ISSN 2186-7437

NII Shonan Meeting Report

No. 2017-16

Geometric Graphs: Theory and Applications

Naoki Katoh
Dept. of Informatics, Kwansei Gakuin University, Japan
Hee-Kap Ahn
Dept. of Computer Science & Engineering, POSTECH, Korea
Subhas C Nandy
Advanced Computing and Microelectronics Unit,
Indian Statistical Institute, India

October 30 – November 2, 2017

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-Ku, Tokyo, Japan

Geometric Graphs: Theory and Applications

Organizers:
Naoki Katoh (Kwansei Gakuin University, Japan)
Hee-Kap Ahn (POSTECH, Republic of Korea)
Subhas C Nandy (Indian Statistical Institute, India)

October 30 – November 2, 2017

This is the report on the NII Shonan Meeting on Geometric Graphs: Theory and Applications, which was held during October 30-November 2, 2017 at Shonan International Village, Japan. The meeting number is 2017-16.

The main goal of the meeting is to facilitate the exchange of tools, techniques, questions, and ideas that will lead to a better understanding of geometric graphs that arise in different applications. To achieve the goal, 26 prominent researchers, including quite a few young researchers, over the world who have been working on these topics participated in the meeting and shared their ideas for solving problems arising in geometric graphs in theory and applications.

The focus of the meeting is mainly on sharing open problems and exploring possible approaches to solve them. There were 4 plenary talks, each for an hour, by eminent researchers, and 6 regular talks, each for a half an hour by young researchers in these fields. Each talk started with an introduction to the research topic and presented the latest results on the topic. It ended with a few related open problems so that participants could discuss on them with the speakers and other participants afterwards. Two open problem sessions were arranged and the invited participants of the meeting posed open problems during the sessions. The talks and open problem sessions were arranged in the morning and in the afternoon we had discussion sessions for those open problems. During the workshop, we focused on the following research topics.

Combinatorial questions on geometric graphs. The study of geometric graphs is a classical topic in computational geometry. Many practical problems have natural models via geometric objects. For example, map labeling, problems in wireless and sensor networks and VLSI physical design, database queries, etc. Here the vertices of the graph are mapped to some geometric objects and an edge between a pair of vertices indicates the existence of some specific relation depending on the problem. Here, the objectives are three fold, namely

(i) characterizing the graph to have some desired embedding in $\mathbb{R}^d$, for some chosen d, depending on the problem specification,

(ii) combinatorial questions regarding various properties of the graph, for example, minimum number of layers required to draw the graph in planar way, coloring vertices/edges with minimum number of colors to avoid conflict among the objects/paths, showing the relationship among different parameters of the graph, namely coloring number, clique size, etc,

(iii) algorithmic questions regarding polynomial time computation of some parameters of the graph, or approximating the parameters to a desired accuracy in polynomial time, etc.

For an example, we may refer to the following problems on a geometric complete graph G with n points:

(i) maximum number of planar spanning trees of G that can be packed in G,

(ii) maximum number of plane perfect matching of the vertices in G that can be packed in G,

(iii) maximum number of plane Hamiltonian paths in G that can be packed in G. All these questions are still unanswered.

Random Geometric Graphs. In recent years, a lot of progress has been made in the study of geometric intersection graphs, but many important combinatorial and algorithmic questions are still open. Recently, in the network science community the interest is growing towards the geometrical characterization of real network. A large real network is usually considered as a random graph (Erdos and Renyi (1960)). It is assumed that each pair of vertices in the network is connected with probability p, and is independent of the other edges of the graph. Here, the desired problems are modeled by random walk on geometric graphs for the average case analysis of the performance of the corresponding network. On the other hand, random geometric graph was first formally suggested by Gilbert (1961), whose vertices are random points in Euclidean plane and an edge between a pair of points (nodes) exists if their distance is less than a given constant r. These classes of graphs are known to have numerous applications as a model for studying communication primitives (broadcasting, routing, etc.) and topology control (connectivity, coverage, etc.) in idealized wireless sensor networks as well as extensive utility in theoretical computer science and many fields of the mathematical sciences, namely routing problems in the internet, data mining, community detection, to name a few.

Many important problems are still unsolved for random geometric graphs. For example, given a geometric k-NN (k nearest neighbor) graph G with n random points in $\mathbb{R}^2$ and each point is connected with its k nearest neighbors for a given value of k, the lower and upper bounds on k for the graph G to be connected are $0.3043 \log n$ and $0.5139 \log n$ respectively [Balister, Bollobás, Sarkar and Walters 2006]. It will be interesting to tighten the bound on k for the connectivity of G. Also for a directed complete graph G with the edge cost of each directed edge $(\overrightarrow{p_i, p_j})$ is α if p_j is the α-th nearest neighbor of p_i, computing (i) the expected cost of minimum spanning tree, (ii) expected value of the spanning ratio between the farthest pair of points, etc. will be interesting problems to study.

In specific, we studied various structural and combinatorial properties of Delaunay graphs, unit disk graphs and visibility graphs during the meeting. Delaunay graphs guarantee a constant spanning ratio or stretch factor, which is the ratio of the shortest paths between two points in a geometric graph and in its spanning subgraph. The geometric separator for unit disks in the plane is a closed curve that separates the unit disks into two disjoint sets, one consisting of disks lying inside of the curve and one consisting of disks lying outside of the curve, while those disks intersected by the curve belong to the separator. We are interested in the existence of a balanced separator for unit disk graphs and efficient algorithms for computing such a separator. Based on the properties, we discussed on efficient algorithms for solving problems on those graphs, including optimal routing in Delaunay and visibility graphs, balanced separators for unit disk graphs, and graph recognition from weak embeddings. In addition, there were discussions on adaptive point location data structures, rigidity of triangulations, and Kakeya-related problems. The details of the problems and discussion are given in the following sections.

Finally, we, the organizers and participants of the meeting, would like express our sincere gratitude to National Institute of Information (NII) for providing us the excellent facility of Shonan International Village and supporting the meeting in various ways.

Overview of Talks

We had four plenary talks by prominent scholars and 6 regular talks by young researchers who have been working on geometric graphs and have presented a number of research results in theory and applications.

This is the list of the plenary talks.

- On spanning properties of various Delaunay graphs *by Prosenjit Bose.*
- The Reverse Kakeya Problem *by Otfried Cheong.*
- Adaptive Planar Point Location *by Siu-Wing Cheng.*
- Recognizing Weak Embeddings of Graphs *by Csaba D. Toth.*

This is the list of regular talks.

- Balanced Line Separators of Unit Disk Graphs *by Yoshio Okamoto.*
- Shortcuts for the Circle *by Sang Won Bae*
- Geodesic Voronoi Diagrams in a Simple Polygon *by Eunjin Oh.*
- Global Rigidity of Triangulations with Braces *by Shin-ichi Tanigawa.*
- Open Problems on Optimal Patrolling *by Akitoshi Kawamura.*
- Routing on the Visibility Graph *by André van Renssen.*

On Spanning Properties of Various Delaunay Graphs

Speaker: Prosenjit Bose, Carleton University, Ottawa, Canada

A geometric graph G is a graph whose vertices are points in the plane and whose edges are line segments weighted by the Euclidean distance between their endpoints. In this setting, a t-spanner of G is a connected spanning subgraph G' with the property that for every pair of vertices x, y, the shortest path from x to y in G' has weight at most $t \geqslant 1$ times the shortest path from x to y in G. The parameter t is commonly referred to as the spanning ratio or the stretch factor. Among the many beautiful properties that Delaunay graphs possess, a constant spanning ratio is one of them. We provide an overview of various results concerning the spanning ratio among other properties of different types of Delaunay graphs and their subgraphs.

The Reverse Kakeya Problem

Speaker: Otfried Cheong (a joint work with Sang Won Bae, Sergio Cabello, Fabian Stehn, Yoonsung Choi, and Sang-duk Yoon)

In 1917, Soichi Kakeya posed the following problem: What is the minimum area region in the plane in which a needle of length 1 can be turned through 360° continuously, and return to its initial position [6]? For convex regions, the problem was solved by Pál [8], who showed that the solution is the equilateral triangle of height one, having area $1/\sqrt{3}$. For the general case, Besicovitch gave the surprising answer that one could rotate a needle using an arbitrary small area [2, 3]. Kakeya-type problems have received considerable attention in the literature, as there are strong connections to problems in number theory [4], geometric combinatorics [10], arithmetic combinatorics [7], oscillatory integrals, and the analysis of dispersive and wave equations [9].

If one generalizes the problem for convex regions slightly, and asks for the smallest convex region in which a given convex shape P can be turned through 360°, the problem seems to be still wide open: the answer is not even known when P is an equilateral triangle or a square.

In this paper, we consider a "reverse" version of the problem, where the convex compact shapes P and Q are already given, and we ask: how large can we make P such that it can turn through 360° inside Q?

Let's assume that the origin is in the interior of P, and denote P rotated by θ around the origin by P_θ. Being able to turn P inside Q obviously implies that P_θ can be translated into Q for any orientation θ. Is this condition also sufficient?

In his 1921 paper solving the convex case of the Kakeya problem, Pál [8] conjectured that this is the case. Intriguingly, the paper contains a footnote added during the proof stage, stating that Harald Bohr had proven this conjecture. Unfortunately, this proof seems to have never been published, and we have not been able to find another proof in the literature. We also do not know how exactly Pál defined "turning" in this context: Is the angle changing in a strictly monotone way, or merely monotonically?

We therefore prove a stronger version of Pál's conjecture: For a given angle θ, let $\lambda(\theta)$ be the largest scaling factor such that $P^*_\theta := \lambda(\theta)P_\theta$ can be translated into Q. We prove that the function $\theta \mapsto \lambda(\theta)$ is continuous.

Our main results are the following two theorems:

Theorem 1. *Given a convex m-gon P and a convex n-gon Q, we can in time $O(mn^2 \log n)$ compute the continuous function $\lambda(\theta)$ and a continuous function $t : [0, 2\pi] \mapsto \mathbb{R}^2$ such that $P^*_\theta + t(\theta) \subseteq Q$.*

Theorem 2. *For any compact convex shapes P and Q, there exists a continuous function $t : [0, 2\pi] \mapsto \mathbb{R}^2$ such that $P^*_\theta + t(\theta) \subseteq Q$.*

In other words, P can be rotated, while continuously scaling and translating it to maintain the largest possible size that will fit inside Q at that orientation.

The polygonal version, Theorem 1, follows quite easily from a construction by Agarwal et al. [1]. The proof for general convex shapes turns out to be much harder—we have not been able to find a limit argument to obtain it from the polygonal version.

Theorem 2 immediately implies Pál's conjecture: If P_θ can be translated into Q for any θ, then $\lambda(\theta) \geqslant 1$ always, and so $P_\theta + t(\theta)$ is a continuous motion that turns P inside Q.

Returning to the polygonal case, where P is a convex m-gon and Q is a convex n-gon, the work by Agarwal et al. [1] implies that the functions $\theta \mapsto \lambda(\theta)$ and $\theta \mapsto t(\theta)$ have complexity $O(mn^2)$, and gives an algorithm to compute a description of these functions in time $O(mn^2 \log n)$. Given these functions, one can answer questions such as:

- What is the largest similar copy of P inscribed into Q?
- What is the largest similar copy of P that can be turned through 360° inside Q?

The answer to the first question is the maximum of $\lambda(\theta)$, and was the original goal of Agarwal et al.'s paper. The answer to the second question is given by the *minimum* of $\lambda(\theta)$.

Agarwal et al. give a construction of a convex m-gon P and a convex n-gon Q such that there are $\Theta(mn^2)$ placements of similar copies of P inscribed into Q and realizing distinct sets of vertex-edge contacts. However, this is not a lower bound on the complexity of the functions $\lambda(\theta)$ and $t(\theta)$, since not all these placements are maximal.

We prove the following lower bounds on the complexity of these functions:

Theorem 3. *For any n there is a convex n-gon Q such that there are $\Theta(n^2)$ maximal placements of an equilateral triangle in Q.*
For any n and m there is a convex n-gon Q and a convex m-gon P such that there are $\Theta(mn^2)$ maximal placements of P in Q (see Figure 1).

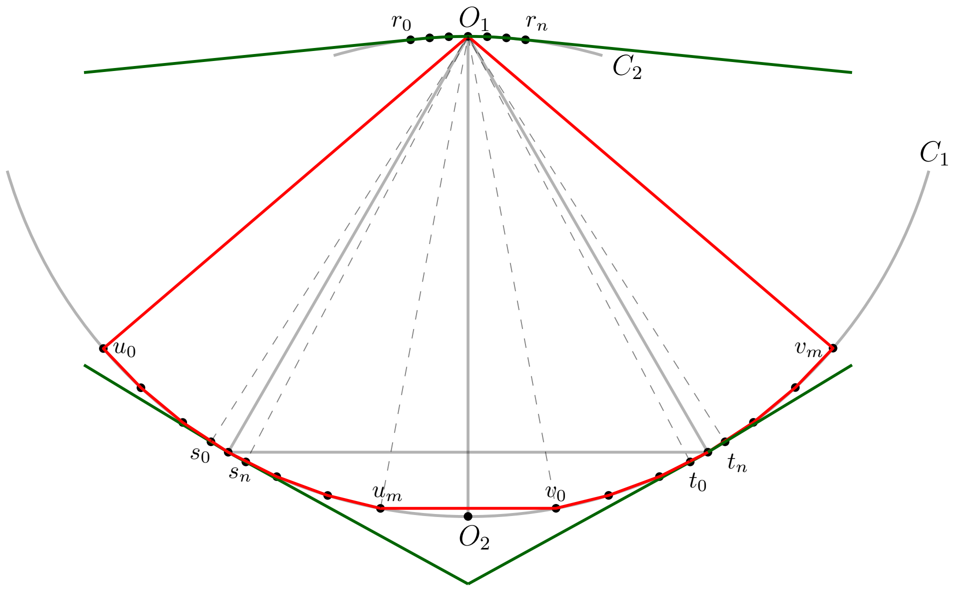

Figure 1: Construction of P (in red) and Q (in green).

These bounds imply that it may be difficult to improve on Agarwal et al.'s algorithm, or on the quadratic-time algorithm for finding the largest equilateral triangle inscribed to a convex polygon by DePano et al. [5].

Finally, we consider the following problem: Given a triangle Q, find three points u, v, w, one on each edge of Q, such that the diameter of the set $\{u, v, w\}$ is minimized. It turns out that this is closely related to our problem.

Theorem 4. *For a given triangle Q, three points u, v, w, one on each edge of Q, such that the diameter of the set $\{u, v, w\}$ is minimized satisfy the following conditions: (a) If the largest angle α of Q is at least $120°$, than u is the intersection of the angular bisector of α with the edge a, while v and w are the orthogonal projections of u onto b and c. (b) If no angle of Q is larger than $120°$, then u, v, w are given by the largest equilateral triangle that can be fully rotated inside Q.*

References

[1] Pankaj K. Agarwal, Nina Amenta, and Micha Sharir. Largest placement of one convex polygon inside another. *Discrete & Computational Geometry*, 19(1):95–104, 1998.

[2] A. S. Besicovitch. Sur deux questions de l'intégrabilité. *Journal de la Société des Math. et de Phys.*, II, 1920.

[3] A. S. Besicovitch. On Kakeya's problem and a similar one. *Math. Zeitschrift*, 27:312–320, 1928.

[4] J. Bourgain. Harmonic analysis and combinatorics: How much may they contribute to each other? In V. I. Arnold, M. Atiyah, P. Lax, and B. Mazur, editors, *Mathematics: Frontiers and Perspectives*, pages 13–32. American Math. Society, 2000.

[5] A. DePano, Yan Ke, and J. O'Rourke. Finding largest inscribed equilateral triangles and squares. In *Proc. 25th Allerton Conf. Commun. Control Comput.* 1987.

[6] S. Kakeya. Some problems on maxima and minima regarding ovals. *The Science Report of the Tohoku Imperial University, Series 1, Mathematics, Physics, Chemistry*, 6:71–88, 1917.

[7] I. Laba. From harmonic analysis to arithmetic combinatorics. *Bulletin (New Series) of the AMS*, 45(1):77–115, 2008.

[8] G. Pál. Ein Minimumproblem für Ovale. *Math. Ann.*, 83:311–319, 1921.

[9] T. Tao. From rotating needles to stability of waves: Emerging connections between combinatorics, analysis and PDE. *Notices of the AMS*, 48(3):297–303, 2001.

[10] T. Wolff. Recent work connected with the Kakeya problem. In H. Rossi, editor, *Prospects in Mathematics*. American Math. Sociaty, 1999.

Adaptive Planar Point Location

Speaker: Siu-Wing Cheng

We present a self-adjusting point location structure for convex and connected subdivisions. Let n be the number of vertices. For a convex subdivision S, our structure uses $O(n)$ space and processes any online query sequence Q in $O(n + OPT)$ time, where OPT is the minimum time required by any linear decision tree for answering point location queries in S to process Q. The $O(n + OPT)$ time bound includes the preprocessing time. Our result is a two-dimensional analog of the static optimality property of splay trees. We will also discuss extension to connected subdivisions.

Recognizing Weak Embeddings of Graphs

Speaker: Csaba D. Toth
Department of Mathematics, California State University Northridge, Los Angeles, CA, USA.
(a joint work with Hugo A. Akitaya and Radoslav Fulek.)

Given a graph G and a 2-dimensional manifold M, one can decide in linear time whether G can be embedded into M [9], although finding the smallest genus of a surface into which G embeds is NP-hard [11]. An **embedding** $\psi : G \to M$ is a continuous piecewise linear injective map where the graph G is considered as a 1-dimensional simplicial complex. Equivalently, an embedding maps the vertices into distinct points and the edges into interior-disjoint Jordan arcs between the corresponding vertices. We would like to decide whether a given map $\varphi : G \to M$ can be "perturbed" into an embedding $\psi : G \to M$. Let M be a 2-dimensional manifold equipped with a metric. A continuous piecewise linear map $\varphi : G \to M$ is a **weak embedding** if, for every $\varepsilon > 0$, there is an embedding $\psi_\varepsilon : G \to M$ with $\|\varphi - \psi_\varepsilon\| < \varepsilon$, where $\|.\|$ is the uniform norm (i.e., sup norm).

Problem Statement and Results. An **embedded graph** H in an orientable 2-manifold M is an abstract graph together with a **rotation system** that specifies, for each vertex of H, the counterclockwise (abbreviated as ccw) cyclic order of incident edges[1]. The **strip system** $\mathcal{H}$ of H (a.k.a. *thickening* of H) is a 2-manifold with boundary constructed as a quotient space of a disjoint union of topological discs, i.e., by gluing together topological discs along boundaries, as follows. For every $u \in V(H)$, create a topological disk D_u, and for every edge $uv \in E(H)$, create a rectangle R_{uv}. For every D_u and R_{uv}, fix an arbitrary orientation of ∂D_u and ∂R_{uv}, respectively. Partition the boundary ∂D_u into $\deg(u)$ arcs, and label them by $A_{u,v}$, for all $uv \in E(H)$, in the cyclic order around ∂D_u determined by the rotation of u in the embedding of H. Finally, the manifold $\mathcal{H}$ is obtained by identifying two opposite sides of every rectangle R_{uv} with $A_{u,v}$ and $A_{v,u}$ via an orientation preserving homeomorphism (i.e., consistently with the chosen orientations of $\partial R_{uv}, \partial D_u$ and ∂D_v).

We formulate a problem instance as a function $\varphi : G \to H$ (for short, φ), where G is an abstract graph, H is an embedded graph, and $\varphi : G \to H$ is a **simplicial map** that maps the vertices of G to vertices of H, and the edges of G to edges or vertices of H, such that incidences are preserved. The simplicial map $\varphi : G \to H$ is a **weak embedding** if there is an embedding $\psi_\varphi : G \to \mathcal{H}$ that maps each vertex $v \in V$ to a point in $D_{\varphi(v)}$, and each edge $uv \in E(G)$ to a Jordan arc in $D_{\varphi(u)} \cup R_{\varphi(u)\varphi(v)} \cup D_{\varphi(v)}$ that has a connected intersection with each of $D_{\varphi(u)}$, $R_{\varphi(u)\varphi(v)}$, and $D_{\varphi(v)}$, and $R_{\varphi(u)\varphi(v)} = \emptyset$ if $u = v$. We say that the embedding ψ_φ **approximates** φ. Our main results is the following.

Theorem 1. *[2]* (i) *Given an abstract graph G with m edges, an embedded graph H, and a simplicial map $\varphi : G \to H$, we can decide in $O(m \log m)$ time whether φ is a weak embedding.*

(ii) *If $\varphi : G \to H$ is a weak embedding, then for every $\varepsilon > 0$ we can also find an embedding $\psi_\varepsilon : G \to M$ with $\|\varphi - \psi_\varepsilon\| < \varepsilon$ in $O(m \log m)$ time.*

Throughout the paper we assume that G has n vertices and m edges. In the plane (i.e., $M = \mathbb{R}^2$), only planar graphs admit weak embeddings hence $m = O(n)$, but our techniques work for 2-manifolds of arbitrary genus, and G may be a dense graph. Our result improves

[1] Our methods extend to **nonorientable** surfaces with minor changes in the combinatorial representations, using a signature $\lambda : E(H) \to \{-1, 1\}$ to indicate whether the edge u (and R_{uv}) is orientation-preserving or -reversing [4].

the running time of the previous algorithm [8] from $O(m^{2\omega}) \leqslant O(m^{4.75})$ to $O(m \log m)$. It also improves the running times of several recent polynomial-time algorithms in special cases, e.g., when the embedding of G is restricted to a given isotopy class [7], and H is a path [3].

Nonsimplicial Maps. If $\varphi : G \to H$ is a continuous map (not necessarily simplicial) that is injective on the edges (each edge is a Jordan arc), we may assume that $\varphi(V(G)) \subseteq V(H)$ by subdividing the edges of H with at most $n = |V(G)|$ additional vertices if necessary. Then φ maps every edge $e \in G$ to a path of length $O(n)$ in H. By subdividing the edges $e \in E(G)$ at all clusters along $\varphi(e)$, we reduce the recognition problem to the regime of simplicial maps (Theorem 1). The total number of vertices may increase to $O(mn)$ and the running time to $O(mn \log(mn))$.

Corollary 1. *Given an abstract graph G with m edges, an embedded graph H with n vertices, and a piecewise linear continuous map $\varphi : G \to H$ that is injective on the interior of every edge in $E(G)$, we can decide in $O(mn \log(mn)$ time whether φ is a weak embedding.*

For example, this applies to straight-line drawings in $\mathbb{R}^2$ if the edges may pass through vertices.

Corollary 2. *Given an abstract graph G with n vertices and a map $\varphi : G \to \mathbb{R}^2$ where every edge is mapped to a straight-line segment, we can decide in $O(n^2 \log n)$ time whether φ is a weak embedding.*

Outline. Our results rely on ideas from [1, 5, 6] and [8]. To distinguish the graphs G and H, we use the convention that G has **vertices** and **edges**, and H has **clusters** and **pipes**. A cluster $u \in V(H)$ corresponds to a subgraph $\varphi^{-1}[u]$ of G.

The main tool in our algorithm is a local operation, called "cluster expansion," which generalizes similar operations introduced previously for the case that G is a cycle. Given an instance $\varphi : G \to H$ and a cluster $u \in V(H)$, it modifies u and its neighborhood (by replacing u with several new clusters and pipes) and it is "reversible" in the sense that the resulting new instance $\varphi' : G' \to H'$ is a weak embedding if and only if $\varphi : G \to H$ is a weak embedding. Our operation increases the number of clusters and pipes, but it decreases the number of "ambiguous" edges (i.e., multiple edges in the same pipe). The proof of termination and the running time analysis use potential functions.

In a preprocessing phase, we perform a cluster expansion operation at each cluster $u \in V(H)$. The main loop of the algorithm applies another operation, "pipe expansion," for two adjacent clusters $u, v \in V(H)$ under certain conditions. It merges the clusters u and v, and the pipe $uv \in E(H)$ between them, and then invokes cluster expansion. If any of these operations finds a local configuration incompatible with an embedding, then the algorithm halts and reports that φ is not a weak embedding (this always corresponds to nonplanar subconfigurations since the neighborhood of a single cluster or pipe is homeomorphic to a disk). We show that after $O(m)$ successive operations, we obtain an irreducible instance for which our problem is easily solvable in $O(m)$ time. Ideally, we end up with $G = H$ (one vertex per cluster and one edge per pipe), and $\varphi = \mathrm{id}$ is clearly an embedding. Alternatively, G and H may each be a cycle (possibly G winds around H multiple times), and we can decide whether φ is a weak embedding in $O(m)$ time by a simple traversal of G. If G is disconnected, then each component falls into one of the above two cases, i.e., the case when $\varphi = \mathrm{id}$ or the case when $\varphi \neq \mathrm{id}$.

The main challenge was to generalize previous reversible local operations (that worked well for cycles [6, 5, 10]) to arbitrary graphs. Our expansion operation for a cluster $u \in V(H)$ simplifies each component of the subgraph $\varphi^{-1}[u]$ of G independently. Each component is planar (otherwise it cannot be perturbed into an embedding in a disk D_u). However, a planar (sub)graph with k vertices may have $2^{O(k)}$ combinatorially different embeddings: some of these may or may not be compatible with adjacent clusters. The embedding of a (simplified) component C of $\varphi^{-1}[u]$ depends, among other things, on the edges that connect C to adjacent clusters. The **pipe-degree** of C is the number of pipes that contain its incident edges. If the pipe-degree of C is 3 or higher, then the rotation system of H constrains the embedding of C. If the pipe-degree is 2, however, then the embedding of C can only be determined up to a reflection, unless C is connected by *two* independent edges to a component in $\varphi^{-1}[v]$ whose orientation is already fixed.

We need to maintain the feasible embeddings of the components in all clusters efficiently. In [8], this problem was resolved by introducing 0-1 variables for the components, and aggregating the constraints into a system of linear equations over $\mathbb{Z}_2$, which was eventually resolved in $O(m^{2\omega}) \leqslant O(m^{4.75})$ time. We improve the running time to $O(m \log m)$ by maintaining the feasible embeddings simultaneously with our local operations.

Another challenge comes from the simplest components in a cluster $\varphi^{-1}[u]$. Long chains of degree-2 vertices, with one vertex per cluster, are resilient to our local operations. Their length may decrease by only one (and cycles are irreducible). We need additional data structures to handle these "slowly-evolving" components efficiently. We use a dynamic heavy-path decomposition data structure and a suitable potential function to bound the time spent on such components.

References

[1] Hugo A. Akitaya, Greg Aloupis, Jeff Erickson, and Csaba D. Tóth. Recognizing weakly simple polygons. *Discrete Comput. Geom.*, 58(4):2017, 785–821.

[2] Hugo A. Akitaya, Radoslav Fulek, and Csaba D. Tóth. Recognizing weak embeddings of graphs. In *Proc. 29th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2018, to appear.

[3] Patrizio Angelini, Giordano Da Lozzo, Giuseppe Di Battista, and Fabrizio Frati. Strip planarity testing for embedded planar graphs. *Algorithmica*, 77(4):1022–1059, 2017.

[4] Henning Bruhn and Reinhard Diestel. MacLane's theorem for arbitrary surfaces. *J. Combin. Theory Ser. B*, 99:275–286, 2009.

[5] Hsien-Chih Chang, Jeff Erickson, and Chao Xu. Detecting weakly simple polygons. In *Proc. 26th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1655–1670, 2015.

[6] Pier Francesco Cortese, Giuseppe Di Battista, Maurizio Patrignani, and Maurizio Pizzonia. On embedding a cycle in a plane graph. *Discrete Math.*, 309(7):1856–1869, 2009.

[7] Radoslav Fulek. Embedding graphs into embedded graphs. In *Proc. 28th Internat. Sympos. on Algorithms and Computation (ISAAC)*, LIPIcs. Schloss Dagstuhl, 2017.

[8] Radoslav Fulek and Jan Kynčl. Hanani–Tutte for approximating maps of graphs. Preprint, arXiv:1705.05243, 2017.

[9] Bojan Mohar. A linear time algorithm for embedding graphs in an arbitrary surface. *SIAM J. Discrete Math.*, 12(1):6–26, 1999.

[10] Mikhail Skopenkov. On approximability by embeddings of cycles in the plane. *Topology Appl.*, 134(1):1–22, 2003.

[11] Carsten Thomassen. The graph genus problem is NP-complete. *J. Algorithms*, 10(4):568–576, 1989.

Balanced Line Separators of Unit Disk Graphs

Speaker: Yoshio Okamoto
The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan
(a joint work with Paz Carmi, Man-Kwun Chiu, Matthew J. Katz, Matias Korman, AndrïğćŢ van Renssen, Marcel Roeloffzen, Taichi Shiitada, and Shakhar Smorodinsky.)

Balanced separators in graphs are a fundamental tool and used in many divide-and-conquer-type algorithms as well as for proving theorems by induction. Given an undirected graph $G = (V, E)$ with V as its vertex set and E as its edge set, and a non-negative real number $\alpha \in [1/2, 1]$, we say that a subset $S \subseteq V$ is an *α-separator* if the vertex set of $G \setminus S$ can be partitioned into two sets A and B, each of size at most $\alpha|V|$ such that there is no edge between A and B. The parameter α determines how balanced the two sets A and B are in terms of size. For a balanced separator to be useful we want both the size $|S|$ of the separator and $\alpha \geqslant 1/2$ to be small.

Much work has been done to prove the existence of separators with certain properties in general sparse graphs. For example, the well-known Lipton–Tarjan planar separator theorem states that for any n-vertex planar graph, there exists a 2/3-separator of size $O(\sqrt{n})$. Similar theorems have been proven for bounded-genus graphs, minor-free graphs, low-density graphs, and graphs with polynomial expansion.

These separator results apply to graph classes that do not contain complete graphs of arbitrary size, and each graph in the classes contains only $O(n)$ edges, where n is the number of vertices. Since any α-separator of a complete graph has $\Omega(n)$ vertices, the study of separators for graph classes that contain complete graphs seems useless. However, it is not clear how small a separator can be with respect to the number of edges for possibly dense graphs.

Our focus of interest is possibly dense geometric graphs, which often encode additional geometric information other than adjacency. Even though one can use the separator tools in geometric graphs, often the geometric information is lost in the process. As such, a portion of the literature has focused on the search of balanced separators that also preserve the geometric properties of the geometric graph.

Among several others, we highlight the work of Miller et al., and Smith and Wormald. They considered intersection graphs of n balls in $\mathbb{R}^d$ and proved that if every point in d-dimensional space is covered by at most k of the given balls, then there exists a $(d+1)/(d+2)$-separator of size $O(k^{1/d}n^{1-1/d})$ (and such a separator can be found in deterministic linear time). More interestingly, the separator itself and the two sets it creates have very nice properties; they show that there exists a $(d-1)$-dimensional sphere that intersects at most $O(k^{1/d}n^{1-1/d})$ balls and contains at most $(d+1)n/(d+2)$ balls in its interior and at most $(d+1)n/(d+2)$ balls in its exterior. In this case, the sphere acts as the separator (properly speaking, the balls that intersect the sphere), whereas the two sets A and B are the balls that are inside and outside the separator sphere, respectively. Note that the graph induced by the set A consists of the intersection graph of the balls inside the separator (similarly, B for the balls outside the separator and S for the balls intersecting the sphere).

We emphasize that, even though the size of the separator is larger than the one from Lipton–Tarjan for planar graphs (specially for high values of d), the main advantage is that the three subgraphs it creates are geometric graphs of the same family (intersection graphs of balls in $\mathbb{R}^d$). The bound on the separator size does not hold up well when k is large, even for $d = 2$:

if $\sqrt{n}$ disks overlap at a single point and the other disks form a path we have $k = \sqrt{n}$ and $m = \Theta(n)$, where m is the number of edges in the intersection graph. Hence, the separator has size $O(\sqrt{kn}) = O(m^{3/4})$.

Fox and Pach gave another separator result that follows the same spirit: the intersection graph of a set of Jordan curves in the plane has a 2/3-separator of size $O(\sqrt{m})$ if every pair of curves intersects at a constant number of points. A set of disks in $\mathbb{R}^2$ satisfies this condition, and thus the theorem applies to disk graphs. Their proof can be turned into a polynomial-time algorithm. However, we need to construct the arrangement of disks, which takes $O(n^2 2^{\alpha(n)})$ time, where $\alpha(n)$ is the inverse Ackermann function, and in practice an efficient implementation is non-trivial.

From a geometric perspective these two results show that, given a set of unit disks in the plane, we can always find a closed curve in the plane (a circle and a Jordan curve, respectively) to partition the set. The disks intersected by the curve are those in the separator, and the two disjoint sets are the disks inside and outside the curve, respectively.

In this paper, we continue the idea of geometric separators and show that a balanced separator always exists, even if we constrain the separator to be a line. Given a set of n unit disks with m pairs of intersecting disks, we show that a line 2/3-separator of size $O(\sqrt{(m+n)\log n})$ can be found in expected $O(n)$ time, and that an axis-parallel line 4/5-separator of size $O(\sqrt{m+n})$ can be found in deterministic $O(n)$ time.

We emphasize that our results focus on *unit* disk graphs, while the other results hold for disk graphs of arbitrary radii, too. Indeed, if we want to separate disks of arbitrary radii with a line, we show that the separator's size may be as large as $\Omega(n)$. We also prove that for unit disks our algorithm may fail to find a line 2/3-separator of size better than $O(\sqrt{m\log(n/\sqrt{m})})$ in the worst case. In this sense, the size of our separators is asymptotically almost tight. We also present experimental results: We evaluate the performance of our algorithm, compare it with the method by Fox and Pach in terms of the size of the produced separators for random instances, and conclude that our algorithm outperforms theirs for the intersection graphs of unit disks.

Shortcuts for the Circle

Speaker: Sang Won Bae
Kyonggi University, Korea
(a joint work with Mark de Berg, Otfried Cheong, Joachim Gudmundsson and Christos Levcopoulos)

Graph augmentation problems have received considerable attention over the years. The goal in such problems is typically to add extra edges to a given graph G in order to improve some quality measure. One natural quality measure is the (vertex- or edge-)connectivity of G. This has led to work where one tries to find the minimum number of edges that can be added to the graph to ensure it is k-connected, for a desired value of k. Another natural measure is the diameter of G, that is, the maximum distance between any pair of vertices. The goal then becomes to reduce the diameter as much as possible by adding a given number of edges, or to achieve a given diameter with a small number of extra edges.

Chung and Garey (1984) studied this problem for the special case where the original graph is the n-vertex cycle. They showed that if k edges are added, then the diameter of the resulting graph is at least $\frac{n}{k+2} - 3$ for even k and $\frac{n}{k+1} - 3$ for odd k, and that there is a way to add k edges so that the resulting graph has diameter at most $\frac{n}{k+2} - 1$ for even k and $\frac{n}{k+1} - 1$ for odd k.

The algorithmic problem of finding a set of $k \geqslant 1$ edges that minimizes the diameter of the augmented graph was first asked by Chung in 1987. Since then many papers have considered the problem for general graphs. Große et al. (2015) were the first to consider the diameter minimization problem in the geometric setting where the graph is embedded in the Euclidean plane. They presented an $O(n \log^3 n)$ time algorithm that determined the optimal shortcut that minimizes the diameter of a polygonal path with n vertices. The running time was later improved to $O(n \log n)$ by Wang (2016).

In the above papers only the discrete setting is considered, that is, shortcuts connect two vertices and the diameter is measured between vertices. In the continuous setting all points along the edges of the network are taken into account when placing a shortcut and when measuring distances in the augmented network. In the continuous setting, Yang (2013) studied the special case of adding a single shortcut to a polygonal path and gave several approximation algorithms for the problem. De Carufel et al. (2016) considered the problem for paths and cycles. For paths they showed that an optimal shortcut can be determined in linear time. For cycles they showed that a single shortcut can never decrease the diameter, while two shortcuts always suffice. They also proved that for convex cycles the optimal pair of shortcuts can be computed in linear time. Recently, Cáceres et al. (2016) gave a polynomial time algorithm that can determine whether a plane geometric network admits a reduction of the continuous diameter by adding a single shortcut.

We are interested in a geometric continuous variant of this problem. Let C be a unit circle in the plane. We define the *distance* $d(p, q)$ between two points $p, q \in C$ to be the length of the smaller arc along C that connects p to q. Thus the diameter of C in this metric is π. We now want to add a number of *shortcuts*—a shortcut is a chord of C—to improve the diameter. Here the distance $d_S(p, q)$ between p and q for a given collection S of shortcuts is defined as the length of the shortest path between p and q that can travel along C and along the shortcuts where, if two shortcuts intersect in their interior, we do not allow the path to switch from one shortcut to the other at the intersection point. In other words, if the path uses a shortcut, it has to traverse it completely. Note that if we view the circle C as a graph with infinitely many vertices (namely all points on C) where the graph distance is the distance along C, then adding shortcuts corresponds to adding edges to the graph. For a set S of shortcuts, define

$$\operatorname{diam}(S) := \max_{p,q \in C} d_S(p, q)$$

to be the diameter of the resulting graph. We are interested in the following question: given k, the number of shortcuts we are allowed to add, what is the best diameter we can achieve? In other words, we are interested in the quantity

$$\operatorname{diam}(k) := \inf_{|S|=k} \operatorname{diam}(S).$$

It is obvious that

$$\pi = \operatorname{diam}(0) \geqslant \operatorname{diam}(1) \geqslant \cdots \geqslant \operatorname{diam}(k) \geqslant \cdots \geqslant \lim_{k \to \infty} \operatorname{diam}(k) = 2.$$

Our main results are as follows.

- For $1 \leqslant k \leqslant 7$, we determine $\operatorname{diam}(k)$ exactly. Our results show that $\operatorname{diam}(k)$ is not strictly decreasing as a function of k. This not only holds at the very beginning—it is easy to see that $\operatorname{diam}(1) = \operatorname{diam}(0)$—but, interestingly also for certain larger values of k. In particular, we show that $\operatorname{diam}(7) = \operatorname{diam}(6)$.
- We have $\operatorname{diam}(8) < \operatorname{diam}(7)$.
- We show that $\operatorname{diam}(k) = 2 + \Theta(1/k^{\frac{2}{3}})$.

We rely on a number of numerical calculations in this work. A Python script that performs these calculations can be found at `http://github.com/otfried/circle-shortcuts`.

Geodesic Voronoi Diagrams in a Simple Polygon

Speaker: Eunjin Oh

Since the early 1980s, many classical geometric problems have been studied in the presence of polygonal obstacles. In the presence of polygonal obstacles, the distance between two points is measured by the length of a shortest path between the two points avoiding obstacles. It is called the geodesic distance to distinguish it from the Euclidean distance. Under the geodesic distance, most of the classical geometric structures such as the Voronoi diagram and the convex hull are naturally extended to the geodesic setting.In this talk, I will introduce some of the recent work on Voronoi diagrams inside a simple polygon. A geodesic Voronoi diagram of point sites in a simple polygon partitions the polygon into cells based on the geodesic distances to the sites. The geodesic nearest-point Voronoi diagram partitions the polygon into cells, exactly one cell per site, such that every point in a cell has the same nearest site under the geodesic metric. Similarly, the geodesic farthest-point Voronoi diagram partitions the polygon into cells, at most one cell per site, such that every point in a cell has the same farthest site under the geodesic metric.

Although the first nontrivial algorithms for computing the geodesic nearest-point and farthest-point Voronoi diagrams were presented in 1980s, the optimal running times for these problems are not known. In other words, there are gaps between the best known running times and the best known lower bounds. However, very recently, there was the first improvement on the computation of the geodesic nearest-point and farthest-point Voronoi diagrams since 1998 and 1993, respectively. This talk will mainly deals with these algorithms.

Global Rigidity of Triangulations with Braces

Speaker: Shin-ichi Tanigawa (Based on a joint work with Tibor Jordán[4].)

Cerebrated Cauchy's theorem[1] states that if the vertex-edge graphs of two convex polyhedra are isomorphic and corresponding faces are congruent then the two polyhedra are the same. This theorem in particular implies that a convex simplicial polyhedron (i.e., a convex polyhedron with triangular faces) is rigid as a bar-and-joint framework. A natural question would be whether simplicial polyhedra have a stronger rigidity property such as global rigidity (i.e., unique realizability). Cauchy's theorem states uniqueness within the family of convex realizations, but the uniqueness fails if we drop the assumption of convexity. For example if the graph of a simplicial polyhedron has a separator of size three, then one can always construct a distinct realization by reflecting one side of the polyhedron along the hyperplane spanned by those three points. Thus 4-connectivity is necessary for the global rigidity of 1-skeleta.

In 1992 B. Hendrickson [3] proved a necessary condition for a generic realization of a graph to be globally rigid, which in turn implies that the 1-skeleton of a generic polyhedron cannot be globally rigid regardless of the connectivity of the underlying graph.

Motivated by this background, W. Whiteley studied the rigidity of simplicial polyhedra with *braces*. See Figure 2. He proved that a simplicial polyhedron with a bracing edge has a substantially stronger rigidity property if the underlying graph is 4-connected:

Theorem 1. *[Whiteley [5]] A generic simplicial polyhedron with one bracing edge is redundantly rigid (i.e., rigid after the removal of any edge) in $\mathbb{R}^3$ if the underlying graph is 4-connected.*

In his talk at the Advances in Combinatorial and Geometric Rigidity Workshop (BIRS, Banff, 2015) he conjectured that every 4-connected uni-braced generic simplicial polyhedron is in fact globally rigid. In this work we prove the following more general statement.

Theorem 2. *A generic simplicial polyhedron with at least one bracing edge is globally rigid in $\mathbb{R}^3$ if the underlying graph is 4-connected.*

As we remarked above, 4-connectivity is a trivial necessary condition for the global rigidity, and hence Theorem 2 chracterizes the global rigidity of generic simplicial polyhedra with braces.

Theorem 2 will follow by showing that a 4-connected uni-braced triangulation (that is, a 4-connected maximal planar graph with one extra edge) is globally rigid in the sense of the rigidity theory. Our proof for the latter claim is first to show an inductive construction of 4-connected uni-braced triangulations and then to show each operation preserves the global rigidity. The operation we consider is the so-called vertex splitting operation, which is defined as follows.

Let $H = (V, E)$ be a graph. For a vertex $v \in V$ we use $N_H(v)$ to denote the set of neighbours of v in H. Given a vertex $v_1 \in V$ and a partition $\{U_{01}, U_0, U_1\}$ of $N_H(v)$ with $|U_{01}| = k$, the *k-vertex splitting* operation at v_1 with respect to $\{U_{01}, U_0, U_1\}$ removes the edges connecting v_1 to U_0 and inserts a new vertex v_0 as well as new edges between v_0 and $\{v_1\} \cup U_{01} \cup U_0$. See Figure 3.

The operation is *nontrivial* if U_0 and U_1 are both non-emtpy.

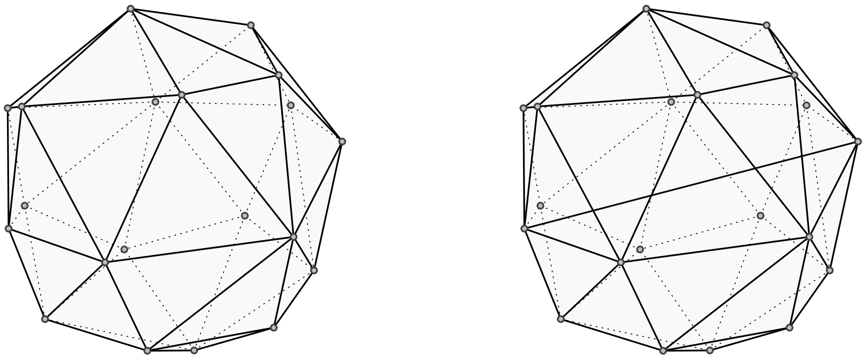

Figure 2: A simplicial polyhedron and a simplicial polyhedron with a brace.

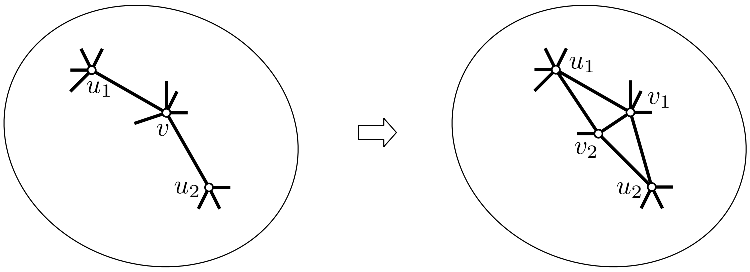

Figure 3: A 2-vertex splitting operation at v with $U_{01} = \{u_1, u_2\}$.

The vertex-splitting operation is well-known in rigidity theory as well as in the theory of polyhedra and triangulations of surfaces. Steinitz proved that every triangulation can be obtained from K_4 by a sequence of 2-vertex splitting operations. Whiteley [6] proved that $(d-1)$-vertex splitting preserves rigidity in $\mathbb{R}^d$.

Whiteley conjectured that $(d-1)$-vertex splitting preserves global rigidity in $\mathbb{R}^d$ provided it does not create vertices of degree d. The corresponding statement for global rigidity was conjectured by Connelly and Whiteley.

Conjecture 1 (Connelly and Whiteley [2])**.** *Let H be globally rigid in $\mathbb{R}^d$ with at least $d+2$ vertices and let G be obtained from H by a nontrivial $(d-1)$-vertex-splitting operation. Then G is globally rigid in $\mathbb{R}^d$.*

This conjecture is still open for $d \geqslant 3$. Our second main result is the following.

Theorem 3. *Suppose that G can be obtained from K_{d+2} by a sequence of non-trivial $(d-1)$-vertex splitting operations. Then G is globally rigid in $\mathbb{R}^d$.*

References

[1] A.L. Cauchy, Sur les polygones et polyedres, second memoire, *J. Ecole Polytechnique*, 1813.

[2] R. Connelly and W. Whiteley, Global rigidity: the effect of coning, *Discrete Comp. Geometry*, Volume 43, Number 4, 717–735, 2010.

[3] B. Hendrickson, Conditions for unique graph realizations, *SIAM J. Comput* 21 (1992), pp 65-84.

[4] T. Jordán and S. Tanigawa, Global rigidity of triangulations with braces, EGRES technical report, TR-2017-06, 2017.

[5] W. Whiteley, Infinitesimally rigid polyhedra. II: Modified spherical frameworks, *Trans. Amer. Math. Soc.* 306, No. 1, 115–139, 1988.

[6] W. Whiteley, Vertex splitting in isostatic frameworks, *Structural Topology* 16, 23–30, 1991.

Open Problems on Optimal Patrolling

Speaker: Akitoshi Kawamura (Kyushu University)

In patrolling problems [2, 3, 4], several mobile agents with predefined speeds move on the edges of a graph and try to ensure that every specified point on the graph is (perpetually) visited at least once in any time period of unit length. Problems of this kind are studied with various motivations and in various forms: the agents may have the same or different speeds; the underlying graph may be a path, a cycle, a tree, or more general graphs; the points to be visited may be just some of the vertices or all points on the edges. Finding an optimal patrolling schedule is not straightforward, even in the simplest settings. I will introduce some recent results and open questions about properties of and algorithms for optimal patrolling. In this abstract I list three specific open problems.

Open Problem 1. *Consider the problem [2] where the goal is to visit every point on the given line segment at least once in every time interval of unit length. What is the smallest constant c such that the following holds for any k and any set of speeds $v_1, \ldots, v_k > 0$?*

> *It is impossible to deploy k agents with speeds $v_1, \ldots, v_k$ and patrol a line segment of length greater than $c(v_1 + \cdots + v_k)$.*

The obvious partition-based strategy (in which each agent i goes back and forth on a subsegment of length $v_i/2$) patrols a total length of $(v_1 + \cdots + v_k)/2$, proving $c \geqslant 1/2$. Perhaps surprisingly, a longer fence can be patrolled [5], and our current best knowledge is that $c \geqslant 2/3$, as shown in [7]. On the other hand, it is relatively easy to see that $c \leqslant 1$. We have not succeeded in proving that $c < 1$.

Open Problem 2. *Consider the patrolling problem on a given tree (with edge lengths) where the goal is to visit every leaf at least once in any time interval of unit length. Is there a polynomial-time algorithm to determine whether a given tree can be patrolled by a given number of agents with speed 1?*

There is a simple (and in particular polynomial-time) way to solve the problem on stars (i.e., in the special case where the depth of the tree is 1) [6]. The analogous problem on general graphs is **NP**-hard, even for a single agent, because it subsumes the traveling salesman problem.

In our setting, a single point can be guarded by multiple agents together. The situation might be different if we require [1] that a point must be visited sufficiently often by some single agent.

Open Problem 3. *Consider the problem [7] where the goal is to protect just one point. That is, instead of the speed of each agent i, we are given a number a_i, which means that i can visit the point only after time a_i has elapsed since its last visit. Can we determine in polynomial time whether this is possible for a given sequence $(a_1, \ldots, a_k)$? Is the problem in* **NP***? Is it* **NP***-hard?*

The problem is in **NP** if there is a polynomial p such that, if patrolling is possible with a given sequence $(a_1, \ldots, a_k)$ at all, then it is possible by a periodic schedule with period at most $p(\lceil \log a_1 \rceil + \cdots + \lceil \log a_k \rceil)$. We do not know whether there is such a polynomial upper bound on the period.

This research is supported in part by JSPS KAKENHI Challenging Research (Exploratory) JP17K19960.

References

[1] S. Coene, F. C. R. Spieksma, and G. J. Woeginger. Charlemagne's challenge: The periodic latency problem. *Operations Research*, 59(3), 674–683, 2011.

[2] J. Czyzowicz, L. Gąsieniec, A. Kosowski, and E. Kranakis. Boundary patrolling by mobile agents with distinct maximal speeds. In *Proc. 19th Annual European Symposium on Algorithms* (ESA 2011), LNCS 6942, pp. 701–712.

[3] A. Dumitrescu and C. D. Tóth. Computational Geometry Column 59. *ACM SIGACT News*, 45(2), 2014.

[4] 河村彰星．最適の警邏に関する諸問題．数理解析研究所講究録2027（最適化技法の最先端と今後の展開）85〜92頁．平成29年．(A. Kawamura. Problems on optimal patrolling. *RIMS Kôkyûroku*, 2027, 85–92, 2017. In Japanese.)

[5] A. Kawamura and Y. Kobayashi. Fence patrolling by mobile agents with distinct speeds. *Distributed Computing*, 28(2), 147–154, 2015. Preliminary version in *Proc. 23rd International Symposium on Algorithms and Computation* (ISAAC 2012), LNCS 7676, pp. 598–608.

[6] A. Kawamura and H. Noshiro. Multi-agent cooperative patrolling of designated points on graphs. In *Proc. 20th Japan Conference on Discrete and Computational Geometry, Graphs, and Games* (JCDCG^3 2017).

[7] A. Kawamura and M. Soejima. Simple strategies versus optimal schedules in multi-agent patrolling. In *Proc. Ninth International Conference on Algorithms and Complexity* (CIAC 2015), LNCS 9079, pp. 261–273.

Routing on the Visibility Graph

Speaker: André van Renssen (a joint work with Prosenjit Bose, Matias Korman, and Sander Verdonschot.)

Routing is a fundamental problem in networking. The goal is to find a path from a source vertex to a destination vertex in the network. When the whole network is known to the routing algorithm, there exist many algorithms to find paths. The problem is more challenging when the only information available is the location of the current vertex, its neighbours and a constant amount of additional information (such as the source and destination vertex). This is often referred to as *local* routing (or k-local for some constant k, when the k-neighbourhood is considered). In our setting, we assume that the network is a graph embedded in the plane, with edges being straight line segments connecting pairs of vertices, weighted by the Euclidean distance between their endpoints. Algorithms routing on such networks are referred to as *geometric* routing algorithms (see [10] and [11] for surveys of the area).

Deterministic routing algorithms that guarantee delivery in these networks typically route on plane subgraphs of the complete Euclidean graph. This means that of the potentially quadratic number of edges available to the routing algorithm, only a linear number are ever considered. This forces these algorithms to use paths that are much longer than necessary. We present the first deterministic local routing algorithm that considers more edges by not restricting its choices to a plane subgraph.

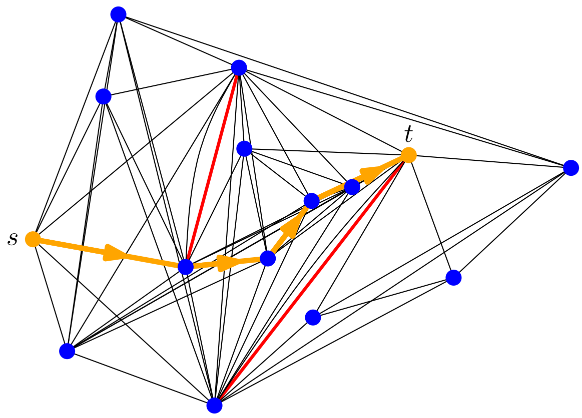

Figure 4: The visibility graph of a point set, where the constraints are thick red line segment. An example of a routing path from s to t is shown in orange.

Moreover, we study routing algorithms in a more general setting. In certain cases, some edges of a network may not be usable if for example there is a large obstacle blocking direct communication between two nodes. We model this impossibility via a set S of non-intersecting *line segment constraints* whose endpoints are vertices of the network (see Figure 4). Given a set P of n points in the plane and a set S of non-intersecting line segment constraints, we say that two vertices u and v can *see each other* provided that either the line segment uv does not properly intersect any constraint in S or uv is itself a constraint in S. If two vertices u and v can see each other, the line segment uv is referred to as a *visibility edge*. The *visibility graph* of P with respect to a set of constraints S has P as vertex set and all visibility edges as edge set. In other words, the visibility graph is the complete graph on P minus all edges that properly intersect one or more constraints in S.

Although this setting has been studied extensively in the context of motion planning amid obstacles ([7, 8, 1, 6]), there has not been much work on routing in this setting. Bose *et al.* [2] showed that it is possible to route locally and 2-competitively between any two visible vertices in the constrained Θ_6-graph. Additionally, an 18-competitive routing algorithm between any two visible vertices in the constrained half-Θ_6-graph was provided. In the same paper it was shown that no deterministic local routing algorithm is $o(\sqrt{n})$-competitive between all pairs of vertices of the constrained Θ_6-graph, regardless of the amount of memory it is allowed to use.

We present three deterministic 1-local $O(1)$-memory routing algorithms on the visibility graph. The first locally computes a plane subgraph of the visibility graph (the constrained half-Θ_6-graph) and uses face routing [9, 5] to route on this subgraph. This algorithm leads to the natural question of whether it is possible to route on the visibility graph without computing a plane subgraph.

Theorem 1. *There exists a 1-local $O(1)$-memory routing algorithm for the visibility graph by routing on the (plane) constrained half-Θ_6-graph, that reaches the destination in $O(n)$ steps.*

Our second algorithm answers this question in the affirmative by locally computing a non-plane subgraph of the visibility graph (the constrained Θ_6-graph) and routing on that.

Theorem 2. *There exists a 1-local $O(1)$-memory routing algorithm for the visibility graph by routing on the (non-plane) constrained Θ_6-graph, that reaches the destination in $O(n)$ steps.*

However, since computing any subgraph of the visibility graph incurs some overhead, we aim to route on the visibility graph directly, i.e., without computing any subgraph. This is indeed also possible, as the second algorithm can be modified to obtain a routing algorithm that routes directly on the visibility graph. To the best of our knowledge, this is the first deterministic local routing algorithm does not compute a plane subgraph of the visibility graph.

Theorem 3. *There exists a 1-local $O(1)$-memory routing algorithm for the visibility graph that reaches the destination in $O(n)$ steps.*

Unfortunately, these algorithms do not give guarantees on the length of the routing path, only on the number of edges used. Hence, designing an algorithm that is competitive with respect to the shortest path remains open. Is this possible when we consider only deterministic algorithms or does this require the local algorithm to be non-deterministic? Or are we even forced to either increase the local information or store some specific additional information at the vertices?

References

[1] Prosenjit Bose, Rolf Fagerberg, André van Renssen, and Sander Verdonschot. On plane constrained bounded-degree spanners. In *Proceedings of the 10th Latin American*

Symposium on Theoretical Informatics (LATIN 2012), volume 7256 of *Lecture Notes in Computer Science*, pages 85–96, 2012.

[2] Prosenjit Bose, Rolf Fagerberg, André van Renssen, and Sander Verdonschot. Competitive local routing with constraints. *Journal of Computational Geometry (JoCG)*, 8(1):125–152, 2017.

[3] Prosenjit Bose, Matias Korman, André van Renssen, and Sander Verdonschot. Constrained routing between non-visible vertices. In *Proceedings of the 23rd Annual International Computing and Combinatorics Conference (COCOON 2017)*, volume 10392 of *Lecture Notes in Computer Science*, pages 62–74, 2017.

[4] Prosenjit Bose, Matias Korman, André van Renssen, and Sander Verdonschot. Routing on the visibility graph. In *Proceedings of the 28th International Symposium on Algorithms and Computation (ISAAC 2017)*, 2017.

[5] Prosenjit Bose, Pat Morin, Ivan Stojmenovic, and Jorge Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7(6):609–616, 2001.

[6] Prosenjit Bose and André van Renssen. Upper bounds on the spanning ratio of constrained theta-graphs. In *Proceedings of the 11th Latin American Symposium on Theoretical Informatics (LATIN 2014)*, volume 8392 of *Lecture Notes in Computer Science*, pages 108–119, 2014.

[7] Ken Clarkson. Approximation algorithms for shortest path motion planning. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC 1987)*, pages 56–65, 1987.

[8] Gautam Das. The visibility graph contains a bounded-degree spanner. In *Proceedings of the 9th Canadian Conference on Computational Geometry (CCCG 1997)*, pages 70–75, 1997.

[9] Evangelos Kranakis, Harvinder Singh, and Jorge Urrutia. Compass routing on geometric networks. In *Proceedings of the 11th Canadian Conference on Computational Geometry (CCCG 1999)*, pages 51–54, 1999.

[10] Sudip Misra, Subhas Chandra Misra, and Isaac Woungang. *Guide to Wireless Sensor Networks.* Springer, 2009.

[11] Harald Räcke. Survey on oblivious routing strategies. In *Mathematical Theory and Computational Practice*, volume 5635 of *Lecture Notes in Computer Science*, pages 419–429, 2009.

Discussion on Open Problems

We also had a number of open problems posed and discussed during the meeting. Many of them are posed by the invited speakers and they are already mentioned in the previous section. Here we list another four open problems that were discussed during the meeting.

Assigning Radius

posed by Ahmad Biniaz, Carleton University, Ottawa, Canada.

Given a set of n points $p_1, p_2, \ldots, p_n$ lying on a line, assign radius r_i to p_i such that the resulting graph is connected and $\sum_{i=1}^{n} r_i^2$ is minimized. We say that two points p_i and p_j are connected by an edge if $|x(p_j) - x(p_i)| \leqslant \max\{r_i, r_j\}$.

A variation is to assign radius r_i to p_i such that the resulting intersection graph of the disks is connected and $\sum_{i=1}^{n} r_i^2$ is minimized. We say that two points p_i and p_j are connected in the intersection graph if $|x(p_j) - x(p_i)| \leqslant r_i + r_j$.

Both problems can be solved using dynamic programming. However, it is slow. Is there any efficient way of solving the optimization problems?

Optimal Patrolling

posed by Akitoshi Kawamura (partly based on joint work with Yusuke Kobayashi, Hideaki Noshiro, and Makoto Soejima.)

In patrolling problems, several mobile agents move on a graph (with edge lengths) and try to cooperate so that every specified point on the graph is (perpetually) visited sufficiently often (that is, no point should be left unattended for a long time).

Problems of this kind are studied with various motivations and in various forms: the agents may have the same or different speeds; the underlying graph may be a path, a cycle, a tree, or more general graphs; the points to be visited may be just the vertices or all points on the edges. Finding an optimal patrolling schedule is not straightforward, even in the simplest settings. I will introduce some results and open questions about properties of and algorithms for optimal patrolling.

Patrolling a fence (line segment). A requirement is that every point of the fence must be visited by at least one agent during one unit time. Let k be the number of agents, each with speed $v_1, v_2, \ldots, v_k$. Here the goal is to maximize the length of the fence. Dividing the fence into pieces, proportional to the speeds gives a total length $\frac{v_1+v_2+\cdots+v_k}{2}$.

PBS(Partition Based Strategy) is optimal for $v_i = v_j$ for all $i, j \in \{1, \ldots k\}$, and for two agents. However, PBS is not always optimal. Then what is the largest value c such that the following is true for all k and $v_1, \ldots, v_k$?

> No fence of length larger than $c(v_1 + \cdots + v_k)$ can be patrolled by agents with speeds $v_1, \ldots, v_k$.

It is known that $c \leqslant 1$, $c \geqslant 0.520$ [2014], and $c \geqslant 0.666$ [CIAC 2015].

Patrolling vertices on a path. When vertices have different idle times, a simple greedy strategy is not optimal. This kind of problem is also known as Charlemagne's Challenge, which is a periodic latency problem.

Sorting Linear Functions

posed by Siu-Wing Cheng.

Preprocess n linear functions so that given a query parameter t, report the functions in the sorted order encounter by the vertical line $x = t$ efficiently.

Given $\Theta(n^2)$ space, a query can be answered (reporting the functions in the sorted order) by a binary search and reporting in $O(n)$ time. But, can we do it using $o(n^2)$ space and $o(n \log n)$ query time? There have been many trials using cutting and bucketing.

It may require sophisticated data structures.

1. Partition the lines (linear functions) into $O(\log n)$ groups of $O(n/\log n)$ lines.
2. Construct the data structure above for each group. Total space is $O(n^2/\log^2 n) \times O(\log n) = O(n^2/\log n)$.
3. Given a query, compute the sorted lists from the groups and merge them into one. This takes $O(n/\log n) \times O(\log n) + O(n \log\log n)$ time.

Diameters, Centers of Polygons with Holes in 2D and 3D

posed by Matias Korman.

There are plenty of works on computing the (geodesic) diameter and (geodesic) center of convex and simple polygons, and polygons with holes in the plane. Surprisingly, it takes much more time for the problem with polygons with holes than with polygons without holes. This is mainly due to the fact that the shortest path connecting two points lying in a polygon is not necessarily unique for polygons with holes.

We consider the problem in higher dimensions. In specific, consider a polyhedron in 3D. Is the shortest path unique? Is it possible that the diametral pair of a polyhedron lie in the interior of the polyhedron? How can we compute the diameter (diametral pair) of a polyhedron efficiently?

Participants

1. Siu Wing Cheng, Hongkong University of Science and Technology, Hongkong
2. Takeshi Tokuyama, Tohoku University, Sendai, Japan
3. Matias Korman, Tohoku University, Sendai, Japan
4. Yoshio Okamoto, The University of Electro-Communications
5. Akitoshi Kawamura, Kyushu University, Japan
6. Shin-ichi Tanigawa, University of Tokyo, Japan
7. Otfried Cheong, Korea Advanced Institute of Science and Technology
8. Sang Won Bae, Kyonggi University, Korea
9. Michiel Smid, Carleton University, Ottawa, Canada
10. André van Renssen, National Institute of Informatics
11. Anil Maheshwari, Carleton University, Ottawa, Canada
12. Ahmad Biniaz, Carleton University, Ottawa, Canada
13. Matthew (Matya) Katz, Ben-Gurion University, Israel
14. Csaba Toth, California State University Northridge, USA
15. Jan Vahrenhold, University of Münster, Germany
16. Wolfgang Mulzer, Freie Universität Berlin, Germany
17. Bhaswar B. Bhattacharya, University of Pennsylvania, USA
18. Minati De, Indian Institute of Science, Bangalore, India
19. Sudeshna Kolay, Eindhoven University of Technology, The Netherlands
20. Marcel Roeloffzen, National Institute of Informatics, Japan
21. Man Kwun Chiu, National Institute of Informatics, Japan
22. Jinhee Chun, Tohoku University, Sendai, Japan
23. Eunjin Oh, POSTECH, Korea
24. Prosenjit Bose, Carleton University, Ottawa, Canada
25. Takashi Horiyama, Saitama University, Japan
26. Yuya Higashikawa, Chuo University, Japan

Meeting Schedule

Check-in Day: October 29, 2017 (Sun)

- Welcome Banquet

Day1: October 30, 2017 (Mon)

- Talks and Discussions

Day2: October 31, 2017 (Tue)

- Talks and Discussions

Day3: November 01, 2017 (Wed)

- Talks and Discussions
- Group Photo Shooting
- Excursion and Main Banquet

Day4: November 02, 2017 (Thu)

- Talks and Discussions
- Wrap up