. Note that a point $\mathcal{T}(q)$ in $\mathcal{T}[P]\cap\mathcal{D}^+$ is below L if and only if q is blue. Thus the vertical decomposition of $\mathcal{T}[P]\cap\mathcal{D}^+$ can be obtained from the vertical decomposition of $\mathcal{T}[P']$ below L. □

Proof. [Proof of Proposition 3] In [2] Alon et al. prove that the maximum combinatorial complexity of the $(\leq k)$-level on arrangements of n straight lines in the plane is $\theta(nk)$. Without loss of generality, suppose that the arrangement $\mathcal{A}$ reaches this bound and the $(\leq k)$-level of $\mathcal{A}$ is contained in $\mathcal{D}^+$. Thus, the combinatorial complexity of $(\leq k)level^+(\mathcal{A})$ is $\theta(nk)$ and by Lemma 14 the combinatorial complexity of $\mathcal{V}_k^+(O,\mathcal{T}[\mathcal{A}])$ is also $\theta(nk)$. □

3 Results in $\mathbb{R}^3$

Let $\mathcal{D}$ be the set of points $(x,y,z) \in \mathbb{R}^3$ such that $z \neq 0$. Throughout this section, O denotes the point $(0,0,0) \in \mathbb{R}^3$ and $\mathcal{T}$ denotes the transformation $\mathcal{T}: \mathcal{D} \to \mathcal{D}$ be such that

$$\mathcal{T}((x,y,z)) = (x/z, y/z, 1/z).$$

The proofs of Proposition 17, Proposition 18, Proposition 19, Proposition 20, Proposition 21 and Lemma 22, can be obtained as in Section 2.

Proposition 17 *$\mathcal{T}$ is self-inverse.*

Proposition 18 *$\mathcal{T}$ sends planes to planes. More precisely, if π is the plane in $\mathcal{D}$ with equation $ax + by + cz + d = 0$ then $\mathcal{T}[\pi]$ is the plane in $\mathcal{D}$ with equation $ax + by + dz + c = 0$.*

Proposition 19 *$\mathcal{T}$ preserve incidences between points and planes. More precisely the point p is in the plane π if and only if $\mathcal{T}(p)$ is in the plane $\mathcal{T}[\pi]$.*

Given a plane π in $\mathcal{D}$, we denote by π^+ the set of points in π whose third coordinate is greater than zero, and we denote by π^- the set of points in π whose third coordinate is less than zero.

Proposition 20 *Let π be a plane in $\mathcal{D}$. Then $\mathcal{T}[\pi^+] = \mathcal{T}[\pi]^+$ and $\mathcal{T}[\pi^-] = \mathcal{T}[\pi]^-$. Moreover, If $p_1, p_2, \ldots, p_k$ are in π^+ (or they are in π^-) ordered by their distance to the plane $z = 0$ from the closest to the furthest, then $\mathcal{T}(p_1), \mathcal{T}(p_2), \ldots, \mathcal{T}(p_k)$ are in $\mathcal{T}[\pi^+]$ (or they are in $\mathcal{T}[\pi^-]$, respectively), ordered by their distance to the plane $z = 0$ from the furthest to the closest.*

Given a $D \subset \mathcal{D}$ we denote by $\overline{D}$ the closure of D in $\mathbb{R}^3$.

Proposition 21 *Let L be a straight line in $\mathcal{D}$. Then $O \in \overline{L}$ if and only if $\mathcal{T}[L]$ is a vertical line.*

Lemma 22 *Let $\mathcal{A}$ be an arrangement of planes. Then:*

1. $\mathcal{V}_k^+(O, \mathcal{A}) = \mathcal{T}[(\leq k)level^+(\mathcal{T}[\mathcal{A}])]$.
2. $\mathcal{V}_k^-(O, \mathcal{A}) = \mathcal{T}[(\leq k)level^-(\mathcal{T}[\mathcal{A}])]$.

The proofs of Theorem 4 and Proposition 5 follows from Theorem 23 and Theorem 24, in a similar way as in the proof of Theorem 2 and the proof of Proposition 3 in Section 2.

Theorem 23 (Chan [6]) *Let $\mathcal{A}$ be an arrangement of n planes in $\mathbb{R}^3$. Then $(\leq k)level(\mathcal{A})$ can be computed in $O(n \log n + k^2 n)$ expected time.*

Theorem 24 (Clarkson et al. [8]) *Let $k \geq 1$. Then the maximum combinatorial complexity of $(\leq k)$-level on arrangements of n hyperplanes in $\mathbb{R}^d$ is $\theta\left(n^{\lfloor d/2 \rfloor} k^{\lceil d/2 \rceil}\right)$.*

References

[1] Oswin Aichholzer, Ruy Fabila-Monroy, David Flores-Peñaloza, Thomas Hackl, Jorge Urrutia, and Birgit Vogtenhuber. Modem illumination of monotone polygons. *Computational Geometry*, 68:101–118, 2018.

[2] Noga Alon and Ervin Györi. The number of small semispaces of a finite set of points in the plane. *Journal of Combinatorial Theory, Series A*, 41(1):154–157, 1986.

[3] Nancy M Amato, Michael T Goodrich, and Edgar A Ramos. A randomized algorithm for triangulating a simple polygon in linear time. *Discrete & Computational Geometry*, 26(2):245–265, 2001.

[4] Yeganeh Bahoo, Prosenjit Bose, Stephane Durocher, and Thomas Shermer. Computing the k-crossing visibility region of a point in a polygon. In Charles J. Colbourn, Roberto Grossi, and Nadia Pisanti, editors, *Combinatorial Algorithms*, pages 10–21, Cham, 2019. Springer International Publishing.

[5] Antonio Bajuelos, Santiago Canales, Gregorio Hernández, and Mafalda Martins. A hybrid metaheuristic strategy for covering with wireless devices. *J. Univers. Comput. Sci.*, 18(14):1906–1932, 2012.

[6] Timothy Chan. Random sampling, halfspace range reporting, and construction of $(\leq k)$-levels in three dimensions. *SIAM Journal on Computing*, 30(2):561–575, 2000.

[7] Bernard Chazelle. Triangulating a simple polygon in linear time. *Discrete & Computational Geometry*, 6(3):485–524, 1991.

[8] Kenneth Clarkson and Peter Shor. Applications of random sampling in computational geometry, ii. *Discrete & Computational Geometry*, 4(5):387–421, 1989.

[9] James Dean, Andrzej Lingas, and Jörg-Rüdiger Sack. Recognizing polygons, or how to spy. *The Visual Computer*, 3(6):344–355, 1988.

[10] Hazel Everett, Jean-Marc Robert, and Marc van Kreveld. An optimal algorithm for the $(\leq k)$-levels, with applications to separation and transversal problems. *International Journal of Computational Geometry Applications*, 06(03):247–261, 1996.

[11] Subir Kumar Ghosh. *Visibility algorithms in the plane*. Cambridge university press, 2007.

[12] Naji Mouawad and Thomas Shermer. The superman problem. *The Visual Computer*, 10(8):459–473, 1994.

[13] Joseph O'Rourke et al. *Art gallery theorems and algorithms*, volume 57. Oxford University Press Oxford, 1987.

Algorithms for efficient solar tracking in CSP plants

J.M Díaz-Báñez*[1], J.M Higes-López†[1], M.A Pérez-Cutiño‡[1,2], and J.S Valverde-García§[1,2]

[1]Department of Applied Mathematics, University of Seville, SPAIN
[2]Virtualmech S.L, Seville, SPAIN

Abstract

Increasing the efficiency in green energy production is mandatory to reduce dependency on fossil fuels. Capturing and storing solar energy is an appealing alternative, but optimizing energy collection with no damage to components of solar plants is a complex problem. In this work, some geometric optimization problems for solar tracking in Concentrated Solar Power plants based on Parabolic Through Collectors are addressed. Using nice properties of a solution, we propose efficient algorithms for optimal scheduling in solar tracking tasks that can be adapted for other solar plants.

1 Introduction

Concentrated Solar Power (CSP) plants are an effective alternative to photovoltaic technologies, as it has the capacity of storing the energy captured from the sun. Parabolic Trough Collectors (PTC) systems are one of the most widespread CSP plants around the globe, including more than 40 plants in Spain alone. PTC systems are composed of a parabolic-shaped surface reflecting the sun rays to a Heat Collector Element (HCE) located at the focus of the parabola. The parabolic-shaped mirror surface together with three HCEs forms a Solar Collector Element (SCE), and 4 SCEs are a Solar Collector Assembly (SCA). For a full decomposition of elements in the solar field of PTC plants, the reader is referred to [1].

During normal operation of PTC plants, SCAs are instructed to follow the sun so that the maximum energy can be collected, see Figure 1. Providing tracking systems to simultaneously improve accuracy and reduce operational cost is a seminal research area in solar plants. Methods for optimizing trackers in plants with arbitrary design and geometry have been proposed in the area of renewable energy [4, 2, 3]. When the operating conditions are optimal, a perfect tracking of the sun results in maximal energy collection.

Considering this scenario, the ray incidence over the HCE for different SCA and solar angles is expected to be unimodal. However, the shape of the function can change due to several errors, such as installation errors of some components of the SCA, cracks/dirt in the mirror surface, HCE bending and vertical/horizontal displacements due to mechanical stress, among others. This work raises some geometric problems to optimize the tracking system considering any shape for the ray incidence function. To the best of our knowledge, we are the first considering the optimization of the solar tracking while reducing the movements of the SCA in a PTC plant.

The rest of the paper is organized as follows: Section 2 provides the necessary background and the definition of the optimization problems; the algorithms are outlined in Sections 3 and 4.

2 Preliminaries

For this initial study, we assume that the weather conditions are constant throughout the day. Thus, *solar irradiance* over the HCE can be expressed as a function $z = f(x, y)$, where the (x, y) coordinates represent the Solar Collector Assembly and the sun angular displacements, respectively, while z corresponds to the number of rays touching the HCE. Since there is no change in the initial conditions, the 3D surface corresponding to f can be interpreted as a shifted 2D curve as illustrated in Figure 2. This visualization allow us to redefine the function as $z = f(\theta)$, where θ represents the difference between the sun and the SCA angular position. Solar tracking is discrete in PTC plants; hence, f can be defined as a step function with n steps as follows:

$$f(\theta) = \sum_{i=1}^{n} \alpha_i \delta_{S_i}(\theta), \tag{1}$$

where α_i is the number of rays touching the HCE in the step $S_i \in f$, and δ_{S_i} is a binary function indicating if $\theta \in S_i$.

Using a ray-tracing software, f can be obtained by moving the sun in a fixed axis with the SCA at 90^{o}. The events at which the sun rays start/end intersecting the HCE can occur at any angular difference between the sun and the SCA position; hence, the length of the steps in f can be a real number. However, in this

*Email: dbanez@us.es
†Email: jhiges@us.es
‡Email: m.perez@virtualmech.com
§Email: jvalverde@us.es

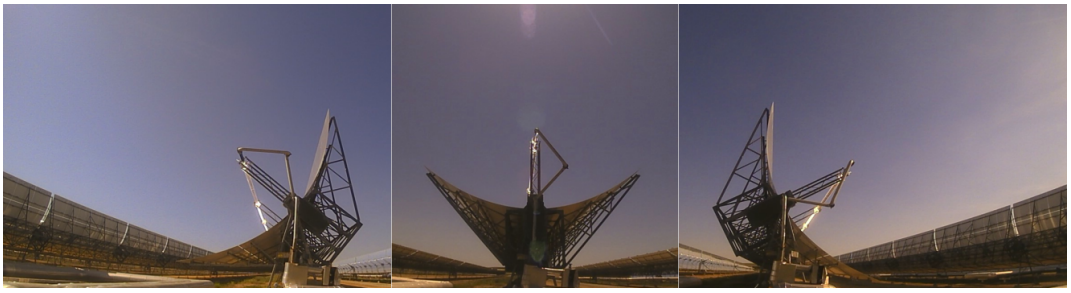


Figure 1: Solar tracking example in a real PTC plant.

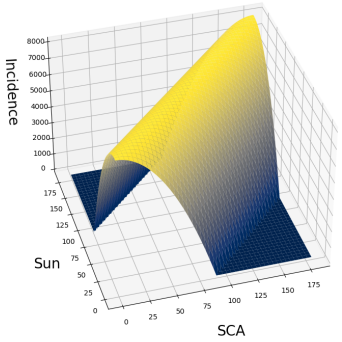


Figure 2: Ray incidence over the HCE depending on the angular position of the sun and the SCA for a simple case, assuming that the SCE has a perfect parabolic shape.

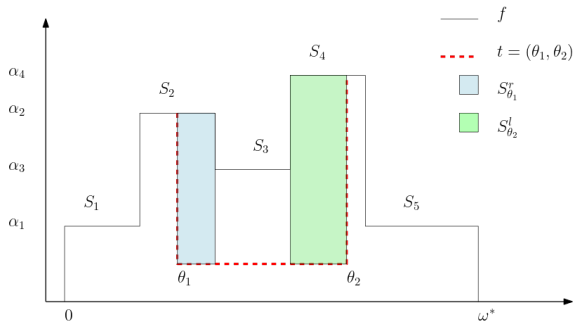


Figure 3: Main elements defining the ray incidence function f. t is a tracking interval from θ_1 to θ_2. Total irradiance in t (g_t) is the area within t below the curve. We consider f shifted to the range $(0, \omega^*)$.

abstract we consider the case in which these numbers are approximated as rational numbers (e.g., accurate to within one thousandth of a unit). This is standard in real-world applications, and involves the computational representation of real numbers. In the rational case, the problem can be reduced to one in which the solar irradiance function has steps with integer length.

2.1 The problems

Let f:$\mathbb{R} \to \mathbb{N}$ be a step function with n steps, defined as in equation 1. Let $S = \{S_1, \cdots, S_n\}$ be the set of steps of f. The elements of S are disjoint, ordered by x, and there is no gap between consecutive elements. A step $S_i \in S$ is an interval of the form $[\theta_{i_1}, \theta_{i_2})$, where $\theta_{i_1}, \theta_{i_2}$ are the edges of the step; let E be the set containing all the edges of S. The ray incidence over a step S_i is defined by α_{S_i}, its length as $l_{S_i} \in \mathbb{N}$, and the associated gain as $g_{S_i} = \alpha_{S_i} l_{S_i}$. For convenience, if $\theta \in S_i$, then $\theta \in S$, $\alpha_\theta = \alpha_{S_i}$, and S^l_θ (S^r_θ) is the portion of S_i from its left (right) edge to θ. In addition, for a tracking interval $t = (\theta_i, \theta_j)$ s.t. $\theta_i \in S_i$, $\theta_j \in S_j$, and $i \leq j$, the length of t is $l_t = \theta_j - \theta_i$, and its total gain can be defined by:

$$g_t = \begin{cases} g_{S^r_{\theta_i}} + g_{S^l_{\theta_j}} + \sum_{k=i+1}^{j-1} g_{S_k} & j > i \\ g_{S^r_{\theta_i}} - g_{S^r_{\theta_j}} & i = j \end{cases} \quad (2)$$

For a multiset $T = \{t_1, \ldots, t_k\}$ the total solar irradiance (gain) of the set is $I_T = \sum g_{t_i}$, and the total length is defined as $L_T = \sum l_{t_i}$. Finally, ω^* is the total length of S, and the initial position of the SCA w.r.t the sun is θ_0. See Figure 3 for an overview of the described notations.

We formulate two optimization problems of particular interest for solar tracking in CSP plants. The first problem looks for the minimum number of movements of the SCA such that the solar irradiance intersecting the HCE at any moment is preserved within a given range. The second one addresses to optimize the total solar irradiance intersecting the HCE with a limited number of allowed movements. More formally:

Problem 1 *(Min-Tracking, or MT-Problem): Given a step function f defined on $[0, \omega^*]$, and two real numbers u_1, u_2, find a set of intervals $T^* = \{t_1, \ldots, t_m\}$ of minimum size s.t. $t_i \subseteq [0, \omega^*]$, $\forall \theta \in t_i$, $u_1 \leq \alpha_\theta \leq u_2$ and $L_{T^*} + \theta_0 = \omega^*$.*

Problem 2 *(Maximal Energy Collection, or MEC-Problem): Given a step function f defined on $[0, \omega^*]$ and $m \in \mathbb{N}$, find a set of intervals $T^* = \{t_1, \ldots, t_j\}$ s.t. $t_i \subseteq [0, \omega^*]$, $|T^*| \leq m$, $L_{T^*} \leq \omega^*$, and I_{T^*} is maximal.*

3 Minimum Tracking

The analysis of Problem 1 must consider the initial position of the SCA with respect to the sun position. Two cases are possible: the SCA is in a feasible configuration, i.e. $u_1 \leq \alpha_{\theta_0} \leq u_2$; or the SCA is violating this restriction. In the former, it can be readily noticed that the optimal solution is to wait while the sun moves until a non-feasible state is reached. Therefore, we can assume, without lost of generality, that the SCA starts from a non-feasible configuration.

Theorem 1 *Let $t^* = (\theta_i, \theta_j)$ be a maximum tracking interval in f such that for any $\theta \in t^*$, $u_1 \leq \alpha_\theta \leq u_2$. If the SCA initially violates the boundary conditions, then the minimum possible cardinality of a solution that satisfies conditions of Problem 1 is $\lceil \frac{\omega^* - \theta_0}{l_{t^*}} \rceil$.*

Proof. Since the SCA violates the boundary condition, it needs to be moved to a feasible configuration. Let us assume that such feasible configuration initiates at θ_i and when the sun reaches θ_j the SCA moves again to θ_i. In such case it is clear that the SCA has rotated $\lceil \frac{\omega^* - \theta_0}{l_{t^*}} \rceil$ times. Hence $T^* = \{t^*, \ldots, t^*, \hat{t}\}$ with the size of T^* equals m and $\hat{t} \subseteq t^*$ is a feasible solution of Problem 1. Moreover, T^* is of miniminum size because $L_{T^*} = \omega^* - \theta_0$. Otherwise, if $T = \{t_1, \ldots, t_n\}$ is a feasible solution with $n < m$, then there would exist a t_i whose length is larger than the length of t^*, this contradicts the maximality of t^*. □

Corollary 2 *The MT-Problem can be solved in $O(n+m)$ time, where n is the number of steps in f and m is the size of a solution.*

Proof. The proof of Theorem 1 provides an additional insight on the optimal value when the SCA starts from a feasible configuration. If l_0 is the length of the interval in which the SCA meets the problem restrictions from the begining, then the minimum number of rotations of the SCA is $m = \lceil \frac{\omega^* - \theta_0 - l_0}{l_{t^*}} \rceil$. Finally, since the maximal interval t^* can be computed in linear time with a sweep from left to right, a greedy algorithm computes the optimal solution T^* in $O(n+m)$ time. □

4 Maximal energy collection

We say that a solar irradiance function f is unimodal if, for exactly one $i \in \{1, \ldots, n\}$, $\alpha_j \leq \alpha_{j+1}\ \forall j < i$ and $\alpha_j \geq \alpha_{j+1}\ \forall j \geq i$. Likewise, f is multimodal or k-modal if it has k local maxima. The unimodal case can be solved using a greedy approach. The main ideas are the following.

Given a real number l, let G_l be the maximum gain with respect to f of an interval of length l. By simplicity, we refer to G_l as the *maximum gain of length l*. Notice that when f is unimodal, any interval of maximum gain of length l contains intervals of maximum gain for lengths lower than l. Hence, given $l_1, l_2 \in \mathbb{R}$ with $l_1 \leq l_2 \leq \omega^*$, it can always be found t_1 and t_2 of lengths $l_{t_1} = l_1$ and $l_{t_2} = l_2$ of maximum gain in f for l_1, l_2, respectively, such that $t_1 \subseteq t_2$.

Theorem 3 *Let $l = \frac{\omega^*}{m}$ and t be a subinterval of $[0, \omega^*]$ s.t. $l_t = l$ and $g_t = G_l$. Then $T^* = \{t, ..., t\}$ with $|T^*| = m$ is optimal for MEC problem when f is unimodal.*

Corollary 4 *The MEC-Problem can be solved in $O(n+m)$ time when f is unimodal.*

When f is k-modal ($k > 1$), it is easy to find an example for which the MEC problem cannot be solved with the same greedy algorithm. Let us introduce some concepts to be used in the proposed solution. Recall that E is defined as the set of edges of f.

Definition 5 *An interval $t = (\theta_1, \theta_2)$ is discrete, called as a d-interval, if $\theta_1 \in E$ and $\theta_2 \in E$. The interval is semi-discrete if it starts or ends in an edge of f.*

Definition 6 *We say that a step of f is modal (m-step) if it is a local maximum.*

Definition 7 *An interval $t = (\theta_1, \theta_2)$ is an md-interval, if it is discrete and contains at least a modal step of f. A semi md-interval is a semi-discrete interval containing at least a modal step of f.*

The following results constitute the heart of our approach.

Lemma 8 *There exists an optimal solution T^* to the MEC problem s.t. for any $t \in T^*$, t is at least semi-discrete.*

Lemma 9 *There exists an optimal solution T^*, to the MEC problem s.t. $\forall i = 1 \ldots |T^*| - 1$, t_i is a discrete interval.*

Theorem 10 *There exist an optimal solution T^* to the MEC problem s.t. $\forall i = 1 \ldots |T^*| - 1$, t_i is an md-interval, and t_m is a semi md-interval.*

4.1 The algorithm

The following property of any optimal solution T^* can be easily proved: *removing any interval t_i from T^* yields a solution $T' = T^* - \{t_i\}$ that is optimal for $m-1$ moves and $\omega^* - l_{t_i}$ total displacement of the SCA*. This property, known as *optimal substructure property*, allows us to find an optimal solution by solving a collection of subproblems and it is the base of

the greedy and dynamic programming paradigms. In addition, and more importantly, according to Theorem 10, the general form of the optimal solution to the MEC problem can be expressed as:

$$I_m^* = D_{m-1}^l + G_{\omega^* - l}, \tag{3}$$

being I_m^* the maximum gain associated to m moves, D_{m-1}^l the maximum gain for length at most l using $m-1$ discrete intervals, and G_{ω^*-l} the maximum gain in f for the remaining length. Because of the optimal substructure of the problem, D_{m-1}^l is optimal for length l. However, we cannot know beforehand the value of l, hence we divide the problem in two tasks:

Task 1: Computing D_{m-1}^l, $\forall l \in (0, \omega^*)$.

Task 2: Computing G_l, $\forall l \in (0, \omega^*)$.

According to (3), a solution with length l for the first task is associated to a solution with length $\omega^* - l'$ in the second, where $l' \leq l$ is the total length of the intervals obtained during the computation of D_{m-1}^l. In addition, notice that $l \in \mathbb{N}$ because the length of the steps of f are integers. Therefore, the following remarks can be stated:

Remark 1 *Combining the solutions from Task 1 and Task 2 takes $O(\omega^*)$.*

Remark 2 *I_m^* is the maximum value obtained after combining the solutions from Task 1 and Task 2.*

Task 2 can be easily solved in linear time for a given l and we have:

Theorem 11 *Task 2 can be solved in $O(n\omega^*)$ time.*

We now focus on solving Task 1. Since the considered intervals are discrete, we design an efficient algorithm based on Dynamic Programming (DP). Our algorithm will solve the MEC problem for any length considering only *md*-intervals, which is the requirement for Task 1. For simplicity, we refer to this version as the MEC-*d* problem.

Let B be the set containing the *md*-intervals of f. In addition, let us consider the table $D[i, j, l]$ indicating the maximum gain for the MEC-*d* problem when using up to interval i of B, with j movements and l as maximum solar displacement. Notice that intervals in B do not need to be sorted, but we assume a fixed order during the execution of the algorithm. Then, the update rule for D can be expressed as:

$$D[i,j,l] = \begin{cases} 0 & \text{(a) } 0 \in \{i, j, l\} \\ D[i-1, j, l] & \text{(b) } l < l_i \\ \max(D[i-1, j, l], & \text{(c) } else \\ \quad g_i + D[i, j-1, l-l_i]) & \end{cases} \tag{4}$$

where g_i represents the gain of the interval i of B.

Theorem 12 *DP is optimal for the MEC-d problem and spends $O(n^2 m \omega^*)$ time.*

Remark 3 *For a given $l \in (0, \omega^*)$, $D[|B|, m-1, l]$ contains the optimal value for Task 1.*

The intervals corresponding to an optimal solution T^* to the MEC problem can be obtained after computing I_m^*. Notice that every decision is associated to an interval, both in Task 1 and Task 2; see Theorems 11 and 12. In Taks 2, the interval associated to G_l (for a given value of l) can be obtained by scanning f. On the other hand, for Task 1, it is easier to use the cases defining equation 4 to retrieve the intervals associated to a decision. Specifically, for any i, j, l, we check (a), (b) or (c); if cases (a) or (b) holds, then the candidate i is not used; if c holds, then we check the equality $D[i, j, l] = D[i-1, j, l]$ and if it holds, then candidate i is not used, otherwise, it is used. Starting this process at $D[|B|, m-1, l]$, being l the length of the optimal solution, it is possible to retrieve the full set of intervals in T^*.

Corollary 13 *The MEC problem can be solved in $O(n^2 m \omega^*)$.*

Acknowledgments

This work is partially supported by grants PID2020-114154RB-I00, TED2021-129182B-I00 and DIN2020-011317 funded by MCIN/AEI/10.13039/501100011033 and the European Union NextGenerationEU/PRTR.

References

[1] Lourdes A Barcia, Rogelio Peón Menéndez, Juan Á Martínez Esteban, Miguel A José Prieto, Juan A Martín Ramos, F Javier de Cos Juez, and Antonio Nevado Reviriego. Dynamic modeling of the solar field in parabolic trough solar power plants. *Energies*, 8(12):13361–13377, 2015.

[2] Ze-Dong Cheng, Ya-Ling He, Bao-Cun Du, Kun Wang, and Qi Liang. Geometric optimization on optical performance of parabolic trough solar collector systems using particle swarm optimization algorithm. *Applied energy*, 148:282–293, 2015.

[3] LM Fernández-Ahumada, FJ Casares, J Ramírez-Faz, and R López-Luque. Mathematical study of the movement of solar tracking systems based on rational models. *Solar Energy*, 150:20–29, 2017.

[4] Hristo Zlatanov and Gerhard Weinrebe. Csp and pv solar tracker optimization tool. *Energy Procedia*, 49:1603–1611, 2014.

Center of maximum-sum matchings of bichromatic points

Pablo Pérez-Lantero*[1] and Carlos Seara†[2]

[1]Departamento de Matemática y Ciencia de la Computación, Universidad de Santiago de Chile, Chile.
[2]Departament de Matemàtiques, Universitat Politècnica de Catalunya, Spain.

Abstract

Let R and B be two disjoint point sets in the plane with $|R| = |B| = n$. Let $\mathcal{M} = \{(r_i, b_i), i = 1, 2, \ldots, n\}$ be a perfect matching that matches points of R with points of B and maximizes $\sum_{i=1}^{n} \|r_i - b_i\|$, the total Euclidean distance of the matched pairs. In this paper, we prove that there exists a point o of the plane (the center of $\mathcal{M}$) such that $\|r_i - o\| + \|b_i - o\| \leq \sqrt{2}\, \|r_i - b_i\|$ for all $i \in \{1, 2, \ldots, n\}$.

1 Introduction

Let R and B be two disjoint point sets in the plane with $|R| = |B| = n$, $n \geq 1$. The points in R are *red*, and those in B are *blue*. A *matching* of $R \cup B$ is a partition of $R \cup B$ into n pairs such that each pair consists of a red and a blue point. A point $p \in R$ and a point $q \in B$ are *matched* if and only if the (unordered) pair (p, q) is in the matching. For every $p, q \in \mathbb{R}^2$, we use pq to denote the segment connecting p and q, and $\|p - q\|$ to denote its length, which is the Euclidean norm of the vector $p - q$. Let $\mathcal{B}(pq)$ denote the disk with diameter equal to $\|p - q\|$, that is centered at the midpoint $\frac{p+q}{2}$ of the segment pq. For any matching $\mathcal{M}$, we use $\mathcal{B}_{\mathcal{M}}$ to denote the set of the disks associated with the matching, that is, $\mathcal{B}_{\mathcal{M}} = \{\mathcal{B}(pq) : (p, q) \in \mathcal{M}\}$.

In this note, we consider the *max-sum* matching $\mathcal{M}$, as the matching that maximizes the total Euclidean distance of the matched points. As our main result, we prove the following theorem:

Theorem 1 *There exists a point o of the plane such that for all $i \in \{1, 2, \ldots, n\}$ we have:*

$$\|r_i - o\| + \|b_i - o\| \;\leq\; \sqrt{2}\, \|r_i - b_i\|.$$

Fingerhut (see Eppstein [3]), motivated by a problem in designing communication networks (see Fingerhut et al. [4]), conjectured that given a set P of $2n$ uncolored points in the plane and a max-sum matching $\{(a_i, b_i), i = 1, \ldots, n\}$ of P, there exists a point o of the plane, not necessarily a point of P, such that

$$\|a_i - o\| + \|b_i - o\| \;\leq\; \frac{2}{\sqrt{3}}\, \|a_i - b_i\| \quad \text{for all } i \in \{1, \ldots, n\}, \tag{1}$$

where $2/\sqrt{3} \approx 1.1547$.

Bereg et al. [2] obtained an approximation to this conjecture. They proved that for any point set P of $2n$ uncolored points in the plane and a max-sum matching $\mathcal{M} = \{(a_i, b_i), i = 1, \ldots, n\}$ of P, all disks in $\mathcal{B}_{\mathcal{M}}$ have a common intersection, implying that any point o in the common intersection satisfies

$$\|a_i - o\| + \|b_i - o\| \;\leq\; \sqrt{2}\, \|a_i - b_i\|,$$

where $\sqrt{2} \approx 1.4142$.

Recently, Barabanshchikova and Polyanskii [1] confirmed the conjecture of Fingerhut.

The statement of Equation (1) is equivalent to stating that the intersection $\mathcal{E}(a_1 b_1) \cap \mathcal{E}(a_2 b_2) \cap \cdots \cap \mathcal{E}(a_n b_n)$ is not empty, where $\mathcal{E}(pq)$ is the region of the plane bounded by the ellipse with foci p and q, and major axis length $(2/\sqrt{3})\, \|p - q\|$ (see [3]).

In our context of bichromatic point sets, given $p \in R$ and $q \in B$, let $\mathcal{E}(pq)$ denote the region bounded by the ellipse with foci p and q, and major axis length $\sqrt{2}\, \|p - q\|$. That is, $\mathcal{E}(pq) = \{x \in \mathbb{R}^2 : \|p - x\| + \|q - x\| \leq \sqrt{2}\, \|p - q\|\}$. Then, the statement of Theorem 1 is equivalent to stating that the intersection $\mathcal{E}(r_1 b_1) \cap \mathcal{E}(r_2 b_2) \cap \cdots \cap \mathcal{E}(r_n b_n)$ is not empty, for any max-sum matching $\{(r_i, b_i), i = 1, 2, \ldots, n\}$ of $R \cup B$.

We note that the factor $\sqrt{2}$ is tight. It suffices to consider two red points and two blue points as vertices of a square, so that each diagonal has vertices of the same color. The center of the square is the only point in common of the two ellipses induced by any max-sum matching.

Hence, to prove Theorem 1 it suffices to consider $n \leq 3$, by Helly's Theorem. Let $X_1, X_2, \ldots, X_n$ be a collection of n convex subsets of $\mathbb{R}^d$, with $n \geq d + 1$. Helly's Theorem [5] asserts that if the intersection of every $d + 1$ of these subsets is nonempty, then the whole collection has a nonempty intersection. That is why we prove our claim only for $n \leq 3$, since we are considering n ellipses in $\mathbb{R}^2$. The arguments that we give in this paper are a simplification and adaptation of the arguments of Barabanshchikova and Polyanskii [1].

*Email: pablo.perez.l@usach.cl
†Email: carlos.seara@upc.edu

Huemer et al. [6] proved that if $\mathcal{M}'$ is any perfect matching of R and B that maximizes the total *squared* Euclidean distance of the matched points, i.e., it maximizes $\sum_{(p,q)\in\mathcal{M}'} \|p-q\|^2$, then all disks of $\mathcal{B}_{\mathcal{M}'}$ have a point in common. With different techniques, the result of Huemer et al. was extended to higher dimensions by Pirahmad et al. [7]. As proved by Bereg et al. [2], the disks of our max-sum matching $\mathcal{M}$ of $R \cup B$ intersect pairwise, a fact that will be used in this paper, but the common intersection is not always possible.

2 Proof of main result

Let R and B be two disjoint point sets defined as above, where $|R| = |B| = n$, $n \leq 3$, and let $\mathcal{M}$ be a max-sum matching of $R \cup B$. Note that for every pair $(p, q) \in \mathcal{M}$ the disk $\mathcal{B}(pq)$ is inscribed in the ellipse $\mathcal{E}(pq)$ (see Figure 1a), which implies $\mathcal{B}(pq) \subset \mathcal{E}(pq)$. Then, for $n = 2$ Theorem 1 is true because the disks of $\mathcal{M}$ intersect pairwise [2, Proposition 2.1]. Trivially, the theorem is also true for $n = 1$. Therefore, we will prove in the rest of the paper that the theorem is also true for $n = 3$, which will require elaborated arguments.

Let $n = 3$, with $R = \{a, b, c\}$ and $B = \{a', b', c'\}$, and let $\mathcal{M} = \{(a, a'), (b, b'), (c, c')\}$ be a max-sum matching of $R \cup B$.

For two points $p, q \in \mathbb{R}^2$, let $r(pq)$ denote the ray with apex p that goes through q, and for a real number $\lambda \geq 1$, let $\mathcal{E}_\lambda(pq)$ be the region bounded by the ellipse with foci p and q and major axis length $\lambda\|p - q\|$. That is, $\mathcal{E}_\lambda(pq) = \{x \in \mathbb{R}^2 : \|p - x\| + \|q - x\| \leq \lambda\|p - q\|\}$. Note that in our context $\mathcal{E}(pq) = \mathcal{E}_{\sqrt{2}}(pq)$, and $\mathcal{E}_\lambda(pq) \subset \mathcal{E}_{\lambda'}(pq)$ for any $\lambda' > \lambda$.

Assume by contradiction that $\mathcal{E}(aa') \cap \mathcal{E}(bb') \cap \mathcal{E}(cc') = \emptyset$. Then, we can "inflate uniformly" $\mathcal{E}(aa')$, $\mathcal{E}(bb')$, and $\mathcal{E}(cc')$ until they have a common intersection. Formally, we can take the minimum $\lambda > \sqrt{2}$ such that $\mathcal{E}_\lambda(aa') \cap \mathcal{E}_\lambda(bb') \cap \mathcal{E}_\lambda(cc')$ is not empty, which means that $\mathcal{E}_\lambda(aa') \cap \mathcal{E}_\lambda(bb') \cap \mathcal{E}_\lambda(cc')$ is singleton. Let o denote the point of $\mathcal{E}_\lambda(aa') \cap \mathcal{E}_\lambda(bb') \cap \mathcal{E}_\lambda(cc')$.

Let $\ell(aa')$ denote the ray with apex o that bisects $r(oa)$ and $r(oa')$. Similarly, we define $\ell(bb')$ and $\ell(cc')$. Let $t(aa')$ denote the line through o tangent to $\mathcal{E}_\lambda(aa')$, oriented so that $\mathcal{E}_\lambda(aa')$ is to its right. Similarly, we define $t(bb')$ and $t(cc')$. It is well known that given an ellipse with foci p and q, and a line tangent at it at some point o, the rays $r(op)$ and $r(oq)$ form equal angles with the tangent line (see Figure 1b). This implies that rays $\ell(aa')$, $\ell(bb')$, and $\ell(cc')$ are perpendicular to the tangent lines $t(aa')$, $t(bb')$, and $t(cc')$, respectively. In other words, they are contained respectively in the normal lines at point o.

Since $\mathcal{E}(aa')$, $\mathcal{E}(bb')$, and $\mathcal{E}(cc')$ intersect pairwise (and also none of them is contained inside other one), we have that o belongs to the boundary of each of $\mathcal{E}_\lambda(aa')$, $\mathcal{E}_\lambda(bb')$, and $\mathcal{E}_\lambda(cc')$. Then, $\mathcal{E}_\lambda(aa')$, $\mathcal{E}_\lambda(bb')$, and $\mathcal{E}_\lambda(cc')$ intersect pairwise, and each pairwise intersection contains interior points. This implies that no two lines of $t(aa')$, $t(bb')$, and $t(cc')$ coincide. Furthermore, the six directions (positive and negative) of $t(aa')$, $t(bb')$, and $t(cc')$ alternate around o, which implies that any two consecutive rays among $\ell(aa')$, $\ell(bb')$, and $\ell(cc')$ counterclockwise around o, have rotation angle strictly less than π (see Figure 1c).

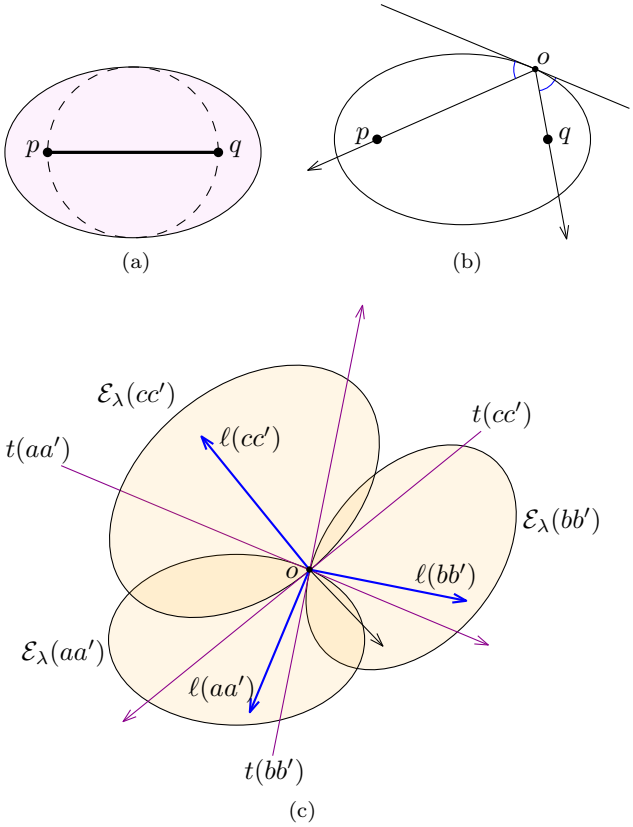


Figure 1: (a) The ellipse $\mathcal{E}(pq)$ and the disk $\mathcal{B}(pq)$. (b) A line tangent to an ellipse forms equal angles with the rays, whose apex is the tangency point, that go through the foci. (c) Point o and the three ellipses.

Let $G = (R \cup B, E)$ be the bipartite graph such that $(p, q) \in E$ if and only if $p \in R$, $q \in B$, and either $(p, q) \in \{(a, a'), (b, b'), (c, c')\}$ or $o \in \mathcal{B}(pq)$. We color the edges into two colors: We say that edge (p, q) is *black* if (p, q) is an edge of the matching, that is, $(p, q) \in \{(a, a'), (b, b'), (c, c')\}$. Otherwise, we say that (p, q) is *white*. Note that this color classification is consistent, since we have that $o \notin \mathcal{B}(pq)$ for all edges $(p, q) \in \{(a, a'), (b, b'), (c, c')\}$ because $\mathcal{B}(pq)$ is contained in the interior of $\mathcal{E}_\lambda(pq)$ and o is in the boundary of $\mathcal{E}_\lambda(pq)$.

The proof of the next lemma is included for completeness.

Lemma 2 ([1]) *If G has a cycle whose edges are color alternating, then $\mathcal{M}$ is not a max-sum matching of*

$R \cup B$.

Proof. For a black edge (p, q) we have that $\|p - o\| + \|q - o\| = \lambda\|p - q\|$. For a white edge (p, q) we have that $\|p - o\| + \|q - o\| < \lambda\|p - q\|$, since $o \in \mathcal{B}(pq)$ and $\mathcal{B}(pq)$ is contained in the interior of $\mathcal{E}_\lambda(pq)$. Let $(r_1, b_1, r_2, b_2, \ldots, r_m, b_m, r_{m+1} = r_1)$ be a cycle of length m, where $r_1, \ldots, r_m \in R$ and $b_1, \ldots, b_m \in B$, and its edges are color alternating. Suppose w.l.o.g. that the edge (r_1, b_1) is black, which means that the edges $(r_1, b_1), \ldots, (r_m, b_m) \in \mathcal{M}$ are all black, and the edges $(b_1, r_2), \ldots, (b_m, r_{m+1}) \in \mathcal{M}$ are all white. Then, we have that:

$$\begin{aligned}\sum_{i=1}^{m} \|r_i - b_i\| &= \frac{1}{\lambda}\sum_{i=1}^{m} (\|r_i - o\| + \|b_i - o\|) \\ &= \frac{1}{\lambda}\sum_{i=1}^{m} (\|b_i - o\| + \|r_{i+1} - o\|) \\ &< \sum_{i=1}^{m} \|b_i - r_{i+1}\|.\end{aligned}$$

Hence, by replacing in $\mathcal{M}$ the black edges of the cycle by the white edges, we will obtain a matching of larger total sum. □

The above alternating cycle idea in the problems about intersections of geometric objects induced by matchings appeared in the proof of Theorem 3 in the paper of Pirahmad et al. [7].

Lemma 3 *Each vertex of G has at least one white edge incident to it.*

Proof. Consider the blue vertex a'. Assume w.l.o.g. that o is the origin of coordinates, and a' is in the positive direction of the y-axis. We have that $\angle aoa' < \pi/2$ because $o \notin \mathcal{B}(aa')$, then assume w.l.o.g. that a is in the interior of the first quadrant Q_1. Let Q_2, Q_3, and Q_4 be the second, third, and fourth quadrants, respectively. Further assume w.l.o.g. that rays $\ell(aa')$, $\ell(bb')$, and $\ell(cc')$ appear in this order counterclockwise.

Assume by contradiction that there is no white edge incident to a'. This implies that b, c belong to the interior of $Q_1 \cup Q_2$. If $c \in Q_2$, then the counterclockwise rotation angle from $\ell(cc')$ to $\ell(aa')$ is larger than π. Hence, $c \in Q_1$. If $b \in Q_1$, then the counterclockwise rotation angle from $\ell(aa')$ to $\ell(bb')$, or that from $\ell(bb')$ to $\ell(cc')$, is larger than π. Hence $b \in Q_2$. Furthermore, if both b' and c' belong to $Q_1 \cup Q_2$, then the counterclockwise rotation angle from $\ell(bb')$ to $\ell(cc')$ is larger than π. Hence, at least one of b', c' belong to the interior of $Q_3 \cup Q_4$. That is, $b' \in Q_3$ and/or $c' \in Q_4$. The proof is divided now into three cases:

Case 1: $b' \in Q_3$ and $c' \in Q_4$. Since $b \in Q_2$ and $c' \in Q_4$, the angle $\angle boc' \geq \pi/2$, which implies that $o \in \mathcal{B}(bc')$ (see Figure 2a). That is, edge (b, c') is white. Similarly, edge (b', c) is also white. The colors of the edges of the cycle (b, c', c, b', b) alternate, then Lemma 2 implies a contradiction.

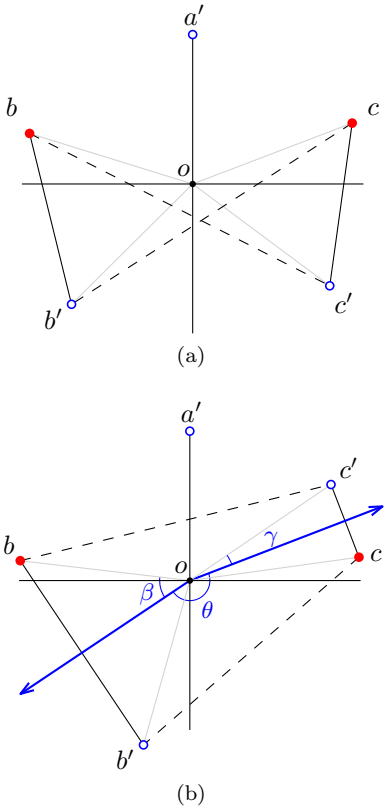


Figure 2: Proof of Lemma 3. Black edges are in normal line style, and white edges in dashed style.

Case 2: $b' \in Q_3$ and $c' \notin Q_4$. Since the counterclockwise rotation angle θ from $\ell(bb')$ to $\ell(cc')$ is smaller than π, we must have that $c' \in Q_1$. As in Case 1, we have that edge (b', c) is white, given that $b' \in Q_3$ and $c \in Q_1$. Let β be the half of the angle $\angle bob'$, and γ be the half of the angle $\angle coc'$ (see Figure 2b). Note that $\angle bob' < \pi/2$ and $\angle coc' < \pi/2$ because $o \notin \mathcal{B}(bb')$ and $o \notin \mathcal{B}(cc')$. We have that $\beta, \gamma < \pi/4$, which implies that $\angle boc' \geq 2\pi - \beta - \gamma - \theta \geq \pi/2$. Hence, edge (b, c') is also white. Again, the colors of the edges of the cycle (b, c', c, b', b) alternate, and Lemma 2 implies a contradiction.

Case 3: $b' \notin Q_3$ and $c' \in Q_4$. The proof of this case is analogous to that of Case 2.

The lemma thus follows. □

Lemma 3 implies that the graph G has always a cycle (of length four or six) whose edges are color alternating. Hence, Lemma 2 implies a contradiction, and we obtain that the max-sum matching $\mathcal{M}$ ensures that $\mathcal{E}(aa') \cap \mathcal{E}(bb') \cap \mathcal{E}(cc') \neq \emptyset$. Therefore, Theorem 1 holds.

Acknowledgements: We thank Alexander Polyanskii and Polina Barabanshchikova for sharing and discuss with

us their manuscript. P. P-L. was partially supported by project DICYT 041933PL Vicerrectoría de Investigación, Desarrollo e Innovación USACH (Chile). C. S. is supported by project PID2019-104129GB-I00/ MCIN/ AEI/ 10.13039/501100011033.

References

[1] P. Barabanshchikova and A. Polyanskii. Intersecting ellipses induced by a max-sum matching. *arXiv preprint arXiv:2212.14200*, 2022.

[2] S. Bereg, O. P. Chacón-Rivera, D. Flores-Peñaloza, C. Huemer, P. Pérez-Lantero, and C. Seara. On maximum-sum matchings of points. *Journal of Global Optimization*, pages 1–18, 2022.

[3] D. Eppstein. Geometry Junkyard. `https://www.ics.uci.edu/~eppstein/junkyard/maxmatch.html`.

[4] J. A. Fingerhut, S. Suri, and J. S. Turner. Designing least-cost nonblocking broadband networks. *J. Algorithms*, 24(2):287–309, 1997.

[5] E. Helly. Über Mengen konvexer Körper mit gemeinschaftlichen Punkte. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 32:175–176, 1923.

[6] C. Huemer, P. Pérez-Lantero, C. Seara, and R. I. Silveira. Matching points with disks with a common intersection. *Discrete Mathematics*, 342(7):1885–1893, 2019.

[7] O. Pirahmad, A. Polyanskii, and A. Vasilevskii. Intersecting diametral balls induced by a geometric graph. *Discrete & Computational Geometry*, pages 1–18, 2022.

A fitting problem in three dimension

Pablo Pérez-Lantero*[1], Carlos Seara†[2], and Jorge Urrutia‡[3]

[1]Departamento de Matemática y Ciencia de la Computación, Universidad de Santiago de Chile
[2]Departament de Matemàtiques, Universitat Politècnica de Catalunya
[3]Instituto de Matemáticas de la Universidad Nacional Autónoma de México

Abstract

Let P be a set of n points in $\mathbb{R}^3$ in general position, and let $RCH(P)$ be the rectilinear convex hull of P. In this paper we use an efficient $O(n \log^2 n)$ time and $O(n \log n)$ space algorithm to compute and maintain the set of vertices of the rectilinear convex hull of P as we rotate $\mathbb{R}^3$ around the Z-axis to obtain an improvement of the time complexity in an optimization algorithm for a fitting problem in $\mathbb{R}^3$.

1 Introduction

Let P be a set of n points in $\mathbb{R}^3$ in general position, and let $RCH(P)$ be the rectilinear convex hull of P. An *open octant* in $\mathbb{R}^3$ is the intersection of the three open halfspaces, whose supporting planes are perpendicular to the X-axis, to the Y-axis, and to the Z-axis, respectively. An octant is called P-free if it contains no elements of P. The rectilinear convex hull of a set of points in $\mathbb{R}^3$ is defined as

$$RCH(P) = \mathbb{R}^3 \setminus \bigcup_{W \in \mathcal{W}(P)} W,$$

where $\mathcal{W}(P)$ is the set of P-free open octants of $\mathbb{R}^3$.

Theorem 1 *[2] The rectilinear convex hull of P, $RCH(P)$, can be computed in optimal $O(n \log n)$ time and $O(n)$ space.*

If we do rotations of the X- and Y-axis around the Z-axis by an angle θ in the clockwise direction, instead of octants we get θ-octants, and the corresponding rectilinear convex hulls generated $RCH_\theta(P)$. Thus, an open θ-octant is the intersection of three open halfspaces whose supporting planes are orthogonal to three mutually orthogonal lines through the origin X_θ, Y_θ, and Z. An open θ-octant is called P-free if it contains no elements of P. The θ-rectilinear convex hull $RCH_\theta(P)$ of a point set P is defined as

$$RCH_\theta(P) = \mathbb{R}^3 \setminus \bigcup_{W \in \mathcal{W}_\theta(P)} W,$$

where $\mathcal{W}_\theta(P)$ denotes the set of all P-free *open θ-octants*. The points of P are labeled $\{p_1, \ldots, p_n\}$ from top to bottom by decreasing z-coordinates.

Theorem 2 *[2] Maintaining the elements of P that belong to the boundary of $RCH_\theta(P)$ as $\theta \in [0, 2\pi]$ can be done in $O(n \log^2 n)$ time and $O(n \log n)$ space. The algorithm stores the set of angular intervals in $[0, \pi]$ at which the points are θ-active.*

2 A 2-fitting problem in 3D

The *oriented 2-fitting problem* [1] is defined as follows: Given a point set P in $\mathbb{R}^3$, find a plane Π, called the *splitting plane* of P (assume that Π is parallel to the XY-plane), and four parallel halfplanes $\pi_1, \pi_2, \pi_3, \pi_4$, called the *supporting halfplanes* of P, such that:

1. Π splits P into two non-empty subsets P_1 and P_2, i.e., $\{P_1, P_2\}$ is the bipartition of P produced by the splitting-plane Π.

2. $\pi_1, \pi_2, \pi_3, \pi_4$ are orthogonal to Π, π_1 and π_2 lie above Π, and π_3 and π_4 lie below Π, each one of $\pi_1, \pi_2, \pi_3, \pi_4$ containing at least a point of P. The point sets P_1 and P_2 are contained between π_1 and π_2, and π_3 and π_4, respectively. See Figure 1.

3. The maximum of ϵ_1 and ϵ_2 is minimized, where ϵ_1 is the error tolerance of P_1 with respect to π_1 and π_2, and ϵ_2 is the error tolerance of P_2 with respect to π_3 and π_4.

4. The solution for the 2-fitting problem is given by the mid halfplane of the supporting halfplanes π_1 and π_2 and the mid halfplane of the supporting halfplanes π_3 and π_4.

*Email: pablo.perez.l@usach.cl. Partially supported by project DICYT 042332PL Vicerrectoría de Investigación, Desarrollo e Innovación USACH (Chile).

†Email: carlos.seara@upc.edu. Partially supported by projects PID2019-104129GB-I00/ MCIN/ AEI/ 10.13039/501100011033.

‡Email: urrutia@matem.unam.mx. Supported by PAPIIT IN102117 Programa de Apoyo a la Investigación e Innovación Tecnológica, UNAM.

The error tolerances ϵ_1 and ϵ_2 are defined by the Euclidean distances between the two parallel supporting-halfplanes on either sides of Π. The problem consists in getting the bipartition $\{P_1, P_2\}$ of P such that $\max\{\epsilon_1, \epsilon_2\}$ is minimum. It is a min-max problem. See Figure 1. If the orientation of the splitting plane

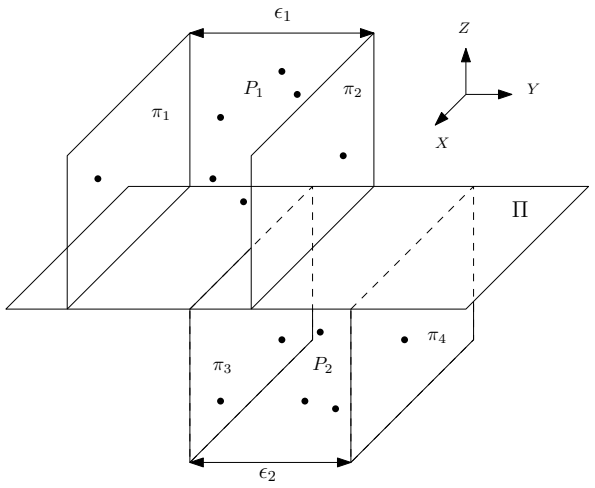


Figure 1: The splitting-plan plane Π and the two pairs of parallel supporting-halfplanes.

is fixed, the problem can be solved in $O(n^2)$ time and $O(n)$ space, as proved by Díaz-Báñez et al. [1]. We will design an algorithm that bounds the solution for this case: when the orientation of the splitting plane is fixed. The complexities are smaller than those in the algorithm Díaz-Báñez et al. [1]. In our algorithm, instead of doing a sequence of bipartitions of P, we will maintaining $RCH_\theta(P)$ as we rotate the space around the Z-axis, and compute optimal solutions in each of the linear number of events at which the $RCH_\theta(P)$ changes for $\theta \in [0, \pi]$.

We assume that the splitting-plane Π is parallel to the XY plane, all the points of P are above the XY plane and sorted with respect to the z-coordinate, where p_1 is the point with the largest z-coordinate, and p_n is the point with the smallest z-coordinate. We also assume that the Z-axis passes through p_1, and thus, its coordinates do not change as we rotate $\mathbb{R}^3$ around the Z-axis, and p_1 and p_n are always in the boundary of $RCH_\theta(P)$.

Suppose that the supporting halfplanes have normal unit vectors. Thus, the 2-fitting problem reduces to computing four parallel supporting halfplanes $\pi_1, \pi_2, \pi_3, \pi_4$ of a bipartition $\{P_1, P_2\}$ of P, and therefore, to computing four points in the boundary of $RCH_\theta(P)$, for some $\theta \in [0, \pi]$. See Figure 2.

We will discretize the problem by considering the angular sub-intervals of $[0, \pi]$ such that in each subinterval the points in the boundary of $RCH_\theta(P)$ do not change. By Theorem 2, their number is linear in n. Then, we will show how to optimize the error tolerance at the endpoints of each of these sub-intervals.

Recall that a point $p \in P$ is said to be θ-active if at least one of the p^θ-octants is P-free. The definition of a θ-active point considering an octant can be easily adapted to considering a *dihedral* (two perpendicular and axis-parallel planes) as follows:

Definition 3 *Let $p \in P$ and $\{s, t\} \in \{\{1,2\},\{3,4\},\{5,6\},\{7,8\}\}$. We say that p is a $\theta_{\{s,t\}}$-active point if p is θ-active for both the s-th and t-th octants.*

For example, a point $p \in P$ is $\theta_{\{1,2\}}$-active if p is θ-active for both the first and second octants. In fact, the union of the first and second p^θ-octants is a dihedral, which is P-free and its edge goes through p.

Lemma 4 *The boundary of the projection of $RCH_\theta(P)$ on the ZY_θ plane is formed by the points of P which are $\theta_{\{1,2\}}$-active, $\theta_{\{3,4\}}$-active, $\theta_{\{5,6\}}$-active, and $\theta_{\{7,8\}}$-active in the direction defined by the ZY_θ plane in the unit circle S^1. Thus, the four staircases of the boundary of the projection of $RCH_\theta(P)$ on the ZY_θ plane are as follows: the first staircase is formed by the $\theta_{\{1,2\}}$-active points, the second staircase is formed by the $\theta_{\{3,4\}}$-active points, the third staircase is formed by the $\theta_{\{5,6\}}$-active points, and the fourth staircase is formed by the $\theta_{\{7,8\}}$-active points.*

Proof. The proof follows by observing that any point p in the interior of the projection of $RCH_\theta(P)$ on the ZY_θ plane is dominated by at least one point for each of the four quadrants, and it is so because p is not $\theta_{\{s,t\}}$-active for any $\{s, t\} \in \{\{1,2\},\{3,4\},\{5,6\},\{7,8\}\}$, see Figure 2. Furthermore, the rightmost point in the projection on the ZY_θ is both $\theta_{\{1,2\}}$-active and $\theta_{\{7,8\}}$-active, and the leftmost point is both $\theta_{\{3,4\}}$-active and $\theta_{\{5,6\}}$-active. □

Let p^θ_{left} and p^θ_{right} denote, respectively, the leftmost and rightmost points of the projection of P on the plane ZY_θ. For any angle θ, let L_θ be the list consisting of the $\theta_{\{1,2\}}$-active points and the $\theta_{\{5,6\}}$-active points of P, sorted in decreasing order of their z-coordinate. By simplicity, we assume with loss of generality that no two elements of P have the same z-coordinate. Let $m = O(n)$ denote the number of elements of L_θ and let $z_1, z_2, \ldots, z_m$ denote the sorted elements of L_θ. The $\theta_{\{1,2\}}$-active points of L_θ, those of the first staircase of $RCH_\theta(P)$ are colored red, and the $\theta_{\{5,6\}}$-active points, those of the third staircase of $RCH_\theta(P)$, are colored blue (see the red and blue staircases of Figure 2, containing as vertices the red and blue elements of L_θ, respectively). We can represent L_θ as a standard binary search tree, with extra $O(1)$-size data at each node, such that inserting/deleting an element can be done in $O(\log n)$ time, and also the following queries can all be answered in $O(\log n)$ time:

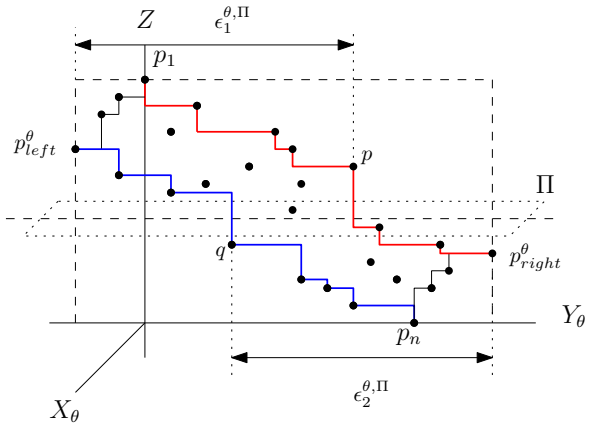


Figure 2: Projection of the $RCH_\theta(P)$ on the ZY_θ plane. The bipartition plane Π is determined by the $\theta_{\{1,2\}}$-active point p and the $\theta_{\{5,6\}}$-active point q, which determine the error tolerance functions $\epsilon_1^{\theta,\Pi}$ and $\epsilon_2^{\theta,\Pi}$.

(1) Given an element in the list, retrieve its position.

(2) Given a position $j \in \{1, 2, \ldots, m\}$, retrieve the rightmost *red* element in the sublist $z_1, z_2, \ldots, z_j$.

(3) Given a position $j \in \{1, 2, \ldots, m\}$, retrieve the leftmost *blue* element in the sublist $z_j, z_{j+1}, \ldots, z_m$.

By simplicity in the explanation, we will assume that p^θ_{left} is always above p^θ_{right} in the z-coordinate order. Hence, any bipartition plane Π must have p^θ_{left} above it and p^θ_{right} below it. Furthermore, Π is determined by the closest $\theta_{\{1,2\}}$-active point p above Π and the closest $\theta_{\{5,6\}}$-active point q below Π (see Figure 2). This is why the list L_θ is for the $\theta_{\{1,2\}}$-active and $\theta_{\{5,6\}}$-active points. If the assumption is not considered, then our arguments must include a similar list with the $\theta_{\{3,4\}}$-active and $\theta_{\{7,8\}}$-active points for the situations in which p^θ_{left} is below p^θ_{right}.

The next facts are the keys for the algorithm:

1. Since each point of P can change its condition of being $\theta_{\{s,t\}}$-active a constant number of times, then the total number of times there is a change in some of the four staircases, hence in L_θ, is $O(n)$. Thus, we have $O(n)$ intervals of $[0, \pi]$ with no change in the staircases. We can then define the sequence Θ of the $N = O(n)$ angles $0 = \theta_0 < \theta_1 < \theta_2 < \cdots < \theta_N = \pi$, such that for each interval $[\theta_i, \theta_{i+1})$, $i = 0, 1, \ldots, N-1$ the list L_θ do not change.

2. For an angle $\theta \in [0, \pi]$ and a point p of P, let p^θ be the projection of p on ZY_θ, and let α_p be the angle formed by the X-axis and the line through the origin O and the projection of p on the XY-plane. For any point q, let $d(q, Z)$ denote the distance from q to the Z-axis. We have that $d(p^\theta, Z) = d(p, Z) \cdot \cos(\alpha_p - \theta)$, which is a function depending only on θ since $d(p, Z)$ and α_p are constants.

3. For a fixed angle $\theta \in [0, \pi]$, a bipartition of P by a plane Π induces a partition of the list $L_\theta = z_1, z_2, \ldots, z_m$ into two sublists: $z_1, z_2, \ldots, z_k$ with the elements above Π, and $z_{k+1}, z_{k+2}, \ldots, z_m$ with the elements below Π. And vice versa, every such a partition of L_θ into two lists induces a plane Π that bipartitions P. Let the $\theta_{\{1,2\}}$-active point p and the $\theta_{\{5,6\}}$-active point q be the *witnesses* of this bipartition. That is, p is the rightmost red element in $z_1, z_2, \ldots, z_k$, and q is the leftmost blue element in $z_{k+1}, z_{k+2}, \ldots, z_m$ (see Figure 2). The error tolerances for this bipartition, denoted $\epsilon_1^{\theta,\Pi}$ and $\epsilon_2^{\theta,\Pi}$, are given by the distances

$$\epsilon_1^{\theta,\Pi} = d(p^\theta_{left}, Z) + d(p^\theta, Z) \text{ and}$$

$$\epsilon_2^{\theta,\Pi} = d(p^\theta_{right}, Z) \pm d(q^\theta, Z),$$

where the $+$ or $-$ depends on whether q^θ is to the left or right of the Z-axis in the ZY_θ plane. Note that when moving Π upwards, the functions $\epsilon_1^{\theta,\Pi}$ and $\epsilon_2^{\theta,\Pi}$ are non-increasing and non-decreasing, respectively. Hence, to find an optimal Π for a given angle θ, we can perform a binary search in the range $\{k_1, k_1 + 1, \ldots, k_2 - 1\} \subset \{1, 2, \ldots, m-1\}$ to find an optimal partition $z_1, z_2, \ldots, z_k$ and $z_{k+1}, \ldots, z_m$ of L_θ, where k_1 and k_2 are the positions of p^θ_{left} and p^θ_{right} in L_θ, respectively.

The binary search does the following steps for a given value $k \in \{k_1, k_1 + 1, \ldots, k_2 - 1\}$: Consider a bipartition plane Π induced by the partition $z_1, z_2, \ldots, z_k$ and $z_{k+1}, \ldots, z_m$ of L_θ, and find the witnesses points p and q, each in $O(\log n)$ time by using the queries of the tree supporting L_θ. Then, compute $\epsilon_1^{\theta,\Pi}$ and $\epsilon_2^{\theta,\Pi}$ in constant time. If $\epsilon_1^{\theta,\Pi} = \epsilon_2^{\theta,\Pi}$, then stop the search. Otherwise, if $\epsilon_1^{\theta,\Pi} < \epsilon_2^{\theta,\Pi}$ (resp. $\epsilon_1^{\theta,\Pi} > \epsilon_2^{\theta,\Pi}$), then we increase (resp. decrease) the value of k accordingly with the binary search and repeat. We return the value of k visited by the search that minimizes $\max\{\epsilon_1^{\theta,\Pi}, \epsilon_2^{\theta,\Pi}\}$. This search makes $O(\log n)$ steps, each in $O(\log n)$ time, thus it costs $O(\log^2 n)$ time.

4. Let θ_i and θ_{i+1} be two consecutive angles of the sequence Θ. It may happen for some angle $\theta \in (\theta_i, \theta_{i+1})$, and some bipartitioning plane Π, that

$$\epsilon_1^{\theta,\Pi} = \epsilon_2^{\theta,\Pi} <$$

$$< \max\left\{\epsilon_1^{\theta_i,\Pi}, \epsilon_2^{\theta_i,\Pi}\right\}, \max\left\{\epsilon_1^{\theta_{i+1},\Pi}, \epsilon_2^{\theta_{i+1},\Pi}\right\}.$$

That is, the objective function improves inside the interval $[\theta_i, \theta_{i+1})$ for the angle θ. In fact, this can be happen for a linear number of angles. For example, suppose that p^θ_{left} and p^θ_{right} are sufficiently far from the Z-axis, and the rest of the elements of L_θ are suffiently close to the Z-axis. Further suppose that the function $d(p^\theta_{left}, Z)$ is increasing, and function $d(p^\theta_{right}, Z)$ is decreasing in (θ_i, θ_{i+1}), and that they coincide for some some $\theta \in (\theta_i, \theta_{i+1})$. For any bipartition plane Π, we will have that the tolerance functions $\epsilon_1^{\theta,\Pi} \approx d(p^\theta_{left}, Z)$ and $\epsilon_2^{\theta,\Pi} \approx d(p^\theta_{right}, Z)$ are increasing and decreasing, respectively, and they will also coincide for some angle $\theta \in (\theta_i, \theta_{i+1})$.

Considering all the facts above, we next describe an approximation algorithm running in subquadratic time for solving the 2-fitting problem in 3D, in the case that the orientation of the splitting-plane is fixed. The approximation consists in computing the best bipartition plane for a discrete set of critical angles. That is, we find such a plane for the $O(n)$ angles of the sequence Θ. Our algorithm leaves apart the fact number 4 above, which would imply to consider a quadratic number of critical angles.

2-FITTING ALGORITHM IN 3D. FIXED ORIENTATION OF THE SPLITTING-PLANE

1. By Theorems 1 and 2, and Lemma 4, we compute in $O(n \log^2 n)$ time and $O(n \log n)$ space, for all points $p \in P$ the angular intervals $I(p)$ in which p is $\theta_{\{s,t\}}$-active for some $\{s,t\} \in \{\{1,2\},\{3,4\},\{5,6\},\{7,8\}\}$. We have $O(1)$ intervals for each p, each one associated with the corresponding $\{s,t\}$. For each p, we intersect pairwise the intervals of $I(p)$ to find the set $I'(p)$ of $O(1)$ intervals such that for each interval we have: p is only $\theta_{\{1,2\}}$-active; p is both $\theta_{\{1,2\}}$-active and $\theta_{\{7,8\}}$-active (i.e., p is p^θ_{right}); p is only $\theta_{\{5,6\}}$-active; or p is both $\theta_{\{3,4\}}$-active and $\theta_{\{5,6\}}$-active (i.e., p is p^θ_{left}).

2. We sort in $O(n \log n)$ time the endpoints of $I'(p)$ for all $p \in P$ to obtain the sequence Θ of the $O(n)$ angles $0 = \theta_0 < \theta_1 < \theta_2 < \cdots < \theta_N = \pi$, such that the list L_θ do not change for all $\theta \in [\theta_i, \theta_{i+1})$, $i = 0, 1, \ldots, N-1$. Thinking on sweeping the sequence Θ with the angle θ from left to right, we associate with each θ_i the point p_i of P and the interval of $I'(p_i)$ with endpoint θ_i. Then, for each θ_i we know which point of P changes some $\theta_{\{s,t\}}$-active condition, and the precise conditions it changes.

3. We sweep Θ from left to right: As a initial step, for $\theta = 0$, we compute the projection of $RCH_0(P)$ on the plane ZY_0, the points p^0_{left} and p^0_{right} in the projection, and build the list L_0 (as a tree) with the $\theta_{\{1,2\}}$-active and $\theta_{\{5,6\}}$-active points in $O(n \log n)$ time.

   In the next steps, for $i = 1, 2, \ldots, N$, we have $\theta = \theta_i$ and we update p^θ_{left} and p^θ_{right} in constant time from $p^{\theta_{i-1}}_{left}$, $p^{\theta_{i-1}}_{right}$, and the point p_i associated with θ_i, and update L_θ by inserting/deleting p_i in $O(\log n)$ time. The color of p_i (red or blue) is known according to the $\theta_{\{s,t\}}$-active condition that p_i changes.

   In each step, the initial one and the subsequent ones, we perform the binary search in L_θ in $O(\log^2 n)$ time to find the bipartition plane Π that minimizes $\epsilon_\theta = \max\{\epsilon_1^{\theta,\Pi}, \epsilon_2^{\theta,\Pi}\}$. At the end, we return the angle θ of Θ (joint with its corresponding optimal plane Π) such that ϵ_θ is the smallest over all angles of Θ.

It is clear that the running time of the above algorithm is $O(n \log^2 n)$. We note that the quality of the solution can be improved in terms of ε-approximations. Indeed, for $\varepsilon > 0$, if we split the interval $[0, \pi]$ into sub-intervals of length $\delta = \varepsilon / D$, where D is an upper bound of the absolute value of the first derivative of the functions $\epsilon_1^{\theta,\Pi}$ and $\epsilon_2^{\theta,\Pi}$ for all θ, and apply the binary search also for θ being the endpoints of these sub-intervals, then the solution $APROX$ given by the algorithm is such that $APROX - OPT \leq \delta D$, where OPT denotes the optimal solution. This implies that $OPT \leq APROX \leq OPT + \varepsilon$. The running time will be $O(n \log^2 n + (\pi/\delta) \log^2 n) = O(n \log^2 n + (D\pi/\varepsilon) \log^2 n)$. A value for D can be twice the maximum distance of a point of P to the Z-axis, and can be considered a constant by scaling the point set P. Hence, the final running time is $O(n \log^2 n + \varepsilon^{-1} \log^2 n)$.

Therefore, we arrive to the following theorem:

Theorem 5 *For any $\varepsilon > 0$, an upper bound of the optimal solution of the oriented 2-fitting problem in 3D, with absolute error at most ε, can be obtained in $O(n \log^2 n + \varepsilon^{-1} \log^2 n)$ time and $O(n \log n)$ space if the orientation of the splitting plane is fixed.*

References

[1] J. M. Díaz-Bañez, M. A. López, M. Mora, C. Seara, and I. Ventura. Fitting a two–joint orthogonal chain to a point set. *Computational Geometry: Theory and Applications*, 44(3), (2011), pp. 135–147.

[2] P. Pérez-Lantero, C. Seara and J. Urrutia. Rectilinear convex hull of points in 3D. *14th Latin American Theoretical Informatics Symposium*, São Paulo, Brazil, January 5-8, (2021), LNCS 12118, pp. 296—307, doi.org/10.1007/978-3-030-61792-9-24

On strip separability of bichromatic point sets

Nicolau Oliver*[1] and Carlos Seara†[1]

[1]Universidad Politécnica de Catalunya

Abstract

In this paper we extend and improve algorithms for the separability of red and blue points in the plane using four parallel lines. We also prove sufficient conditions to meet this separability criteria.

1 Introduction

Separability problems of a bichromatic point set in $\mathbb{R}^2$ by a set of parallel lines were studied in Arkin et al. [1, 2], Hurtado et al. [4], and Seara [9]. The number of these lines is denoted by k. It is well known that two object sets are line separable ($k = 1$) if and only if their convex hulls do not intersect. The decision problem of linear separability for two disjoint sets of points, segments, polygons, or circles can be solved in linear time, see Megiddo [6], and O'Rourke et al. [8].

Let B be a set of blue points and R a disjoint set of red points $B \cap R = \emptyset$, both in the plane, $|B| = |R| = n$, and let $CH(B)$ and $CH(R)$ be their convex hulls. The minimum number k of parallel lines separating R and B into monochromatic strips can be computed in $O(n^2 \log n)$ time and $O(n^2)$ space. If $k \leq 4$, there exists an algorithm that solves the problem in $O(n \log n)$ time and $O(n)$ space under a constraint on the point sets. We ask for determining the minimum k for separating R and B, or whether there exists some direction such that R and B can be separated with k lines.

Open problem 1 *Can it be decided if B and R are separable by four parallel lines in $O(n \log n)$ time and $O(n)$ space if $CH(B)$ does not contain any red point?*

The algorithm in [9] has a constraint over the points. We try to solve the problem without the constraint. If a general algorithm for $k = 4$ is found, maybe we can generalize it for any k. Another open problem in [9]:

Open problem 2 *Can it be decided in $O(kn \log n)$ time if B and R are separable with at most $k \geq 5$ parallel lines?*

*Email: nicolau.oliver@estudiantat.upc.edu.

†Email: carlos.seara@upc.edu. Supported by PID2019-104129GB-I00/ MCIN/ AEI/ 10.13039/501100011033.

Let "/" be denote one of the separating parallel lines. We refer to *red/blue/red/blue...* as the notation for the subsets $R_1/B_1/R_2/B_2/\ldots$, and the separators as $s_1, s_2, s_3, s_4, \ldots$. Assuming k is minimal for any direction, only two possible orderings of the subsets are possible: *red/blue/red/blue/...*, or *blue/red/blue/red/...*. From now on consider only *red/blue/red/blue/...* separability. A caliper rotates clockwise from 0 to π around a convex polygon, and the separators are denoted as $s_i - s_j$. Calipers are always parallel and rotate in sync.

Observation 3 *All points outside the polygon enter and leave the rotating caliper once, and all points inside the polygon are always inside the caliper.*

We say that a point p inside a caliper is "alive" and is otherwise "dead". The supporting lines of p with respect to the polygon give an interval of directions in which p is alive; one such line corresponds to the entrance slope of p (its "birth") and the other to the departure slope of p (its "death"). The clockwise slope of the supporting lines with respect to the x-axis are computed to be inside the $[0, \pi]$ interval.

2 Separability using four lines

For the $k = 2, 3$ strip separability, Arkin et al. [2] shows $O(n \log n)$ time optimal algorithms. From now on we only consider relevant directions that need at least $k = 4$ lines for separability. The algorithm starts by constructing a caliper that rotates around $CH(B)$. This caliper will constitute the separators s_1 and s_4, see Figure 1. Then, it computes the support lines of the red points with respect to $CH(B)$ and builds the sorted list of birth/death events in $O(n \log n)$ time. The slopes of the support lines belong to $[0, \pi]$. With this event list the caliper can be rotated over $CH(B)$, yielding the sets R_1,R_2,R_3 for all directions. See Figure 1. Next, the algorithm separates the blue points into B_1 and B_2. Take a red point g inside $CH(B)$ as a guard.

Observation 4 *(i) The caliper around $CH(B)$ is never empty of red points. (ii) Red points inside $CH(B)$, including g, belong to R_2. (iii) A line through g separates B_1 from B_2. See Figure 1.*

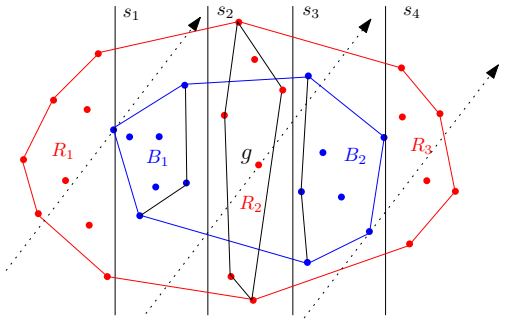


Figure 1: Separability by 4 lines. A red point g inside $CH(B)$ is a *guard* that separates B_1 and B_2 for any direction. A jump event classifying B_1 and B_2.

Jump events describe how the blue points belong to B_1 or B_2 as the caliper rotates. The list of jump events follows from Observation 4, from sorting the angles of the lines through g and each blue point in $O(n \log n)$ time. This list with $O(n)$ angles is the bi-partition list of the guard g. By merging the jump events with the birth and death events we can keep track of all the subsets $R_1/B_1/R_2/B_2/R_3$ as we rotate the caliper. See Figure 1.

In the interval between consecutive events $[e_i, e_{i+1}]$, the subsets $R_1/B_1/R_2/B_2/R_3$ don't change. Neither do their dynamic convex hulls, abbreviated DCH. This interval is separable if $DCH(B_1)$ with $DCH(R_2)$ and $DCH(R_2)$ with $DCH(B_2)$ are separable.

Thus, we compute the supporting lines between the adjacent pairs of dynamic convex hulls (B_1/R_2, R_2/B_2). Intersecting these intervals Θ_1, Θ_2, Θ_3, and Θ_4 with $[e_i, e_{i+1}]$. Repeat for each consecutive pair of events, and merge the results by calculating the union of intervals.

Computing these dynamic convex hulls takes $O(n \log n)$ time, and updating them takes $O(\log n)$ time according to Brodal and Jacob [3]. So the algorithm has $O(n \log n)$ time complexity.

2.1 New algorithm for four line separability

Assume that there are no red points (guards) inside $CH(B)$. Let m guards $G =< g_1, \ldots, g_m >$ be a sequence of guards sorted by birth angle. As above for g, for all $g_i \in G$ compute the sorted bi-partition list in total $O(mn \log n)$ time. This guarantees that we can use the bi-partition lists for the entire rotation.

To separate the blue points into B_1 and B_2, we need at least one guard g_i at any direction inside the caliper, and we use the current guard bi-partition list to do it. When a guard dies, another takes its place.

Observation 5 *There always exists the set $G \subseteq R$, such that for all relevant directions there is at least one guard $g_i \in G$ inside the caliper around $CH(B)$.*

2.2 Minimizing the guard set

The guards have a birth and death event associated, forming the living angle interval. Before an alive guard dies, the next must already be alive. The living angle interval is referred to as the angle that is "guarded/covered" by that guard, see Figure 2; and those intervals must overlap totally covering $[0, \pi]$. Thus, the problem of minimizing guards is equivalent to that of minimizing the sets to cover $[0, \pi]$.

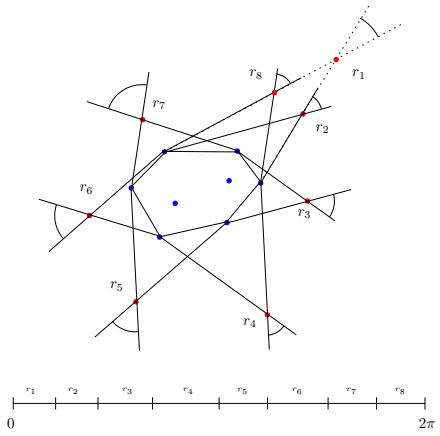


Figure 2: A set of 8 red points covering $[0, 2\pi]$.

3 Sufficient conditions for four line separability

As m could be linear in n, this begs the question of finding conditions over R and B that guarantee that m is constant. There exists a family of configurations with a constant number of guards.

Condition 1 *If there exist guards $a, b, c \in R$ such that all the sides of the triangle $\widehat{abc}$ cross $CH(B)$, then $G = \{a, b, c\}$ is a guard set that covers the entire rotation of the caliper.*

Thus, all configurations that satisfy the Condition 1 have an optimal set of guards of constant size. The guard set $G = \{a, b, c\}$ had a close relationship with the triangle it formed. So, we extend this geometrical analysis to other guard sets: Trace the polygonal line given by the sequence, closing it by adding an edge from the last to the first guard.

Condition 2 *$< g_1, \ldots, g_m >$ is a guard sequence and all the edges of the closed polygonal line traced by the sequence of guards cross $CH(B)$, and m is odd.*

Lemma 6 *If $G =< g_1, \ldots, g_m >$ satisfies Condition 2, then G covers the entire rotation of the caliper.*

This family of closed polygonal lines can be considered for a constant value of m, as $m = 3$ yields the triangle.

The polygons that result from Condition 2 can be from convex to self intersecting. It includes the family of star polygons, with a special interest in those of the form $\{m/\lfloor \frac{m}{2} \rfloor\}$. For an example, see the pentagram $\{5/2\}$ for $m = 5$ and the heptagram $\{7/3\}$ in Figure 3.

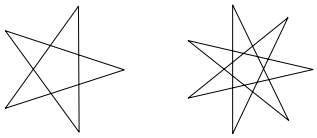


Figure 3: A pentagram $\{5/2\}$ and a heptagram $\{7/3\}$.

Corollary 7 *If $G =< g_1, \ldots, g_m >$ traces an odd star polygon, whose segments all intersect $CH(B)$, then G covers the entire rotation of the caliper.*

The star polygons show that the new algorithm does indeed allow for solving much more general configurations. Nevertheless, Lemma 6 is a sufficient but not necessary condition for an input R and B to have a constant size set of guards. See Figure 4.

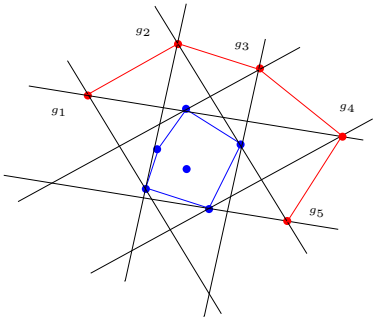


Figure 4: Guards covering the entire rotation, but not tracing a star polygon. Equivalent guards $g_1 \equiv g_5$.

If two guards have the same living angle, they would receive identical use by the algorithm. Thus, two guards are said to be equivalent if the interval they cover is identical, i.e. the support lines of both guards are pairwise parallel. For each red point exterior to $CH(B)$ there is precisely only one other point that satisfies this equivalence: find the support lines of the original point, trace the two parallel support lines tangent to $CH(B)$ on the antipodal points, its intersection point is the equivalent guard. See Figure 4. This equivalence relation reflects that it is possible to swap a guard with its equivalent one, without it affecting the execution of the algorithm.

Observation 8 *If $G =< g_1, \ldots, g_m >$ covers the entire rotation with m odd, there exists equivalent guards $G' =< g'_1, \ldots, g'_m >$ that trace a star polygon.*

4 Finding the optimal guards

As pointed out before, the new algorithm complexity depends on the set G of guards. So it is central to find a constant size G. The guards can be understood as the interval of the caliper rotation they cover, it is the only relevant attribute for the algorithm. This suggests representing the guards as intervals in the unit circle, and each guard being their covering interval. See Figure 5.

Such representation leads to a minimization interval cover problems in $\mathbb{R}$: to find, amongst a set of intervals, the minimum amount such that the union covers an interval $[a, b]$. This is a well-known problem [7] that can be solved via a greedy algorithm in $O(n \log n)$ time with respect to the number of intervals in the set. This indeed can be adapted to find a minimal G. The intersection graph of the intervals has as

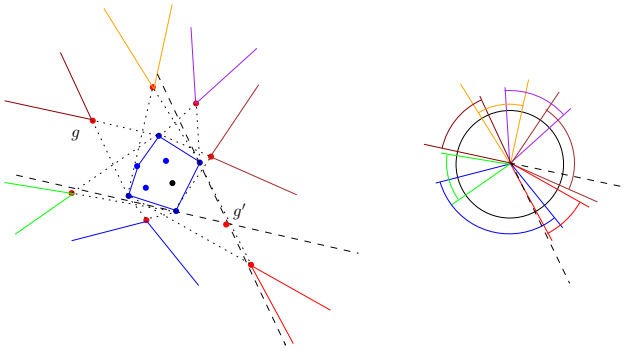


Figure 5: Red guards and their angular intervals.

vertices the intervals, and two vertices are connected if the intersection of the intervals is not empty [5]. See Figure 6. This graph represents pairs of guards that are both alive in some direction. The rotation of the caliper is clockwise, so the edges are arcs and satisfy that the origin guard dies before the destination guard. The direction of the arc captures how the sequence $< g_1, g_2, \ldots, g_m >$ jumps from one guard to the next.

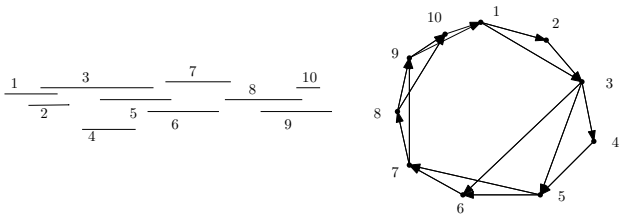


Figure 6: Intersection graph of angular intervals.

Observation 9 *g_i dominates g_j if all directions covered by g_j are also covered by g_i.*

Only the red points that are not dominated by other

red points are considered as candidates. This pruning can be done by sorting the guards in $O(n \log n)$ time.

4.1 An algorithm to find optimal guards

Once dominated red points have been pruned, we find the smallest size set of guards. Let G^{opt} be an optimal guard set, and $< g_1, \ldots, g_m >^{opt}$ its sequence. A local optimality can be found among consecutive guards of the sequence. Given $g_i^{opt} \in < g_1, \ldots, g_m >^{opt}$ there might be several guards that cover the angle of death of g_i^{opt}. Let $candidates(g_i)$ be the set of guards that could "succeed" g_i. It follows that $g_{i+1}^{opt} \in candidates(g_i)$. These candidates can be understood more easily as the adjacent vertices of g_i in the intersection graph.

From the set $candidates(g_i)$, a greedy heuristic chooses the guard that dies the last. The candidate selected by this heuristic will be alive at least for all directions covered by any guard in $candidates(g_i^{opt})$, clockwise from g_i^{opt} death. Let g_{i+1}^{greedy} be this greedy choice of candidate.

Observation 10 *g_{i+1}^{greedy} is as optimal as g_{i+1}^{opt}.*

This is the same greedy heuristic used to solve the classical interval cover problem. Given this heuristic, instead of representing all the outgoing edges for each g_i in the graph, just draw the ones chosen by the greedy heuristic. Omitting degenerate cases, all vertices now have at most one outgoing degree. If we assume that R and B are separable using $k = 4$ lines, the whole rotation must be covered, so $\forall g_i \ |candidates(g_i)| \geq 1$. This results in $deg_{out}(g_i) = 1$. This graph will be referred to as the *greedy intersection graph*, $GIG = (V, E)$, where the vertices $V = R_{pruned}$ exclude dominated guards, and it is computed in $O(n \log n)$ time from the intervals.

Observation 11 *If R and B are separable by $k = 4$ lines, the vertices of GIG have $deg_{out}(g_i) = 1$ and the number of edges is $\sum_i^n deg_{out}(g_i) = n$.*

First, GIG can't be acyclic, meaning that it is not possible to cover the entire rotation. So, GIG has at least a directed cycle of length 2, and the cycle corresponds to a $< g_1, \ldots, g_m >^{opt}$. In fact, each connected component of the GIG must have one and only one cycle. Any path starting from any vertex eventually ends up in one of those cycles.

Observation 12 *GIG cycles have the same length.*

A way of finding $< g_1, \ldots, g_m >^{opt}$ can be described in terms of executing search algorithms over GIG, see Tarjan [10]: for any vertex in GIG, follow the arcs until detecting a cycle, which is an $O(n \log n)$ time algorithm. Finally, we have an algorithm that finds the smallest set G of guards, even in the worst case that $m = n$.

Thus, in $O(n \log n)$ time we can detect if the input can be solved by the new algorithm in optimal $O(n \log n)$ time.

5 The second open question

The main insight is that the new $k = 4$ algorithm works because it imposes a similar structure to the $k = 2$ algorithm. The new $k = 4$ algorithm rotates an extra caliper around the $CH(R_{\text{rec}})$, where R_{rec} (Red recursive) are the points inside $CH(B)$. If a few blue points lay inside $CH(R_{\text{rec}})$, call them B_{rec}, then the recursive algorithm rotates a third caliper around $CH(B_{\text{rec}})$. While the substructure repeats, the recursive algorithm can nest further calipers. These are used in a very similar manner to how they are used in the new $k = 4$ algorithm.

For each of these recursively defined convex hulls, compute the same events as for the new $k = 4$ algorithm. Each nested hull bi-partitions the hull that contains it. Each caliper determines a birth and death for each point. The cost of all these operations amounts to repeating the computations k times, once per each nested caliper. The events generated are thus $O(kn)$, and the total cost is $O(kn \log n)$.

References

[1] E. M. Arkin, F. Hurtado, J. S. Mitchell, C. Seara, and S. S. Skiena. Some separability problems in the plane. *16th EuroCG, Eilat, Israel, March 13-15*, 2000.

[2] E. M. Arkin, F. Hurtado, J. S. Mitchell, C. Seara, and S. S. Skiena. Some lower bounds on geometric separability problems. *IJCGA*, 16(01):1–26, 2006.

[3] G. Brodal and R. Jacob. Dynamic planar convex hull. In *43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 617–626, 2002.

[4] F. Hurtado, M. Noy, P. A. Ramos, and C. Seara. Separating objects in the plane by wedges and strips. *Discrete Applied Mathematics*, 109(1):109–138, 2001.

[5] J. Kratochvil and J. Matousek. Intersection graphs of segments. *Journal of Combinatorial Theory, Series B*, 62(2):289–315, 1994.

[6] N. Megiddo. Linear-time algorithms for linear programming in $\mathbb{R}^3$ and related problems. *SIAM J. on Comput.*, 12(4):759–776, 1983.

[7] V. Mäkinen, V. Staneva, A. Tomescu, D. Valenzuela, and S. Wilzbach. Interval scheduling maximizing minimum coverage. *Discrete Applied Mathematics*, 225:130–135, 2017.

[8] J. O'Rourke, S. Rao Kosaraju, and N. Megiddo. Computing circular separability. *Discrete & Computational Geometry*, 1:105–113, 1986.

[9] C. Seara. On geometric separability. *PhD thesis, Universitat Politènica de Catalunya*, 2002.

[10] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM J. on Comput.*, 1(2):146–160, 1972.

Algebraically-informed deep networks for associative evolution algebras

Desamparados Fernández-Ternero*[1], Víctor Manuel Gómez-Sousa†[1], Juan Núñez-Valdés‡[1], and Eduardo Paluzo-Hidalgo§[2]

[1]Department of Geometry and Topology. Faculty of Mathematics, University of Seville, Seville, Spain.
[2]Department of Applied Mathematics I. University of Seville, Seville, Spain.

Abstract

In this paper, we extend the application of the algebraically-informed deep networks (AIDNs) introduced in [4] in the case of evolution algebras. For the associative evolution algebra, the performance of AIDNs is tested for known theoretical results. This is a first step towards neural-network-aided classification of evolution algebras.

1 Introduction

Evolution algebras were firstly introduced by Tian in his Ph.D. Thesis [6] in 2004 and later published in a book in 2008 [7]. These types of algebras belong to the family of genetic algebras and have direct applications in non-Mendelian genetics [8]. In addition, its applications to other branches of mathematics are numerous, being connected, among others, with graph theory, stochastic processes and Markov chains, group theory and mathematical physics.

Classification problems in evolution algebras generally involve a large number of nonlinear equations. This makes classifying evolution algebras a difficult task. For instance, the classification of those evolution algebras whose evolution operator (the main operator of the algebra) is a homomorphism is still incomplete [2]. The purpose of this paper is to help in this task by searching for representations of evolution algebras through neural networks, which will give us an approximation of possible solutions to the problem that we can later prove theoretically. To verify the effectiveness of this method, we will use it for classifications already achieved theoretically, such as the classification of associative evolution algebras [1] or that of those evolution algebras whose evolution operator is a derivation [3].

Algebraically-informed deep networks were introduced in [4]. There, the authors provide a correspondence between a representation of an algebraic structure and feed-forward neural networks where each generator is associated with a neural network. To find an appropriate representation, those networks are trained using a traditional training algorithm which error function is based on the relations needed to be satisfied by the generators. In the case of associative evolution algebras, it is known that its structure matrix is diagonal for any dimension. In this paper, we adapt the AIDN implementation for associative evolution algebras and test its performance.

This paper is organized as follows: in Section 2, the basic concepts of evolution algebras, neural networks and AIDN are provided. Next, in Section 3, AIDNs are extended for associative evolution algebras and experiments are depicted. Finally, in Section 4, some conclusions and future work are described.

2 Background

This section provides a brief introduction to evolution algebras, neural networks and algebraically-informed deep networks over the real field. However, it can be easily extended to the complex field.

2.1 Evolution algebras

An algebra $E \equiv (E, +, \cdot)$ is said to be an *evolution algebra* if there exists a basis $\mathcal{B} = \{e_i\}_{i=1}^n$ of E such that $e_i \cdot e_j = 0$, for all $i \neq j$. Since $\mathcal{B}$ is a basis, the product $e_j \cdot e_j = e_j^2$ can be written as $\sum_{i\in\Lambda} a_{ij}e_i$, for some *structure constants* $a_{ij} \in \mathbb{R}$. So, the product on E is determined by the *structure matrix* $A = (a_{ij})$.

In general, evolution algebras are non-associative. However, in this paper we will work with those that are. This type of evolution algebras have been studied in depth in [1], where it is shown the following theorem

Theorem 1 *Let E be an evolution algebra. It is equivalent:*

1. *E is associative.*

*Email: desamfer@us.es
Research supported by national projects PID2020-118753GB-100 and PID2020-114474GB-I00, autonomic project ProyExcel-00827 and autonomic P.A.I. Group FQM-326.
†Email: victor.manuel.gomez.sousa@gmail.com
Research supported by national project PID2020-118753GB-100 and autonomic P.A.I. Group FQM-326.
‡Email: jnvaldes@us.es
Research supported by autonomic P.A.I. Group FQM-326.
§Email: epaluzo@us.es
Research partially supported by national project PID2019-107339GB-I00.

2. $e_i \cdot e_j^2 = 0$, *for all* $i \neq j$.

3. $a_{ij}a_{ki} = 0$, *for all* $i \neq j$ *and for all* k.

4. *There exists a rearrangement of the basis* $\mathcal{B}$ *such that the structure matrix has the form*

$$\begin{pmatrix} D_{r\times r} & 0_{r\times s} \\ M_{s\times r} & 0_{s\times s} \end{pmatrix},$$

where $r + s = n$ *and*

- $D_{r\times r}$ *is a diagonal matrix of order* r.
- $M_{s\times r}$ *is a* $s \times r$ *matrix.*
- $0_{r\times s}$ *and* $0_{s\times s}$ *are null matrices of dimension* $r \times s$ *and* $s \times s$*, respectively.*

An evolution algebra is said to be *non-degenerate* if $e_j^2 \neq 0$, for all j. In this case, the previous theorem turns out to be

Corollary 2 *Let* E *be a non-degenerate evolution algebra with evolution operator* L*. Then, the following assertions are equivalent*

1. E *is associative.*

2. $e_i \cdot e_j^2 = 0$, *for all* $i \neq j$.

3. $a_{ij}a_{ki} = 0$, *for all* $i \neq j$ *and for all* k.

4. *The structure matrix is a diagonal matrix with nonzero diagonal elements.*

For example, the evolution algebra with basis $\{e_1, e_2, e_3\}$ and product defined by

$$\begin{aligned} e_1^2 &= e_1, \\ e_2^2 &= 2e_2, \\ e_3^2 &= 3e_3, \end{aligned}$$

is a non-degenerate associative evolution algebra, since its structure matrix is $Diag(1, 2, 3)$.

2.2 Neural networks

A *neural network* is a function $Net : \mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out}}$ defined by a composition of *layer functions* $f_i : \mathbb{R}^{n_i} \to \mathbb{R}^{m_i}$, that is to say, $Net = f_L \circ \cdots \circ f_1$. The layer functions are of the form $f_i(x) = \alpha_i(W_i(x) + b_i)$, where W_i is a $m_i \times n_i$ matrix, b_i is a vector in $\mathbb{R}^{m_i}$ and α_i is the *activation function*, a chosen function (generally non linear) applied coordinate-wisely to an input vector.

We will denote the set of neural networks of the form Net: $\mathbb{R}^n \to \mathbb{R}^n$ as $\mathcal{N}(\mathbb{R}^n)$. This set is closed under composition of functions and hence has a natural algebraic structure $(\mathcal{N}(\mathbb{R}^n), +, \circ)$.

The set of weights of a neural network are trained using a gradient-based training algorithm induced by a loss function $\mathcal{L}$ that measures how far is the output of the neural network from the desired output. There exist plenty of different hyperparameters to be tuned in a neural network and a training algorithm. To name a few: the number of layers and nodes, the loss function, the learning rate, the number of epochs, among others.

2.3 Algebraically-informed deep networks (AIDN)

Let $S = \{s_i\}_{i=1}^n$ be a set of formal symbols (generators) and $R = \{r_i\}_{i=1}^k$ a formal set of equations satisfying these generators. The system $\langle S \mid R\rangle$ is called a *presentation*.

Presentations can encode different algebraic objects depending on the algebraic operations that we are willing to allow while solving the algebraic equations of R. For example, if we allow operations of addition and scalar multiplication by elements of a field satisfying the axioms implied in the definition of vector space, in addition to a bilinear product, then the resulting algebraic structure induced by the presentation $\langle S \mid R\rangle$ is an algebra. Depending on the properties that we allow in the product, this algebra can be, for example, associative or unitary.

We are interested in finding neural networks $\{f_i(x; \theta_i)\}_{i=1}^n$, where $\theta_i \in \mathbb{R}^{k_i}$ is the parameter vector of the network f_i, such that these neural networks correspond to the generators of S and satisfy the same relations of R. Formally, this is equivalent to finding a homomorphism from the algebraic structure $\langle S \mid R\rangle$ to $(\mathcal{N}(\mathbb{R}^n), +, \circ)$.

The AIDN algorithm finds the weights $\{\theta_i\}_{i=1}^n$ of the networks $\{f_i(x; \theta_i)\}_{i=1}^n$ by defining the loss function as follows

$$\mathcal{L}(f_1, \ldots, f_n) = \sum_{i=1}^{k} \parallel \mathcal{F}(r_i) \parallel_2^2,$$

where $\mathcal{F}(r_i)$ is the relation r_i written in terms of the networks $\{f_i(x; \theta_i)\}$ and $\| \cdot \|_2$ is the L^2 norm. This loss function is minimized using any known neural network training algorithm such as gradient descent or RMSprop [5].

3 AIDN for associative evolution algebras

In the case of associative evolution algebras, we considered a fixed canonical basis $\mathcal{B}$ in dimension n, hence we want to find a set of generators corresponding to the structure constants $a_{ij} \in \mathbb{R}$. Let us describe, firstly, the set of relations needed to be satisfied. The third statemen of Theorem 1 specifies the rest of the relations needed for the desired representation.

Finally, we desired to find those algebras that are not trivial (i.e. its structure matrix is not null), so we can add a final set of relations for the representation, guaranteeing that a column in the structure matrix is not null by minimizing $\frac{1}{\sum_{i=1}^n |a_{ij}|}$ for all $j \in [\![1, n]\!]$[1].

[1]Let us denote $\{1, \ldots, n\}$ by $[\![1, n]\!]$.

Therefore, an associative algebra representation is given by n^2 generators $\{a_{ij}\}_{i,j\in[\![1,n]\!]}$ subject to the following relations:

1. For all $i \neq j$ and for all k: $a_{ij}a_{ki} = 0$.

2. For all $j \in [\![1, n]\!]$: minimize $\frac{1}{\sum_{i=1}^{n} |a_{ij}|}$.

Let us remark that the first set of relations is composed of $(n^2 - n) \cdot n$ equations, and the second set is composed of n equations. As mentioned above, the basis of the associative evolution algebra will be fixed as the canonical basis which is composed of one-hot vectors and the definition of the product induced by the obtained generators. Therefore, the AIDN representation of the algebra will be based on finding a set of feed-forward neural networks $\{f_{ij}\}_{i,j\in[\![1,n]\!]} \subset \mathcal{N}(\mathbb{R})$ associated to each of the structure constants which satisfy the conditions stated above. The architecture of those neural networks can be tuned by the user. In our case, we decided to use very simple feed-forward neural networks with just one hidden layer $(1 \times 12 \times 1)$ and a linear activation function. In the case of higher dimensional representations, the number of layers and nodes can be increased. The neural networks were trained using the RMSprop training algorithm and the structure matrices were computed for dimensions 2, 3, and 4. Each of the generators is the result of applying neural networks to a specific real number parameter that was used during the training process as input data. In Table 1, examples of the matrices obtained are displayed with the error values of the error function which history is depicted in Figure 1. Let us remark that, as expected, the structure matrices obtained are diagonal matrices, satisfying the already known theoretical results.

Dimension	Structure matrix
2	$\begin{pmatrix} 145.22 & 0 \\ 0 & -143.69 \end{pmatrix}$
3	$\begin{pmatrix} 19.94 & 0 & 0 \\ 0 & -20.81 & 0 \\ 0 & 0 & 20.68 \end{pmatrix}$
4	$\begin{pmatrix} -23 & 0 & 0 & 0 \\ 0 & -23.93 & 0 & 0 \\ 0 & 0 & -22.74 & 0 \\ 0 & 0 & 0 & -22.27 \end{pmatrix}$

Table 1: Examples of structure matrices obtained after training AIDNs are depicted together with the mean of the loss values for all relations for different dimensions. The reached loss values were of the order of 10^{-4} to 10^{-5}.

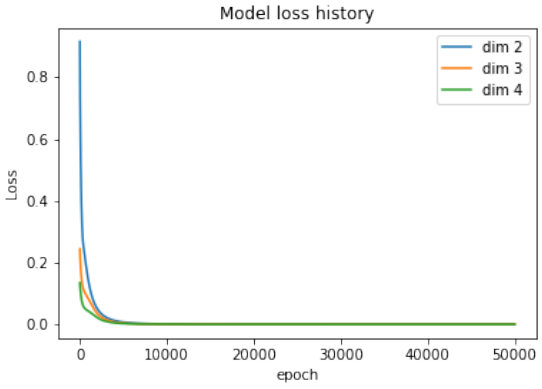


Figure 1: Neural networks were trained for 50000 epochs with a learning rate of 10^{-3}. As depicted, the evolution of the loss function during the training algorithm was similar in the three cases.

4 Conclusions

In this paper, we have applied AIDNs to associative evolution algebras. These networks were trained for different dimensions. After the training process, the matrices obtained satisfied the known theoretical results for the structure matrix of associative evolution algebras, known to be diagonal. In future work, we plan to apply AIDNs towards patterns discovery for other types of evolution algebras.

Code availability The code of the experiments is available in the following GitHub repository `https://github.com/Cimagroup/AIDN-for-Evolution-Algebras`.

References

[1] Desamparados Fernández-Ternero, Víctor M. Gómez-Sousa, and Juan Núñez-Valdés. "A Characterization of Associative Evolution Algebras". In: *Contemp. Math.* 4.1 (2023), pp. 42–48.

[2] Desamparados Fernández-Ternero, Víctor M. Gómez-Sousa, and Juan Núñez-Valdés. "Evolution algebras whose evolution operator is a homomorphism". In: *Comput. Math. Methods.* 3.6 (2021). e1200.

[3] Desamparados Fernández-Ternero, Víctor M. Gómez-Sousa, and Juan Núñez-Valdés. "Using the Evolution Operator to Classify Evolution Algebras". In: *Math. Comput. Appl.* 26.3 (2021). Art. 57.

[4] Mustafa Hajij et al. *Algebraically-Informed Deep Networks (AIDN): A Deep Learning Approach to Represent Algebraic Structures*. 2021. arXiv: 2012.01141v3 [cs.LG].

[5] Sebastian Ruder. *An overview of gradient descent optimization algorithms*. 2017. arXiv: 1609.04747v2 [cs.LG].

[6] Jianjun Paul Tian. "Evolution Algebra Theory". PhD thesis. Riverside, CA: University of California, 2004.

[7] Jianjun Paul Tian. *Evolution Algebras and Their Applications*. Berlin, Germany: Springer, 2008.

[8] Jianjun Paul Tian and Petr Vojtechovsky. "Mathematical concepts of evolution algebras in non-Mendelian genetics". In: *Quasigr. Relat. Syst.* 14.1 (2006), pp. 111–122.

On the rectilinear crossing number of complete balanced multipartite graphs and layered graphs

Ruy Fabila-Monroy *[1], Rosna Paul†[2], Jenifer Viafara-Chanchi ‡[1], and Alexandra Weinberger§[2]

[1]Departamento de Matemáticas, CINVESTAV.
[2]Institute for Software Technology, Graz University of Technology, Graz, Austria

Abstract

A rectilinear drawing of a graph is a drawing of the graph in the plane in which the edges are drawn as straight-line segments. The rectilinear crossing number of a graph is the minimum number of pairs of edges that cross over all rectilinear drawings of the graph. Let $n \geq r$ be positive integers. The graph K_n^r, is the complete r-partite graph on n vertices, in which every set of the partition has at least $\lfloor n/r \rfloor$ vertices. The layered graph, L_n^r, is an r-partite graph on n vertices, in which for every $1 \leq i \leq r-1$, all vertices in the i-th partition are adjacent to all vertices in the $(i+1)$-th partition. In this paper, we give upper bounds on the rectilinear crossing numbers of K_n^r and L_n^r.

1 Introduction

Let G be a graph on n vertices and let D be a drawing of G. The crossing number of D is the number, $\mathrm{cr}(D)$, of pairs of edges that cross in D. The *crossing number* of G is the minimum crossing number, $\mathrm{cr}(G)$, over all drawings of G in the plane. A *rectilinear drawing* of G is a drawing of G in the plane in which its vertices are points in general position, and its edges are drawn as straight-line segments joining these points. The *rectilinear crossing number* of G, is the minimum crossing number, $\overline{\mathrm{cr}}(G)$, over all rectilinear drawings of G in the plane.

Computing crossing and rectilinear crossing numbers of graphs are important problems in Graph Theory and Combinatorial Geometry. For a comprehensive review of the literature on crossing numbers, we refer the reader to Schaefer's book [1].

Most of the research on crossing numbers have been focused around the complete graph, K_n, and the complete bipartite graph $K_{n,m}$. For the complete graph, Hill [2] gave the following drawing of K_n; see Figure 1 for an example. Place half of the vertices equidistantly on the top circle of a cylinder, and the other half equidistantly on the bottom circle. Join the vertices with geodesics on the cylinder. Hill showed that the following number, $H(n)$, is the crossing number of this drawing, and it is now conjectured to be optimal. Let

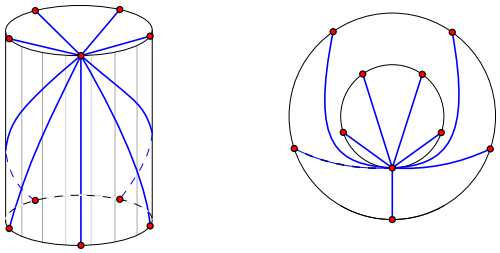


Figure 1: An example of Hill's drawings of K_8, where here for convenience only the edges of one vertex are drawn. Left: The drawing on a cylinder. Right: An equivalent representation of Hill's drawings via concentric cycles.

$$H(n) := \frac{1}{4}\left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor \left\lfloor \frac{n-2}{2} \right\rfloor \left\lfloor \frac{n-3}{2} \right\rfloor$$

Conjecture 1 (Harary-Hill [3])

$$\mathrm{cr}(K_n) = H(n).$$

For the complete bipartite graph, Zarankiewicz gave a rectilinear drawing with the following number, $Z(n,m)$, as crossing number of this drawing, and it is now conjectured to be optimal. Let

$$Z(n,m) := \left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor \left\lfloor \frac{m}{2} \right\rfloor \left\lfloor \frac{m-1}{2} \right\rfloor$$

and

$$Z(n) := Z(n,n).$$

Conjecture 2 (Zarankiewicz [4])

$$\mathrm{cr}(K_{n,m}) = Z(n,m).$$

*Email: ruyfabila@math.cinvestav.edu.mx
†Email: ropaul@ist.tugraz.at Supported by the Austrian Science Fund (FWF) grant W1230.
‡Email: viafara@math.cinvestav.mx
§Email: weinberger@ist.tugraz.at Supported by the Austrian Science Fund (FWF) grant W1230.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 734922.

The number $Z(n,m)$ is also conjectured to be the general optimal crossing number, directly implying the following conjecture.

Conjecture 3

$$\overline{\mathrm{cr}}(K_{n,m}) = \mathrm{cr}(K_{n,m}).$$

Much less is known for the rectilinear crossing number of the complete graph. For $n \geq 10$, it is known that

$$\mathrm{cr}(K_n) < \overline{\mathrm{cr}}(K_n).$$

In contrast to the case of the complete bipartite graph, there is no conjectured value for $\overline{\mathrm{cr}}(K_n)$, nor drawings conjectured to be optimal. The best bounds to date are

$$0.379972\binom{n}{4} < \overline{\mathrm{cr}}(K_n) < 0.380445\binom{n}{4} + O(n^3).$$

The lower bound is due to Ábrego, Fernández-Merchant, Leaños, and Salazar [5], and the upper bound to Aichholzer, Duque, Fabila-Monroy, García-Quintero, and Hidalgo-Toscano [6]. It is known that

$$\lim_{n\to\infty} \frac{\overline{\mathrm{cr}}(K_n)}{\binom{n}{4}} = \overline{q},$$

for some positive constant $\overline{q}$; this constant is known as the *rectilinear crossing constant*.

Let $K_{n_1 n_2 \ldots n_r}$ be the complete r-partite graph with n_i vertices in the i-th set of the partition; and let K_n^r be the complete balanced r-partite graph in which there are at least $\lfloor n/r \rfloor$ vertices in every partition set. Harborth [7] gave a drawing that provides an upper bound for $\mathrm{cr}(K_{n_1 n_2 \ldots n_r})$; and gave an explicit formula for this number. He claims that for the case of $r = 3$, his drawing can be made rectilinear. More recently, Gethner, Hogben, Lidický, Pfender, Ruiz and Young [8] independently studied the problem of the crossing number and rectilinear crossing numbers of complete balanced r-partite graphs. For $r = 3$, they obtain the same bound as Harborth; and their drawing is rectilinear.

Let $H(n,r)$ be the number of crossings in Harborth's drawing for $\mathrm{cr}(K_n^r)$. Due to the complexity of the formula, we use the following approximation to $H(n,r)$ instead. (All missing proofs can be found in the Appendix)

Lemma 4 *If n is a multiple of r, then*

$$H(n,r) \leq \frac{1}{16}\left(\frac{r-1}{r}\right)^2 \left(\frac{n^4}{4} - 2n^3\right) + O(n^2).$$

In this paper, we mainly focus on the rectilinear crossing number of K_n^r. If n is fixed and r tends to n, then K_n^r tends to K_n. Thus, we believe that studying the rectilinear crossing number of K_n^r might shed some light on how optimal rectilinear drawings of K_n look like.

2 Random Embeddings into Drawings of K_n with Few Crossings

Suppose that we have a drawing (that can be rectilinear but doesn't have to be) D' of K_n. If $\mathrm{cr}(D')$ is small, it might be a good idea to use this drawing to produce a drawing of a graph G on n vertices. Let D be the drawing of G that is produced by mapping the vertices of G randomly to the vertices of D', and where the edges are drawn as their corresponding edges of D'. We call D a *random embedding* of G into D'. In every 4-tuple of vertices of D', there are three pairs of independent edges, which could cross. Of these three pairs at most one pair is crossing. For every pair of independent edges of G, we have a possible crossing in D; thus, the probability that this pair of edges is mapped to a pair of crossing edges is equal to

$$\frac{1}{3} \cdot \frac{\mathrm{cr}(D')}{\binom{n}{4}}.$$

By defining, for every pair of independent edges of G, an indicator random variable with value equal to one if the edges cross and zero otherwise, we obtain the following result where $||G||$ is the number of edges in G and $d(v)$ is the degree of a vertex v.

Lemma 5

$$\mathrm{E}(\mathrm{cr}(D)) = \frac{cr(D')}{3\binom{n}{4}} \left(\binom{||G||}{2} - \sum_{v \in V(G)} \binom{d(v)}{2} \right).$$

Suppose that n is a multiple of r. For the crossing number of K_n^r, we use Lemma 5 and Hill's drawing of K_n to obtain the following.

Theorem 6 *Suppose that n is a multiple of r. Let D be a random embedding of K_n^r into Hill's drawing of K_n. Then,*

$$E(\mathrm{cr}(D)) \leq \frac{1}{16}\left(\frac{r-1}{r}\right)^2 \left(\frac{n^4}{4} - \frac{3n^3}{2}\right) + O(n^2).$$

In [8], the authors obtain same bound on $\mathrm{cr}(K_n^r)$ by considering a random mapping of the vertices of K_n^r into a sphere, and then joining the corresponding vertices with geodesics. This type of drawing is called a *random geodesic spherical drawing.* In 1965, Moon [9], showed that the expected number of crossings of a random geodesic spherical drawing of K_n is equal to

$$\frac{1}{16}\binom{n}{2}\binom{n-2}{2} = H(n) - O(n^3);$$

which explains why the bound of Theorem 6 matches the bound of [8].

Note that by Lemma 4 together with Theorem 6, it holds that

$$E(\mathrm{cr}(D)) - H(n,r) \leq \frac{1}{32}\left(\frac{r-1}{r}\right)^2 n^3 + O(n^2) = O(n^3).$$

Thus, the random embedding gives an upper bound on $\mathrm{cr}(D)$ that matches the conjectured value up to the leading term, but it is a little worse in the lower terms.

2.1 Rectilinear crossing number of K_n^r

Let $\overline{D}$ be a random embedding of K_n^r into an optimal rectilinear drawing of K_n.

Theorem 7 *Let r be a positive integer and n a multiple of r. Then*

$$\begin{aligned}\overline{\mathrm{cr}}(K_n^r) &\leq E(\mathrm{cr}(\overline{D})) \\ &\leq \frac{\overline{q}}{4!}\left(\frac{r-1}{r}\right)^2 n^4 + o(n^4) \\ &< 0.015852\left(\frac{r-1}{r}\right)^2 n^4 + o(n^4).\end{aligned}$$

For a lower bound we have the following.

Theorem 8 *Let r be a positive integer and n a multiple of r. Then*

$$\overline{\mathrm{cr}}(K_n^r) \geq \overline{\mathrm{cr}}(K_r)\left(\frac{n}{r}\right)^4.$$

Theorems 7 and 8 imply the following.

Corollary 9 *Let $r = r(n)$ be a monotone function of n such that $r \to \infty$ as $n \to \infty$. Then*

$$\lim_{n\to\infty} \frac{\overline{\mathrm{cr}}(K_n^r)}{\binom{n}{4}} = \overline{q}.$$

In both [7] and [8], it is conjectured that

$$\mathrm{cr}\left(K_n^3\right) = \overline{\mathrm{cr}}\left(K_n^3\right).$$

Using the order type database [10], we have verified the following.

Observation 10

$$\overline{\mathrm{cr}}\left(K_8^4\right) = 8 \text{ and } \overline{\mathrm{cr}}\left(K_9^4\right) = 15.$$

On the other hand

$$\mathrm{cr}\left(K_8^4\right) \leq H(8,4) = 6 \text{ and } \mathrm{cr}\left(K_9^4\right) \leq H(9,4) = 15.$$

See Figure 2 for an example. From the above results we conjecture the following.

Conjecture 11 *There exists a natural number $n_0 > 9$ such that for all $n \geq n_0$,*

$$\mathrm{cr}\left(K_n^4\right) < \overline{\mathrm{cr}}\left(K_n^4\right).$$

Further, we pose the following problem.

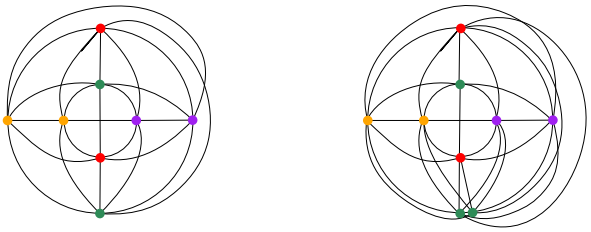


Figure 2: A drawing of K_8^4 with 6 crossings (left) and K_9^4 with 15 crossings (right).

Open problem 12 *Let $r \geq 4$ be a constant positive integer. Does*

$$\lim_{n\to\infty} \frac{\overline{\mathrm{cr}}(K_n^r)}{\binom{n}{4}} = \overline{q}\left(\frac{r-1}{r}\right)^2 ?$$

We believe that finding a good rectilinear drawing of K_n^r, even for the case of $r = 4$, will help in understanding how crossing optimal rectilinear drawings of K_n look like. Theorem 7 implies that

$$\overline{\mathrm{cr}}(K_n^4) \leq 0.00892n^4 + O(n^3).$$

We have found an explicit rectilinear drawing of K_n^4 with

$$0.00953n^4 + O(n^3)$$

crossings.

2.2 Layered graphs

Let r be a positive integer and let n be a multiple of r. The *layered graph*, L_n^r, is the graph defined as follows. Its vertex set is partitioned into sets $V_1, \ldots, V_r$, each consisting of n/r vertices. We call the set V_i, the i-th layer of L_n^r. The edge set of L_n^r is given by

$$\{uv : u \in L_i \text{ and } v \in L_{i+1}, \text{ for } i = 1, \ldots, r-1\};$$

that is, the edges are exactly all possible edges between vertices on consecutive layers. The random embedding into crossing optimal drawings of K_n, seems to give drawings of K_n^r with crossings close to the minimum value. In this section, we show that for layered graphs, the random embedding gives drawings with considerably more crossings than the optimal drawings. Note that $L_n^2 = K_{n/2,n/2}$, and $L_n^3 = K_{2n/3,n/3}$. In the reminder of the section assume that $r \geq 4$, and for convenience assume also that n/r is even. Using the random embedding technique into Hill's drawing of K_n, we obtain the following upper bound.

Theorem 13

$$\mathrm{cr}(L_n^r) \leq \frac{(r-1)^2}{16r^4}n^4 + O(n^3).$$

We can improve on this bound with the following rectilinear drawing, $\overline{D}$, of L_n^r. Since we are describing a rectilinear drawing it is sufficient to specify the location of the vertices of $\overline{D}$:

- $$V_1 := \left\{ \left(1 + \frac{1}{2^j}, 0\right) : 1 \leq j \leq \frac{n}{r} \right\};$$
- for $2 \leq i \leq r-1$,

  $$V_i := \left\{(i, j) : 1 \leq j \leq \frac{n}{2r}\right\} \cup \left\{(i, -j) : 1 \leq j \leq \frac{n}{2r}\right\};$$

  and
- $$V_r := \left\{ \left(r + \frac{1}{2^j}, 0\right) : 1 \leq j \leq \frac{n}{r} \right\}.$$

Thus, the first and r-th layer are horizontal and the remaining layers are vertical. The vertices of $\overline{D}$ are not in general position; a small random perturbation of the vertices of $\overline{D}$ is sufficient to ensure general position, while at the same time not changing the number of crossings.

Theorem 14

$$\mathrm{cr}(\overline{D}) = \frac{r-2}{4r^4} n^4 + O(n^3).$$

The leading constant in Theorem 14 is better, for all $r \geq 4$, than the constant obtained by the random embedding into Hill's drawing of K_n.

For $r = 4$, Theorem 14 implies that

$$\mathrm{cr}(\overline{D}) \leq \frac{3}{1024} n^4 + O(n^3) \leq 0.00293 n^4 + O(n^3).$$

We have found a rectilinear drawing of L_n^4 with considerable fewer crossings; see Figure 3 for a depiction.

Proposition 15 *There exists a rectilinear drawing D' of L_n^4 such that*

$$\mathrm{cr}(D') = \frac{56}{32768} n^4 + O(n^3) \leq 0.00171 n^4 + O(n^3).$$

We believe that the bound of Theorem 14 is far from optimal, and that computing $\mathrm{cr}(L_n^r)$ and $\overline{\mathrm{cr}}(L_n^r)$ are interesting problems.

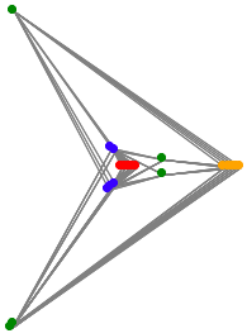


Figure 3: A rectilinear drawing of L_{20}^4

References

[1] M. Schaefer, *Crossing numbers of graphs*. CRC Press, 2018.

[2] F. Harary and A. Hill, "On the number of crossings in a complete graph," *Proceedings of the Edinburgh Mathematical Society*, vol. 13, no. 4, pp. 333–338, 1963.

[3] R. K. Guy, "A combinatorial problem," *Nabla (Bulletin of the Malayan Mathematical Society)*, vol. 7, pp. 68–72, 1960.

[4] K. Zarankiewicz, "On a problem of P. Turan concerning graphs," *Fund. Math.*, vol. 41, pp. 137–145, 1954.

[5] B. M. Ábrego, S. Fernández-Merchant, J. Leaños, and G. Salazar, "A central approach to bound the number of crossings in a generalized configuration," in *The IV Latin-American Algorithms, Graphs, and Optimization Symposium*, vol. 30 of *Electron. Notes Discrete Math.*, pp. 273–278, Elsevier Sci. B. V., Amsterdam, 2008.

[6] O. Aichholzer, F. Duque, R. Fabila Monroy, O. E. García-Quintero, and C. Hidalgo-Toscano, "An ongoing project to improve the rectilinear and the pseudolinear crossing constants." Preprint.

[7] H. Harborth, "Über die Kreuzungszahl vollständiger, n-geteilter Graphen," *Mathematische Nachrichten*, vol. 48, no. 1-6, pp. 179–188, 1971.

[8] E. Gethner, L. Hogben, B. Lidickỳ, F. Pfender, A. Ruiz, and M. Young, "On crossing numbers of complete tripartite and balanced complete multipartite graphs," *Journal of Graph Theory*, vol. 4, no. 84, pp. 552–565, 2017.

[9] J. W. Moon, "On the distribution of crossings in random complete graphs," *J. Soc. Indust. Appl. Math.*, vol. 13, pp. 506–510, 1965.

[10] O. Aichholzer, F. Aurenhammer, and H. Krasser, "Enumerating order types for small point sets with applications," *Order*, vol. 19, no. 3, pp. 265–281, 2002.

Measuring cocircularity in a point set

Andrea de las Heras*[1], Guillermo Esteban†[2, 3], Delia Garijo‡[4], Clemens Huemer§[1], Antoni Lozano¶[1], Nicolau Oliver‖[1], and David Orden**[3]

[1]Universitat Politècnica de Catalunya, Barcelona, Spain
[2]Carleton University, Ottawa, Canada
[3]Universidad de Alcalá, Alcalá de Henares, Spain
[4]Universidad de Sevilla, Sevilla, Spain

Abstract

In a given set S of n points in the plane, how close are four points of S to be cocircular? We define several measures to study this question, and present bounds on this almost-cocircularity in a point set. Algorithms for cocircularity are presented as well.

1 Introduction

A set S of points in the plane is in general position if no three points of S are collinear and no four points are cocircular. Most algorithms in Computational Geometry require the input points to be in general position. This simplifies the design of the algorithms as most degenerate situations arise from collinearity, but also from cocircularity. It is well known that any sufficiently large set of points contains three points that are almost collinear. In particular, a result by Erdős and Sekeres [1] states that for every set S of 2^n points in the plane, the largest angle defined by points of S is bounded from below by $\pi \cdot (1 - 1/n)$.

We study how close are four points from S to being cocircular. We define several measures of cocircularity in point sets and give bounds on these measures.

On the algorithmic side, the minimum area triangle defined by points of a given set S, which can be considered a measure of collinearity, can be found in $O(n^2)$ time using duality [3, 4]. Our goal is to design algorithms to find the tuple of four points of S closest to cocircularity. We present several $O(n^3)$-time algorithms for this problem. A related, and well studied, algorithmic problem to our research is computing the annulus of smallest width that contains S, see e.g. [5] and references therein.

In Section 2 we define three measures for cocircularity, the *Thales* cocircularity, the *Voronoi* cocircularity, and the *Determinant* cocircularity. We present properties and differences among them. In Section 3 we show bounds on the Thales cocircularity, and Section 4 is mainly devoted to the design of algorithms.

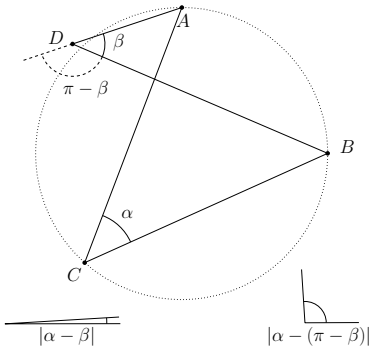


Figure 1: Angles used in Thales cocircularity.

2 Measures of cocircularity

Definition 1 *The Thales cocircularity of four points A, B, C, D is $\mathcal{T}(A, B, C, D) = \min_P\{\min\{|\alpha-\beta|, |\alpha-(\pi-\beta)|\}\}$, where $\alpha = \angle ACB$ and $\beta = \angle ADB$, and the minimum $\min_P$ is taken over all permutations P of the four points A, B, C, D. See Figure 1.*

The Thales cocircularity is motivated by Thales' theorem, also known as the inscribed angle theorem. $\mathcal{T}(A, B, C, D)$ is invariant under translation and scaling.

Definition 2 *The Determinant cocircularity $\mathcal{D}(A, B, C, D)$ of four points $A = (A_x, A_y)$, $B = (B_x, B_y)$, $C = (C_x, C_y)$, $D = (D_x, D_y)$ is the absolute value of the determinant:*

$$\begin{vmatrix} A_x & A_y & A_x^2 + A_y^2 & 1 \\ B_x & B_y & B_x^2 + B_y^2 & 1 \\ C_x & C_y & C_x^2 + C_y^2 & 1 \\ D_x & D_y & D_x^2 + D_y^2 & 1 \end{vmatrix}$$

*Email: andrea.de.las.heras@estudiantat.upc.edu.
†Email: g.esteban@uah.es.
‡Email: dgarijo@us.es.
§Email: clemens.huemer@upc.edu.
¶Email: antoni.lozano@upc.edu.
‖Email: nicolau.oliver@estudiantat.upc.edu.
**Email: david.orden@uah.es.

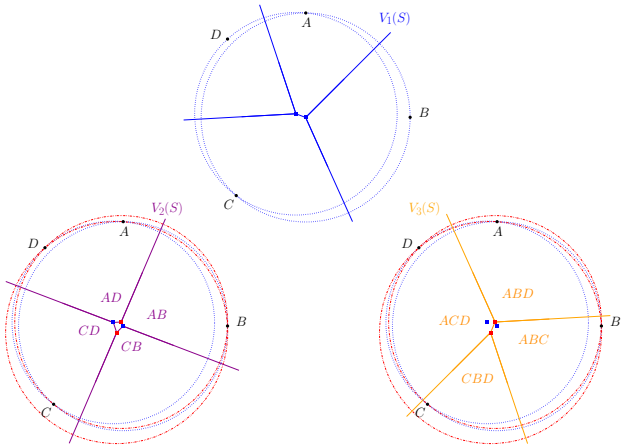


Figure 2: Top: $V_1(S)$ of a set of four almost cocircular points $S = \{A, B, C, D\}$. Bottom-left: $V_2(S)$, where for each unbounded cell the closest pair of points of S is indicated, and the bounded cell has AC as the closest pair of points. Bottom-right: $V_3(S)$, where in each cell the closest triplet of points of S is indicated. Centers of circles are depicted as red or blue squares.

The Determinant cocircularity is a very common tool to check if four points are cocircular: four points A, B, C, D in the plane lie on a common circle if and only if $\mathcal{D}(A, B, C, D) = 0$. More general, a determinant test is often used to check if $d + 1$ points in $\mathbb{R}^d$ are in general position, since the volume the simplex defined by $d + 1$ points is given by a determinant. $\mathcal{D}(A, B, C, D)$ is invariant under translation, but under a scaling by a factor c, the determinant varies in a factor of c^4. We therefore only consider the Determinant cocircularity in Section 4 on algorithms to show a relation to the 4-SUM problem.

We now introduce another measure of cocircularity. The order-k Voronoi diagram of a point set S, denoted by $V_k(S)$, is a partition of the plane into cells that have the same k closest points of S. The order-1 Voronoi diagram of four cocircular points is composed of one vertex of degree four and four rays from it, the vertex being the center of the circle. If we perturb the points slightly so that the cocircularity disappears, the Voronoi diagram changes: the vertex of degree four gets replaced by two vertices of degree three, connected by a short segment (there are two rays from each of them). Each vertex is the center of a circle through three of the four points considered, with none of them in the interior. See Figure 2, top. Note that there can be shorter segments in $V_2(A, B, C, D)$ or in $V_3(A, B, C, D)$, see Figure 2, bottom.

Definition 3 *The* Voronoi cocircularity of four points A, B, C, D, *denoted by* $\mathcal{V}(A, B, C, D)$, *is the length of the shortest edge in all* $V_k(A, B, C, D)$ *for* $k = 1, 2, 3$. $\mathcal{V}(A, B, C, D)$ *is zero if some* $V_k(A, B, C, D)$ *has a vertex of degree four.*

$\mathcal{V}(A, B, C, D)$ is invariant under translation. Also, under scaling by a factor c, the length of the shortest Voronoi edge scales linearly with c.

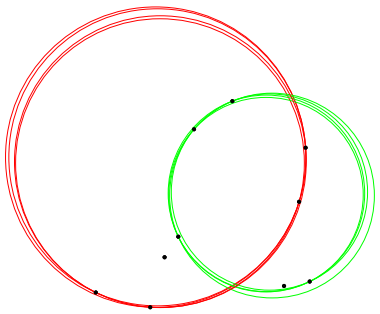


Figure 3: Green (resp., red) circles pass through the four most cocircular points of a set S of $n = 10$ points according to Voronoi (resp., Thales) measure.

Definition 4 *The* Thales/Determinant/Voronoi cocircularity of a set S of points *is the minimum of the Thales/Determinant/Voronoi cocircularity among all 4-tuples of points of* S.

2.1 Properties and differences

All the measures described in the previous section are zero when the points are cocircular.

We next give a formula for the length of an edge in a Voronoi diagram of S. Each endpoint of such an edge is the center of a circle passing through three points of S, and where two of these three points are the same for both circles.

Proposition 5 *Let* A, B, C, D *be four points in the plane, and let* C_A, C_B *be the centers of the circles through* C, A, D *and* C, B, D, *respectively. If* C_A *and* C_B *are on the same side of the line* $\overline{CD}$, *then*

$$|C_A C_B| = \frac{|CD|}{2}(\big||\cot\beta| - |\cot\alpha|\big|),$$

where α *and* β *are the angles* $\angle CAD$ *and* $\angle CBD$, *respectively. Otherwise,*

$$|C_A C_B| = \frac{|CD|}{2}(|\cot\beta| + |\cot\alpha|).$$

We show in Figure 3 that the measures of Voronoi and Thales cocircularity of a point set are different, in the sense that they do not yield the same four-tuple of points minimizing the cocircularity measure.

3 Bounds on cocircularity

We present two bounds on the measure of Thales cocircularity.

Algorithm 1 Detecting cocircularities with inversions

for every point $p_i \in S$ **do**

• Invert all points of $S \backslash \{p_i\}$ with respect to the center p_i and radius $r = 1$.

• Execute as subroutine the sweep-line algorithm to detect collinearities, see [3].

end for

Proposition 6 *For any set S of n points and any two points $A, B \in S$ there exist $C, D \in S$ such that $\mathcal{T}(A, B, C, D) \leq \pi/(n-3)$.*

Theorem 7 *For any set S of n points in convex position, there exist $A, B, C, D \in S$ such that $\mathcal{T}(A, B, C, D) \in O\left(\frac{1}{n^2}\right)$.*

4 Algorithms

Another line of research is to detect these cocircularities. The decision problem is: Are there four cocircular points in S? The optimization problem is: Find the 4-tuple of points of S that minimizes the measure.

4.1 Inversions

Proposition 8 *There is an $O(n^3)$-time algorithm that decides whether there exist four cocircular points in a set of n points, using inversions.*

An *inversion* transformation is determined by two parameters: The center O and the radius R of inversion. Two points P, P' are *i*nverses if they lie in the same half-line with origin at O and the Euclidean distances $|OP|$ and $|OP'|$ satisfy $|OP| \cdot |OP'| = R^2$. We need the following property of inversions:

Property 9 *Given a center O and a radius R of inversion, any circle containing O is inverted into a line.*

Thus, if we perform an inversion at a point A that is cocircular with points B, C, D, then inverting B, C, D results in three points being collinear. Property 9 allows us to propose Algorithm 1. The cost of the subroutine is $O(n^2)$, and it is executed in a loop with $O(n)$ iterations. Thus, the total cost is $O(n^3)$ time and $O(n)$ space.

We note that inversions do not preserve a relationship between the measures of cocircularity and collinearity. Then, by using inversions, we can only solve the decision problem of detecting cocircularities.

4.2 Higher order Voronoi diagrams

Proposition 10 *The Voronoi cocircularity of a set S of n points in general position in the plane can be computed in $O(n^3)$ time.*

To compute the Voronoi cocircularity of S is equivalent to finding the shortest edge among the diagrams $V_k(S)$, for $k = 1, \ldots, n-1$. All diagrams $V_k(S)$ can be obtained in time $O(n^3)$ [4, 6], and the number of edges of a diagram $V_k(S)$ is at most $O(k(n-k))$. Hence, the computation can be done in $O(n^3)$ time.

4.3 Reduction from 4-SUM

The k-SUM problem asks if a list of n integers contains k integers whose sum is zero. This is a prominent problem for $k = 3$ since there is a large list of problems, called 3SUM-hard, that have been proved to be as difficult as 3-SUM; among them, the problem to decide if three points of a given set are collinear [2].

There are easy quadratic-time algorithms to solve both the 3-SUM and the 4-SUM problem in the integer RAM model. As the quadratic-time algorithm for the 4-SUM problem uses hashing, under the real RAM model there is a similar algorithm without hashing of complexity $O(n^2 \log n)$. We prove the following result, which has as a consequence that the problem to decide whether four points are cocircular is 4SUM-hard.

Proposition 11 *Let $[x_1, x_2, \ldots, x_n]$ be a list of n integers, and let S be the set of n points on the parabola $y = x^2$, with coordinates (x_i, x_i^2). Then, S contains four cocircular points if and only if the sum of the x-coordinates of these four points is zero.*

4.4 Cost of the exact problem

We show that to decide whether there exist four cocircular points in a set of n points can be done in $\mathrm{O}(n^3 \log n)$ time in the worst case or, using hashing, in expected $\mathrm{O}(n^3)$ time[1]. Both algorithms can find a solution, thus solving the optimization problem, and work by storing the radius and center of the circle defined by each triplet of points, and then detecting collisions —in the first case, by sorting the circles, which contributes with the additional factor of $\log n$.

Proposition 12 *There is an algorithm working in expected $O(n^3)$ time that decides whether there exist four cocircular points in a set of n points.*

Proof. Suppose S is a set of n points. Our algorithm uses hashing with separate chaining. In particular, H represents a hash table of point sets indexed by triples of points. Let f be the function that, given a set T of three points, returns the triple (x, y, r) s.t. (x, y) and r are the center and radius, resp., of the circle defined by the points in T. We will use the operations:

- **Insertion**: $H.insert(T)$ adds information T with key $f(T)$ to H.

[1]The use of hashing was also proposed in `https://cs.stackexchange.com/questions/49316/largest-set-of-cocircular-points`

Algorithm 2 Exact cocircularity with hashing

given a set S of n points
let H be a hash table of size n^3
initialise H with all entries containing $\emptyset$
for every distinct $T \subseteq S$ with $|T| = 3$ **do**
  if $H[T] = \emptyset$ **then**
    $H[T] \leftarrow T$
  else
    return $H[T] \cup T$
  end if
end for
return "no cocircular points"

- **Search**: $H.search(T)$ returns $\emptyset$ if H does not contain information for key $f(T)$ or a set R s.t. $f(T) = f(R)$, otherwise.

The method is shown in Algorithm 2. Since insertion and search have constant average cost, the for loop and the initialization of H give an $O(n^3)$ expected cost for the whole algorithm. □

Proposition 13 *There is an $O(n^3 \log n)$-time algorithm that decides whether there exist four cocircular points in a set of n points.*

Proof. In Algorithm 3, the hash table of Algorithm 2 has been substituted by a vector V of size n containing 4-tuples (x, y, r, T), where n is the number of points; here T represents a set of three points and (x, y) and r are the center and radius resp. of the circle defined by the points in T. We use a stable sorting algorithm (like *mergesort*) and apply it to V with respect to the first, second, and finally third component of the vector. Since the sorting algorithm is stable, after the three iterations the entries corresponding to the same circle are contiguous in V and can be detected in linear time. The cost of the first and third for loops is $O(n^3)$, while the second for loop has a cost $O(n^3 \log n^3) = O(n^3 \log n)$, which is also the cost of Algorithm 3 in the worst case. □

5 Conclusions

We initiated the study of almost cocircularity in point sets. We chose measures of cocircularity that we considered to be very natural, though other measures could be studied as well. Several questions remain open, and we plan to continue this line of research.

Open problem 14 *Can the decision problem of detecting cocircularities be solved in sub-cubic time?*

Open problem 15 *Can the bound for the Thales measure for convex point sets be extended to arbitrary point sets in general position?*

Algorithm 3 Exact cocircularity

given a set S of n points
let V be a size n^3 vector of tuples (x, y, r, T)
  for reals x, y, r and a three point set T
$i \leftarrow 1$
for every distinct $T \subseteq S$ with $|T| = 3$ **do**
  let (x, y) and r be the center and radius, resp.,
    of the circle defined by the points in T
  $V[i] \leftarrow (x, y, r, T)$
  $i \leftarrow i + 1$
end for
for $i \in \{1, 2, 3\}$ **do**
  sort V with respect to component i
end for
for $i = 1$ to $n^3 - 1$ **do**
  if $V[i][j] = V[i+1][j]$ for each $j \in \{1, 2, 3\}$ **then**
    return $V[i][4] \cup V[i + 1][4]$
  end if
end for
return "no cocircular points"

Open problem 16 *Find families of point sets that are "far away" from having four cocircular points.*

Acknowledgments. This work was initiated at the TOPPING workshop in Barcelona, July 2022. We thank Mercè Claverol, Dolores Lara, Alejandra Martínez, and Pablo Perez for their valuable discussions and contributions. The authors were supported by projects PID2019-104129GB-I00/MCIN/AEI/10.13039/501100011033 and Gen.Cat. 2021 SGR 00266.

References

[1] P. Erdős, G. Szekeres, On some extremum problems in elementary geometry, *Ann. Univ. Sci. Budapest* **3-4** (1960/1961), 53–62.

[2] A. Gajentaan, M. H. Overmars, On a class of $O(n^2)$ problems in computational geometry. *Comput. Geom. Theory Appl.* **5**(3) (1995), 165–185.

[3] H. Edelsbrunner, L. J. Guibas, Topologically sweeping an arrangement, in: *Proc. eighteenth Annual ACM Symposium on Theory of Computing*, 1986, 389–403.

[4] H. Edelsbrunner, J. O'Rourke, R. Seidel, Constructing arrangements of lines and hyperplanes with applications, *SIAM J. on Comput.*, **15**(2), 1986, 341–363.

[5] J. García-López, P. A. Ramos, J. Snoeyink, Fitting a set of points by a circle, *Discrete Comput. Geom.* **20** (1998), 389–402.

[6] K. Mulmuley, On levels in arrangements and Voronoi diagrams, *Discrete Comput. Geom.* **6** (1991), 307–338.

Minsum m watchmen's routes in Stiegl polygons*†

Alireza Bagheri‡[1], Anna Brötzner§[2], Faezeh Farivar¶[3], Rahmat Ghasemi‖[3], Fatemeh Keshavarz-Kohjerdi**[4], Erik Krohn††[5], Bengt J. Nilsson‡‡[2], and Christiane Schmidt△[6]

[1]Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran
[2]Malmö University, Malmö, Sweden
[3]Science and Research Branch, Islamic Azad University, Tehran, Iran
[4]Shahed University, Tehran, Iran
[5]University of Wisconsin, Oshkosh, USA
[6]Linköping University, Campus Norrköping, Sweden

1 Introduction

In the classical watchman route problem, we aim for the shortest closed route R within a polygon P, such that all points of P are visible to some point of R.

Carlsson *et al.* [1] introduced the m-watchmen problem as a generalization of this problem: instead of a single mobile guard, we are given m mobile watchmen (with or without a given starting point) and we aim to find routes for all watchmen, such that all points in P are visible from at least one route and such that the sum of the watchman-route lengths is minimized. Carlsson et al. showed that the problem is NP-hard in simple polygons and provided a polytime algorithm in histograms. Nilsson and Wood [2, 3] gave an $O(n^2m)$ time and $O(n^2)$ storage algorithm for spiral polygons without given starting points for the m watchmen. Nilsson and Schuierer [4] also considered histograms, but altered the objective to minimizing the length of the longest of the m watchmen routes, for which they provided an $O(n^2 \log n)$ algorithm. Also, Mitchell and Wynters [5] considered the minmax criterion. They gave an $O(n^4m)$ algorithm for rectilinear vision in rectilinear monotone polygons and showed that the problem is NP-hard for $m = 2$ in general. Nilsson and Packer [6] gave an approximation algorithm for two watchmen in simple polygons. Packer [7] presented heuristics for both the minmax and the minsum criterion in polygons with and without holes. In this paper, we present an $O(n^2 \cdot \min\{m, n\})$ time and $O(n \cdot \min\{m, n\})$ storage algorithm to compute the minsum set of m watchmen routes given their starting points in a Stiegl polygon—which we define in Section 2. Without starting points the solution is trivially a single point.

2 Watchmen routes' properties in Stiegl polygons

A staircase polygon P, as defined in [8], is called a *Stiegl polygon* if the floor solely consists of one horizontal and one vertical edge, which we call the *base* and the *wall* of P, respectively. Moreover, we call the vertex between the base and the wall the *origin* of P.

Let S be a set of m points in the interior of P which we consider as starting positions for the watchmen. We say that point p sees point q if the segment $[p, q]$ lies in P. In particular, it can partly be on the boundary of P, hence, one can see along a boundary edge of P. We denote the x-coordinate of a point p by $x(p)$, and the y-coordinate by $y(p)$. We furthermore denote the horizontal and the vertical segment that goes through point p and lies inside P with $h(p)$, and $v(p)$, respectively.

Definition 1 *For two points p and q, if $x(p) \geq x(q)$ and $y(p) \leq y(q)$, we say that p* dominates *q.*

Observe that, if p dominates q its visibility polygon is a superset of the visibility polygon of q and that any watchman can be limited to walk to the right and downwards from its starting point, because the bottom-rightmost position on its route w dominates all other positions on w. Hence:

Lemma 2 *A watchman route having starting point s and rightmost x-coordinate x, and lowest y-coordinate y can be replaced by the segment $[s, (x, y)]$ without increasing the minmax or minsum value of the solution.*

*Any connection to an Austrian beer manufacturer is purely coincidental.

†A.Br., B.J.N, and C.S. are supported by the Swedish Research Council project "ILLUMINATE provably good methods for guarding problems" (2021-03810). B.J.N. and C.S. are supported by the Swedish Research Council project "New paradigms for autonomous unmanned air traffic management" (2018-04001).

‡Email: ar_bagheri@aut.ac.ir.
§Email: anna.brotzner@mau.se.
¶Email: f.farivar@srbiau.ac.ir.
‖Email: rahmat.ghasemi@srbiau.ac.ir.
**Email: f.keshavarz@shahed.ac.ir.
††Email: krohne@uwosh.edu.
‡‡Email: bengt.nilsson.TS@mau.se.
△Email: christiane.schmidt@liu.se.

Because all watchmen routes are a segment that the watchman walks back and forth, for the rest of this paper, we only compute that length of the segment and the routes have twice the length we compute.

Lemma 3 *If s and s' are starting points such that s dominates s', and w, w' are two watchmen routes starting at s and s', respectively, then w' will have zero length in an optimal minsum solution.*

Proof. Everything s' sees is seen by s. If a point p is unseen both from s and s', but the watchman starting at s' sees it, then the watchman has to cross either the horizontal or vertical line through s. W.l.o.g., the horizontal one, $h(s)$. But then the distance from s to a point that sees p is shorter than that from s'. □

Hence, let S be the set of non-dominated starting points. S admits a total order, so let the points be sorted from bottom-left to top-right: $s_1 < \cdots < s_m$. Define the x-overlap of two watchmen w and w' as the intersection between the projection of w and w' onto the base. Similarly, define the y-overlap as the intersection of the projections onto the wall.

Lemma 4 *Let W be a set of optimal watchmen routes and let $w, w' \in W$ be two watchmen routes with starting points s and s', respectively. If neither w nor w' has length zero, then w and w' have no x- and no y-overlap.*

Proof. Assume w.l.o.g. $s < s'$ and that the x-overlap of w and w' is non-empty. Let w and w' be disjoint (otherwise, we can shorten the routes). Let p be w's endpoint, and p' be (w')'s endpoint. Observe that $x(p) < x(p')$ as otherwise p would dominate the route w'. Let the overlap be the interval $[x_1, x_2]$, then $x(s') = x_1$. We can shorten w: Substitute w by $[s, (x(s'), y(p))]$. The vertical segment $[s', (x(s'), y(p))]$ is fully contained in P. Hence, no convex corner that w saw before is unseen by the new w and w'. By symmetry, it also holds for the y-overlap. □

Let P be a Stiegl polygon, and let C be the convex corners on the ceiling. Enumerate the corners in C from bottom-left to top-right by $c_1, \ldots, c_{\tilde{n}}$, where $\tilde{n} = \frac{n-2}{2}$. For a convex corner c_k let $h_k = h(c_k)$ be the extension of the horizontal edge at c_k, and $v_k = v(c_k)$ the extension of the vertical edge at c_k.

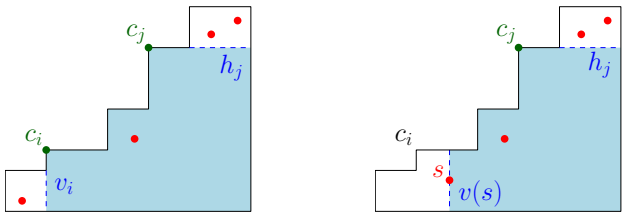
Lemma 5 *Let W be a set of optimal watchmen routes. Then, for every convex corner c_k that is not seen from S, either extension h_k or v_k is visited, no such extension is visited twice, and every watchman stops at an extension.*

Proof. First, we show that every watchman stops at an extension. Assume w.l.o.g. that watchman w crosses extension v_k in $v_k^\times$, and that this is the last extension on its route. Let q be the last point on its route. When walking along segment $[v_k^\times, q]$, w will not see any yet unseen convex corner that he did not see at $v_k^\times$. Hence, w can be replaced by $[s, v_k^\times]$, contradicting the assumption that it was optimal. Next, we argue that for each c_k that is not seen from S, either h_k or v_k is visited. Assume w.l.o.g. that watchman w visits v_k and stops there. Then, he can see all of the rectangle between c_k and the origin, but he will not see c_{k+1}. Let c_{k+1} be guarded by watchman w'. If w' moves to extension h_k, he will not see anything that w does not see yet. In order to see some convex corner c_j, $j < k$, that is not seen from w, w' has to walk downwards to a point below $v_k^\times$, but then w and w' have nonempty y-overlap, contradicting Lemma 4. Finally, no extension is visited twice since this would mean that two watchmen have nonempty x- or y-overlap, again contradicting Lemma 4. □

Let W be a set of watchmen that guard P optimally. Then the watchmen can be separated into solutions of subpolygons with bottom and right edges given by the extensions that are visited by the watchmen, and where the subpolygons are separated by *crates*, which are solely guarded by the starting points inside, and the watchmen outside, but no watchman moves inside these crates. A *crate* is a Stiegl polygon with precisely two convex corners on the ceiling that are not seen by the set of starting points in the crate. Specifically we define a crate by (a) two unseen convex corners c_i and c_j where every $c_k, i < k < j$, is seen, where we cut along the extensions v_i, h_j, or (b) one unseen convex vertex c_j and a starting point s where we cut along $v(s)$ and h_j, or v_j and $h(s)$ depending on the position of s, and s is considered to be outside the crate. In case there are two starting points s, s' with $x(c_i) < x(s) < x(s') < x(c_{i+1})$, we only consider the crate cut at $v(s)$, but not the crate cut at $v(s')$. Similarly, for two starting points s, s' with $y(c_{j-1}) < y(s) < y(s') < y(c_j)$, we only consider the crate cut at $h(s')$. We say such a crate starts at i and ends at j. (We do not define a crate if both cuts pass starting points as then the cuts are automatically visited.) The two unseen convex corners on the crate's ceiling are precisely those incident to these cuts. See Figure 1 for the different types of crates.

3 A dynamic programming algorithm

We describe an algorithm, which iteratively splits the polygon into two independent subpolygons, called *sub-Stiegl polygons*, that are separated by a crate and which computes the minimum length watchmen routes in each of them recursively. In each recursion,



(a) Crate cut at v_i and h_j

(b) Crate cut at $v(s)$ and h_j

Figure 1: The two different types of crates.

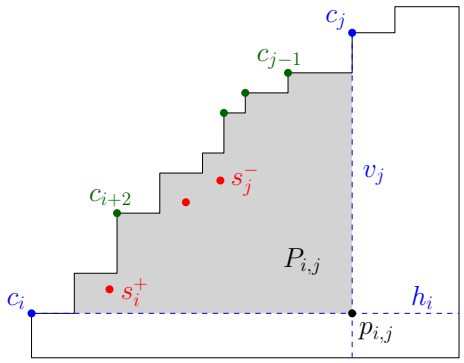


Figure 2: The sub-Stiegl polygon $P_{i,j}$

it is ensured that the neighboring crates are seen by forcing watchmen to walk to the base and the wall of the sub-Stiegl polygon.

3.1 Idea

After cutting out a crate, we are left with two sub-Stiegl polygons, which need to be guarded. For the lower sub-Stiegl polygon, we consider the minimum length of a watchman route guarding it immediately, and all possible crates that split the subpolygon. The upper one will be guarded immediately.

Since it is not necessary to walk to the bottom and the right boundary of the initial polygon P, some preprocessing is necessary. We do this by cutting off a horizontal strip along the lower boundary of P, and a vertical strip along the right boundary of P such that the interior of the strip is seen by a watchman that visits the extension along which we cut.

For defining the horizontal strip, consider the first convex corner c_1. If no starting point lies below h_1, this is an extension which needs to be visited in order to see c_1. Therefore we cut off the horizontal strip below h_1. If there is a starting point s below h_1, we consider $h(s)$ as the first extension, and cut off the horizontal strip below. Note that it does not matter whether we cut off the strip below the lowermost starting point or any other starting point as long as we do not cut off an unseen convex corner on the ceiling, because the extension will be visited from the starting point we choose with a watchman of length 0, and the shortest watchman route to a vertical extension will never start at any of the lower starting points as they also lie further to the left. Analogously, we cut off a vertical strip at the wall of P.

3.2 Sub-Stiegl polygons

In each iteration, the algorithm considers a subpolygon of P and computes the optimal solution within that.

Let $P_{i,j}$ be the Stiegl polygon that evolves when cutting off a crate along h_i, or along $h(s)$ for any $s \in S$ satisfying $y(c_{i-1}) < y(s) \leq y(c_i)$, and v_j, or $v(s')$ for any $s' \in S$ satisfying $x(c_j) < x(s') \leq x(c_{j+1})$. This definition is unique up to the choice of the starting point that defines the cut. Here, we simply choose the leftmost possible starting point for vertical cuts, and the uppermost possible starting point for horizontal cuts in order to remove a maximal crate. This will not change the solution in $P_{i,j}$ because among all possible starting points s satisfying $y(c_{i-1}) < y(s) \leq y(c_i)$, the uppermost one has the shortest direct path to the wall of $P_{i,j}$ among all starting points below h_i, and the same holds for any possible vertical cut.

Let $p_{i,j}$ be the origin of $P_{i,j}$, let $S_{i,j}$ be the subset of starting points in S that lie in $P_{i,j}$ (possibly on the boundary) and let $s_i^+ < \cdots < s_j^-$ be the points in $S_{i,j}$. Let furthermore $C_{i,j}$ be the subset of C that lies in $P_{i,j}$ and is not yet seen. See Figure 2 for an illustration. The goal is to visit both the floor and the wall of $P_{i,j}$ with watchmen routes that start at $S_{i,j}$, such that all corners in $C_{i,j}$ are seen.

Let $\mathcal{L}(i)$ be the length of the minimum watchmen routes in the subpolygon $P_{1,i}$, starting at the points $S_{1,i}$.

Lemma 6 *If every convex corner in a sub-Stiegl polygon $P_{i,j}$ is already seen by $S_{i,j}$, then the optimal watchmen routes inside $P_{i,j}$ consists either of one watchman starting at $s \in S_{i,j}$ who directly moves to $p_{i,j}$, or s_i^+ who moves vertically down to h_i and s_j^- who moves horizontally right to v_j.*

Proof. As all convex corners are already seen, there is no extension inside $P_{i,j}$ that needs to be visited by a watchman. Hence, any watchman will directly walk to h_i or v_j and stop there. Moreover, if a watchman walks towards only one of the extensions, but does not visit the other one, its shortest route will be the orthogonal onto the extension. For any such watchman route starting neither at s_i^+ nor at s_j^-, its route can be replaced by the parallel route starting at s_i^+ or s_j^-, respectively. □

We define the *uninorm* of a polygon $P_{i,j}$, denoted $\|P_{i,j}\|_u$, as the length of the shortest possible watchmen routes from which $P_{i,j}$ is guarded, using starting

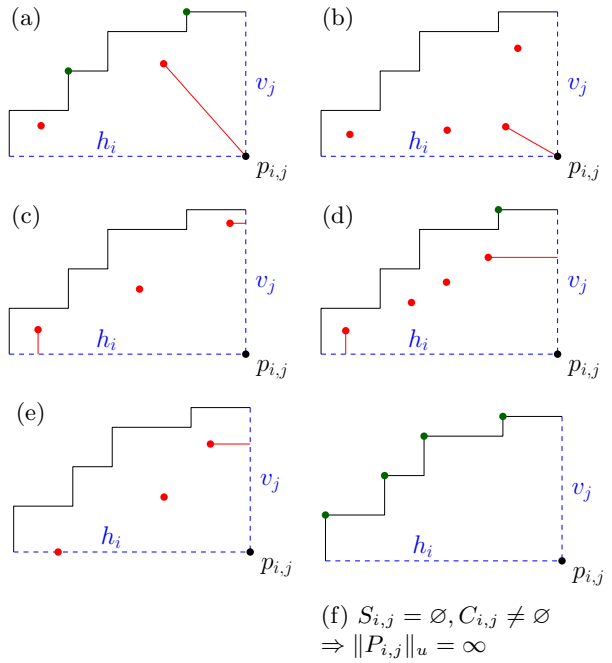


Figure 3: Different solutions for sub-Stiegl polygon $P_{i,j}$, depending on the starting points $S_{i,j}$ (red points). The floor and the right wall (blue dashed lines) need to be visited and all unseen convex corners $C_{i,j}$ (green) need to be guarded.

points in $S_{i,j}$, and that visit both the base and the wall of $P_{i,j}$, and such that the solution is not split into a set of independent solutions. The value of the uninorm depends on the unseen convex corners and the starting points in $S_{i,j}$ (the precise dependency is given by the $(*)$-condition that we define in the last paragraph of this section),

$$\|P_{i,j}\|_u = \begin{cases} \min\limits_{s \in S_{i,j}} \|s, p_{i,j}\| & \text{if } C_{i,j} \neq \varnothing, |S_{i,j}| \geq 1 \text{or } \neg(*) \\ \min\left\{ \|s_i^+, h_i\| + \|s_j^-, v_j\|, \min\limits_{s \in S_{i,j}} \|s, p_{i,j}\| \right\} & \text{if } |S_{i,j}| \geq 1 \text{ or } (*) \\ \infty & \text{if } C_{i,j} \neq \varnothing, S_{i,j} = \varnothing \text{ or } P_{i,j} \text{ degenerates.} \end{cases}$$

In case $S_{i,j} = \varnothing$ while $C_{i,j} \neq \varnothing$, then $P_{i,j}$ cannot be guarded from its interior. Hence, $\|P_{i,j}\|_u$ is defined to be ∞. See Figure 3f. If $P_{i,j}$ is a degenerate crate with no area, then again $\|P_{i,j}\|_u = \infty$.

If every unseen convex corner $c \in C_{i,j}$ satisfies $y(c) < y(s_i^+)$ or $x(c) > x(s_j^-)$ then we say that $P_{i,j}$ satisfies the $(*)$-condition. If $P_{i,j}$ satisfies $(*)$, a watchman starting at s_i^+ who moves vertically down to h_i and a second watchman starting at s_j^- who moves horizontally right to v_j is a candidate solution (see Figure 3c–3e). The other candidate solutions are given by a single watchman moving from a starting point in $S_{i,j}$ to the origin $p_{i,j}$ (see Figure 3a–3b). The uninorm is then the minimum over all candidate solutions.

3.3 The algorithm

The total length of the minimum watchmen routes is computed by the recursion

$$\mathcal{L}(j) = \min \begin{cases} \|P_{1,j}\|_u & \text{or} \\ \min\limits_{\substack{1<i<j-1 \\ i \text{ unseen or} \\ \exists s: x(c_{i-1}) < x(s) < x(c_i)}} \{\mathcal{L}(i) + \min\limits_{\substack{i<k<j \\ i,k \text{ define a crate}}} \|P_{k,j}\|_u\}, \end{cases}$$

where the current Stiegl polygon is either guarded immediately, using watchmen routes of length $\|P_{1,j}\|_u$, or split into two sub-Stiegl polygons where the upper one, $P_{k,j}$, is guarded immediately. We precompute the uninorm of all sub-Stiegl polygons in $O(n(n+m)\log^2 m)$ time (per subpolygon $P_{i,j}$, query the closest point to the origin in $O(\log^2 m)$ time using a dynamic closest point data structure [9]). To fill out the lookup-table position $\mathcal{L}(j)$, the dynamic programming algorithm considers all values $\mathcal{L}(i)$, $i < j$, and corresponding values $\|P_{k,j}\|_u$ with index $k > i$, where i and k define a crate, and computes their sum. There are less than j values for the start of the crate i, and at most $j - i$ ends of the crate that need to be verified since for every convex corner, we only consider the maximum possible crate. As every lookup takes constant time, we can compute each entry in $O(n \cdot \min\{m, n\})$ time. Hence, the algorithm takes $O(n^2 \cdot \min\{m, n\})$ time and $O(n \cdot \min\{m, n\})$ storage.

References

[1] Svante Carlsson, Bengt J. Nilsson, and Simeon C. Ntafos. Optimum guard covers and m-watchmen routes for restricted polygons. *Int. J. Comput. Geom. Appl.*, 3(1):85–105, 1993.

[2] B.J. Nilsson and D. Wood. Optimum watchmen routes in spiral polygons: Extended abstract.

[3] B.J. Nilsson and D. Wood. Watchmen routes in spiral polygons. Technical Report LU-CS-TR:90-55, Dept. of Computer Science, Lund University, 1990.

[4] Bengt J. Nilsson and Sven Schuierer. Shortest m-watchmen routes for histograms: the minmax case. In *Proc. 4th ICCI '92:*, pages 30–33, 1992.

[5] Joseph SB Mitchell and Erik L Wynters. Watchman routes for multiple guards. In *Proc. 3rd CCCG*, volume 9, pages 293–327, 1991.

[6] Bengt J. Nilsson and Eli Packer. An approximation algorithm for the two-watchman route in a simple polygon. In *EuroCG*, 2016.

[7] Eli Packer. Computing multiple watchman routes. In Catherine C. McGeoch, editor, *WEA*, pages 114–128. Springer Berlin Heidelberg, 2008.

[8] Mireille Bousquet-Melou, Anthony J Guttmann, William P. Orrick, and Andrew Rechnitzer. Inversion relations, reciprocity and polyominoes, 1999.

[9] Haim Kaplan, Wolfgang Mulzer, Liam Roditty, Paul Seiferth, and Micha Sharir. Dynamic Planar Voronoi Diagrams for General Distance Functions and Their Algorithmic Applications. *D&CG*, 64(3):838–904, 2020.

Special constructions to understand the structure of higher order Voronoi diagrams

Mercè Claverol[1], Andrea de las Heras-Parrilla[1], and Clemens Huemer[1]

[1]Universitat Politècnica de Catalunya

Abstract

We study the structure of faces in Voronoi diagrams of order k, $V_k(S)$, of sets S of n points in the plane. While the number of faces of $V_k(S)$ is well known, not so much is known about the numbers of quadrilaterals, of pentagons, of hexagons in $V_k(S)$. We present two extremal point sets and calculate the number of faces of each type in $V_k(S)$. Among the obtained results, we show that there exists a set S of n points, where all bounded faces of $V_k(S)$ are hexagons, for $k \geq (n+3)/2$, and where $V_k(S)$ contains no quadrilateral for $3 \leq k \leq (n+1)/4$. Finally, we prove that for no point set S, $V_k(S)$ can have two adjacent quadrilaterals, for $k \geq 2$, and we present some experimental result.

1 Introduction

We present a study on higher order Voronoi diagrams, that continues our previous work [3]. Voronoi diagrams are a very useful tool in diverse disciplines, see e.g. [1, 11]. Many of their properties were already obtained by Lee [7]. For a given set S of n points in general position in the plane, meaning that no three points of S are collinear and no four points of S are cocircular, the Voronoi diagram of order k of S, $V_k(S)$, is a subdivision of the plane into faces such that points in the same face have the same k nearest points of S. A face of $V_k(S)$ is denoted by $f(P_k)$ where P_k is the subset of k points of S that is closest to every point of this face. It is well known that $V_k(S)$ has $(2k-1)n - (k^2-1) - \sum_{j=0}^{k-2} e_j$ many faces, see e.g. [7, 3]. Here, e_j denotes the number of j-edges of S. A j-edge is a half-plane defined by the oriented line through a pair of points of S that contains j points of S in its interior. The set P_k associated to an unbounded face can be separated from $S \setminus P_k$ by a straight line, and the number of unbounded faces of $V_{k+1}(S)$ is e_k.

Miles and Maillardet [9] proved that $V_k(S)$ never contains a triangle for $k \geq 2$, also see [3, 8]. We are interested in the number of quadrilateral faces, of pentagonal faces, etc., of $V_k(S)$. This question has been studied extensively for $k = 1$ and for random point sets, especially with respect to a homogeneous Poisson point process, see e.g. [2, 4, 10, 6]. Several of these results are experimental and are summarized in [11]. In this setting, the expected number of sides of a face of $V_k(S)$ is 6 for every $1 \leq k \leq n-2$ [10]. In order to study how many faces with i sides, for $i = 4, 5, \ldots$, are there at least and at most in $V_k(S)$ among all sets S of n points, and to better understand the structure of $V_k(S)$, we present two special point sets S, determine subsets $P_k \subset S$ that define a face $f(P_k)$ of $V_k(S)$ and count the number of i-sided faces. For the first point set S, studied in Section 2, all its points are placed very close to the coordinate axes. Among the properties of $V_k(S)$ for this set, we point out that $V_k(S)$ contains no quadrilateral for $3 \leq k \leq \frac{n+1}{4}$, and, for $k \geq \frac{n+3}{2}$, all the bounded faces of $V_k(S)$ are hexagons. The second considered point set S consists of n points on the positive branch of the parabola $y = x^2$, i.e. on the two-dimensional moment curve. We describe all faces of $V_k(S)$ precisely. Interestingly, for every $2 \leq k \leq n-2$, $V_k(S)$ contains exactly one quadrilateral, and for $k \geq 3$, all hexagonal faces are alternating (this is defined in the following). A similiar study of counting the number of i-sided faces in a special point set was carried out for Voronoi diagrams of order 1 in [5], where the points are placed on the Archimedean spiral.

The i-sided faces of $V_k(S)$ can be classified even more precisely: each vertex of a face $f(P_k)$ is either the circumcenter of two points from P_k and one point from $S \setminus P_k$, a type II vertex, or of one point from P_k and two points from $S \setminus P_k$, a type I vertex [3]. Such vertices are also called inner and outer vertices [9], or old and new vertices [7]. It is known that in $V_k(S)$, for $2 \leq k \leq n-2$, every bounded face has vertices of type I and of type II [7]. For $k \geq 2$, every quadrilateral has two vertices of each type, which appear in alternating order. There exist two classes of pentagonal faces: Class I are pentagons with three vertices of type I and two vertices of type II, and Class II are pentagons with three vertices of type II and two vertices of type I. We say that a hexagonal face is *alternating* if its vertices alternate between type I and type II. See [3] for some structural results on alternating hexagons in $V_k(S)$. We then also study the number of faces according to this classification for type I and type

II vertices. We will need the *edge labeling* of $V_k(S)$, defined in [3]. An edge that delimits a face of $V_k(S)$ is a (possibly unbounded) segment of the perpendicular bisector of two points i and j of S. This well-known observation induces a natural labeling of the edges of $V_k(S)$ with the following rules:
Edge rule: An edge of $V_k(S)$ from the perpendicular bisector of points $i, j \in S$ has labels i and j, where label i is on the side (half-plane) of the edge that contains point i and label j is on the other side.
Vertex rule: Let v be a vertex of $V_k(S)$ and let $\{i, j, \ell\} \in S$ be the set of labels of the edges incident to v. The cyclic order of the labels of the edges around v is i, i, j, j, ℓ, ℓ if v is of type I, and it is i, j, ℓ, i, j, ℓ if v is of type II.
Face rule: In each face of $V_k(S)$, the edges that have the same label i are consecutive, and these labels i are either all in the interior of the face, or are all in the exterior of the face.

Using the edge labeling, we prove a structural result that holds for every set of points S, namely that no two quadrilaterals can share an edge in $V_k(S)$, for $k \geq 2$. We also describe the labels of the edges of $V_k(S)$ for the point set on the parabola, studied in Section 3. Proofs are omitted in this abstract.

2 Points close to the axes

Let $S = H \cup V$ where H are all the points of the form $H_i = (i, 0)$ with $i \in \mathbb{Z}, -n \leq i \leq n, n \geq 1$, and V are all of the form $V_j = (0, j)$ with $j \in \mathbb{Z}, -(n+m) \leq j \leq -n, m > 1$, or $n \leq j \leq n+m$. H_n, H_{-n} are called extremes of H, and $V_n, V_{-n}, V_{n+m}, H_{-n-m}$ are extremes of V. We slightly perturb the points of H and V so that the points of S are in general position. The structure of Voronoi diagrams stays the same when the perturbation of the points is sufficiently small; values of k where this perturbation can make a difference are not considered. Note that $|S| = |H| + |V| = (2n+1) + 2m + 2$.

Lemma 1. *Every circle C passing through the points H_i and $H_{i'}$, where $i, i' \in \mathbb{Z}$, encloses all points H_p, with $-n \leq i < p < i' \leq n$. If in addition C passes through V_j, $n \leq j$, then C encloses the points V_ℓ such that $n \leq \ell < j$. Analogously if $j \leq -n$, then C encloses the points V_ℓ such that $j < \ell \leq -n$.*

Lemma 2. *Let C be a circle passing through $H_i \in H$, V_j and $V_{j'} \in H$, $j, j' \in \mathbb{Z}$, where $n \leq j < j'$ or $j' < j \leq -n$. Then, if $i > 0$, C encloses H_p with $i < p$; if $i < 0$, C encloses H_p with $p < i$.*

2.1 Quadrilaterals

Property 3. *$V_1(S)$ has $|H| + |V| - 6 = 2(n+m) - 3$ quadrilateral faces. Also, if the points of S are on the coordinate axes, two edges of each quadrilateral are tangent to the parabolas with focus H_n, H_{-n}, V_n, V_{-n} and directrix an axis.*

To illustrate Property 3, see Figure 1.

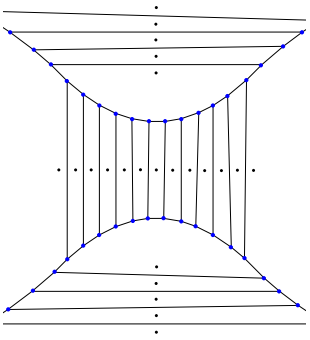


Figure 1: All bounded faces of $V_1(S)$ are quadrilaterals except two of them which have $|H| + 2$ sides.

Property 4. *$V_2(S)$ has four quadrilateral faces: $f(\{V_n, H_n\})$, $f(\{V_n, H_{-n}\})$, $f(\{V_{-n}, H_n\})$ and $f(\{V_{-n}, H_{-n}\})$. Moreover, $V_k(S)$ does not have quadrilateral faces for $3 \leq k \leq |V|/2$ and $k \geq |H| + 2$.*

2.2 Pentagons

It is possible to find a collection of pentagons joined two by two, sharing an edge. We find this configuration in the $V_k(S)$, where $2 \leq k \leq |V|/2 = m + 1$ (if $m = n$, then $2 \leq k \leq (|S| + 1)/4$). See Figure 2.

Property 5. *In each $V_k(S)$, $2 \leq k \leq |V|/2$, there are two chains of pentagons. Further, if P_k is a set of points associated to a pentagonal face of $V_k(S)$, then P_k has either a single point from V and an extreme point of H, or a single point from H and an extreme point of V, except in the case where $k = 2$, in which the two points of P_2 cannot be one of them extreme of V and the other one extreme of H. The number of pentagonal faces is $2(|V| + |H|) - 12$ in $V_2(S)$ and $2(|V| + |H|) - 4$ in $V_k(S)$, for $k \geq 3$.*

2.3 Hexagons

Property 6. *Let $f(P_k)$ be a non-alternating hexagonal face of $V_k(S)$. Then, P_k is either:*

- *A set of k consecutive points of $H \setminus \{H_{-n}, H_n\}$ where $2 \leq k \leq |H| - 2$.*
- *A set of k consecutive points of $V \setminus \{V_{-n}, V_n, V_{-(n+m)}, V_{n+m}\}$ where $2 \leq k \leq |V|/2 - 2$.*
- *A set of k_1 consecutive points of H that contains either H_{-n} or H_n, together with k_2 consecutive points of $V \setminus \{V_{-(n+m)}, V_{n+m}\}$ that contain either*

V_{-n} or V_n, where $k = k_1 + k_2 \geq 4$, $2 \leq k_1 < |H| - 1$, $2 \leq k_2 < (|V|/2) - 1$.

Property 7. *Let $f(P_k)$ be an alternating hexagonal face of $V_k(S)$. Then P_k is either:*

- *A set of k_1 contiguous points of $H \setminus \{H_{-n}, H_n\}$ together with $k_2 = k - k_1$ contiguous points of $V \setminus \{V_{-(n+m)}, V_{n+m}\}$ that contain V_{-n} or V_n where $k \geq 3$, $2 \leq k_1 < |H| - 2$, $k_2 < (|V|/2) - 1$.*
- *A set of k_2 contiguous points of $V \setminus \{V_{-n}, V_n, V_{-(n+m)}, V_{n+m}\}$ together with $k_1 = k - k_2$ contiguous points of H that contain H_{-n} or H_n, where $2 \leq k_2 < |V|/2 - 2$, $k_1 < |H|$.*

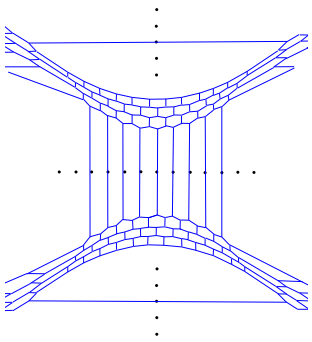


Figure 2: Hexagonal and pentagonal faces in $V_4(S)$.

Property 8. *The numbers of hexagons in $V_k(S)$, for $2 \leq k \leq \min\{|H| - 2, |V|/2 - 2\}$ are:*

$k = 2$	$2n + 2m - 6$
$k = 3$	$6n + 6m - 21$
$k \geq 4$	$(\|S\| - 3)(2k - 3) - 3k^2 + 4k - 6$

Property 9. *If $|V| \leq |H|$ and $|H| + 2 \leq k < |S| - 1$, then all bounded faces of $V_k(S)$ are hexagons. Moreover, if the set P_k associated to the bounded face $f(P_k)$ of $V_k(S)$ does not contains an extreme point of H, then $f(P_k)$ is an alternating hexagon.*

3 Points on the positive branch of a parabola

Let S be the ordered set of points of the form $Q_i = (x_i, x_i^2)$, where $x_i \in \mathbb{R}$, $x_i > 0$, $i \in \mathbb{N}$, $1 \leq i \leq n$ and $Q_i < Q_j$ if and only if $i < j$ and $x_i < x_j$. We count the bounded faces $V_k(S)$, which can only be a quadrilateral, pentagons and alternating hexagons.

Lemma 10. *Every circle C passing through the points Q_i, Q_j and Q_ℓ, with $i < j < \ell$, encloses all points Q_m with $m < i$ or $j < m < \ell$.*

3.1 Quadrilaterals

Property 11. *$V_k(S)$ with $2 \leq k \leq n - 2$ has a unique quadrilateral face $f(P_k)$. The two labels at the interior of $f(P_k)$ are $k - 1$ and $k + 1$ with $Q_{k-1}, Q_{k+1} \in P_k$ and the two labels at the exterior of $f(P_k)$ are k and $k + 2$ with $Q_k, Q_{k+2} \notin P_k$. Also, $P_k = \{Q_1, Q_2, ..., Q_{k-2}, Q_{k-1}, Q_{k+1}\}$.*

3.2 Pentagons

There exists two classes of pentagonal faces with both types of vertices: Class I are pentagons with three vertices of type I and two vertices of type II and Class II are pentagons with three vertices of type II and two vertices of type I.

Property 12. *Let $f(P_k)$ be a class I pentagonal face of $V_k(S)$ with $2 \leq k \leq n - 2$, and let i and j be the two labels at the interior of $f(P_k)$ with $i < j$ and $Q_i, Q_j \in P_k$. Then, $i = k - 1$, $k + 2 \leq j \leq n - 1$ and the three labels at the exterior of $f(P_k)$ are k, $j - 1$ and $j + 1$, with $Q_k, Q_{j-1}, Q_{j+1} \notin P_k$. Also, $P_k = \{Q_1, Q_2, ..., Q_{k-2}, Q_{k-1}, Q_j\}$.*

Property 13. *Let $f(P_k)$ be a class II pentagonal face of $V_k(S)$ with $3 \leq k \leq n - 3$, and let i, j, ℓ be the three labels at the interior of $f(P_k)$ with $i < j < \ell$ and $Q_i, Q_j, Q_\ell \in P_k$. Then, $1 \leq i \leq k - 2$, $j = i + 2$, $\ell = k + 1$ and the three labels at the exterior of $f(P_k)$ are $i + 1$ and $k + 2$, with $Q_{i+1}, Q_{k+2} \notin P_k$. Also, the points Q_m with $m < i$ or $i + 2 < m < k + 1$ are the remaining points of P_k.*

Property 14. *$V_k(S)$ with $2 \leq k \leq n - 2$, has exactly $(n - k - 2)$ class I pentagonal faces.*

Property 15. *$V_k(S)$ with $3 \leq k \leq n - 3$, has exactly $(k - 2)$ class II pentagonal faces.*

3.3 Hexagons

Property 16. *Let $f(P_k)$ be an alternating hexagonal face of $V_k(S)$ with $3 \leq k \leq n - 3$, and let i, j and ℓ be the three labels at the interior of $f(P_k)$ with $i < j < \ell$ and $Q_i, Q_j, Q_\ell \in P_k$. Then, $1 \leq i \leq k - 2$, $i + 2 \leq j \leq n - k + i$, $\ell \leq n - 1$ and the three labels at the exterior of $f(P_k)$ are $i + 1$, $j - 1$ and $\ell + 1$ with $Q_{i+1}, Q_{j-1}, Q_{\ell+1} \notin P_k$. Also, the points Q_m with $m < i$ or $j < m < \ell$ are the remaining points of P_k.*

Property 17. *$V_k(S)$ with $3 \leq k \leq n - 3$, has exactly $(k - 2)(n - k - 2)$ alternating hexagons.*

4 Experimental and theoretical results

Previous properties have been additionally verified computationally. For this, a generator algorithm for the order-k Voronoi diagram was implemented in Python, so n-sided bounded faces can be counted. This code was used to seek for more general properties. We generated 1000 sets of n uniformly distributed random points on the unit square in general position for each n from 4 to 20. We obtained all order-k Voronoi

diagrams for these sets. Then, minimum, maximum and mean of the n-sided bounded faces for all of the same order Voronoi diagrams for the sets with the same number of points were computed. We get tables like the ones below shown for $n = 10$.

$n = 10$	Quadrilateral	Pentagons	Hexagons
$k = 1$	min= 0 max= 5 mean= 1.379	min= 0 max= 5 mean= 1.464	min= 0 max= 3 mean= 0.684
$k = 2$	min= 0 max= 6 mean= 3.077	min= 0 max= 12 mean= 4.177	min= 0 max= 10 mean= 2.988
$k = 3$	min= 0 max= 8 mean= 3.957	min= 0 max= 12 mean= 4.644	min= 0 max= 11 mean= 4.16
$k = 4$	min= 0 max= 9 mean= 4.092	min= 0 max= 11 mean= 4.719	min= 0 max= 13 mean= 4.342
$k = 5$	min= 0 max= 10 mean= 3.726	min= 0 max= 12 mean= 4.249	min= 0 max= 11 mean= 3.908
$k = 6$	min= 0 max= 7 mean= 3.007	min= 0 max= 10 mean= 3.432	min= 0 max= 9 mean= 3.017
$k = 7$	min= 0 max= 5 mean= 2.048	min= 0 max= 8 mean= 2.328	min= 0 max= 7 mean= 1.866
$k = 8$	min= 0 max= 3 mean= 0.978	min= 0 max= 5 mean= 1.190	min= 0 max= 3 mean= 0.615

$n = 10$	Class I Pentagons	Class II Pentagons	Alternating Hexagons
$k = 1$	min= 0 max= 0 mean= 0	min= 0 max= 0 mean= 0	min= 0 max= 0 mean= 0
$k = 2$	min= 0 max= 12 mean= 4.177	min= 0 max= 0 mean= 0	min= 0 max= 0 mean= 0
$k = 3$	min= 0 max= 9 mean= 3.335	min= 0 max= 6 mean= 1.309	min= 0 max= 7 mean= 1.081
$k = 4$	min= 0 max= 7 mean= 2.678	min= 0 max= 7 mean= 2.678	min= 0 max= 8 mean= 1.304
$k = 5$	min= 0 max= 6 mean= 1.983	min= 0 max= 7 mean= 2.266	min= 0 max= 7 mean= 1.304
$k = 6$	min= 0 max= 5 mean= 1.241	min= 0 max= 7 mean= 2.190	min= 0 max= 7 mean= 1.216
$k = 7$	min= 0 max= 3 mean= 0.534	min= 0 max= 6 mean= 1.793	min= 0 max= 4 mean= 0.489
$k = 8$	min= 0 max= 0 mean= 0	min= 0 max= 5 mean= 1.190	min= 0 max= 0 mean= 0

Note that, since for $k = n - 1$ the Voronoi diagram $V_k(S)$ has no bounded faces, there is no row in the tables for $k = 9$ as all the values are always 0.

With these tables we try to find general properties for the number of quadrilaterals, pentagons, and hexagons in higher order Voronoi diagrams. We proved the next results for the bounded faces of the Voronoi diagrams of any set of points in general position.

Property 18. *Only in Voronoi diagrams of order one, it is possible to find two quadrilaterals sharing an edge.*

Property 19. *$V_k(S)$ with $k \geq 2$, cannot have a bounded face with only two type II vertices and sharing a type I vertex with two Class II pentagonal faces.*

Acknowledgments. M. C., A. d.H.-P. and C. H. were supported by project PID2019-104129GB-I00/MCIN/AEI/10.13039/501100011033 and Gen.Cat. 2021 SGR 00266.

References

[1] F. Aurenhammer. *Voronoi diagrams - a survey of a fundamental geometric data structure*, volume 23. ACM Computing Surveys, 1991.

[2] M. W. Bern, D. Eppstein, and F. F. Yao. The expected extremes in a Delaunay triangulation. *Int. J. Comput. Geom. Appl.*, 1(1):79–91, 1991.

[3] M. Claverol, A. de las Heras, C. Huemer, and A. Martínez-Moraian. The edge labeling of higher order Voronoi diagrams. *Proc. of Spanish meeting on Computational Geometry 2021*, pages 23–26. `https://arxiv.org/abs/2109.13002`.

[4] I. K. Crain. The monte-carlo generation of random polygons. *Computers & Geosciences*, 4(2):131–141, 1978.

[5] M. Frenkel, I. Legchenkova, N. Shvalb, S. Shoval, and E. Bormashenko. Voronoi diagrams generated by the archimedes spiral: Fibonacci numbers, chirality and aesthetic appeal. *Symmetry*, 15(3):746, 2023.

[6] A. Hinde and R. Miles. Monte carlo estimates of the distributions of the random polygons of the Voronoi tessellation with respect to a Poisson process. *Journal of Statistical Computation and Simulation*, 10(3-4):205–223, 1980.

[7] D. T. Lee. On k-nearest neighbor Voronoi diagrams in the plane. *IEEE Trans. Comput.*, pages 478–487, 1982.

[8] J. E. Martínez-Legaz, V. Roshchina, and M. I. Todorov. On the structure of higher order Voronoi cells. *J. Optim. Theory Appl.*, 183(1):24–49, 2019.

[9] R. Miles and R. Maillardet. The basic structures of Voronoi and generalized Voronoi polygons. *Journal of Applied Probability*, 19(A):97–111, 1982.

[10] R. E. Miles. On the homogeneous planar Poisson point process. *Mathematical Biosciences*, 6:85–127, 1978.

[11] A. Okabe, B. Boots, K. Sugihara, and S. Chiu. *Spatial Tessellations: Concepts and Applications of Voronoi diagrams*. Wiley, 2 edition, 2000.

VC dimension with half guards

Erik Krohn*[1] and Alex Pahlow†[2]

[1]University of Wisconsin-Oshkosh
[2]University of Wisconsin-Madison

Abstract

In this paper, we show that the VC dimension of half-guarding a terrain is exactly 2 or 3, depending on certain assumptions. We also show that the VC dimension of half-guarding a monotone polygon is exactly 4.

1 Introduction

A full guard is a guard that can see 360°. In our paper, we define a *half guard* as a guard that sees 180° and only sees to the right. VC dimension is a measure of the complexity of some set system. It has been studied by researchers for many variants of the art gallery problem. Guarding simple polygons with full guards has a VC dimension between 6 and 14 [1]. Guarding monotone polygons (simple polygons) with full guards where guards are limited to being on the boundary of the polygon was shown to be exactly 6 in both types of polygons [2, 3]. The structure half guards add to the art gallery problem is interesting because the difference, as compared to, full guards, is not trivial. For example, convex polygons have a VC dimension of 1 with half guards despite having a VC dimension of 0 with full guards. Monotone polygons, where all guards and viewpoints are located on the boundary, have a VC dimension of 4 with half guards despite having a VC dimension of 6 with full guards.

A set of guards G in P is *shattered* if for every $G_s \subseteq G$, there exists a point that is seen by the guards in G_s and by no guards in $G \setminus G_s$. With half guarding, we show that the VC dimension is exactly 4. The terrain guarding problem with full guards has a VC dimension of exactly 4 [4]. With half guarding, we show that the VC dimension is exactly 2 or 3, depending on certain assumptions.

Notation: Let $p < q$ mean that point p is to the left of q, i.e. the x coordinate of $p.x < q.x$. With half guarding a polygon (resp. terrain), a point p sees a point q if the line segment connecting p and q does not go outside of the polygon (resp. below the terrain) and $p.x \leq q.x$. Let p and q be two points such that $p.x < q.x$, then $[p, q)$ denotes every point in the polygon between p and q (including the vertical line containing p but excluding the vertical line containing q). Let l be the leftmost point of the polygon and let r be the rightmost point of the polygon. The *ceiling (resp. floor)* denotes every boundary point in $[l, r]$ as we travel clockwise (resp. counterclockwise) from l to r. We define *viewpoint* as a point that is exactly seen by a subset of the guards. For example, the viewpoint $vp(AC)$ is a point in the polygon that is seen by guards A and C but is not seen by any other guards.

*Email: krohne@uwosh.edu
†Email: apahlow22@alumni.uwosh.edu

2 VC dimension of terrains

We start by discussing the VC dimension of terrains with regards to half guards. The VC dimension of a terrain with regards to half guards depends on if a point on the terrain can be considered both a guard and a viewpoint. If guards and viewpoints must be disjoint, then the VC dimension is 2. If a point on the terrain can be both a guard and a viewpoint, then the VC dimension is 3. Figure 1 shows an example of a terrain being shattered with 2 guards. We use the standard *order claim* without proof.

Claim: Let A, B, C, D be 4 points on a terrain with $A.x < B.x < C.x < D.x$. If A sees C and B sees D, then A must see D.

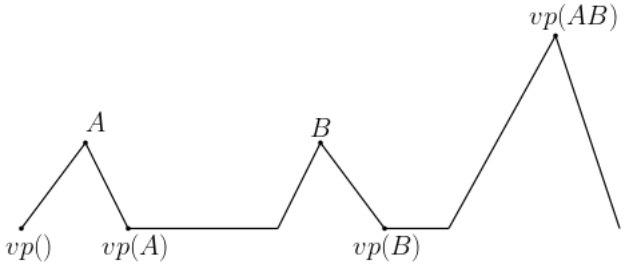


Figure 1: A terrain shattered by 2 half guards.

Theorem 1 *If a terrain guarding problem does (resp. does not) allow a guard and a viewpoint to be the same point, then the VC dimension of a terrain is exactly 3 (resp. 2).*

Proof. We will first consider the case where guards and viewpoints cannot be located at the same point. Let A, B and C be guards such that $A.x < B.x < C.x$. Assuming that a guard and viewpoint cannot be the same point, the viewpoints that are seen by C must be strictly to the right of C. It follows that $B.x < C.x < vp(BC).x$ and $B.x < C.x < vp(AC).x$. If $vp(BC).x < vp(AC).x$, then we have $B.x < C.x < vp(BC).x < vp(AC).x$. By the order claim, B sees $vp(AC)$, a contradiction. If $vp(AC).x < vp(BC).x$, then $A.x < B.x < vp(AC).x < vp(BC).x$. By the order claim, A sees $vp(BC)$, a contradiction.

Next we consider the VC dimension of terrains where a guard and a viewpoint can be at the same point. In this case, the VC dimension is 3. We achieve a lower bound of 3 by giving an example of a terrain shattering 3 guards in Figure 2.

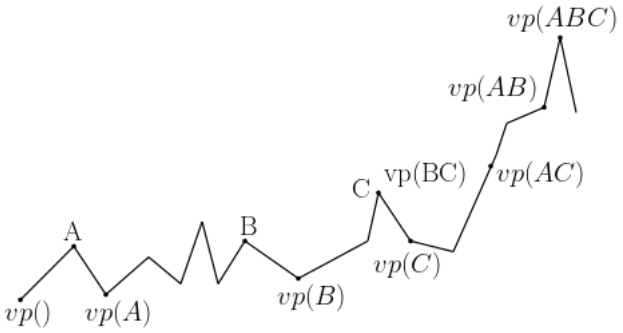


Figure 2: A terrain shattered by 3 half guards. In this example, C and $vp(BC)$ are the same point.

We will show that it is impossible for such a terrain to have a VC dimension of 4. Let A, B, C, D be the guards of this polygon with $A.x \leq B.x \leq C.x \leq D.x$. Consider the following cases:

1. If the viewpoint $vp(AC).x < vp(BD)$, then $A.x < B.x < vp(AC).x < vp(BD).x$. By the order claim using $A, B, vp(AC), vp(BD)$, A sees $vp(BD)$.

2. If the viewpoint $vp(BD).x < vp(AC)$, then $B.x < C.x < vp(BD).x < vp(AC).x$. By the order claim using $B, C, vp(BD), vp(AC)$, B sees $vp(AC)$.

□

3 VC dimension of monotone polygons

We show that the VC dimension of half guarding a monotone polygon is exactly 4. We obtain the lower bound for monotone polygons by giving an example of a monotone polygon being shattered by 4 guards as seen in Figure 3. We now show that the 5 guards cannot be shattered with a case analysis. A few cases are shown in the paper with the remaining ones omitted due to lack of space. We use the following lemma:

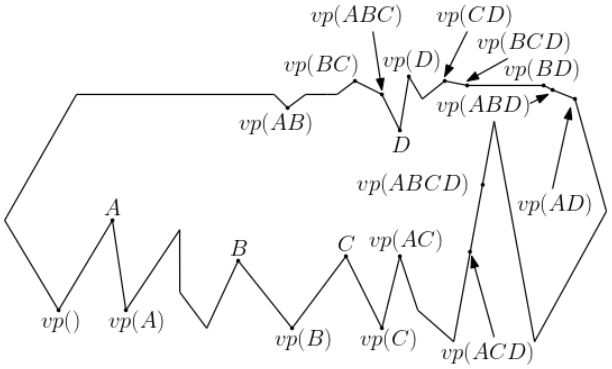


Figure 3: Polygon shattered by 4 half guards.

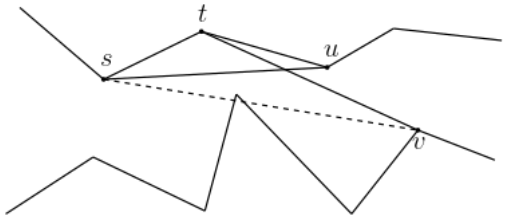


Figure 4: Lemma 2 where s, t and u are on the ceiling.

Lemma 2 *Let $s < t < u < v$ where s, t, u are on the same side of the polygon, s sees u, t sees v, and s does not see v. The opposite side of the polygon must block s from seeing v.*

Proof. W.l.o.g., assume s, t and u are on the ceiling. If a point p' on the ceiling is used to block s from v such that $s.x < p'.x < u.x$, then s is blocked from u. If a point p' on the ceiling is used to block s from v such that $t.x < p'.x < v.x$, then t is blocked from v. If the ceiling wraps underneath v to block s from v, then the polygon is not monotone. Therefore, if s does not see v, the floor must block it. □

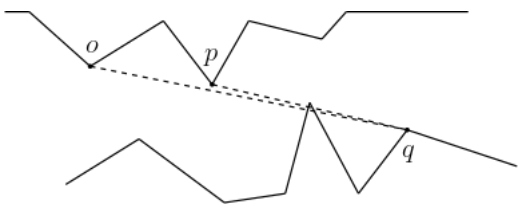


Figure 5: Lemma 3 where p is on the ceiling and the floor blocks p from q.

Lemma 3 *Let p and q be two points in the polygon and let $p < q$. If p is blocked from q using the side opposite p, then no point in $[l, p]$ can see q.*

Proof. W.l.o.g, assume that p is on the ceiling and the floor is blocking p from q. Let o be some point to the left of p. The $\overrightarrow{oq}$ ray lies in between the $\overrightarrow{pq}$ ray

and the floor. If this were not the case, then p would have blocked o from q. If the floor blocks p from q, the $\overrightarrow{oq}$ ray must also go through the floor and therefore, q must also be blocked from o. □

Lemma 4 *Let $s < t < u$, where t and u are on opposite sides of the polygon, s sees u, and t does not see u. It must be that t cannot see any point in $[u, r]$.*

Proof. Assume, w.l.o.g., that t is on the floor. Note that t cannot be blocked from u using the ceiling since by Lemma 3, s would not see u. Thus, t must be blocked from u using the floor. Let v denote some point to the right of u. If the $\overrightarrow{tv}$ line crosses above u, then the ceiling will block t from v. If the $\overrightarrow{tv}$ line crosses below u, then the floor will block t from v since the floor is blocking t from u. □

Corollary 4.1 *Let $t < u$, t is on the floor (resp. ceiling), u is on the ceiling (resp. floor), and the floor is blocking t from seeing u. It must be that t cannot see any point in $[u, r]$.*

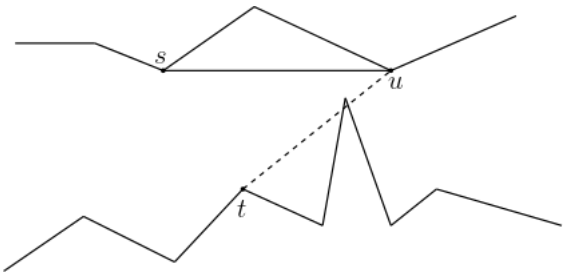


Figure 6: Visualization of Lemma 4.

We obtain an upper bound of 4 by showing that it is impossible to shatter 5 half guards in a monotone polygon. The upper bound proof is obtained by breaking the problem up into different cases. Unfortunately, every viewpoint, when considered by itself without placing any other viewpoints, can be placed when there are 5 guards. However, depending on the location of the guards, certain viewpoint combinations are impossible. We provide a few cases below. Consider a monotone polygon with 5 guards: $\{A, B, C, D, E\}$ such that $A.x \leq B.x \leq C.x \leq D.x \leq E.x$.

Case 1: Let $\{A, C\}$ be on the floor (resp. ceiling) and $\{B, D\}$ be on the opposite side. The position of E does not matter (with respect to the ceiling or floor). We show that it is impossible to place the points $vp(BCE)$ and $vp(ADE)$. Note that $vp(BCE)$ and $vp(ADE)$ must be to the right of, or on the same vertical line, as E.

Case 1a: If $vp(BCE)$ is on the ceiling to the left of $vp(ADE)$, or on same line as $vp(ADE)$, then consider how B must be blocked from $vp(ADE)$. The B guard cannot be blocked from $vp(ADE)$ using the ceiling because of Lemma 2 where $s = B, t = D, u = vp(BCE)$ and $v = vp(ADE)$. The floor must then be used to block B from $vp(ADE)$. By Lemma 3, using $o = A, p = B, q = vp(ADE)$, the A guard would not be able to see $vp(ADE)$. Therefore, B cannot be blocked from $vp(ADE)$. This case is illustrated in Figure 7.

Case 1b: If $vp(ADE)$ is on the ceiling to the left of $vp(BCE)$, or on same line, then consider how C is blocked from seeing $vp(ADE)$. This case is illustrated in Figure 8. Similar to the previous argument, if C is blocked from seeing $vp(ADE)$ using the floor, then by Lemma 4 using $s = A, t = C, u = vp(ADE), v = vp(BCE)$, C cannot see $vp(BCE)$. If the ceiling blocks C from seeing $vp(ADE)$, then by Lemma 3 using $o = A, p = C, q = vp(ADE)$, A is blocked from seeing $vp(ADE)$.

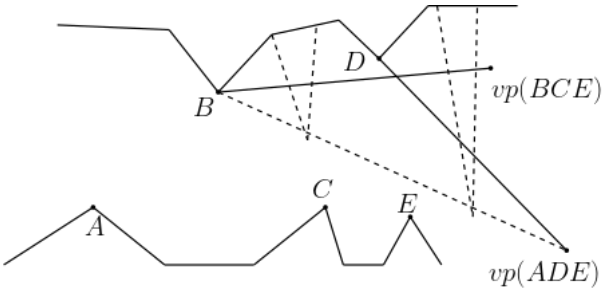


Figure 7: Visualization of Case 1a.

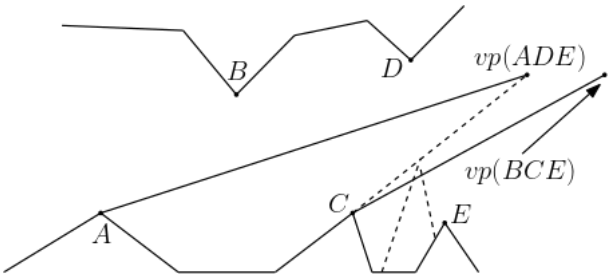


Figure 8: Visualization of Case 1b.

Case 1c: If $vp(BCE)$ is on floor to the left of $vp(ADE)$, or on same line as $vp(ADE)$, then consider how D must be blocked from $vp(BCE)$. If the ceiling blocks D from seeing $vp(BCE)$, then D does not see $vp(ADE)$ by Corollary 4.1 when $t = D, u = vp(BCE), v = vp(ADE)$. If the floor blocks D from seeing $vp(BCE)$, then by Lemma 3 with $o = C, p = D, q = vp(BCE)$, C cannot see $vp(BCE)$.

Case 1d: If $vp(ADE)$ is on floor to the left of $vp(BCE)$, or on same line as $vp(ADE)$, then consider how B is blocked from $vp(ADE)$. If the floor blocks B from $vp(ADE)$, then by Lemma 3 with $o = A, p = B, q = vp(ADE)$, A does not see $vp(ADE)$. If the ceiling blocks B from $vp(ADE)$, then by Corollary 4.1 with $t = B, u = vp(ADE), v = vp(BCE)$, B does not

see $vp(BCE)$.

Therefore, $\{A, C\}$ and $\{B, D\}$ cannot be on opposite sides of the polygon. We provide 1 more case.

Case 2: In this case, $\{A, E\}$ are on the floor (resp. ceiling) and $\{B, C, D\}$ are on the opposite side. In this case, it is impossible to place both $vp(BDE)$ and $vp(ACD)$.

Case 2a: The viewpoint $vp(ACD)$ is on the ceiling to the left of $vp(BDE)$ or on same line as $vp(BDE)$. We consider how C is blocked from $vp(BDE)$. We can't block C from $vp(BDE)$ with the ceiling by Lemma 2, where $s = C, t = D, u = vp(ACD), v = vp(BDE)$. If we try to block C from $vp(BDE)$ using the floor, we end up blocking B from $vp(BDE)$ by Lemma 3 with $o = B, p = C, q = vp(BDE)$).

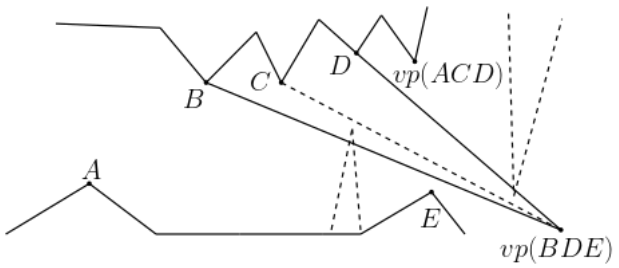


Figure 9: Visualization of Case 2a.

Case 2b: The viewpoint $vp(BDE)$ is on the ceiling to the left of $vp(ACD)$, or on same line as $vp(ACD)$. By Lemma 2 with $s = B, t = C, u = vp(BDE), v = vp(ACD)$, we must use the floor to block B from $vp(ACD)$. However, if we use the floor to block B from $vp(ACD)$, then by Lemma 3 with $o = A, p = B, q = vp(ACD)$, the A guard is blocked from seeing $vp(ACD)$.

Case 2c: The viewpoint $vp(ACD)$ on floor to the left of $vp(BDE)$. In this case, we consider how B is blocked from $vp(ACD)$. If the ceiling blocks B from $vp(ACD)$, then by Corollary 4.1 with $t = B, u = vp(ACD), v = vp(BDE)$, B does not see $vp(BDE)$. If the floor blocks B from $vp(ACD)$, then by Lemma 3 with $o = A, p = B, q = vp(ACD)$, the A guard does not see $vp(ACD)$.

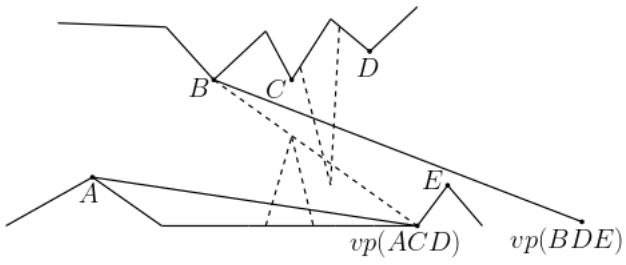


Figure 10: Visualization of Case 2c.

Case 2d: The viewpoint $vp(BDE)$ is on floor to the left of $vp(ACD)$. In this case, consider how C is blocked from seeing $vp(BDE)$. If the floor blocks C from seeing $vp(BDE)$, then by Lemma 3 with $o = B, p = C, q = vp(BDE)$, B would not see $vp(BDE)$. If the ceiling blocks C from $vp(BDE)$, then by Corollary 4.1 with $t = C, u = vp(BDE), v = vp(ACD)$, C would not see $vp(ACD)$.

These cases are just a few examples of how to show the VC dimension of a monotone polygon with half guards is exactly 4. The $2^5 = 32$ cases that we consider are the following: $\{A, C\}$ are on the same side and $\{B, D\}$ are on the opposite side (4 cases), $\{A, E\}$ are on some side and $\{B, C, D\}$ are on the opposite side (2 cases), $\{C, E\}$ are on the same side and $\{A, B, D\}$ are on the opposite side (2 cases), there are any 4 guards that are on the same side (12 cases), $\{A, B\}$ are on the same side and $\{C, D\}$ are on the opposite side (4 cases), $\{A, D\}$ are on the same side and $\{B, C\}$ are on the opposite side (4 cases), $\{B, E\}$ are on the same side and $\{A, C, D\}$ are on the opposite side (2 cases), and $\{A, B, C\}$ are on the same side and $\{D, E\}$ are on the opposite side (2 cases).

These cases give us the following theorem.

Theorem 5 *The VC dimension of half guarding a monotone polygon is exactly 4.*

4 Conclusions

We show the VC dimension exactly for several variants of half guarding in the art gallery problem. The VC dimension for half guarding a terrain is 2 or 3 depending on the assumption of whether or not guards and viewpoints can occupy the same space. The VC dimension for monotone polygons with half guards is exactly 4.

Open problem 6 *What is the VC dimension of half-guarding other variants of the art gallery problem, for example: simple polygons, spiral polygons, orthogonal polygons, etc?*

References

[1] Alexander Gilbers and Rolf Klein. A new upper bound for the VC-dimension of visibility regions, *Comput. Geom.* **47** (2014), 61–74.

[2] Gibson, Matt, Krohn, Erik and Wang, Qing. The VC-dimension of visibility on the boundary of monotone polygons, *Comput. Geom.* **77** (2019), 62–72.

[3] Gibson, Matt, Krohn, Erik and Wang, Qing. The VC-Dimension of Visibility on the Boundary of a Simple Polygon, *ISAAC* **9472** (2015), 541–551.

[4] James King. VC-Dimension of Visibility on Terrains, *Proceedings of the 20th Annual Canadian Conference on Computational Geometry* (2008)

Comparing box and disk bichromatic discrepancy

Nicolau Oliver Burwitz*[1] and Carlos Seara†[1]

[1]Universitat Politècnica de Catalunya

Abstract

In this paper we consider the problem of computing the discrepancy of a bichromatic point set by using boxes and disks and comparing the respective algorithmic complexities.

1 Introduction

Let S be a bichromatic d-dimension n-point set. Let R and B be a the set of red and blue points from S respectively, so $S = R \cup B$. The colouring of the points in S is expressed in the mapping $\chi : S \to \{-1, 1\}$, where blue points are negative and red points are positive. Let us define the bichromatic discrepancy of a geometric shape $\mathcal{SH}$:

$$\Delta(\mathcal{SH}) = \sum_{x \in (\mathcal{SH} \cap S)} \chi(x).$$

This is, the discrepancy of the shape is the number of red points minus the number of blue points. The maximum bichromatic discrepancy of a family of shapes $\mathcal{SH} \in \mathcal{F}$ is defined as:

$$\text{Max}\Delta(S, \chi, \mathcal{F}) = \max_{\mathcal{SH} \in \mathcal{F}} |\Delta(\mathcal{SH})|.$$

The main goal of this paper is to compare existing algorithms and approaches to solve the problem of computing the maximum bichromatic discrepancy of the set S using various families of shapes. The shapes considered are boxes and disks, in various dimensions.

Applications of computing discrepancy are present in several areas of computer science. Three major ones are mentioned in the introduction of the paper by Dobkin et al. [9] are the Agnostic PAC-Learning, ϵ-Approximations, and Sampling Patterns in Graphics.

Sampling patterns are used in ray-tracing for rendering digital images. If the pattern is ill designed it yields visible biasing artefacts. Computing the boxes discrepancy in $2d$ is related to the design of good patterns.

If instead of pixels we instead assume more circular shaped receptors, as the human eye's photo-receptors roughly are, and furthermore take into account other optical effects natural to human eyesight, it suggests studying discrepancy on disks instead of boxes.

There is a lot literature about discrepancy, and the books by Matousek [13] and Chazelle [5] cover in depth the topic. For related results see Bereg et al. [4] and Díaz-Báñez et al. [6, 7].

2 Boxes discrepancy

We first introduce the approach presented by Dobkin et al. [9], and Gunopulos [11], to compute the boxes bichromatic discrepancy in $1d$ and $2d$.

2.1 $1d$: Intervals

As boxes and disks both define intervals in the 1-dimension ($1d$) case, the results from Dobkin et al. [9] apply to both shapes. We specially want to highlight some results for the $1d$ case, as they provide the properties that are fundamental for the algorithms in further sections.

Lemma 1 *[9] Given an interval $[l, r]$ on our $1d$ setting, the discrepancy $\Delta([l, r]) = \Delta([l, m]) + \Delta([m, r])$ where $m \in [l, r]$ and $m \notin S$. See Figure 1.*

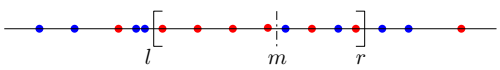


Figure 1: Example points in $1d$ with discrepancy 4.

This allows us to divide and conquer the computation of the discrepancy.

Lemma 2 *[9] Given an interval $[l, r]$ on our $1d$ setting, let the maximum discrepancy interval be $[a, b] \in [l, r]$. Then for any $m \in [a, b]$ the interval $[a, m]$ maximises the discrepancy among all intervals in $[l, m]$ that have m as the right endpoint. Analogously, for the discrepancy among all intervals in $[m, r]$ with m as the left endpoint.*

*Email: nicolau.oliver@estudiantat.upc.edu

†Email: carlos.seara@upc.edu. Supported by project PID2019-104129GB-I00/ MCIN/ AEI/ 10.13039/501100011033.

Lemma 2 introduces the connection between computing the maximum discrepancy in intervals with some fixed endpoints and computing the maximum discrepancy among all intervals. It specifically implies:

Observation 3 *[9] Given the maximum discrepancy of two consecutive intervals $[l, m]$ and $[m, r]$ with the fixed endpoint m, we can compute the maximum discrepancy for their union interval $[l, r]$.*

From Observation 3 we can intuitively see how we can build a segment tree of the points to represent all possible intervals. Computing this tree takes $O(n \log n)$ time and allows us to solve the static $1d$ maximum discrepancy.

But this tree also allows for a dynamic algorithm. Updates need only to traverse a $O(\log n)$-LENGTH path from the new leaf (or deleted leaf) to the root of the tree. This allows us to solve the dynamic $1d$ maximum discrepancy.

Theorem 4 *[9] The maximum discrepancy for intervals in $1d$ can be computed in $O(n \log n)$ time (linear if input is sorted) and $O(n)$ space. Computing updates after insertion/deletion of a point can be done in $O(\log n)$ time and $O(n)$ space.*

2.1.1 Axis-parallel boxes

The key strategy to tackle the $2d$ setting, is to find projections back to $1d$. The axis-parallel boxes is a perfect example.

Fix the y-coordinates of the axis-parallel box. We have $\Theta(n^2)$ pairs to choose from, and each of them defines a horizontal strip. Because we fixed them, the y-coordinates of the points inside the strip become irrelevant. See Figure 2.

Lemma 5 *[9] Computing the maximum discrepancy box in a fixed horizontal strip is equivalent to finding the maximum discrepancy interval of the points inside the strip projected onto the x-axis.*

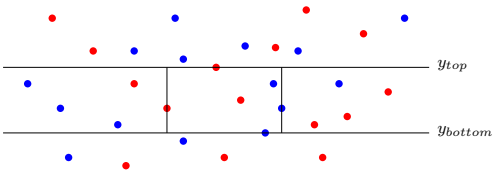


Figure 2: Discrepancy with boxes in $2d$.

These projections allow us to use the previous results to design algorithms. Let S again be the input set of n points. An outline of the algorithm step by step is:

1. Sort S by y-coordinate[1], obtaining the order $p_1, \ldots, p_n$.
2. For each p_i do:
   (a) Initialise the segment tree with only p_i, this represents the strip that only contains p_i.
   (b) For each p_j such that $i < j$ do:
      i. Update the segment tree by inserting p_j.
      ii. If the new maximum discrepancy is larger than the one seen so far, record the box.

Step 1 has cost $O(n \log n)$ time. Step 2.b.i has cost $O(\log n)$ time. Step 2.a and 2.b.ii have cost $O(1)$ time. Both loops 2 and 2.b have $O(n)$ iterations, so the total complexity is $O(n^2 \log n)$ time and $O(n)$ space. This algorithm is straightforward to extend to higher dimensions, for each new dimension the "strip" is determined by two extra points, so the complexity is $O(n^{2(d-1)} \log n)$ time and $O(n)$ space.

This can be improved by applying divide and conquer to the y-axis, after sorting the input by both coordinates as a pre-computation. This and more improvements where shown by Barbay et al. [2] to result in an $O(n^2)$ time algorithm, or even faster under some parametrizations of the input. They extend this approach to higher dimensions, resulting in an $O(n^d)$ time algorithm. This running time is tight up to subpolynomial factors, as proven by Backurs et al. [1].

3 Disk discrepancy

Disks are equivalent to intervals in $1d$, but disks in $2d$ do not satisfy the analogous of Lemma 1. There is no easy way to decompose a disk into smaller disks. Analogously, Lemma 2 doesn't hold.

Nevertheless, we can still apply the key strategy presented in Subsection 2.1.1, finding projections back to $1d$. The projection we present is equivalent to the one used by Bereg et al. [3].

For $p_i, p_j \in S$ consider all disks that pass through them. All their centers lie on the bisector of the segment $\overline{p_i p_j}$. See Figure 3.

Definition 6 *The* oriented angle $\alpha_k^{ij} \in [-\pi, \pi]$ *of a point $p_k \in S$ with respect to $p_i, p_j \in S$ is the supplementary angle of $\angle p_i p_k p_j$. It is positive if p_k is to the right of the directed line $\overrightarrow{p_i p_j}$, otherwise negative.*

If we order the points in S by the oriented angle, the *furthest* left point in Figure 3 has the smallest oriented angle. The *furthest* right point has the largest oriented angle. Points *close* to segment $\overline{p_i p_j}$ have oriented angle close to 0.

[1]The following for loops iterate in this order.

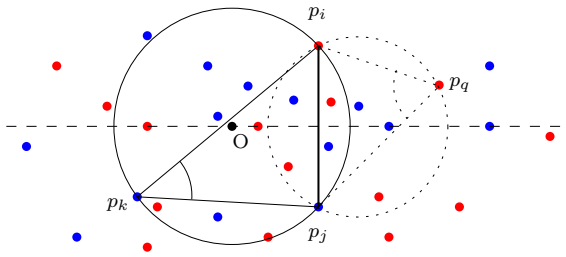


Figure 3: Discrepancy with disks in $2d$. Oriented angle α is negative, and oriented angle β is positive.

Traversing all the points $p_k \in S$ by their oriented angle $\alpha_k^{ij} = \pm(\pi - \angle p_i p_k p_j)$ can be visualised as sliding the center O of the disk over the bisector from left to right.

Thus, for fixed points $p_i, p_j \in S$, let

$$\Lambda^{ij} = [\alpha_1^{ij}, \ldots, \alpha_k^{ij}, \ldots, \alpha_{n-2}^{ij}]$$

be the list (in fact, a multi-set) of oriented angles of the points p_k with respect to p_i, p_j. Each α_k^{ij} retains the colour of the point it represents.

Definition 7 *The inverse of an angle α_k^{ij} is:*

$$inv(\alpha_k^{ij}) = \begin{cases} \text{if } \alpha_k^{ij} < 0 : swap\_color(\pi - |\alpha_k^{ij}|) \\ \text{else } \alpha_k^{ij}. \end{cases}$$

Analogously, the inverse of a list

$$inv(\Lambda^{ij}) = [\,\forall k : inv(\alpha_k^{ij})\,]$$

is just the list of the inverse of its elements.

In the circular discrepancy, the intervals of angles must be of the form $[\beta - \pi, \beta]_{\Lambda^{ij}}$ where $\beta \in [0, \pi]$. This interval represents the disk through p_i, p_j with inscribed angle β, where β is positive; so it lies to the right of $\overrightarrow{p_i p_j}$. Let $[0, \beta]_{inv(\Lambda^{ij})}$ be the same interval/disk over the inverse angles of Λ^{ij}.

Lemma 8 *The disk discrepancy of Λ^{ij} is equal to a constant with respect to β, plus the interval discrepancy of the inverse angles with fixed endpoint 0.*

$$\Delta([\beta - \pi, \beta]_{\Lambda^{ij}}) = \Delta([-\pi, 0]_{\Lambda^{ij}}) + \Delta([0, \beta]_{inv(\Lambda^{ij})}).$$

Definition 9 *The projection of a list of angles is:*

$$\mathcal{P}(\Lambda^{ij}) = [\,\forall k : \{\alpha_k^{ij} \cup inv(\alpha_k^{ij})\}\,].$$

In few words, the projection is duplicating the negative angles with its inverses. So a negative red angle is duplicated by inserting its positive value in blue. As a consequence of Lemma 8 we have the following result.

Theorem 10 *The disk discrepancy of a list of angles is equal to the interval discrepancy of its projection, with the restriction of containing the interval $[-\pi, 0]$,*

$$\Delta([\beta - \pi, \beta]_{\Lambda^{ij}}) = \Delta([-\pi, \beta]_{\mathcal{P}(\Lambda^{ij})}), \quad \beta \in [0, \pi].$$

The algorithm for circular discrepancy starts by fixing two points. We have $O(n^2)$ pairs to choose from, and each of them defines a bisector. Because we fixed the points, the remaining points can be sorted by their oriented angle. These angles are then projected via $\mathcal{P}$. Using this projection, the algorithm is straightforward:

1. For each pair $(p_i, p_j) \in S \times S$ such that $i \neq j$ do:
   (a) Compute the list of oriented angles Λ^{ij} of all points with respect to p_i, p_j.
   (b) Compute the projection $\mathcal{P}(\Lambda^{ij})$.
   (c) Compute the maximum discrepancy interval with the restriction:
   $$\Delta(\mathcal{P}(\Lambda^{ij})) = [-\pi, \beta], \quad \beta \in [0, \pi].$$

To compute Step 1.c it is enough to modify slightly the algorithm for interval discrepancy.

Steps 1.a and 1.b have cost $O(n)$ and Step 1.c has cost $O(n \log n)$ time. The Loop 1 is $O(n^2)$ iterations so the total complexity is $O(n^3 \log n)$ time and $O(n)$ space. This is equivalent to the complexity of the algorithm presented by Bereg et al. [3].

In comparison to this approach, a faster and more general algorithm exists by Dobkin and Eppstein [8]. Their approach extends to shapes bounded by algebraic curves, such as circles and ellipses in $2d$. Lifting the points to the paraboloid, in order to compute the discrepancy inside the disks they compute the lifted points below the corresponding plane, using the topological sweep algorithm by Edelsbrunner et al. [10]. Their resulting complexity is $O(n^3)$ time for disks, and $O(n^5)$ time for ellipses.

We think that finding the maximum discrepancy disk in $2d$ could be 4-SUM hard. The reduction can be done using Proposition 11 in Heras et al. [12], and Theorems 6, 7 and Lemma 12 in Bereg et al. [4].

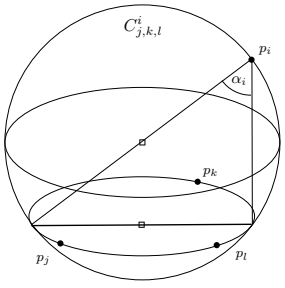


Figure 4: Generalization of the oriented angle in 3d.

It is straightforward to generalise to $d > 2$ the algorithm we present above and the one by Dobkin and Eppstein [8]. See Figure 4.The respective complexities are $O(n^{d+1} \log n)$ and $O(n^{d+1})$ time and $O(n)$ space.

4 Unoriented boxes

In contrast to disks in $2d$, boxes with arbitrary orientations in $2d$ can be decomposed into smaller boxes of that same orientation. Again this impacts positively the complexity of the algorithm.

Lemma 11 *Computing the maximum discrepancy box in a fixed strip with orientation $\vec{v}$ is equivalent to finding the maximum discrepancy interval of the points inside the strip projected onto a line with direction $\vec{v}$.*

This is just a generalisation of Lemma 5, and allows us to reuse the results for axis parallel boxes.

Lemma 12 *A set S of n points in $2d$ has $O(n^2)$ unique linear projections onto $1d$.*

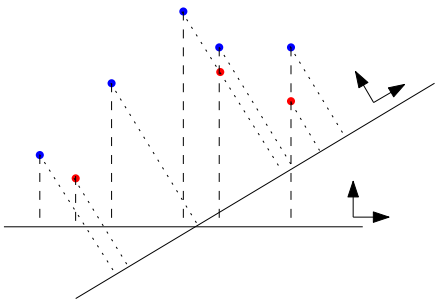


Figure 5: Two projections on a line.

Furthermore, the projections of the set S can be sorted by their angle. Two consecutive projections differ by the swap of two points. This can be processed in two insertion/deletion updates using the dynamic algorithm for intervals in $1d$. The trivial algorithm reusing the boxes discrepancy algorithm is $O(n^4 \log n)$ time. We are currently studying other approaches to improve this trivial complexity.

5 Conclusions

The following table illustrates the time complexities of the algorithms for computing the discrepancies for boxes, disks and unoriented boxes.

	Boxes	Disks	Unoriented Boxes
$d = 1$	$O(n \log n)$		
$d = 2$	$O(n^2)$	$O(n^3)$	$O(n^4 \log n)$
$d \geq 3$	$O(n^d)$	$O(n^{d+1})$	?

Open problem 13 *Is the maximum bichromatic disk discrepancy problem in $2d$ 4-SUM hard?*

Exploiting advanced data structures in the $2d$ setting could be promising. Specifically in the case of oriented boxes, we attempted fruitlessly to use quad-trees to extend the algorithm for discrepancy on intervals in $1d$. The hope was that quad-trees of the points can be rotated with only $O(n^2)$ updates.

Open problem 14 *Is there a faster algorithm for unoriented boxes in $d \geq 2$?*

References

[1] A. Backurs, N. Dikkala and C. Tzamos. Tight Hardness Results for Maximum Weight Rectangles. *ArXiv abs/1602.05837*, (2016).

[2] J. Barbay, T.M. Chan, G. Navarro and P. Pérez-Lantero. Maximum-weight planar boxes in $O(n^2)$ time (and better). *Information Processing Letters*, Vol. 114(8), (2014), pp. 437–445.

[3] S. Bereg, O. Daescu, M. Zivanic and T. Rozario. Smallest Maximum-Weight Circle for Weighted Points in the Plane. *ICCSA*, (2015), pp. 244–253.

[4] S. Bereg, J.M. Díaz-Báñez, D. Lara, P. Pérez-Lantero, C. Seara, and J. Urrutia. On the coarseness of bicolored point sets. *Computational Geometry: Theory and Applications*, 46(1), (2013), pp. 65–77.

[5] B. Chazelle. *The Discrepancy Method in Computational Geometry*. Handbook of Discrete and Computational Geometry, CRC Press 44, (2004), pp. 983–996.

[6] J.M. Díaz-Báñez, R. Fabila, P. Pérez-Lantero, I. Ventura. New results on the coarseness of bicolored point sets. *Information Processing Letters*, 123, (2017), pp. 1–7.

[7] J.M. Díaz-Báñez, M.A. López, C. Ochoa, P. Pérez-Lantero. Computing the coarseness with strips or boxes. *Discrete Applied Mathematics*, 224(19), (2017), pp. 80–99.

[8] D.P. Dobkin and D. Eppstein. Computing the Discrepancy. *9th Annual Symposium on Computational Geometry*, (1993).

[9] D.P. Dobkin, D. Gunopulos, and W. Maass. Computing the maximum bichromatic discrepancy, with applications to computer graphics and machine learning. *Journal of Computer and Systems Sciences*, 52(3), (1996), pp. 453–470.

[10] H. Edelsbrunner and L.J. Guibas. Topologically sweeping an arrangement. *Journal of Computer and System Sciences*, Vol. 38(1), (1989), pp. 165–194.

[11] D. Gunopulos. Computing the Discrepancy. *Thesis in Princeton University*, Princeton, N.J. (1995).

[12] A. de las Heras, G. Esteban, D. Garijo, C. Huemer, A. Lozano, N. Oliver and D. Orden. Measuring cocircularity in a point set. *ECG23* (2023).

[13] J. Matoušek. *Geometric Discrepancy: An Illustrated Guide*. Springer-Verlag, (1999).

Approximate shortest paths on weighted disks

Prosenjit Bose*[1], Jean-Lou De Carufel†[2], Guillermo Esteban‡[1,3], and Anil Maheshwari§[1]

[1]School of Computer Science, Carleton University, Canada
[2]School of Electrical Engineering and Computer Science, University of Ottawa, Canada
[3]Departamento de Física y Matemáticas, Universidad de Alcalá, Spain

In this paper, we study optimal obstacle-avoiding paths from a source point s to a target point t in the 2-dimensional plane. A general version of the shortest path problem allows the two-dimensional space to be subdivided into regions. Each of the regions has a (non-negative) weight associated to it, representing the cost per unit distance of traveling in that region. This variant, called the Weighted Region Problem (WRP), was proposed by Mitchell and Papadimitriou [3]. They propose an approximating algorithm which computes a $(1+\varepsilon)$-approximation path in $O(n^8 \log \frac{nNW}{w\varepsilon})$ time, where N is the maximum integer coordinate of any vertex of the subdivision, W (respectively, w) is the maximum finite (respectively, minimum non-zero) integer weight assigned to faces of the subdivision.

Recently, it has been shown that the WRP cannot be solved exactly within the Algebraic Computation Model over the Rational Numbers (ACM$\mathbb{Q}$) [2], i.e., the solutions to some instances of the WRP cannot be expressed as a closed formula in ACM$\mathbb{Q}$.

This result probably explains the lack of exact algorithms for the WRP, and the fact that several authors propose algorithms for computing approximated paths. The most common scheme followed in the literature is to position Steiner points, and then build a graph by connecting pairs of Steiner points, see, e.g., [1, 4]. An approximate solution is constructed by finding a shortest path in this graph, by using well-known combinatorial algorithms (e.g., Dijkstra's algorithm).

Let D be a set of disjoint disks in the plane, and s, t be two points. To compute a shortest path, which avoids D, between s and t, we can use an algorithm based on Dijkstra's shortest path algorithm. However, we are not aware of any work where the shortest path is allowed to go through the disks $D_i \in D$, where each D_i has a non-negative value k_i associated to it.

It is straightforward to prove that if the weight of the disks is at least $\frac{\pi}{2}$, then the disks act as obstacles, and the problem can be solved in $O(|D|^2 \log |D|)$ time. Thus, we focus in finding an algorithm to compute an approximate shortest path from s to t, when $0 < k_i < \frac{\pi}{2}$.

Our first approach is the particular case in which $|D| = 1$, and s is a fixed source point on the boundary of the disk. In this case we obtained a promising result: a $(1+\varepsilon)$-approximation for the shortest path by carefully placing Steiner points on the boundary of the disk. In addition, we are currently working on the generalization of this partial result to the case of multiple disks.

Acknowledgments. This work is partially supported by NSERC. G. E. is also supported by project PID2019-104129GB-I00 funded by MCIN/AEI/10.13039/501100011033, and an FPU of the Universidad de Alcalá.

References

[1] L. Aleksandrov, A. Maheshwari, and J.-R. Sack. Determining approximate shortest paths on weighted polyhedral surfaces. *Journal of the ACM (JACM)*, 52(1):25–53, 2005.

[2] J.-L. De Carufel, C. Grimm, A. Maheshwari, M. Owen, and M. Smid. A note on the unsolvability of the weighted region shortest path problem. *Computational Geometry*, 47(7):724–727, 2014.

[3] J. S. B. Mitchell and C. H. Papadimitriou. The weighted region problem: finding shortest paths through a weighted planar subdivision. *Journal of the ACM (JACM)*, 38(1):18–73, 1991.

[4] Z. Sun and J. H. Reif. On finding approximate optimal paths in weighted regions. *Journal of Algorithms*, 58(1):1–32, 2006.

*Email: jit@scs.carleton.ca.
†Email: jdecaruf@uottawa.ca.
‡Email: g.esteban@uah.es.
§Email: anil@scs.carleton.ca.

Isomorphisms of simple drawings of complete multipartite graphs

Oswin Aichholzer*[1], Birgit Vogtenhuber†[1], and Alexandra Weinberger*[1]

[1]Institute of Software Technology, Graz University of Technology, Austria.

Simple drawings are drawings of graphs on the sphere in which any two edges intersect at most once (either at a common endpoint or a proper crossing), and no edge intersects itself. When investigating simple drawings, it usually is sufficient to study one representative of each isomorphism class (for different types of isomorphism). One essential element when studying simple drawings are the rotations of crossings or vertices, that is, the cyclic order in which edges emanate from the vertex or crossing. Via rotations and crossings, different types of isomorphisms are defined. Two labeled simple drawings are RS-isomorphic if either all vertices have the same rotations or all have the inverse rotations, CE-isomorphic (also known as weakly isomorphic) if the same pairs of edges cross, CR-isomorphic if either all crossings have the same rotations or all have the inverse rotations, ERS-isomorphic if either all crossings and vertices have the same rotations or all have the inverse rotations, and CO-isomorphic if for each edge the crossings along the edges are in the same order. Finally, two labeled simple drawings are strongly isomorphic if there is a homeomorphism of the sphere that transforms one drawing into the other. Unlabeled simple drawings are isomorphic w.r.t. some type of isomorphism if there exists a labeling such that the labeled drawings are isomorphic w.r.t. that type.

All listed isomorphisms and some combinations of those isomorphisms can be relevant for general graphs. However, there are some isomorphisms implying each other by definition. CR-isomorphism and CO-isomorphism each imply CE-isomorphism, and ERS-isomorphism implies RS-isomorphism and CR-isomorphism (and thus CE-isomorphism). Moreover, for connected graphs, any two simple drawings of the same connected graph are strongly isomorphic if and only if they are ERS-isomorphic and CO-isomorphic (with the same labeling) [1]. For complete graphs, CE-isomorphism, CR-isomorphism, RS-isomorphism, and ERS-isomorphism are all equivalent [2, 3]. Thus, the only relevant types of isomorphism for simple drawings of complete graphs are CE-isomorphism (weak isomorphism) and strong isomorphism.

*Emails: {oaich,weinberger}@ist.tugraz.at. Supported by the Austrian Science Fund (FWF) grant W1230.

†Email: bvogt@ist.tugraz.at. Supported by the FWF grant I 3340-N35.

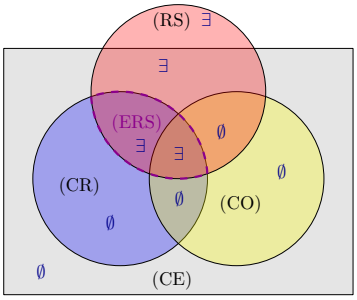


Figure 1: Implications between different isomorphisms for graphs G as defined in Theorem 1. An area is marked with ∃ if there exist drawings of G that are isomorphic w.r.t. exactly the overlapping types of isomorphisms (and no others), and ∅ if there aren't.

As opposed to complete graphs, there are simple drawings of (general) complete multipartite graphs that are RS-isomorphic but not CE-isomorphic. We give a complete characterization which implications do or do not always hold for drawings of complete multipartite graphs, depending on the cardinalities of the partition classes; see Figure 1 for some of our findings. As a main result, we prove the following.

Theorem 1 *Let G be a complete multipartite graph in which each partition class has at least three vertices. Then for any two simple drawings of G it holds that*
(1) CE-isomorphism implies RS-isomorphism, and
(2) CO-isomorphism implies strong isomorphism.

For simple drawings of $K_{2,n}$, we show that RS-isomorphism and CO-isomorphism together imply strong isomorphism, while CE-isomorphism does not imply RS-isomorphism for $n \geq 4$.

References

[1] J. Kynčl, Improved enumeration of simple complete topological graphs, *Discrete Comput. Geom.* **50** (2013), 727–770.

[2] J. Kynčl, Simple realizability of complete abstract topological graphs in P, *Discrete Comput. Geom.* **45** (2011), 383–399.

[3] E. Gioan, Complete Graph Drawings Up to Triangle Mutations, *Discrete Comput. Geom* **67** (2022), 985–1022.

The free-linking task: Graphs for better discrimination of sensory similarity

Jacob Lahne*[1], David Orden†[2], Katherine Phetxumphou‡[3], and Marino Tejedor-Romero§[4]

[1]Department of Food Science & Technology, Virginia Tech, Blacksburg, VA 14060, USA.
[2]Departamento de Física y Matemáticas, Universidad de Alcalá, 28805, Spain.
[3]Berry Tasting Lab, Driscoll's, Watsonville, CA 95076, USA.
[4]Departamento de Automática, Universidad de Alcalá, 28805, Spain.

Identifying similarities and differences between foods is of great importance for both sensory science and industry, particularly useful to understand how consumers perceive a product. This analysis was traditionally performed by a trained panel, but the need for fine training implies large costs in time and money. Therefore, in the last few decades a number of alternative, cheaper and faster, methods have been proposed.This talk reviews a new rapid method which, for the first time, used graphs for both gathering and processing consumers' opinions [1].

One of the most popular sensory rapid methods is *free-sorting*, in which the participants are asked to distribute the products in disjoint groups according to their own criteria, without restrictions on the number of groups or the number of products in each group. See Figure 1, left.

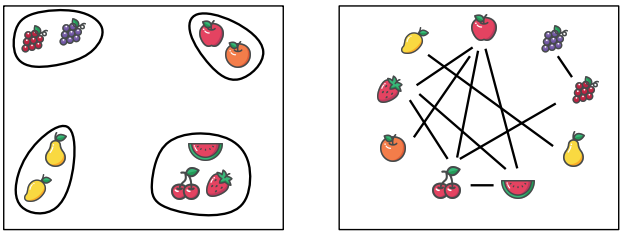


Figure 1: Examples of a participant's opinion using free-sorting (left) and free-linking (right).

Despite its usefulness, free-sorting has disadvantages. First, the groups being disjoint implies transitive similarity, i.e., each group corresponding to a clique. In the example, the participant had to choose whether to group the red grape with the black grape (as in the figure) or with the cherry (included in a group of other red fruits). Second, free-sorting accounts for purely binary similarity. In the example, the participant might consider the black grape being just two steps away from the cherry, since both are similar to the red grape.

In order to overcome these issues, we have recently proposed the *free-linking* method for gathering opinions [1]. In this method, the participants are asked to join with a link those pairs of samples they consider similar. For this connect-the-dots task, the samples are presented on the vertices of a regular polygon, randomizing the sample positions for each participant in order to avoid bias. See Figure 1, right.

This method was tested against free-sorting in two tasting sessions, with spice blends (10 samples, 58 participants) and chocolate bars (10 samples, 63 participants). The results were compared using both standard statistical techniques and graph parameters, finding that the latter allowed to highlight that the results from free-linking were more robust and realistic. Figure 2 compares some parameters for the linking and sorting global graphs obtained merging all the participants' opinions.

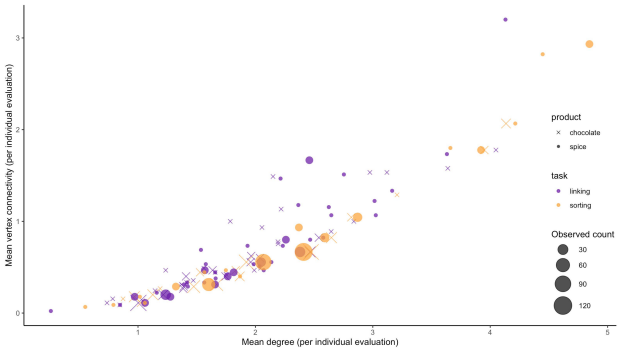


Figure 2: Comparison of mean degree versus mean vertex connectivity for free-sorting and free-linking.

References

[1] J. Lahne, K. Phetxumphou, M. Tejedor-Romero, D. Orden, The free-linking task: A graph-inspired method for generating non-disjoint similarity data with food products, *Food Quality and Preference* **95** (2022) 104355.

*Email: jlahne@vt.edu.
†Email: david.orden@uah.es. Supported by Project PID2019-104129GB-I00/MCIN/AEI/10.13039/501100011033.
‡Email: Kat.Phetxumphou@driscolls.com.
§Email: marino.tejedor@uah.es. Supported by Project PID2019-104129GB-I00/MCIN/AEI/10.13039/501100011033.

The Borsuk number of geometric graphs

José Cáceres[1], Delia Garijo[2], Alberto Márquez [2], and Rodrigo I. Silveira[3]

[1]Dpto. de Matemáticas, Universidad de Almería
[2]Dpto. de Matemática Aplicada I, Universidad de Sevilla
[3]Dept. de Matemàtiques, Universitat Politècnica de Catalunya

In 1933, Karol Borsuk wondered if every set X in $\mathbb{R}^d$ could be partitioned into $d+1$ closed (sub)sets each with diameter smaller than that of X [1]. Here, the diameter is defined as the maximum of the distances between two points in the set, under the Euclidean metric. This leads to the concept of the *Borsuk number*. For a set $X \subset \mathbb{R}^d$, the Borsuk number $b(X)$ is the smallest number such that X can be partitioned into $b(X)$ subsets, each with diameter smaller than that of X. The answer to Borsuk's question was shown to be positive for $d = 2, 3$, and for general d for centrally symmetric convex bodies and smooth convex bodies. To the surprise of many researchers, the general answer turned out to be negative, as shown in 1993 by Kahn and Kalai [2]. Since then, research on variants of the Borsuk problem has continued in a plethora of directions, see [3] for a recent survey.

In this work, we propose a formulation of the problem in the context of geometric graphs. A *(plane) geometric graph* is a plane undirected graph $G = (V, E)$ whose vertices are points in $\mathbb{R}^2$, and whose edges are straight-line segments connecting pairs of points. In addition, each edge has a weight equal to the Euclidean distance between its endpoints. We are interested in the *locus* of G, denoted by $\mathcal{L}_G$, which is the set of all points of the plane that are on G. Thus, we treat both G and $\mathcal{L}_G$, interchangeably, as a closed point set. The distance between two points in $\mathcal{L}_G$ is the length of a shortest path between them in G (note such a path will contain up to two fragments of edges, if the points are not vertices). The *diameter* of $\mathcal{L}_G$ or *(continuous) diameter* of G is the maximum distance between any two points in $\mathcal{L}_G$. In contrast to (abstract) graphs, in a geometric graph, there can be an infinite number of pairs of points whose distance is equal to the diameter.

We extend the concept of Borsuk number to geometric graphs. Conceptually, it is the smallest number $b(G)$ such that G can be partitioned into $b(G)$ subgraphs, each with smaller diameter than $\mathcal{L}_G$. However, we need to define carefully how a geometric graph can be partitioned. We consider partitions of $\mathcal{L}_G$ by a sequence of cuts with straight lines. A line ℓ naturally partitions $\mathcal{L}_G$ into two geometric subgraphs (possibly, one empty). Moreover, to guarantee that the partition by ℓ does not produce a disconnected subgraph, we add to both subgraphs the maximal segment of ℓ intersecting $\mathcal{L}_G$.[1] Figure 1 (left) illustrates this for a square. After partitioning the square with a vertical line ℓ (dashed) through its center point, we obtain two subgraphs: all points of $\mathcal{L}_G$ on each halfplane induced by ℓ, union the longest segment in ℓ with endpoints in $\mathcal{L}_G \cap \ell$. Since this partitions the graph into two subgraphs (of $\mathcal{L}_G \cup \ell$), each with smaller diameter than that of $\mathcal{L}_G$, its Borsuk number is two (best possible). However, sometimes more subgraphs are needed. The example in Figure 1 (right) shows a 4-star graph, requiring at least two lines, giving at least three subgraphs. Thus its Borsuk number is three.

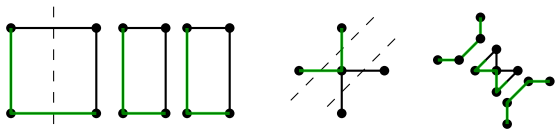


Figure 1: Left: a square with side length 1 and diameter 2 (given by green paths), and a partition with a line. Right: a 4-star partitioned into three subgraphs.

This illustrates the main question studied in this work: What is the Borsuk number of a geometric graph? Clearly, the answer depends on the graph.

In this talk, we will show that, in general, any geometric graph with n vertices can be partitioned with lines into $2n$ subgraphs with smaller diameter. Moreover, we will give upper bounds for $b(G)$ that depend on the number of disjoint diameter paths in the graph. We will also show that the Borsuk number of a tree can be two or three, and discuss how to efficiently figure it out. Finally, we will mention several open problems in relation to this new concept.

References

[1] K. Borsuk. Three theorems on the n-dimensional sphere. *Fund. Math*, 20:177–190, 1933.

[2] J. Kahn and G. Kalai. A counterexample to borsuk's conjecture. *B. Am. Math. Soc.*, 29(1):60–62, 1993.

[3] C. Zong. Borsuk's partition conjecture. *Jpn. J. Math.*, 16:185–201, 2021.

[1]So, actually, the partition gives two subgraphs of $\mathcal{L}_G \cup \ell$.

Crossing minimal and generalized convex drawings: 2 non-hard problems

Joachim Orthaber*[1]

[1]Institute of Software Technology, Graz University of Technology, Austria

Simple drawings are drawings of graphs in the plane such that each pair of edges meets in at most one point, either a common endvertex or a crossing. In this work we study two problems on simple drawings that are hard in general but get easy on a certain subclass. As the first problem, Arroyo et al. [1] showed that it is NP-complete to decide whether a specific edge can be added to a simple drawing of a non-complete graph without violating simplicity. In contrast to this, by Levi's Extension Lemma, every pseudolinear drawing can be extended by any set of edges. We show a similar result for *crossing minimal drawings*, that is, drawings of a graph G which contain the minimum number of crossings over all drawings of G.

Theorem 1 *Let $\mathcal{D}$ be a crossing minimal drawing of a graph on n vertices. Then $\mathcal{D}$ can be extended to a simple drawing of the complete graph K_n.*

Proof idea. Note that every crossing minimal drawing $\mathcal{D}$ is a simple drawing. In a first step we show that adding a single edge such that it creates a minimum number of additional crossings results in a simple drawing $\mathcal{D}'$. However, $\mathcal{D}'$ need not be crossing minimal anymore. So in a second step we add a set of edges simultaneously to $\mathcal{D}$ such that each single added edge has a minimum number of crossings with $\mathcal{D}$. Over all possibilities to do so, we then show that choosing a drawing $\mathcal{D}''$ which in addition minimizes the total number of crossings ensures that $\mathcal{D}''$ is simple. □

While it is known that no crossing minimal drawing of K_n is pseudolinear for large enough n, Arroyo et al. [2] asked the question whether all crossing minimal drawings of K_n might be *generalized convex drawings* (short *g-convex*). These are simple drawings where every triangle has a *convex side* Δ, that is, for each pair of vertices in Δ also the edge connecting them lies completely inside Δ. If there exists a choice of a convex side for each triangle such that every triangle T_2, being contained in the convex side Δ_1 of a triangle T_1, has its convex side Δ_2 contained in Δ_1, then the drawing is called *hereditarily convex* (short *h-convex*).

This brings us to the second problem. García et al. [3] showed that it is NP-complete to decide whether a simple drawing $\mathcal{D}$ of K_n contains a plane (no two edges cross) subdrawing with a given number of edges. In this context we call a subdrawing of $\mathcal{D}$ *maximal plane* if it is plane and no edge of $\mathcal{D}$ can be added to it without violating planarity. We call a subdrawing *maximum plane* if it is plane and contains the highest number of edges over all plane subdrawings of $\mathcal{D}$. If a plane subdrawing contains $3n-6$ edges, then we call it a *combinatorial triangulation*.

Theorem 2 *Let $\mathcal{D}$ be a g-convex drawing of K_n. Then every maximal plane subdrawing of $\mathcal{D}$ is maximum plane. Moreover, if $\mathcal{D}$ is h-convex but not pseudolinear, then every maximal plane subdrawing of $\mathcal{D}$ is a combinatorial triangulation.*

We can show this by combining some results from [2] and [3]. We can further confirm by computer that all crossing minimal drawings of K_n for $n \leq 12$ are h-convex. Since all crossing minimal straight-line or pseudolinear drawings are known to have a triangular convex hull, this gives rise to the following conjecture.

Conjecture 3 *Let $\mathcal{D}$ be a crossing minimal drawing of K_n for $n \geq 3$. Then every maximal plane subdrawing of $\mathcal{D}$ is a combinatorial triangulation.*

Acknowledgments. We thank O. Aichholzer, H. Bergold, M. Scheucher, B. Vogtenhuber, and A. Weinberger for inspiring discussions and helpful comments.

References

[1] Alan Arroyo, Fabian Klute, Irene Parada, Raimund Seidel, Birgit Vogtenhuber, and Tilo Wiedera. Inserting one edge into a simple drawing is hard. *Discrete & Computational Geometry*, 69(3):745–770, 2023. doi:10.1007/s00454-022-00394-9.

[2] Alan Arroyo, Dan McQuillan, R. Bruce Richter, and Gelasio Salazar. Convex drawings of the complete graph: topology meets geometry. *Ars Mathematica Contemporanea*, 22(3):27, 2022. doi:10.26493/1855-3974.2134.ac9.

[3] Alfredo García, Alexander Pilz, and Javier Tejel. On plane subgraphs of complete topological drawings. *Ars Mathematica Contemporanea*, 20(1):69–87, 2021. doi:10.26493/1855-3974.2226.e93.

*Email: orthaber@ist.tugraz.at. Research supported by the Austrian Science Fund (FWF) grant W1230.

Some routing problems on a half-line with release times and deadlines

Alfredo García*[1] and Javier Tejel*[1]

[1]Departamento de Métodos Estadísticos and IUMA, Universidad de Zaragoza, Spain

The following problem is studied in [1, 3]. Let $N = \{1, 2, \ldots, n\}$ be a set of customers located on the real half-line $\mathbb{R}^+$ and let D be a depot located at $x = 0$. The distance (and also the travel time) from customer i to the depot is denoted by τ_i. A vehicle has to deliver goods from the depot to the customers. Each customer places an order to the depot and this order is associated with a time window $[r_i, l_i]$, with $l_i = r_i + S - \tau_i$. The release time r_i specifies the earliest possible time the vehicle can depart the depot to deliver at i. S can be seen as a service guarantee such that customer i cannot be served after $r_i + S$. Thus, $r_i + S - \tau_i$ is the latest dispatch time for customer i. For example, we can think of a restaurant delivering meals at home as a depot, the release time of an order (a customer ordering for a meal) as the time that the order can be dispatched from the restaurant after preparing the meal, and S as the time in which the restaurant guaranties that the order will be delivered.

The problem analyzed in [3] is determining the minimum possible completion time c^* of a schedule of delivery routes that can be executed by a single driver, each starting and ending at the depot, such that each order i is dispatched at or after r_i and delivered at or before $r_i + S$. Any feasible solution to the problem will consist of a set of k routes, visiting a subset of customers in each route. Assuming that the customers are ordered according to their release times, that is, $r_1 \leq \ldots \leq r_n$, it is proved in [3] that there is always an optimal delivery schedule with non-interlacing routes, where two routes K_1 and K_2, with $\min\{i | i \in K_1\} < \min\{j | j \in K_2\}$, are non-interlacing if and only if $\max\{r_i | i \in K_1\} < \min\{r_j | j \in K_2\}$. As a consequence, in any delivery schedule with non-interlacing routes, the customers visited in any route have consecutive release times.

Geometrically, one can imagine the set of customers as a set of n horizontal segments $s_1, \ldots, s_n$ such that the y-coordinate of s_i is τ_i and the x-coordinates of the endpoints of s_i are r_i and l_i, respectively. The abscissa axis represents time. For instance, Figure 1 shows a set of eight customers and a feasible schedule (in red), consisting of three routes, K_1, K_2 and K_3, to serve the eight customers. After waiting at the depot, K_1 starts at r_2, dispatches orders 2, 1 and 3, and ends at $r_2 + 2\tau_3$. K_2 starts at $r_2 + 2\tau_3$, dispatches orders 4 and 5, and ends at $r_2 + 2\tau_3 + 2\tau_5$. Finally, after waiting again at the depot for a while, K_3 starts at r_8, dispatches orders 8, 7 and 6, and ends at $r_8 + 2\tau_6$.

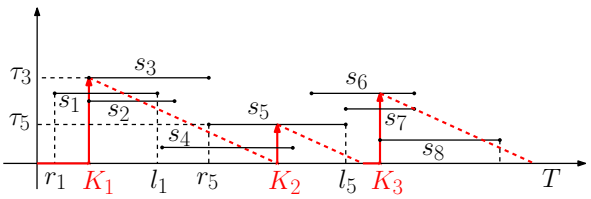


Figure 1: A feasible schedule to serve eight customers.

Let $c(i)$ be the minimum completion time of a non-interlacing schedule serving orders $\{1, \ldots, i\}$, or ∞ if it is not possible to serve $\{1, \ldots, i\}$ feasibly with a single server. Thus, c^* will be given by $c(n)$. Defining $c(0) = 0$, the following recurrence [3] allows one to compute $c(i)$, for $i = 1, \ldots, n$:

$$c(i) = \min_{0 \leq j < i} \left\{ \max\{c(j), r_i\} + 2 \max_{j < k \leq i} \{\tau_k\} \mid \max\{c(j), r_i\} \leq \min_{j < k \leq i} \{l_k\} \right\}$$

In this talk, we will show how to solve this recurrence in $O(n \log n)$ time, improving the $O(n^2)$ algorithm given in [3]. In addition, using the algorithm described in [2], if $S = \infty$, that is, there are no deadlines, then the previous recurrence can be solved in $O(n)$ time, improving the $O(n^2)$ algorithm provided in [3].

References

[1] C. Archetti, D. Feillet and M.G. Speranza, Complexity of routing problems with release times, *European Journal of Operational Research* **247** (2015), 797–803.

[2] L. Larmore and B. Schieber, On-line dynamic programming with applications to the prediction of RNA secondary structure, *J. of Algorithms* **12** (1991), 490–515.

[3] D. Reyes, A.L. Erera and M.W.P. Savelsbergh, Complexity of routing problems with release dates and deadlines, *European Journal of Operational Research* **266** (2018), 29–34.

*Emails: {olaverri, jtejel}@unizar.es. Research supported by project PID2019-104129GB-I00 / AEI / 10.13039/501100011033 of the Spanish Ministry of Science and Innovation, and by Gobierno de Aragón project E41-23R.

Recognizing rotation systems of generalized twisted drawings in $O(n^2)$ time

Oswin Aichholzer*[1], Alfredo García†[2], Javier Tejel‡[2], Birgit Vogtenhuber§[1], and Alexandra Weinberger*[1]

[1]Institute of Software Technology, Graz University of Technology, Austria.
[2]Departamento de Métodos Estadísticos. IUMA. Universidad de Zaragoza. Spain.

Simple drawings are drawings of graphs in which the edges are Jordan arcs and each pair of edges share at most one point (a proper crossing or a common endpoint). The rotation of a vertex is the cyclic order of its incident edges. The rotation system of a simple drawing is the collection of the rotations of all vertices. Two simple drawings of K_n have the same crossing edge pairs if and only if they have the same or inverse rotation systems [1].

A simple drawing D of K_n is generalized twisted if there exists a point O such that each ray emanating from O crosses each edge of D at most once and there exists a ray r emanating from O such that all edges of D cross r. We call a rotation system generalized twisted if there exists a generalized twisted drawing with that rotation system. Generalized twisted drawings have been used to improve bounds on plane substructures in (general) simple drawings of complete graphs [2]. Moreover, they are the biggest class for which it is known that each drawing has exactly $2n-4$ empty triangles [3], which is conjectured to be the minimum for all simple drawings. In addition to being useful for proving general results, generalized twisted drawings are quite interesting in their own right.

In this talk, we present an efficient algorithm to decide if a rotation system is generalized twisted.

Theorem 1 *Let R be the rotation system of a simple drawing of K_n. Then deciding whether R is generalized twisted can be done in $O(n^2)$ time.*

To obtain the algorithm, we first show the following statement for generalized twisted drawings; see Figure 1 for an illustration.

Theorem 2 *Let R be the rotation system of a simple drawing D of K_n and let V be the set of vertices of D. Then R is generalized twisted if and only if there exist two vertices v_1 and v_2 in V and a bipartition $A \cup B$ of the vertices in $V \setminus \{v_1, v_2\}$, where some of A or B can be empty, such that: 1. For every pair of vertices u_1 and u_2 that are either both in A or both in B, the edge (u_1, u_2) crosses the edge (v_1, v_2). 2. For every pair of vertices $a \in A$ and $b \in B$, the edge (a, b) does not cross (v_1, v_2). 3. Beginning at v_2 (respectively v_1), in the rotation at v_1 (respectively v_2), all the vertices in B appear before all the vertices in A.*

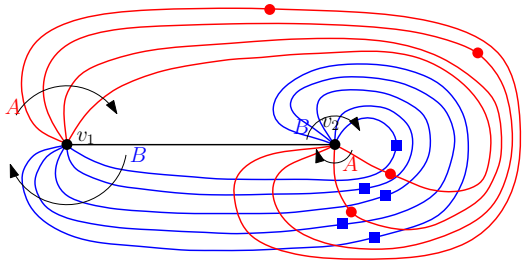


Figure 1: The vertices v_1 and v_2 are as in Theorem 2; arrows around them and colors indicate the partition classes. Only edges incident to v_1 or v_2 are depicted.

The algorithm then runs in two steps. In the first step, we find $O(1)$ possible candidates for v_1 and v_2 as in Theorem 2, using properties on empty triangles in generalized twisted drawings from [3]. In the second step, we check whether one of those candidates fulfills the requirements of Theorem 2. We show that each of the two steps can be done in $O(n^2)$ time.

References

[1] J. Kynčl, Improved enumeration of simple complete topological graphs, *Discrete Comput. Geom.* **50** (2013), 727–770.

[2] O. Aichholzer, A. García, J. Tejel, B. Vogtenhuber, and A. Weinberger, Twisted Ways to Find Plane Structures in Simple Drawings of Complete Graphs, *38th International Symposium on Computational Geometry (SoCG 2022)* (2022), 5:1–5:18.

[3] A. García, J. Tejel, B. Vogtenhuber, and A. Weinberger, Empty Triangles in Generalized Twisted Drawings of K_n, *Graph Drawing and Network Visualization* (2023), 40–48.

*Emails: {oaich,weinberger}@ist.tugraz.at. Supported by the Austrian Science Fund (FWF) grant W1230.

†Email: olaverri@unizar.es. Supported by the Gobierno de Aragón project E41-17R.

‡Email: jtejel@unizar.es. Supported by the he project PID2019-104129GB-I00 / AEI / 10.13039/501100011033 of the Spanish Ministry of Science and Innovation, and by the Gobierno de Aragón project E41-17R.

§Email: bvogt@ist.tugraz.at. Supported by the FWF grant I 3340-N35.

Characterizing rotation systems of generalized twisted drawings via 5-tuples

Oswin Aichholzer*[1], Alfredo García†[2], Javier Tejel‡[2], Birgit Vogtenhuber§[1], and Alexandra Weinberger*[1]

[1]Institute of Software Technology, Graz University of Technology, Austria.
[2]Departamento de Métodos Estadísticos. IUMA. Universidad de Zaragoza. Spain.

Simple drawings are drawings of graphs in which the edges are Jordan arcs and each pair of edges share at most one point (a proper crossing or a common endpoint). The rotation of a vertex is the cyclic order of its incident edges. The rotation system of a simple drawing is the collection of the rotations of all vertices. An abstract rotation system of K_n gives for every vertex v, a cyclic order of the other $n-1$ vertices. Not every abstract rotation system can be realized as a simple drawing, but this realizability can be checked in $O(n^5)$ time [1]. Two simple drawings of K_n have the same crossing edge pairs if and only if they have the same or inverse rotation systems [2]; such drawings are called weakly isomorphic.

A simple drawing D of K_n is generalized twisted if there exists a point O such that each ray emanating from O crosses each edge of D at most once and there exists a ray r emanating from O that crosses all edges of D. We call a rotation system generalized twisted if there exists a generalized twisted drawing with that rotation system. Generalized twisted drawings have been used to improve bounds on plane substructures in (general) simple drawings of complete graphs [3]. Moreover, they are the biggest class for which it is known that each drawing has exactly $2n-4$ empty triangles [4], which is conjectured to be the minimum for all simple drawings. In addition to possibly being useful for further general results, generalized twisted drawings are quite interesting in their own right.

In this talk, we present the following new characterization of generalized twisted drawings via abstract rotation systems.

Theorem 1 *Let R be an abstract rotation system of K_n with $n \geq 7$. Then R is generalized twisted if and only if every rotation sub-system induced by 5 vertices is generalized twisted.*

Up to relabeling, there is one unique rotation system of K_5 that is generalized twisted[1] , and one unique rotation system R of K_6 such that every rotation subsystem of R induced by 5 vertices is generalized twisted, but R is not generalized twisted [3]. In particular, Theorem 1 does not hold for $n = 6$.

To prove Theorem 1, we computationally verify it for $7 \leq n \leq 10$ and then use this as an induction base to prove the statement for general n. We use the following two concepts as our main ingredients. (1) A pair of cells in a drawing D is an *antipodal vi-cell pair*, if both cells have a vertex on their boundary and for any triangle of D, the two cells lie on different sides. A simple drawing D of K_n is weakly isomorphic to a generalized twisted drawing if and only if D has two antipodal vi-cells [3]. We show that any simple drawing D has at most two antipodal vi-cell pairs, and it has exactly two if and only if there is an edge e of D such that e crosses every edge in D not adjacent to e. (2) A vertex-empty triangle xyz is an *empty star triangle at x* if no edge incident to x crosses yz. For every vertex x of a generalized twisted drawing D, there are two empty star triangles at x [4]. We show that D has exactly two pairs of antipodal vi-cells if and only if there is a vertex v such that the two empty star triangles at v are adjacent.

References

[1] J. Kynčl, Simple realizability of complete abstract topological graphs in P, *Discrete Comput. Geom.* **45** (2011), 383–399.

[2] J. Kynčl, Improved enumeration of simple complete topological graphs, *Discrete Comput. Geom.* **50** (2013), 727–770.

[3] O. Aichholzer, A. García, J. Tejel, B. Vogtenhuber, and A. Weinberger, Twisted Ways to Find Plane Structures in Simple Drawings of Complete Graphs, *38th International Symposium on Computational Geometry (SoCG 2022)* (2022), 5:1–5:18.

[4] A. García, J. Tejel, B. Vogtenhuber, and A. Weinberger, Empty Triangles in Generalized Twisted Drawings of K_n, *Graph Drawing and Network Visualization* (2023), 40–48.

*Emails: {oaich,weinberger}@ist.tugraz.at. Supported by the Austrian Science Fund (FWF) grant W1230.

†Email: olaverri@unizar.es. Supported by the Gobierno de Aragón project E41-17R.

‡Email: jtejel@unizar.es. Supported by the he project PID2019-104129GB-I00 / AEI / 10.13039/501100011033 of the Spanish Ministry of Science and Innovation, and by the Gobierno de Aragón project E41-17R.

§Email: bvogt@ist.tugraz.at. Supported by the FWF grant I 3340-N35.

[1]The unique rotation system of K_5 that is 'twisted'.

On the number of drawings of a combinatorial triangulation

Belén Cruces Mateo*[1], Clemens Huemer†[1], and Dolores Lara‡[2]

[1]Universitat Politècnica de Catalunya
[2]Centro de Investigación y de Estudios Avanzados

In 1962, Tutte [5] proved that the number of triangulations, that is maximal planar graphs with a fixed face with vertices a, b, and c, and n additional vertices is $\psi_{n,0} = \frac{2}{n(n+1)}\binom{4n+1}{n-1} = \Theta\left(\frac{1}{n^{5/2}} 9,\overline{481}^n\right)$. See [5] for a precise definition. We call these triangulations *combinatorial* triangulations. Note that in a combinatorial triangulation, the edges need not be straight-line segments. In contrast to combinatorial triangulations, there is no general formula for the number of *geometric* triangulations, which are defined for a given set S of n points in the plane. A geometric triangulation on S is a maximal planar straight-line graph with vertices the set S. Finding the maximum number $tr(n)$ of geometric triangulations, among all sets S of n points in general position in the plane, is a long-standing open problem in Discrete Geometry. The current best bounds are $\Omega(9,08^n) \leq tr(n) \leq O(30^n)$, [2, 3]. In [4] the question was raised if the numbers of combinatorial and geometric triangulations are somehow related? See Fig. 1 for an example that shows the three combinatorial triangulations on five vertices, but only two of them are geometric triangulations on the shown set S of five points. We study the following problem:

Question: *In how many ways can a combinatorial triangulation with n vertices be drawn on a set of n points in the plane?*

Note that any upper bound c^n on this number yields trivially an upper bound for $tr(n)$ of $O((c \cdot 9,\overline{481})^n)$. It turns out to be very difficult to find examples of combinatorial triangulations which can be drawn in many different ways on a given point set S. A first simple bound is shown in the following:

• **A triangulation formed by nested triangles** The set S of n points in the plane in general position has $\frac{n}{3}$ layers of three points as in Fig. 2. The combinatorial triangulation T we consider is the one shown in this figure. We observe that each triangular layer can be rotated to produce a different geometric triangulation of S, while maintaining the combinatorial triangulation T. This yields a lower bound of $\Omega\left(2^{\frac{n}{3}}\right) = \Omega\left(1, 2599^n\right)$ different drawings of T on S.

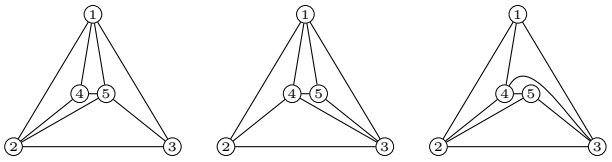


Figure 1: Combinatorial triangulations on 5 vertices.

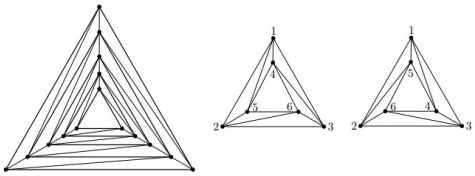


Figure 2: A triangulation formed by nested triangles and a rotation between consecutive layers.

• **A triangulation on the double chain** We improve upon this bound by defining another combinatorial triangulation T recursively, and show a lower bound on the number of drawings of T on the so-called *double chain* point configuration [1].

Theorem 1 *There exists a combinatorial triangulation T and a set S of n points in the plane such that T has at least $\Omega(1, 31^n)$ different drawings on S.*

References

[1] A. García, M. Noy, J. Tejel, Lower bounds on the number of crossing-free subgraphs of K_N, *Computational Geometry* **16** (2000), 211-221.

[2] D. Rutschmann, M. Wettstein, Chains, Koch chains, and point sets with many triangulations, arXiv preprint, 2022, `arXiv:2203.07584`.

[3] M. Sharir, A. Sheffer, Counting triangulations of planar point sets, *The Electronic Journal of Combinatorics* **18** (2011).

[4] M. Sharir, E. Welzl, Random triangulations of planar point sets. *Proc. of the 22nd Annual Symposium on Computational Geometry* (2006) 273–281.

[5] W. T. Tutte, A census of planar triangulations, *Canadian Journal of Mathematics* **14** (1962), 21–38.

*Email: belen.cruces@estudiant.upc.edu
†Email: clemens.huemer@upc.edu. Research supported by PID2019-104129GB-I00/ MCIN/ AEI/ 10.13039/501100011033 and Gen. Cat. 2021 SGR 00266.
‡Email: dlara@cs.cinvestav.mx.

Coverage maps on domains with obstacles

Oriol Balló*[1], Narcís Coll†[2], and Marta Fort‡[2]

[1]Universitat de Girona
[2]Graphics and Imaging Laboratory, Universitat de Girona

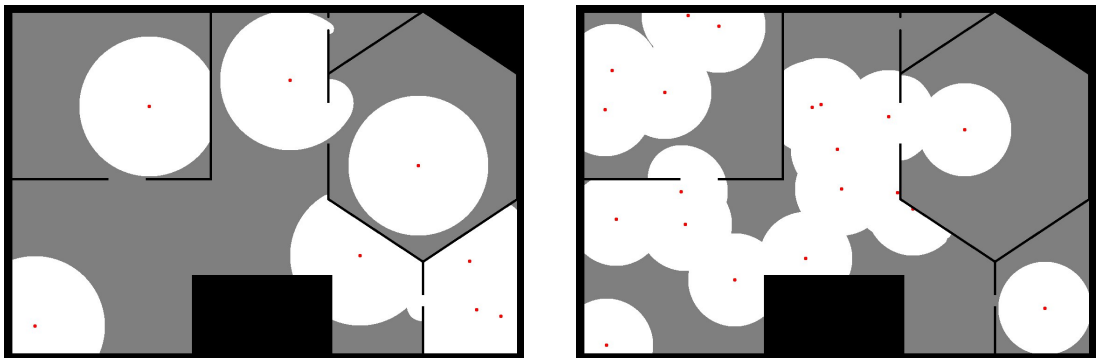


Figure 1: Coverage map of eight facilities (left) and twenty facilities (right)

Abstract

The coverage map of a set of facilities represents, for each point within a domain, whether at least one facility covers it (see Figure 1). That is, we know if at least one facility is at most at a given distance (the facility coverage radius) through the free space of each point of the domain. In this work, we propose a parallel method that runs on the GPU to compute coverage maps over domains given by binary images. The input is a binary image, where each pixel is marked depending on whether it is free space or not, the set of facilities, and their coverage radius. The output is an image where each pixel is marked as covered or uncovered by the set of facilities.

We use a two-step iterative process that combines a quasi-Euclidean distance [1] propagation along free space and an exact Euclidean distance computation (without propagation). We iteratively repeat these steps until no updates occur. During the process, we obtain the distance of each pixel to its nearest site and the pixel-*id* of the last corner (of the current shortest path) by using two CUDA-kernels executed on 2d-grids and 2d-blocks and considering a thread per pixel.

The quasi-Euclidean distance propagation along free space uses a GPU parallel Bellman-Ford algorithm. The used graph (computed on the fly from the binary image) has as vertices the free pixels, and connects a free pixel with its, at most 8, neighboring free pixels. At the beginning of the process, pixels containing sites store a 0 as distance and its pixel-*id* as last-corner-*id*. The rest store ∞ as distance and -1 as last-corner-*id* (uncovered pixel). Throughout the process, in a similar way to the Euclidean distance transformation algorithms [2], each pixel propagates towards itself the paths that reach its free neighbors. If the length of any of these paths is less than the current distance, this distance and the last corner-*id* are updated accordingly.

An inner-block propagation followed by an inter-block propagation expands the current paths through free space according to the quasi-Euclidean distance. A boolean variable and two synchronizing points (per block) stop the inner-block propagation when no updates occur within the block. The inter-block propagation uses a global boolean to keep calling the kernel while updates occur.

The second step computes the exact Euclidean length of the obtained paths. The covered-pixel threads retrieve the path reaching the pixel while determining its Euclidean length. They add the Euclidean distance from the pixel to its last-corner-*id* to the distance of this last corner to the previous one, and so on, until reaching the original site. This two-step iteration leads to exact free-space coverage maps.

Funding

Research supported by grants PID2019-106426RB-C31 and PDC2021-120997-C32 funded both by MCIN/AEI/10.13039/501100011033 and the 2nd one also by European Union NextGenerationEU/PRTR.

References

[1] Montanari, U. (1968). A Method for Obtaining Skeletons Using a Quasi-Euclidean Distance. Journal of the ACM (JACM), 15, 600–624.

[2] Maurer, C. R., Qi, R., and Raghavan, V. (2003). A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(2), 265–270.

*Email: u1962391@campus.udg.edu
†Email: narcis.coll@udg.edu
‡Email: marta.fort@udg.edu

Fault-tolerant resolvability in maximal outerplanar graphs

Carmen Hernando*[1], Montserrat Maureso†[1], Mercè Mora‡[1], and Javier Tejel§[2]

[1]Universitat Politècnica de Catalunya, Barcelona, Spain
[2]Universidad de Zaragoza, Zaragoza, Spain

Resolving sets are useful to distinguish the vertices of a graph. A set S of vertices of a graph G is a *resolving set* if for every pair of vertices u and v of G there is at least one vertex w in S such that the distances from w to u and from w to v are distinct. The *metric dimension* of G, denoted by $\beta(G)$, is the minimum cardinality of a resolving set. These concepts were introduced for general graphs independently by Slater [4] and by Harary and Melter [2], and have since been widely investigated.

Fault-tolerant resolving sets were introduced to distinguish the vertices of a graph even if a vertex fails [3]. A resolving set S of a non-trivial connected graph G is *fault-tolerant* if $S \setminus \{v\}$ is also a resolving set for each $v \in S$. The *fault-tolerant metric dimension* of G, denoted by $\beta'(G)$, is the minimum cardinality of a fault-tolerant resolving set of G. Since $V(G)$ and $V(G) \setminus \{v\}$ are both resolving sets for every vertex v of a graph G of order at least 2, this parameter is well-defined whenever G is a non-trivial graph. Moreover, $\beta'(G) \leq n$ and, obviously, $\beta'(G) \geq \beta(G) + 1$.

A planar graph is *outerplanar* if it admits a plane embedding such that all the vertices belong to the unbounded face. A *maximal outerplanar* graph is an outerplanar graph such that the addition of an edge results in a non-outerplanar graph. Maximal outerplanar graphs can be viewed as triangulated polygons.

This ongoing work is devoted to studying the fault-tolerant resolvability for maximal outerplanar graphs. Our first goal is to prove lower and upper bounds on the fault-tolerant metric dimension. Hence, the study of resolving sets and the metric dimension of maximal outerplanar graphs will be useful for our purpose. In [1], the authors prove that the metric dimension of a maximal outerplanar graph of order n, $n \geq 5$, is at most $\lceil \frac{2n}{5} \rceil$. Moreover, this bound is tight and is attained for some fan graphs, a special family of maximal outerplanar graphs.

It is easy to see that $\beta'(G) \geq 3$, if G is a maximal outerplanar graph. Furthermore, we show that only two maximal outerplanar graphs, having orders 3 and 6, attain this lower bound. For maximal outerplanar graphs of order at least 7, the lower bound for $\beta'(G)$ is 4 and there is an infinite family of maximal outerplanar graphs such that $\beta(G) = 2$ attaining this bound.

Regarding to the upper bound, we conjecture that $\beta'(G) \leq \lceil \frac{n}{2} \rceil$ for a maximal outerplanar graph G of order n, $n \geq 7$. At the moment, we have proved that fan graphs attain this upper bound. Moreover, fan graphs of even order have only one fault-tolerant resolving set of minimum size, concretely, the set formed by alternating vertices of the unbounded face and not containing the vertex of degree $n - 1$ (see Figure 1).

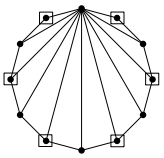


Figure 1: The only fault-tolerant resolving set of size 6 of the fan of order 12 is formed by the squared vertices.

References

[1] M. Claverol, A. García, G. Hernández, C. Hernando, M. Maureso, M. Mora and J. Tejel, Metric dimension of maximal outerplanar graphs, *Bulletin of the Malaysian Mathematical Sciences Society* **44(4)** (2021), 2603–2630.

[2] F. Harary, R.A. Melter, On the metric dimension of a graph, *Ars Combinatoria* **2** (1976), 191–195.

[3] C. Hernando, M. Mora, P. Slater and D.R. Wood, Fault-tolerant metric dimension, in *Ramanujan Mathematical Society*, Lecture Note Series, **5** (2008) 81–85.

[4] P. J. Slater, Leaves of trees, *Congressus Numerantium* **14** (1975), 549–559.

*Email: carmen.hernando@upc.edu. Research supported by PID2019-104129GB-I00/MCIN/AEI/10.13039/501100011033 of the Spanish Ministry of Science and Innovation and Gen.Cat. DGR2021-SGR-00266.

†Email: montserrat.maureso@upc.edu. Research supported by Gen.Cat. DGR2021-SGR-00266.

‡Email: merce.mora@upc.edu. Research supported by PID2019-104129GB-I00/MCIN/AEI/10.13039/501100011033 of the Spanish Ministry of Science and Innovation and Gen.Cat. DGR2021-SGR-00266.

§Email: jtejel@unizar.es. Research supported by PID2019-104129GB-I00/MCIN/AEI/10.13039/501100011033 of the Spanish Ministry of Science and Innovation and Gobierno de Aragón project E41-23R.

Augmenting plane geometric graphs to meet degree parity constraints

Aleksander Bjørn Grodt Christiansen*[1], Linda Kleist†[2], Irene Parada‡[3], and Eva Rotenberg§[1]

[1]Technical University of Denmark, Denmark
[2]TU Braunschweig, Germany
[3]Universitat Politècnica de Catalunya, Spain

A fundamental class of problems in graph theory and graph drawing concerns augmenting existing graphs to achieve some desired properties. In this talk, we approach the natural question of augmenting plane geometric graphs to meet degree constraints.

A *geometric graph* $G = (V, E)$ is a graph drawn in the plane such that its vertex set V is a point set in general position (no three points are collinear) and its edge set E is a set of straight-line segments between those points. A geometric graph is *plane* if no two of its edges cross and it is *convex* if its vertices are in convex position. The *visibility graph* of a plane geometric graph G, denoted by $\mathrm{Vis}(G)$, is a geometric graph that has V as its vertex set and two vertices u and v share an edge in $\mathrm{Vis}(G)$ if and only if $uv \notin E$ and uv does not cross any edge in E (so G can be augmented by uv); see Figure 1 (left) for an example.

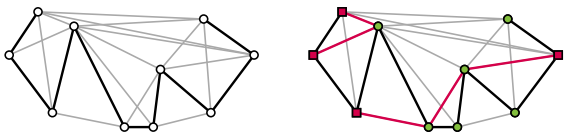


Figure 1: Left: A plane geometric graph G (in black) and its visibility graph $\mathrm{Vis}(G)$ (in gray). Right: A solution set for G and the unhappy vertices (red squares).

Given a plane geometric graph G, the problem we study is to augment it with straight-line edges such that the result is a plane geometric graph in which constraints concerning the degrees of the vertices are met. Even in the simplest version of this problem, where the degree constraints are modulo two, even or odd degree, the problem is NP-hard for general graphs [1], and conjectured to be NP-hard even for trees [1]. In this talk, we answer the question about the tractability for geometric paths; see Table 1.

Table 1: Summary of results on augmenting a plane geometric graph $G = (V, E)$ to meet parity constraints.

Plane geometric G	Any R / $R = V$ / Eulerian
General	NP-hard [1]
Convex	P (Theorem 1)
Paths	P (Theorem 2)
Trees	?

The degree parity constraints can be interpreted as a set of *unhappy* vertices R that would like to change the parity of their degree. We refer to vertices that are not unhappy as *happy*. A subgraph H of $\mathrm{Vis}(G)$ is called a *solution set* for (G, R) if H is crossing-free and the vertices that have odd degree in H are exactly those in R; see Figure 1 (right). Thus, the problem can be reformulated as deciding the existence of a solution set for (G, R). Note that by the handshaking lemma, a solution set can only exist if $|R|$ is even.

For convex plane geometric graph we obtain a surprisingly simple efficient algorithm:

Theorem 1 *Let $G = (V, E)$ be a convex plane geometric graph, and let $R \subseteq V$ be the set of unhappy vertices. There exists a linear-time algorithm to decide whether (G, R) admits a solution set.*

Using this result, we obtain a polynomial-time algorithm for plane geometric paths, solving an open problem by Catana et al. [1].

Theorem 2 *Let $P = (V, E)$ be a plane geometric path and let $R \subseteq V$ with $|R|$ even. There exists an algorithm to decide whether (P, R) admits a solution set in $O(|V| \log |V|)$ time.*

Finally, we characterize those paths that admit a positive answer for any even set R of unhappy vertices.

References

[1] J. C. Catana, A. G. Olaverri, J. Tejel, and J. Urrutia. Plane augmentation of plane graphs to meet parity constraints. *Appl. Math. Comput.*, 386:125513, 2020.

*Email:abgch@dtu.dk. Supported by the VILLUM Foundation grant 37507 "Efficient Recomputations for Changeful Problems"

†Email:kleist@ibr.cs.tu-bs.de.

‡Email:irene.parada@upc.edu. Supported by Independent Research Fund Denmark grant 2020-2023 (9131-00044B) "Dynamic Network Analysis" and by the Margarita Salas Fellowship funded by the Ministry of Universities of Spain and the European Union (NextGenerationEU).

§Email:erot@dtu.dk. Partially supported by the VILLUM Foundation grant 37507 "Efficient Recomputations for Changeful Problems" and the Independent Research Fund Denmark grant 2020-2023 (9131-00044B) "Dynamic Network Analysis".

Computing the (α, k)-hull for points in convex position

Luis H. Herrera*[1], Pablo Pérez-Lantero†[2], and Carlos Seara‡[3]

[1]Dept. de Ingeniería Informática, Universidad Tecnológica Metropolitana (UTEM), Chile
[2]Dept. de Matemática y Ciencia de la Computación, Universidad de Santiago de Chile (USACH)
[3]Dept. de Matemàtiques, Universitat Politècnica de Catalunya (UPC), Spain

Abstract

We present an efficient $O(n \log n)$-time and $O(n)$-space algorithm for computing the (α, k)-hull of a set P of n points in convex position in the plane.

1 Introduction

Let P be a set of $n \geq 3$ points in convex position in the plane, $\alpha \in (0, \pi]$ an angle, and $k \in \mathbb{N}$, $1 \leq k \leq \lfloor n/2 \rfloor$. The (α, k)-hull of the set P is the (possible empty) curvilinear region defined by the intersection of all α-halfplanes that contain at least $n - k + 1$ points of P, where an α-halfplane is the complement of an open wedge of aperture angle α. That is, the (α, k)-hull of P is the locus of the points u such that any wedge with apex u and aperture angle α contains at least k points of P. See Claverol et al. [1], where an $O(n^2 \log n)$-time algorithm was presented for computing this hull.

The (α, k)-hull of a point set has tentative real applications. Namely, suppose that P represents a set of key points of a terrain, and we need to install a surveillance camera which has an angle of vision equal to α, rotates constantly, and at every moment should watch at least k of the key points. Hence, the (α, k)-hull of P is the locus of the possible locations in the terrain to install the rotating camera.

2 The algorithm

Each $k + 1$ consecutive vertices of $conv(P)$ define an open disk D such that all points of $conv(P) \setminus D$ see these vertices with angle at least α. We have n disks in total, and it can be proved that $conv(P) \setminus \mathcal{D}$ is precisely the (α, k)-hull of P, where $\mathcal{D}$ is the union of all disks D.

The contour of $\mathcal{D}$ consists of $O(n)$ circular arcs [2], and by using the algorithm of Imai et al. [2] based on a Voronoi diagram for disks, they can be computed in $O(n \log n)$ time. The interior contour of $\mathcal{D}$, bounding the holes of $\mathcal{D}$, defines the (connected components of) the (α, k)-hull of P. See Figure 1 for a couple of examples of the (α, k)-hull of P. Thus, we have the following result.

Theorem 1 *Given $\alpha \in (0, \pi]$ and $1 \leq k \leq \lfloor n/2 \rfloor$, the (α, k)-hull of a set P of n points in convex position has $O(n)$ complexity, and can be computed in efficient $O(n \log n)$ time and $O(n)$ space.*

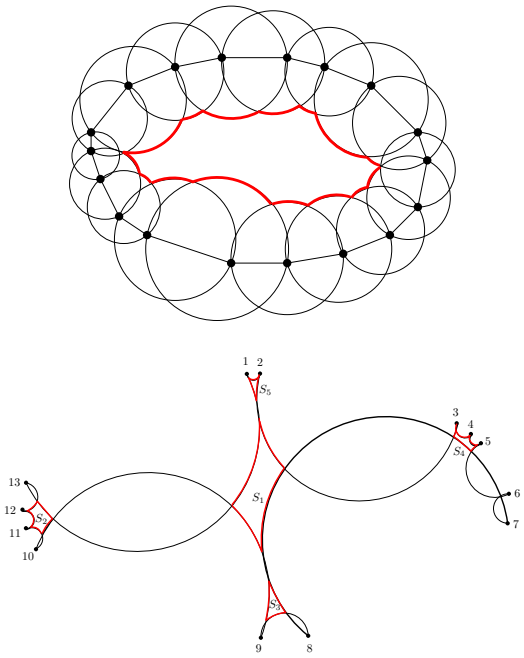


Figure 1: (Top) The $(\pi/2, 2)$-hull formed by a connected region. (Bottom) The $(\pi/2, 1)$-hull formed by 5 connected regions.

References

[1] M. Claverol, L. H. Herrera, P. Pérez-Lantero, and C. Seara. On (α, k)-sets and (α, k)-hulls in the plane. *XIX EGC*, (2021), pp. 37–40.

[2] H. Imai, M. Iri, and K. Murota. Voronoi diagram in the Laguerre geometry and its applications. *SIAM J. Comput.*, **14**, (1985), pp. 93–105.

*Email: luis.herrerab@utem.cl
†Email: pablo.perez.l@usach.cl.
‡Email: carlos.seara@upc.edu.

On the Realizability of Planar and Spherical Occlusion Diagrams

Kimberly Kokado*[1] and Csaba D. Tóth†[1]

[1]California State University Northridge, Los Angeles, CA, USA

Abstract

Spherical occlusion diagrams (SOD) were introduced as an axiomatic framework to analyze the visibility maps of points in the interior of a nonconvex polyhedron from which no vertex is visible. Planar occlusion diagrams (POD), corresponding to a viewpoint at infinity, can be defined analogously in terms of orthogonal projections of a lower envelope of polyhedra. We have recently constructed PODs and SODs that are not realizable as visibility maps. Here we show that every axis-aligned POD is realizable and follow up with open problems.

Introduction. The classical *art gallery problem* asks for the minimum number of point guards that can jointly see all points in a nonconvex polyhedron $\mathcal{P}$ in Euclidean space, where points s and t *see* each other if the line segment st does not cross any face of $\mathcal{P}$. It is well known that guards stationed at the vertices of $\mathcal{P}$ do not always suffice in $\mathbb{R}^3$, as some points $s \in \mathbb{R}^3$ in the interior of a polyhedron $\mathcal{P}$ may not see any of the vertices [4, Sec. 10.2]. Viglietta [7] recently introduced *spherical occlusion diagrams* (*SOD*, for short) to analyze the visibility map $V_{\mathcal{P}}(s)$ of such a point s with respect to $\mathcal{P}$. An SOD is defined so that it satisfies key properties of visibility maps. In particular, if no vertices of a polyhedron $\mathcal{P}$ are visible from a viewpoint s, then the visibility map $V_{\mathcal{P}}(s)$ is an SOD. Viglietta conjectured that the converse also holds, that is, every SOD is the visibility map $V_{\mathcal{P}}(s)$ for some point s and polyhedron $\mathcal{P}$ in $\mathbb{R}^3$. We have recently disproved this conjecture, by constructing an SOD that is not realizable as a visibility map in $\mathbb{R}^3$ [3].

Related work. Our results show that SODs are not always visibility maps. Nevertheless, SODs have already been used in 3-dimensional visibility problems: Cano et al. [2] proved that every polyhedron $\mathcal{P}$ in $\mathbb{R}^3$ can be guarded by at most $\frac{5}{6}$ of its edges; moreover, when $\mathcal{P}$ is homeomorphic to a ball and all its faces are triangles, it can be guarded by at most $\frac{29}{36}$ of its edges. Tóth et al. [6] proved that every point that does not see any vertex of a polyhedron $\mathcal{P}$ must see at least 8 edges of $\mathcal{P}$, and this bound is tight.

The realizability of visibility maps has been previously studied for lines. A *weaving pattern* is a simple arrangement of n lines in $\mathbb{R}^2$ together with a binary relation between intersecting lines; a weaving pattern is *realizable* if it is the orthogonal projection of an arrangement of disjoint lines in $\mathbb{R}^3$ such that the above-below relation between lines matches the given binary relation between their orthogonal projections. Pach et al. [5] showed that almost all weaving patterns of n lines are nonrealizable for sufficiently large n. Basu et al. [1] generalized the result to arrangements of semialgebraic curves.

Outlook. We have shown that spherical occlusion diagrams (SODs) are not equivalent to visibility maps in 3-space. Our result raises several open problems: Is there a simple (axiomatic) characterization of visibility maps? Can one decide efficiently whether a given SOD is a visibility map? If so, can one find a realization efficiently? What is the maximum (combinatorial, topological, or bit) complexity of the realization space for an SOD with n arcs for a given positive integer n?

References

[1] S. Basu, R. Dhandapani, and R. Pollack, On the realizable weaving patterns of polynomial curves in $\mathbb{R}^3$. in *Proc. 12th Sympos. Graph Drawing (GD)*, volume 3383 of *LNCS*, Springer, 2004, pp. 36–42.

[2] J. Cano, C. D. Tóth, J. Urrutia, and G. Viglietta, Edge guards for polyhedra in three-space, *Comput. Geom.* **104** (2022), article 101859.

[3] K. Kokado and C. D. Tóth, Nonrealizable planar and spherical occlusion diagrams, in *Proc. Japanese Conf. Discrete & Comput. Geom. Graphs, and Games (JCDCG^3)*, Tokyo, 2022, pp. 60–61.

[4] J. O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, 1987.

[5] J. Pach, R. Pollack, and E. Welzl, Weaving patterns of lines and line segments in space, *Algorithmica* **9(6)** (1993), 561–571.

[6] C. D. Tóth, J. Urrutia, and G. Viglietta, Minimizing visible edges in polyhedra, in *Proc. 23rd Thailand-Japan Conf. Discrete & Comput. Geom. Graphs, and Games (TJCDCG^3)*, Chiang Mai, 2021, pp. 70–71.

[7] G. Viglietta, A theory of spherical diagrams, in *Proc. 34th Canadian Conf. Comput. Geom. (CCCG)*, Toronto, ON, 2022, pp. 306-313.

*Email: kimberly.kokado.43@my.csun.edu
†Email: csaba.toth@csun.edu

Maintaining low congestion potential among moving entities using minimal query frequency *

David Kirkpatrick†

Dept. of Computer Science, University of British Columbia, Vancouver, Canada

Abstract

Consider a collection of entities moving continuously with bounded speed, but otherwise unpredictably, in some low-dimensional space. Two such entities encroach upon one another at a fixed time if their separation is less than some specified threshold. Encroachment, of concern in many settings such as collision avoidance, may be unavoidable. However, the associated difficulties are compounded if there is uncertainty about the precise location of entities, giving rise to potential encroachment and, more generally, potential congestion within the full collection.

We consider a model in which individual entities can be queried for their current location (at some fixed cost) and the uncertainty regions associated with an entity grows in proportion to the time since that entity was last queried. The goal is to maintain low potential congestion, measured in terms of the (dynamic) intersection graph of uncertainty regions, using the lowest possible query cost. Previous work [EKLS2013, EKLS2014, EKLS2016, BEK2019], in the same uncertainty model, addressed the problem of minimizing the congestion potential *of point entities* using location queries of some bounded frequency. It was shown that it is possible to design a query scheme that is $O(1)$-competitive, in terms of worst-case congestion potential, with other query schemes (even those that correctly guess the trajectories of all entities), subject to the same bound on query frequency.

In this talk we outline recent results that address a more general problem with the dual optimization objective: minimizing the query frequency, measured in terms of the minimum spacing between queries (query granularity), over any fixed time interval, while guaranteeing a fixed bound on congestion potential *of entities with positive extent*. This complementary objective necessitates quite different algorithms and analyses. Nevertheless, our results parallel those of the earlier papers, specifically tight competitive bounds on required query frequency, with a few surprising differences.

Applied to collision avoidance, our results show that it is possible to maintain, at all times, a certificate for each entity e_i that identifies some pre-specified number of other entities that could potentially encroach upon e_i (those warranting more careful local monitoring), using competitively-optimal query frequency An additional application, considered in earlier work, concerns entities that are mobile transmission sources, with associated broadcast ranges, where the goal is to minimize the number of broadcast channels so as to eliminate potential transmission interference. In this case, using competitively-optimal frequency, our query strategy maintains a fixed bound on the number of broadcast channels, an objective that seems to be at least as well motivated as optimizing the number of channels for a fixed query frequency (the objective in earlier work).

*Based on joint work with William Evans (see arxiv:2205.09243). Research supported in part by Discovery Grants from the Natural Sciences and Engineering Research Council of Canada.

†Email: kirk@cs.ubc.ca.

[EKLS2013] W. Evans, D. Kirkpatrick, M. Löffler, and F. Staals. Competitive query strategies for minimizing the ply of the potential locations of moving points. *29th Annual Symposium on Computational Geometry*, SoCG '13, pages 155-164, 2013.

[EKLS2014] W. Evans, D. Kirkpatrick, M. Löffler, and F. Staals. Query strategies for minimizing the ply of the potential locations of entities moving with different speeds. *Abstr. 30th European Workshop on Computational Geometry*, 2014.

[EKLS2016] W. Evans, D. Kirkpatrick, M. Löffler, and F. Staals. Minimizing co-location potential of moving entities. *SIAM Journal on Computing*, 45, 1870-1893, 2016.

[BEK2019] D. Busto, W. Evans and D. Kirkpatrick. Minimizing Interference Potential Among Moving Entities. *Proc. ACM-SIAM Symposium on Discrete Algorithms*, SODA'19, pages 2400-2418, 2019.