JCDCG^3 2017

The 20th Japan Conference on Discrete and Computational Geometry, Graphs, and Games

29th August – 1st September 2017

Tokyo University of Science

The 20th Japan Conference on Discrete and Computational Geometry, Graphs, and Games

August 29 – September 1, 2017
Tokyo University of Science

Invited Plenary Speakers

Jin Akiyama	(Tokyo University of Science, Japan)
Vašek Chvátal	(Concordia University, Canada)
Erik Demaine	(MIT, USA)
David Eppstein	(University of California, Irvine, USA)
Mikio Kano	(Ibaraki University, Japan)
Naoki Katoh	(Kwansei Gakuin University, Japan)
Evangelos Kranakis	(Carleton University, Canada)
János Pach	(EPFL, Switzerland and Renyi Institute, Hungary)
Jorge Urrutia	(UNAM, México)

Conference Chair

Toshinori Sakai	(Tokai University, Japan)

Program Comittee

Jean Cardinal	(ULB, Belgium)
Ruy Fabila-Monroy	(Cinvestav, México)
Takashi Horiyama	(Saitama University, Japan)
Hiro Ito	(UEC, Japan; Chair)
Matias Korman	(Tohoku University, Japan)
Stefan Langerman	(ULB, Belgium)
Chie Nara	(Meiji University, Japan)
Yoshio Okamoto	(UEC, Japan)
Hirotaka Ono	(Kyushu University, Japan)
Vera Sacristán	(UPC, Spain)
Toshinori Sakai	(Tokai University, Japan)
Tomohiro Tachi	(University of Tokyo, Japan)
Ryuhei Uehara	(JAIST, Japan)
Yushi Uno	(Osaka Prefecture University, Japan)
Aaron Williams	(Bard College at Simon's Rock, USA)

Organizing Committee

Hiro Ito (UEC, Japan)
Takako Kodate (Tokyo Woman's Christian University, Japan)
Keiko Kotani (Tokyo University of Science, Japan; Co-chair)
Yasuko Matsui (Tokai University, Japan)
Atsuki Nagao (Seikei University, Japan)
Toshinori Sakai (Tokai University, Japan; Co-chair)
Kazuhisa Seto (Seikei University, Japan)
Xuehou Tan (Tokai University, Japan)
Shin-ichi Tokunaga (Tokyo Medical and Dental University, Japan)

Sponsored by

Tokyo University of Science

In cooperation with

JST CREST: Foundations of Innovative Algorithms for Big Data (ABD14)
Tokai University

Table of Contents

Invited Talks

Memorial Session

Contributed Talks (August 29)

Session1A

Session1B

Session2A

Session2B

Contributed Talks (August 30)

Session3A

Session3B

Session4A

Session4B

Contributed Talks (August 31)

Session5A

Session5B

Session6A

Session6B

Contributed Talks (September 1)

On Orthogonal Polyhedra

Jorge Urrutia
Instituto de Matemáticas
Universidad Nacional Autónoma de México

Joint work with I. Aldana-Galván, J.L. Álvarez-Rebollar,
J.C. Catana-Salazar, M. Jiménez-Salinas, N. Marín-Nevárez,
E. Solís Villarreal, and C. Velarde.

1 Introduction

An orthogonal polygon is a simple polygon all of whose sides are parallel to the x- or the y-axis. Orthogonal polygons have been studied by many authors in the Computational Geometry community mainly on problems related to Art Galleries Problems [8]. A *polyhedron* in $\mathbb{R}^3$ is a compact connected set bounded by a piecewise linear 2-manifold. An orthogonal polyhedron is called orthogonal if all of its faces are all parallel to the xy-, xz- or yz-planes of $\mathbb{R}^3$.

In this talk we will review some of the known results for orthogonal polygons, and present some generalizations of them to orthogonal polyhedra. We will focus on two problems: Minimizing the solid angle sum of orthogonal polyhedra, and beacon coverage. Most of the results presented here, are contained in [1, 2].

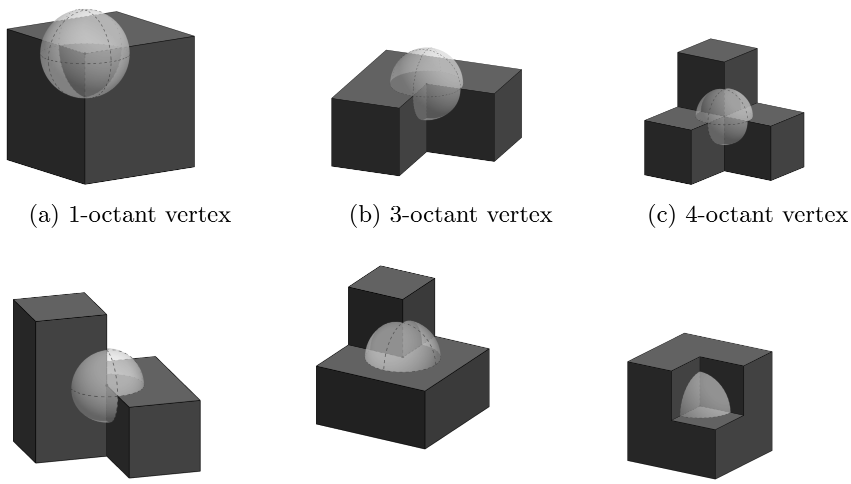
2 Angle Sum

Let $\mathcal{P}$ be a polyhedron in $\mathbb{R}^3$, and let v be a vertex of $\mathcal{P}$. The *solid angle* of $\mathcal{P}$ at v is defined as follows: Consider a small enough sphere $\mathcal{C}$ centered at v. The size of the angle of $\mathcal{P}$ at v is the *area* of the portion of the boundary of $\mathcal{C}$ that lies within $\mathcal{P}$ divided it by the square of the radius of $\mathcal{C}$. Since the area of a unit sphere is 4π, it follows that the maximum size of the solid angles at vertices of a polyhedron is at most 4π.

In the plane we all know that the sum of the angles of a triangle is always π. We note first that this result does not generalize to tetrahedra in $\mathbb{R}^3$, it is a well-known result that for any α, $0 < \alpha < 2\pi$ there is a tetrahedron such that the sum of its solid angles is α, see [7]. In fact, it is not hard to see that there are polyhedra with an arbitrarily large number of vertices such that the sum of their solid angles is arbitrarily small.

Observe that for orthogonal polyhedra, the size of the solid angle at each of their vertices is at least $\pi/2$, and at most $7\pi/2$, each vertex of an orthogonal polyhedron covers one, three, four, five, and seven octants, see Figure 1. Thus a natural question that arises is the following: Can we characterize orthogonal polyhedra with n vertices that minimize or maximize the sum of their solid angles?

Let $\mathcal{P}$ be an orthogonal polyhedron in $\mathbb{R}^3$. We classify the vertices of P according to the size of interior solid angles. A vertex x of $\mathcal{P}$ is classified as 1-*octant* if its interior solid angle is $\pi/2$ (see Figure 1a), and 3-*octant* if its interior solid angle is $3\pi/2$ (see Figure 1b). The 4-*octant*, 5-*octant* and 7-*octant* vertices are defined in a similar way, as illustrated in Figures 1c, 1d, 1e and 1f respectively.



(a) 1-octant vertex (b) 3-octant vertex (c) 4-octant vertex

(d) 4-octant vertex (e) 5-octant vertex (f) 7-octant vertex

Figure 1: Vertex classification for orthogonal polyhedra.

We will prove the following result:

Theorem 1. *The minimum interior solid angle sum of orthogonal polyhedra with n vertices and genus g is $(n-4+4g)\pi$ and is achieved by polyhedra having only* 1*-octant,* 3*-octant, and possibly* 4*-octant vertices.*

To prove our previous result, we will prove the following results that generalizes the well known result on orthogonal polygons.

Lemma 1. *Let P be an orthogonal polygon with n vertices, and h holes. Then the number of reflex and convex vertices is, respectively, $r = (n+4h-4)/2$ and $c = (n-4h+4)/2$.*

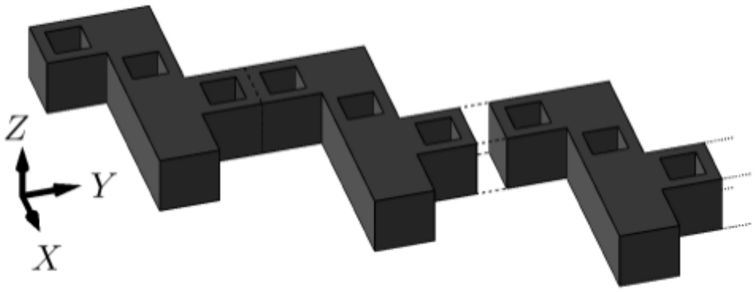
Let k_3 be the vertices of degree 3, k_4 the vertices of degree 4, and k_6 the vertices of degree 6 in the 1-skeleton of a polyhedron. We prove:

Theorem 2. *Let $\mathcal{P}$ be an orthogonal polyhedron in $\mathbb{R}^3$ with $n = k_3 + k_4 + k_6$ vertices and arbitrary genus g. Then $\mathcal{P}$ has $(n - 3(k_4 + k_6) + 8g - 8)/2$ reflex vertices and $(n + 3(k_4 + k_6) - 8g + 8)/2$ convex vertices.*

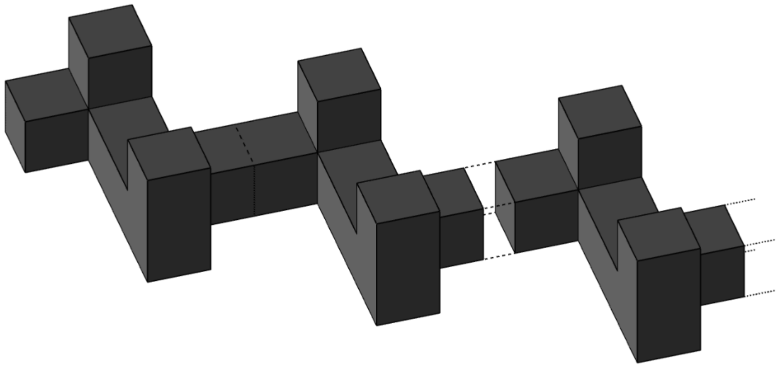
For $g = 0$, we have:

Theorem 3. *Let $\mathcal{P}$ be an orthogonal polyhedron in $\mathbb{R}^3$ homeomorphic to the sphere, with $n = 2k$ vertices, and such that its* 1*-skeleton is a* 3-regular *graph. Then $\mathcal{P}$ has $(n + 8)/2$ convex vertices and $(n - 8)/2$ reflex vertices.*

Observe that the bound given in Theorem 1 is minimized when $g = 0$.



(a) A family of lifting orthogonal polyhedra that minimize its solid angle sum.



(b) A family of orthogonal polyhedra with 4-octant vertices that minimize the solid angle sum.

Figure 2: Family of polyhedra that minimize the solid angle sum.

3 Beacon Coverage

A *beacon* is a fixed point in a polyhedron P that can induce a magnetic pull toward itself over all points in P. When a beacon b is activated, points in P move greedily to decrease their euclidean distance to b. A point p can move along any obstacles it hits on its way to a beacon b as long as its distance to b keeps on decreasing. Thus, the path from the initial position of p to a beacon b may alternate between moving in straight line segments contained in the interior of P and line segments on the faces of P.

The piecewise linear path created by the movement of p under the attraction of b is called the *attraction path* of p with respect to b. If the attraction path of p ends in b, we say that p is *covered* by b. If p is in a position where it is unable to move in such a way that its distance to b decreases, we say that it is *stuck* and it has reached a *local minimum*, or a *dead end*, see Figure 3.

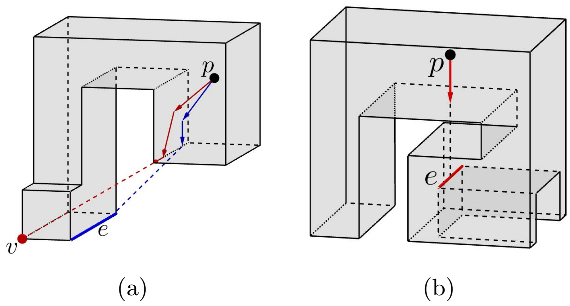


Figure 3: Two examples of points that reach a local minimum: (a) The attraction path of a point with respect to a vertex beacon and an edge beacon, both unreachable, and (b) the point gets stuck on its way to an edge beacon.

Beacon attraction was introduced by Biro et al. [4, 5, 6]. This model extends the classical notion of visibility; if an object p is visible from a beacon q, then p moves towards q along the straight line segment joining p to q.

4 Covering orthogonal polyhedra

Bae et al. [3] proved that the interior of any orthogonal n-gon can be covered with $\lfloor \frac{n}{6} \rfloor$ vertex beacons. A natural question is to see if this result generalizes to $\mathbb{R}^3$. We will prove now that not every orthogonal polyhedron can be covered with vertex beacons by showing an orthogonal polyhedron P such that if we place a vertex beacon on each vertex of P there is a point p not covered by any of the vertex beacons, see Figure 4. We call it the *notched orthoplex*. It consists of a cuboid with six channels, each one of them going across a different face. We attach to each channel a cuboid of the same length. Each attached cuboid is slightly narrower than its channel and is placed at its center, thus creating a notch on that channel, as shown in Figure 4.

Let p be the point at the *center* of P. It is easy to see that the attraction path of p to any vertex beacon of the polyhedron leads to a local minimum generated by a notch on the corresponding channel, as shown in Figure 5.

Since vertex beacons are not enough to cover orthogonal polyhedra, it is

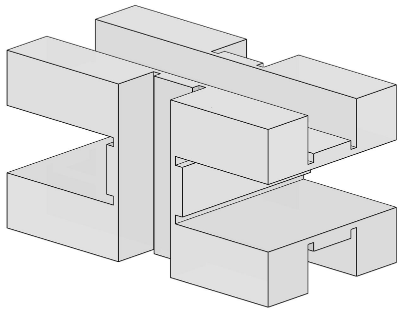


Figure 4: The notched octoplex cannot be covered with vertex beacons.

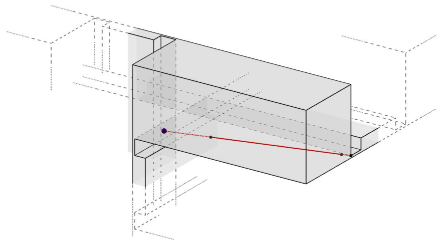


Figure 5: The point in the center of the notched octoplex is unable to reach a beacon placed in any vertex.

natural to study the edge beacon model. It is straightforward to see that if we place an edge beacon at each edge of a polyhedron P (orthogonal or not) these edge beacons always cover P.

We will prove:

Theorem 4. *Let P be an orthogonal polyhedron with e edges. Then $\frac{e}{12}$ edge beacons are always sufficient to cover P.*

Theorem 5. *Let P be an orthogonal polyhedron with e edges. Then $\lfloor \frac{e}{6} \rfloor$ edge beacons are always sufficient to cover both the interior and exterior of P.*

Finally we will prove:

Theorem 6. *There exists a family of orthogonal polyhedra with e edges, such that $\lfloor \frac{e}{21} \rfloor$ edge beacons are necessary to cover their interior.*

References

[1] I. Aldana-Galván, J.L. Álvarez-Rebollar, J.C. Catana-Salazar, N. Marín-Nevárez, E. Solís Villarreal, J. Urrutia, and C. Velarde, Beacon Coverage in Orthogonal Polyhedra. In CCCG 2017.

[2] I. Aldana-Galván, J.L. Álvarez-Rebollar, J.C. Catana-Salazar, M. Jiménez-Salinas, E. Solís-Villareal, and J. Urrutia, Minimizing the Solid Angle Sum of Orthogonal Polyhedra and Guarding them with $\frac{\pi}{2}$-Edge Guards. In Proc. CCCG 2016, pp. 175-181, August 3-5, 2016 Simon Fraser University Vancouver, British Columbia Canada.

[3] S. W. Bae, C.S. Shin, and A. Vigneron, Tight bounds for beacon-based coverage in simple rectilinear polygons. In Latin American Symposium on Theoretical Informatics, pp. 110-122. Springer, 2016.

[4] M. Biro. Beacon-based routing and guarding. PhD thesis, State University of New York at Stony Brook, 2013.

[5] M. Biro, J. Gao, J. Iwerks, I. Kostitsyna, and J. Mitchell. Beacon-based routing and coverage. In 21st Fall Workshop on Computational Geometry, 2011.

[6] M. Biro, J. Iwerks, I. Kostitsyna, and J. S. Mitchell. Beacon-based algorithms for geometric routing. In Workshop on Algorithms and Data Structures, pages 158-169. Springer, 2013.

[7] Gaddum, J., The sums of the dihedral and trihedral angles in a tetrahedron. The American Mathematical Monthly 59(6), pp. 370-371, (1952).

[8] J. Urrutia, "Art gallery and illumination problems", In J.-R. Sack and J. Urrutia, (eds.), *Handbook on Computational Geometry*, North Holland (2000) 973–1127.

Forbidden Configurations in Discrete Geometry

David Eppstein*
Computer Science Department, Univ. of California, Irvine

We review and classify problems in discrete geometry that depend only on the order-type or configuration of a set of points, and that can be characterized by a family of forbidden configurations. These include the happy ending problem, no-three-in-line problem, and orchard-planting problem from classical discrete geometry, as well as Harborth's conjecture on integer edge lengths and the construction of universal point sets in graph drawing. We investigate which of these properties have characterizations involving a finite number of forbidden subconfigurations, and the implications of these characterizations for the computational complexity of these problems.

*Supported in part by the National Science Foundation under Grants CCF-1228639, CCF-1618301, and CCF-1616248.

New crossing lemmas

János Pach[*]

One of the most useful tools in topological graph theory is the so-called *Crossing Lemma* of Ajtai, Chvátal, Newborn, Szemerédi (1982) and Leighton (1983). It states, roughly speaking, that if a graph drawn in the plane has much more edges than vertices, then the number of crossings between its edges is much larger than the number of edges.[1]

Natan Rubin, Gábor Tardos and the speaker discovered a similar phenomenon for families of curves in the plane. If two curves have precisely one point p in common, and at this point they do not properly cross, then p is called a *touching point.* Any other point that belongs to two curves is called an *intersection point.* Let X and T stand for the number of intersection points and touching points, respectively, in a family of n curves, no three of which pass through the same point. If T/n is larger than a fixed constant, then $X \geq \Omega(T(\log\log(T/n))^{1/336})$. In particular, if $T/n \to \infty$, then the number of intersection points is much larger than the number of touching points.[2]

What happens if, instead of the number X of crossing *points*, we want to estimate the number x of crossing *pairs* of curves? Obviously, we have $x \leq X$. In a joint paper with Géza Tóth, it was shown that $x \geq \Omega(T^2/n^2)$. The order of magnitude of this bound is best possible.[3]

References

[1] M. Ajtai, V. Chvátal, M. Newborn, E. Szemerédi, Crossing-free subgraphs. In: Theory and Practice of Combinatorics, North-Holland Mathematics Studies 60, North-Holland, Amsterdam, 1982, 9–12.

[2] J. Pach, N. Rubin, and G. Tardos: Beyond the Richter-Thomassen conjecture, Proc. 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2016, Arlington), SIAM, 2016, 957–968.

[3] J. Pach and G. Tóth: Many touchings force many crossings, arXiv:1706.06829

[*]Ecole Politechnique Fédérale de Lausanne and Rényi Institute, Hungarian Academy of Sciences, P.O.B. 127 Budapest, 1364, Hungary; `pach@renyi.hu`. ; `pach@cims.nyu.edu`.

Characterizing minimal rigidity of square-grid frameworks with holes

Naoki Katoh
Department of Computer Science, Kwansei Gakuin University
naoki.katoh@gmail.com.

1 Introduction

Bolker and Crapo gave a necessary and sufficient condition in their seminal paper [1] for an $m \times n$ square grid framework with some diagonal braces of squares to be infinitesimally rigid. They defined a bipartite graph corresponding to the square grid framework with some diagonal braces such that the infinitesimal rigidity of the framework can be tested by checking the connectivity of the graph.

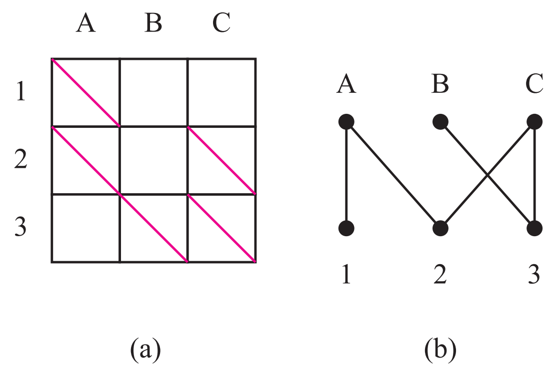


Figure 1: A braced $m \times n$ square grid framework and the corresponding bipartite graph.

In particular, the minimum number of diagonal braces that are necessary and sufficient to make the $m \times n$ square grid framework infinitesimally rigid is $m + n - 1$ (see Figure 1).

Radics and Recski [5] studied the case with holes where the outer boundary of the square grid framework is a simple rectilinear polygon, and long diagonal bars as well as cables can be used. See Figure 2 for the square grid framework with holes.

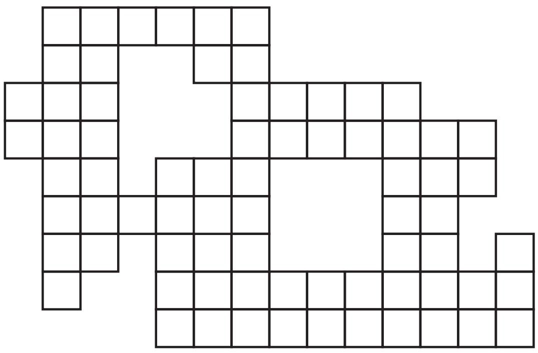


Figure 2: A square grid framework with holes

They showed a lower bound for the number of diagonal bars and cables required to make the framework rigid; this bound matches the one for the case where only short braces (diagonal edges of unit squares) are allowed. However, in this case, they noted that the characterization based on a bipartite graph is no longer valid.

Gáspár, Radics and Recski [3] studied the case with holes where the outer boundary is rectangular, and derived a necessary and sufficient condition in terms of the rank of a certain matrix for an $m \times n$ square grid framework with some diagonal braces of squares to be infinitesimally rigid, The advantage of using the matrix introduced by [3] is that the matrix size is much reduced compared with using an original rigidity matrix, which helps to substantially reduce the running time of checking the rigidity. The paper [3] mentioned that the result can be generalized to the case where the outer boundary is a rectilinear simple polygon. However, the details are not given.

Ito, Kobayashi, Higashikawa, Katoh, Poon and Saumell [4] have recently proposed an algorithm for the bracing problem: given a square-grid framework with holes in which there is no brace, the objective is to add the minimum number of braces which makes the framework infinitesimally rigid.

A square-grid framework is called minimally rigid if the framework is infinitesimally rigid and removing any brace makes the framework infinitesimally flexible. Then the characterization by [1] immediately provides a necessary and sufficient condition for the minimal rigidity of a framework with no holes, however the results by [3, 5] do not provide such a condition for the case with holes. Recently, we give the first necessary and sufficient condition for the minimal rigidity of a square-grid framework with holes [2]. In this talk, we will review results related to the rigidity of square-grid frameworks and will explain our recent result of [2]. This is a joint work with Siu-Wing Cheng, Yuya Higashikawa and Adnan Sljoka.

Acknowledgement

Research was supported by JST CREST Grant Number JPMJCR1402, Japan.

References

[1] E. D. Bolker and H. Crapo. Bracing rectangular frameworks. I. *SIAM Journal on Applied Mathematics*, 36(3) 473–490, 1979.

[2] Siu-Wing Cheng, Yuya Higashikawa, Naoki Katoh, Adnan Sljoka, Characterizing minimal rigidity of square-grid frameworks with holes, *Proc. of the 10th Hungarian-Japanese Symposium on Discrete Mathematics and Applications* (2017), 93-102.

[3] Z. Gáspár and N. Radics and A. Recski. Rigidity of square grids with holes. *Computer Assisted Mechanics and Engineering Sciences*, 6(3-4) 329–335, 1999.

[4] Yoshihiko Ito, Yuki Kobayashi, Yuya Higashikawa, Naoki Katoh, Sheung-Hung Poon, Maria Saumell, Optimally Bracing Grid Frameworks with Holes, *Theoretical Computer Science*, vol. 607, pp. 337-350, 2015,

[5] N. Radics and A. Recski. Applications of combinatorics to statics – rigidity of grids. *Discrete Applied Mathematics*, 123 473-485, 2002.

Oichan and I

Vašek Chvátal
Department of Computer Science & Software Engineering
Concordia University, Montreal

I have been blessed by having a number of wonderful people come into my life. Jin Akiyama is one of them.

Jin and I met in 1979. It was a charmed decade, the nineteen seventies. The Pill was used widely and AIDS lay dormant, patchouli disguised the scent of marijuana, the Rolling Stones released *Sticky Fingers* in 1971 and *Some Girls* in 1978. More to the point, Canadian travel grants were dispensed liberally. "We award prize money", said later a member of the NSERC Grants Selection Committee and continued, "we don't care if he uses it to take his mistress to Rio de Janeiro, as long as he brings back a first-class theorem". Such was the atmosphere where David Avis and I sensed a license to travel around the world and let NSERC pay.

The vehicle for our round-the-world junket was Pan Am Flight 1 from New York to New York, westbound from beginning to end, with a number of stopovers allowed along the way. To justify its expense, I picked my stopovers in places to which I had some professional connection: first Honolulu, where the Institute of Management Sciences held its 24th International Meeting on June 18–22, 1979, then Tokyo, where the IEEE Circuits and Systems Society held its International Symposium on Circuits and Systems on July 17–19, 1979, and further along the trajectory graph theorists in Singapore, Bangkok, Bombay, and Paris. But the three-day meeting in Tokyo did not seem to justify the stay of several weeks in Japan that David and I insisted on. We insisted on staying in Japan for several weeks since our desire to get to know this country was the motivation for the whole trip.

The idea of going to Japan crossed my mind first when I read Koestler's *The Lotus and the Robot:* his venomous sneers at Zen notwithstanding, his portrayal of the 1950s Japan intrigued me. If, even having read it, I still imagined Japanese life as revolving around tea ceremonies, listening to geishas plucking shamisen strings, and watching cherry blossoms float gently to the ground, two films disabused me of such stereotypes. I was fascinated by the cross between Chandler and Kafka in *The Man Without a Map* and I was riveted by the refined perversion of *Odd Obsession.* Now I really wanted to get to know Japan.

But my contacts in Japan were only tenuous, and so one day I phoned Frank Harary in his Ann Arbor office and asked him if he could introduce me to any Japanese graph theorists. He replied that one was sitting right next to him and he handed the receiver to Jin. I am immensely grateful to Frank Harary: He may have saved my life in the summer of 1971 and less than eight years later he introduced me to Jin Akiyama.

And so it came to pass that one fine June evening Jin, his assistant Hiroshi Era, and Hiroshi's wife Akiko waited at Narita to welcome us to Japan. David (nonstop from New York) and I (via Hawaii) landed there twenty-four hours later: oblivious of the International Date Line, I had misinformed Jin about the date of our arrival. In spite of this setback, Jin and Hiroshi got us to a happy reunion in Jin's parents' house in Eifukuchō and, after a reinvigorating nap, the four boys were off to a hostess bar in Kichijōji. Where we met the beautiful convivial Fumiko Yano. To whom we dedicated our joint paper *Balancing signed graphs* nine months later. I still have her self-portrait. And later that night, futons unrolled once more on the tatami floor in Eifukuchō, we slept and slept and slept. And next day Jin took us to our new home away from home, the Sun Route Hotel in Shibuya.

Our first impressions of Japan? Alice in Wonderland does not begin to describe it. The sights and smells and sounds were intoxicating. The elegant tall buildings with a number of bars on each floor and the floors stacked up up up all the way to the indigo heaven flanked by beautiful brightly lit kanji and kana, the occasional shout of rōmaji a reminder of how mysterious the rest was in the aroma of yakitori, shioyaki, yakisoba floating through the din of pachinko parlors and the incessant *irasshaimase* of greeters in the street.In short, Shinjuku san-chōme.

Soon there were five of us carousing around the magic city: in addition to the original team of Jin, Hiroshi, David, myself, there was also Dr. Umemori, a recent medical graduate. Gaijins in Japan realize sooner or later that (with the single exception of 'n') consonants in Japanese cannot stand on their own:

not Vašek, but Basheku and not Chvátal, but Fubātaru. Which goes towards my excuse for parsing our new friend's name as *dokutoru Memory* when he was first introduced to us. Umechan's round face was beaming beatitude and calm well-being. In his presence you felt reassured and safe. "It's all right: I will take knife", he liked to say with a smile that hovered between eerie and comforting when the slightest medical mishap seemed to threaten any of us. His misheard name radiated mystical significance as our humid summer Tōkyō nights were being recorded in eternity behind time warps and black holes. Dr. Memory was looking after us then and is looking after us still.

From the first moment, Jin and I recognized kindred spirits in each other. This intuitive revelation was confirmed by the following years and the confirmation was superfluous. We are brothers, our birthdays 84 days apart, and I like to imagine how, if the circumstances of our births were reversed, Jin could have been me and I could have been him. Speaking of kinship: Jin liked to accost vagrants in the street and, in the manner of Japanese children speaking to strangers, address them as uncle, *oichan.* Eventually he extended the use of this sobriquet to me and I reciprocated, so now I am his oichan and he is mine.

Playing tourist guide to acquaintances and friends is a delicate task. Shepherding them too much may suffocate them and its opposite may leave them feeling neglected. The art is to strike the right balance and Jin Akiyama is one of its undisputed masters. In the beginning, he took us to places and introduced us to people. Then, having settled us in the new environment, he let us run around on our own and luxuriate in unfettered adventures. Like a rocket, he launched us into orbit. Keeping himself in the background, he was ready to spring to our rescue in any emergency and we knew it. Thank you, Jin.

We may have been intrepid in the early days, David and I, but our digestive systems were less so. Fortunately, compassionate and brilliant mamasans understood our exotic dietary needs and supplied us with fatty foods as a matter of course. One of these angels, the owner of an open-air snack bar, was getting on in years and was nearly blind. One day, as she was preparing an omelet for us and stirring it with her long chopsticks while she tried to peer at a small TV in the corner, a cockroach appeared on the wooden counter and began marching toward the smell. This was not one of your dainty *aburamushi,* this was a big mother *gokiburi.* Fascinated, David and I watched the animal's progress to the spot directly above the frying pan, where it waved its atennae voluptuosly in the intoxicating aroma, teetered momentarily on the edge and then plunged straight into the pan. The sound of sizzling eggs and vegetables differs from the sound of a sizzling cockroach. David and I can attest to this and so could the mama. Unerringly, without taking her eyes off the TV screen, she grasped the insect in her chopsticks and neatly dropped it to the ground. Need I say that we were impressed?

There is an art to making people feel at home in a foreign country. *You belong here,* you say to them, implicitly at least, *you belong here and all this is yours. Of course, you will come across people who disagree. Ignore such idiots. They are beneath us. We know better, you and I.* This is how Claude Berge gave me France. This is how Jin Akiyama gave me Japan.

One evening during my first Japanese summer, Jin took me to a restaurant owned by a friend of his. Little by little, the customers thinned out until there were none and the owner locked the door. Now only he, Jin, I, and two beautiful waitresses remained. With the lights dimmed and sandalwood wafting in gently, the two enchantresses sat me between them and sang to me *Ue o Muite Arukō,* the song that, with Czech lyrics, had been one of my favourites fifteen years earlier, four years before the Russian invasion of Czechoslovakia. I knew then, quite rationally and quite lucidly, that I was in paradise. I knew that I had come home.

I never knew where the powerful magic that Japan attracted me with was coming from. When I was a child, a small tin statue of Buddha, painted black, stood watch over my sleep night after night. My maternal grandfather had brought it from Yokohama on his return from the First World War via the long route —Siberia, Japan, North America. Growing up, I was hopelessly inept at almost all sports. All except one, to which I took like fish to water and at which I excelled. That sport was judo. At the age of eighteen, I experienced a *satori* just hours before finding out from a literary magazine that there was something called Zen Buddhism.

"You will die in Japan," Jin said to me once and I was happy to see that he understood the intensity of my love affair with our country. I did my best to make his prophesy come true when I went on an alcoholic binge during the

Kyōto conference celebrating our sixtieth birthdays, but that was accidental, I did not do it on purpose, and it is a different story altogether. Except that I am glad of the opportunity to say now what I meant to say then and to add that I am sorry to have screwed up ten years ago.

There is an art to making people feel at home in a foreign country and Jin Akiyama is one of its undisputed masters. For years I had on my key ring, next to the key to my Montreal apartment and the key to my McGill office, a slender key stamped with the Lion's Mansion logo. It was the key to Jin's Nishi Eifuku apartment. Just looking at it back in Motreal warmed my heart and made me feel safe. I would get off the plane at Narita and hop on the train — Ueno, Shinjuku, Meidaimae — and the key fit, of course, and here I was, back in the familar haven. After a thorough wash outside the ofuro, with much hot water sloshed around on the floor (none of the Western shower nonsense for me, thank you) and a slow sensuous soak up to my neck in the ofuro, a restorative nap on the futon rolled out on the tatami and then off to Golden Gai. With renewed thanks to Jin, silent ones if he happened to be away from Tōkyō.

In Confucianism, Jin's first name denotes the ethical constant of human kindness. The name fits him. He is compassionate and he is kind-hearted. What redeems him from appearing sanctimonious is his gift for laughing at himself and his voracious pursuit of fun .

About to return to Japan after a five-year hiatus, I was apprehensive that reality might not live up to my reminiscences. I could not have been more mistaken: Tōkyō proved even better than my memories of it. It felt even more exhilirating, even more welcoming, even more soothing. Blueprint for Utopia.

On this first return to Japan, I made a survey of our old haunts and could not find the snack bar where the cockroach incident had taken place. The neighbourhood was rebuilt and none of the people I asked knew what happened to the owner. That night, soaking in Jin's ofuro, I meditated wistfully on what Japan teaches us. The transience of life and its ephemeral beauty. Gokiburi mama gone the way of all flesh and cherry blossoms.

Jin Akiyama is an endearing figure. Rather than relying on help from rich relatives, he left home at the age of around 20 to earn his own livelihood, including the university tuition fees. He worked as a tutor and as a computer operator in IBM Japan at night during terms and he worked as a guard at a mountain cottage near Japan Alps during long holidays. This was in keeping with the principles of simplicity, fortitude, and self-reliance, by which Jin's relatives on his father's side lived. Among these relatives, there were scientists such as Dr. Shōten Oka, one of the pioneers of biorheology; Jin's father was a physicist and studied acoustics. They held academic achievements in high regard. On Jin's mother's side, there were many relatives who succeeded in business. His grandfather was the president of Japan Victor Co. Ltd and his uncle-in-law was the Vice Minister of Finance of Japan. With their privileges and strong connections to high society, these relatives tended to regard wealth and status as important. Such was the atmosphere where Jin Akiyama came to disdain the snobbery of class distinctions and inherited entitlements.

Jin Akiyama is a romantic figure. When he ventured to Ann Arbor in 1977 in order to study graph theory with Frank Harary, it was in the best tradition of Meiji scholars seeking knowledge throughout the world. In the best tradition of Meiji scholars, he brought the knowledge back to Japan and then he built on it. In the summer of 1979, on the heels of his two-year Ann Arbor apprenticeship, he organized a week-long graph theory seminar in Nikkō with a number of Japanese participants as well as the foreign contingent of David Avis, Chung Laung (Dave) Liu, András Recski, and myself. (Its strenuous mathematical program was lightened by Dr. Umemori's folk dance exhibition.) Afterward, Jin kept the interest of Japanese mathematicians in graph theory alive by organizing a regular Saturday seminar in Tōkyō, to which he invited many foreign speakers. At the same time, he served as an editor of the *Journal of Graph Theory* and this experience helped him in his next big step, the creation of *Graphs & Combinatorics.* Jin was the moving force behind this project and, when the new journal was launched in December 1985, he became its first Managing Editor. Then he organized the First Japan Conference on Graph Theory, which took place on June 1–5, 1986 in Hakone. With around a half of its 200 participants coming from foreign countries (and including luminaries such as Paul Erdős and Ron Graham), this event was unprecedented in the context of Japanese mathematics. In less than a decade after meeting Frank Harary, Jin Akiyama made graph theory flourish in Japan.

Jin Akiyama is a glamourous figure. His meteoric rise to stardom is the stuff legends are made of. In 1972, he joined the faculty of the prestigious medical school Nippon Ika (the Japanese respect mathematics and insist that all medical doctors be thoroughly trained in logical thinking) and he also took on a part-time teaching job at the Sundai Preparatory School. At Sundai, students could choose the teacher whose courses they wanted to attend and teachers were paid by the number of students attending their courses. Jin's charisma, his infectious energy, and his penchant for the outrageous (he made the *risqué* condom problem a standard part of his syllabus) quickly made him the most popular mathematics teacher. Soon he was assigned Sundai's biggest lecture room, an auditorium that held 450 people (those sitting in the back rows had to use binoculars) and his lectures in this room were simultaneously followed by students in Sundai's satellite campuses all over Japan via closed-circuit television. This alone would have been enough to make Jin a wealthy man.

But there was more. Preparatory schools train students for university entrance examinations, which are regarded as the most important hurdle in the path of outstanding education. At Sundai Yobikō, one of the oldest and the best of these schools, Jin taught some 5000 students every year. Most of them were good students, who later formed the crème de la crème of Japanese society. This alone would have been enough to make Jin famous.

But there was more still. In 1991, just before the end of his twenty-year Sundai stint, Jin got affiliated with Japan's national public broadcasting organization NHK and in that single year produced for them 30 television shows dealing with mathematics. Audience rating of this program was very high, and so Jin's career at NHK continued for the next twelve years. In this program, he performed as a lecturer and edited its textbooks that were sold all over Japan, ('Turn to page 77', says Jin and thousands of pages are flipping from Hokkaidō to Okinawa.) So Jin became a TV star and his name became a household word.

The summer of 1979. A gray misty Tōkyō morning and ravens cawing overhead as we were returning home after yet another night of debauchery, "I want Chinese noodles," announced Jin and off we went in search of a ramen stand. When I woke up later, I marvelled at his wisdom, full of gratitude for this cushion against hangover. “Look, Vašek, your father,” Jin cried out one of these mornings and pointed to a homeless man sleeping in a cardboard box. He must have liked this theme because, gradually, revel by revel, he developed it into a comedian’s routine. “Come speak to your father, Vašek. Why aren’t you speaking to him? Are you ashamed of your father? You are making him so sad!” But once he suddenly changed his monologue. “Look, here is my father,” he exclaimed when we saw another homeless man sitting on the sidewalk. “He has no money, my father, and he is worried. Don’t worry about money, my father! I will make much money, because I am a great educationalist!” Later on, I wondered: did Jin have his brilliant Sundai/NHK career all mapped out at that moment? James Clavell’s *Shōgun* came out in 1975. Finding parallels between its archetype of the sophisticated strategist Yoshi Toranaga and the real Jin Akiyama may have been pardonable.

On the rare occasions when I feel a pang of jealousy over my brother’s success, I like to reassure myself that I have not had the same opportunity: where else would you find society that holds mathematics in high esteem and the highly competitive system of university entrance examinations with its attendant preparatory schools? Only in Japan, I say when I try to comfort myself. But that excuse is flimsy: who knows what fireworks Jin would come up with in a different environment.

Once Jin took me to the taping of a television game show. The two teams — red against white, of course — were deployed in two buses and the buses were suspended above water from large cranes. Penalty for incorrect answers consisted of lowering the bus below the water level and the TV cameras faithfully recorded the contestants’ struggle till the last string of air bubbles rising to the surface before reprieve was granted and the dripping vehicle slowly cranked up again. Celebrities formed the panel of judges: Jin Akiyama, a bantamweight boxing champion, a *tanka* poet, two charming porn actresses. *Only in Japan.* Eat your heart out, NBC.

Having a celebrity oichan means having his face confront one from TV commercials for Tokyo Gas and from posters advertising Suntory Premium Malt’s. When I brought Jin to a Japanese restaurant in Montreal for the first time, the waitress called out *Akiyama sensei!* in surprise. Several times I answered a question in a Tōkyō bar and told the stranger that I was not staying in a hotel, I was staying in a friend's apartment. This information elicited no response to speak of, but the atmosphere changed dramatically if I revealed the friend's name. "What? Are you trying to tell me that you know Akiyama? Who do you think you are kidding?" The Japanese, too, can be quite direct when confronted with obvious delusions of grandeur.

Once a magnificient mamasan closed her shop for the night and, as is the mamasans' wont, took me to another drinking establishment. Which was presided over by a vigorous granny dispensing wisdom to her clientele, mainly young girls coming there for advice on their love lives. My memory of faces is atrocious: as a child, I was often scolded by my parents for failing to greet our neighbours, whom I simply did not recognize. My memory of faces is atrocious, but something kept tugging at it after I returned to Jin's Nishi Eifuku that night. Something kept tugging at it until I gave in and went back to observe the granny some more. "Excuse me, but didn't you run an open-air snack bar near Yasukuni dōri just behind Kabukichō?" I asked eventually and she said yes, she did. "But," I continued flabbergasted, Alice in Wonderland once again, "your eyesight was very bad then, wasn't it?" and she confirmed this, too, and went on to tell me how, at the age of 70, she got fed up with her current way of life, and so she divorced her husband, got a cataract surgery, and moved her business here.

That night, soaking in Jin's ofuro, I meditated on what Japan really teaches us. Zest for life. Joyous energy. Not falling pompously for trite clichés.

Jin Akiyama gave me Japan, that moveable feast, and he gave me his friendship. Thank you, Oichan. Thank you, Jin. Thank you for these precious permanent presents.

Acknowledgment

I am grateful to David Avis, Elizabeta Stoinová, and Markéta Vyskočilová for their insightful comments that led to improvements of my manuscript.

20 Years of JCDCGGG

Erik D. Demaine*

Abstract. I will reminisce in the past 20 years of JCDCGGG, in particular reviewing selected papers over the years, their remaining open problems, and their influence on computational geometry, graphs, and games. I have been attending this conference for more than half of my life — at my first JCDCG'98, I was 17 years old, while the oldest participant, Gisaku Nakamura, was 71. This conference has led to many amazing collaborations, and defined its own research direction that combines mathematics, computer science, public outreach, and recreation. In particular, it shaped my own career goals: solve problems while having fun.

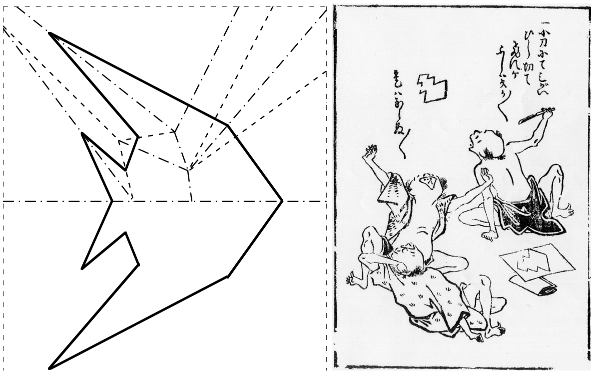


Figure 1: "Folding and Cutting Paper" [JCDCG'98]

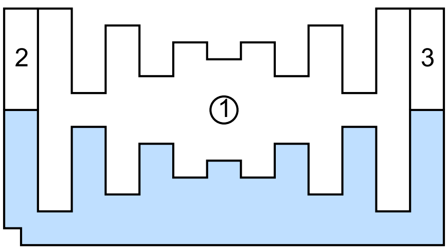


Figure 2: "Flipturning Polygons" [JCDCG 2000]

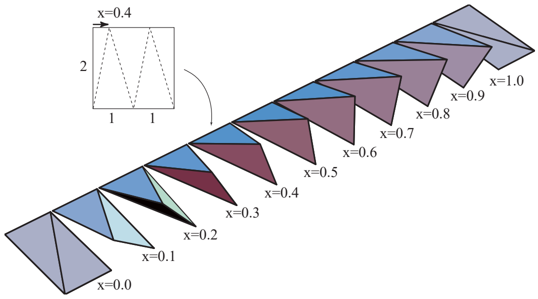


Figure 3: "Enumerating Foldings and Unfoldings between Polygons and Polytopes" [JCDCG 2000]

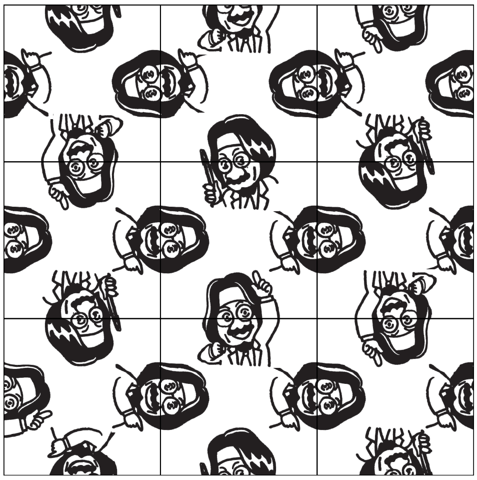


Figure 4: "Jigsaw Puzzles, Edge Matching, and Polyomino Packing: Connections and Complexity" [KyotoCGGT 2007]

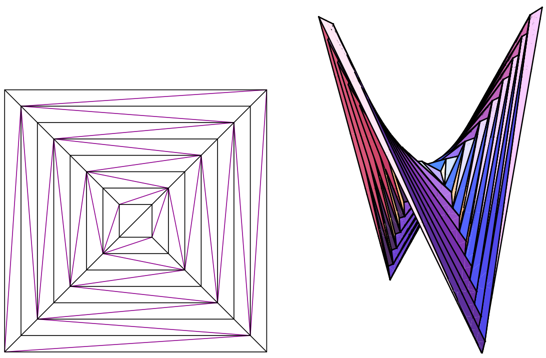


Figure 5: "(Non)existence of Pleated Folds: How Paper Folds Between Creases" [JCCGG 2009]

*MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar St., Cambridge, MA 02139, USA, edemaine@mit.edu

Balanced subdivisions of three colored point sets in the plane

Mikio Kano
Ibaraki University, Hitachi, Ibaraki, Japan
mikio.kano.math@vc.ibaraki.ac.jp
http://gorogoro.cis.ibaraki.ac.jp

This is a joint work with Toshinori Sakai and Jorge Urrutia.

1 Introduction

Let R, B and G be a set of red points, a set of blue points and a set of green points in the plane, respectively, such that $R \cup B \cup G$ is in general position. A path of order n is denote by P_n, and a path drawn in the plane is called an *properly colored path* if every edge is a straight line segment and any two adjacent vertices have distinct colors. A P_n*-covering* of $R \cup B$ or $R \cup B \cup G$ is a set of disjoint non-crossing properly colored paths P_n that cover $R \cup B$ or $R \cup B \cup G$. A generalization of the following two theorems to three colored point sets is our motivation.

Theorem 1 (Kaneko, Kano and Suzuki [4]) *Let g and h be non-negative integers. If n is an even integer such that $2 \leq n \leq 14$, then for any given $(n/2)g$ red points and $(n/2)g$ blue points in the plane in general position, there exists a P_n-covering. If n is an odd integer such that $3 \leq n \leq 11$, then for any given $\lfloor n/2 \rfloor g + \lceil n/2 \rceil h$ red points and $\lceil n/2 \rceil g + \lfloor n/2 \rfloor h$ blue points in the plane in general position, there exists a P_n-covering.*

Moreover, for any integer n such that $n = 13$ or $n \geq 15$, there exists a configuration with $\lfloor n/2 \rfloor$ red points and $\lceil n/2 \rceil$ blue points for which there exists no P_n-covering.

In order to prove Theorem 1, the next theorem is essentially important.

Theorem 2 (Kaneko, Kano and Suzuki, [4]) *Let g and h be non-negative integers. If n is an even integer, then for any given $(n/2)g$ red points and*

$(n/2)g$ blue points in the plane in general position, there exists a subdivision of the plane into g disjoint convex regions $X_1, X_2, \ldots, X_g$ such that every X_i contains exactly $n/2$ red points and $n/2$ blue points. If n is an odd integer, then for any given $\lfloor n/2 \rfloor g + \lceil n/2 \rceil h$ red points and $\lceil n/2 \rceil g + \lfloor n/2 \rfloor h$ blue points in the plane in general position, there exists a subdivision of the plane into $g + h$ disjoint convex regions $X_1, X_2, \ldots, X_g, Y_1, Y_2, \ldots, Y_h$ such that every X_i contains exactly $\lfloor n/2 \rfloor$ red points and $\lceil n/2 \rceil$ blue points and every Y_j contains exactly $\lceil n/2 \rceil$ red points and $\lfloor n/2 \rfloor$ blue points.

We consider the following problems in general.

Problem 3 *(Even case) Assume that $|R| + |B| + |G| = 2kn$, where k and n are positive integers. If $|R|, |B|, |G| \leq kn$, then the plane is subdivided into disjoint n convex regions $X_1, X_2, \ldots, X_n$ so that every X_i contains precisely $2k$ points and contains at most k points with the same color.*

Problem 1 with $k = 1$ holds, and hamburger theorem plays essential role in this proof. If $k = 2$, then every convex region X_i contains 4 points, and contain at most two points with the same color.

Problem 4 *(Odd case) Assume that $|R| + |B| + |G| = (2k + 1)n$, where k and n are positive integers. If $|R|, |B|, |G| \leq (k + 1)n$, then the plane is subdivided into disjoint n convex regions $X_1, X_2, \ldots, X_n$ so that every X_i contains precisely $2k + 1$ points and contains at most $k + 1$ points with the same color.*

Problem 2 with $k = 1$ holds as shown below (Theorem 6). On the other hand, Problem 2 with $k = 2$ seems to be correct, and partial proof is obtained, but not complete (Theorems 8 and 9).

2 Some results

Lemma 5 *If $|R| + |B| = 3n$, $|R| \leq 2n$ and $|B| \leq 2n$, then the plane can be subdivided into n disjoint convex regions $X_1, X_2, \ldots, X_n$ so that every X_i contains either two red points and one blue point or one red point and two blue points.*

Theorem 6 *If $|R| + |B| + |G| = 3n$, $|R| \leq 2n$, $|B| \leq 2n$ and $|G| \leq 2n$, then the plane can be subdivided into n convex regions $X_1, X_2, \ldots, X_n$ so that every X_i contains precisely three points and contains at most two points with the same color (see Figure 1).*

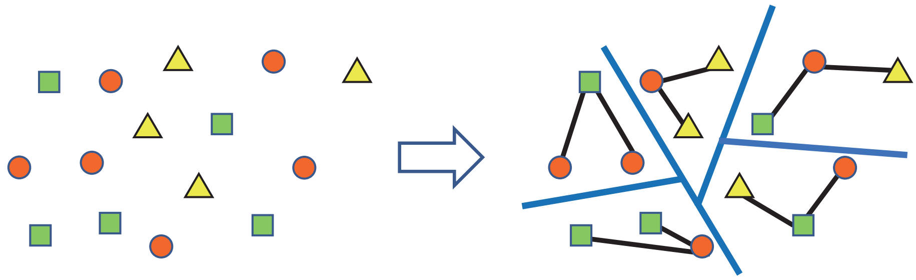


Figure 1: A P_3-covering of $R \cup B \cup G$.

Lemma 7 *If $|R| + |B| = 5n$, $|R| \leq 3n$ and $|B| \leq 3n$, then the plane can be subdivided into n disjoint convex regions $X_1, X_2, \ldots, X_n$ so that every X_i contains either three red points and two blue points or two red points and three blue points.*

Theorem 8 *If $|R| + |B| + |G| = 3n$, $|R| \leq 2n$, $|B| \leq 2n$ and $|G| \leq 2n$, then the plane can be subdivided into n convex regions $X_1, X_2, \ldots, X_n$ so that every X_i contains precisely three points and contains at most two points with the same color.*

Theorem 9 *Let μ_1, μ_2, μ_3 be three continuous measures on the plane $\mathbf{R}^2$. Let $a = \mu_1(\mathbf{R}^2)$, $b = \mu_2(\mathbf{R}^2)$ and $c = \mu_3(\mathbf{R}^2)$. If*

$$0 < \ a, \ b, \ c \ \leq \frac{2}{5}(a + b + c), \tag{1}$$

then there exists a line l such that each half plane H defined by l satisfies

$$\mu_1(H) + \mu_2(H) + \mu_3(H) \geq \frac{4}{5}\min\{a, b, c\}, \tag{2}$$

and

$$0 \leq \mu_1(H), \ \mu_2(H), \ \mu_3(H) \leq \frac{3}{5}\big(\mu_1(H) + \mu_2(H) + \mu_3(H)\big). \tag{3}$$

In order to prove our results, we need the following theorems.

Theorem 10 (Balanced subdivision Theorem, [1], [2], [7]) *Assume that $|R| = rn$ and $|B| = bn$. Then the plane can be subdivided into disjoint convex regions $X_1, X_2, \ldots \cup X_n$ so that every X_i contains exactly r red points and b blue point*

Theorem 11 (Kaneko, Kano, and Suzuki [4]) *Let $s \geq 1$, $g \geq 0$ and $h \geq 0$ be integers such that $g + h \geq 1$. Assume that $|R| = (s+1)g + sh$ and $|B| = sg+(s+1)h$. Then, there exists a subdivision $X_1 \cup \cdots \cup X_g \cup Y_1 \cup \cdots \cup Y_h$ of the plane into $g+h$ disjoint convex polygons such that every X_i $(1 \leq i \leq g)$ contains exactly $s+1$ red points and s blue points and every Y_j $(1 \leq j \leq h)$ contains exactly s red points and $s+1$ blue points*

Theorem 12 (Kano and Kynčl, [5]) *Assume that $|R| + |B| + |G| = 2n$ and $|R|, |B|, |G| \leq n$, where $n \geq 1$ is an integer. Then there exists a line ℓ passing through no point of $R \cup B \cup G$ such that each open half-plane H_i determined by ℓ satisfies (i) H_i contains $2n_i$ points and (ii) $|H_i \cap R|$, $|H_i \cap B|$, $|H_i \cap G| \leq n_i$, where $n_i \geq 1$ for $i \in \{1, 2\}$ and $n_1 + n_2 = n$.*

Theorem 13 (Borsuk-Ulam theorem, Theorem 2.1.1 of [6]) *Let $f : S^n \to \mathbf{R}^n$ be a continuous mapping. If $f(-\boldsymbol{u}) = -f(\boldsymbol{u})$ for all $\boldsymbol{u} \in S^n$ (i.e., f is antipodal), then there exists a point $\boldsymbol{v} \in S^n$ such that $f(\boldsymbol{v}) = \mathbf{0} = (0, \ldots, 0)$.*

References

[1] S. Bespamyatnikh, D. Kirkpatrick, and J. Snoeyink, Generalizing ham sandwich cuts to equitable subdivisions, *Discrete and Computational Geometry*, **24** (2000), 605-622.

[2] H. Ito, H. Uehara, and M. Yokoyama, 2-dimensional ham-sandwich theorem for partitioning into three convex pieces, *Discrete Comput. Geom.* LNCS, **1763** (2000), 129-157.

[3] A. Kaneko and M. Kano, Discrete geometry on red and blue points in the plane — A survey, *Discrete and Computational Geometry*, Algorithms and Combinatorics **25**, Springer (2003) 551-570.

[4] A. Kaneko, M. Kano and K. Suzuki, Path Coverings of Two Sets of Points in the Plane *Contemporary Mathematics series of AMS; Towards a theory of geometric graphs*, edited by J. Pach, Vol. 342 (2004) 99-111.

[5] M. Kano and J. Kynčl, The hamburger theorem, *Computational geometry; theory and applications*, to appear.

[6] J. Matoušek, *Using the Borsuk-Ulam Theorem*, Springer (2003)

[7] T. Sakai, Balanced Convex Partitions of Measures in R^2, *Graphs and Combinatorics*, **18** (2002), 169–192.

Survey
Miscellaneous Properties of Polyhedral Nets
— Tessellability, Reversibility and Foldability —

Jin Akiyama
Tokyo University of Science
ja@jin-akiyama.com

Many results on tessellability, reversibility and foldability of planar figures, in particular of polyhedral nets, have been obtained recently. In this talk, changes of techniques adopted for analyzing these properties of planar figures, such as superimposition method of tilings, lattice adjustment method and interchange method of trunks, are elaborated. Moreover, mutual relations among these results are also exposed. Most importantly, this presentation focuses on the relevant relations connecting these properties, which were studied independently at the beginning of the research. In addition, many research problems and conjectures are presented.

References

(1) T. Abbott, S. Abel, D. Charlton, E. Demaine, M. Demaine, S. Kominers, *Hinged dissections exist*, Discrete and Computational Geometry 47 (1), 2012, 150-186.

(2) Jin Akiyama, Kiyoko Matsunaga, *Only Isotetrahedra Can Be Stampers,* to appear in Math. Mag.

(3) Jin Akiyama, Kiyoko Matsunaga, *Unfoldings of an Envelope,* to appear in European J. Combin.

(4) Jin Akiyama, Kiyoko Matsunaga, *Generalization of Haberdasher's Puzzle,* Discrete Comput. Geom. 58 (2017), no. 1, 30-50, 52B45 (52C20).

(5) Jin Akiyama, Kiyoko Matsunaga, *Reversibility and foldability of Conway tiles,* Comput. Geom. 64 (2017), 30-45, 52C20.

(6) Jin Akiyama, Stefan Langerman, Kiyoko Matsunaga, *Reversible nets of polyhedra,* Discrete and computational geometry and graphs, 13-23, Lecture Notes in comput. Sci., 9943, Springer, Cham, 2016, 52B70.

(7) Jin Akiyama, Kiyoko Matsunaga, *Treks into intuitive geometry—The world of polygons and polyhedra,* Springer, Tokyo, 2015, xv+425 pp. ISBN: 978-4-431-55841-5; 978-4-431-55843-9, 52-01 (00A 66).

(8) Jin Akiyama, Hyunwoo Seong, *A criterion for a pair of convex polygons to be reversible,* Graphs Combin. 31 (2015), no. 2, 347-360, 52B45 (52C20).

(9) Jin Akiyama, David Rappaport, Hyunwoo Seong, *A decision algorithm for reversible pairs of polygons,* Discrete Appl. Math. 178 (2014), 19-26, 52B45.

(10) Jin Akiyama, Ikuro Sato, Hyunwoo Seong, *Tessellabilities, reversibilities, and decomposabilities of polytopes (a survey),* Geometric science of information, 215-223, Lecture Notes in Comput. Sci., 8085, Springer, Heidelberg, 2013, 52C20.

(11) Jin Akiyama, Ikuro Sato, Hyunwoo Seong, *On reversibility among parallelhedra, essellabilities,* Computational geometry, 14-28, Lecture Notes in Comput. Sci., 7579, Springer, Cham, 2011, 52B45.

(12) Jin Akiyama, Takayasu Kuwata, Stefan Langerman, Kenji Okawa, Ikuro Sato, Geoffrey C. Shephard, *Determination of all tessellation polyhedra with regular polygonal, faces,* Computational geometry, graphs and applications, 1-11, Lecture Notes in Comput. Sci., 7033, Springer, Heidelberg, 2011, Indexed.

(13) Jin Akiyama, Chie Nara, *Chromatic numbers of specified isohedral tilings,* Computational geometry and graph theory, 14-24, Lecture Notes in Comput. Sci., 4535, Springer, Berlin, 2008, 05B45 (52B05).

(14) Jin Akiyama, Midori Kobayashi, Gisaku Nakamura, *Dudeney transformation of normal tiles,* Computational geometry and graph theory, 1-13, Lecture Notes in

Comput. Sci., 4535, Springer, Berlin, 2008, 52C15 (05B45).

(15) Jin Akiyama, *Tile-makers and semi-tile-makers,* Amer. Math. Monthly, 114 (2007), no. 7, 602-609, 52C20 (05B45 52A10 52B45).

(16) Jin Akiyama, Chie Nara, *Developments of polyhedra using obliqui coordinates*, J. Indones. Math. Soc. 13 (2007), no. 1, 99-114, 52B55 (51M04 51M20 52C20).

(17) Jin Akiyama, Koich Hirata, Midori Kobayashi, Gisaku Nakamura, *Convex developments of a regular tetrahedron,* Comput. Geom. 34 (2006), no. 1, 2-10, 52B55 (51M04 51M20 52C20).

(18) Jin Akiyama, Gisaku Nakamura, *Foldings of regular polygons to convex polyhedra, I. Equilateral triangles,* Combinatorial geometry and graphtheory, 34-43, Lecture Notes in Comput Sci., 3330, Springer, Berlin, 2005, 52A10 (52B45).

(19) Jin Akiyama, Koich Hirata, Mari-Jo P. Ruiz, Jorge Urrutia, *Flat 2-foldings of convex polygons,* Combinatorial geometry and graph theory, 14-24, Lecture Notes in Comput. Sci., 3330, Springer, Berlin, 2005, 52A10 (52B45).

(20) Jin Akiyama, Koich Hirata, *On convex developments of a doubly-covered square,* Combinatorial geometry and graph theory, 1-13, Lecture Notes in Comput. Sci., 3330., Springer, Berlin, 2005, 52A10 (52B45).

(21) Jin Akiyama, Ferran Hurtado, Criel Merino, Jorge Urrutia, *A problem on hinged dissections with colours,* Graphs Combin. 20 (2004), no. 2, 145-159, 52 C20 (05B45 52B45).

(22) Jin Akiyama, Gisaku Nakamura, *Congruent Dudeney dissections of polygons all the hinge points on vertices of the polygon,* Discrete and computational geometry, 14-21, Lecture Notes in Comput. Sci., 2866, Springer, Berlin, 2003, 52B45 (05B25).

(23) Jin Akiyama, Gisaku Nakamura, *Congruent Dudeney dissections of triangles and convex quadrilaterals—all hinge points interior to the sides of the polygons,* Discrete and computational geometry, 43-63, Algorithms Combin., 25, Springer, Berlin, 2003, 52B45.

(24) Jin Akiyama, Gisaku Nakamura, *Foldings of regular polygons to convex polyhedra, II. Regular pentagons,* J. Indones. Math. Soc. 9 (2003), no. 2, 89-99, 52B45.

(25) Jin Akiyama, Gisaku Nakamura, *Congruent Dudeney dissections of polygons, I. All the hinge points are vertices of the polygon,* The Ninth Quadrennial International Conference on Graph Theory, Combinatorics, Algorithms and Applications, 8 pp., Electron. Notes Discrete Math., 11, Elsevier Sci. B. V., Amsterdam, 2002, 52B45 (05B25).

(26) Jin Akiyama, Gisaku Nakamura, *Dudeney dissections of polygons and polyhedrons — a survey,* Discrete and computational geometry, Tokyo, 2000, 1-30, Lecture Notes in Comput. Sci., 2098, Springer, Berlin, 2001, 52B45.

(27) Jin Akiyama, Gisaku Nakamura, *Dudeney dissections of polygons,* Discrete and computational geometry, Tokyo, 1998, 14-29, Lecture Notes in Comput. Sci., 1763, Springer, Berlin, 2000, 68U05 (52B45).

(28) Erik D. Demaine and Joseph O'Rourke, *Geometric Folding Algorithms*, Cambridge University Press, 2007.

(29) Henry E. Dudeney, *The Canterbury Puzzles and Other Curious Problems*, W. Heinemann, 1907.

(30) Greg N. Frederickson, *Dissections: Plane & Fancy*, Cambridge University Press, 1997.

(31) Greg N. Frederickson, *Hinged Dissections: Swinging & Twisting*, Cambridge University Press, August 2002.

(32) Harry Lindgren, *Geometric Dissections*, Princeton, N.J.: D. Van Nostrand Company, 1964.

(33) S. Langerman and A. Winslow, *A complete classification of tile-makers*, In abstracts from the 18th Japan Conference on Discrete and Computational Geometry and Graph, Kyoto, Japan, September 2015.

(34) Joseph O' Rourke, *How to fold it,* Cam−bridge University Press, 2011.

Mobile Robot Search: Problems and Results

Evangelos Kranakis[1,2]

[1] School of Computer Science, Carleton University, Ottawa, Ontario, Canada..
[2] Research supported in part by NSERC Discovery grant.

Abstract. We consider the problem of searching an underlying geometric domain for a treasure. There are n mobile communicating robots, of which at most f are faulty, and the remaining $n - f$ are reliable. The treasure is placed on the geometric domain at a location unknown to the robots. Reliable robots can find the treasure when they reach its location, but faulty robots either cannot detect the treasure (crash faulty) or may maliciously report a wrong location (Byzantine faulty). Our goal is to design (collaborative) search algorithms minimizing the competitive ratio, represented by the worst case ratio between the time of arrival of the first reliable robot at the treasure, and the distance from the source to the treasure. The faults considered are crash and Byzantine. We present several recent results that illuminate tradeoffs on the impact of fault tolerance and communication on search time and discuss related open problems.

A Memory of the Late Professor Narong Punnim

Wanida Hemakul[1], Ratinan Boonklurb[2] and Jinnadit Laorpaksin[3]

Abstract

This talk is dedicated to Professor Narong Punnim who made an important contribution to JCDCG[3]. Professor Narong's biography and some of his main research will be presented.

[1](retired professor) Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Bangkok 10330 Thailand (email: hewanida@gmail.com)

[2]Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Bangkok 10330 Thailand (email: ratinan.b@chula.ac.th)

[3]Department of Curriculum and Instruction, Faculty of Education, Chulalongkorn University, Bangkok 10330 Thailand (email: ljinnadit@hotmail.com)

Cookie Clicker

Erik D. Demaine[*] Hiro Ito[†] Stefan Langerman[‡] Jayson Lynch[*]
Mikhail Rudoy[§] Kai Xiao[*]

Cookie Clicker[1] is a popular online incremental game where the goal of the game is to generate as many cookies as possible. In the game, you can click on a big cookie icon to bake a cookie, which we model as an initial cookie generation rate. You can also use the cookies you have generated as currency to purchase various items that increase your cookie generation rate. In this paper, we analyze strategies for playing Cookie Clicker optimally. While simple to state, the game gives rise to interesting analysis involving ideas from NP-hardness, approximation algorithms, and dynamic programming.

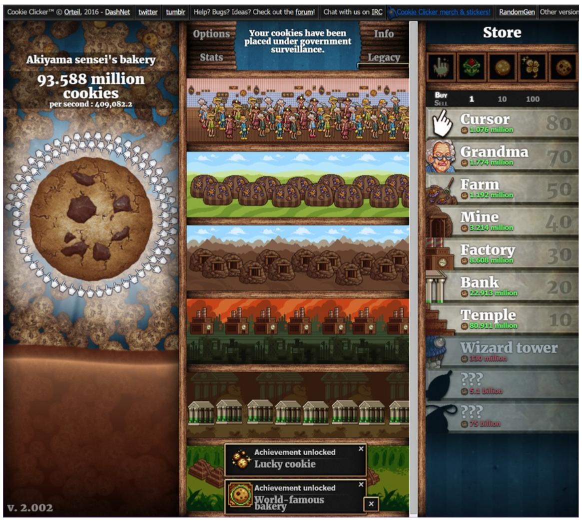


Figure 1: Screenshot of Cookie Clicker.

Each cookie-generating item in this game can be purchased multiple times, but after each item purchase, the item's cost will increase at an exponential rate, given by $C_n = C_1 \cdot \alpha^{n-1}$, where C_1 is the cost of the first item and C_n is the cost of the nth item. In the actual game, $\alpha = 1.15$. There is no real end condition in the game, but in this paper we have two possible end conditions: reaching a certain number M of cookies or reaching a certain cookie generation rate R.

Cookie Clicker falls into a broader class of popular online games called "incremental" games, in which the primary mechanic of the game is acquiring income and spending that income on income generators in order to acquire even more income. Some of the other well-known games in this genre include Adventure Capitalist, Cow Clicker, Clicker Heros, Shark Souls, Kittens Game, Egg Inc., and Sandcastle Builder (Based around the xkcd comic Time, number 1190).

Models. In most of this paper, we will assume that you start with 0 cookies and that the initial cookie generation rate from clicking on the big cookie icon is 1. We will describe each item by a tuple (x, y, α), where x denotes how much the item will increase your cookie generation rate, y denotes the initial cost of the item, and α denotes the multiplicative increase in item cost after each purchase. The case where $\alpha = 1$ for every item is a special case called the *fixed-cost* case. The goals of the game is to find the optimal sequence and timing of item purchases that minimizes some objective function.

There are multiple possible objective functions that we could want to optimize for, but we will focus on the following two:

1. Reaching M cookies in as little time as possible

[*]CSAIL, Massachusetts Institute of Technology
[†]University of Electro-Communications
[‡]Directeur de Recherches du F.R.S.-FNRS, Université Libre de Bruxelles
[§]Google Inc. Work completed at MIT CSAIL.
[1]`http://orteil.dashnet.org/cookieclicker/`

2. Reaching a generation rate of R in as little time as possible

Our analysis of various versions of Cookie Clicker gives rise to interesting and varied results; refer to Table 1. First, we present some general results, such as the fact that the optimal strategy involves a "Buying Phase" where items are purchased in some sequence as quickly as possible, and then a "Waiting Phase" where no items are purchased.

We begin our version-by-version analysis by examining the case where exactly 1 item is available for purchase, and we present formulas describing how many copies of the item should be purchased in both the fixed-cost case and the increasing-cost case.

Next, we analyze cases involving 2 items. In the 2 item fixed-cost case, we prove that the optimal solution always involves buying some number of copies of one item, followed by some number of copies of the other item.

Then, we analyze the case involving k items. In the k items fixed-cost case, a weakly polynomial time dynamic programming solution can be used to find the optimal sequence of items to buy, and in the increasing-cost case, a strongly polynomial time dynamic programming solution can be used. Additionally, a greedy algorithm can be devised with an approximation ratio that approaches 1 for sufficiently large values of M.

Afterwards, we present negative results, including proofs of Weak NP-hardness of the decision version of the problem of reaching a generation rate of R as quickly as possible, as well as for a version of Cookie Clicker that allows you to start with a nonzero number of cookies. Finally, we define a discretized version of Cookie Clicker where decisions regarding whether or not to buy an item happen in discrete time steps and prove Strong NP-hardness for that version.

Table 1: Summary of Results

Problem Variant	**Result for M version**	**Result for R version**
1 Item Fixed-Cost with item $(x, y, 1)$	Final answer is $\approx \frac{y}{x} \ln \frac{M}{y}$ $O(1)$ to solve	Final answer is $\approx \frac{y}{x} \ln \frac{R}{x}$ $O(1)$ to solve
1 Item Increasing-Cost with item (x, y, α)	Stop "Buying Phase" after $\log_\alpha \frac{M}{y}$ items $O(1)$ to solve	Stop "Buying Phase" after $\frac{R}{x}$ items $O(1)$ to solve
2 Items Fixed-Cost with items $(x_i, y_i, 1)$ where $y_2 > y_1$	Solutions of the form $[1, 1, \ldots, 1, 2, \ldots, 2]$ for large enough M $u_1 \log_\phi u_2 + O(u_1)$ to solve, where $u_i \approx \frac{y_i}{x_i} \log \frac{M}{y_i}$	Solutions of the form $[1, 1, \ldots, 1, 2, \ldots, 2, 1, 1]$ for a small number of 1's at the end for large enough R.
k Items Fixed-Cost with items $(x_i, y_i, 1)$	Dynamic Programming solution, runtime $O(\max_i(\frac{Mx_ik}{y_i}))$	Dynamic Programming solution, runtime $O(kR)$
k Items Increasing-Cost with items (x_i, y_i, α_i)	$O(\max_i(k \log^k_{\alpha_i} \frac{M}{y_i}))$ using Dynamic Programming Greedy Algorithm has Approximation Ratio of $1 + O(\frac{1}{\log M})$ for $k = 2$	$O(\max_i(k(\frac{R}{x_i})^k))$ using Dynamic Programming Weakly NP-hard by reduction from PARTITION
k Items Increasing-Cost with items (x_i, y_i, α_i) with Initial Cookies	Weakly NP-hard by reduction from PARTITION	Weakly NP-hard by reduction from M version
Discrete k Items Increasing-Cost with items (x_i, y_i, α_i) with Initial Cookies	Strongly NP-hard by reduction from 3-PARTITION	Strongly NP-hard by reduction from M version

Periodic structures of five-way cylinder packing

Yoshinori Teshima*[1], Ryo Higashida[1], & Takeo Matsumoto[2]
(1) Chiba Institute of Technology, 2-17-1 Tsudanuma, Narashino-shi, Chiba 275-0016, JAPAN
* yoshinori.teshima@it-chiba.ac.jp
(2) Emeritus Prof., Kanazawa University

Keywords: cylinder packing, packing density, space group, symmetry

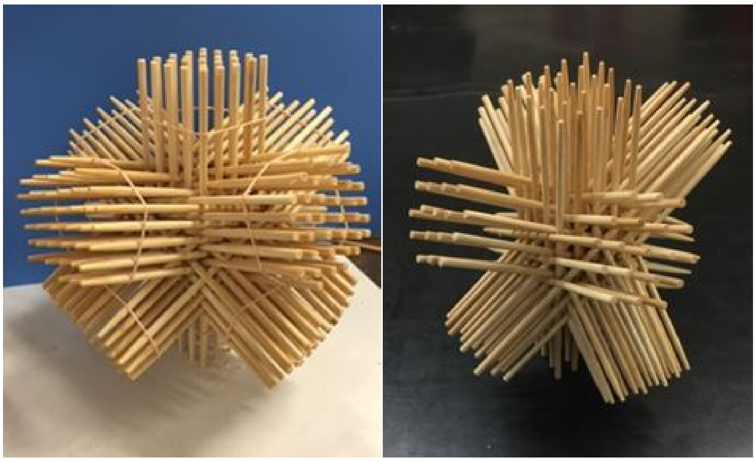
Packing problems are an important aspect of crystallography. In particular, sphere packings have played an important role in improving our understanding of crystal structures. Although cylinder packings have not received as much research attention as sphere packings, they are also important for the same reason and have been investigated in the fields of both science and engineering.

In the field of science, the complex structure of garnet has been explained on the basis of cylinder packing to be a periodic structure with a cubic four-way cylinder packing [1]. Since then, cylinder packings have been extensively applied in the field of crystal chemistry. In particular, homogeneous cubic cylinder packings have been thoroughly investigated.

In the field of engineering as well, cylinder packings are important for determining the fiber packings of composite materials [2]. Apart from cylinders, bars with various cross-sectional shapes are used in composite materials to enhance the packing density. Some regular fiber packing structures have been designed.

Motivated by structures of composite materials [2], periodic six-way cylinder packing structures have also been investigated [3] [4].

In this study, authors will focus on two structures of cylinder packing with five directions. One of them was described in the pioneering paper [2]. Other one is new and should be distinguished from the former one. The two structures are derived from two distinct structures of cylinder packing with four directions.



References

[1] O' Keeffe, M. & Andersson, S. (1977). Acta Cryst., A33, 914–923.

[2] Hijikata, A. & Fukuta, K. (1992). Nihon Fukugoh Zairyou Gakkaishi (J. Japan Soc. Composite Materials), 18, 231–238 [in Japanese].

[3] Teshima, Y., Watanabe, Y., and Ogawa, T. (2001). Lect. Notes. Comput. Sci., 2098, 351-361. (Proc. of JCDCG 2000)

[4] Teshima, Y. & Matsumoto, T. (2012). Glass Phys. Chem., 38-1, 41-48.

Laguerre Voronoi Diagram as a Model for Generating the Tessellation Patterns on the Sphere

Supanut Chaidee[1] and Kokichi Sugihara[2]

[1]Department of Mathematics, Faculty of Science, Chiang Mai University
239, Huay Kaew Rd., Muang, Chiang Mai, 50200, Thailand
[2]Graduate School of Advanced Mathematical Sciences, Meiji University
4-21-1 Nakano, Nakano-ku, Tokyo 164-8525, Japan
[1]supanut.c@cmu.ac.th, [2]kokichis@meiji.ac.jp

Abstract

We propose a model for generating tessellation patterns on the sphere using the spherical Laguerre Voronoi diagram which satisfies the real world assumptions. The generator pushing model is presented to generate the tessellation dynamically. The simulations were done for the different distribution of spherical circles on the sphere, and the results show the tendency of the distribution of resulting spherical circles.

1 Introduction

There are many phenomena displayed as polygonal tessellations. Voronoi diagram is one of the possible candidates for modeling those phenomena. In the application viewpoints, the Laguerre Voronoi diagram (or power diagram) is widely used for analyzing and modeling the tessellation phenomena in the real world as mentioned in [4, 5] since it involves weights, but still Voronoi edges are straight lines.

Recently, we were interested in analysis the tessellation patterns of fruit skins which were regarded as spherical tessellations. We used the *spherical Laguerre Voronoi diagram* (SLVD) which was defined in [6] by defining the spherical circle on the unit sphere U, i.e.

$$\tilde{c}_i = \{p \in U | \tilde{d}(p_i, p) = r_i\} \qquad (1.1)$$

where p_i and r_i are the center and radius of the circle $\tilde{c}_i$ such that $0 \leq r_i < \pi/2$, respectively. The Laguerre proximity is defined by

$$\tilde{d}_L(p, \tilde{c}_i) = \cos \tilde{d}(p, p_i)/\cos r_i. \qquad (1.2)$$

We solved the SLVD approximation problem which finds the best fit SLVD of the given spherical tessellation in the case of the spike-containing object [1] and object without spikes [2] where tessellations were chosen from fruit skins. The results of the previous studies showed that it is promising to use the SLVD as a tool for modeling the pattern formation of tessellation patterns on the fruit skins. Based on the biological viewpoint of the fruits we used in experiments, we also found the common characteristics of fruits which lead to the modeling rules of the patterns formations.

In this study, we propose a framework for modeling the tessellation patterns on the sphere using the SLVD. The basic assumptions are modified from the Voronoi growth model proposed by Okabe et al. in [3]. The energy model is presented to control the dynamics of the tessellations. We finally simulate the models and observe the resulting tessellation in the distribution of spherical circles.

2 Modeling Assumptions

From the biological information, the patterns of fruit skins are generated from the small particles whose shapes are the tubed-like structures which are radially attached to the large object. Based on the fact, we assume that the attached structure is the unit sphere U whose center is located at $(0, 0, 0)$, and each particle is recognized as a spherical circle $\tilde{c}_i$ on the sphere U. The set of spherical circles on the sphere U is defined by $\mathcal{G} = \{\tilde{c}_1, ..., \tilde{c}_N\}$. Remark that the SLVD can be constructed from the set $\mathcal{G}$ using the distance (1.2).

Let t be the time variable in the processes. We consider the discrete version of the model. For each spherical circle $\tilde{c}_i$, we assume that the radius of spherical circle $\tilde{c}_i$ is specified by a function $R_i(t)$ at the time t, i.e. the equation (1.1) is rewritten as

$$\tilde{c}_i(t) = \{p \in U : \tilde{d}(p_i(t), p(t)) = R_i(t)\} \qquad (2.1)$$

such that $0 \leq R_i(t) < \pi/2$, and $p_i(t)$ is the spherical circle center at time t. We also assume that each spherical circle center p_i is radially fixed for all i.

Note that the determination of function $R_i(t)$ for all i is based on the observation of the growth in the real world. By the growth characteristics, we assume that $R_i(t)$ are nondecreasing bounded functions for all i.

The following assumptions are defined for modeling the patterns using SLVD which are modified from [3].

- (V1) The generating circles occur at the same time.
- (V2) The position of the generating circles are allowed to moved during the growth.
- (V3) The functions of generating circle radii are not necessarily the same.
- (V4) The growth happens in all direction of the spherical center.

3 Main Framework

We propose the model which allows the dynamical change of the spherical circles called *the generator pushing model.* Intuitively, we suppose that each spherical circle grows with its radius function $R_i(t)$, and each circle is considered as an elastic circle. When any two distinct circles $\tilde{c}_i$ and $\tilde{c}_j$ touch each other, the energy from pushing of two circles occurs. Therefore, we would like to minimize the energies of all touched circles in each step of t. Based on the biological information, the circle center at time s should not move too much from the position of time $s-1$. This means that the sum of the squared distance between the center of spherical circle $\tilde{c}_i$ at time s and $s-1$ for all i is minimized.

In detail, let $p_i(t) = (1, \theta_i^t, \phi_i^t)$ be the center of the spherical circle $\tilde{c}_i(t)$ in the form of spherical coordinates such that θ_i^t, ϕ_i^t are angles measured from the north-pole, and positive side of x-axis to the point $p_i(t)$, respectively. Let $(\mathcal{G}(t))_{t=0}^{K}$ be a sequence of set of spherical circles $\mathcal{G}(t) = \{\tilde{c}_0(t), ..., \tilde{c}_N(t)\}$ at time t starting from $t = 1$ to $t = K$, and assume that $\mathcal{G}(0)$ is the set of spherical circles at the initial time $t = 0$.

For any two spherical circles $\tilde{c}_i(t), \tilde{c}_j(t)$, we denote $\tilde{c}_i(t) \sim \tilde{c}_j(t)$ if and only if $R_i(t-1) + R_j(t-1) \geq \tilde{d}(p_i(t-1), p_j(t-1))$. Otherwise, $\tilde{c}_i(t) \nsim \tilde{c}_j(t)$. Therefore, the energy of the pair of circles is defined by

$$\Delta E_{i,j} = \begin{cases} 0 & \text{if } \tilde{c}_i(t) \nsim \tilde{c}_j(t); \\ R_i(t) + R_j(t) - \tilde{d}(p_i(t), p_j(t)) & \\ & \text{if } \tilde{c}_i(t) \sim \tilde{c}_j(t). \end{cases} \tag{3.1}$$

Therefore, the energy function E is defined with respect to the variable set $\overrightarrow{\varphi} = (\theta_1^t, ..., \theta_N^t, \phi_1^t, ..., \phi_N^t) \in \mathbb{R}^{2N}$ as

$$E(\theta_1^t, \phi_1^t, ..., \theta_N^t, \phi_N^t) = \sum_{i,j} (\Delta E_{i,j})^2 \tag{3.2}$$

such that $\Delta E_{i,j}$ is defined by (3.1).

Remark that we can compute the corresponding pairs of spherical circles in a set $\mathcal{G}(t)$ by constructing the spherical Laguerre Delaunay diagram of the set of spherical circles $\mathcal{G}(t-1)$ using algorithm in [6] to obtain the topological structure of any two circles, i.e. if any pair of vertices of the spherical Laguerre Delaunay diagram at time $t-1$ has a connected edge, then those circles are adjacent.

In addition to the energy, the function of distance between the generators at time $t-1$ and t, which is a function of variables $\theta_1^t, \phi_1^t, ..., \theta_N^t, \phi_N^t$, can be denoted by

$$F(\theta_1^t, \phi_1^t, ..., \theta_N^t, \phi_N^t) = \sum_{i=1}^{N} \tilde{d}(p_i(t-1), p_i(t))^2. \tag{3.3}$$

Hence, we use multiobjective optimization for solving the minimization of function $E(\theta_1^t, \phi_1^t, ..., \theta_N^t, \phi_N^t)$ and $F(\theta_1^t, \phi_1^t, ..., \theta_N^t, \phi_N^t)$. In this study, we use weighting method, i.e. for $\omega \geq 0$, we define

$$\mathcal{F}(\theta_1^t, \phi_1^t, ..., \theta_N^t, \phi_N^t) = \omega E(\theta_1^t, \phi_1^t, ..., \theta_N^t, \phi_N^t) + (1-\omega) F(\theta_1^t, \phi_1^t, ..., \theta_N^t, \phi_N^t) \tag{3.4}$$

then solve

$$\begin{aligned} \text{minimize} \quad & \mathcal{F}(\theta_1^t, \phi_1^t, ..., \theta_N^t, \phi_N^t) \\ \text{subject to} \quad & (\theta_1^t, \phi_1^t, ..., \theta_N^t, \phi_N^t) \in S \subset \mathbb{R}^{2N}. \end{aligned}$$

Therefore, the framework can be concluded into the following steps: for $t = 1$ to N, compute the SLVD at time $t-1$, computing the energies of all corresponding spherical circles and sum of the squared distance between point at time $t-1$ and t, minimize $\{E, F\}$, and update the spherical circle centers.

4 Experiments and Conclusion

We performed the experiments by generating N spherical circles $\mathcal{G}(t) = \{\tilde{c}_1(t), ..., \tilde{c}_N(t)\}$ on the sphere. In this study, the spherical radii are determined by the logistic function

$$R_i(t) = \frac{L_i}{1 + e^{-k(t-t_0)}} \tag{4.1}$$

for chosen k, t_0 such that L_i is a pseudorandom number for all i.

To terminate the procedure of the simulation, the threshold ϵ is defined for determining the number K of iteration. In the case of logistic function defined in (4.1), $K = \max_{i=1,...,N}\{\lceil t_0 - \frac{1}{k}\ln\left(\frac{L_i}{\epsilon + L_i}\right)\rceil\}$.

To observe the tessellation pattern after simulation, we mainly focus on the distribution of generating circles on the sphere after simulation. Therefore, we set the experiments to two groups. The first group is the case that initial spherical circles are located uniformly, i.e. the centers of generating circles are equidistributed on the sphere. Another group is set in a way that generating circles are randomly located on the sphere.

The simulation results showed the tendency that the spherical circles were uniformly distributed after employing the generator pushing model to the set of spherical circles. Figure 1 shows the example of the simulation results when the initial spherical circles are randomly distributed on the sphere implemented by Wolfram Mathematica ®11.1. We plan to interpret the resulting diagram to describe the pattern formation of fruit skin patterns.

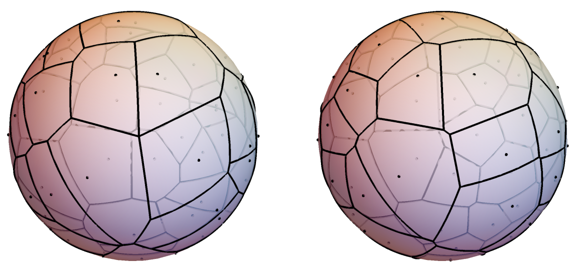


Figure 1: The results from the simulation when the spherical circle were randomly distributed with $N = 50, k = 0.2$, $t_0 = 15$, $\epsilon = 10^{-8}$, $L_i \in [\arccos(1-\frac{1}{N}) - \frac{\pi}{36}, \arccos(1-\frac{1}{N}) + \frac{\pi}{36}]$ (left) Initial tessellation; (right) final tessellation at $t = 112$

References

[1] S. Chaidee and K. Sugihara (2016), Fitting Spherical Laguerre Voronoi Diagrams to Real World Tessellations Using Planar Photographic Images, J. Akiyama et al. (eds.): Discrete and Computational Geometry and Graphs (LNCS 9943), Springer, 2016, pp. 73-84.

[2] S. Chaidee and K. Sugihara, Spherical Laguerre Voronoi Diagram Approximation to Tessellations without Generators, Submitted to *Graphical Models*

[3] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu (2009). Spatial tessellations: concepts and applications of Voronoi diagrams (Vol. 501). John Wiley & Sons.

[4] H. Telley, T. M. Liebling, and A. Mocellin (1996). The Laguerre model of grain growth in two dimensions I. Cellular structures viewed as dynamical Laguerre tessellations. Philosophical Magazine B, 73(3), 395-408.

[5] H. Telley, T. M. Libeling, and A. Mocellin, (1996). The Laguerre model of grain growth in two dimensions II. Examples of coarsening simulations. Philosophical Magazine B, 73(3), 409-427.

[6] K. Sugihara (2002), *Laguerre Voronoi Diagram on the Sphere*, J. Geom. Graph., 6 (1), 69-81.

Pentagonal Subdivision and Double Pentagonal Subdivision

Min Yan
Hong Kong University of Science and Technology

Pentagonal subdivision and double pentagonal subdivision are two constructions that convert any tiling of an oriented surface into tilings by pentagons. When the constructions are applied to Platonic solids, we get tilings of the sphere by congruent pentagons. Specifically, the pentagonal subdivision produces three 2-variable families of tilings of the sphere by 12, 24 and 60 congruent pentagons of edge length combination a^2b^2c, and the double pentagonal subdivision produces three rigid tilings of the sphere by 24, 48 and 120 congruent pentagons of edge length combination a^3bc.

We discuss some combinatorial problems related to the two constructions. We also prove that, under the assumption that there is enough variation in edge lengths, the two constructions are the only tilings of sphere by congruent pentagons.

This is a joint work with Erxiao Wang of the Hong Kong University of Science and Technology.

The Independence and Domination Numbers of the Hanoi Graphs

Agnes D. Garciano, Reginaldo M. Marcelo[1],
Mari-Jo P. Ruiz & Mark Anthony C. Tolentino

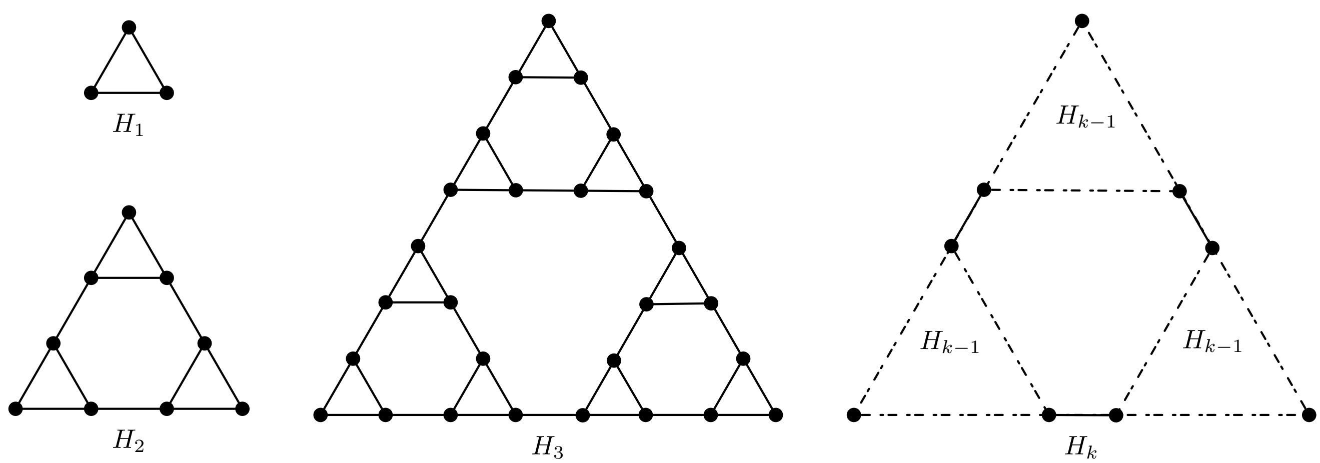
Mathematics Department
Ateneo de Manila University, Quezon City, Philippines

A vertex subset S of a graph $G = (V, E)$ is an *independent set* if no two vertices in S are adjacent, and it is a *dominating set* if every vertex that is not in S is adjacent to a vertex in S. The *independence number* of G, denoted by $\alpha(G)$, is the maximum cardinality of an independent set; the *domination number* of G, denoted by $\gamma(G)$, is the minimum cardinality of a dominating set; and the *independent domination number* of G, denoted by $i(G)$, is the minimum cardinality of an independent dominating set.

Given any graph G, it follows from the definitions that $\gamma(G) \leq i(G) \leq \alpha(G)$. The first inequality is discussed in [1], and a sufficient condition for equality is given. A survey on recent results on independent domination in graphs is given in [3].

In this paper, we study independent dominating sets in the Hanoi graphs. The Hanoi graphs are derived from the states of the Tower of Hanoi problem [2, 4]. These graphs also belong to the family of Sierpinski-like graphs [5]. They are derived in an iterative manner in much the same way as the process used in the derivation of the Sierpinski triangle fractal [4].

Let H_k denote the kth Hanoi graph. Then H_k is constructed as shown below.



In this paper we prove the following:

$$\gamma(H_k) = i(H_k) = \left\lceil \frac{3^k}{4} \right\rceil \text{ and } \alpha(H_k) = 3^{k-1}.$$

[1]Presenter

References

[1] R.B. Allan and R. Laskar, On domination and independent domination numbers of a graph, *Discrete Mathematics* **23(2)**(1978), 73-76.

[2] D. Arett and S. Dorée, Coloring and Counting on the tower of Hanoi graphs, *Mathematics Magazine* **83(3)**(2010), 200-209.

[3] W. Goddard and M.A. Henning, Independent domination in graphs: a survey and recent results, *Discrete Mathematics* **313(7)**(2013), 839-854.

[4] A.M. Hinz, *The Tower of Hanoi in Algebra and Combinatorics (ICAC '97, Hongkong)*, K.P. Shum, E.J. Taft, Z.X. Wan (eds.), Springer, Singapore 1999, 277-289.

[5] S. Klavzar and S. Zemljic, A survey and classification of Sierpinski-like graphs, *Discrete Applied Mathematics* **217**(2017), 560-600.

Sigma Chromatic Number of the Sierpinski Gasket Graphs and the Hanoi Graphs

Agnes D. Garciano, Reginaldo M. Marcelo,
Mari-Jo P. Ruiz, Mark Anthony C. Tolentino[1]
agarciano@ateneo.edu,rmarcelo@ateneo.edu
mruiz@ateneo.edu,mtolentino@ateneo.edu
Ateneo de Manila University

In [2], G. Chartrand, F. Okamoto, and P. Zhang defined the concept of the *sigma chromatic number* of a graph as follows: For a non-trivial connected graph G, let $c : V(G) \to \mathbb{N}$ be a vertex coloring of G. For each $v \in V(G)$, let $N(v)$ denote the *neighborhood of* v, i.e., the set of vertices adjacent to v. Moreover, the *color sum* of v, denoted by $\sigma(v)$, is defined to be the sum of the colors of the vertices in $N(v)$. If $\sigma(u) \neq \sigma(v)$ for every two adjacent $u, v \in V(G)$, then c is called a *sigma coloring* of G. The minimum number of colors required in a sigma coloring of G is called its sigma chromatic number and is denoted by $\sigma(G)$.

In [6], Klavžar defines geometrically the Sierpiński gasket graphs S_n, $n \geq 1$ as the graphs whose vertices are the intersection points of the finite Sierpiński gasket σ_n and whose edges are the line segments of the gasket. The following figure shows how the Sierpiński gasket graphs are constructed.

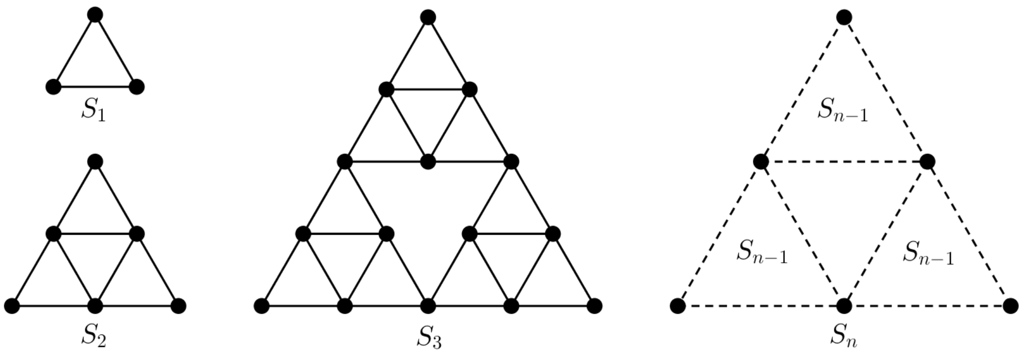


Figure 1: Sierpiński gasket graphs

The Hanoi graphs are derived from the states of the Tower of Hanoi problem [1, 4, 6]. Like the Sierpiński gasket graphs, the Hanoi graphs are also constructed in an interative manner. These graphs also belong to the general class of Sierpiński-like graphs [5]. The following figure shows how the Hanoi graphs are constructed.

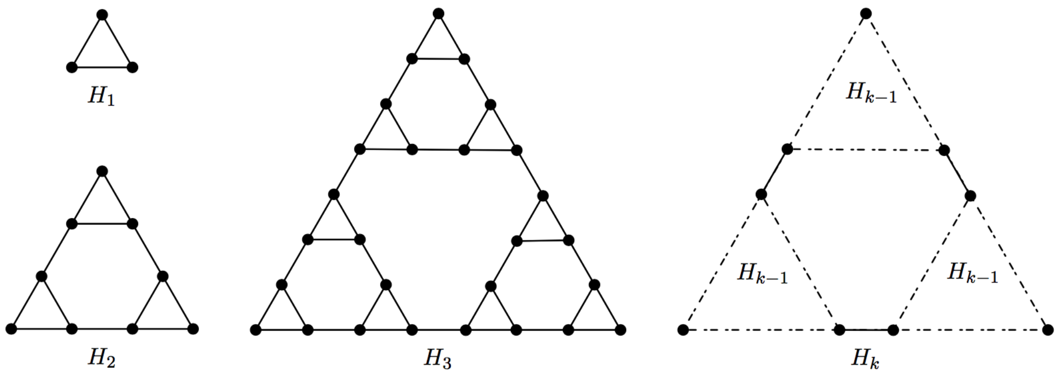


Figure 2: Hanoi graphs

[1] speaker

In this work, we study the sigma coloring of the Sierpiński gasket graphs and the Hanoi graphs. We prove the following results:

1. The sigma chromatic number of the Sierpiński gasket graph S_n is 2 for any $n \geq 2$.

2. The sigma chromatic number of the Hanoi graph H_n is 3 for any $n \geq 3$.

References

[1] Danielle Arett and Suzanne Dorée. Coloring and counting on the tower of hanoi graphs. *Mathematics Magazine*, 83(3):200–209, 2010.

[2] Gary Chartrand, Futaba Okamoto, and Ping Zhang. The sigma chromatic number of a graph. *Graphs and Combinatorics*, 26:755–773, 2010.

[3] Ali Dehghan, Mohammad-Reza Sadeghi, and Arash Ahadi. The complexity of the sigma chromatic number of cubic graphs. arXiv:1403.6288v1 [math.CO], March 2014.

[4] Andreas Hinz. The tower of hanoi in algebra and combinatorics. In K.P. Shum, E.J. Taft, and Z.X. Wan, editors, *ICAC '97, Hongkong*, Lecture Notes in Computer Science, pages 277–289. Springer, Singapore, 1999.

[5] Andreas Hinz, Sandi Klavžar, and Sara Sabrina Zemljič. A survey and classification of sierpinski-like graphs. *Discrete Mathematics*, 217:560–600, 2017.

[6] Sandi Klavžar. Coloring sierpiński graphs and sierpiński gasket graphs. *Taiwanese Journal of Mathematics*, 12(2):513–522, 2008.

[7] Paul Adrian Luzon, Mari-Jo Ruiz, and Mark Anthony Tolentino. The sigma chromatic number of the circulant graphs $C_n(1,2)$, $C_n(1,3)$, and $C_{2n}(1,n)$. In Akiyama J., Ito H., Sakai T., and Uno Y., editors, *Discrete and Computational Geometry and Graphs. JCDCGG 2015.*, volume 9943 of *Lecture Notes in Computer Science*. Springer, Cham, 2016.

Results on Additive Chromatic Numbers and Additive Choice Numbers of Halin Graphs

Hsin-Hao Lai

Department of Mathematics, National Kaohsiung Normal University, Kaohsiung, Taiwan

hsinhaolai@nknucc.nknu.edu.tw

The notation of additive coloring and additive chromatic number was first introduced in [2]. An *additive coloring* of a graph G is a labeling from the vertex set of G to the set of integers such that for every two adjacent vertices the sums of integers assigned to their neighbors are different. The *additive chromatic number*, denoted $\eta(G)$, is the least integer k such that G has an additive coloring from the vertex set of G to $\{1, 2, \ldots, k\}$. The following conjecture was proposed in [2], where $\chi(G)$ denotes the chromatic number of G.

Conjecture 1 *For any graph G, $\eta(G) \leqslant \chi(G)$.*

The notation of additive choice number was first introduced in [1]. A *list* L of a graph G is a mapping that assigns a finite set of integers to each vertex of G. A list is a *k-list* if $|L(v)| \geqslant k$ for each vertex v. An additive coloring f of G such that $f(v) \in L(v)$ for each vertex v is called an *additive L-coloring* of G. A graph G is said to be *additive k-choosable* if it has an additive L-coloring for any k-list L. The *additive choice number*, denoted $\eta_l(G)$, is the least integer k such that G is additive k-choosable. Obviously, $\eta(G) \leqslant \eta_l(G)$ for any graph G.

A *Halin graph* is a plane graph $G = T \cup C$ constructed as follows. Let T be a tree of order at least 4. All vertices of T are either of degree 1 or of degree at least 3. Let C be a cycle connecting the leaves of T in such a way that C forms the boundary of the unbounded face.

In this paper, I will apply mathematical induction and discharging method to obtain the following result on planar graphs.

Theorem 2 *If G is a planar graph, then $\eta(G) \leqslant \eta_l(G) \leqslant 2\Delta(G) + 25$.*

And I will apply mathematical induction and Combinatorial Nullstellensatz to obtain the following result on Halin graphs.

Theorem 3 *If G is a Halin graph, then $\eta(G) \leqslant \eta_l(G) \leqslant 5$.*

Keywords: additive coloring, lucky labeling, additive chromatic number, additive choice number, Halin graph.

References

[1] S. Akbari, M. Ghanbari, R. Manaviyat and S. Zare, On the lucky choice number of graphs, Graphs Combin. 29 (2013), 157-163.

[2] S. Czerwiński, J. Grytczuk and W. Żelazny, Lucky labelings of graphs, Inform. Process. Lett. 109 (2009), 1078-1081.

New Developments on r-Equitable Coloring of Cross Products of Graphs

Chao-Chun Chen, Zi-Yi Yang, and Chih-Hung Yen*

Department of Applied Mathematics

National Chiayi University, Chiayi 60004, Taiwan

Abstract

A graph G consists of a nonempty vertex set $V(G)$ and an edge set $E(G)$. All graphs considered in this paper are finite, loopless, and without multiple edges. For a positive integer k, a (proper) *k-coloring* of a graph G is a mapping $\phi : V(G) \to \{1, 2, \ldots, k\}$ such that adjacent vertices are mapped to distinct numbers. The images $1, 2, \ldots, k$ are called *colors* and all pre-images of a fixed i, $1 \leq i \leq k$, form a *color class*. Then a k-coloring of a graph G is said to be *equitable* if the sizes of any two color classes differ by at most one. And a graph G is *equitably k-colorable* if G has an equitable k-coloring. The smallest integer k for which G is equitably k-colorable, denoted by $\chi_{=}(G)$, is called the *equitable chromatic number* of G. The notion of an equitable coloring was first introduced in [9] by W. Meyer. So far, quite a few results on the equitable coloring of graphs have been obtained. Please refer to the survey of Lih [7].

In 2011, Hertz and Ries [6] generalized the notion of equitable colorability. They said that a k-coloring of a graph G is *r-equitable* for an integer $r \geq 0$ if the sizes of any two color classes differ by at most r. And a graph G is *r-equitably k-colorable* if there exists an r-equitable k-coloring of G. Clearly, an equitably k-colorable graph is 1-equitably k-colorable, and vice versa. Similarly, the smallest integer k for which G is r-equitably k-colorable, denoted by $\chi_{r=}(G)$, is called the *r-equitable chromatic number* of G. It is clear that an r-equitably k-colorable graph is certainly $(r+1)$-equitably k-colorable. However, unlike proper colorings of graphs, an r-equitably k-colorable graph may not be r-equitably $(k+1)$-colorable. Hence, we also have an interest in finding the smallest integer n such that a graph G is r-equitably k-colorable for all $k \geq n$, called the *r-equitable chromatic threshold* of G and denoted by $\chi^*_{r=}(G)$. That is, $\chi^*_{r=}(G)$ may be greater than $\chi_{r=}(G)$.

The *cross product* $G \times H$, also known as the Kronecker, direct, tensor, weak tensor, or categorical product, of two graphs G and H is the graph with vertex set $\{(x, y) : x \in V(G) \text{ and } y \in V(H)\}$ and edge set $\{(x, y)(x', y') : xx' \in$

*E-mail address: chyen@mail.ncyu.edu.tw

$E(G)$ and $yy' \in E(H)\}$. By the way, cross product is so named because the product of two single edges is a cross. In this paper, we introduce (new) results obtained on r-equitable chromatic numbers and thresholds of cross products of graphs, especially for $r \neq 1$.

Keywords: Equitable coloring; r-Equitable coloring; r-Equitable chromatic number; r-Equitable chromatic threshold.

References

[1] B.-L. Chen, K.-W. Lih, J.-H. Yan, Equitable coloring of graph products, manuscript, 1998.

[2] B.-L. Chen, K.-W. Lih, J.-H. Yan, Equitable coloring of interval graphs and products of graphs, 2009, arXiv:0903.1396v1.

[3] D. Duffus, B. Sands, R. E. Woodrow, On the chromatic number of the product of graphs, J. Graph Theory 9 (1985) 487-495.

[4] H. Furmańczyk, Equitable colorings of graph products, Opuscula Mathematica 26(1) (2006) 31-44.

[5] S. Hedetniemi, Homomorphism of graphs and automata, Univ. Michigan, Technical Report 03105-44-T, 1966.

[6] A. Hertz, B. Ries, On r-equitable colorings of trees and forests, Les Cahiers du GERAD, 2011, G-2011-40.

[7] K.-W. Lih, Equitable coloring of graphs, P.M. Pardalos et al. (eds.), Handbook of Combinatorial Optimization, Springer Science+Business Media New York 2013, 1199-1248.

[8] W.-H. Lin, G. J. Chang, Equitable coloring of Kronecker products of graphs, Discrete Appl. Math. 158 (2010) 1816-1826.

[9] W. Meyer, Equitable coloring, Amer. Math. Monthly, 80 (1973) 920-922.

[10] W. Wang, Z. Yan, X. Zhang, On r-equitable chromatic threshold of Kronecker products of complete graphs, Discrete Appl. Math. 175 (2014) 129-134.

[11] Z. Yan, W. Wang, Equitable chromatic threshold of direct products of complete graphs, Ars Combin. 114 (2014) 235-243.

[12] Z. Yan, W. Wang, Equitable coloring of Kronecker products of complete multipartite graphs and complete graphs, Discrete Appl. Math. 162 (2014) 328-333.

A result on balanced partitions of 3 colored point sets

Toshinori Sakai
Tokai University, Japan
sakai@tokai-u.jp

This is a joint work with Mikio Kano.

We show the following results concerning partitions of three colored point sets:

Theorem 1 *Let $n \geq 2$ be an integer, and let R, B, and G be disjoint sets of red points, blue points, and green points in general position in the plane, respectively. If $|R \cup B \cup G| = 4n$ and $|R|, |B|, |G| \leq 2n$, then there exists a straight line l such that l passes through no point of $R \cup B \cup G$, and such that each half plane H defined by l satisfies (i) $|(R \cup B \cup G) \cap H| = 4m$ and (ii) $|R \cap H|, |B \cap H|, |G \cap H| \leq 2m$ for some positive integer m.*

Applying this result repeatedly, we obtain the following corollary:

Corollary 1 *Let n be an integer, and let R, B, and G be disjoint sets of red points, blue points, and green points in general position in the plane, respectively. If $|R \cup B \cup G| = 4n$ and $|R|, |B|, |G| \leq 2n$, then there exists a subdivision $D_1 \cup D_2 \cup \cdots \cup D_n$ of the plane into n disjoint convex regions such that (i) $|(R \cup B \cup G) \cap D_i| = 4$ and (ii) $|R \cap D_i|, |B \cap D_i|, |G \cap D_i| \leq 2$ for $1 \leq i \leq n$.*

Planarity Preserving Augmentation of Plane Graphs to Meet Parity Constraints*

J.C. Catana-Salazar[1], A. García[2], J. Tejel[2], and J. Urrutia[3]

[1]Posgrado en Ciencia e Ingeniería de la Computación, UNAM.
[2]Departamento de Métodos Estadísticos, IUMA. Universidad de Zaragoza.
[3]Instituto de Matemáticas, UNAM.

1 Introduction

Let $G = (V, E)$ be a plane graph and let $C_G = \{c_1, c_2, ..., c_n\}$ be a set of parity constraints, where the vertex $v_i \in V$ has assigned the constraint c_i. The *augmentation problem to meet parity constraints* refers to find a graph $H = (V, E')$ such that (i) $G' = G \cup H$ is a simple plane graph (H is compatible with G), (ii) H and G are disjoint ($E' \cap E = \emptyset$) and (iii) the degree of the vertices in G' meet the parity constraint set C_G. The well-known problem of transforming a graph into Eulerian is an example of this type of problem.

Hereafter we denote by $S(G, C_G)$ the set of vertices in G do not satisfying its parity constraint in C_G. We call (and represent) a vertex $v_i \in S(G, C_G)$ a red vertex of G, otherwise we call it a blue vertex of G. Note that every red vertex of G must have odd degree in H and every blue vertex of G must have even degree in H. We say that an edge is red, if it's both endpoints are red vertices (analogously for blue vertices). We say that an edge is red-blue, if its endpoints have different colors.

An outerplanar graph is a graph having a plane embedding in such a way that all the vertices belong to the outer face of the drawing. An outerplanar graph is maximal (MOP for short) if it is not possible to add an edge such that the resulting graph is still outerplanar.

It is know in [1] that every topological tree $T = (V, E)$ with an arbitrary constrains set C_T is augmentable to meet C_T (except for the star with its center being red) with at most $\frac{k}{2} + 1$ edges, i.e., T admits a compatible and disjoint topological graph $H = (V, E')$ with $|E'| \leq \frac{k}{2} + 1$, $k = |S(T, C_T)|$. Nevertheless, it is *NP-Complete* to decide if a topological plane graph G admits a topological matching H, such that $G' = G \cup H$ meets a parity constraints set C_G, where $S(G, C_G) \subset V$.

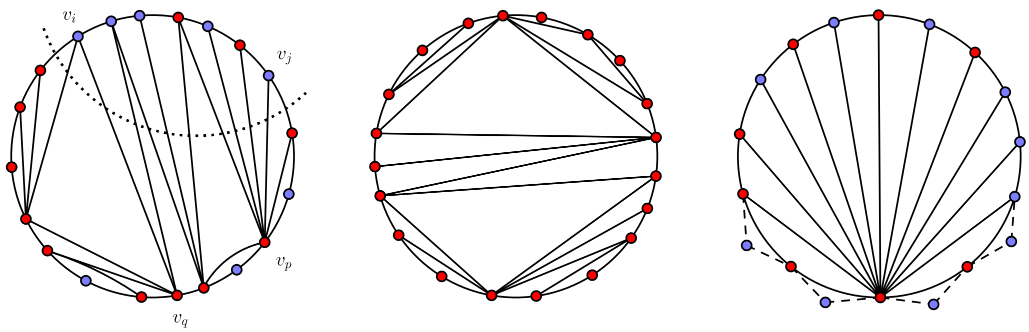
We study two variants of the augmentation problem to meet parity constraints in outerplanar graphs. In Section 2, we tackle the augmentation problem when the embedding of G is fixed, i.e., G preserves its embedding as a subgraph of G'. The second variant of the problem, Section 3, arises when it is allowed to change the embedding of G. In the last section we study this augmentation problem for plane geometric graphs and, in particular, we show that it is *NP-Complete* to decide if a plane geometric tree or path admits a compatible and disjoint perfect matching.

2 Augmentation of graphs fixing their embedding

First we characterize the family of MOP graphs that are non augmentable to meet parity constraints, when its required to preserve the embedding of the input graph.

Theorem 2.1. *Let $G = (V, E)$ be a plane MOP graph and B the set of blue vertices adjacent to a diagonal such that its opposite endpoint is a red vertex. Then, G is non-augmentable preserving its embedding if (i) it is possible to draw a line that separates the ends of all red-blue diagonals such that on one side are all the blue vertices (possibly with red vertices between them, all with degree greater than or equals to 3) and on the other side all red vertices (see Figure 1 (a)), or (ii) G only has red diagonals (see Figure 1 (b)).*

*This research has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 734922. Also supported by project MTM2015-63791-R.



(a) (b) (c)

Figure 1: (a) A non-augmentable plane MOP graph that has blue-red diagonals. (b) A non-augmentable plane MOP graph with only red diagonals. (c) A MOP graph non-augmentable even when it its allowed to change its embedding.

Next we give two polynomial time algorithms: The first to compute a minimum edge set to augment a MOP graph, and the second to augment an outerplanar graph.

Theorem 2.2. *Let $G = (V, E)$ be a plane MOP graph and C_G a set of parity constraints. Then, finding a compatible and disjoint graph $H = (V, E')$ with edge set E' of minimum size, such that $G' = G \cup H$ meets C_G, can be computed in $\mathcal{O}(n^3)$ time.*

Theorem 2.3. *Let $G = (V, E)$ be a plane outerplanar graph and C_G a set of parity constraints. Then, deciding if there exists a compatible and disjoint graph $H = (V, E')$ such that $G' = G \cup H$ meets C_G, can be done in $\mathcal{O}(n)$ time.*

3 Augmentation of graphs with mobile embedding

When it is allowed to choose the embedding of the input MOP graph, we have the following theorem.

Theorem 3.1. *Let $G = (V, E)$ be a MOP graph and C_G a set of parity constraints. Then, deciding if G can be drawn in such a way that there exists a compatible and disjoint topological graph H, such that $G' = G \cup H$ meets C_G can be done in $\mathcal{O}(n^2)$ time.*

Figure 1 (c) shows an example of a MOP graph that cannot be augmented even changing the embedding.

4 Geometric plane graphs

Finally we give some results about the hardness of the augmentation problem in geometric plane graphs.

Theorem 4.1. *Let $G = (V, E)$ be a geometric plane graph and C_G a parity constraints set. Then, the problem of deciding if there exists a topological plane graph H disjoint and compatible with G such that $G' = G \cup H$ meets C_G is $\mathcal{NP}$-Complete. The problem remains $\mathcal{NP}$-Complete even when $S(G, C_G) = V$.*

Theorem 4.2. *Let $T = (V, E)$ be a geometric plane tree. Then, the problem of deciding if T admits a compatible and disjoint perfect matching is $\mathcal{NP}$-Complete.*

Theorem 4.3. *Let $P = (V, E)$ be a geometric plane path. Then, the problem of deciding if P admits a compatible and disjoint perfect matching is $\mathcal{NP}$-Complete.*

References

[1] Israel Aldana Galván, José Luis Álvarez Rebollar, Juan Carlos Catana-Salazar, Erick Solís Villarreal, Jorge Urrutia and Carlos Bruno Velarde Valázquez. *Planarity Preserving Augmentation of Topological and Geometric Plane Graphs to Meet Parity Constraints*, 29th Canadian Conference on Computational Geometry. 2017.

Acute triangulations of the sphere

SANG-HYUN KIM* AND G. S. WALSH

1. Statement of the result

A *combinatorial triangulation* of the unit two-sphere $\mathbb{S}^2$ means a simplicial complex homeomorphic to $\mathbb{S}^2$. An *acute triangulation* of $\mathbb{S}^2$ is a triangulation of $\mathbb{S}^2$ into geodesic triangles whose dihedral angles are all acute. We say a combinatorial triangulation L of $\mathbb{S}^2$ is *realized by an acute triangulation of* $\mathbb{S}^2$ (or in short, *acute*) if there is an acute triangulation T of $\mathbb{S}^2$ and a simplicial homeomorphism from T to L.

In this talk, we completely characterize combinatorial triangulations of $\mathbb{S}^2$ that can be realized by acute triangulations. A simplicial complex Y is *flag* if every complete subgraph in $Y^{(1)}$ spans a simplex. We say Y is *no-square* if every 4-cycle in $Y^{(1)}$ has a chord in $Y^{(1)}$.

Theorem 1. *A combinatorial triangulation L of S^2 is acute if and only if L is flag no-square.*

2. Brief History and Applications

J. Itoh pioneered the problem of finding acute triangulations on the sphere. Itoh and Zamfirescu proved that a geodesic triangle contained in one hemisphere of $\mathbb{S}^2$ can be triangulated into at most ten acute triangles and this bound is sharp [2]. For triangulations of $\mathbb{S}^2$, Itoh proved the following theorem by explicit constructions.

Theorem 2 (J. Itoh [1])**.** (1) *If there exists an acute triangulation of* $\mathbb{S}^2$ *with n faces, then n is even, $n \geq 20$ and $n \neq 22$.*
(2) *If n is even, $n \geq 20$ and $n \neq 22, 28, 34$, then there exists an acute triangulation of* $\mathbb{S}^2$ *with n faces.*

Itoh then asked whether or not there exists an acute triangulation of $\mathbb{S}^2$ with either 28 or 34 faces. We exhibit examples of flag no-square triangulations with 28 and 34 faces; see Figure 1. Using Theorem 1, we have a full answer to Itoh's question:

Corollary 3. *There exists an acute triangulation of* $\mathbb{S}^2$ *with n faces if and only if n is even, $n \geq 20$ and $n \neq 22$.*

As for the theme of fixed combinatorics, Maehara determined exactly when a given abstract triangulation of a polygon can be realized as an acute triangulation in $\mathbb{E}^2$. Given a triangulation L of a disk, a cycle C in $L^{(1)}$ is said to be *enclosing* if C bounds a disk with least one interior vertex. We say that a cycle C is L is *separating* if each component of $L \setminus C$ contains a vertex of L. We say L is *flag-no-separating-square* if L is flag and has no separating 4-cycle.

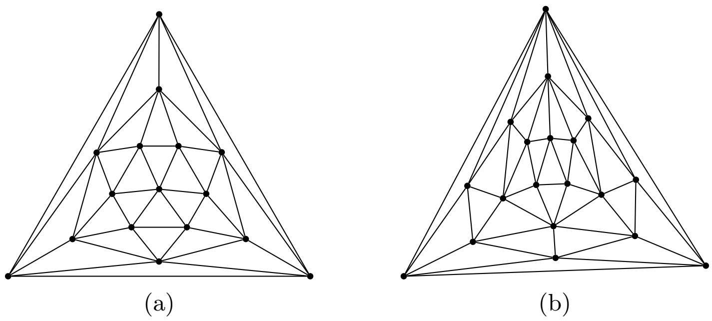


FIGURE 1. Acute spherical triangulations with 28 and 34 faces.

Theorem 4 (H. Maehara [3]). *An abstract triangulation L of a disk is acute in $\mathbb{E}^2$ if and only if L does not have an enclosing 3- or 4-cycle.*

We prove a spherical version of Maehara's theorem.

Theorem 5. *Let L be an abstract triangulation of a compact planar surface such that L is flag no-separating-square. Then:*
(1) *L is acute in $\mathbb{S}^2$.*
(2) *L is acute in $\mathbb{E}^2$ if and only if at least one boundary component of L is not a square.*

The hard part of Maehara's theorem is the existence of acute triangulations. Interestingly, the existence part of our main theorem is relatively simple after using Koebe–Andreev–Thurston theorem on hyperbolic polytopes. The obstruction part is more sophisticated and for this, we use properties of a metric space of singular negative curvature.

REFERENCES

1. J. Itoh, *Acute triangulations of sphere and icosahedron*, Differential geometry (Sakado, 2001), Josai Math. Monogr., vol. 3, Josai Univ., Sakado, 2001, pp. 53–62. MR 1824599 (2002c:52019)
2. Jin-Ichi Itoh and Tudor Zamfirescu, *Acute triangulations of triangles on the sphere*, Rend. Circ. Mat. Palermo (2) Suppl. (2002), no. 70, part II, 59–64, IV International Conference in "Stochastic Geometry, Convex Bodies, Empirical Measures & Applications to Engineering Science", Vol. II (Tropea, 2001). MR 1962585 (2004a:52021)
3. H. Maehara, *Plane graphs with straight edges whose bounded faces are acute triangles*, Journal of Combinatorial Theory. Series B **88** (2003), no. 2, 237–245.

DEPARTMENT OF MATHEMATICAL SCIENCES, SEOUL NATIONAL UNIVERSITY, KOREA
E-mail address: `s.kim@snu.ac.kr`
URL: `http://cayley.kr`

DEPARTMENT OF MATHEMATICS, TUFTS UNIVERSITY, MEDFORD, MA 02155, USA
E-mail address: `genevieve.walsh@tufts.edu`
URL: `http://emerald.tufts.edu/~gwalsh01`

Enumeration of binary matroids using degree sequences

Ken Sugimori*, Sonoko Moriyama, Kunihiko Sadakane

Consider enumerating exhaustively non-isomorphic matroids of a certain size n and a certain rank r. The database of matroids is useful as a test data for verifying mathematical conjectures or a benchmark for numerical experiments. Matsumoto et al. enumerated matroids of $n \leq 10$ and $r \leq 4$ [2]. When enumerating binary matroids, a subclass of matroids, n and r can be larger. Fripertinger and Wild enumerated binary matroids of $n \leq 15$ and $r \leq 7$ [1]. In this paper, we enumerated binary matroids of $n \leq 17$ and $r \leq 8$ by using a more efficient method.

A binary matroid is a matroid that can be representable over the finite field F_2. A binary matroid of size n and rank r can be represented by an $r \times n$ full rank matrix over F_2. Conversely, a matroid represented by an $r \times n$ full rank matrix over F_2 is a binary matroid. The theorem below follows from Theorem 6.4.1 of [3].

Theorem 1. *Let M_1 and M_2 be binary matroids of size n and rank r, and let A_1 and A_2 be $r \times n$ full rank matrices over F_2 that represent M_1 and M_2 respectively. Then, M_1 and M_2 are isomorphic if and only if A_1 and A_2 can be transformed into each other by a sequence of elementary row operations and column-swapping.*

Therefore, enumerating non-isomorphic binary matroids of size n and rank r is equivalent to enumerating $r \times n$ full rank matrices over F_2 that cannot be transformed into each other by a sequence of elementary row operations and column-swapping.

Fripertinger and Wild enumerated binary matroids by the orderly algorithm [4]. In the orderly algorithm, we fix a rank r, and generate in order a list of matroids of size n and rank r from that of size $n-1$ and rank r. The definition of canonical forms is essential for removing duplicate matroids in each list. We need to define canonical forms such that there is exactly one canonical form in each isomorphic class of matroids. We check whether each matroid is a canonical form, and remove non-canonical forms. Fripertinger and Wild defined canonical forms as below. An $r \times n$ full rank matrix over F_2 can be transformed into a standard form $[I_r|D]$ ($D \in F_2^{r\times(n-r)}$) by a sequence of elementary row operations and column-swapping. Thus, we define a standard form $[I_r|D]$ as a canonical form when the matrix D is lexicographically smallest. Note that a matrix $D = (d_{i,j})$ is lexicography smallest if a sequence $(d_{1,1}, \ldots, d_{r,1}, d_{1,2}, \ldots, d_{r,2}, \ldots, d_{1,(n-r)}, \ldots, d_{r,(n-r)})$ is lexicographically smallest.

In the method of Fripertinger and Wild, it is a bottleneck to check whether each matrix is a canonical form. In this paper, we improve efficiency of this process and enumerate binary matroids of larger size n and rank r. First, we explain an efficient method of checking a canonical form regarding the definition of canonical forms by Fripertinger and Wild. Then, we propose a more efficient method by improving the definition of canonical forms.

Consider checking whether an $r \times n$ full rank matrix over F_2 is a canonical form. The most naive method is trying all the possible sequences of elementary row operations and column-swapping on A to obtain a standard form $[I_r|D']$, and checking whether $D \preceq D'$ always holds. In order to improve efficiency of this process, we divide a sequence of elementary row operations and column-swapping on A into the following three steps. First, determine the arrangement of the leftmost r columns, i.e., choose r columns from the n columns to be placed in the leftmost square submatrix. The leftmost r columns must be transformed into I_r by a sequence of elementary row operations, so they must be independent. We need to try at most ${}_nP_r$ ways of arrangements. Second, apply a sequence of elementary row operations to A such that the leftmost r columns are transformed into I_r. This transformation is unique. Finally, sort the rightmost $n-r$ columns in ascending order, and check $D \preceq D'$.

Our contribution: In the above method, we need to try ${}_nP_r$ ways of arrangements. We reduce the number of ways by improving the definition of canonical forms. To be more precise, we define some invariant for each

*The expected presenter.

column under elementary row operations, and fix the values of the invariant for the leftmost r columns of the canonical form. Let $E = \{e_1, e_2, \ldots, e_n\}$ be a set of columns of A. We define d_i, the degree of e_i $(1 \leq i \leq n)$ as below.

$$d_i = \#\{X \subseteq E \mid e_i \in X \text{ and } X \text{ is independent.}\}$$

The degree d_i is invariant under elementary row operations. Further, the following theorem holds.

Theorem 2. *The multiset of degrees $\{d_1, d_2, \ldots, d_n\}$ is invariant under elementary row operations and column-swapping.*

Therefore, the multiset of degrees is common in each isomorphic class. By using this information, we fix the degrees of the leftmost r columns of the canonical form. Formally, we define $(d_1^*, d_2^*, ..., d_r^*)$ for each isomorphic class, so that the canonical form of the isomorphic class must hold the condition that $d_1 = d_1^*$, $d_2 = d_2^*$, ..., $d_r = d_r^*$. Thereby, we make the number of ways of arrangements as small as possible. We run the following algorithm. The input of the algorithm is A, an $r \times n$ full rank matrix over F_2, and the output is some $(d_1^*, d_2^*, ..., d_r^*)$. First, we calculate a multiset of degrees $\{d_1, d_2, \ldots, d_n\}$ of A. Then, we iterate $\{d'_1, d'_2, ..., d'_r\}$, which is a r-subset of $\{d_1, d_2, \ldots, d_n\}$, in ascending order of the number of ways of arrangements. For each iteration, we check whether there exists an independent r-subset of columns of A such that the corresponding r-subset of degrees is equal to $\{d'_1, d'_2, ..., d'_r\}$. If exists, the algorithm outputs $\{d'_1, d'_2, ..., d'_r\}$ and terminates. Otherwise, the algorithm proceeds to the next iteration.

Theorem 3. *The above algorithm outputs the identical $(d_1^*, d_2^*, ..., d_r^*)$ for every matrix in an isomorphic class.*

Therefore, the canonical form is well-defined for each isomorphic class.

We enumerated (simple) binary matroids of $n \leq 17$ and $r \leq 8$ by using the above method (Table 1). The numbers in bold indicate matroids enumerated for the first time. It took about two days on 32 cores to enumerate matroids $n = 17$ and $r = 8$.

Table 1: The number of non-isomorphic simple binary matroids of size n and rank r

$n \setminus r$	$\cdots$	5	6	7	**8**
$\vdots$					
12		89	700	1285	**821**
13		112	1794	5632	**5098**
14		128	4579	26792	**37191**
15		144	11635	137493	**320663**
16		**145**	**29091**	**745413**	**3186083**
17		**129**	**70600**	**4145064**	**34799393**

References

[1] H. Fripertinger and M. Wild, A catalogue of small regular matroids and their Tutte polynomials, arXiv:1107.1403v1, 2011.

[2] Y. Matsumoto, S. Moriyama, H. Imai and D. Bremner, Matroid enumeration for incidence geometry, *Discrete Comput Geom*, 47:17–43, 2012.

[3] J. G. Oxley, Matroid Theory, *Oxford Graduate Texts in Mathematics*, second edition, 1997.

[4] R. C. Read, Every one a winner or how to avoid isomorphism search when cataloguing combinatorial configurations, *Ann. Discrete Mathematics*, 2:107–120, 1978.

$C_4 \odot P_m-$ Supermagic Labeling and Super $(a,d) - C_4 \odot P_m-$ Antimagic Labeling for a Gear Graph Corona with a Path Graph

Titin Sri Martini, Mania Roswitha, Hardina Sandariria, T A Kusmayadi
Department of Mathematics, Faculty of Mathematics and Natural Sciences
Sebelas Maret University
E-mail : titinsmartini@gmail.com, mania_ros@yahoo.co.id,
sandaririahardina@gmail.com, tri.atmojo.kusmayadi@gmail.com

Abstract

A simple graph $G = (V, E)$ admits an H-covering if every edge in E belongs to a subgraph of G isomorphic to H. A graph G is H-magic if there is a total labeling $f : V(G) \cup E(G) \rightarrow \{1, 2, \dots, |V(G)| + |E(G)|\}$, such that each subgraph $H' = (V', E')$ of G isomorphic to H satisfies

$$f(H') \stackrel{\text{def}}{=} \sum_{v \in V'} f(v) + \sum_{e \in E'} f(e) = m(f),$$

where $m(f)$ is a constant magic sum. Furthermore, f is a $H-$supermagic covering if $f(V) = \{1, 2, \dots, |V(G)|\}$. A simple graph $G = (V, E)$ admits an $(a, d) - H-$ antimagic covering if every edge in E belongs to at least one subgraph of G isomorphic to H and there exists a bijective function $f : V(G) \cup E(G) \rightarrow \{1, 2, \dots, |V(G)| + |E(G)|\}$ such that $m(f)$ constitutes an arithmetic progression $a, a+d, ..., a+(t-1)d$, where a and d are positive integers and t is the number of subgraphs of G isomorphic to H. If $f(V) = \{1, 2, \dots, |V(G)|\}$, then it is called a super $(a, d) - H-$ antimagic covering. This research aims to find $H-$supermagic covering and super $(a, d) - H-$ antimagic covering on $G_n \odot P_m$ with $H = C_4 \odot P_m$.

Keywords : *$H-$supermagic covering, super $(a, d) - H-$ antimagic covering, corona, gear graph, $G_n \odot P_m$ graph.*

Maryati et al.[3] introduced a technique of partitioning a multiset. A multiset is a set that allows the existence of the same elements in it. Let X be a set containing some positive integers. We use the following notation $[a, b]$ to mean $\{x \in N | a \leq x \leq b\}$, and $\sum X$ to mean $\sum_{x \in X} x$. For any $k \in N$, the notation $k + [a, b]$ means $\{k + x | x \in [a, b]\}$. According to Gutiérrez and Lladó [1], the set X is equipartition if there exist k subsets of X, say $X_1, X_2, \dots, X_k$ such that $\bigcup_{i=1}^{k} X_i = X$ and $|X_i| = \frac{|X|}{k}$ for every $i \in [1, k]$
Let Y be a multiset containing positive integers. Y is said to be $k-$ balanced if there exists k subsets of Y, i.e. $Y_1, Y_2, \dots, Y_k$ such that for every $i \in [1, k]$, $|Y_i| = \frac{|Y|}{k}$, $\sum Y_i = \frac{\sum Y}{k} \in N$, and $\uplus_{i=1}^{k} Y_i = Y$. Inayah et al.[2] in 2013 introduced (k, δ)-anti balanced as follows. The multiset Y is $(k, \delta)-$anti balanced if there exist k subsets of Y, say $Y_1, Y_2, \dots, X_k$ such that for every $i \in [1, k]$, $|Y_i| = \frac{|Y|}{k}$, $\bigcup_{i=1}^{k} Y_i = Y$ and for $i \in [1, k]$, $\sum Y_{i+1} - \sum Y_i = \delta$ is satisfied.

We use some lemmas to prove theorems

Lemma 1 *[5] Let x and y be nonegative integers. Let $X = [x+1, x+k]$ with $|X|$=k and Y=$[x+k+1, x+2k]$ where $|Y|$=k. Then, the multiset K=X$\uplus$Y is k-balanced for $j \in [1, k]$.*

Lemma 2 *[3] Let x,y, and z be positive integers. Then multiset* $Y = [x+1, x+k] \uplus [y+1, y+k] \uplus [z+1, z+k]$ *is k-balanced for* $k \geq 3$ *odd.*

Lemma 3 *[4] Let x,y, and k be integers, such that* $1 \leq x \leq y$ *and* $k > 1$. *If X=*$[x, y]$ *and* $|X|$ *is a multiple 2k, then X is k-balanced.*

Then we have the following lemma and theorems

Lemma 4 *Let x, y, and k be nonnegative integers. Let X=*$[x+1, x+k]$ *with* $|X| = k$ *and Y=*$[y+k+1, y+2k]$ *with* $|Y| = k$. *Then, the multiset* $M = X \uplus Y$ *is k-balanced if there is k subsets from* M *where* $M_j = M_1, M_2, \ldots, M_k$.

Theorem 1 *A* $G_n \odot P_m$ *graph is* $C_4 \odot P_m$*-supermagic for n odd and* $m \geq 3$.

Furthermore, the constant supermagic sum of a subgraph $C_4 \odot P_m$ is

$$f(C_4 \odot P_m) = \begin{cases} 18m^2(2n+1) + 14m(2n+1) - n + \lceil \frac{n}{2} \rceil + 7, & \text{for } m \text{ odd;} \\ (8m^2+40m)(2n+1) + 4m + 107n + \lceil \frac{n}{2} \rceil + \lceil \frac{2n+1}{2} \rceil + 43, & \text{for } m = 4\text{ ;} \\ (18m^2+12m)(2n+1) + n(4m-5) + \lceil \frac{n}{2} \rceil + 4\lceil \frac{2n+1}{2} \rceil + 11, & \text{for } m \text{ even, } m \geq 6. \end{cases}$$

Theorem 2 *Let* n *and* m *be positive integers. Then* $G_n \odot P_m$ *is a super* $(a, d) - C_4 \odot P_m-$ *antimagic total labeling for* $d = 1$

References

[1] A. Gutiérrez and A. Lladó, Magic Coverings, J. Combin. Math. Comput. 55 (2005), 43 - 46

[2] N. Inayah, R. Simanjuntak, A.N.M. Salman and K.I.A.Syuhada, Super $(a, d) - H-$ antimagic total labelings for shackles of a connected graph H, Australian Journal of Combinatorics 57 (2013), 127 - 138

[3] T.K Maryati, A.N.M. Salman, E.T. Baskoro, J.Ryan, and M. Miller, On H − Supermagic Labelings for Certain Shackles and Amalgamations of a Connected Graph, *Utilitas Mathematica* 88 (2010), 333-342

[4] T.K Maryati, E. T. Baskoro, and A. N. M Salman, *P_h-supermagic Labelings of Some Trees*, J. Combin. Math. Combin. Computing **65** (2008), 182-189.

[5] Roswitha, M., E. T. Baskoro, T. K. Maryati, N. A. Kurdhi, and I. Susanti, *Further Result on Cycle-Supermagic Labeling*, ACKE Int. J. Graphs Comb. **10** (2013), 211-220.

A Characterization of $(C_n, K_{1,n})$-Supermagic of Trees Corona Paths and Trees Join A Trivial Graph

Yeva Fadhilah Ashari [1], A.N.M. Salman [2], and Khreshna Imaduddin Ahmad Syuhada[3]
Combinatorial Mathematics Research Group, Faculty of Mathematics and Natural Sciences, Institut Teknologi Bandung, Jalan Ganesa no. 10, Bandung 40132, Indonesia
yevafadhilahashari@students.itb.ac.id[1], msalman@math.itb.ac.id[2], khreshna@math.itb.ac.id[3]

Extended Abstract A simple graph $G = (V(G), E(G))$ admits an H-covering, where H is a subgraph of G, if every edge in E belongs to a subgraph of G isomorphic to H. Graph G is H-magic, if there exists a total labeling $f : V(G) \cup E(G) \to \{1, 2, \ldots, |V(G)| + |E(G)|\}$, such that $\sum_{v \in V(H')} f(v) + \sum_{e \in E(H')} f(e) = k_1$ (constant) for every subgraph H' isomorphic to H. Additionally, G admits H-supermagic if $f(V(G)) \to \{1, 2, \ldots, |V(G)|\}$. This concept was introduced by Gutiérrez and Lladó [2]. They studied about star magic coverings in star graphs, complete graphs, and complete bipartite graphs. There have been many results on cycle C_n-supermagic and star $K_{1,n}$-supermagic total labelings of graphs. Lladó and Moragas [4] proved that some graphs such as wheels, windmills, and books are C_n-magic. Ngurah et al. [6] showed that kC_n-paths are C_n-supermagic for all integers $k \geq 2$ and $n \geq 3$; for all integers $n \geq 2$, fans $K_1 + P_n$, triangle ladders Tl_n, and stars join a trivial graph $K_{1,n} + K_1$ are C_3-supermagic; and books B_n are C_4-supermagic for all integers $n \geq 2$. Some other results can be found in [3].
We use a k-balanced multiset method to prove the main theorems. Maryati et al. [5] introduced this method in 2010. A multiset is a set that allows the existence of the same elements in it. In this paper, we use the notation $[a, b]$ for $\{x \in \mathbb{N} | a \leq x \leq b\}$ and the notation $\sum X$ for $\sum_{x \in X} x$. Let k be a positive integer and Y be a multiset that contains positive integers. The multiset Y is said to be a k-balanced multiset, if there exist k subsets of Y, namely $Y_1, Y_2, \ldots, Y_k$, such that $|Y_i| = \frac{|Y|}{k}$, $\sum Y_i = \frac{\sum Y}{k}$, and $\biguplus_{i=1}^{k} Y_i = Y$ for every $i \in [1, k]$.
In 2016, Salman and Ashari [7] generalized the idea of H-magic graph into (H_1, H_2)-magic graph. A graph G admits an (H_1, H_2)-covering, where H_1 and H_2 are two connected subgraphs of G, if every edge in $E(G)$ belongs to at least one subgraph of G isomorphic to H_1 or H_2. The graph G is called (H_1, H_2)-magic, if there are two positive integers k_1 and k_2, and a bijective function $f : V(G) \cup E(G) \to \{1, 2, \ldots, |V(G)| + |E(G)|\}$ such that $\sum_{v \in V(H')} f(v) + \sum_{e \in E(H')} f(e) = k_1$ and $\sum_{v \in V(H'')} f(v) + \sum_{e \in E(H'')} f(e) = k_2$, for every subgraph $H' = (V(H'), E(H'))$ of G isomorphic to H_1 and for every subgraph $H'' = (V(H''), E(H''))$ of G isomorphic to H_2. Moreover, it is said (H_1, H_2)-supermagic, if $f(V(G)) = \{1, 2, \ldots, |V(G)|\}$. In [7], Salman and Ashari proved that some subgraph amalgamations of H_1 and H_2 are (H_1, H_2)-magic. They also found a necessary condition of $(C_n, K_{1,n})$-magic graph for any positive integer $n \geq 3$. Besides that, they proved a characterization of path power graphs being $(C_3, K_{1,3})$-magic. In [1], Ashari and Salman showed that some shackles of H_1 and H_2 are (H_1, H_2)-supermagic.
In this paper, we prove that some classes of graphs are not $(C_n, K_{1,n})$-supermagic for any possible integers n such that the graphs admit $(C_n, K_{1,n})$-covering. We also prove a characterization of trees corona a path P_m being $(C_n, K_{1,n})$-supermagic for $n \in [3, m+1]$ and a characterization of trees join a trivial graph being $(C_n, K_{1,n})$-supermagic for $n = 3$ or 4. We consider that a complete graph K_m admits $(C_n, K_{1,n})$-covering for any positive integer $n \in [3, m-1]$ and a cocktail party graph $CP(s)$ admits $(C_n, K_{1,n})$ for any positive integer $n \in [3, 2(s-1)]$. A cocktail party $CP(s)$ is a graph obtained from a complete graph K_{2s} by deleting s disjoint edges. We have the following two Theorems.

Teorema 1. *Let n and m be positive integers such that $n \geq 3$ and $m \geq 4$. A complete graph K_m is not a $(C_n, K_{1,n})$-supermagic for $n \in [3, m-1]$.*

Teorema 2. *Let n and s be positive integers such that $n \geq 3$ and $s \geq 2$. A cocktail party graph $CP(s)$ is not a $(C_n, K_{1,n})$-supermagic for $n \in [3, 2(s-1)]$.*

Besides that, we investigate a characterization of $(C_n, K_{1,n})$-supermagic in some operation graphs. A graph G corona with H, denoted by $G \odot H$, is a graph which is obtained from G and $|V(G)|$ copies of H, namely $H_1, H_2, ..., H_{|V(G)|}$ then joining every $v_i \in V(G)$ to all vertices in $V(H_i)$ for $i \in [1, |V(G)|]$. We consider that trees corona a path $T \odot P_m$ admits $(C_n, K_{1,n})$-covering for $n \in [3, m+1]$. We have the following Theorem.

Teorema 3. *Let k, m, and n be positive integers such that $k \geq 1$ and $n \in [3, m+1]$. Trees corona path $T \odot P_m$ is $(C_n, K_{1,n})$-supermagic if and only if T is isomorphic to $K_{1,k}$ and $n = m+1$.*

In the last part of this paper, we prove a characterization of $(C_n, K_{1,n})$-supermagic of trees join a trivial graph $T + K_1$ for $n \in \{3, 4\}$. The graph G join H, denoted by $G + H$, is a graph that is obtained from G union H then joining every vertex in G to all vertices in H. Trees join a trivial graph $T + K_1$ admits $(C_3, K_{1,3})$-covering and $(C_4, K_{1,4})$-covering. We have the following two Theorems.

Theorem 4. *Let T be a tree of order at least three. A Tree join a trivial graph $T + K_1$ is $(C_3, K_{1,3})$-supermagic if and only if T is isomorphic to $K_{1,2}$.*

Theorema 5. *Let m be a positive integer such that $m \geq 4$ and T be a tree of order at least four. A tree join a trivial graph $T + K_1$ is $(C_4, K_{1,4})$-supermagic if and only if T is isomorphic to $K_{1,3}$ or path P_m.*

Keywords: $(C_n, K_{1,n})$-covering, $(C_n, K_{1,n})$-supermagic, a trivial graph, corona, join, path, tree

References

[1] Ashari, Y.F., and A.N.M. Salman.: (H_1, H_2)-supermagic labelings for some shackles of connected graphs H_1 and H_2. ***submitted*** (2016)

[2] Gutiérrez, A., and A. Lladó.: Magic coverings. Journal of Combinatorial Mathematics and Combinatorial Computing **55**, 43-56 (2005)

[3] Kojima, T.: On C_4-supermagic labelings of the cartesian product of paths and graphs. Discrete Mathematics **313**, 141-148 (2013)

[4] Llado , A., and J. Moragas.: Cycle-magic graphs. Discrete Mathematics **307**, 2925-2933 (2007)

[5] Maryati, T.K., A.N.M. Salman, E.T. Baskoro, J. Ryan, and M. Miller.: On H-supermagic labelings for certain shackles and amalgamations of a connected graph. Utilitas Mathematica **83**, 333-342 (2010)

[6] Ngurah, A.A.G., A.N.M. Salman, and L. Susilowati.: H-supermagic labelings of graphs. Discrete Mathematics **310 (8)**, 1293-1300 (2010)

[7] Salman, A.N.M., and Y.F. Ashari.: (H_1, H_2)-supermagic labelings of certain graphs. ***submitted*** (2016)

The (strong) 3-rainbow index of amalgamation of some graphs

A.Z. Yumni[1] and A.N.M. Salman[2]
[1,2]Combinatorial Mathematics Research Group, Faculty of Mathematics and Natural Sciences,
Institut Teknologi Bandung, Jalan Ganesa 10, Bandung 40132, Indonesia
[1]zata.yumni@s.itb.ac.id, [2]msalman@math.itb.ac.id

Extended Abstract

All graphs considered in this paper are simple, finite and undirected. We follow the terminology and notation of Diestel [4]. Let G be a nontrivial, connected and edge-colored graph of order n, where adjacent edges may be colored the same. A path P is a *rainbow path*, if no two edges of P receive the same color. The graph G is *rainbow connected*, if G contains a $u-v$ rainbow path for each pair u and v of distinct vertices of G. An edge-coloring of G that results in a rainbow connected graph is a *rainbow coloring* of G. The minimum number of colors needed in a rainbow coloring of G is the *rainbow connection number* $rc(G)$ of G which was first introduced by Chartrand et al. in 2008 [3].

In 2007, Ericksen [5] stated that the terrorist attacks on September 11, 2001 occured because intelligence agencies were not able to communicate with each other through their regular channels from radio systems to databases. Although such information needed to be protected because it is critical to national security, procedures must be in place that permit access between appropriate parties. This issue can be addressed by assigning information transfer paths between agencies which may have other agencies as intermediaries that require a large enough number of passwords that are prohibitive to intruders, yet small enough that any path between agencies has no password repeated. An immediate question arises: What is the minimum number of passwords needed that allow a path between every two agencies so that the passwords along each path are distinct? The minimum number of these passwords is represented by rainbow connection number. In [6], Fitriani and Salman gave the rainbow connection number of amalgamation of some graphs.

Another generalization of rainbow connection number was introduced by Chartrand et al. in 2010 [2]. A tree T in G is a *rainbow tree*, if no two edges of T receive the same color. Let k be an integer with $2 \leq k \leq n$. A *k-rainbow coloring* of G is an edge-coloring of G having property that for every set S of k vertices of G, there exists a rainbow tree T such that $S \subseteq V(T)$. The minimum number of colors needed in a k-rainbow coloring of G is the *k-rainbow index* of G, denoted by $rx_k(G)$. It is obvious that $rx_2(G) = rc(G)$. For every nontrivial connected graph G of order n, it is easy to see that $rx_2(G) \leq rx_3(G) \leq ... \leq rx_n(G)$.

The *distance* $d(u, v)$ of two vertices u and v in G is the length of a shortest $u-v$ path in G. The greatest distance between any two vertices in G is the *diameter* of G, denoted by $diam(G)$. The *Steiner distance* $d(S)$ of a set S of vertices in G is the minimum size of a tree in G containing S. Such a tree is called a *Steiner S-tree* or simply a *Steiner tree*. The *k-Steiner diameter* of G, denoted by $sdiam_k(G)$, is the maximum Steiner distance of S among all sets S with k vertices in G. Thus if $k = 2$ and $S = \{u, v\}$, then $d(S) = d(u, v)$ and $sdiam_2(G) = diam(G)$. In [2], they provided a simple upper bound and a lower bound for $rx_k(G)$ as follows.

Proposition 1 *Let G be a nontrivial connected graph of order $n \geq 3$. For each integer k with $3 \leq k \leq n-1$, $rx_k(G) \leq n-1$, whereas $rx_n(G) = n-1$.*

Proposition 2 *For every connected graph G of order $n \geq 3$ and each integer k with $3 \leq k \leq n$,*

$$k-1 \leq sdiam_k(G) \leq rx_k(G).$$

They showed that trees are composed of a class of graphs whose k-rainbow index attains the upper bound in Proposition 1.

Proposition 3 *Let T be a tree of order $n \geq 3$. For each integer k with $3 \leq k \leq n$, $rx_k(T) = n-1$.*

In this paper, we give a lower bound and an upper bound for the (strong) 3-rainbow index of amalgamation of some graphs. Additionally, we determine the (strong) 3-rainbow index of amalgamation of trees, ladders, and wheels. For simplifying, we define $[a, b] = \{x \in \mathbb{Z} | a \leq x \leq b\}$.

A *strong k-rainbow coloring* of G is an edge-coloring of G having property that for every set S of k vertices of G, there exists a rainbow Steiner S-tree T such that $S \subseteq V(T)$. The minimum number of colors needed in a strong k-rainbow coloring of G is the *strong k-rainbow index* of G, denoted by $srx_k(G)$.

The following definition of *amalgamation* of graphs is taken from [1]. For $t \in \mathbb{N}$ with $t \geq 2$, let $\{G_1, G_2, ..., G_t\}$ be a collection of finite, simple, and connected graphs and each G_i has a fixed vertex v_{o_i} called a *terminal*. The amalgamation $Amal(G_i, v_{o_i})$ is a graph obtained by taking all the G_i's and identifying their terminals. If for each $i \in [1, t]$, $G_i \cong G$ and $v_{o_i} = v$, $Amal(G_i, v_{o_i})$ denoted by $Amal(G, v, t)$.

For $n \geq 3$, a *wheel* W_n is a graph constructed by joining a vertex v to every vertex of a cycle C_n : $v_1, v_2, ..., v_n, v_{n+1} = v_1$. The vertex v is called the *center vertex* of W_n. For each $i \in [1, n]$, edge vv_i is called *spokes* of W_n.

In [3], Chartrand et al. gave a lower bound and an upper bound for strong rainbow connection number of G, that is $diam(G) \leq rc(G) \leq src(G) \leq n-1$. It is easy to see that

$$sdiam_k(G) \leq rx_k(G) \leq srx_k(G) \leq n-1.$$

In the following theorem, we provide a lower bound and an upper bound for the (strong) 3-rainbow index of amalgamation of arbitrary graphs.

Theorem 1 *For $t \in \mathbb{N}$ with $t \geq 2$, let $\{G_i | i \in [1,t]\}$ be a collection of finite, simple, and connected graphs and each G_i has a fixed vertex v_{o_i} called a terminal. If $G \cong Amal(G_i, v_{o_i})$, then*

$$sdiam_3(G) \leq srx_3(G) \leq \sum_{i=1}^{t} srx_3(G_i) \text{ and } sdiam_3(G) \leq rx_3(G) \leq \sum_{i=1}^{t} rx_3(G_i).$$

The upper bound in Theorem 1 is tight. In the next theorem, we provide the existence of an amalgamation of graph whose 3-rainbow index attains the upper bound in Theorem 1.

Theorem 2 *Let $t \in \mathbb{N}$ with $t \geq 2$, T_{n_i} is a tree of order n_i, and v is an anbitrary vertex in T_{n_i}. If for each $i \in [1,t]$, $G_i \cong T_{n_i}$ and $v_{o_i} = v$, then $srx_3\left(Amal(T_{n_i}, v_{o_i})\right) = \sum_{i=1}^{t} srx_3(T_{n_i})$ and $rx_3\left(Amal(T_{n_i}, v_{o_i})\right) = \sum_{i=1}^{t} rx_3(T_{n_i})$.*

In the next theorems, we provide the (strong) 3-rainbow index of amalgamation of wheels.

Theorem 3 *Let $t \in \mathbb{N}$ and $n \in \mathbb{N}$ with $t \geq 2$ and $n \geq 4$, W_n is a wheel of order $n+1$, and v is the center vertex of W_n. If for each $i \in [1,t]$, $G_i \cong W_n$ and $v_{o_i} = v$, then $srx_3\left(Amal(W_n, v, t)\right) = \lceil \frac{n}{2} \rceil t$.*

Theorem 4 *Let $t \in \mathbb{N}$ and $n \in \mathbb{N}$ with $t \geq 2$ and $n \geq 3$, W_n is a wheel of order $n+1$, and v is an arbitrary vertex in W_n where v is not the center vertex. If for each $i \in [1,t]$, $G_i \cong W_n$ and $v_{o_i} = v$, then*

$$srx_3\left(Amal(W_n, v, t)\right) = \begin{cases} t+2, & \text{for } n = 3; \\ 2t+1, & \text{for } n \in \{4,5\}; \\ \left(\left\lceil \frac{n-5}{2} \right\rceil + 1\right) t + 2, & \text{for odd } n \text{ with } n \geq 6; \\ \left(\left\lceil \frac{n-5}{2} \right\rceil + 1\right) t + 1, & \text{for even } n \text{ with } n \geq 6. \end{cases}$$

Theorem 5 *Let $t \in \mathbb{N}$ and $n \in \mathbb{N}$ with $t = 2$ and $n \geq 3$, W_n is a wheel of order $n+1$, and v is the center vertex of W_n. If for each $i \in [1,2]$, $G_i \cong W_n$ and $v_{o_i} = v$, then*

$$rx_3\left(Amal(W_n, v, 2)\right) = \begin{cases} 3, & \text{for } n = 3; \\ 4, & \text{for } n \in [4,8]; \\ 5, & \text{for } n \geq 9. \end{cases}$$

Theorem 6 *Let $t \in \mathbb{N}$ and $n \in \mathbb{N}$ with $t \geq 3$ and $n \geq 10$, W_n is a wheel of order $n+1$, and v is the center vertex of W_n. If for each $i \in [1,t]$, $G_i \cong W_n$ and $v_{o_i} = v$, then $rx_3\left(Amal(W_n, v, t)\right) = 5$ for $n = 5y$ and $y \geq 2$ or $n = 6y$ and $y \geq 2$.*

Keywords : *(strong) 3-rainbow index, amalgamation, rainbow steiner tree, rainbow tree*

References

[1] Carlson, K., Generalized books and C_m-snakes are prime graphs, *Ars Combin.* (2006) **80**, 215-221.

[2] Chartrand, G., F. Okamoto, and P. Zhang, Rainbow trees in graphs and generalized connectivity, *Networks* (2010) **55**, 360-367.

[3] Chartrand, G., G.L. Johns, K.A. McKeon, and P. Zhang, Rainbow connection in graphs, *Math. Bohem.* (2008) **133**, 85-98.

[4] Diestel, R., *Graph Theory* 4^{nd} *Edition*, New York: Springer (2010).

[5] Ericksen, A.B., A matter of security, *Graduating Engineer and Computer Careers* (2007), 24-28.

[6] Fitriani, D. and A.N.M. Salman, Rainbow connection number of amalgamation of some graphs, *AKCE International Journal of Graphs and Combinatorics*, Vol. 13 (1) (2016), 90-99.

Decompositions of the complete twisted graph T_n into isomorphic plane spanning trees.

Eduardo Rivera-Campo
Universidad Autónoma Metropolitana-Iztapalapa

A collection of pairwise edge disjoint subgraphs $G_1, G_2, \ldots, G_m$ of a graph G is a *decomposition* of G if $E(G_1) \cup E(G_2) \cup \cdots \cup E(G_m) = E(G)$.

Let P be a set of points in the plane. A *topological graph* with vertex set P is a simple graph drawn in the plane with edges as curves in such a way that any two edges have at most one point in common.

A *geometric graph* G is a topological graph in which all edges are straight line segments. A geometric graph G is a *convex graph* the vertex set P is in convex position. In [*Computational Geometry: Theory and Applications* 34 (2006), no. 2, 116 - 125], Bose *et al* characterize all decompositions of the complete convex graph C_{2m} into isomorphic plane spanning trees.

A *twisted graph* T_n is a complete topological graph with n collinear vertices $v_1, v_2, \ldots, v_n$ in which two edges $v_i v_j$ $(i < j)$ and $v_s v_t$ $(s < t)$ cross each other if and only if $i < s < t < j$ or $s < i < j < t$ (T_6 and a decomposition of T_6 into plane spanning trees are depicted in Fig 1). For this talk we characterize all decompositions of T_{2m} into isomorphic plane spanning trees.

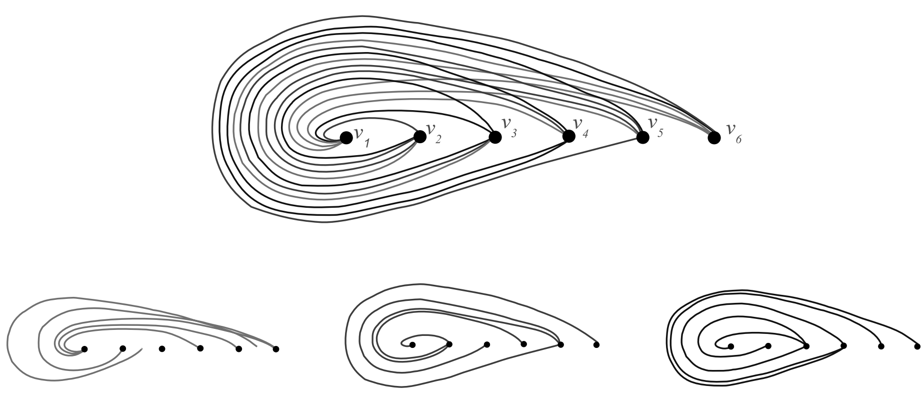


Figure 1: A decomposition of T_6 into three spanning trees.

Continuous Flattening of the set of the square faces in a hypercube

Jin-ichi Itoh[1] and Chie Nara[2]

[1] Faculty of Education, Kumamoto University, Kumamoto, 860-8555, Japan.*j-itoh@kumamoto-u.ac.jp*.
[2] Meiji Institute for Advanced Study of Mathematical Sciences, Meiji University, Tokyo, 164-8525, Japan. *cnara@jeans.ocn.ne.jp*.

Abstract: It is known that we can continuously flatten the surface of a 3-dimensional cube onto any of its faces by moving creases to change the shapes of some faces successively, following the bellow theorem. Let C be a n-dimensional cube with $n \geq 4$, and F be the set of its 2-dimansional faces, in other words, the 2-dimensional skeleton of the square faces in C. We show that F can be continuously flattened onto any face f of F such that the faces of F that are parallel to f do not have any crease, that is, they are rigid during the motion.

There are several results on continuous flattening problems of three dimensional polyhedra (see [1, 2, 3]). But it seems that there are no published results on such problems for high dimensional polyhedra. In this talk, we study such problems on high dimensional cubes (hypercubes) .

Definition 1. We denote by C the 4-dimensional cube described in the 4-dimensional Euclidean space $\mathbb{R}^4$ with the vertex set $V = \{(x, y, z, w) : x = \pm 1, y = \pm 1, z = \pm 1, w = \pm 1\}$. An edge, a face, and a facet of C are a line segment joining two vertices with distance two, a square with area four, and a cube with volume eight, respectively. We denote by E and F the edge set and the 2-dimensional face set, respectively. There are eight facets $C_{x=-1}$, $C_{x=1}$, $C_{y=-1}$, $C_{y=1}$, $C_{z=-1}$, $C_{z=1}$, $C_{w=-1}$, and $C_{w=1}$ whose subscripts show the hyperplanes including them.

Definition 2. Suppose a family $\{F_t : 0 \leq t \leq 1\}$ of sets in $\mathbb{R}^4$ satisfies the following three conditions;
(i) each F_t is composed of (folded) squares and is intrinsically isometric to F, where we allow some parts of F_t to touch each other,
(ii) F_t converges continuously to F_1, and
(iii) $F_0 = F$, and F_1 is a multilayer 2-dimesional flat figure.
We say F is continuously flat-foldable to F_1 (or, F is flat-folded to F_1 by a continuous motion).

Theorem 1. *Let C, F, and f be the four-dimensional cube, the set of the two-dimensional faces of C, and an element of F, respectively. The set F is continuously flat-foldable onto the face f such that the four faces in F parallel to f have no creases, that is, they are rigid during the motion.*

Lemma 1. *Let C and E be the four-dimensional cube and the edge set of C, respectively. For any edge e in E and a facet C_s which does not include e,* *there is a continuous motion from the edge set E to C_s such that the eight edges parallel to e are folded in half and that the other edges are rigid during the motion.*

Lemma 2. *There is a continuous motion from the face set F onto the surface of $C_{w=-1}$ such that all faces in F are folded continuously on $C_{w=-1}$ and that all eight edges parallel to the w-axis are folded in half and the other edges are rigid during the motion.*

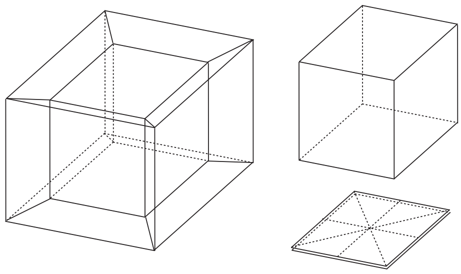


Figure 1: A four-dimensional cube, its facet, and its 2-dimensional flat folded state.

By using the mathematical induction on n, we can prove that for any n-dimensional hypercube with $n \geq 4$, the set of the 2-dimensional faces in the hypercube can be flattened continuously onto any of its 2-dimentional faces f such that all 2-dimensional faces parallel to f have no creases during the continuous motion.

Acknowledgement. The first author is supported by Grant-in-Aid for Scientific Research (B)(15KT0020) and (C) (17K05222), and the second author is supported by Grant-in-Aid for Scientific Research (C)(16K05258).

References

[1] Zachary Abel, Erik Demaine, Martin Demaine, Jin-ichi Itoh, Anna Lebiw, Chie Nara, Joseph O'Rourke, Continuously flattening polyhedra using straight skeletons, Proc. 30th Annual Symposium on Computational Geometry (SoCG), 2014. pp. 396–405.

[2] J. Itoh and C. Nara Continuous flattening of Platonic polyhedra *Computational geometry, Graphs and Applications*, volume 7033 of Lecture Note Comput. Sci., pp. 108–121. Springer-Verlag, 2011.

[3] J. Itoh, C. Nara, and C. Vîlcu. Continuous flattening of convex polyhedra In A. Márquez, P. Ramos and J. Urrutia, editors, *Computational Geometry*, volume 7579 of *Lecture Notes Comput. Sci.*, 85–97. Springer-Verlag, 2012.

Internal continuous flattening of prisms

Kazuki Matsubara[1] and Chie Nara[2]

[1] Faculty of Commerce, Chuo Gakuin University,
Chiba, 270-1196, Japan.
additivestructure@gmail.com

[2] Meiji Institute for Advanced Study of Mathematical Sciences,
Meiji University, Tokyo, 164-8525, Japan.
cnara@jeans.ocn.ne.jp.

We use the terminology *polyhedron* for a polyhedral surface in $\mathbb{R}^3$ which is permitted to touch itself but not self-intersect. There are several ways of continuous flattening of polyhedra in [1, 2, 4, 5, 6, 7]. We propose the open question to find a continuous flattening motion that minimizes its sweeping volume. There is value in investigating this problem because there does not always exist enough area around a polyhedron. On the other hand, in any flattening motion of a polyhedron, it is clear that the spatial area through which the faces of the polyhedron pass must contain the internal area of the original polyhedron. A flattening motion is said to be *internal* if every face does not pass through the outside of the original polyhedron.

We show an example in Figure 1.

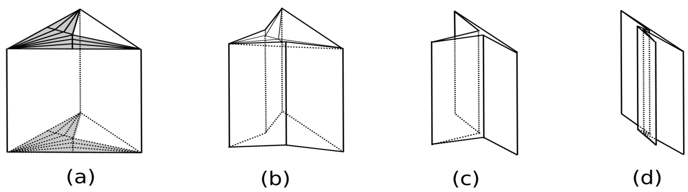


Figure 1: (a) A triangular prism, (b)(c) an internal folding motion, (d) a flattening state.

In this talk, we provide the internal continuous flattening motion of prisms so that there are no moving creases on every side face. Note that the methods given in [2, 6] show the existence of a continuous flattening motion not only for any prism, but also for more general polyhedra defined in [2, 3, 6]. However, the continuous flattening motion provided in the literature is not always internal, and many faces have moving creases. For an internal continuous flattening motion, we focus on the number of faces with moving creases and the number of creases on each face.

Now, an internal continuous flattening motion of a prism with one arbitrary rigid side face can be obtained as follows.

Lemma 1. *For any right triangular prism, there exists an internal continuous* *flattening motion so that one arbitrary side face is rigid and each of other side faces has at most a finite number of creases.*

Theorem 1. *For any right prism, there exists an internal continuous flattening motion so that one arbitrary side face is rigid and each of other side faces has at most a finite number of creases.*

Moreover, an internal continuous flattening motion with two arbitrary rigid side faces can also be obtained for a special class of prisms as follows.

Lemma 2. *For any right triangular prism with bottom face having non-acute angle, there exists an internal continuous flattening motion so that two arbitrary side faces whose dihedral angle is acute are rigid and another side face has only one crease.*

Theorem 2. *For any convex right prism whose bottom face does not have an acute angle, there exists a continuous flattening motion so that two arbitrary side faces are rigid and each of other side faces has at most a finite number of creases.*

Acknowledgement. The second author is supported by Grant-in-Aid for Scientific Research (C)(16K05258).

References

[1] Z. Abel, E. D. Demaine, M. L. Demaine, J.-i. Itoh, A. Lubiw, C. Nara, and J. O'Rourke, Continuously flattening polyhedra using straight skeletons, *Proc. 30th Annual Symposium on Computational Geometry (SoCG)*, pp. 396–405 (2014).

[2] E. D. Demaine, M. L. Demaine, J.-i. Itoh, and C. Nara, Continuous flattening of orthogonal polyhedra, *Proc. JCDCGG 2015*, LNCS, Vol. 9943, pp. 85–93, Springer (2016).

[3] E. D. Demaine and J. O'Rourke, *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*, Cambridge University Press (2007).

[4] J.-i. Itoh and C. Nara, Continuous flattening of Platonic polyhedra, *Proc. Computational geometry, Graphs and Applications (CGGA 2010)*, LNCS, Vol. 7033, pp. 108–121, Springer (2011).

[5] J.-i. Itoh, C. Nara and C. Vîlcu, Continuous flattening of convex polyhedra, *Proc. 16th Spanish Meeting on Computational Geometry (EGC 2011)*, LNCS, Vol. 7579, pp. 85–97, Springer (2012).

[6] K. Matsubara and C. Nara, Continuous flattening of α-trapezoidal polyhedra, to appear.

[7] C. Nara, Continuous flattening of some pyramids. *Elem. Math.* Vol. 69(2), pp. 45–56 (2014).

Infinite All-Layers Simple Foldability

Hugo Akitaya* Cordelia Avery† Joseph Bergeron†
Erik D. Demaine† Justin Kopinsky† Jason Ku†

A classic problem in computational origami is *flat foldability*: given a crease pattern (planar straight-line graph with n edges) on a polygonal piece of paper P, can P be folded flat isometrically without self-intersection while creasing at all creases (edges) in the crease pattern? The problem can also be defined for *assigned* crease patterns, in which every crease is labeled *mountain* or *valley* depending on the direction it is allowed to fold. The decision problem (for both assigned or unassigned) is NP-hard [5], even when the paper is an axis-aligned rectangle and the creases are at multiples of 45° [2]. But even when a crease pattern does fold flat, the motion to achieve that folding can be complicated [6], making the process impractical in some physical settings.

Motivated by practical folding processes in manufacturing such as sheet-metal bending, Arkin et al. [3] introduced the idea of *simple foldability*—flat foldability by a sequence of simple folds. Informally, a *simple fold* is defined by a line segment and rotates a portion of the paper around this segment by ±180°, while avoiding self-intersection. The problem generalizes to *d*-dimensions. In particular, for 1D paper, P is a line segment and creases are defined by points in P. In [3], they defined several models for simple folds and, for many models, showed that deciding simple foldability is polynomial for 1D paper, polynomial for rectangular paper with axis-aligned creases, weakly NP-complete for rectangular paper with creases at multiples of 45°, and weakly NP-complete for orthogonal paper with axis-aligned creases. In particular, they provided an algorithm to determine simple foldability of a 1D paper in $O(n \log n)$ deterministic time and $O(n)$ randomized time in the *all-layers* model, requiring that a simple fold through one crease, also folds through all layers overlapping that crease. Akitaya et al. [1] extended the list of simple folding models, and for many models showed simple foldability to be strongly NP-hard for 2D paper. In particular, they introduced the *infinite all-layers* model of simple folds for 2D paper which is studied here, requiring that each simple fold be defined by an infinite line, and that all layers of paper intersecting this line must be folded. This model is probably the most practical simple folding model; for example, Balkcom's robotic folding system [4] is restricted to this model.

In this paper, we improve on [3] giving a deterministic $O(n)$-time algorithm to decide simple foldability of 1D crease patterns in the all-layers model. Then, we prove two results concerning the complexity of one of the few remaining open problems in this area [1]: infinite all-layers simple foldability on *orthogonal crease patterns*, axis-aligned orthogonal 2D paper with axis-aligned creases. First, we prove that this problem can be solved in linear time when creases are *fully unassigned*. On the other hand, when the creases are *partially assigned* (some creases must fold mountain, some creases must fold valley, while others can freely fold mountain or valley), this problem becomes strongly NP-complete, even for an axis-aligned rectangle of paper.

Theorem 1 *All-layers simple foldability of a 1D crease pattern can be decided in deterministic linear time.*

Proof sketch: We reduce the problem to a "string folding" problem as in [3], representing the input as a string of the form $\ell_0 d_1 \ell_1 d_2 \dots d_{n-1} \ell_{n-1} d_n \ell_n$ where each $d_i \in \{M, V\}$ represents the assignment of the i-th crease and $\ell_i \in \mathbb{R}$ represents the length of the i-th uncreased line segment in P. For an instance to be simple foldable in this model, any fold must map a crease onto another crease of opposite assignment. After a fold is performed, we obtain a smaller crease pattern by ignoring paper overlap. By [3] the smaller crease pattern is simple foldable if and only if the original one is. The *size* of a fold is defined by the difference on the length of the strings representing the crease patterns. We adapt the algorithm in [7] to recognize the smallest possible fold in a crease pattern that runs in linear time on the size of the output fold, leading to an amortized linear time algorithm overall. Unassigned crease patterns can also be solved by a simple modification of this algorithm.

Theorem 2 *Infinite all-layers simple foldability of a fully unassigned orthogonal crease pattern can be decided in deterministic linear time.*

Proof sketch: We first provide a linear time reduction of infinite all-layers simple foldability of unassigned orthogonal crease patterns to instances on rectangular paper. Such instances are equivalent in the finite and infinite all-layers models [2]. We then reduce the problem on a rectangle to two instances of 1D simple foldability which by Theorem 1 can each be decided in deterministic linear time.

*Tufts University, hugo.alves_akitaya@tufts.edu
†CSAIL, Massachusetts Institute of Technology, {cavery, jbergero,edemaine,jkopin,jasonku}@mit.edu

Theorem 3 *Deciding infinite all-layers simple foldibility of partially assigned orthogonal crease patterns is NP-Complete, even for creases on a square grid on rectangular paper.*

Proof sketch: The problem is in NP as a valid sequence of simple folds represents a certificate of at most linear size that can be checked in polynomial time. We show NP-hardness via a reduction from 3SAT. Given an instance of 3SAT on n variables and m clauses, we build a partially assigned simple foldability instance as illustrated in Figure 1. Informally, yellow dots on the same vertical line represent a clause. The partial assignment forces any legal sequence of simple folds to fold through either t_1 or f_1 but not both, forcing a yellow dot onto either a green or red dot respectively, encoding the boolean assignment of the variable x_1. A vertical fold on the right edge of the paper must occur next, followed by folding along either t'_1 or f'_1, consistent with whether t_1 or f_1 was initially chosen. These folds force the yellow dots directly below t_1 and f_1 to coincide with the yellow dots directly above t_1 and f_1, and adds a valley assignment to a crease incident to a yellow dot if its corresponding clause contains a literal involving x_1 that evaluates to FALSE. We apply induction on the resulting crease pattern to bring all the yellow dots to lie on top of the m upper-most yellow dots. A crease pattern containing a vertex incident to only valley creases is not flat foldable. After folding as described for any given assignment of the variables, we show that the each resulting yellow dot will be incident to at least one non-valley crease if and only if the SAT instance has a positive solution, and that the resulting crease pattern can be folded by a sequence of infinite all-layers simple folds.

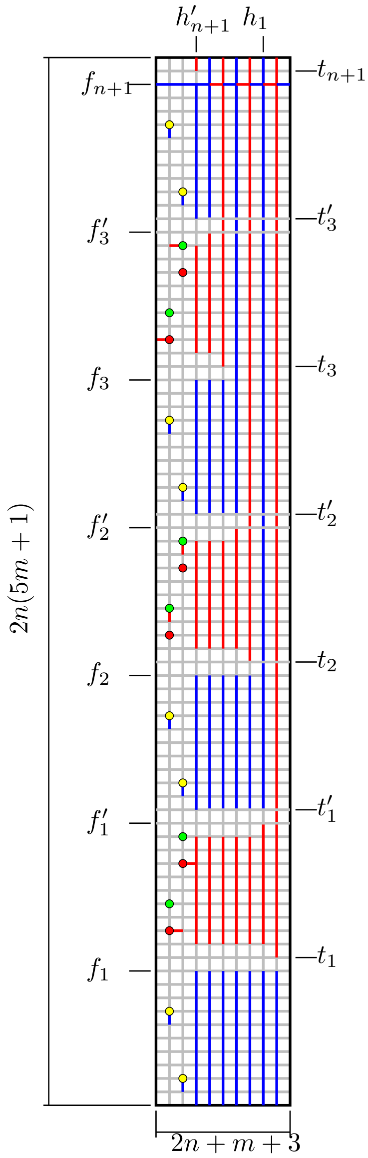


Figure 1: Reduction from the instance $(x_1 \vee \overline{x_2} \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3})$ of 3SAT to partially assigned simple foldability under the infinite all-layers model.

Acknowledgments. This work was initiated during an open problem session associated with MIT class 6.849 (Geometric Folding Algorithms) taught by E. Demaine. We thank the other participants of that class for providing a stimulating research environment. Supported in part by NSF grants EFRI-1240383, CCF-1138967 and CCF-1422311, and Science without Borders.

References

[1] H. Akitaya, E. D. Demaine, and J. S. Ku. Simple folding is really hard. *Journal of Information Processing*. To appear. Special issue of papers from the 19th Japan Conference on Discrete and Computational Geometry, Graphs, and Games, September 2016.

[2] H. A. Akitaya, K. C. Cheung, E. D. Demaine, T. Horiyama, T. C. Hull, J. S. Ku, T. Tachi, and R. Uehara. Box pleating is hard. In *Revised Papers from the 18th Japan Conference on Discrete and Computational Geometry and Graphs*, pages 167–179. 2015.

[3] E. M. Arkin, M. A. Bender, E. D. Demaine, M. L. Demaine, J. S. B. Mitchell, S. Sethia, and S. S. Skiena. When can you fold a map? *Computational Geometry: Theory and Applications*, 29(1):23–46, 2004.

[4] D. J. Balkcom and M. T. Mason. Robotic origami folding. *The International Journal of Robotics Research*, 27(5):613–627, 2008.

[5] M. Bern and B. Hayes. The complexity of flat origami. In *Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 175–183, 1996.

[6] E. D. Demaine and J. S. B. Mitchell. Reaching folded states of a rectangular piece of paper. In *Proceedings of the 13th Canadian Conference on Computational Geometry*, pages 73–75, Waterloo, August 2001.

[7] G. Manacher. A new linear-time "on-line" algorithm for finding the smallest initial palindrome of a string. *J. ACM*, 22(3):346–351, July 1975.

Which convex polyhedra can be made by gluing regular hexagons?⋆

Elena Khramtcova⋆⋆ and Stefan Langerman

Computer Science Department, Université libre de Bruxelles (ULB), Belgium,
{elena.khramtcova,stefan.langerman}@ulb.ac.be

Abstract. Which convex 3D polyhedra can be obtained by gluing several regular hexagons edge-to-edge? It turns out that there are only 15 possible types of shapes, 5 of which are doubly-covered 2D polygons. We give examples for most of the them, including all simplicial and all flat shapes, and give a characterization for the latter ones. It is open whether the remaining can be realized.

1 Introduction

Given a 2D polygon P, which convex 3D polyhedra can be obtained by folding it and gluing its boundary to itself? Alexandrov's theorem [1] states that for any gluing pattern homeomorphic to a sphere that does not yield the total facial angle more than 2π at any point, there is a unique 3D convex polyhedron that can be constructed in this manner. Nevertheless, answering the above question requires checking exponentially many possibilities [2]. There are two ways to restrict the setting: to consider a particular polygon (e.g., all regular polygons [4], and the Latin cross [3] were studied), or to consider only *edge-to-edge* gluing, where an edge of P needs to be glued to an entire other edge of P [4]. We are interested in gluing together several copies of a same regular polygon edge-to-edge, thus fusing these two settings, while at the same time extending and restricting each of them. The case of regular k-gons for $k > 6$ is trivial. Indeed, since gluing three k-gons in one point would violate the above Alexandrov's condition, the only two possibilities are: two k-gons glued together and forming a doubly covered k-gon, or one k-gon folded in half (if k is even). Thus the first interesting case is $k = 6$, and we study it here. Note that the problem we are solving here for $k = 6$ is actually decidable (in constant time) for any constant k by Tarski's theorem, but the problem is probably too large even for $k = 6$ to be handled by any existing computer.

2 Gluing regular hexagons

Let P be a convex 3D polyhedron. Gaussian curvature at a vertex v of P equals $2\pi - \sum_{j=1}^{t} \alpha_j^v$, where t is the number of faces of P incident to v, and α_j^v is the angle of the j-th face incident to v. Gaussian curvature at each vertex of P is non-negative.

Theorem 1 (Gauss, Bonnet'1848). *The total sum of the Gaussian curvature at each vertex of a 3D polyhedron P equals 4π.*

Let P be a convex polyhedron obtained by gluing several regular hexagons edge-to-edge. Vertices of P are vertices of the hexagons, and the sum of facial angles around a vertex v of P equals $2\pi/3$ (the interior angle of the regular hexagon) times the number of hexagons glued together at v. Since the Gaussian curvature at v is in $(0, 2\pi)$, the number of hexagons glued at v can be either one or two, implying the Gaussian curvature of v to be respectively $4\pi/3$ or $2\pi/3$. If three hexagons are glued at a point p, p has zero Gaussian curvature, and thus is a (flat) point on the surface of P. Thus P has at most 6 vertices.

There are 10 distinct 3-connected simple planar graphs of at most 6 vertices; these are all combinatorially different graph structures of convex polyhedra of at most 6 vertices. There also are 4 combinatorially different doubly-covered plane polygons that can be obtained by gluing hexagons. The quadrilaterals come in 2 variants depending on the sequence of their angles. Thus we list 5 types of polygons.

Below we give examples for different polyhedra obtained by gluing regular hexagons. Namely we give an example for each doubly-covered flat polygon, for each simplicial polyherdon, and for one non-simplicial polyhedron. It remains open whether all the non-simplicial polyhedra can be constructed as well (five polyhedra are in question).

Doubly-covered flat polygons (see Figure 1):

(a) Equilateral triangle.
(b) Quadrilateral with angles $\pi/3, \pi/3, 2\pi/3, 2\pi/3$. This is a parallelogram.
(c) Quadrilateral with angles $\pi/3, 2\pi/3, \pi/3, 2\pi/3$. This is an isosceles trapezoid.

⋆ E. K. was supported by F.R.S.-FNRS, and by the SNF Early PostDoc Mobility grant P2TIP2-168563. S. L. is directeur de recherches du F.R.S.-FNRS.

⋆⋆ E. K. is the expected presenter.

(d) Pentagon with 1 angle $\pi/3$, and 4 angles $2\pi/3$.
(e) Regular hexagon.

We further characterize such shapes:

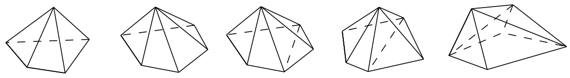
Proposition 1. *Any polygon of type (a)–(e), that can be drawn on the hexagonal grid, can be obtained by gluing regular hexagons. (See Appendix.)*

(Non-flat) Polyhedra (see Figure 2):

(i) Tetrahedron.
(ii) Hexahedron with 5 vertices (3 vertices of degree 4, and 2 vertices of degree 3), and 6 triangular faces.
(iii) Octahedron with 6 vertices of degree 4 each, and 8 triangular faces. In our example, it is a regular octahedron.
(iv) Octahedron with 6 vertices (two of which are of degree 5, two of degree 4, and two of degree 3) and 8 triangular faces.
(v) Pentahedron with 6 vertices of degree 3 each, and 5 faces (2 triangles, and 3 quadrilaterals). In our example, it is a triangular prism.

We do not characterize these polyhedra in terms of side lengths, as opposed to the case of polygons.

Open question. Can the following polyhedra be realized by gluing regular hexagons?



References

1. Alexandrov, A.: Convex Polyhedra. Springer-Verlag, Berlin (2005), translation from Russian
2. Demaine, E., Demaine, M., Lubiw, A., O'Rourke, J.: Enumerating foldings and unfoldings between polygons and polytopes. Graphs and Combinatorics 18(1), 93–104 (2002)
3. Demaine, E., Demaine, M., Lubiw, A., O'Rourke, J., Pashchenko, I.: Metamorphosis of the cube. In: Proc. 15th Annu. ACM Symp. Comput. Geom. pp. 409–410. ACM (1999), video and abstract
4. Demaine, E., O'Rourke, J.: Geometric folding algorithms. Cambridge university press (2007)

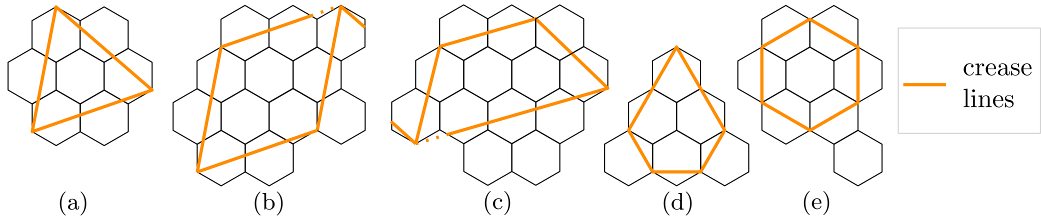


Fig. 1. Doubly-covered polygons (a)-(e): Their nets and crease lines

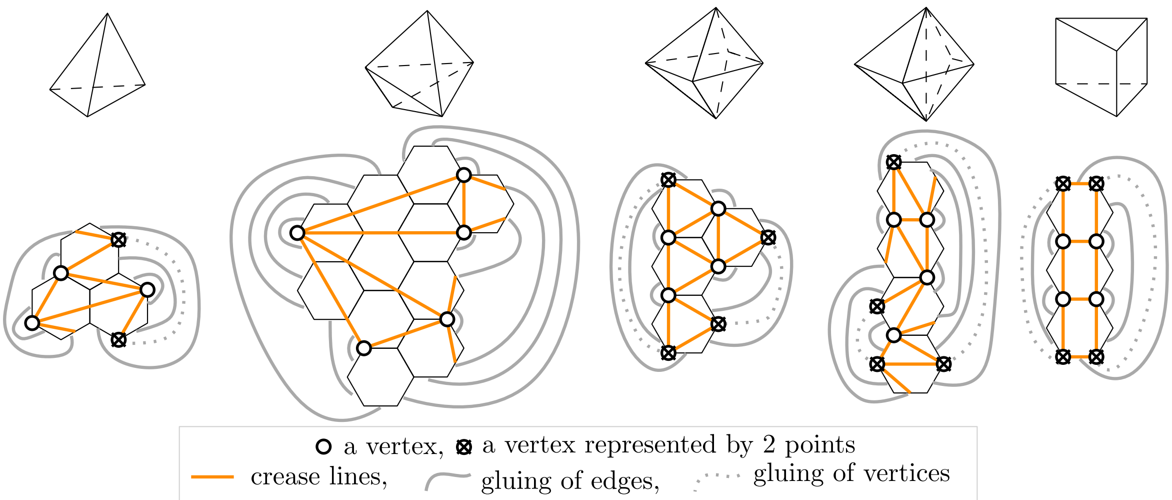


Fig. 2. Examples of polyhedra (i)–(v). Above: graphs of their skeletons. Below: their nets, crease lines, and gluing rules.

Some extension of reversing a polyhedral surface

Naofumi Horio#, Jin-ichi Itoh

Kumamoto University Graduate School of Education#, Kumamoto University Faculty of Education
holynet@me.com#, j-itoh@kumamoto-u.ac.jp

Recently H. Maehara considered the reversibility of several polyhedral surfaces in [1]. He defined an origami-deformation of a polyhedral surface in $\mathbf{R}^3$ and he showed that if an orientable polyhedral surface with boundary is reversible, then its genus is 0, and for every interior vertex, the sum of face angles at the vertex is at least 2π. He showed that every rectangular tube and some polyhedra (obtained by tube-attachment operation) can be subdivided so that it becomes reversible (it is called s-reversible). Moreover he posed several open problems.

In this talk we continue his research and announce several results. Before that, we have to stand for his main theorem.

Theorem. *Every rectangular tube is s-reversible.*

Now we will announce our new results. First we discuss a 3 times extended cube with 4 square holes XC_4 (see Figure 1). We define a semi-flattening operation and deform XC_4 to a subset of square tube and use Maehara's result.

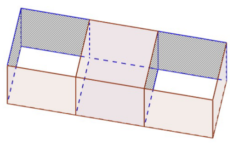


Figure 1: XC_4

A semi-flattening operation is defined by the following way. We make creases as shown in Figure 2 on the three consecutive square faces which are perpendicular to the bottom plane and also stand perpendicularly each other, and deform continuously as Figure 2. At last the middle face is flattened and the other two side faces are folded along their diagonals.

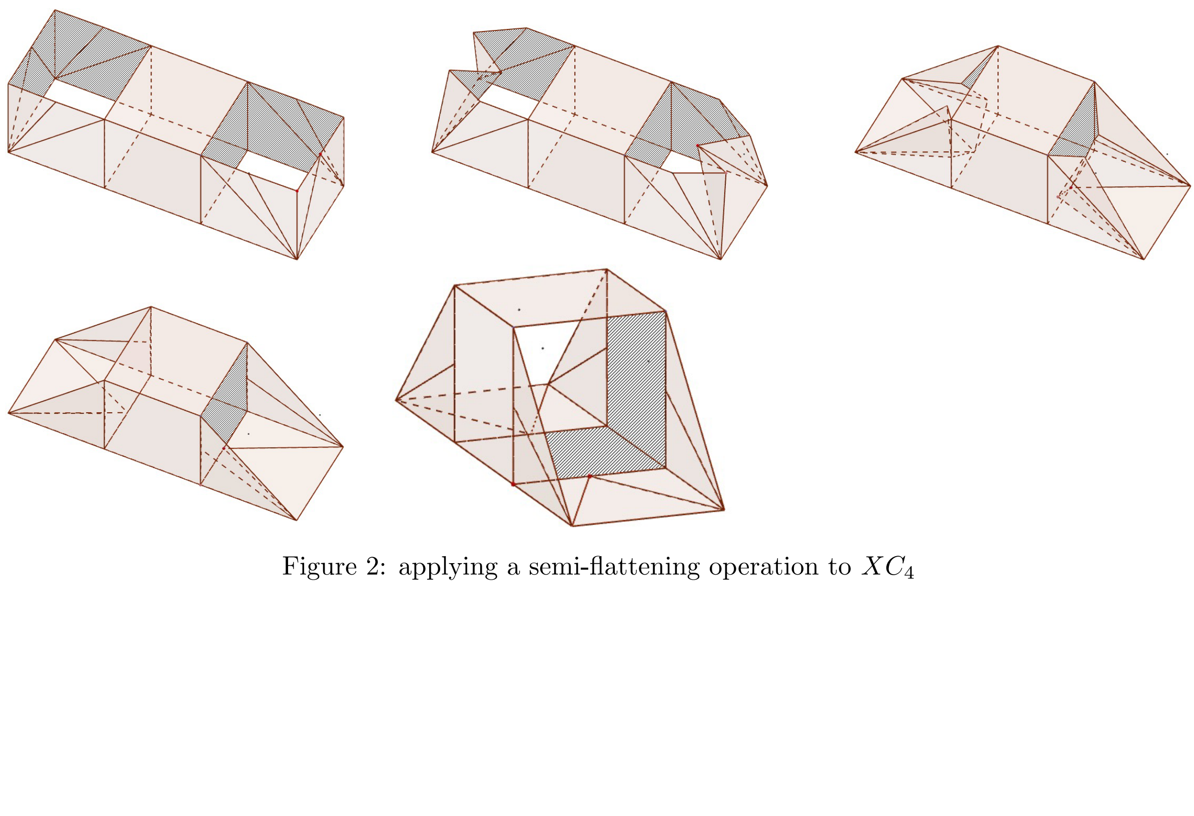


Figure 2: applying a semi-flattening operation to XC_4

Proposition 1. *XC_4 is s-reversible.*

Next we discuss a surface of cube with 8 holes C_8 (which is a surface of the unit cube removed 8 neighborhoods around each vertex, where each neighborhood is composed of 3 squares whose edge length is equal to 1/3)(see Figure 3).

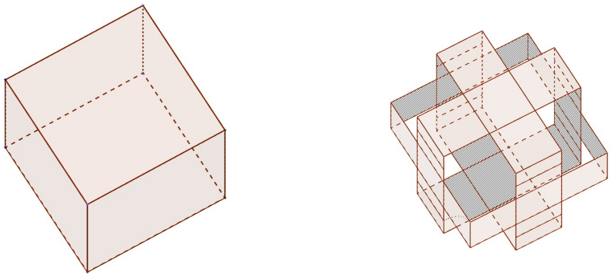


Figure 3: make a cube to C_8

Proposition 2. *C_8 is s-reversible.*

C_8 can be deformed into XC_4. The procedure is as follows (see Figure 4). Then we can get the result.

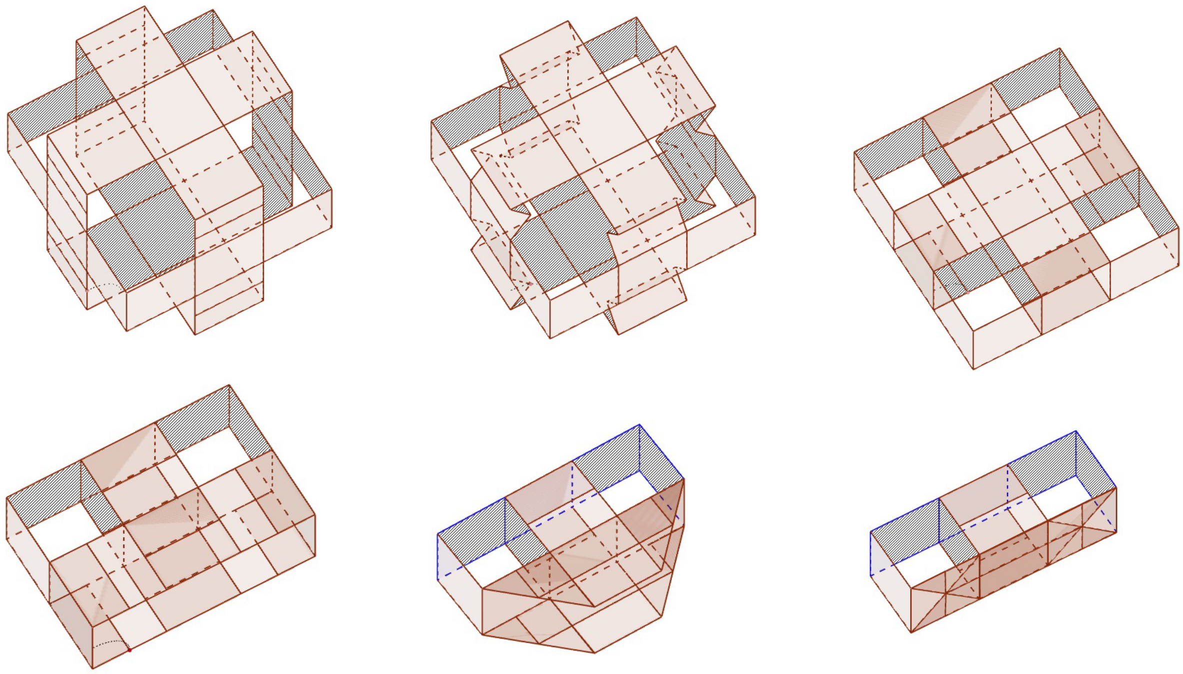


Figure 4: a procedure of folding C_8 to XC_4

Finally we define a cubical-tube-unit attachment operation and get the following theorem.

Theorem. *Every polyhedral surface made of cubical-tube-unit-attachment operation is s-reversible.*

References

[1] H. Maehara, *Reversing a polyhedral surface by origami-deformation*, European Journal of Combinatorics, **31** (2010), 1171-1180

The bipartite cylindrical crossing number of $K_{n,m}$

B. M. Ábrego[*], S. Fernández-Merchant[*][†], A. Sparks[‡]

Abstract

A bipartite cylindrical drawing of the complete bipartite graph $K_{m,n}$ is a drawing on the surface of a cylinder, where the vertices are placed on the top and bottom rims of the cylinder, one vertex-partition per rim, and the edges do not cross the rims of the cylinder. The bipartite cylindrical crossing number $cr_{\circledcirc}(K_{m,n})$ of $K_{m,n}$ is the minimum number of crossings among all bipartite cylindrical drawings of $K_{m,n}$. We determine $cr_{\circledcirc}(K_{m,n})$.

1 Introduction

The *crossing number* of a graph G, denoted by $cr(G)$, is the minimum number of edge-crossings over all drawings of G on the plane. We consider a subclass of drawings in the plane. A *cylindrical drawing* of a graph has the vertices placed along the top and bottom rims of a cylinder and the edges lie on the surface of the cylinder and do not cross the rims. We look specifically at *bipartite cylindrical drawings* of the complete bipartite graph $K_{m,n}$, where we impose the additional condition that the m-set of vertices is placed on one rim and the n-set of vertices on the other rim. (See Figure 1, left.) The *bipartite cylindrical crossing number* $cr_{\circledcirc}(K_{m,n})$ of the complete bipartite graph $K_{m,n}$ is the minimum number of crossings among all bipartite cylindrical drawings of $K_{m,n}$.

Cylindrical drawings can be represented in the plane by placing the vertices on two concentric circles corresponding to the top and bottom rims of the cylinder: the *inner boundary* C_i and the *outer boundary* C_o. Therefore, bipartite cylindrical drawings of $K_{m,n}$ in the plane have the m-set of vertices on C_i, the n-set of vertices on C_o, and all edges lie between C_i and C_o. (See Figure 1, right.) We determine $cr_{\circledcirc}(K_{m,n})$ in this setting.

These type of drawings were first studied by Harary and Hill [2], who constructed cylindrical drawings of the complete graph K_n with few crossings. These drawings consist of a bipartite cylindrical drawing of $K_{\lfloor n/2\rfloor,\lceil n/2\rceil}$ extended to a drawing of K_n by adding straight line segments between vertices in the same rim of the cylinder. They conjectured that their drawings achieve $cr(K_n)$. This conjecture was recently proved by Ábrego et al. [1] for the class of cylindrical drawings of K_n. The crossing number of a graph restricted to cylindrical drawings whose edges completely remain in the region between the two rims/boundaries is also called the *annulus crossing number* [4].

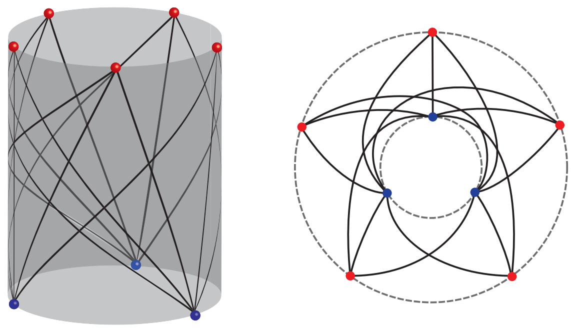


Figure 1: *Left:* A cylindrical drawing of $K_{3,5}$ drawn on a cylinder. *Right:* The same cylindrical drawing (up to isomorphism) represented in the plane.

2 Results

In 1997, Richter and Thomassen [3] proved that the bipartite cylindrical crossing number of $K_{n,n}$ is

$$cr_{\circledcirc}(K_{n,n}) = n\binom{n}{3}.$$

In this work, we extend this results to unbalanced complete bipartite graphs.

Theorem 1. *If $m \le n$, the bipartite cylindrical crossing number of $K_{m,n}$ is*

$$\begin{aligned} cr_{\circledcirc}(K_{m,n}) &= \binom{n}{2}\binom{m}{2} \qquad (1)\\ &+ \sum_{1\le i<j\le m}\left(\left\lfloor \frac{n}{m}(j-1)\right\rfloor - \left\lfloor \frac{n}{m}(i-1)\right\rfloor\right)^2 \\ &- n\sum_{1\le i<j\le m}\left(\left\lfloor \frac{n}{m}(j-1)\right\rfloor - \left\lfloor \frac{n}{m}(i-1)\right\rfloor\right) \end{aligned}$$

[*]California State Univ., Northridge, [bernardo.abrego, silvia.fernandez]@csun.edu.
[†]Supported by the NSF grant DMS-1400653.
[‡]Univ. Colorado Boulder, athena.sparks@colorado.edu

When m divides n, this reduces to the following result, which agrees with that in [3] when $m = n$.

Theorem 2. *If m divides n, the bipartite cylindrical crossing number of $K_{m,n}$ is*

$$cr_{\circledcirc}(K_{m,n}) = \frac{1}{12} n(m-1)(2mn - 3m - n). \quad (2)$$

3 Sketch of proofs

It is known that for any graph G the minimum $cr(G)$ can be restricted to *good drawings* of G, that is, drawings where (a) no edge crosses itself, (b) two edges that share a vertex do not cross, and (c) two edges with no shared vertices cross each other at most once. Any drawing with one of these crossings is not optimal because for each of these three types of crossings, the edges can be altered slightly to produce a drawing of the same graph with fewer crossings. These modifications preserve the property of being a bipartite cylindrical drawing. Then from now on, we only consider good bipartite cylindrical drawings of $K_{m,n}$.

We use the notation in [3]. Given a good bipartite cylindrical drawing D of $K_{m,n}$ in the plane, label the vertices on C_i clockwise 1 through m and the vertices on C_o clockwise 1 through n. For each vertex $i \in C_i$, there is a unique vertex $x_i \in C_o$ such that the edges from i to x_i and i to $x_i + 1 \pmod{n}$ together with the segment of C_o that connects x_i to $x_i + 1 \pmod{n}$ clockwise creates a simple closed curve that contains C_i. (See Figure 2, left.) With this labeling system, it was proved in [3] that the number of crossings in D is given by

$$\sum_{1 \le i < j \le m} \binom{|x_j - x_i|}{2} + \binom{n - |x_j - x_i|}{2}.$$

Assuming without loss of generality that $x_1 = 1$, we prove that $1 = x_1 \le x_2 \le \cdots \le x_m \le n$ and thus

$$\mathrm{cr}(D) = F(x_1, x_2, \ldots, x_m) = \binom{n}{2}\binom{m}{2} + \sum_{1 \le i < j \le m} (x_j - x_i)^2 - n \sum_{1 \le i < j \le m} (x_j - x_i).$$

Moreover, given a sequence $1 = x_1 \le x_2 \le \cdots \le x_m$, there is a unique good cylindrical drawing of $K_{m,n}$, up to isomorphism, and this drawing has $F(x_1, x_2, ..., x_m)$ crossings. (See Figure 2,right.)

We minimize F over all integer m-tuples $1 = x_1 \le x_2 \le \ldots \le x_m \le n$. As in [3], we first determine when the real-valued function F attains its minimum. Setting the partial derivatives of F to 0,

$$\frac{\partial F}{\partial x_i} = 2mx_i - 2 \sum_{1 \le j \le m}^{x_j} + n(m - 2i + 1) = 0,$$

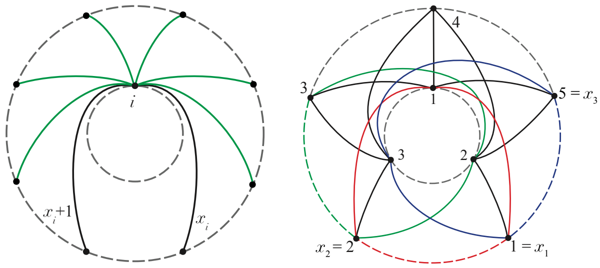


Figure 2: *Left:* The vertex $x_i \in C_o$ assigned to the vertex $i \in C_i$. *Right:* The unique good cylindrical drawing of $K_{3,5}$ with $x_1 = 1$, $x_2 = 2$, $x_3 = 5$ and $F(1, 2, 5) = 16$ crossings.

yields

$$x_i = \frac{1}{m} \sum_{1 \le j \le m} (x_j - n(m - 2i + 1)).$$

So F achieves its (real) minimum when $x_{i+1} - x_i = \frac{n}{m}$. Setting $x_1 = 1$, gives that the minimum of F is attained when $x_i = \frac{n}{m}(i-1) + 1$ for all $1 \le i \le m$. When m divides n, these values are integers between 1 and n, which in turn proves Identity (2).

In the case when m does not necessarily divide n, the previous argument suggests that the minimum of F when restricted to m-tuples of integers between 1 and n should be achieved when x_i is close to $\frac{n}{m}(i-1) + 1$. In fact, we prove that this (discrete) minimum of F is achieved when $x_i = f_i$ for $1 \le i \le m$, where $f_i = \lfloor \frac{n}{m}(i-1) + 1 \rfloor$.

Using a partial variations argument, we first prove that the discrete minimum is achieved when each x_i is either f_i or $f_i + 1$. Then, for $I \subset \{1, 2, \cdots, m\}$, we study the difference $H(I) = F(x_1, x_2, \ldots, x_m) - F(f_1, f_2, \ldots, f_m)$, where $x_i = f_i + 1$ for $i \in I$ and $x_i = f_i$ for $i \notin I$. Our goal is then to prove that $H(I) \ge 0$, which in turn proves Identity (1). We use induction on $|I|$ and several number theory lemmas that focus on the residual classes of $\{n(i-1) : 1 \le i \le m\}$ modulo m.

References

[1] B.M. Ábrego, O. Aichholzer, S. Fernández-Merchant, P. Ramos, and G. Salazar, Shellable drawings and the cylindrical crossing number of K_n. *Discrete Comput. Geom.* **52-4** (2014) 743–753.

[2] F. Harary and H. Hill, On the number of crossings in a complete graph, *Proc. Edinburgh Math. Soc.* **13** (1963) 333–338.

[3] R.B. Richter and C. Thomassen, Relations between crossing numbers of complete and complete bipartite graphs. *The American Mathematical Monthly* **104-2** (1997) 131–37.

[4] Schaefer, M.: The Graph Crossing Number and its Variants: A Survey, *E. J. Combinatorics* DS21 (2014).

Minimizing crossings of 2-page drawings of K_n with prescribed number of edges in each page

B. M. Ábrego,* S. Fernández-Merchant*†, E. Lagoda‡, and P. Ramos§

Abstract

We consider the problem of determining the 2-page book crossing number of the complete graph K_n when the number of edges in each page is given. We find upper and lower bounds of the right order of magnitude depending on the number of edges in the page with the least number of edges.

1 Introduction

A *k-page book* is the union of k-half planes in the space with common boundary and disjoint interiors. The common boundary is a straight line called the *spine* and the half planes are called *pages*. We are concerned only with 2-page book drawings of K_n, that is, drawings of K_n such that the vertices are placed on the spine and each edge (except by its endpoints) is completely contained in the interior of a single page. Ábrego et al. [1] proved Harary-Hill's conjecture (see [3] and [2]) for 2-page drawings, that is, they proved that a 2-page drawing of the complete graph has at least $\frac{1}{4}\lfloor\frac{n}{2}\rfloor\lfloor\frac{n-1}{2}\rfloor\lfloor\frac{n-2}{2}\rfloor\lfloor\frac{n-3}{2}\rfloor$ crossings. This inequality is tight; in fact the authors of [1] partially classified the drawings that achieve equality and, as expected, these drawings have essentially the same number of edges in each page. In this paper we consider the problem of minimizing $\nu_2(K_n, r)$, the number of crossings of the complete graph K_n on n vertices where r edges are drawn on one page (the red edges) and the remaining $\binom{n}{2} - r$ edges are drawn on the other page (the blue edges). Given n and r, this problem is equivalent to determining the minimum number of monochromatic crossings (red-red or blue-blue) over all different ways to color exactly r red and $\binom{n}{2} - r$ blue diagonals of the regular n-gon.

*California State Univ., Northridge, [bernardo.abrego, silvia.fernandez]@csun.edu.

†Supported by the NSF grant DMS-1400653.

‡Berlin Mathematical School, lagoda@math.tu-berlin.de

§Universidad de Alcalá, pedro.ramos@uah.es

2 Results

Let D be a 2-page drawing of the complete graph on n vertices with exactly r edges in one page and $\binom{n}{2} - r$ on the other page. We call R the set of r red edges, and B the set of the remaining $\binom{n}{2} - r$ blue edges. In this paper we concentrate on the case when $r = o(n^2)$. The full range of values of r will be included in the full version. For simplicity, we assume that n is even. Our main result follows.

Theorem 1. *If $r = o(n^2)$, then*

$$\frac{1}{2}r^2 - o(r^2) \le \nu_2(K_n, r) - \binom{n}{4} + \frac{1}{4}r(n-2)^2 \le 0.6109r^2 + o(r^2).$$

The proof of this theorem is outlined in what follows. The proofs of the lemmas are omitted for lack of space. Using the model of the regular n-gon, we define $\text{cr}(X, Y)$ as the number of crossings of edges in X with edges in Y. If $X = \{e\}$, we write $\text{cr}(e, Y)$ instead of $\text{cr}(\{e\}, Y)$. In addition, every edge e separates $(n-2)/2 - i(e)$ points from the remaining $(n-2)/2 + i(e)$ points, where $0 \le i(e) \le (n-2)/2$ is twice the difference of the absolute value of the difference of points in each side.

Theorem 2. *If D is a 2-page drawing of the complete graph on n vertices with exactly r edges in one page, then*

$$\text{cr}(D) = \binom{n}{4} - \frac{1}{4}r(n-2)^2 + \sum_{e \text{ red}} \left(\text{cr}(e, R) + i(e)^2\right).$$

Proof. In the model of 2-colored edges in the regular n-gon, every red edge e has $(n-2)/2 - i(e)$ vertices on one side and $(n-2)/2 + i(e)$ vertices on the other, and any edge joining vertices on different sides crosses e. Thus

$$\frac{1}{4}(n-2)^2 - i(e)^2 = \text{cr}(e, R) + \text{cr}(e, B).$$

Adding this equation over all red edges yields

$$\frac{1}{4}r(n-2)^2 - \sum_{e \text{ red}} i(e)^2 = \sum_{e \text{ red}} \text{cr}(e, R) + \text{cr}(R, B).$$

Because $\text{cr}(D) = \binom{n}{4} - \text{cr}(R, B)$, it follows that

$$\text{cr}(D) = \binom{n}{4} - \frac{1}{4}r(n-2)^2 + \sum_{e \text{ red}} \left(\text{cr}(e, R) + i(e)^2\right).$$

□

For the rest of the paper, we assume that D is a 2-page drawing of the complete graph on n vertices with exactly $r = |R|$ edges in one page and with $\nu_2(K_n, r)$ crossings.

We construct an upper triangular matrix $M(D)$ which corresponds to the coloring of the edges. We call this the *2-page matrix* of D. For $i < j$ an entry (i, j) (row,column) in the 2-page matrix $M(D)$ is a point with the same color as the edge ij in the drawing D. In order to find a lower bound for $\text{cr}(D)$, in the model of the 2-page matrix, it is convenient to have most of the r edges in the square submatrix with entries (a, b) with $1 \le a \le n/2$ and $n/2 + 1 \le b \le n$. The next lemma accomplishes this.

Lemma 3. *For every $1 \le j \le n/2$, let S_j be set of the edges*

$$\{(a(\text{mod} n), b(\text{mod} n)) :$$
$$j \le a \le j + n/2 - 1, j + n/2 \le b \le j + n - 1\}.$$

There is $1 \le j \le n/2$ such that

$$|S_j \cap R| \ge r\left(1 - \frac{2}{n}\sqrt{2(r-1)}\right) > r - \frac{(2r)^{3/2}}{n}.$$

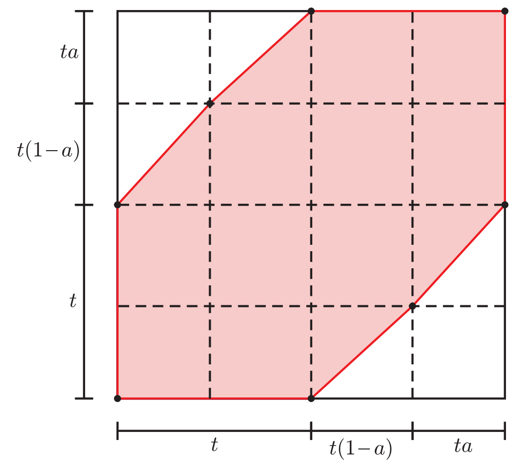
By using a suitable rotation of the indices, we can assume that the j from the previous lemma is equal to 1. Note that if $r < n^{2/3}/2$, then $|S_1 \cap R| = r$. Similarly, if $r = o(n^2)$, then $|S_1 \cap R| = r - o(r)$, so in effect S_1 contains asymptotically almost all of the red edges.

Lemma 4. *Let $s = |R \cap S_1|$. Then*

$$\sum_{e \text{ red}} \left(\text{cr}(e, R) + i(e)^2\right) \ge \frac{1}{2}s(s-1) - \frac{4\sqrt{2}}{3}(s^{3/2} - 1).$$

By Lemma 3, we can assume $s = r - o(r)$. Now the proof of the lower bound of Theorem 1 follows by using Theorem 2 and Lemma 4.

To prove the upper bound, we consider a drawing induced by a 2-page matrix $M(D)$ where the red points are those in the interior of a centrally symmetric octagon with the dimensions described in Figure 1. The octagon is also mirror-symmetric with respect to the diagonal of slope 1, and its center coincides with a point e in $M(D)$ with $i(e) = 0$. The area of the octagon is r and so $t = \sqrt{r/(4-2a)}$. The optimal value of a is the



real solution of $2a^3 - 12a^2 + 43a - 18 = 0$, which is about $a \approx 0.477$. The corresponding number of crossings of the drawing is

$$\text{cr}(D) = \binom{n}{4} - \frac{1}{4}r(n-2)^2 + h(a)r^2 + o(r^2),$$

where

$$h(a) = \frac{-2a^3 + 21a^2 - 41a + 32}{12(a-2)^2} \approx 0.6109.$$

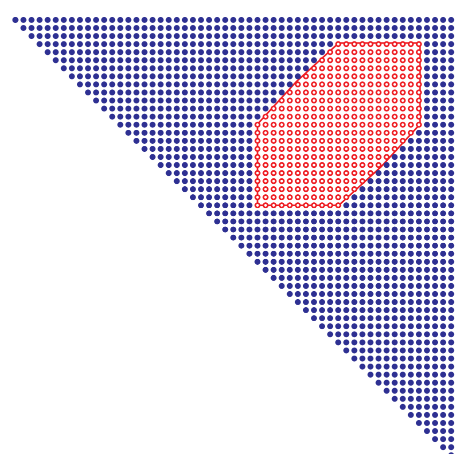


Figure 1: $M(D)$ of a drawing with few crossings

References

[1] B. M. Ábrego, O. Aichholzer, S. Fernández-Merchant, P. Ramos, and G. Salazar, The 2-page crossing number of K_n. *Discrete and Computational Geometry*, **49** (4) 747–777 (2013).

[2] L. Beineke and R. Wilson. The early history of the brick factory problem. *Math. Intelligencer*, 32:41–48, 2010.

[3] F. Harary and A. Hill. On the number of crossings in a complete graph. *Proc. Edinburgh Math. Soc.*, 13:333–338, 1963.

A characterization of tree-tree quadrangulations on closed surfaces

Raiji Mukae* & Terukazu Sano†
National Institute of Technology, Kisarazu College

A graph G is said to be *tree-tree* if there is a partition of $V(G) = V_1 \cup V_2$ such that V_i induces a tree for $i = 1, 2$. This terminology is first introduced as "tree-tree triangulation" in [2]. In that paper, the results discussed the dual form of Barnette conjecture and the vertex arboricity for planar graphs. In [1], they focus on constructing tree-tree triangulations on the plane.

A *triangulation* on a closed surface F^2 is a simple graph embedded on F^2 so that each face is triangular.

Theorem 1 (Z.Skupień [2]). *A triangulation G on a sphere is tree-tree if and only if the dual G^* is hamiltonian.*

It is easy to see that there is no tree-tree triangulations on non-spherical closed surfaces. However, there are tree-tree graphs on closed surfaces in general. In this talk, we shall focus on "tree-tree quadrangulations".

A *quadrangulation* on a closed surface F^2 is a simple graph embedded on F^2 so that each face is quadrangular. First, we obtain an analogous theorem for tree-tree triangulations on a sphere.

Theorem 2. *A quadrangulation G on a sphere is tree-tree if and only if the dual G^* is hamiltonian.*

For extending the result to other surfaces, we define some terminologies. An *open* 2*-cell* 2*-face embedding* (open 2C2F embedding) of a multigraph G into a closed surface F^2 represents an embedding with only two faces so that each face is homeomorphic to an open 2-cell. (See Figure 1.) For a closed surface F^2, *Euler genus* is defined as $\varepsilon(F^2) = 2 - \chi(F^2)$, where $\chi(F^2)$ is Euler characteristic of F^2.

We have proved that tree-tree quadrangulations can be characterized with special 4-regular multigraphs in the dual.

Theorem 3. *A quadrangulation G on a non-spherical closed surface F^2 is tree-tree if and only if the dual G^* has a spanning subgraph H^* such that H^* is homeomorphic to some $\varepsilon(F^2)$-vertex* 4*-regular connected multigraph with open* 2C2F *embedding into F^2.*

Moreover, we have determined the specific 4-regular multigraphs in Theorem 3 for a projective plane and a torus, as follows.

Theorem 4. *A quadrangulation G on a projective plane P^2 is tree-tree if and only if the dual G^* has a spanning subgraph H^* such that H^* is homeomorphic to a* 2*-bouquet with open* 2C2F *into P^2.*

*E-mail: mukae@cc.miyakonojo-nct.ac.jp
†E-mail: sano@nebula.n.kisarazu.ac.jp

Theorem 5. *A quadrangulation G on a torus is tree-tree if and only if the dual G^* has a spanning subgraph H^* of G^* such that H^* is homeomorphic to a* 4*-dipole with open* 2C2F *into T^2.*

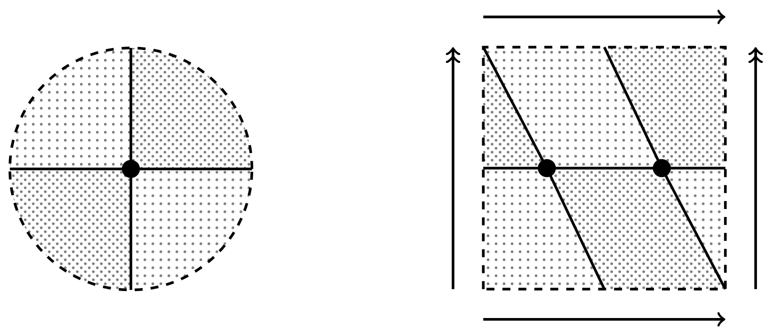


Figure 1: Open 2C2F embeddings of a 2-bouquet on P^2 and a 4-dipole on T^2

In this talk, we will introduce a difference between triangular and quadrangular cases, and show an outline of our main results.

References

[1] Günter Schaar and Zdzisł aw Skupień, Pairs of trees in tree-tree triangulations, *Discrete Math.*, 307(11-12):1499–1505, 2007.

[2] Zdzisł aw Skupień, Hamiltonicity of planar cubic multigraphs, *Discrete Math.*, 251(1-3):163–168, 2002.

On Ramsey (P_4, P_4)-minimal graphs

H. Assiyatun[1], E.T. Baskoro[2], D. Rahmadani[2], B. Rahajeng[2], Y. Yoshie[3]

[1]Combinatorial Mathematics Research Group ITB Indonesia, email :hilda@math.itb.ac.id

[2]Combinatorial Mathematics Research Group ITB Indonesia

[3]Graduate School of Information Science, Tohoku University, Japan

Abstract

For any two given graphs G and H, the notation $F \to (G, H)$ means that any red-blue coloring of all edges of F creates either a red subgraph isomorphic to G or a blue subgraph isomorphic to H. A graph F is a Ramsey $(G, H)-$minimal graph if $F \to (G, H)$ but $F - e \nrightarrow (G, H)$, for every $e \in E(F)$. The class of all Ramsey $(G, H)-$minimal graphs is denoted by $\mathcal{R}(G, H)$. It is known that $\mathcal{R}(P_m, P_n)$, for $n \geq m \geq 3$ is infinite. We show that $\mathcal{R}(P_m, P_n)$, for $n \geq m \geq 2$ contains no disconnected graphs, and $\mathcal{R}(P_m, P_n)$, for $n \geq m \geq 4$ contains no trees. We also show that there is no unicyclic graph in $\mathcal{R}(P_m, P_n)$, for $n > m \geq 4$. In particular we investigate some classes of graphs in $\mathcal{R}(P_4, P_4)$, and prove the uniqueness of unicyclic graphs in $\mathcal{R}(P_4, P_4)$.

2010 Mathematics Subject Classification: *05C55, 05D10*

Keywords: *Ramsey minimal graph, coloring, Ramsey infinite, unicyclic graph*

Unicyclic Ramsey (path,path)-minimal graphs

Desi Rahmadani*, Hilda Assiyatun, Edy Tri Baskoro

Combinatorial Mathematics Research Group
Faculty of Mathematics and Natural Sciences
Institut Teknologi Bandung Jl. Ganesa 10 Bandung 40132, Indonesia

Email: desi.rahmadani33@students.itb.ac.id, $\{$*hilda,ebaskoro*$\}$*@math.itb.ac.id*

Abstract

For any given two graphs G and H, the notation $F \to (G, H)$ means that any red-blue coloring of all the edges of F will create either a red subgraph isomorphic to G or a blue subgraph isomorphic to H. A graph F is a Ramsey (G, H)-minimal graph if $F \to (G, H)$ but $F - e \nrightarrow (G, H)$, for every $e \in E(F)$. The set of all Ramsey (G, H)-minimal graphs (up to isomorphism) is denoted by $\mathcal{R}(G, H)$. The pair (G, H) will be called Ramsey-finite or Ramsey-infinite depending upon whether $\mathcal{R}(G, H)$ is finite or infinite, respectively. Several papers have discussed the problem of determining whether for a pair (G, H) of graphs the class $\mathcal{R}(G, H)$ is finite or infinite. It is known that the set $\mathcal{R}(P_m, P_n)$, for $n \geq m \geq 3$ is Ramsey-infinite.

Some partial results for $\mathcal{R}(P_3, P_n)$, for odd $n \geq 7$ have been obtained. However, since the set $\mathcal{R}(P_3, P_n)$, for $n \geq 7$ is Ramsey-infinite then it is interesting to know some infinite families in this set. In this paper, we construct an infinite set of unicyclic graphs in $\mathcal{R}(P_3, P_n)$, for each odd $n \geq 7$. These unicyclic graphs are formed by attaching a tree in each vertex of odd cycles. This process will be done recursively.

Keywords : *edge coloring, Ramsey minimal graph, path, unicyclic graph.*

Polyhedral Characterization of Reversible Hinged Dissections

Jin Akiyama[*] Erik D. Demaine[†] Stefan Langerman[‡]

Abstract. We prove that two polygons A and B have a reversible hinged dissection (a chain hinged dissection that reverses inside and outside boundaries when folding between A and B) if and only if A and B are two non-crossing nets of a common polyhedron. Furthermore, *monotone* hinged dissections (where all hinges rotate in the same direction when changing from A to B) correspond exactly to non-crossing nets of a common convex polyhedron. By envelope/parcel magic, it becomes easy to design many hinged dissections.

1 Introduction

Given two polygons A and B of equal area, a *dissection* is a decomposition of A into pieces that can be re-assembled (by translation and rotation) to form B. In a (chain) *hinged* dissection, the pieces are hinged together at their corners to form a chain, which can fold into both A and B, while maintaining connectivity between pieces at the hinge points. Many known hinged dissections are *reversible* (originally called *Dudeney dissection* [3]), meaning that the outside boundary of A goes inside of B after the reconfiguration, while the portion of the boundaries of the dissection inside of A become the exterior boundary of B. In particular, the hinges must all be on the boundary of both A and B. Other papers [4, 2] call the pair A, B of polygons *reversible*.

Without the reversibility restriction, Abbott et al. [1] showed that any two polygons of same area have a hinged dissection. Properties of reversible pairs of polygons were studied by Akiyama et al. [3, 4]. In a recent paper [2], it was shown that reversible pairs of polygons can be generated by unfolding a polyhedron using two non-crossing nets. The purpose of this paper is to show that this characterization is in some sense complete.

An *unfolding* of a polyhedron P cuts the surface of P using a *cut tree* T,[1] spanning all vertices of P, such that the cut surface $P \setminus T$ can be unfolded into the plane without overlap by opening all dihedral angles between the (possibly cut) faces. The planar polygon that results from this unfolding is called a *net* of P. Two trees T_1 and T_2 drawn on a surface are *non-crossing* if pairs of edges of T_1 and T_2 intersect only at common endpoints and, for any vertex v of both T_1 and T_2, the edges of T_1 (respectively, T_2) incident to v are contiguous in clockwise order around v. Two nets are non-crossing if their cut trees are non-crossing.

Lemma 1. *Let T_1, T_2 be non-crossing trees drawn on a polyhedron P, each of which spans all vertices of P. Then there is a cycle C passing through all vertices of P such that C separates the edges of T_1 from edges of T_2, i.e., the (closed) interior (yellow region) of C includes all edges of T_1 and the (closed) exterior of C includes all edges of T_2.*

We can now state our first characterization.

Theorem 2. *Two polygons A and B have a reversible hinged dissection if and only if A and B are two non-crossing nets of a common polyhedron.*

Proof sketch. To prove one direction, it suffices to glue both sides of the pieces of the dissection as they are glued in both A and B to obtain a polyhedral metric homeomorphic to a sphere, and note that this metric corresponds to the surface of some polyhedron [2]. In the other direction, we use Lemma 1 to define the sequence of hinges. Now the cut tree T_B of net B is completely contained in the net A and determines the dissection. □

Often times, reversible hinged dissections are also *monotone*, meaning that the turn angles at

[*]Tokyo University of Science

[†]CSAIL, Massachusetts Institute of Technology

[‡]Directeur de Recherches du F.R.S-FNRS, Université Libre de Bruxelles

[1]For simplicity we assume that the edges of T are drawn using segments along the surface of P, and that vertices of degree 2 can be used in T to draw any polygonal path.

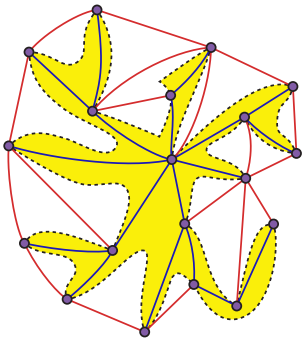


Figure 1: Example of Lemma 1. The edges of T_1, T_2 are colored blue, red, respectively.

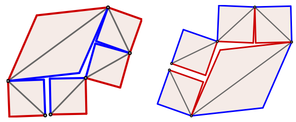


Figure 2: Reversible hinged dissection that is not monotone (or simple).

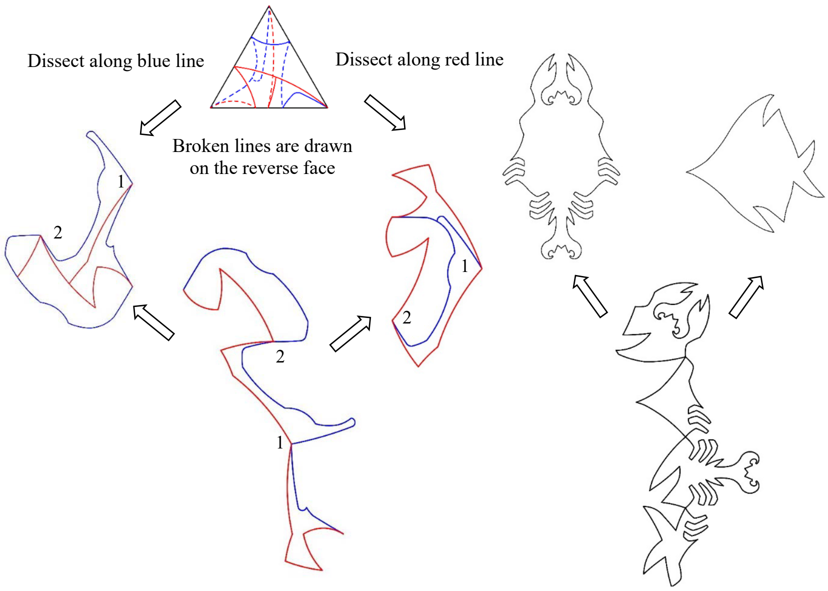


Figure 3: Two simple reversible hinged dissections found by our technique. Left: two non-crossing nets of a doubly covered triangle. Right: Lobster to fish.

all hinges in A increase to produce B. Figure 2 shows a hinged dissection that is reversible but not monotone. Monotone reversible hinged dissections also have a nice characterization:

Theorem 3. *Two polygons A and B have a monotone reversible hinged dissection if and only if A and B are two non-crossing nets of a common convex polyhedron.*

An interesting special case of a monotone reversible hinged dissection is when every hinge touches only its two adjacent pieces in both its A and B configurations, and thus A and B are only possible such configurations. We call these *simple* reversible hinged dissections. (For example, Figure 2 is not simple.)

Lemma 4. *Every simple reversible hinged dissection is monotone.*

Corollary 5. *If two polygons A and B have a simple reversible hinged dissection, then A and B are two non-crossing nets of a common convex polyhedron.*

Figure 3 shows two examples of hinged dissections resulting from these techniques. Historically, many hinged dissections (e.g., in [5]) have been designed by overlaying tessellations of the plane by shapes A and B. This connection to tiling is formalized by the results of this paper, combined with the characterization of shapes that tile the plane isohedrally as unfoldings of certain convex polyhedra [6].

References

[1] T. G. Abbott, Z. Abel, D. Charlton, E. D. Demaine, M. L. Demaine, and S. D. Kominers. Hinged dissections exist. *Discrete & Computational Geometry*, 47(1):150–186, 2012.

[2] J. Akiyama, S. Langerman, and K. Matsunaga. Reversible nets of polyhedra. In *Revised Papers from the 18th Japan Conference on Discrete and Computational Geometry and Graphs*, pages 13–23, Kyoto, Japan, September 2015.

[3] J. Akiyama and G. Nakamura. Dudeney dissection of polygons. In *Revised Papers from the Japan Conference on Discrete and Computational Geometry*, volume 1763 of *Lecture Notes in Computer Science*, pages 14–29, Tokyo, Japan, December 1998.

[4] J. Akiyama and H. Seong. An algorithm for determining whether a pair of polygons is reversible. In *Proceedings of the 3rd Joint International Conference on Frontiers in Algorithmics and Algorithmic Aspects in Information and Management*, pages 2–3, Dalian, China, June 2013.

[5] G. N. Frederickson. *Hinged Dissections: Swinging & Twisting*. Cambridge University Press, August 2002.

[6] S. Langerman and A. Winslow. A complete classification of tile-makers. In *Abstracts from the 18th Japan Conference on Discrete and Computational Geometry and Graphs*, Kyoto, Japan, September 2015.

Packing Developments of Cubes

Tatsuya Inoha*† Yoshitaka Inoue*‡ Takayuki Ozawa*§ Yushi Uno*¶

*Osaka Prefecture University, Japan

{†sxb01196,‡sxb01025,§syb01040}@edu.osakafu-u.ac.jp, ¶uno@cs.osakafu-u.ac.jp

1 Introduction

We study a problem of packing developments of cubes into a rectangle area. More specifically, given an $m \times n$ rectangle board with grid lines, we pack arbitrary developments of a unit cube aligned with the grid lines, and we are interested in the largest number of developments that can be packed in the board. Since the problem setting is quite simple and could have potential practical applications, it has attracted a lot of interest and attention since early times. However, it does not seem to have been studied systematically and very few results can be seen formally. Among those, Odawara [1] shows records on boards of sizes from 4×4 to 12×12 without proofs, and this is the best known records to the best of our knowledge.

In this research, we marshal and verify the existing results, and try to update the known records or obtain new results for the aim of promoting further systematic research on this topic. Also we pose some conjectures based on our newly obtained results.

2 Approach

There are eleven distinct developments of a cube (identifying mirror images and rotations), as shown in Fig. 1, and they are parts of 35 hexominoes.

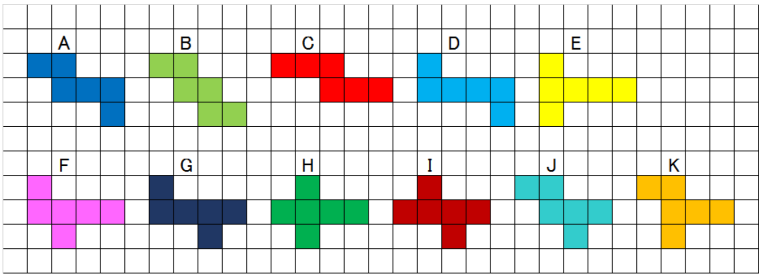


Figure 1: Eleven developments of a unit cube. We distinguish them by giving unique identifiers from A to K.

Our approach to this problem is to do exhaustive search by computer programs. However, a naive implementation of a brute-force search in our environment could only solve up to 7×7 board. We also use BurrTools [2], which is a well-known free software designed by Röver for the purpose of solving these kinds of puzzles efficiently, however, this is again not fast enough even for relatively small sizes of boards.

To accelerate our exhaustive search of using BurrTools, we incorporate the following two heuristics that we obtained by observing large number of optimal solutions for several board sizes. This is especially effective for obtaining the first feasible solution for each board size.

1. Frequently used developments first.
   By statistic analyses, A, B, C and J are the most frequently used four developments in optimal solutions, and H, I and K are not. Especially, I is rare to appear. Therefore, we try to use them preferentially.
2. Eliminate a pair of opposite corner cells from the board.
   By observation, we found that a pair of opposite corner cells are not covered by most of optimal solutions. Therefore, we set to avoid using those corner cells.

These two heuristic ideas could reduce the computational time approximately to 50%–30% to the original on the average to obtain a feasible solution.

3 Computational Results

Before we proceed, we recall the following fact.

Fact (folklore)**.** *Any rectangle cannot be exactly packed by developments of a cube.*

This implies that if the number of remaining empty cells of a solution is less than 7, it is automatically optimal. Furthermore, Odawara [1] conjectures that for boards of sizes 7×7 or larger, the number of remaining empty cells in an optimal solution is in the range of 6 to 11.

Now we present our new results as well as previously known ones in Fig. 2. We verified that all the known results are correct.

6	4								
7	6	7							
8	7	8	9						
9	8	9	11	12					
10	9	10	12	14	15				
11	10	11	13	15	17	19			
12	10	13	15	17	19	21	23		
13	12	14	16	18	20	22	25	27	
14	13	15	17	20	22	24	27	29	31
×	6	7	8	9	10	11	12	13	14

6	12								
7	6	7							
8	6	8	10						
9	6	9	6	9					
10	6	10	8	6	10				
11	6	11	10	9	8	7			
12	12	6	6	6	6	6	6		
13	6	7	8	9	10	11	6	7	
14	6	8	10	6	8	10	6	8	10
×	6	7	8	9	10	11	12	13	14

Figure 2: [Left] the optimal (largest) number of developments that are packed, and [Right] the number of remaining empty cells; yellow: newly obtained and optimal, beige: newly obtained but not verified to be optimal.

4 Discussion

By our computational experiments, we newly obtained optimal solutions for boards of sizes from 6×13 to 13×14 (with two exceptions). We show an optimal solution for 13×14 board in Fig. 3.

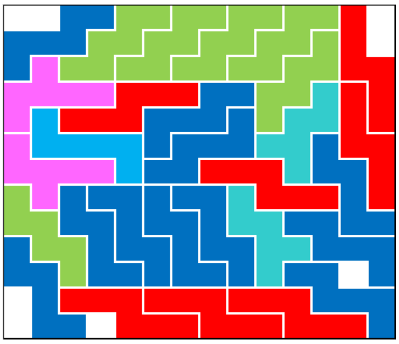


Figure 3: An optimum solution for 13×14 board. Developments of types A, B and C appear 11, 6 and 7 times, respectively; 24 developments in total out of 29 pieces.

From the results, we can verify that the conjecture by Odawara about the number of remaining empty cells holds for them, and we pose a slightly stronger conjecture.

Conjecture 1 *For boards of sizes* 7×7 *or* 6×13 *or larger, the number of remaining cells ranges from* 6 *to* 11*.*

Also by examining optimal solutions we have the following conjectures.

Conjecture 2 *There is an optimal solution for any rectangle board where one pair of opposite corner cells are not covered.*

Conjecture 3 *There is an optimal solution where none of developments H, I or K is used.*

We are trying to prove these conjectures mathematically, and we hope that they could be useful for more intelligent exhaustive search.

References

[1] Mitsuhiro Odawara. `http://www.torito.jp/puzzles/hako047.shtml` (in Japanese), 2008 (last accessed Dec. 2016).

[2] Andreas Röver. BurrTools. `http://burrtools.sourceforge.net/` (last accessed Mar. 2017).

Packing polyominoes into a rectangle is constant-time testable

ITO Hiro*† TAKEDA Yoshihiro*†
*School of Informatics and Engineering,
The University of Electro-Communications (UEC); Tokyo, Japan.
†CREST, JST; Tokyo, Japan.
*{itohiro@, t1631087@edu.cc.}uec.ac.jp

For a given shape of a polyomino S (typically, a rectangle) and a multiset of polyominoes $\mathcal{P} = \{P_1, \ldots, P_n\}$, where the sum of the area of every polyomino is equal to the area of S, a problem to determining whether all polyominoes can be packed into S, i.e., to make S without any overlap and gap by gathering $P_1, \ldots, P_n$ arbitrarily. This problem is called POLYOMINO and known to be NP-complete even if every polyominoes are restricted to rectangles with unit width[1]. If S is not restricted to a rectangle and holes are allowed, it is known to be #P-hard and ASP-Complete [2].

We consider constant-time testability of this problem. Researches on constant-time testers have been developed rapidly, mainly in this century, and many problems have been shown to be constant-time testable [1]. Even in the area of games and puzzles, it was shown that the generalized shogi, chess, and xianqi are all constant-time testable [4]. From the view point of packing problems, the knapsack problem also have been shown to be constant-time testable [3].

For discussing constant-time testability of POLYOMINO, we first give how to represent an instance, definitions of distance between instances, and what kind of oracles can be used. For treating in constant-time, we introduce an assumption that the size of every polyominoes in $\mathcal{P}$ is bounded by a constant k (otherwise, it becomes hard to be treated in constant-time). Based on them, we give a constant-time tester for this puzzle. Thus we prove the following theorem.

Theorem 1 POLYOMINO *is constant-time testable.*

References

[1] O. Goldreich (Ed.): Property Testing — Current Research and Surveys, LNSC 6390, 2010.

[2] T. Horiyama, T. Ito, K. Nakatsuka, A. Suzuki and R. Uehara: Packing Trominoes is NP-Complete, #P-hard and ASP-Complete, The 24th Canadian Conference on Computational Geometry (CCCG 2012), pp. 219–224, 2012/8/8-10, PEI, Canada.

[3] H. Ito, S. Kiyoshima, Y. Yoshida, Constant-time approximation algorithms for the knapsack problem, Proc. of The 9th Annual Conference on Theory and Applications of Models of Computation (TAMC 2012), LNCS, Springer, # 7287, pp. 131–142, 2012.

[4] H. Ito, A. Nagao, and T. Park, Generalized shogi and chess are constant-time testable, Proc. of the 12th International Symposium on Operations Research & Its Applications (ISORA 2015), IET Digital Library, pp. 1–6, 2015.

[1]The proof is easy: 3-partition can be reduced to POLIOMINO.

Spherical embeddings of symmetric association schemes in 3-dimensional Euclidean space

Eiichi Bannai and Da Zhao

1 Extended Abstract

Let $\mathfrak{X} = (X, \{R_i\}_{0 \le i \le d})$ be a symmetric association schemes, and let A_i be the adjacency matrix of the relation R_i and let E_i $(0 \le i \le d)$ be the primitive idempotents. The spherical embedding of a symmetric association scheme $\mathfrak{X}$ with respect to E_i is the mapping: $X \to \mathbb{R}^{m_i}$ defined by

$$x \to \overline{x} = \sqrt{\frac{|X|}{m_i}} E_i \phi_x,$$

where ϕ_x is the characteristic vector of x (regarded as a column vector of size $|X|$) and $m_i = \text{rank} E_i$. Then the $\overline{x}$ are all in the unit sphere $S^{m_i - 1} \subset \mathbb{R}^{m_i}$. In what follows, we identify $\overline{X}$ and X when the embedding is faithful. The reader is referred to [3,4,2,1, etc.] for the basic concept of association schemes and spherical embeddings of association schemes.

In [1], Bannai-Bannai studied the spherical embeddings of symmetric association schemes with $m_1 = 3$, i.e., in $\mathbb{R}^3$. and determined that there exist only one such faithful spherical embedding if we assume the association scheme is primitive. Namely, it must be a regular tetrahedron, i.e., the association scheme with $d = 1$ corresponding to the complete graph K_4. On the other hand, in [1] it was known that the method used there could be applied to study imprinitive association schemes as well, but it was left unanswered. The proof in [1] is of completely elementary geometric nature, and it has close connections with the classification of regular polyhedrons and quasi-regular polyhedrons, etc.. We first remark that the method in [1] essentially proves the following result.

Proposition 1. Let X be a faithful spherical embedding of $\mathfrak{X}$. Let $A(X) = \{\langle x, y \rangle \mid x, y \in X, x \neq y\}$ and let $\alpha = \max\{\langle x, y \rangle \mid x, y \in X, x \neq y\}$, where $\langle x, y \rangle$ is the usual inner product on $\mathbb{R}^n$. Suppose $m_1 = 3$ and that the spherical embedding of X is faithful (i.e., $1 \notin A(X)$), and moreover, we assume that the relation $R_\alpha = \{(x, y) \mid \langle x, y \rangle = \alpha\}$ makes a single relation R_j of the association scheme for some j. Then
(1) $k_1 = 3, 4$, or 5 where k_1 is the valency of the graph (X, R_j) corresponding to the j (corresponding to the inner product α). (We regard this old R_j as our new R_1.)
(2) For each $k_1 = 3, 4, 5$, we can show that $X \subset S^2$ is as follows.
(a) If $k_1 = 5$, then X is the regular icosahedron ($|X| = 12$).
(b) If $k_1 = 4$, then X is the regular octahedron ($|X| = 6$), the quasi-regular polyhedron of type $[3, 4, 3, 4]$ ($|X| = 12$), or the quasi-regular polyhedron of type $[3, 5, 3, 5]$ ($|X| = 30$).
(c) If $k_1 = 3$, then X is the regular tetrahedron ($|X| = 4$), the cube ($|X| = 8$), or the regular dodecahedron ($|X| = 20$).

Corollary 2. Let $\mathfrak{X}$ be a Q-polynomial association scheme spherically embedded in $\mathbb{R}^3$, then such $\mathfrak{X}$ are classified. (They are in a part of those in the list of Theorem 1 above.)
This is immediately obtained from Theorem 1, since if $\mathfrak{X}$ is a Q-polynomial association scheme, then the $Q_1(i)$ $(0 \le i \le d)$ are all distinct, and so the assumption of Theorem 1 is satisfied.
(We believe that the result obtained in Corollary 2 should be expected to be known, but it seems that

School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai 200240, China.
E-mail addresses: bannai@sjtu.edu.cn, bannai@math.kyushu-u.ac.jp (E. Bannai), jasonzd@sjtu.edu.cn (Da Zhao).

this was not explicitly mentioned in the literature, as far as we could check.)

So, the difficulty of treating imprimitive association schemes (with $m_1 = 3$) is in dealing with the cases (1) $k_1 = 1$ and $k_1 = 2$, and (2) the cases where R_α is split into more than one relations.

To our great joy, we were able to deal with these two difficulties completely. Our main theorem is stated as follows.

Theorem 3. Let $\mathfrak{X}$ be a symmetric association scheme. If $\mathfrak{X}$ has a faithful spherical embedding with $m_1 = 3$ in $\mathbb{R}^3$, then it must be one of those listed in Proposition 1.

The proof is done, first assuming either $k_1 = 1$ or $k_1 = 2$, then $\mathfrak{X}$ cannot have a faithful spherical embedding with $m_1 = 3$ in $\mathbb{R}^3$. Then that technique can be generalized to deal with the case where R_α is split into more than one relations, because if R_α is split, then some relation with valency 1 or 2 must appear, and then a similar technique as in the previous step can be used. The proof is ingenious, but not long, and can be clearly understood.

Concluding Remarks.
(1) This paper would be interesting as an interplay of the theory of association schemes and the elementary geometric considerations in discrete geometry. Association schemes can be a more standard tool to study good geometric structures such as regular polyhedron, quasi-regular polyhedrons, as well as similar or more general objects in higher dimensions.
(2) There are some considerable differences between determining all symmetric association schemes with $m_1 = 3$ and determining all faithful spherical embeddings with $m_1 = 3$ of symmetric association schemes. For example, van Dam, Koolen, Park [5, Section 2.5, page 6] describes the difficulty of the former problem. On the other hand, from the geometric point, the most crucial problem would be the latter one that is answered in this paper.
(3) It would be very interesting to study spherical embeddings of symmetric association schemes with $m_1 = 4$, say. In particular, it would be interesting to try to classify (primitive) Q-polynomial association schemes which are spherically embedded with $m_1 = 4$. We hope that our method for $m_1 = 3$ is somehow useful for that. On the other hand, the complete classification of faithful spherical embedding of symmetric association schemes with $m_1 = 4$ seems to be still somehow distant, as there are infinitely many such examples.

References

[1] Ei. Bannai and Et. Bannai, On primitive symmetric association schemes with $m_1 = 3$, Contrib. Discrete Math., 1 (2006), 68–79.

[2] Ei. Bannai, Et. Bannai, and T. Ito, Introduction to Algebraic Combinatorics, Kyoritsu Shuppan, Tokyo, (in Japanese), 2016.

[3] Ei. Bannai and T. Ito, Algebraic Combinatorics I: Association Schemes, Benjamin/Cummings, Menlo Park, California, (1984).

[4] A. E. Brouwer, A. M. Cohen and A. Neumaier, *Distance-regular graphs*, Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)], 18. Springer-Verlag, Berlin (1989).

[5] E. R. van Dam, J. H. Koolen, J. Park, Partially metric association schemes with a multiplicity three, arXiv:1701.03193.

CONSTRUCTION OF SPHERICAL t-DESIGNS FROM BALL DESIGNS

M. SAMY BALADRAM

A *spherical t-design* on the sphere S^n is a finite set of points $X \subseteq S^n$ for which

$$\frac{1}{\sigma(S^n)} \int_{S^n} f(x) \mathrm{d}\sigma(x) = \frac{1}{|X|} \sum_{x \in X} f(x)$$

holds for all polynomials $f(x) = f(x_0, x_1, x_2, \ldots, x_n)$ of degree at most t, where σ denotes the surface measure on S^n.

A general and explicit construction was first given by Rabau and Bajnok in 1992 [2], however no further generalization of this has been proven yet. We try to generalize their result by defining the following:

Let $B^d = \{(a_1, \ldots, a_d) \in \mathbb{R}^d \mid \sum_{i=1}^d a_i^2 \leq 1\}$. A finite subset V of B^d is called a *ball t-design* with weight function $w : B^d \to \mathbb{R}$ if

$$\frac{1}{\int_{B^d} w(x) \mathrm{d}x} \int_{B^d} f(x) w(x) \mathrm{d}x = \frac{1}{|V|} \sum_{v \in V} f(v) \tag{1}$$

holds for all polynomials $f(x) = f(x_1, x_2, \ldots, x_d)$ of degree at most t.

For positive integers n, d, with $n > d$, we can express an integral over S^n as a double integral over S^{n-d} and B^d as follows

$$\int_{S^n} f(x) \mathrm{d}\sigma(x) = \int_{B^d} \int_{S^{n-d}} f(\sqrt{1 - \|v\|^2} y, v)(1 - \|v\|^2)^{\frac{n-d-1}{2}} \mathrm{d}\sigma(y) \mathrm{d}v \tag{2}$$

with $v = (v_1, v_2, \ldots, v_d)$, $\|v\| = \sqrt{v_1^2 + \cdots + v_d^2}$, $\mathrm{d}v = \mathrm{d}v_1 \mathrm{d}v_2 \cdots \mathrm{d}v_d$. By a similar way of proving Rabau-Bajnok's theorem, this implies the following:

Theorem 1. *Let n, d be positive integers, $n > d$, $Y \subseteq S^{n-d}$ be a spherical t-design, and let $V \subseteq B^d$ be a ball t-design in $\mathbb{R}^d$ with respect to the weight function $w_{n-d-1}(x) = (1 - \|x\|^2)^{\frac{n-d-1}{2}}$. Then*

$$X = \{(\sqrt{1 - \|v\|^2} y, v) \mid y \in Y, v \in V\}$$

is a spherical t-design on S^n.

We will use this theorem by first constructing some ball t-design in $\mathbb{R}^d$. The following lemma is from [1]:

Lemma 2. *Let d be a positive integer, and*

$$F(x) = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha_d} \quad (\alpha_1, \ldots, \alpha_d \in \mathbb{Z}_{\geq 0}).$$

Then,

$$\text{(i)} \int_{S^{d-1}} F(x) \mathrm{d}\sigma(x) = \frac{\prod_{i=1}^{d} \frac{(1+(-1)^{\alpha_i})}{2} \cdot \Gamma(\frac{\alpha_i+1}{2})}{\frac{1}{2}\Gamma(\frac{1}{2}(\sum_{i=1}^{d} \alpha_i + d))}, \qquad \text{(ii)} \int_{B^d} F(x)\, \mathrm{d}x = \frac{\int_{S^{d-1}} F(x)\, \mathrm{d}\sigma(x)}{\sum_{i=1}^{d} \alpha_i + d}.$$

By this, we proved the following:

Theorem 3. *Let t, d be positive integers and m be a non-negative integer. Also, let R be a finite subset of the interval $(0, 1]$, $Z_r \subset S^{d-1}$ be a spherical $(2t + 1)$-design for all $r \in R$. Then*

$V := \bigcup_{r\in R} rZ_r \subset B^d$ is a ball $(2t+1)$-design with weight function $w_m(x) = (1-\|x\|^2)^{\frac{m}{2}}$ if and only if the following equation holds for $1 \le k \le t$:

$$\frac{1}{\sum\limits_{r\in R}|Z_r|}\sum_{r\in R}|Z_r|r^{2k} = \frac{\Gamma(k+\frac{d}{2})\Gamma(\frac{m+d}{2}+1)}{\Gamma(\frac{d}{2})\Gamma(k+\frac{m+d}{2}+1)}. \tag{3}$$

Now, we can incorporate this result with Theorem 1 as follows.

Theorem 4. *Let n, d be positive integers with $n > d \ge 2$, and R be a finite subset of the interval $(0,1]$. Also, let $Y \subseteq S^{n-d}$, $Z_r \subseteq S^{d-1}$ with $r \in R$, each be a spherical $(2t+1)$-design in their respective dimension. If*

$$\frac{1}{\sum\limits_{r\in R}|Z_r|}\sum_{r\in R}|Z_r|r^{2k} = \frac{\Gamma(\frac{n}{2}+k)\Gamma(n+\frac{1-k}{2})}{\Gamma(\frac{n}{2})\Gamma(n+\frac{1+k}{2})} \tag{4}$$

holds for $1 \le k \le t$ then $X = \{(\sqrt{1-r^2}y, rz) \mid y \in Y, r \in R, z \in Z_r\}$ is a spherical $(2t+1)$-design on S^n.

We wish to construct some ball designs in $\mathbb{R}^2$ using a union of polygons, which is a spherical design in S^1. The regular n-gon is defined as $P_n = \{(\cos(\frac{2\pi i}{n}), \sin(\frac{2\pi i}{n})) \mid 1 \le i \le n\}$. In Theorem 3, setting $t = 2$, $d = 2$, $R = \{r_1, r_2\}$, and Z_r to be P_{n_i} for some integers n_i with $n_i \ge 2t+2$, $i = \{1, 2\}$, we have the following theorem.

Lemma 5. *For non negative integers m and $n_1, n_2 \ge 6$ with*

$$\frac{m+2}{m+6} < \frac{n_1}{n_2} \le \frac{1}{4}(m+2)(m+6),$$

the set $V = r_1P_{n_1} \cup r_2P_{n_2} \subset B^2$ with

$$r_1 = \sqrt{\frac{2(n_1\sqrt{m+6} - \sqrt{(m+2)n_1n_2})}{n_1(m+4)\sqrt{m+6}}}, \quad r_2 = \sqrt{\frac{2(n_2\sqrt{m+6} + \sqrt{(m+2)n_1n_2})}{n_2(m+4)\sqrt{m+6}}}$$

is a disk 5-design with weight function $w(x) = (1-\|x\|^2)^{\frac{m}{2}}$.

It is well known that a regular 6-gon and 12 points of an icosahedron are spherical 5-designs in S^1 and S^2, respectively. By this fact, Theorem 1 can be used to produce the following spherical designs.

Theorem 6. *Let m be a non negative integer and $V_m = r_+(m)P_6 \cup r_-(m)P_6$ with*

$$r_\pm(m) = \sqrt{\frac{2((m+6) \pm \sqrt{(m+2)(m+6)})}{(m+4)(m+6)}}.$$

Also, let Y_1 be the 6-gon, Y_2 be the 12 points of an icosahedron, and define Y_n recursively as

$$Y_n = \{(\sqrt{1-\|v\|^2}y, v) \mid y \in Y_{n-2}, v \in V_{n-3}\}, \quad n \ge 3.$$

Then, Y_n is a spherical 5-design in S^n.

Computing the size of Y_n in this theorem, we get $|Y_n| = 6 \cdot 12^{\frac{n-1}{2}}$ if n is odd, and $|Y_n| = 12^{\frac{n}{2}}$ if n is even.

References

[1] G. B. Folland, How to Integrate a Polynomial over a Sphere, The American Mathematical Monthly Vol. 108, No. 5 (May, 2001), pp. 446-448

[2] P. Rabau and B. Bajnok, Bounds for the number of nodes in Chebyshev type quadrature formulas, J. Approximation Theory **67** (1991), 199–214.

Division of Mathematics, Graduate School of Information Science, Tohoku University
E-mail address: samy@ims.is.tohoku.ac.jp

The Orbits of Folded Crossed Cubes*

Jia-Jie Liu[1,†]

[1] Department of Information Management,
Shih Hsin University, Taipei, Taiwan

Abstract

An *automorphism* of a graph $G = (V, E)$ is a mapping ϕ: $V(G) \to V(G)$ such that there is an edge $uv \in E(G)$ if and only if $\phi(u)\phi(v)$ is also an edge in $E(G)$. A graph is *vertex-transitive* if, for any two vertices u and v of G, there is an automorphism ϕ such that $\phi(u) = v$. Clearly, every vertex-transitive graph is regular, e.g., hypercubes. However, not all regular graphs are vertex-transitive, e.g., the crossed cubes [1, 2]. Two vertices u and v in a graph $G = (V, E)$ are in the same orbit if there exists an automorphism ϕ of G such that $\phi(u) = v$. The orbit number of a graph G, denoted by $Orb(G)$, is the smallest number of orbits, which form a partition of $V(G)$, in G. All vertex-transitive graphs G are with $Orb(G) = 1$. Since the n-dimensional hypercube (n-cube for short), denoted by Q_n, is vertex-transitive, it follows that $Orb(Q_n) = 1$ for $n \geq 1$. The crossed cubes which are a variant of hypercubes. In [3], Kulasinghe and Bettayeb showed that $Orb(CQ_n) > 1$ when $n \geq 5$. Shiau, Wang, and Pai showed that $Orb(CQ_n) = 2^{\lceil \frac{n}{2} \rceil - 2}$ for $n \geq 3$ in [4], where CQ_n are the n-dimensional crossed cubes.

We investigate the orbit number of the folded crossed cubes. The folded crossed cubes which are a variation of crossed cubes. The n-dimensional folded crossed cube, denoted FCQ_n, is constructed from CQ_n by adding a set of complement edges. In [5], K. J. Pai, J. M. Chang, and J. S. Yang proved that FCQ_n is vertex-transitive if and only if $n \in \{1, 2, 4\}$, namely, $Orb(FCQ_1) = Orb(FCQ_2) = Orb(FCQ_4) = 1$. In this paper, we prove that $Orb(FCQ_n) = 2^{\lceil \frac{n}{2} \rceil - 1}$ if n is odd and $Orb(FCQ_n) = 2^{\lceil \frac{n}{2} \rceil - 2}$ if n is even, for $n \geq 3$.

Keywords: Crossed cubes, Folded crossed cubes, Automorphism, Vertex-transitive, Orbits.

*This work was supported in part by the Ministry of Science and Technology of Republic of China under contracts MOST 105-2221-E-128-004-.

†All correspondence should be addressed to Professor Jia-Jie Liu, Department of Information Management, Shih Hsin University, 1 Lane 17 Sec.1, Mu-Cha Rd., Taipei, Taiwan 10607, Republic of China (e-mail: jjliu@mail.shu.edu.tw).

References

[1] K. Efe, A variation on the hypercube with lower diameter, *IEEE Transactions on Computers* 40 (1991) 1312–1316.

[2] K. Efe, The crossed cube architecture for parallel computation, *IEEE Transactions on Parallel and Distributed Systems* 3 (1992) 513–524.

[3] P. Kulasinghe and S. Bettayeb, Multiply-twisted hypercube with five or more dimensions is not vertex-transitive, *Information Processing Letters* 53 (1995) 33–36.

[4] T. H. Shiau, Y. Li Wang, and K. J. Pai, The orbits of Crossed Cubes, manuscript.

[5] K. J. Pai, J. M. Chang, J. S. Yang, Vertex-transitivity on folded crossed cubes, *Information Processing Letters* 116 (2016) 689–693.

Some Stabbing Problems of Line Segments Solved with Linear Programming

Gerardo González Martínez * Mazay O. Jiménez-Salinas* Carlos Seara†
Jorge Urrutia ‡

1 Introduction

In this paper we introduce a class of stabbing problems that can be solved using linear programming in $O(n)$ time. We start addressing the following:

Problem 1. Let $P = \{p_1, \ldots, p_n\}$ be a set of n points in the plane. Suppose that the elements of P start moving vertically at time $t = 0$ and at the same speed v. As p_i moves up, at time t the point p_i has traversed a line segment l_t^i of length $t \cdot v$, starting at p_i, let us denote as $p_i(t) = p_i + t \cdot v$. Our problem is to find the smallest t such that there exist a line ℓ that stabs $l_t^1, l_t^2, \cdots, l_t^n$, see Figure 1a. We prove that this problem can be solved in $O(n)$ time.

We also address the following variations to our problem:

Problem 2. Each point p_i moves vertically at its own speed v_i.

Problem 3. Each point p_i moves at its own direction s_i and at its own speed v_i.

Problem 4. Same problems as above for $p_i \in \mathbb{R}^d$ where d is fixed.

We will show that all of the above problems can be solved using linear programming in $2d - 1$, and thus can be solved in $f(d) \times n$ time, which is linear time for constant d.

*Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, Ciudad de México, México, {gmg_17, m.jimenez}@uxmcc2.iimas.unam.mx.

†Matemàtiques, Universitat Politècnica de Catalunya, Spain, carlos.seara@upc.edu. Supported by projects Gen. Cat. DGR 2014SGR46, MINECO MTM2015-63791-R and H2020-MSCA-RISE project 734922 - CONNECT.

‡Instituto de Matemáticas, Universidad Nacional Autónoma de México, Ciudad de México, México, urrutia@matem.unam.mx.

In Section 2, we define some concepts of linear programming and point to line transformations. In Section 3, we demonstrate that all the described problems can be solved in $O(n)$ time, when d is fixed.

2 Preliminaries

The problem of geometric separability of two sets of points R and B in $\mathbb{R}^d$ is to decide if there is a hyperplane that leaves all of the elements of B in one of the open semiplanes determined by the hyperplane, and all of the elements of R in the other. It is well known that a linear programming problem with d dimension and n variables can be solved in $O(n)$ time when d is fixed [2].

The dual of a point $p = (a, b)$ of the plane, denoted by ℓ_p, is the non-vertical line with equation $y = ax + b$. The dual of ℓ_p is p. Recall that in the dual plane the *lower envelope* is the boundary of the intersection of the halfplanes lying below the lines. Similarly, the *upper envelope* is formed by considering the intersection of the halfplanes lying above the lines.

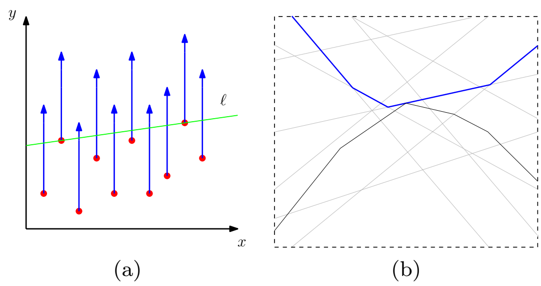


Figure 1: *a*) Set of n points in the plane moving vertically at the same speed. *b*) Dual plane showing the intersection of the upper and lower envelopes.

3 Stabbing line segments

In this section we describe our algorithm to obtain the smallest time t, if it exists, such that at time t there is a line ℓ that stabs all of the line segments l_t^i. Let P_r be the set of red points (resp. lines) containing $P = \{p_1, \ldots, p_n\}$, and $P_b(t) = \{p_1(t), \ldots, p_n(t)\}$. A transformation to the dual plane considering the time is given as follows: every point $p_i = (a_i, b_i + t)$ is mapped to the line $y = a_i x + b_i + t$. The elements of P_r are mapped to the lines $\mathcal{L}_r = \{a_i x + b_i \mid i = 1, \ldots, n\}$. Similarly, the elements of P_b are mapped to the lines $\mathcal{L}_b = \{y = a_i x + b_i + t \mid i = 1, \ldots, n\}$. We note that while points start moving in the primal plane their corresponding lines in the dual plane move upward. After sometime if a feasible region exists the upper envelope of $\mathcal{L}_b$ will intersect the lower envelope of $\mathcal{L}_r$ and that point would be the solution, see Figure 1b. $\mathcal{L}_r$ and $\mathcal{L}_b$ represent the below and above constraints, respectively. So $\mathcal{L}_r$ can be represented as $a_i x - y + b_i + t \leq 0$ and $\mathcal{L}_b$ can be represented as $a_i x - y + b_i \geq 0$. Finally our problem can be stated as a linear programming problem in $\mathbb{R}^3$ as follows:

$$\begin{aligned} &\text{minimize } t \\ &\text{subject to } a_i x - y + b_i \geq 0 \\ &\qquad\qquad a_i x - y + b_i + t \leq 0 \end{aligned}$$

Thus using Meggido's partition algorithm [1], the linear programming problem is solved in $O(n)$ time.

Thus we have the following result:

Theorem 1. *The smallest time t such that a line ℓ stabs the line segments l_t^i can be calculated in $O(n)$ time.*

Let us consider now Problem 2. The set of lines $\mathcal{L}_r$ does not change and their upper envelope remains the same, however now the set of lines $\mathcal{L}_b$ move upwards at different speeds and the lower envelope changes over the time. To solve these new constraints we asociate the speeds as follows: for every $p_i = (a_i, b_i + v_i \cdot t)$, the line $y = a_i x + b_i + v_i \cdot t$ is mapped. Then $\mathcal{L}_b = \{y = a_i x + b_i + v_i \cdot t \mid i = 1, \ldots, n\}$. Problem 2 can be stated as the following linear programming problem in $\mathbb{R}^3$:

$$\begin{aligned} &\text{minimize } t \\ &\text{subject to } a_i x - y + b_i \geq 0 \\ &\qquad\qquad a_i x - y + b_i + v_i \cdot t \leq 0 \end{aligned}$$

For Problem 3. The set of lines $\mathcal{L}_r$ remains the same, for the case of $\mathcal{L}_b$ we associate the *inclination* of every point as follows: every $p_i = (a_i + s_i, b_i + v_i \cdot t)$ is mapped to the line $y = (a_i + s_i)x + b_i + v_i \cdot t$ then $\mathcal{L}_b = \{y = (a_i + s_i)x + b_i + v_i \cdot t \mid i = 1, \ldots, n\}$. Finally Problem 3 can be stated as the following linear programming problem in $\mathbb{R}^3$:

$$\begin{aligned} &\text{minimize } t \\ &\text{subject to } a_i x - y + b_i \geq 0 \\ &\qquad\qquad (a_i + s_i)x - y + b_i + v_i \cdot t \leq 0 \end{aligned}$$

Consider the Problem 4 for points in $\mathbb{R}^3$. The points move vertically at different speeds, now the transformation to the dual space is defined as follows: every point $p_i = (a, b, c + v_i \cdot t)$ is mapped to the plane $z = a_i x + b_i y + c_i + v_i \cdot t$. The below constraints are defined as $a_i x + b_i y - z + c_i \geq 0$ while the above constraints $a_i x + b_i y - z + c_i + v_i \cdot t \leq 0$. Problem 4 can be defined as the following linear programming problem in $\mathbb{R}^4$:

$$\begin{aligned} &\text{minimize } t \\ &\text{subject to } a_i x + b_i y - z + c_i \geq 0 \\ &\qquad\qquad a_i x + b_i y - z + c_i + v_i \cdot t \leq 0 \end{aligned}$$

The d-dimensional case can be solved in linear time, for lake of space we do not give more details but we enunciate the following theorem.

Theorem 2. *For any fixed dimension d, Problems 2, 3, and 4 can be solved in $O(n)$ time.*

References

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

[2] Nimrod Megiddo. Linear programming in linear time when the dimension is fixed. *Journal of the ACM*, 31(1):114–127, 1984.

Routing in Polygonal Domains*

Bahareh Banyassady† Matias Korman‡ Wolfgang Mulzer† André van Renssen§, ¶
Marcel Roeloffzen§, ¶ Paul Seiferth† Yannik Stein† Birgit Vogtenhuber‖ Max Willert†

1 Introduction

Routing is a crucial problem in distributed graph algorithms [1, 2]. We would like to preprocess a given graph G in order to support the following task: given a data packet that lies at some *source* node p of G, route the packet to a given *target* node q in G that is identified by its *label*. We expect three properties from our routing scheme: first, it should be *local*, i.e., in order to determine the next step for the packet, it should use only information stored with the current node of G or with the packet itself. Second, the routing scheme should be *efficient*, meaning that the packet should not travel much more than the shortest path distance between p and q. Third, it should be *compact*: the total space requirement should be as small as possible.

There is an obvious solution: for each node v of G, we store at v the complete shortest path tree for v. Thus, given the label of a target node w, we can send the packet for one more step along the shortest path from v to w. Then, the routing scheme will have perfect efficiency, sending each packet along a shortest path. However, this method requires that each node stores the entire topology of G, making it not compact. Thus, the challenge lies in finding the right balance between the conflicting goals of compactness and efficiency.

Thorup and Zwick introduced the notion of a *distance oracle* [4]. Given a graph G, the goal is to construct a compact data structure to quickly answer *distance queries* for any two nodes in G. A routing scheme gives a distributed implementation of a distance oracle [3].

The problem of constructing a compact routing scheme for a general graph has been studied for a long time. Recently, Roditty and Tov [3] developed a routing scheme for a general graph G with n vertices and m edges. Their scheme needs to store a polylogarithmic number of bits with the packet, and it routes a message from s to t on a path with length $\mathcal{O}(k\Delta + m^{1/k})$, where Δ is the shortest path distance between s and t and $k > 2$ is any fixed integer. The routing tables use $mn^{\mathcal{O}(1/\sqrt{\log n})}$ total space. In general graphs, any efficient routing scheme needs to store $\Omega(n^c)$ bits per node, for some constant $c > 0$ [2]. Thus, it is natural to ask whether there are better algorithms for specialized graph classes.

Here, we consider the class of visibility graphs of a polygonal domain P with h holes and n vertices. Two vertices p and q in P are connected by an edge if and only if they can *see* each other, i.e., if and only if the line segment between p and q is contained in the (closed) region P. The problem of computing a shortest path between two vertices in a polygonal domain has been well-studied in computational geometry. Nevertheless, to the best of our knowledge, prior to our work there have been no routing schemes for visibility graphs of polygonal domains that fall into our model. For any $\varepsilon > 0$, our routing scheme needs $\mathcal{O}((\varepsilon^{-1} + h)\log n)$ bits in each routing table, and for any two vertices s and t, it produces a routing path within a factor of $1 + \varepsilon$ of the optimal which constitutes a dramatic improvement over the traditional geometric routing approach. Thus, we believe that it makes sense to look for compact routing schemes for geometrically defined graphs.

2 Cones in Polygonal Domains

Let P be a polygonal domain with n vertices and h holes. Furthermore, let $t > 2$ be a parameter, to be determined later. Following Yao [5], we subdivide the visibility polygon of each vertex in P into t *cones* with a small enough apex angle. This will allow us to achieve small stretch and compact routing tables.

Let p be a vertex in P and p' the clockwise neighbor of p if p is on the outer boundary, or the counterclockwise neighbor of p if p lies on a hole boundary. We denote with $\mathbf{r}$ the *ray* from p through p'. To obtain our cones, we rotate $\mathbf{r}$ by certain angles. Let α be the inner angle at p. For $j = 0, \dots, t$, we write $r_j(p)$ for the ray $\mathbf{r}$ rotated clockwise by angle $j \cdot \alpha/t$.

Now, for $j = 1, \dots, t$, the cone $C_j(p)$ has apex p, boundary $r_{j-1}(p) \cup r_j(p)$, and opening angle α/t; see Figure 1. For technical reasons, we define $r_j(p)$ not to be part of $C_j(p)$, for $0 \leq j < t$, whereas we

*AvR and MR were supported by JST ERATO Grant Number JPMJER1305, Japan. BB was supported in part by DFG project MU/3501-2. MK was supported in part by KAKENHI Nos. 12H00855 and 17K12635, Japan. PS was supported in part by DFG project MU/3501-1. YS was supported by the DFG within the research training group 'Methods for Discrete Structures' (GRK 1408) and by GIF grant 1161.

†Institut für Informatik, Freie Universität Berlin, Germany, {bahareh,mulzer,pseiferth,yannikstein,willerma}@inf.fu-berlin.de

‡Tohoku University, Sendai, Japan, mati@dais.is.tohoku.ac.jp

§National Institute of Informatics, Tokyo, Japan, {andre, marcel}@nii.ac.jp

¶JST, ERATO, Kawarabayashi Large Graph Project

‖Institute of Software Technology, Graz University of Technology, Graz, Austria, bvogt@ist.tugraz.at

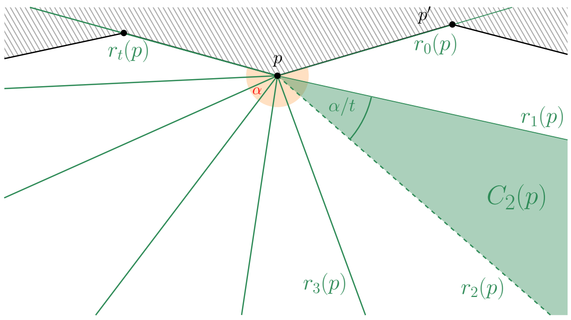


Figure 1: The cones and rays of a vertex p with apex angle α.

consider $r_t(p)$ to be part of $C_t(p)$. Furthermore, we write $\mathcal{C}(p) = \{C_j(p) \mid 1 \leq j \leq t\}$ for the set of all cones with apex p. Since the opening angle of each cone is $\alpha/t < 2\pi/t$ and since $t > 2$, each cone is convex.

3 The Routing Scheme

Let $\varepsilon > 0$, let P be a polygonal domain with n vertices and h holes, and let $\mathrm{VG}(P)$ denote the visibility graph of P. We describe a routing scheme for $\mathrm{VG}(P)$ with stretch factor $1+\varepsilon$. The idea is to compute for each vertex p the corresponding set of cones $\mathcal{C}(p)$ and to store a certain interval of indices for each cone $C_j(p)$ in the routing table of p. If an interval of a cone $C_j(p)$ contains the target vertex t, we proceed to the nearest neighbor of p in $C_j(p)$.

In the preprocessing phase, we first compute the label of each vertex $p_{i,k}$. The label of $p_{i,k}$ is the binary representation of i, concatenated with the binary representation of k, that is, $\ell(p_{i,k}) = (i, k)$. Thus, all labels are distinct binary strings of length $\lceil \log h \rceil + \lceil \log n \rceil$.

Let p be a vertex in P. The routing table of p is constructed as follows: first, we compute a shortest path tree T for p. For a vertex s of P, let T_s be the subtree of T with root s, and denote the set of all vertices on the i-th hole in T_s by $I_s(i)$. The following well-known observation lies at the heart of our routing scheme.

Observation 1 *Let q_1 and q_2 be two vertices of P. Let π_1 be the shortest path in T from p to q_1, and π_2 the shortest path in T from p to q_2. Let l be the lowest common ancestor of q_1 and q_2 in T. Then, π_1 and π_2 do not cross or touch in a point x with $d(p, x) > d(p, l)$.*

Lemma 2 *Let $e = (p, s)$ be an edge in T. Then, the indices of the vertices in $I_s(i)$ form an interval. Furthermore, let $f = (p, s')$ be another edge in T, such that e and f are consecutive in the cyclic order around p in T. Then, the indices of the vertices in $I_s(i) \cup I_{s'}(i)$ are again an interval.*

Lemma 2 indicates how to construct the routing table $\rho(p)$ for p. We set

$$t = \pi / \arcsin\left(\frac{1}{2\,(1+\varepsilon^{-1})}\right), \qquad (1)$$

and we construct a set $\mathcal{C}(p)$ of cones for p as in Section 2. Let $C_j(p) \in \mathcal{C}(p)$ be a cone, and let Π_i be a hole boundary or the outer boundary. We define $C_j(p) \sqcap \Pi_i$ as the set of all vertices q on Π_i for which the first edge of the shortest path from p to q lies in $C_j(p)$. By Lemma 2, the indices of the vertices in $C_j(p) \sqcap \Pi_i$ form a (possibly empty) cyclic interval $[k_1, k_2]$. If $C_j(p) \sqcap \Pi_i = \emptyset$, we do nothing. Otherwise, if $C_j(p) \sqcap \Pi_i \neq \emptyset$, there is a vertex $r \in C_j(p)$ closest to p, and we add the entry (i, k_1, k_2, r) to $\rho(p)$. This entry needs $\lceil \log h \rceil + 3 \cdot \lceil \log n \rceil$ bits.

Now, the routing function $f : V \times \ell(V) \to V$ is quite simple. Given a current vertex p and a target label $\ell(t) = (i, k)$, we search the routing table $\rho(p)$ for an entry (i, k_1, k_2, r) with $k \in [k_1, k_2]$. By construction, this entry is unique. We then forward the packet from p to the neighbor r.

Theorem 3 *Let P be a polygonal domain with n vertices and h holes. For any $\varepsilon > 0$ we can construct a routing scheme for $\mathrm{VG}(P)$ with labels of $\mathcal{O}(\log n)$ bits and routing tables of $\mathcal{O}((\varepsilon^{-1} + h) \log n)$ bits. For any two sites $p, q \in P$, the scheme produces a routing path with stretch factor at most $1 + \varepsilon$. The preprocessing time is $\mathcal{O}(n^2 \log n + hn^2 + \varepsilon^{-1} hn)$.*

For simple polygons, this result can be improved to the following.

Theorem 4 *Let P be a simple polygon with n vertices. For any $\varepsilon > 0$, we can construct a routing scheme for $\mathrm{VG}(P)$ with labels of $\lceil \log n \rceil$ bits and routing tables of $\mathcal{O}(\varepsilon^{-1} \log n)$ bits. For any two vertices $p, q \in P$, the scheme produces a routing path with stretch $1 + \varepsilon$. The preprocessing time is $\mathcal{O}(n^2 + \varepsilon^{-1} n)$.*

References

[1] S. Giordano and I. Stojmenovic. Position based routing algorithms for ad hoc networks: A taxonomy. In *Ad hoc wireless networking*, pages 103–136. Springer-Verlag, 2004.

[2] D. Peleg and E. Upfal. A trade-off between space and efficiency for routing tables. *J. ACM*, 36(3):510–530, 1989.

[3] L. Roditty and R. Tov. Close to linear space routing schemes. *Distributed Computing*, 29(1):65–74, 2016.

[4] M. Thorup and U. Zwick. Approximate distance oracles. *J. ACM*, 52(1):1–24, 2005.

[5] A. C.-C. Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SICOMP*, 11(4):721–736, 1982.

Gossiping with interference in radio ring networks

Jean-Claude Bermond*
COATI joint project CNRS-Inria-UNS, France,
jean-claude.bermond@inria.fr
Takako Kodate
Department of Information and Sciences,
Tokyo Woman's Christian University, Japan,
kodate@lab.twcu.ac.jp
Joseph Yu
Department of Mathematics,
University of the Fraser Valley, B.C., Canada,
joseph.yu@ufv.ca

1 Introduction

In this paper, we study the problem of gossiping with interference constraint in radio ring networks. Gossiping (or total exchange information) is a protocol where each node in the network has a message and wants to distribute its own message to every other node in the network. The gossiping problem consists in finding the minimum running time (makespan) of a gossiping protocol and efficient algorithms that attain this makespan.

Transmission model A radio network consists of communication devices equipped with an half duplex interface. The network is assumed to be synchronous and the time is slotted into *rounds*. The half-duplex hypothesis implies that a node can transmit or receive at most one message during a round. The network is modeled as a digraph, where the vertices represent the nodes and the arcs represent the possible communications. The messages are transmitted through the communication over the arcs and we will denote a call such a transmission.

Interference model We use a binary asymmetric model of interference based on the distance in the communication digraph like the ones used in [1, 2, 5]. Let $d(u, v)$ denote the distance, that is the length of a shortest directed path, from u to v in G and d_I be a non negative integer. We assume that when a node u transmits, all nodes v such that $d(u, v) \leq d_I$ are subject to the interference from u's transmission. So two calls (u, v) and (u', v') do not interfere if $d(u, v') > d_I$ and $d(u', v) > d_I$.

This problem has been studied in [4] where approximation results are given (see also the survey [3]). Here we focus on the case where the transmisson network is a ring network C_n on n nodes with the interference distance $d_I = 1$. We presented some partial results at `JCDCG^G 2013`, and we have now solved completely the gossiping problem by giving the minimum running time (makespan). We show lower bounds and then give gossiping algorithms which meet these lower bounds and so are optimal.

*Funded by ANR STINT

2 Main Result

It appears that the determination of the minimum gossiping time needs various tools. The optimal time depends on the congruence of n modulo 12. Our results are summarized in the following theorem.

Theorem 1 *The minimum number of rounds R needed to achieve a gossiping in a ring network C_n ($n \geq 3$), with the interference model $d_I = 1$ is :*

$$\begin{cases} 2n-3 & \text{if } n \equiv 0 (mod\,12) \\ 2n-2 & \text{if } n \equiv 4, 8 (mod\,12) \\ 2n-1 & \text{if } n \text{ is odd, except when } n = 3 \text{ for which } R = 3, \text{ and } n = 5 \text{ for which } R = 10 \\ 2n+1 & \text{if } n \equiv 2, 6, 10 (mod\,12) \text{ except for } n = 6 \text{ for which } R = 12 \end{cases}$$

References

[1] J.C. Bermond, R. Correa and M.L. Yu.: Optimal gathering protocols on paths under interference constraints, *Discrete Mathematics* 309: 5574–5587, 2009.

[2] J.-C. Bermond, J. Galtier, R. Klasing, N. Morales, and S. Pérennes. Hardness and approximation of gathering in static radio networks. *Parallel Processing Letters*, 16(2): 165–183, 2006.

[3] L. Gasieniec. On efficient gossiping in radio networks. Proc. Int. Conference on Theoretical Computer ScienceColloquium on structural information and communication complexity, SIROCCO 2009, Lectures Notes in Computer science, springer Verlag, vol 5869: 2–14, 2010.

[4] L. Gasieniec and I. Potapov. Gossiping with unit messages in known radio networks. Proc. 2nd IFIP Int. Conference on Theoretical Computer Science: 193–205, 2002.

[5] R. Klasing, N. Morales, and S. Pérennes. On the complexity of bandwidth allocation in radio networks. *Theoretical Computer Science*, 406(3): 225–239, 2008.

Edge Patrolling Beacon

Bahram Kouhestani David Rappaport *

1 Introduction

Consider the following scenario where a powerful magnet, which for historical reasons we call a *beacon*, moves on a rail located along the walls of a bounded interior space. Inside the same space is an iron ball, that due to magnetic attraction will move freely within the space. The ultimate goal in this scenario is to devise a strategy for the beacon to move on its rail so that eventually it and the ball are touching. The interior space is modelled by a simple polygon, and the beacon and ball are each modelled by points, labelled b and p respectively.

This work extends results that were introduced by Biro et al. [1], where beacons using magnetic pull generalize classic art gallery problems. A beacon b as defined in [1] is a point inside simple polygon P, and uses an attractive force to pull a point p. The point p moves directly towards b and either reaches it, or hits an edge of P. Then p may continue its movement towards b by sliding on an edge as long as its trajectory takes it closer to b. The motion of point p alternates between moving directly towards b or by sliding closer to b on an edge, until p either reaches b or becomes stuck on an edge at a point where it cannot get closer to b. If p reaches b we say that b *attracts* p, otherwise p is left stranded on an edge at a *dead point.*

We formally define the *edge patrolling beacon problem* as follows. Given a simple polygon P, an initial position for beacon b on the boundary of P and a point $p \in P$ find a moving strategy for b, if one exists, along the boundary of P, such that b attracts p. We assume that p moves arbitrarily faster than b thus precluding the possibility of the beacon using speed as a capture tactic. In this paper we give an efficient algorithm to solve the edge patrolling problem. We know of examples that show that some simple strategies do not always work. However, we have yet to find an example where no moving strategy exists.

2 An algorithm for edge patrolling beacon

We begin with some preliminary definitions.

Given a polygon P containing fixed beacon b, the *attraction region* of b is defined as the set of points in P that b attracts. As shown in [2] the attraction region is a simply connected subset of the interior of P. The set of points that are left stranded on a dead point d forms a simply connected subset of P that is called the *dead region* of b with respect to d. The boundary between two distinct dead regions or between a dead region and the attraction region of b is called a *split edge* (Fig. 1). The *attraction decomposition* of polygon P with respect to beacon b is defined as the decomposition of P by split edges of b. In a static case where the beacon is fixed, we can easily determine whether a point p is attracted to the beacon, or to a dead point and which one, by simply determining which region of the decomposition contains p. As beacon b moves on the boundary of P the attraction decomposition of P changes. We exploit the fact that there are a polynomial number of critical points on the boundary of P that can be used to partition these different attraction decompositions into equivalence classes.

Let v be a reflex vertex of P incident to edges e_1 and e_2. Let H_1 be the open half-plane perpendicular to e_1 at v that contains e_1, and similarly let H_2 be the open half-plane perpendicular to e_2 at v that contains e_2. The *deadwedge* of v is defined as the intersection of H_1 and H_2 (Fig. 1). Biro et al. [2] prove that a split edge s is a line segment inside P starting from some reflex vertex v along the ray $\overrightarrow{br}$ to the next boundary point of P if and only if b is in the deadwedge of v (Fig. 1). We say v is a *split vertex* with respect to b to describe this occurrence. We define two attraction decompositions with respective beacons b_1 and b_2 (both on the boundary of P) to be equivalent if and only if b_1 and b_2 and all beacons on the boundary of P between them (either in clock-wise or counter clock-wise direction) have the same set of split vertices. As a result two attraction decompositions in the same equivalence class have the same number of split edges and regions. A set of critical points on the boundary of P consists of reflex vertices with an internal angle greater than $\pi/2$, and intersections of rays bounding deadwedges, and the boundary of P (Fig. 2). Note that the attraction decomposition of a beacon b on a reflex vertex r with an internal angle greater

*School of Computing, Queen's University, Kingston, Ontario, Canada, {kouhesta,daver}@cs.queensu.ca

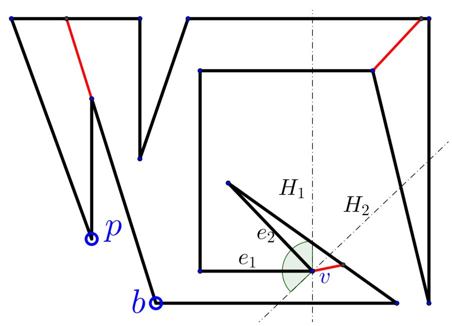


Figure 1: Split edges is shown in red. The deadwedge of v is shown by the green angle.

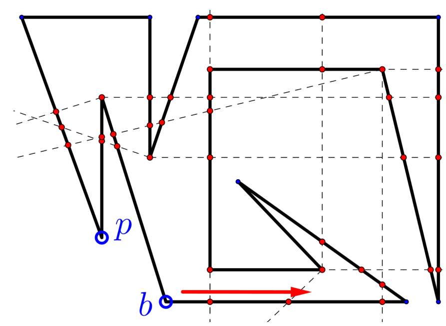


Figure 2: Critical points are shown in red.

than $\pi/2$ belongs to an equivalence class with a single element as b is not in the deadwedge of r and any other beacon on the adjacent edges of r with an infinitesimal distance to b is in the deadwedge of P.

We can now model the attractive ability of a moving beacon with a directed graph $G = (N, A)$. We begin by defining the nodes of G, and then follow that with specifying the directed edges. First we choose an arbitrary "test" attraction decomposition for each equivalence classes. Each node in G represents a region in a test attraction decomposition. Let m denote the number of different equivalence classes of attraction decompositions. We can obtain a cyclic order of these classes as they occur in a traversal of the boundary of P, numbered from $0 \dots m-1$, such that two attraction decompositions classes are adjacent if their cyclic order number differ by 1 (modulo m). We can model the regions of test attraction decomposition i by a set of nodes S_i such that there is a bijection between the nodes in S_i and the regions in attraction decomposition i. The set of nodes of G are precisely $N = \bigcup_{i=0}^{m-1} S_i$.

The directed edges of G are exclusively between nodes representing regions in adjacent classes. Therefore, consider the nodes in S_i and S_{i+1} with subscript addition modulo m. We first observe that the number of nodes in S_i and S_{i+1} differs by exactly one, reflecting the fact that the beacon either left a dead wedge or entered a new one. The nodes in S_i and S_{i+1} are related if one of the following conditions holds:

- Consider node x in S_i and node y in S_{i+1} such that the regions represented by x and y contain points that are attracted to the same dead point, or attracted to the beacon. In this case we have the directed edges (x, y) and (y, x).
- Consider node x and y in S_i and node z in S_{i+1} such that the split edge between regions associated to x and y disappears because the beacon moved out of a dead wedge. If either x or y represent an attraction region then z also represents an attraction region and from the previous case we already have directed edges (x, z), (z, x) or (y, z), (z, y). Otherwise, both x and y and therefore z represent dead regions. Furthermore, we assume without loss of generality that the dead point associated with region z is contained in region x. In this case we use directed edges (x, z), (y, z) and (z, x).

Next we mark a node in G as the starting node as follows. Consider the given initial position of b and p. We construct the attraction decomposition of P with respect to b and determine the region that contains p. We mark the node that represents this region in the corresponding attraction decomposition class as the starting node. Now we use Dijkstra's algorithm to find a path from the starting node to the first node that represents an attraction region in an attraction decomposition class. Note that such a node represents a configuration where p is in a attraction region. There are $O(n^3)$ nodes and edges in G and therefore the running time of the algorithm is $O(n^3 \log n)$.

References

[1] M. Biro and J. Gao, J. Iwerks, I. Kostitsyna and J. S. B. Mitchell. Beacon-based routing and coverage. Proceedings of the 21st Fall Workshop on Computational Geometry, 2011.

[2] M. Biro, J. Iwerks, I. Kostitsyna and J. S. B. Mitchell. Beacon-Based Algorithms for Geometric Routing. Proceedings of the 13th International Symposium on Algorithms and Data Structures, WADS, 2013.

Constructions of Ramanujan Cayley graphs

SHOHEI SATAKE (Kobe University)

1 Introduction

Ramanujan graph was introduced by Lubotzky-Phillips-Sarnak [4]. Let $k > 0$. Let G be a k-regular graph and $\lambda(G) = \max\{|\lambda| \mid \lambda \in \mathrm{Spec}(G), |\lambda| \neq k\}$. Here, $\mathrm{Spec}(G)$ is the set of all eigenvalues of the adjacency matrix of G. A k-regular graph G is a *Ramanujan graph* if $\lambda(G) \leq 2\sqrt{k-1}$. It is well known that Ramanujan graphs have wide applications in coding theory, computer science and so on (see [2]). In graph theory, to give explicit constructions of Ramanujan graphs is recognized as an interesting problem. In these two decades, many explicit constructions of Ramanujan graphs were given (for example [1], [3]).

2 Our results

Our main purpose is to give explicit constructions of Ramanujan graphs. In this talk, we focus on Cayley graphs over finite abelian groups. Let A be a finite abelian group with identity 0 and $S \subset A \setminus \{0\}$ which satisfies that $s \in S$ implies $-s \in S$. Then, the *Cayley graph* $X(A, S)$ over A with respect to S is defined as follows.

$$V(X(A,S)) = A,\ E(X(A,S)) = \{\{x,y\} \mid x - y \in S\}.$$

Clearly, $X(A, S)$ is well-defined and $|S|$-regular. As remarked by Li-Feng [3],

$$\lambda(X(A,S)) = \max\left\{\left|\sum_{s\in S}\psi(s)\right| \;\middle|\; \psi \text{ is non-trivial character of } A\right\}.$$

Here, a *non-trivial character* ψ of A is a homomorphism from A to the complex multiplicative group $\mathbb{C}^*$ such that there exists $a \in A$ with $\psi(a) \neq 1$. Thus, we shall find a suitable subset $S \subset A \setminus \{0\}$ such that

$$\max\left\{\left|\sum_{s\in S}\psi(s)\right| \;\middle|\; \psi \text{ is non-trivial character of } A\right\} \leq 2\sqrt{|S|-1}.$$

In this talk, following this idea, we construct some infinite families of Ramanujan Cayley graphs over finite fields and finite commutative rings.

If time permits, we also discuss some graph-theoretic properties like existentially closedness of our graphs.

References

[1] K. Bibak, B. M. Kapron, V. Srinivasan. The Cayley graphs associated with some quasi-perfect Lee codes are Ramanujan graphs. *IEEE Trans. Inform. Theory* **62** (2016), no. 11, 6355–6358.

[2] S. Hoory, N. Linial, A. Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc. (N.S.)* **43** (2006), no. 4, 439–561.

[3] W.-C. W. Li, K. Feng, Character sums and abelian Ramanujan graphs, *J. Number Theory* **41** (1992), no. 2, 199–217.

[4] A. Lubotzky, R. Phillips, P. Sarnak, Ramanujan graphs, *Combinatorica* **8** (1988), no. 3, 261–277.

Subgroups as Total Perfect Codes in Cayley Graphs of Abelian and Dihedral Groups

Agnes D. Garciano, Reginaldo M. Marcelo, Luis S. Silvestre, Jr.[1]
agarciano@ateneo.edu, rmarcelo@ateneo.edu, silvestre2117@yahoo.com
Ateneo de Manila University

Let G be a group and $S \subset G$ such that $S = S^{-1}$ and $1 \notin S$. The *Cayley graph* $\Gamma = \text{Cay}(G, S)$ on G with respect to S is a graph with vertex set $V(\Gamma) = G$ and edge set $E(\Gamma) = \{xy : x, y \in G, xy^{-1} \in S\}$. S is referred to as the *connection set* of the Cayley graph Γ. We say that a subset C of G is a *total perfect code (TPC)* in the Cayley graph Γ if every vertex in $V(\Gamma)$ has exactly one neighbor in C. A consequence of G admitting a total perfect code is that the order of G must be even.

In [4], total perfect codes were referred to as efficient dominating sets. In [1] and [3], characterizations of grid graphs which contain total perfect codes were studied. In [2], total perfect codes in the lattice graph of $\mathbb{Z} \times \mathbb{Z}$ were investigated. Zhou [5] used pseudocovers and gave criteria under which a Cayley graph of a group will admit a total perfect code.

In this study, we consider abelian and dihedral groups and determine when a subgroup will be a total perfect code in some Cayley graph of the group. If G is the cyclic group $\mathbb{Z}_n$ where n is even and $g \in G$, we will show that the subgroup $\langle g \rangle$ generated by g is a total perfect code in $\text{Cay}(G, S)$ for some $S \subset G$ if and only if g is odd. In general, if $A = \prod_{i=1}^{w} \mathbb{Z}_{2^{k_i}} \times G$ is an abelian group, where $k_i \geq 1$ for $1 \leq i \leq w$ and G is an odd ordered abelian group, then the cyclic subgroup $C = \langle (r_1, r_2, \ldots, r_j, \ldots, r_w, g) \rangle$ of A where $(r_1, r_2, \ldots, r_j, \ldots, r_w, g) \in A$ is a total perfect code in some Cayley graph of A if and only if there exists $j, 1 \leq j \leq w$ such that r_j is odd. Moreover, we will show that a non cyclic subgroup $C = \prod_{i=1}^{w} C_i \times C'$ of A where $C_i \leq \mathbb{Z}_{2^{k_i}}$ for $1 \leq i \leq w$ and $C' \leq G$ is a total perfect code in some Cayley graph of A if and only if $C_i = \{0\}$ or $C_i \cong \mathbb{Z}_{2^{k_i}}$ for every $i, 1 \leq i \leq w$, and $C_i \neq \{0\}$ for at least one $i, 1 \leq i \leq w$.

In the dihedral group $D_{2n} = \langle r, s : r^n = s^2 = 1, rs = sr^{-1} \rangle$, subgroups are either of the form $\langle r^d \rangle$ or $\langle r^d, sr^e \rangle$ for some divisor d of n and $0 \leq e < d$. We will show the following results:

i. If n is odd, then a proper subgroup C of D_{2n} is a total perfect code in some Cayley graph of D_{2n} if and only if $C = \langle r^d, sr^e \rangle$ for some divisor d of $n, d > 1$ and $0 \leq e < d$.

ii. If n is even, then a proper subgroup C of D_{2n} is a total perfect code in some Cayley graph of D_{2n} if and only if C is one of the following:

- $C = \langle r^d \rangle$ for some odd divisor d of n
- $C = \langle r^d, sr^e \rangle$ for some divisor $d > 1$ of n and $0 \leq e < d$.

Keywords: Cayley graphs, total perfect codes, abelian group, dihedral group

References:

[1] Cowen, Robert, Hechler, Stephen, Kennedy, John, and Steinberg, Arthur. Odd neighborhood transversals on grid graphs. Discrete Math. no. 307 (2007): 2200- 2208.

[2] Dejter, Italo. Perfect domination in regular grid graphs. Australasian Journal of Combinatorics, no. 42 (2008) : 99-114

[3] Klostermeyer, William and Goldwasser, John. Total perfect codes in grid graphs. Bull. Inst. Combin. Appl. no. 46 (2006): 61-68.

[4] Kuziak, Dorota, Peterin, Iztok, and Yero, Ismael. Efficient open domination in graph products. Discrete Math. and Theoret. Comp. Sci. no. 16 (2014): 105-120

[5] Zhou, Sanming. Total Perfect Codes in Cayley Graphs. Designs, Codes and Cryptography 81, no. 3 (December 2016): 489-504.

Public key cryptosystems using magic cubes

Tomoko Adachi
Department of Information Sciences, Toho University
2-2-1 Miyama, Funabashi, Chiba, 274-8510, Japan
E-mail: adachi@is.sci.toho-u.ac.jp
Xiao-Nan Lu ⋆
Department of Industrial Administration, Tokyo University of Science
2641 Yamazaki, Noda, Chiba, 278-8510, Japan
E-mail: lu@rs.tus.ac.jp

Keywords: Cryptography, Latin cube, Magic cube.

A cryptosystem is developed for secure message transmission, which converts plain texts (messages) to coded texts, called cipher texts, and back. The cryptosystems can be categorized into two types, private key cryptosystems and public key cryptosystems. The first public key cryptosystem was presented by Diffie and Hellman in 1976. Since then, public key cryptosystems have been variously studied. For example, the most widely used RSA and ElGamal are both public key cryptosystems, whose algorithms are based on the integer factorization problem and the discrete logarithm problem, respectively. In recent years, the public key cryptosystems using magic squares were introduced in the literature (see [1], [2], and [5]).

Magic squares can be traced back to ancient times in China and India, and have been popular puzzles for hundreds of years. A magic square of order n is an arrangement of n^2 integers from $\{1, 2, \cdots, n^2\}$ into an $n \times n$ square array with the property that the sums of each row, each column, and each of the main diagonals are the same. The study of magic squares is deeply related to Latin squares, which have been intensively investigated in combinatorial design theory (see [3] and [4]).

A magic cube is a 3-dimensional generalization of a magic square (see Trenkler [6], [7], and [8]). Let M_n be an $n \times n \times n$ cubical array with different entries $m_n(i, j, k) \in \{1, 2, \cdots, n^3\}$ for $1 \le i, j, k \le n$ such that the sums of the numbers along every row and every diagonal are the same. Note that each row in a cube is n-tuple of elements having the same coordinates in two dimensions. It can be easily checked that each row sum is $n(n^3 + 1)/2$. For example, the following gives a magic cube of order 3.

$$M_3(i,j,1) = \begin{bmatrix} 10 & 26 & 6 \\ 24 & 1 & 17 \\ 8 & 15 & 19 \end{bmatrix}, M_3(i,j,2) = \begin{bmatrix} 23 & 3 & 16 \\ 7 & 14 & 21 \\ 12 & 25 & 5 \end{bmatrix}, M_3(i,j,3) = \begin{bmatrix} 9 & 13 & 20 \\ 11 & 27 & 4 \\ 22 & 2 & 18 \end{bmatrix}.$$

Here, the element $m_3(1, 1, 1) = 10$ is contained in three rows (triples) $\{10, 26, 6\}$, $\{10, 24, 8\}$, and $\{10, 23, 9\}$, and a diagonal $\{10, 14, 18\}$.

In this talk, we will investigate a public key cryptosystem using magic cubes, and the corresponding algorithm.

References

[1] G. Ganapathy and K. Mani (2009); Add-on security model for public-key cryptosystem based on magic square implementation, *Proceedings of the World Congress on Engineering and Computer Science 2009 Vol. WCECS 2009*, October 20-22, 2009, San Francisco, USA

[2] S. Jamel, T. Herawan, and M. M. Deris (2010); A cryptographic algorithm based on hybrid cubes, *D. Taniar et al. (Eds.) : Proceedings of International Conference on Computational and Its Applications, ICCSA2010, Part IV*, LNCS 6019, pp. 175–187.

[3] C. F. Laywine and G. L. Mullen (1998); *Discrete Mathematics Using Latin Squares*, John Weiley & Sons, INC.

[4] I. H. Morgan (1998); Construction of complete sets of mutually equiorthogonal frequency hypercubes, *Discrete Mathematics*, Vol. 186, pp. 237–251.

[5] M. S. Rahma, M. J. Abdul Hossen, and O. A. Dawood (2016); Public key cipher with signature based on Diffie-Hellman and the magic square problem, *Eng. & Tech. Journal*, Vol. 34, Part(B), No.1.

[6] M. Trenkler (1998); Magic p-dimensional cubes, *Acta Arithmetica*, Vol. 96, pp. 361–364.

[7] M. Trenkler (2000); A construction of magic cubes, *The Mathematical Gazette*, Vol. 84, pp. 36–41.

[8] M. Trenkler (2005); An algorithm for making magic cubes, *The* Π*ME Journal*, Vol. 12, No. 2, pp. 105–106.

COMPLETE k-ZERO-DIVISOR HYPERGRAPHS OF SOME COMMUTATIVE RINGS

Ratinan Boonklurb*, Sajee Pianskool, and Pinkaew Siriwong

Abstract

Let R be a commutative ring with nonzero identity and $k > 1$ be a fixed integer. The k-zero-divisor hypergraph $\mathcal{H}_k(R)$ of R consists of the vertex set $Z(R,k)$, the set of all k-zero-divisors of R, and the (hyper)edges of the form $\{a_1, a_2, a_3, ..., a_k\}$ where $a_1, a_2, a_3, ..., a_k$ are k distinct elements in $Z(R,k)$, which means (i) $a_1a_2a_3 \cdots a_k = 0$ and (ii) the products of all elements of any $(k-1)$-subsets of $\{a_1, a_2, a_3, ..., a_k\}$ are nonzero. This talk provides a necessary condition of commutative rings that implies the completeness of their k-zero-divisor hypergraphs. Moreover, the diameter and the minimum length of all cycles of those hypergraphs are determined.

Throughout this paper, we consider a commutative ring R/I where R is a PID and I is the appropriate ideal of R instead of considering directly a commutative ring R. The existence of a prime element p of R and the finiteness of R/Rp^k together with $|(Rp/Rp^k) - (Rp^2/Rp^k)| \geq k$ enable us to construct a complete hypergraph $\mathcal{H}_k(R/Rp^k)$. By the completeness, the diameter of $\mathcal{H}_k(R/Rp^k)$ is 1 and the minimum length of all cycles of $\mathcal{H}_k(R/Rp^k)$ can be either 3; 2 or 0 depends on k and $|Z(R/Rp^k, k)|$. As for the minimum length of all cycles, 2 is not a general case for complete graphs because each edge can be formed by more than two vertices and then some cycles can be constructed by at least two edges.

Keywords: k-zero-divisor; k-zero-divisor hypergraph; complete k-uniform hypergraph; k-partite σ-uniform hypergraph.
2010 Mathematics Subject Classification: 05C65

*Speaker and Corresponding author

R. Boonklurb, S. Pianskool and P. Siriwong are with the Department of Mathematics and Computer Science, the Faculty of Science, Chulalongkorn University, Bangkok 10330, Thailand (email: ratinan.b@chula.ac.th, sajee.pianskool.p.s@gmail.com and psiriwong@yahoo.com).

MazezaM Levels with Exponentially Long Solutions (Extended Abstract)

Aster Greenblatt* Justin Kopinsky† Ben North‡ Malcolm Tyrrell§ Aaron Williams¶

Abstract

MazezaM is a sliding block tour puzzle and popular video game. It has similar mechanics to Sokoban except that boxes only move horizontally, and every box in a row moves simultaneously. It was previously known that deciding if a given level can be solved is NP-hard (North 2008). In this paper we provide a family of levels whose shortest solutions are exponentially long with respect to their size, which suggests that MazezaM could be PSPACE-complete. Interestingly, the levels are constructed to simulate the binary reflected Gray code.

1 Introduction

Sliding block puzzles were among the first puzzles to be studied by computational complexity. For example, the hit Japanese video game *Sokoban* was shown to be PSPACE-complete by Culbertson [1] in 1998. Sliding block puzzles have also been very influential in the literature. For example, the ThinkFun puzzle *Rush Hour* was shown to be PSPACE-complete by Flake and Baum [2] using a form of dual-rail logic. This technique was refined by Hearn and Demaine into *constraint logic* which is the basis for their well-known and influential textbook *Games, Puzzles, and Computation* [4].

When a decision problem is suspected to be PSPACE-hard it is prudent to show that the problem is not obviously in NP. For sliding block puzzles this corresponds to finding puzzles whose shortest solutions are exponential in the size of the puzzle.

We consider *MazezaM*, a sliding block tour puzzle invented by Malcolm Tyrrell and released as a free and open source video game. Each level is played on a rectangular grid surrounded by walls. The player's goal is to move the token from the entrance corridor to the exit corridor. The obstacles are 1-by-1 boxes that can be pushed horizontally by the token. However, every box in a row moves simultaneously when any box in the row is pushed. Boxes do not move vertically, and they cannot move into the corridors. A level and solution are shown in Figure 1. A free playable version is available at `www.puzzlescript.net/play.html?p=7718522`.

*Bard College Simon's Rock `mgreenblatt15@simons-rock.edu`
†Massachusetts Institute of Technology `jkopin@mit.edu`
‡`ben@redfrontdoor.org`
§`malcoh01@yahoo.ie`
¶Bard College at Simon's Rock, `awilliams@simons-rock.edu`

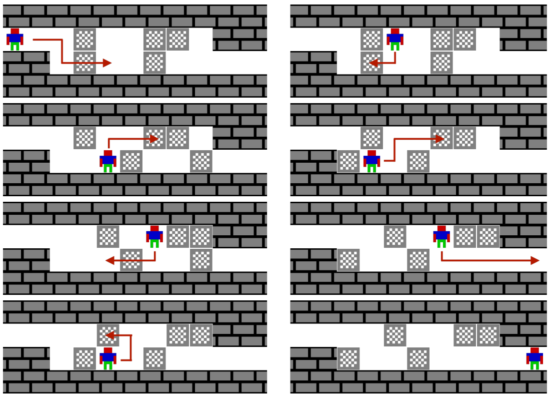


Figure 1: MazezaM level "*Humble Origins*". The wiggle room for the top row is 2 and for the bottom row is 3.

Determining if a MazezaM level is solvable is NP-hard by an unpublished manuscript by North [6]. We provide a family of 'Gray code' levels whose shortest solutions are exponentially long with respect to the size of level.

2 Binary Reflected Gray Code

Let $\mathcal{B}(n)$ be the set of n-bit binary strings. The *weight* of $b_1 b_2 \cdots b_n \in \mathcal{B}(n)$ is its bitwise sum $\sum_{i=1}^{n} b_i$. We use exponents to denote bitwise concatenation. For example, $1^4 = 1111$ is the only string of weight four in $\mathcal{B}(4)$.

The *binary reflected Gray code* (BRGC) is an ordering of $\mathcal{B}(n)$ in which each pair of consecutive strings have Hamming distance one (i.e. they differ in exactly one bit) [3]. The order starts with 0^n and ends with $0^{n-1}1$. The BRGC for $n = 4$ is below with overlines showing the bit that changes to create the next string:

$$\bar{0}000, 1\bar{0}00, \bar{1}100, 01\bar{0}0, \bar{0}110, 1\bar{1}10, \bar{1}010, 001\bar{0},$$
$$\bar{0}011, 1\bar{0}11, \bar{1}111, 01\bar{1}1, \bar{0}101, 1\bar{1}01, \bar{1}001, 0001.$$

Now we explain how to create each successive string in the BRGC starting from the initial string 0^n.

Definition 1 *Each $b_1 b_2 \cdots b_n \in \mathcal{B}(n)$ has up to two* active bits*: (a) its leftmost bit b_1, and (b) its bit immediately to the right of its leftmost 1.*

For example, the leftmost 1 in $b_1 b_2 b_3 b_4 b_5 b_6 = 000111$ is $b_4 = 1$; therefore, its active bits are b_1 and b_5. Every binary string has two active bits except 0^n and $0^{n-1}1$.

The following theorem is well-known (see Knuth [5]).

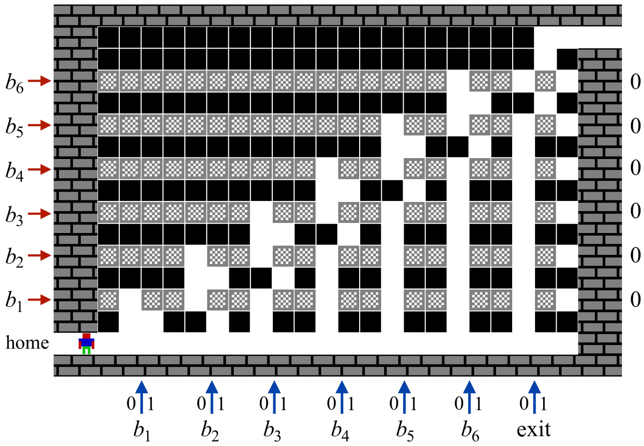


Figure 2: **Gray**(6) with b_i row labels (left), b_j columns (bottom), in initial state 000000 (right).

Theorem 1 *If $b_1 b_2 \cdots b_n$ has even weight, then complementing active bit (a) gives the next string in the BRGC. Otherwise, if $b_1 b_2 \cdots b_n$ has odd weight, then complementing active bit (b) gives the next string.*

On the other hand, complementing the 'other' active bit of $b_1 b_2 \cdots b_n$ gives the previous string in the BRGC.

For example, 000111 has odd weight, so 000101 is the next string in the BRGC and 100111 is the previous.

3 Exponentially Long Levels

Now we construct a MazezaM level **Gray**(n) that simulates the BRGC. The key features of **Gray**(n) are:

- There are n rows that simulate each of the n bits. The overall 'state' is a binary string $b_1 b_2 \cdots b_n$.
- There are n sets of columns to ensure only 'active' bit rows (as per Definition 1) can be modified.
- The initial state is 0^n and the exit state is 10^{n-1}.

Each b_i *row* has two positions: If its boxes are shifted left/right, then its *state* is 0/1. The *state* of **Gray**(n) (excluding the token) is then $b_1 b_2 \cdots b_n \in \mathcal{B}(n)$. The boxes in other rows cannot move and are drawn in black.

Each pair of b_j *columns* has a *0-column* and *1-column* as do the *exit columns*. The *column pairs* have *barrier columns* to the left/right to prevent horizontal moves. The column structure is described further in Lemma 2.

The bottom row is empty and is called the *home row*. The level starts in state 0^n with the player in the home row, as illustrated for **Gray**(6) in Figure 2.

The player *modifies* b_i when they leave the home row, change row b_i, and return to the home row. Lemma 2 proves that modifications can only be done to rows and columns whose bit is active. For example, see Figure 3. A similar proof establishes the exit criteria in Lemma 3.

Lemma 2 *The player can modify b_i via the b_j columns if and only if $i = j$ and b_i is active in* **Gray**(n)*'s state.*

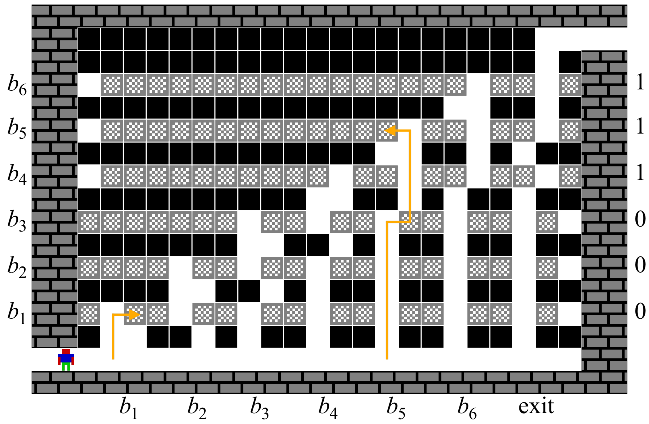
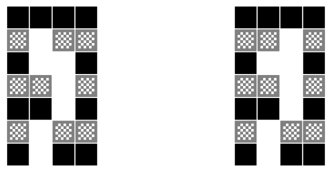


Figure 3: When **Gray**(6) is in state $b_1 b_2 b_3 b_4 b_5 b_6 =$ 000111 the player can modify the active bits b_1 or b_5.

Proof Sketch: Consider the b_3 columns (and barriers) in state $b_1 b_2 b_3 = 010$ (left) and $b_1 b_2 b_3 = 011$ (right).



In both cases b_3 can be modified (which involves changing b_1 twice). This isn't true if $b_1 b_2 \neq 01$ (i.e. when b_3 is not active). The remaining b_j columns are similar. □

Lemma 3 *The player can pass through the exit columns if and only if* **Gray**(n)*'s state is $0^{n-1}1$.*

Theorem 4 *The level MazezaM level* **Gray**(n) *can be solved in no fewer than $2^n - 1$ moves.*

Proof. The player must change the state of **Gray**(n) from 0^n to $0^{n-1}1$ by Lemma 3. Modifications can only be done to active bits by Lemma 2. Therefore, $2^n - 1$ moves are required by Theorem 1. □

References

[1] J. Culbertson. Sokoban is PSPACE-complete. In *Fun With Algorithms (FUN 1998)*, Lecture Notes in Computer Science, pages 65–76. Carleton Scientific, 1998.

[2] G. W. Flake and E. B. Baum. Rush hour is PSPACE-complete, or "why you should generously tip parking lot attendants". *Theor. Comput. Sci.*, 270:895–911, 2002.

[3] F. Gray. Pulse code communication. *U.S. Patent 2,632,058*, 1947.

[4] R. A. Hearn and E. D. Demaine. *Games, Puzzles, and Computation.* A K Peters/CRC Press, 1st edition, 2009.

[5] D. E. Knuth. *The Art of Computer Programming*, volume 4: Generating All Tuples and Permutations. Addison-Wesley, 2005.

[6] B. North. Mazezam is NP-complete. http://redfrontdoor.org/blog/?p=174, 2008.

Path Puzzles: Discrete Tomography with a Path Constraint is Hard

Jeffrey Bosboom* Erik D. Demaine* Martin L. Demaine* Adam Hesterberg*
Roderick Kimball† Justin Kopinsky*

Path Puzzles are a type of logic puzzle introduced in Roderick Kimball's 2013 book [5]. A puzzle consists of a (rectangular) grid of cells with two exits (or "doors") on the boundary and numerical constraints on some subset of the rows and columns. A solution consists of a single non-intersecting path which starts and ends at two boundary doors and which passes through a number of cells in each constrained row and column equal to the given numerical clue. Figure 1 shows some example path puzzles and Figure 4 shows their (unique) solutions. Many variations of path puzzles are given in [5] and elsewhere, for example using non-rectangular grids, grid-internal constraints, and additional candidate doors, but these generalizations make the problem only harder.

A path puzzle can be seen as 2-dimensional *discrete tomography* [3] problem with partial information (not all row and column sums) and an additional Hamiltonicity constraint on the output image. Vanilla 2-dimensional discrete tomography is known to have efficient (polynomial-time) algorithms [3], though it becomes hard under certain connectivity constraints on the output image [2].

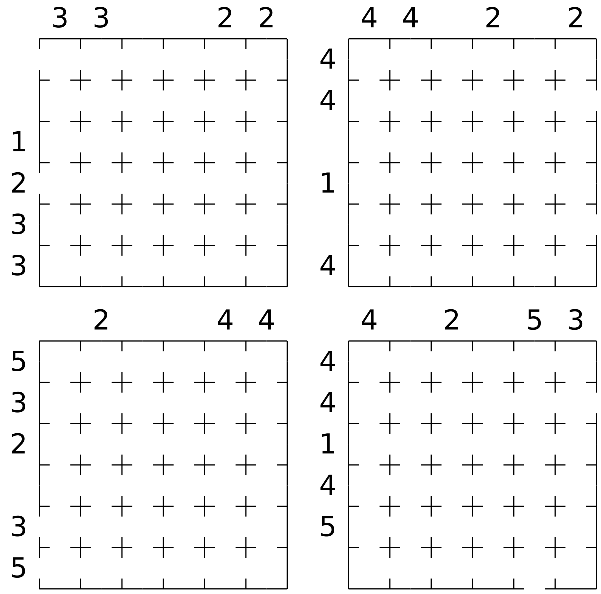


Figure 1: Four path puzzles. Solutions in Figure 4 on the next page.

Our results. Unlike 2-dimensional discrete tomography, we show that path puzzles (with partial information and the added Hamiltonicity constraint) are in fact NP-complete. In fact, we prove the stronger results that path puzzles are Another Solution Problem (ASP) hard and (to count solutions) #P-complete. Figure 2 shows the chain of reductions we prove. To preserve hardness for the ASP and #P classes, our reductions are *parsimonious*, that is, they preserve the number of solutions between the source and target problem instances, generally by showing a one-to-one correspondence thereof. We start from the source problem of Positive Exactly-1-In-3-SAT which is known to be ASP-hard [6] and (to count solutions) #P-complete [4]. We newly establish ASP-hardness and #P-completeness for 3-Dimensional Matching, Numerical 4-Dimensional Matching, Numerical 3-Dimensional Matching, and a new problem Length Offsets, in addition to Path Puz-

*Massachusetts Institute of Technology, {jbosboom, edemaine,mdemaine,achester,jkopin}@mit.edu
†Enigami Puzzles & Games

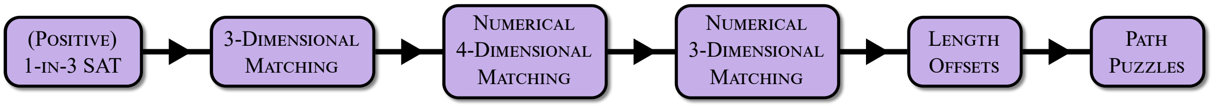


Figure 2: The chain of reductions used in our proof.

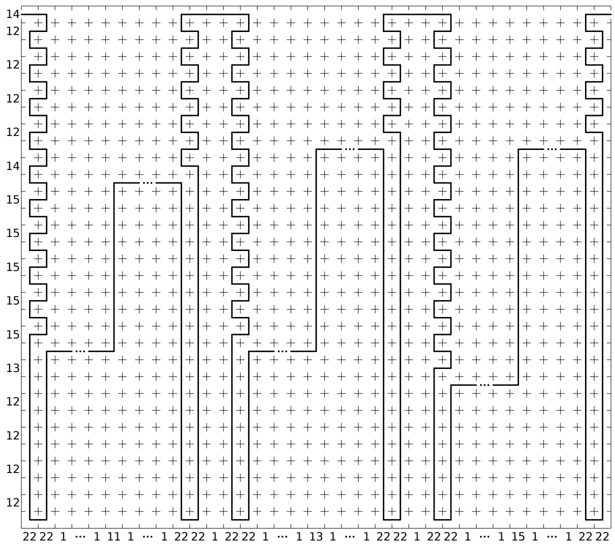


Figure 3: The reduction Numerical 3-Dimensional Matching → Length Offsets → Path Puzzles (with intended solution) representing Numerical 3-Dimensional Matching instance $X = \{5, 6, 7\}$, $Y = \{4, 5, 5\}$, $Z = \{4, 4, 5\}$, and target sum $t = 15$. Ellipses elide sections of $6n = 18$ columns each labeled 1.

zles. Figure 3 gives a flavor of our reductions.

We also present a path puzzles font—a set of 26 path puzzles whose (unique) solutions depict the alphabet. Figures 1 and 4 show the J, C, D, and G puzzles.

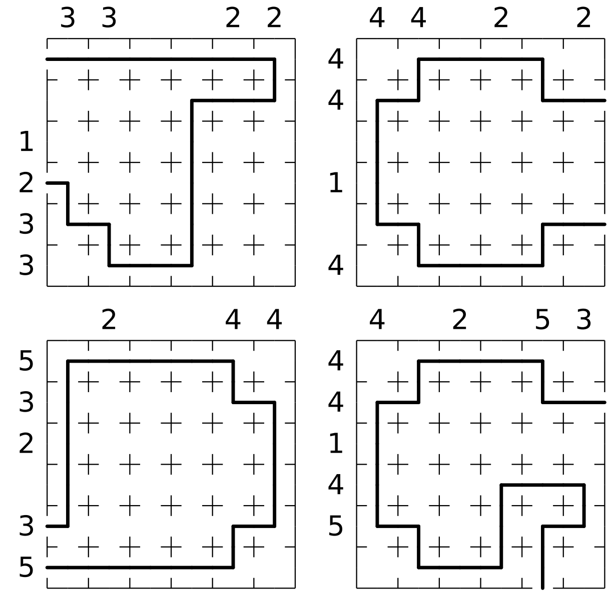


Figure 4: Solutions to the path puzzles in Figure 1. What can you spell?

References

[1] Gary Antonick. Roderick Kimball's path puzzles. *The New York Times: Numberplay*, 28 July 2014. https://wordplay.blogs.nytimes.com/2014/07/28/path-2/.

[2] Alberto Del Lungo and Maurice Navat. Reconstruction of connected sets from two projections. In Herman and Kuba [3], chapter 7.

[3] Gabor T. Herman and Attila Kuba, editors. *Discrete Tomography: Foundations, algorithms, and applications.* Birkhäuser, 2012.

[4] Harry B. Hunt, III, Madhav V. Marathe, Venkatesh Radhakrishnan, and Richard E. Stearns. The complexity of planar counting problems. *SIAM Journal on Computing*, 27(4):1142–1167, 1998.

[5] Roderick Kimball. *Path Puzzles.* Roderick Kimball, 2nd edition, 10 2013.

[6] Takahiro Seta. The complexities of puzzles, Cross Sum, and their Another Solution Problems (ASP), 2002.

Elmsley's Problem Revisited

Rudolf Fleischer

GUtech, Muscat, Oman
Email: rudolf.fleischer@gutech.edu.om

Abstract. In this paper we revisit the perfect shuffling problem. We propose a slightly different mathematical description of the problem which gives rise to a simple solution for Elmsley's problem, which is the problem of finding the shortest perfect shuffle sequence that brings a particular card to the top of the deck, or more general to any fixed position in the deck. We also give a negative answer to the open problem whether it is always possible to bring two particular cards to the two top positions of the deck, at least if the number of cards is a power of 2.

1 Introduction

A *perfect shuffle* (or *riffle shuffle*) of a deck of $2n$ cards first cuts the deck in half and then perfectly interlaces the two halves. This can be done in one of two ways. An *out shuffle* (**O**) leaves the top card on top, while an *in shuffle* (**I**) moves the top card into the second position. Perfect shuffles are popular among magicians who usually refer to it as the *Faro shuffle* since it was first described as a method of cheating at the game of Faro nearly 200 years ago [5, p. 195]. For a more comprehensive review of the history of the Faro shuffle and its applications in magic and mathematics we refer to [3, 8].

The combinatorial structure of Faro shuffles has been extensively studied [3, 2, 7, 8, 11], including generalizations to k-shuffles [4, 9, 10] and other variants of the problem [1]. While most of the literature has been concerned with the structure of the group generated by in and out shuffles and the problem of moving the top card of the deck to an arbitrary position by a sequence of in and out shuffles, much less is known about the problem of moving an arbitrary card to the top of the deck, or more general to any fixed position in the deck. This problem is also known as *Elmsley's problem*.

Ramnath and Scully [11] proposed a brute-force search among all possible shuffle sequences, ordered by increasing length, to find the shortest solution to Elmsley's problem. Diaconis and Graham [2] proposed an algorithm that avoids the tedious enumeration of all shuffle sequences by computing the solution directly using a formula that becomes complicated if $2n$ is not a power of 2. However, contrary to what they claim (but not prove), this direct approach does not always yield a shortest shuffle sequence. For example, if we want to move the card in position 39 to the top in a deck of 52 cards (we assume cards are numbered $0, 1, 2, \ldots$ from the top), their algorithm computes a shuffle of length six, **OOOIOI**, while there exist a shuffle of length two, **II**. This can easily be verified by playing with the Faro shuffle simulator by Kiewel [6], for example. They also raised the question whether any two cards can be moved to the two top positions by a sequence of Faro shuffles.

We propose a different mathematical description of Faro shuffles which yields an easy algorithm for Elmsley's problem and gives more insight into the combinatorial structure of the problem. For example, we can show that not every pair of cards can be simultaneously moved to the two top positions of the deck if $2n$ is a power of 2. Although it is known that Faro shuffles cannot always generate all possible permutations, this is a new result and answers the open question raised in [2].

Let k denote the number of bits of $bin(2n)$, where $bin(z)$ is the binary representation of integer z, and p, q, x, y card positions.

Theorem 1.

(a) *If c is the smallest non-negative integer such that the first k bits of $bin(2cn + q)$ (we may add abitrarily many 0's at the left) are identical to $bin(p)$, then the remaining $z = k - size(bin(2cn + q))$ bits (from left to right) determine the shortest shuffle sequence that moves a card from position p to q, where a digit 0 (1) denotes an out shuffle if the current position of the card is in the upper (lower) half of the deck, otherwise it indicates an in shuffle.*
(b) *There also exist shuffle sequences of length $z + 1, z + 2, \ldots$ that move the card from position p to q.*
(c) *If $2n = 2^k$, then the shortest shuffle sequence is determined by $x \oplus y$ (bitwise XOR), where $p = x \circ t$ and $q = t \circ y$ ($\circ$ denotes the concatenation of two bit strings) and t is the longest bit string that is a suffix of p and a prefix of q.*

Part (c) has been known in the special case that either p or q is 0 $[1, 3, 11]$. We now assume that $2n = 2^k$. Let $rot(s, t)$ denote the shortest left rotation distance from bit string s to t, where one rotation step removes the leftmost digit of s and puts it back at the right end of s. It is easy to see that for positions p and q the string $bit(p \oplus q)$ is an invariant up to rotation that does not change when we apply Faro shuffles. This implies, for example, that any pair of cards at positions with complementary bit strings (e.g., the top and bottom cards of the stack) will always be in complementary positions after Faro shuffles [1].

Theorem 2. *If $2n = 2^k$, then we can move the card at position p to q and at the same time the card at position x to y if and only if $p \oplus q$ and $x \oplus y$ are identical up to rotation. In this case, the shortest shuffle sequence has length $ck + rot(p \oplus q, x \oplus y)$, where c is minimal such that this value is at least the length of the shortest shuffle sequence from position p to q.*

For example, we can only move two cards to the two top positions if their bit strings differ in exactly one bit, i.e., in a deck of 64 cards we can move the cards at positions 22 and 23 to the top via **OIOIIO**, but not the cards at positions 22 and 21. We can easily find the shortest Faro shuffle by Theorem 1.

References

1. S. Butler, P. Diaconis, and R. Graham. The mathematics of the flip and horseshoe shuffles. *American Mathematical Monthly*, 123:542–556, 2016.
2. P. Diaconis and R. L. Graham. The solutions to Elmsley's problem. *Math Horizons*, 14:22–27, 2007.
3. P. Diaconis, R. L. Graham, and W. N. Kantor. The mathematics of perfect shuffles. *Advances in Applied Mathematics*, 4(2):175–196, 1983.
4. J. Ellis, H. Fan, and J. Shellit. The cycles of the multiway perfect shuffle permutation. *Discrete Mathematics and Theoretical Computer Science*, 5:169–180, 2002.
5. J. H. Green. *An Exposure to the Arts and Miseries of Gambling.* James, Cincinnati, 1843.
6. N. Kiewel. Faro shuffle simulator, 2004. http://www.natedog.com/cards/faro.html.
7. S. B. Morris. The basic mathematics of the Faro shuffle. *Pi Mu Epsilon Journal*, 6(2):85–92, 1975.
8. S. B. Morris. *Everything You Always Wanted To Know About The Faro Shuffle: but were afraid to ask.* eBook, 2011.
9. S. B. Morris and R. E. Hartwig. The generalized Faro shuffle. *Discrete Mathematics*, 15:333–346, 1976.
10. R. W. Packard and E. S. Packard. The order of a perfect k-shuffle. *The Fibonacci Quaterly*, 32(2):136–144, 1994.
11. S. Ramnath and D. Scully. Moving card i to position j with perfect shuffles. *Mathematics Magazine*, 69:361–365, 1996.

An efficient algorithm for judicious partition of hypergraphs

Tunzi Tan[1,2], Jihong Gui[1,2], Sainan Wang[1,2], Suixiang Gao[1,2], and Wenguo Yang[1,2]

[1] School of Mathematical Sciences, University of Chinease Academy of Sciences
tantunzi13@mails.ucas.ac.cn,
[2] Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, China

Let $\mathcal{H} = (V, H)$ be a hypergraph and $S \subseteq V$. A hyperedge h is contained in the part S if all the vertices of h is in the set of S and a hyperedge h is incident to the part S if at least one vertex of h is in the set of S. We write $e(S) := |\{h \in H, h \cap (V/S) = \emptyset\}|$ and $L(S) := |\{h \in H, h \cap S \neq \emptyset\}|$, where $e(S)$ is the number of hyperedges contained S and $L(S)$ is the number of hyperedges incident to S. The hyperdegree of a vertex equals the number of hyperedges incident to it and an r-uniform hypergraph has r vertices in each of its hyperedges. The hypergraph partitioning problem is to partition a hypergraph into smaller components satisfying specified constraints so as to minimize (or maximize) some objective functions. A judicious partitioning problem on a hypergraph is a problem in which one seeks a partition that optimizes several quantities simultaneously [1], such as minimizing $\max\{e(V_1), e(V_2), \ldots, e(V_K)\}$ or $\max\{L(V_1), L(V_2), \ldots, L(V_K)\}$. The judicious partition of hypergraphs aiming at minimizing $\max\{ L(V_1), L(V_2), \ldots, L(V_K)\}$ is a NP-hard problem [1]. The paper interprets judicious partition of hypergraphs as an integer nonlinear programming problem and some connections between the maximal hyperdegree and the optimal solution have been analyzed.

Algorithm 1 Minimum k & d algorithm

Require: $E = \{e_1, e_2, \ldots, e_N\}$ where $e_i \subseteq \{H_1, H_2, \ldots, H_M\}$ standing for the hyperedges where the vertex v_i is in and all the possible (CM+d)-sets of $\{H_1, H_2, \ldots, H_M\}$: $T = \{t_1, t_2, \ldots, t_{\binom{M}{CM+d}}\}$.
Ensure: k
1: Generate $S = \{S_1, S_2, \ldots, S_Q\}$, where $S_j = \{e_i | e_i \subseteq t_j, i \in \{1, 2, \ldots, N\}\}$, $j = 1, 2, \ldots, Q$
2: Find the minimum set covering of E in S.
3: Output the minimum number of the covering sets: k .

We analyze a sub-problem of judicious partition of hypergraphs and a general algorithm (Algorithm 1) for solving the problem has been proposed. Given a hypergraph $\mathcal{H}$=(V,H) with maximum hyperdegree CM, where $0 < C \leq 1$ and M is the number of hyperedges, find the minimum k for the partition, so that the objective function value of judicious partition of $\mathcal{H}$ is at most $CM+d$, where $0 \le d \le (1-C)M$ and d is a given integer. In the step 2 of the "Minimum k & d algorithm", minimum set covering of E should be found in S. A greedy algorithm have been applied to solve this problem, which is an LN factor approximation algorithm for the minimum set cover problem, where $LN = 1 + \frac{1}{2} + \cdots + \frac{1}{N}$ [2].

The judicious partition of hypergraphs can be regarded as several sub-problems mentioned above. Each time an objective value $CM+d$ is given, where d ranges from 0 to $M-CM$, k is generated by the "Minimum k & d algorithm". The first time the constraint: $k \le K$ is meet, the $CM+d$ is the objective value we found for the problem. In the Minimum k & d algorithm, Q can be as large as $\binom{M}{CM}$,

Algorithm 2 Judicious partition of hypergraphs

Require: K, the correlation between $(CM+d)$-sets and $(CM+d+1)$-sets, $0 \le d \le (M-CM)$.
1: Initial $minmaxL \leftarrow CM$, $E = \{e_1, e_2, \ldots, e_N\}$.
2: **for** d$\leftarrow$ 0 to $(1-C) * M$ **do**
3: Run the minimum k & d algorithm with E and $T = \{t_1, t_2, \ldots, t_{\binom{M}{CM+d}}\}$, then k, $\{V_1, V_2, \ldots, V_k\}$ and $S^* = \{S_1^*, S_2^*, \ldots, S_k^*\}$ are generated.
4: **if** $k > K$ **then**
5: $E = S^*$.
6: **break**
7: **else**
8: $minmaxL$=$CM+d$.
9: **return**
10: **end if**
11: **end for**
12: Output: $minmaxL$, $\{V_1, V_2, \ldots, V_k\}$.

but Q will not be $\binom{M}{CM+d}$ in the Minimum k & d algorithm, $1 \le d \le (M-CM)$. Since the correlation between $(CM+d)$-sets and $(CM+d+1)$-sets is given, $0 \le d \le (M-CM)$, and E is replaced by the k $(CM+d)$-sets in the step 5 of Algorithm 3 if an optimal value is not found, less than $k(M-(CM+d))$ $(CM+d+1)$-sets will be chosen as S in the next stage. From $k \le N$, we can obtain that $Q \le N(M-(CM+d))$, $1 \le d \le (M-CM)$.

Theorem 1. If the Algorithm 3 is a factor-α approximation algorithm, the α is at least $\frac{1}{M}\frac{logK-logLN}{1-logC}$.

References

1. Y. Zhang, Y. C. Tang, G. Y. Yan, On judicious partitions of hypergraphs with edges of size at most 3, European Journal of Combinatorics 49 (2015) 232–239.
2. V. V. Vazirani, Approximation algorithms, Springer Science & Business Media, 2013.

Tangle and Ultrafilter: Game Theoretic Interpretation

Takaaki Fujita (Gunma University) Koichi Yamazaki (Gunma University)

Abstract: Tangle is a dual notion of a well studied graph (matroid) parameter branch-width. Ultrafilter is an important notion which plays a foundational role in set, model, and topology theories. In this paper, we give a game-theoretic interpretation of relation between tangle and ultrafilter.

1 Preliminary

Branch-width is a well-studied graph parameter, and it can be generalized to connectivity systems [1].

Definition 1 *Let X be an underlying set, f a symmetric submodular function on $2^X \to \mathsf{N}$. A pair (T, μ) is a branch decomposition tree of (X, f) if T is a ternary tree such that $|L(T)| = |X|$ and μ is a bijection from X to $L(T)$, where $L(T)$ denotes the leaves in T. For each $e \in E(T)$, we denote $f(\bigcup_{v \in L(T_1)} \mu^{-1}(v))$ by $bw(T, \mu, e)$, where T_1 is a tree obtained by deleting e from T, note here the symmetry property of f. The width of (T, μ) is defined by $\max_{e \in E(T)} bw(T, \mu, e)$. The branch-width of (X, f), denoted by $bw(X, f)$, is defined as the minimum width over all possible branch decomposition trees of (X, f).*

Tangle was first introduced in [3] as a dual notion of branch-width and it was extended to connectivity systems [1].

Definition 2 *Let X be an underlying set, f a symmetric submodular function on $2^X \to \mathsf{N}$, and an integer k. A family $\mathscr{T} \subseteq 2^X$ is a tangle of order $k+1$ on (X, f) if $\mathscr{T}$ satisfies the following axioms.*

(TB) $\forall A \in \mathscr{T},\ f(A) \le k$,
(TE) $A \subseteq X, f(A) \le k \Rightarrow$ *either* $(A \in \mathscr{T})$ *or* $(X \setminus A \in \mathscr{T})$,
(TC) $A, B, C \in \mathscr{T} \Rightarrow A \cup B \cup C \ne X$,
(TL) $\forall e \in X,\ X \setminus \{e\} \notin \mathscr{T}$.

Theorem 1 ([3]) *There is no tangle order $k+1$ on (X, f) if and only if the branch-width of (X, f) is at most k.*

In this paper, we generalise an ultrafilter on a set X into a connectivity system (X, f).

Definition 3 *Let X be an underlying set, f a symmetric submodular function on $2^X \to \mathsf{N}$, and an integer k. A family $\mathscr{U} \subseteq 2^X$ is a ultrafilter of order $k+1$ on (X, f) if $\mathscr{U}$ satisfies the following axioms.*

(FB) $A \in \mathscr{U} \implies f(A) \le k$,
(FI) $A, B \in \mathscr{U},\ f(A \cap B) \le k \implies A \cap B \in \mathscr{U}$,
(FH) $A \in \mathscr{U},\ A \subseteq B \subseteq X,\ f(B) \le k \implies B \in \mathscr{U}$,
(FW) $\emptyset \notin \mathscr{U}$.
(TE) $A \subseteq X, f(A) \le k \Rightarrow$ *either* $(A \in \mathscr{U})$ *or* $(X \setminus A \in \mathscr{U})$.

Additionally, we will consider the following an extra axiom.

(FN') $[(e \in X) \wedge (f(\{e\}) \le k)] \implies X \setminus \{e\} \in \mathscr{U}$.

Then, we have the following cryptomorphism between ultrafilter on (X, f) and co-tangle on (X, f), where co-tangle is a complementary concept of tangle.

Theorem 2 *Let X be a finite underlying set and f a symmetric submodular function on 2^X. Then, $\mathscr{U}$ is a co-tangle with order k of (X, f) if and only if $\mathscr{U}$ is an ultrafilter of order k on (X, f) satisfying the additional axiom (FN').*

1.1 Trivial case: Principal filter

In this study, we are interested in the existence of ultrafilter on (X, f) of order k for given X, f, and k. Consider a case in which there exists an element $e \in X$ such that $f(\{e\}) > k$. Then, $\mathscr{U}_P := \{A \subseteq X : e \in A, f(A) \le k\}$ is an ultrafilter of order $k+1$. Such $\mathscr{U}_P$ is called *principal (ultra)filter*. The trivial case is not of interest in this study. In order to avoid the trivial case, we assume the following assumption (AS):

(AS) $\forall e \in X,\ f(\{e\}) \le k$.

1.2 Non-trivial case: Non-principal filter

As mentioned above, we are not interested in principal ultrafilter, namely we are interested in non-principal ultrafilter, especially *free ultrafilter*. An ultrafilter $\mathscr{U}$ (on both X and (X, f)) satisfying the following axiom (FF) is called *free ultrafilter*.

(FF) $\bigcap_{A \in \mathscr{U}} A = \emptyset$.

Notice that $\mathscr{U}_P$ in Subsection 1.1 is not free. A well-known example of free ultrafilter on X is *Fréchet filter*: For an infinite set X, $\mathscr{U}_F := \{X \setminus Y : Y \subseteq X \text{ is a finite set}\}$. It is known that no free ultrafilter on X can exist *when X is finite*. However, quite interestingly, there does exist a free ultrafilter on (X, f) even when X is finite, which we will explain in this paper.

1.3 Construction scheme for free filter

The construction method of Fréchet filter can be formalized as follows.

- $\mathscr{S} := \{A \subseteq X : A \text{ is a finite set}\}$, $\bigcup_{A \in \mathscr{S}} A = X$,
- $\forall A \in \mathscr{S}, X \setminus A \in \mathscr{S}$.

With this in mind, let us consider the following scheme, which we will refer to as the *construction scheme for free filter* ($\mathsf{CS\text{-}FF}$). Intuitively, $\mathsf{CS\text{-}FF}$ makes a filter free.

Construction scheme for free filter ($\mathsf{CS\text{-}FF}$):

- $\mathscr{S} := \{A : A \text{ is a } \textit{small} \text{ set }\}$, $\bigcup_{A \in \mathscr{S}} A = X$,
- $\forall A \in \mathscr{S}, X \setminus A \in \mathscr{S}$.

It is significant to set $\mathscr{S}$ to be consisting of small sets. Because, in doing so, we can naturally derive the *pairwise intersection property* as shown below. As we will see later, for a finite underlying set X, we will apply the scheme to a ultrafilter on (X, f) rather than on X. There is an interesting relation between the pairwise intersection property and $\mathsf{CS\text{-}FF}$. We explain the relation through an example. It is obvious that if a family $\mathscr{F}$ satisfies the *finite intersection property*, then $\mathscr{F}$ also satisfies the *pairwise intersection property*. However, the converse is not true in general. Indeed, the following is a counter-example to this.

Example 1 (Cf., Linkedness in [2]) *Let X be a finite set and let $\mathscr{S} := \{A : |A| < |X|/2\}$. Then, $\mathscr{F} := \{A : X \setminus A \in \mathscr{S}\}$ satisfies the pairwise intersection property, and it does not satisfy the finite intersection property. (Hence $\mathscr{F}$ is not a filter.)*

It is worth mentioning that the linkedness is deeply related to branch-width. It should be emphasized that $\mathscr{F}$ in the example is constructed in accordance with CS-FF, although $\mathscr{F}$ is not a filter.

1.4 Ultrafilter interpretation of (FN')

Let us now consider the meaning of the extra axiom (FN'). Essentially, (FN') corresponds to CS-FF. In fact, (FN') is conformed with the CS-FF by setting $\mathscr{S} := \{\{e\} : e \in X\}$. Thus, intuitively, (FN') makes an ultrafilter free as well as CS-FF. By combining the assumption (AS) in Subsection 1.1 and the extra axiom (FN'), we have the following axiom (FN):

(FN) $\forall e \in X,\ X \setminus \{e\} \in \mathscr{U}$.

Since an ultrafilter on (X, f) is a co-tangle, (FN) is essentially the same as the axiom (TL) in the definition of tangle.

1.5 Compass mechanism

The following theorem is known as a folklore.

Theorem 3 *Given an ultrafilter $\mathscr{U}$ on a finite underlying set X and a partition $\mathscr{P} = \{A_1, \ldots, A_n\}$ of X, exactly one of the blocks in $\mathscr{P}$ belongs to $\mathscr{U}$.*

We will refer to the property stated in Theorem 3 as *compass mechanism.* A known application of the compass mechanism is the Ramsey's theorem for infinite graphs. The theorem can be generalized as follows.

Lemma 1 *Given a partition $\mathscr{P} := \{A_1, \ldots, A_n\}$ $(n \geq 3)$ of X such that $f(A_i) \leq k$ for each $1 \leq i \leq n$, and an ultrafilter $\mathscr{U}$ with order $k+1$ on (X, f), exactly one of the blocks in $\mathscr{P}$ belongs to $\mathscr{U}$ unless there exists a partial branch decomposition tree (T, μ) of width at most k such that each block in $\mathscr{P}$ is associated with a leaf of T.*

Corollary 1 *Given a partition $\mathscr{P} := \{A_1, A_2, A_3\}$ of three blocks and an ultrafilter $\mathscr{U}$ of order $k+1$ on (X, f), if $f(A_i) \leq k$ holds for each $1 \leq i \leq 3$, then exactly one of blocks in $\mathscr{P}$ belongs to $\mathscr{U}$.*

2 Game-theoretic interpretations

In this section, we introduce a game on (X, f), which we call *monotone search game.* The game is described in terms of robber and cops. By using the game, we demonstrate how (FN') works in the game. Then, we show that a branch decomposition tree of (X, f) is essentially the same as a cops' winning strategy (see Lemma 2). On the other hand, we also show that an ultrafilter on (X, f) corresponds to a robber's winning strategy for the game (see Lemma 3). Hence, the ultrafilter essentially the same as the robber's *haven.* We also explain how the compass mechanism works in a robber's winning strategy.

Monotone search game on (X, f) with k cops:

1. $i := 1$, $U_i := X$
2. If $|U_i| = 1$ and $f(U_i) \leq k$, then the cops win the game.
3. In order to corner the robber, the cops choose a set M_i such that $\emptyset \neq M_i \subset U_i$, $f(M_i) \leq k$, and $f(U_i \setminus M_i) \leq k$. If the cops cannot choose such a set, then the cops lose the game.
4. The robber must flee into M_i or $U_i \setminus M_i$. The set into which the robber flees is referred to as U_{i+1}.
5. $i := i + 1$ and go to Step 2.

We demonstrate how (FN') works under the assumption (AS). If there is an element $e \in X$ such that $f(\{e\}) > k$, then there is a winning strategy for the robber: The robber just chooses the set containing the element e. This trivial case corresponds to the case of principal filter. In the game, we assume (AS) in order to avoid the trivial case. Then, from (AS), we have (FN) by the axioms (TE) and (FN'), that is, $\{e\} \notin \mathscr{U}$ holds for each $e \in X$. This can be interpreted that every $\{e\}$ is not safe place for the robber. As a result, the Step 2 in the game can be replaced with the following:

2. If $|U_i| = 1$, then the cops win the game.

Lemma 2 *Let X be a finite underlying set and f a symmetric submodular function on X. For the monotone search game on (X, f) with k cops, there is a branch decomposition tree (T, μ) of width at most k on (X, f) if and only if there is a cops' winning strategy for the game.*

Lemma 3 *Let X be a finite underlying set and f a symmetric submodular function on 2^X. For the monotone search game on (X, f) with k cops, suppose that there is an ultrafilter $\mathscr{U}$ of order $k+1$ on (X, f), then there is a robber's winning strategy for the game.*

Now, we explain how the compass mechanism works in a robber's winning strategy. As we have seen, in the robber's winning strategy, an ultrafilter indicates which set the robber flees into. Namely, an ultrafilter works like a compass which is pointing the robber's haven $\{\overline{U_i}, M_i, U_i \setminus M_i\} \cap \mathscr{U}$ for a given partition $\{\overline{U_i}, M_i, U_i \setminus M_i\}$ by the cops. Corollary 1 indicates that the condition "$f(M_i) \leq k$ and $f(U_i \setminus M_i) \leq k$" described in Step 3 plays a key role in the game.

Acknowledgements This work was supported by JSPS KAKENHI Grant Number 15K00007.

References

[1] J. Geelen, B. Gerards, N. Robertson, and G. Whittle. Obstructions to branch-decomposition of matroids. *Journal of Combinatorial Theory, Series B*, 96(4):560–570, 2006.

[2] B. Reed. Tree width and tangles: A new connectivity measure and some applications. *Surveys in combinatorics*, 241:87–162, 1997.

[3] N. Robertson and P. Seymour. Graph minors. X. Obstructions to tree-decomposition. *Journal of Combinatorial Theory, Series B*, 52(2):153–190, 1991.

Linear-width and Single ideal

Takaaki Fujita (Gunma University) Koichi Yamazaki (Gunma University)

Abstract: Branch-width is a well studied graph parameter, and it was extended into parameter on a connectivity system. Linear-width is also a parameter on a connectivity system, which can be thought of as path version of branch-width. In this paper, we define a sort of ideal on a connectivity system which we call *single ideal* and show a duality between single ideal and linear-width. We also give a characterization of maximal single ideal.

1 Preliminary

In this paper, we introduce a new concept of ideal, which is strongly related to linear-width on a *connectivity system* (X, f), where X is an underlying set and f is a symmetric submodular function on X. The linear-width was originally introduced as a graph parameter related to well studied graph parameter branch-width [8]. Intuitively, linear-width is a path version of branch-width, i.e., the relation between linear-width and branch-width is like the relationship between path-width and tree-width. Branch-width was generalized into a parameter on connectivity systems [2, 4], so linear-width as well.

The notion of ideal can be found in several fields, such as ring theory (ring), set theory (boolean algebra), order theory (lattice, poset), and is defined in their different contexts. Ideals discussed in this paper are ones appeared in the context of set theory (see cf. [7]).

There is a dual notion of branch-width, which is called *tangle*. The notion of tangle was first introduced by Robertson and Seymour in [6] for (hyper)graphs and was then extended to matroids (connectivity systems) [2, 4], and it was shown that there is no tangle of (X, f) with order $k+1$ if and only if the branch-width of (X, f) is at most k. In addition, in [5], Oum and Seymour introduced a relaxed notion of tangle, which is called *loose tangle*, and showed that there is a tangle of order k if and only if there is a loose tangle of order k.

It has been shown recently that a loose tangle and a tangle can be considered as an ideal and a maximal ideal on a connectivity system, respectively [9]. Hence, it is natural to ask whether or not similar results can be obtained not for branch-width but for linear-width. In this paper, we introduce a new concept of ideal, which we call *single ideal*, then we show that there is no single ideal of (X, f) with order $k+1$ if and only if the linear-width of (X, f) is at most k. Similar results to ours can be found in [1, 3]. The difference from them is that our approach is based on the concept of ideal.

2 Definitions and known results

In this section, we give definitions and known results related to tangle and ideal which we will need in this paper.

A function $f : 2^X \to \mathsf{N}$ is *symmetric submodular* if f satisfies (1) symmetry property: $\forall A \subseteq X$, $f(A) = f(\overline{A})$, and (2) submodularity property: $\forall A, B \subseteq X$, $f(A) + f(B) \geq f(A \cap B) + f(A \cup B)$. We call the pair (X, f) *connectivity system*. For an integer $k > 0$, a set $A \subset X$ is *k small* (or simply small) if $f(A) \leq k$ holds.

Definition 1 (see cf., [7]) *Let X be an underlying set. A collection $\mathscr{I} \subseteq 2^X$ is an ideal on boolean algebra $(2^X, \cap, \cup)$ (or we simply write X) if $\mathscr{I}$ satisfies the following axioms:*

(IH) $B \in \mathscr{I}$, $A \subseteq B \implies A \in \mathscr{I}$.
(IU) $A, B \in \mathscr{I} \implies A \cup B \in \mathscr{I}$.
(IW) $X \notin \mathscr{I}$.

An ideal $\mathscr{I}$ on boolean algebra is *maximal* if there is no $\mathscr{I}'$ properly containing $\mathscr{I}$. It is known that an ideal $\mathscr{I}$ with an underlying set X is maximal iff $\mathscr{I}$ satisfies the following additional axiom:

(IE) $\forall A \subseteq X$, either $(A \in \mathscr{I})$ or $(\overline{A} \in \mathscr{I})$.

In Definition 1, we give the definition of ideal on *boolean algebra*. In Definition 2, we introduce a new concept which is modeled on the ideal on boolean algebra but is based on connectivity system rather than on boolean algebra.

Definition 2 *Let (X, f) be a connectivity system and $k > 0$ an integer. A collection $\mathscr{S} \subseteq 2^X$ is a single ideal on (X, f) of order $k+1$ if $\mathscr{S}$ satisfies the following axioms:*

(SIB) $\forall A \in \mathscr{S}$, $f(A) \leq k$.
(SIH) $A, B \subseteq X$, $A \subset B$, $B \in \mathscr{S}$, *and* $f(A) \leq k \implies A \in \mathscr{S}$.
(SIS) $A \in \mathscr{S}$, $f(\{e\}) \leq k$, $f(A \cup \{e\}) \leq k \implies A \cup \{e\} \in \mathscr{S}$.
(SIN) $\forall e \in X$, $f(\{e\}) \leq k \implies \{e\} \in \mathscr{S}$.
(IW) $X \notin \mathscr{S}$.

We will refer to the family $\mathscr{S}$ as single ideal on the connectivity system (X, f) of order $k+1$.

There are two main differences between the definition of ideal on boolean algebra and that of single ideal on connectivity system. One is the restriction by the function f and the other is that B in (IU) is replaced with a singleton set $\{e\}$ of $f(\{e\}) \leq k$ in (SIS). To avoid a trivial case (in which the single ideal is essentially the same as a *principal ideal*), we include the axiom (SIN) in the definition. As is the case with the ideal on boolean algebra, a single ideal $\mathscr{I}$ is *maximal* (on connectivity system) if there is no $\mathscr{I}'$ properly containing $\mathscr{I}$.

2.1 Linear branch-decomposition

The following definition of linear-width is slightly different from the original definition. In our definition, we use ternary caterpillar to define the linear-width, because we want to adapt the linear-width to the axiom (SIS) (i.e., the check $f(\{e\}) \leq k$). In the study, we are not interested in the case that there is an element e such that $f(\{e\}) > k$. Thus, in this paper, we may assume that $f(\{e\}) \leq k$ holds for every $e \in X$. The assumption makes the difference between the definitions not substantial.

A tree C is *ternary caterpillar* if C consists of a path $(\ell_1, b_2, b_3, \ldots, b_{n-1}, \ell_n)$ and vertices $\ell_2, \ell_3, \ldots, \ell_{n-1}$ such that ℓ_i is only adjacent to b_i for each $2 \leq i \leq n-1$. The path $(b_2, \ldots, b_{n-1})$ is called *backbone* and the vertices ℓ_1 and ℓ_n are called *start vertex* and *end vertex*, respectively.

Definition 3 *A linear branch-decomposition of a connectivity system (X, f) is a ternary caterpillar C whose leaves are associated one-to-one with the elements of X. This can be generalized as follows: A partial linear branch-decomposition of (X, f) is a ternary caterpillar C in which the leaves are associated one-to-one with the blocks in a partition $(P, \{e_2\}, \ldots, \{e_{n-1}\}, Q)$ of X such that the start and end vertices are associated with the blocks the subsets P and Q, respectively, and each vertex b_i in the backbone $(b_2, \ldots, b_{n-1})$ of C is associated with the singleton set $\{e_i\}$. Hence, if P and Q are singleton sets, then we have a linear branch-decomposition. For a partition $(P, \overline{P})$ of X, a decomposition consisting of just start and end vertices associated with P and $\overline{P}$ can be also considered as a partial linear branch-decomposition. Moreover, the width of the partial linear branch-decomposition of C is defined by $\max\left\{f(P), f(Q), \max_{i=2}^{n-1} f(\{e_i\}), \max_{i=2}^{n-1} f(P \cup \{e_2, \cdots, e_i\})\right\}$. The linear-width of (X, f) is the minimum width overall linear branch-decompositions of (X, f).*

Definition 4 *Let C be a partial linear branch-decomposition of a connectivity system (X, f), P and Q the sets associated with the start and end vertices of C, respectively, and $\mathscr{F}$ a family of subsets of X. We say that C conforms to $\mathscr{F}$ if there are sets $P', Q' \in \mathscr{F}$ such that $P \subseteq P'$ and $Q \subseteq Q'$. A family $\mathscr{F}$ is called k-non-conforming (to (X, f)) if there is no partial linear branch-decomposition of width at most k that conforms to $\mathscr{F}$.*

3 Results

In this section, we first show, in Lemma 2, that there does exist a family which satisfies the axioms in the definition of single ideal. Each set in the family corresponds to *k-branched set* in terms of tangle (see for details [5]). In the proof of Lemma 2, we heavily use techniques developed in [5]. From the lemma (and Lemma 1), we have a duality between single ideal and linear-width (i.e., Theorem 1). On the basis of the existence of single ideal, we next show a characterization of a maximal single ideal in Theorem 2.

Let (X, f) be a connectivity system and let $\mathscr{B}^{(X,f)} := \{A \subseteq X : \exists (e_1, e_2, \ldots, e_\ell) \text{ s.t. } \bigcup_{i=1}^{\ell} \{e_i\} = A \text{ and } f(\{e_1, \ldots, e_j\}) \leq k \text{ for } \forall 1 \leq j \leq \ell\}$, and let $\mathscr{B}{\downarrow}^{(X,f)} := \{A : A \subseteq B, f(A) \leq k, B \in \mathscr{B}^{(X,f)}\}$. We write simply $\mathscr{B}$ and $\mathscr{B}{\downarrow}$ instead of $\mathscr{B}^{(X,f)}$ and $\mathscr{B}{\downarrow}^{(X,f)}$, when it is clear from context. Unfortunately, $\mathscr{B}^{(X,f)}$ is not closed under taking subsets. Thus, it is not obvious whether $\mathscr{B}{\downarrow}^{(X,f)}$ satisfies the axiom (SIH).

We now give a relation between $\mathscr{B}{\downarrow}^{(X,f)}$ and single ideal on a connectivity system (X, f).

Lemma 1 *Let (X, f) be a connectivity system. If the linear-width of (X, f) is at most k, then (X, f) does not allow any single ideal on (X, f) with order $k + 1$.*

Lemma 2 *Let (X, f) be a connectivity system. If the linear-width of (X, f) is at least $k+1$, then $\mathscr{B}{\downarrow}^{(X,f)}$ is a single ideal on (X, f) with order $k+1$. (Recall that we are assuming that $f(\{e\}) \leq k$ holds for every $e \in X$.)*

The following theorem can be obtained directly from the above two lemmas.

Theorem 1 *For a connectivity system (X, f), there is no single ideal on (X, f) with order $k + 1$ if and only if the linear-width of (X, f) is at most k. (Recall again that we are assuming that $f(\{e\}) \leq k$ holds for every $e \in X$.)*

As mentioned above, an ideal $\mathscr{I}$ on a boolean algebra is maximal if and only if $\mathscr{I}$ satisfies the axiom (IE). It would be natural to ask whether or not a similar characterization holds for a maximal single ideal on a connectivity system. We show below that a similar characterization holds. To show this, we need the following Lemma 3. In the proof of Lemma 3, the techniques developed in [6] are heavily used.

Lemma 3 *Let (X, f) be a connectivity system such that $f(\{e\}) \leq k$ holds for each $e \in X$ and the linear-width is more than k. Let $\mathscr{M}$ be a maximal family with respect to:*

(C1) satisfying the axioms (SIB) and (SIN), and
(C2) being k-non-conforming to (X, f).

Then, $\mathscr{M}$ is a single ideal on (X, f) with order $k + 1$.

From Lemmas 1 and 3, we have the following theorem.

Theorem 2 *For a connectivity system (X, f), the linear-width of (X, f) is at least $k+1$ if and only if there is a single ideal on (X, f) with order $k + 1$ which satisfies the following additional axiom (SIE):*

(SIE) $A \subseteq X,\ f(A) \leq k \implies$ either $(A \in \mathscr{S})$ or $(\overline{A} \in \mathscr{S})$.

Acknowledgements This work was supported by JSPS KAKENHI Grant Number 15K00007.

References

[1] D. Bienstock, N. Robertson, P. Seymour, and R. Thomas. Quickly excluding a forest. *Journal of Combinatorial Theory, Series B*, 52(2):274–283, 1991.

[2] J. Dharmatilake. A min-max theorem using matroid separations. *Matroid Theory, Contemporary Mathematics*, 197:333–342, 1996.

[3] F. V. Fomin and D. M. Thilikos. On the monotonicity of games generated by symmetric submodular functions. *Discrete Applied Mathematics*, 131(2):323–335, 2003.

[4] J. Geelen, B. Gerards, N. Robertson, and G. Whittle. Obstructions to branch-decomposition of matroids. *Journal of Combinatorial Theory, Series B*, 96(4):560–570, 2006.

[5] S.-i. Oum and P. Seymour. Testing branch-width. *Journal of Combinatorial Theory, Series B*, 97(3):385–393, 2007.

[6] N. Robertson and P. D. Seymour. Graph minors. X. Obstructions to tree-decomposition. *Journal of Combinatorial Theory, Series B*, 52(2):153–190, 1991.

[7] R. Sikorski. *Boolean algebras.* Springer, Third edition, 1969.

[8] R. Thomas. Tree-decompositions of graphs (lecture notes). *School of Mathematics. Georgia Institute of Technology, Atlanta, Georgia.*

[9] K. Yamazaki. Tangle and maximal ideal. In *International Workshop on Algorithms and Computation*, pages 81–92. Springer, 2017.

Pascal-Like Triangles and Fibonacci-Like Sequences

Ryohei Miyadera[1] Masaru Kitagawa[1] Shota Suzuki[1] Yuki Tokuni[1]
Yushi Nakaya[2] Masanori Fukui[3]

[1]Kwansei Gakuin [2] Tohoku University [3] Hyogo University of Teacher Education

1. Pascal-Like Triangles and Fibonacci-Like Sequences

In [1], one of the authors demonstrated how Pascal-like triangles (triangles that look like Pascal's triangle) and Fibonacci-like sequences (sequences that look like Fibonacci sequences) arise from the probabilities associated with the various outcomes of a game of Definition 1 when $s = 1$ and demonstrated the existence of simple relationships between these Fibonacci-like sequences and the Fibonacci sequence itself. The following Definition 1 is the same as Definition 1 in [1] when $s = 1$. Originally, we studied this game as a form of a Russian roulette.

Definition 1. Let p, n, m and s be fixed positive integers with $m \leq n$. There are p players $\Theta_1, \Theta_2, ..., \Theta_p$ seated around a circular table, and the game starts with player Θ_1. Proceeding in order, a box containing n identically sized cards is passed from hand to hand. All of these cards are white except for m of them, which are red. When a player receives the box, he or she draws out a card at random (i.e., the player cannot see inside the box) s times, and these cards are not returned to the box. Therefore, the number of cards in the box decreases in each round. In this way, Player Θ_1 picks up a card in the first,...,s-th rounds, and Player Θ_2 picks up a card in the $s+1$-th,...,$2s$-th rounds where by 'in the y-th rounds' we mean 'upon the selection of the y-th card from the box.' The first player to draw a red card loses the game, and the game then ends.

This game is mathematically the same as a Russian roulette game in which p players take turns and shoot themselves. To calculate the probability of the game of Definition 1, it is easier to use the data structure of Russian roulette. We suppose that cards are arranged in a cylinder-like component into which n cards are placed. First, the card on the far left is to be picked up, and the last card to be picked up is on the far right. Then we calculate probability by calculating the number of combinations of red cards.

Definition 2. Let $f(p, n, m, s, v)$ be the probability that the v-th player loses in the game of Definition 1.

In [1], it was shown that the set $\{f(p, n, m, s, v) : m \leq n, n = 1, 2, ...\}$ has a pattern similar to Pascal's triangle for fixed positive integers p, v and $s = 1$. As an illustrative example, the Pascal-like triangle formed from $\{f(4, n, m, 1, 1) : 1 \leq m \leq n, n = 1, 2, ..., 6, 7\}$ is shown in Figure 1. For example, note that $f(4, 6, 2, 1, 1) = \frac{6}{15}, f(4, 6, 3, 1, 1) = \frac{10}{20}$, $f(4, 7, 3, 1, 1) = \frac{16}{35}$. Observe that $6 + 10 = 16$ and $15 + 20 = 35$. As shown in Figure 1, the denominators and numerators of the fractions form Pascal-like triangles.

Figure 1.

$$\frac{1}{1}$$
$$\frac{1}{2}, \frac{1}{1}$$
$$\frac{1}{3}, \frac{2}{3}, \frac{1}{1}$$
$$\frac{1}{4}, \frac{3}{6}, \frac{3}{4}, \frac{1}{1}$$
$$\frac{2}{5}, \frac{4}{10}, \frac{6}{10}, \frac{4}{5}, \frac{1}{1}$$
$$\frac{2}{6}, \mathbf{\frac{6}{15}}, \mathbf{\frac{10}{20}}, \frac{10}{15}, \frac{5}{6}, \frac{1}{1}$$
$$\frac{2}{7}, \frac{8}{21}, \mathbf{\frac{16}{35}}, \frac{20}{35}, \frac{15}{21}, \frac{6}{7}, \frac{1}{1}$$

Figure 2.

1
1,1
1,2,1
1,3,3,1
2,4,6,4,1
2,6,10,10,5,1
2,8,16,20,15,6,1

Numbers in Figure 2 are the numerators of the fractions in Figure 1. It is well-known that the numbers on the diagonals of Pascal's triangle add to the Fibonacci sequence, but the numbers on the diagonals of the triangle in Figure 2 add to Fibonacci-like sequences. Let b_n be the sequence made in this way. Then, $b_1 = 1, b_2 = 1, b_3 = 1+1 = 2, b_4 = 1+2 = 3, b_5 = 2+3+1 = 6, b_6 = 2+4+3 = 9, b_7 = 2+6+6+1 = 15, ...$ It is clear that the rule of this sequence is $b_n = b_{n-1}+b_{n-2}+1$ when $n = 1 \pmod 4$, and $b_n = b_{n-1}+b_{n-2}$ when $n \neq 1 \pmod 4$. The relation of this sequence to the Fibonacci sequence is also clear. $b_{4n} = F(2n)F(2n+2) = F(2n+1)^2 - 1$, $b_{4n+1} = F(2n+1)F(2n+2)$,$b_{4n+2} = F(2n+2)^2$ and $b_{4n+3} = F(2n+2)F(2n+3)$ where $F(n)$ is the Fibonacci sequence.

Next, we generalize these results.

Definition 3. We denote by $U(p,n,m,s,v)$ the number of combinations of positions of cards for which the $v-$th player loses the game of Definition 1.

We generalize the game of Definition 1 and obtain the following game of Definition 4.

Definition 4. Let p,n and m be fixed positive integers with $m \leq n$. There are p players $\Theta_1, \Theta_2, ..., \Theta_p$. All of these cards are white except for m of them, which are red. Player Θ_1 picks up in the $\theta_{1,1}$th, $\theta_{1,2}$th, $\theta_{1,3}$th,..., and θ_{1,s_1}th rounds where $\theta_{1,1} < \theta_{1,2} < \theta_{1,3} < \cdots < \theta_{1,s_1}$. Θ_2 picks up in the $\theta_{2,1}$th, $\theta_{2,2}$th, $\theta_{2,3}$th,..., and θ_{2,s_2}th rounds where $\theta_{2,1} < \theta_{2,2} < \theta_{2,3}, ..., < \theta_{2,s_2}$. Finally, Player Θ_p picks up in the $\theta_{p,1}$th, $\theta_{p,2}$th, $\theta_{p,3}$th,..., and θ_{p,s_p}th rounds where $\theta_{p,1} < \theta_{p,2} < \theta_{p,3}, ..., < \theta_{p,s_p}$. The first player to draw a red card loses the game, and the game then ends. Here, we assume that $\bigcup_{v=1}^{p}\{\theta_{v,t}, t = 1,2,...,s_v\} = \{1,2,3,...,n\}$ and $\{\theta_{v,t}, t = 1,2,...,s_v\} \cap \{\theta_{w,t}, t = 1,2,...,s_w\} = \emptyset$ for any natural numbers v,w such that $v \neq w$. This condition guarantees that only one player plays in each round.

Definition 5. We denote by $U_g(p,n,m,v)$ and $f_g(p,n,m,v)$ the number of combinations of positions of cards and the probability that Player Θ_v loses the game of Definition 4. Since Definition 4 does not contain a variable s, $U_g(p,n,m,v)$ and $f(p,n,m,v)$ do not contain s.

Theorem 1. For natural numbers p,n,m,s,v such that $n \leq m$ and $v \leq s$, $U_g(p,n,m,v) = U_g(p,n-1,m,v) + U_g(p,n-1,m-1,v)$.

Theorem 2. $\{f_g(p,n,m,s,v) : m \leq n, n = 1,2,...\}$ has a pattern similar to Pascal's triangle.

Proof. Since $f_g(p,n,m,s,v) = U_g(p,n,m,v)/\binom{n}{m}$, this arises directly from Theorem 1 and the properties of ${}_nC_m$. □

We generalize the sequence b_n and define $B_{p,s}(n), n = 1,2,3,...$ in Definition 6.

Definition 6. Let $B_{p,s}(n) = \sum_{k=0}^{\lfloor\frac{n-1}{2}\rfloor} U(p,n-k,k+1,s,1)$.

Theorem 3.

$$B_{p,s}(n) = B_{p,s}(n-1) + B_{p,s}(n-2) + \begin{cases} 1 & (1 \leq n \leq s \pmod{ps}) \\ 0 & (n = 0 \text{ or } n \geq s+1 \pmod{ps})\,. \end{cases} \tag{1.1}$$

Here we compare the Fibonacci sequence $F(n)$ and $B_{p,s}(n)$.
(1) $F(n)$ is {1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610}.
(2) $B_{3,1}(n)$ is {1, 1, 2, 4, 6, 10, 17, 27, 44, 72, 116, 188, 305, 493, 798}.
(3) $B_{4,1}(n)$ is {1, 1, 2, 3, 6, 9, 15, 24, 40, 64, 104, 168, 273, 441, 714}.
(4) $B_{2,2}(n)$ is {1, 2, 3, 5, 9, 15, 24, 39, 64, 104, 168, 272, 441, 714, 1155 }.
(5) $B_{3,2}(n)$ is {1, 2, 3, 5, 8, 13, 22, 36, 58, 94, 152, 246, 399, 646, 1045 }.
$B_{3,1}(n) = \lfloor(F(n+2)/2)\rfloor$. (See A052952 of [2].) $B_{2,2}(n) = \lfloor((1+\sqrt{5})/2)^{n+3})/5\rfloor$. (See A097083 of [2].) It seems that $B_{2,2}(n) + (1 + i^n + (-1)^n + (-i)^n)/4 = B_{4,1}(n+1)$. $B_{3,2}(n)$ is $\lfloor\frac{F(n+4)}{4}\rfloor$. (See A097083 of [2].)

Here, we study a further generalization of the previous games. For simplicity, we assume that the number of players p who participate in the game is 2.

Definition 7. Here, the rules of the game are the same as the rules of Definition 1 except that $p = 2$ and the first player to collect two red cards loses the game.
We denote by $U_2(n,m)$ the number of combinations of positions of cards for which the first player loses this game.

Theorem 4. $U_2(n,m) = \sum_{k=1}^{\lfloor\frac{n-m+1}{2}\rfloor} k({}_{n-2k-1}C_{m-2}) + \sum_{k=1}^{\lfloor\frac{n-m+2}{2}\rfloor} k^2({}_{n-2k-1}C_{m-3})$.

Theorem 5.

$$U_2(n,m) + U_2(n,m+1) = U_2(n+1,m+1). \tag{1.2}$$

References

[1] Matsui, H., Minematsu, D., Yamauchi T., Miyadera, R.: Pascal-like triangles and Fibonacci-like sequences, Mathematical Gazette, pp.27–41, 2010.

[2] The On-Line Encyclopedia of Integer Sequences. http://www.research.att.com/˜njas/sequences/

Two-Dimensional Maya Game and Two-Dimensional Silver Dollar Game

Ryohei Miyadera[1] Yuusuke Sakamoto [1] Kazunari Mizuta [1] Ryoji Takano [1]

Kai Hirokawa [1] Masanori Fukui[2]

[1]Kwansei Gakuin [2]Hyogo University of Teacher Education

1. Corner the Two Rooks 1: a Two-Dimensional Maya Game

Here, we introduce the impartial game of "Corner the Two Rooks", a variant of the classical game of "Corner the Queen" which was studied in [1]. Instead of the queen used in Wythoff's game of nim, we use the two rooks of chess. This game can be considered as a two-dimensional Maya game. The set of $\mathcal{P}$ positions and the set of $\mathcal{N}$ positions can be described using a nim sum, but the mathematical structure of this game is different from that of traditional nim. The nim sum of some $\mathcal{P}$ positions are positive, and the nim sum of some $\mathcal{N}$ positions are zero. Next, we introduce a variant of "Corner the Two Rooks" that can be considered as a two-dimensional silver dollar game. Although the authors could not find a formula for $\mathcal{P}$ positions, they prove that this two-dimensional silver dollar game is mathematically the same as a variant of a two-dimensional Maya game. Let $Z_{\geq 0}$ be the set of non-negative integers. Let us break with chess traditions here and denote fields on the chessboard by pairs of numbers. The field in the upper left corner is denoted by $(0,0)$ and the others are denoted according to a Cartesian scheme: field (x,y) denotes x fields to the right followed by y fields downward (see Figure 1).

Definition 1. (i) We define "Corner the Two Rooks". Two rooks are placed on a chessboard of unbounded size, and two players take turns choosing one of the rooks and moving it. Rooks are to be moved to the left or upward vertically as far as desired. A rook may jump over another rook but not onto another. The first player who cannot make a valid move loses.
(ii) By (x,y,z,w) we denote the positions of the two rooks, where (x,y) is the position of one rook and (z,w) is the position of the other rook.
(iii) Let $\mathcal{E} = \{(0,0,1,0),(0,0,0,1),(1,0,0,0),(0,1,0,0)\}$, which consists of the end positions from which a rook cannot move to another position.

Remark 1. (i) By Definition 1, there is no difference between the two rooks, and hence the position (a,b,c,d) is the same as (c,d,a,b).
(ii) By Definition 1, any rule of the game is symmetric with respect to x and y for any position (x,y) of a rook. Therefore, a proof for the position (x,y,z,w) is sufficient proof of the same for (y,x,w,z).
(iii) By Definition 1, two rooks should not be at the same position, i.e., $(x,y) \neq (z,w)$ for any position (x,y,z,w).

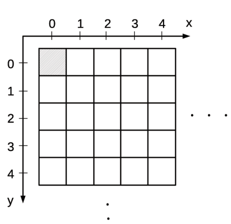


Figure 1: Definition of coordinates

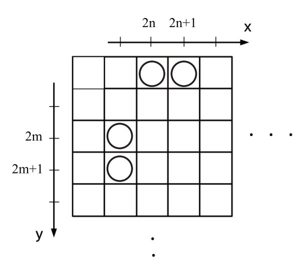


Figure 2: $\mathcal{P}_1$

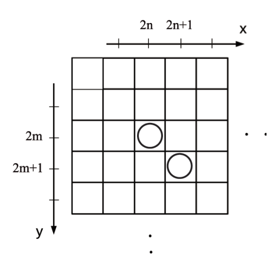


Figure 3: $\mathcal{N}_0$

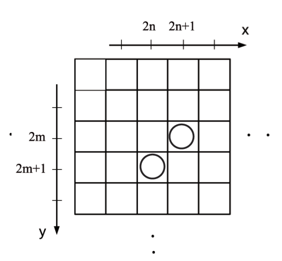


Figure 4: $\mathcal{N}_0$

Definition 2. We define $move((x,y,z,w))$ of "Corner the Two Rooks". $move((x,y,z,w))$ is the set of all positions that can be reached from (x,y,z,w). For any $(x,y,z,w) \notin \mathcal{E}$, let $move((x,y,z,w)) = \{(s,y,z,w):$

$0 \leq s < x$ and $s \in Z_{\geq 0}\} \cup \{(x,t,z,w) : 0 \leq t < y$ and $t \in Z_{\geq 0}\} \cup \{(x,y,u,w) : 0 \leq u < z$ and $u \in Z_{\geq 0}\} \cup \{(x,y,z,v) : 0 \leq v < w$ and $v \in Z_{\geq 0}\} - \{(x,y,x,y),(z,w,z,w)\}$.

Remark 2. In Definition 2, $move((x,y,z,w))$ does not contain $\{(x,y,x,y),(z,w,z,w)\}$ since a rook should not jump onto another rook.

As "Corner the Two Rooks" is an impartial game without draws, there are only two outcome classes:

Definition 3. (a) $\mathcal{N}$ positions, from which the next player can force a win as long as he plays correctly at every stage.
(b) $\mathcal{P}$ positions, from which the previous player (the player who will play after the next player) can force a win as long as he plays correctly at every stage.

For the detailed theory of combinatorial games, see [2].

Definition 4. For $x, y, z \in Z_{\geq 0}$, we let

$$\mathcal{P}_1 = \{(2n,m,2n+1,m) : n,m \in Z_{\geq 0}\} \cup \{(2n+1,m,2n,m) : n,m \in Z_{\geq 0}\}$$
$$\cup \{(n,2m+1,n,2m) : n,m \in Z_{\geq 0}\} \cup \{(n,2m,n,2m+1) : n,m \in Z_{\geq 0}\}, \tag{1.1}$$
$$\mathcal{N}_0 = \{(2n,2m,2n+1,2m+1) : n,m \in Z_{\geq 0}\} \cup \{(2n+1,2m+1,2n,2m) : n,m \in Z_{\geq 0}\}$$
$$\cup \{(2n+1,2m,2n,2m+1) : n,m \in Z_{\geq 0}\} \cup \{(2n,2m+1,2n+1,2m) : n,m \in Z_{\geq 0}\}, \tag{1.2}$$
$$\mathcal{P} = (\{(x,y,z,w) : x \oplus y \oplus z \oplus w = 0 \text{ and } x,y,z,w \in Z_{\geq 0}\} \cup \mathcal{P}_1) - \mathcal{N}_0 \tag{1.3}$$
and
$$\mathcal{N} = \{(x,y,z,w) : x \oplus y \oplus z \oplus w \neq 0 \text{ and } x,y,z,w \in Z_{\geq 0}\} \cup \mathcal{N}_0 - \mathcal{P}_1. \tag{1.4}$$

Example 1. For the example of elements belonging to the set $\mathcal{P}_1$, see Figure 2. Figures 3 and 4 present examples of elements of the set $\mathcal{N}_0$.

Theorem 1. $\mathcal{P}$ and $\mathcal{N}$ are the sets of $\mathcal{P}$ positions and $\mathcal{N}$ positions, respectively.

Remark 3. By Theorem 1, a two-dimensional Maya game has a simple formula for $\mathcal{P}$ positions when two rooks are used. If we use three rooks, there seems to be no simple formula for $\mathcal{P}$ positions.

2. Corner the Two Rooks 2: a Two-Dimensional Silver Dollar Game

Here we introduce the impartial game of "Corner the Two Rooks 2", which is considered to be a two-dimensional silver dollar game.

Definition 5. We define "Corner the Two Rooks 2". The rules are the same as the rules of the game of Definition 1 except that a rook cannot jump over another rook.

Next, we introduce a new game that is a variant of the game in Definition 1.

Definition 6. We define "Corner the Two Rooks 3". The rules are the same as in Definition 1 except for the following rules (i) and (ii):
(i) If Rook A is in the position (x,y) and Rook B is in the position $(x,y+1)$, then Rook B cannot move to the position $(x,y-1)$.
(ii) If Rook A is in the position (x,y) and Rook B is in the position $(x+1,y)$, then Rook B cannot move to the position $(x-1,y)$.

Lemma 1. The set of $\mathcal{P}$ positions of the game of Definition 5 is the same as the set of $\mathcal{P}$ positions of the game of Definition 6.

Remark 4. The authors have not discovered any formula for $\mathcal{P}$ positions of the game of Definition 5.

References

[1] W.A.Wythoff: A modification of the game of Nim, *Nieuw Arch. Wiskd.* **7** pp.199–202 (1907).

[2] M.H. Albert, R.J. Nowakowski, and D. Wolfe, *Lessons In Play*, A K Peters/CRC Press (2007).

3D-puzzles, as an application of infinite A_n hyperspace partition

Ikuro Sato[1] and Hiroshi Nakagawa[2]

2017 JCDCG3, Aug. 29-Sept. 1

1. Miyagi Cancer Center
sato-ik510@miyagi-pho.jp
2. Gallery of Wooden Polyhedra
woodenpolyhedra@gmail.com

Extended Abstract

To find out space-fillers of arbitrary n-dimensional hyperspace, there exist two approaches; (a) polytope first and (b) lattice first. The latter provides us broad and profound geometrical problems more than the former. Here, we will follow the latter. Among infinite reflection groups of Euclidean n-space A_n, B_n, C_n, D_n, E_6, E_7, E_8, F_4 and G_2, their fundamental regions are orthogonal simplices except A_n. In contrast, the fundamental region for A_n is isohedral simplex. For examples, the quadri-rectangular tetrahedron of Hill belongs to C_3 (Table 1-1), the isosceles tetrahedron of Sommerville to A_3 and equilateral triangle to A_2 (Table 1-2).

Sommerville tetrahedron, a 3-dimensional example of A_n, is composed of 4 congruent faces whose edge length are $2, \sqrt{3}\ and\ \sqrt{3}$. In 3-space, it splits into 4 equal Hill tetrahedra (Figure 1-1). But, generally, such a split-property is not held in hyperspaces. Sommerville tetrahedron is a self-replicative space-filler, i.e., reptile, and is well-known to embed into the prism with equilateral triangular base 3-periodically. Simultaneously its faces and edges embed respectively into the lateral surface of the prism 2-periodically and the ridge of the prism 1-periodically (Figure 1-2). Such multiple-duty properties are held in infinite A_n hyperspace partition.

Following [1, chapter 21, p. 462], we constructed space-filling isohedral n-simplex (n-Sommerville simplex abbreviated to Δ_n). Its facet, (n-2)-face, (n-3)-face and regular n-simplex are also abbreviated to F_n, G_n, H_n and α_n. F_n, G_n and H_n are not isohedral simplices but space-fillers. In Table 2 and 3, notice that the shortest edge length is $\sqrt{n}$, not a unit. After laborious calculation of their metric without computers, we can derive new embedding-properties of Δ_n as follows; At least in some direction, it is possible to

(1) embed Δ_n into Δ_{n-1} prism,
(2) embed F_n into F_{n-1} prism,
(3) embed G_n into G_{n-1} prism.
(4) embed H_n into H_{n-1} prism, and,
(5) k-face of Δ_n is asymptotically close to the fundamental simplex for C_k, when $n \to \infty$.

By applying these properties to 3-space, we can construct infinite family of tetrahedral space-fillers, embedding by different 2 ways into the prisms with equilateral or isosceles triangular bases (Figure 2). Nowadays, the determination of finite and infinite reflection groups has been completed [2-4]. However, even for the mathematicians、it is not easy to imagine its results and conclusions concretely. For such reasons, to visualize the above-mentioned models is worthful. In this talk, not only the infinite A_n hyperspace partition is shown but its available applications for the 3D-puzzles are presented.

References

[1] Conway J.H. and Sloane N.J.A.: "Sphere packings, lattices and groups", 3rd ed. Springer-Verlag, New York (1998).
[2] Coxeter, H.S.M.: "Regular polytopes", Dover Publications Inc., New York (1973).
[3] Coxeter, H.S.M.: Regular and semi-regular polytopes. III, *Mathematische Zeitschrift*, Vol. 200, 3-45 (1988).
[4] Coxeter, H.S.M.: Orthogonal trees, *Bull. ICA 3*. 83-91 (1991).

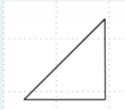
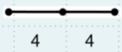
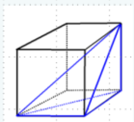
lattice	fundamental simplex	Coxeter-Dynkin diagram
C_2		
C_3		

Table 1-1: crystallographic C_n infinite reflection groups

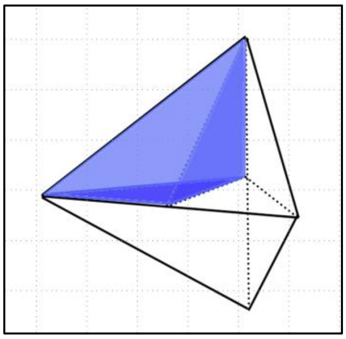


Figure 1-1: Hill and Sommerville

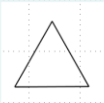
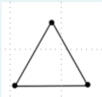
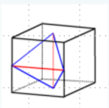
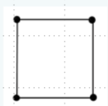
lattice	fundamental simplex	Coxeter-Dynkin diagram
A_2		
A_3		

Table 1-2: crystallographic A_n infinite reflection groups

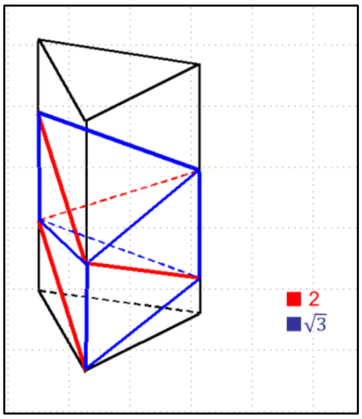


Figure 1-2: 3-periodic Δ_3 embedding into Δ_2-prism

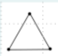
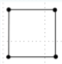
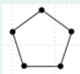
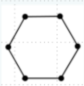
Δ_n	edge length	Coxeter-Dynkin diagram
Δ_2	$P_0P_1=P_1P_2=\sqrt{2}$ $P_0P_2=\sqrt{2}$	
Δ_3	$P_0P_1=P_1P_2=P_2P_3=\sqrt{3}$ $P_0P_2=P_1P_3=2$ $P_0P_3=\sqrt{3}$	
Δ_4	$P_0P_1=P_1P_2=P_2P_3=P_3P_4=2$ $P_0P_2=P_1P_3=P_2P_4=\sqrt{6}$ $P_0P_3=P_1P_4=\sqrt{6}$ $P_0P_4=2$	
Δ_5	$P_0P_1=P_1P_2=P_2P_3=P_3P_4=P_4P_5=\sqrt{5}$ $P_0P_2=P_1P_3=P_2P_4=P_3P_5=\sqrt{8}$ $P_0P_3=P_1P_4=P_2P_5=3$ $P_0P_4=P_1P_5=\sqrt{8}$ $P_0P_5=\sqrt{5}$	

Table 2: scaling of fundamental simplex for A_n

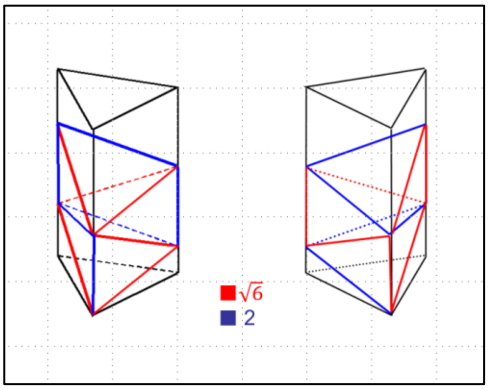


Figure 2:
two-way F4 embedding into F3-prism (left) and Δ2-prism (right)

metric	Δ_n	α_n
V^2	$(n+1)^{n-1}/(n!)^2$	$(n+1)n^n/2^n(n!)^2$
S^2	$2(n+1)^{n-2}/(n-1!)^2$	$n^n/2^{n-1}(n-1!)^2$
h^2	$(n+1)/2$	$(n+1)/2$
r^2	$1/2(n+1)$	$1/2(n+1)$
R^2	$n(n+2)/12$	$n^2/2(n+1)$
$R^2-(nr)^2$	$n(n-1)(n-2)/12(n+1)$	0

Table 3: metric of Δ_n and α_n

The Stable Roommates Problem with Unranked Entries

Hiroaki Suto[1,*] Aleksandar Shurbevski[2] Hiroshi Nagamochi[2]

The stable roommates problem (SR) and its extensions are one of the most well-studied problems of the area of matching under preferences [4]. Given a set of an even number N of participants, each of which has ranked all of the others in a strict order of preference, the SR asks to determine pairs of participants such that each participant is matched in a pair, and there do not exist two participants that are not matched with each other, yet prefer each other to their assigned partners. Given a matching M, a pair of unmatched participants that prefer each other to their partners matched by M is said to be a *blocking pair* of M, and a matching that has no blocking pair is said to be *stable*.

Among the various extensions of the SR the most common ones are the SR with incomplete preference lists (SRI), where "acceptable" pairs are limited by only allowing participants that have included each other in their preference lists to be matched together; and the SR with partially ordered preference lists (SRP), which introduces "indifference" in participants' preferences. In the context of indifference, more than one definition of stability are possible [3]:
- A matching is called *weakly stable* if there is no pair of unmatched participants that strictly prefer each other to their matched partners,
- A matching is called *super-stable* if no two unmatched participants prefer each other to their matched partners, or are indifferent between each other and their respective matched partners.

There exist linear time algorithms for the SR and the SRI which give a stable matching or correctly determine that none exists [2]. While the problem of finding a super-stable matching for an instance of the SRP (or determining that none exists) is solvable in polynomial time, the equivalent problem is NP-hard under weak stability [3]. Henceforth, we focus on weak stability and omit the quantifier "weak".

In light of the NP-hardness results from introducing indifference in the preference lists of the SR, we restrict the notion of indifference and introduce *unranked* entries, such that a participant is indifferent between an unranked and any other entry in her preference list. A pair of participants u and v such that u is unranked in v's preference list will never become a blocking pair since v is indifferent between u and any other participant in v's preference list. Likewise, if u and v are matched in a matching M, then no pair including v can become a blocking pair to M. We call this problem SRU, for stable roommates problem with unranked entries.

Theorem 1. *The SRU is NP-complete, even when*
- all pairs are acceptable,
- each participant has two unranked entries in her preference list, or is an unranked entry of three other participants, and
- each participant does not have unranked entries, or is herself not unranked for any of the other participants.

Proof sketch. We model an instance of the SRU by a bidirected graph $G = (V, E)$ where the vertex set V represents the set of participants, and the edge set E represents the set of acceptable pairs such that for a participant $v \in V$ the set of edges $E^+(v)$ with v as a tail gives the preference list of v. Edges are weighted by a function $\omega : E \to \{1, 2, \ldots, |V|\} \cup \{*\}$ such that for an unranked entry u on v's preference list, for the directed edge $e = (v, u)$ it holds that $\omega(e) = *$, and for two edges $e_1 = (v, u_1)$ and $e_2 = (v, u_2)$, $\omega(e_1) < \omega(e_2)$ means that v prefers u_1 to u_2. Note that according to our model of the SRU, for a participant v and two edges $e_1, e_2 \in E^+(v)$, $\omega(e_1) = \omega(e_2)$ holds if and only if $\omega(e_1) = \omega(e_2) = *$. It should be obvious how to relate the notions of matching and stability in terms of finding a "stable" matching as a set of pairwise non-incident edges in the graph G under the weight function ω. Let $E_* \subseteq E$ denote the set of unranked directed edges. A *base triangle* is defined to be a set $\{a_1, a_2, a_3\}$ of three vertices such that
- $\{\omega(e) \mid e \in E^+(a_2)\} = \{\omega(e) \mid e \in E^+(a_3)\} = \{1, 2, \ldots, N-1\}$,
- $E^+_*(a_1) = \{(a_1, b_i) \mid i = 2, 3, \ldots, k\}$ for some positive integer k,
- $\{\omega(e) \mid e \in E^+(a_1) - E_*\} = \{4, 5, \ldots, N-k-1\}$,
- $\omega(a_1, a_2) = \omega(a_2, a_1) = \omega(a_3, a_2) = 2$,
- $\omega(a_1, b_1) = \omega(a_2, a_3) = \omega(a_3, a_1) = 1$, and
- $\omega(a_1, a_3) = 3$,

as illustarted in Fig. 1. We call the vertex a_1 the *attaching vertex* of the base triangle $\{a_2, a_2, a_3\}$. Any stable matching in (G, ω) must contain the edge a_2a_3 and one of the edges a_1b_i, $i = 1, 2, \ldots, k$.

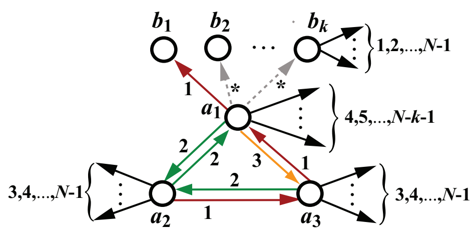


Figure 1: A base triangle $\{a_1, a_2, a_3\}$, where the weight ω is indicated by a number next to each edge, and dashed lines indicate unranked edges.

[1]Toyota Motor Corporation, suto.hiroaki@gmail.com
[2]Kyoto University, {shurbevski,nag}@amp.i.kyoto-u.ac.jp
* This work was done while the author was a graduate student at Kyoto University

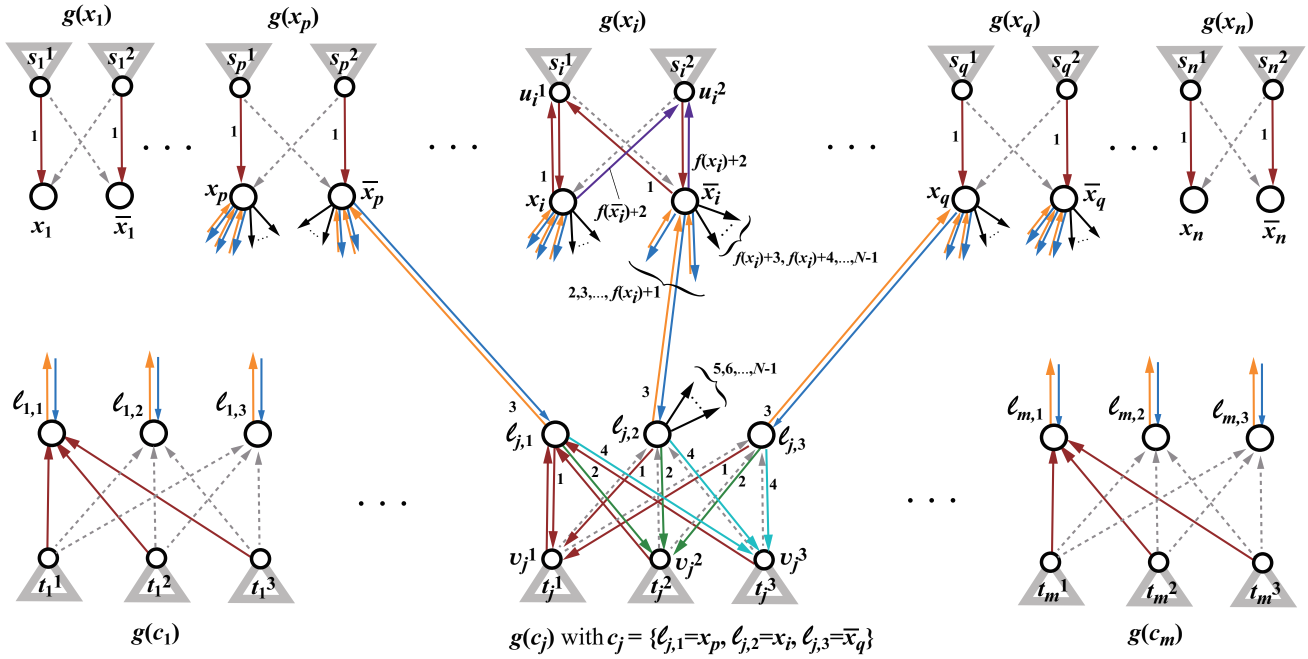


Figure 2: Illustration of an instance (G_I, ω) reduced from a 3SAT instance $I = (X = \{x_i \mid i = 1, 2, \ldots, n\}, C = \{c_j = \{\ell_{j,1}, \ell_{j,2}, \ell_{j,3}\} \mid j = 1, 2, \ldots, m\})$, where $N = |V(G_I)|$ is equal to $8n+12m$, dashed lines indicate unranked edges, and for a literal ℓ, $f(\ell)$ stands for the number of clauses in C containing this literal.

As the SRU is trivially in NP, we show its NP-hardness by a reduction from the well-known 3SAT [1]:
Instance: A set C of m clauses over a set X of n boolean variables such that each clause $c \in C$ has exactly three literals.
Question: Is there a truth assignment for X that satisfies all the clauses in C?

Given an instance $I = (X = \{x_i \mid i = 1, 2, \ldots, n\}, C = \{c_j = \{\ell_{j,1}, \ell_{j,2}, \ell_{j,3}\} \mid j = 1, 2, \ldots, m\})$ of 3SAT, we will construct an SRU instance (G_I, ω) with a complete digraph on $N = 8n + 12m$ vertices which consists of
- n *variable gadgets*, that is, 8-vertex graphs $g(x_1), g(x_2), \ldots, g(x_n)$;
- m *clause gadgets*, that is, 12-vertex graphs $g(c_1), g(c_2), \ldots, g(c_m)$; and
- the set of edges between these gadgets,
as illustrated in Fig. 2. The gadgets and edges between them are defined as follows.
• For each variable $x_i \in X$, define $g(x_i)$ to be a graph with eight vertices that consists of two base triangles s_i^1 and s_i^2 and two vertices, named x_i and $\overline{x}_i$ such that, for $h = 1$ (resp., $h = 2$), the attaching vertex u_i^h of the base triangle s_i^h has exactly one unranked edge $(u_i^1, \overline{x}_i)$ (resp., (u_i^2, x_i)) in $E^+(u_i^h)$.
• For each clause $c_j = \{\ell_{j,1}, \ell_{j,2}, \ell_{j,3}\} \in C$, define $g(c_i)$ to be a graph with 12 vertices that consists of three base triangles, t_j^1, t_j^2, and t_j^3, and three vertices, named $\ell_{j,1}$, $\ell_{j,2}$, and $\ell_{j,3}$, such that, for each $h = 1, 2, 3$, the attaching vertex v_j^h of the base triangle t_j^h has exactly two unranked edges, $(v_j^h, \ell_{j,2})$ and $(v_j^h, \ell_{j,3})$ in $E^+(v_j^h)$.

It can be shown that with a proper choice of the edge weight function ω, the graph (G_I, ω) has a stable matching if and only if the 3SAT instance $I = (X, C)$ has a satisfiable assignment. □

On the other hand, given an SRU instance with m acceptable pairs and k unranked entries in total, we can devise an $O(2^k m)$-time algorithm to find a stable matching or determine that none exists. This algorithm is based on an enumeration procedure which for each of the $O(2^k)$ subsets of pairs including an unranked entry checks if there exists a stable matching containing that subset by applying the $O(m)$-time algorithm due to Gusfield and Irving [2] for the SRI.

It is an interesting future research direction to investigate the nature of SR instances where indifference in participants' preferences are further restricted, and in particular, determine if there exists a particular class of SRU instances that are solvable in polynomial time. It seems possible that the SRU where unranked entries form a matching in the corresponding graph can be solved in polynomial time.

References

[1] M. R. Garey and D. S. Johnson, Computers and Intractability: A guide to the Theory of NP-completeness, Freeman, 1979.

[2] Dan Gusfield and Robert W. Irving, The Stable Marriage Problem Structure and Algorithms, The MIT Press, 1989.

[3] R. W. Irving and D. F. Manlove, "The stable roommates problem with ties," Journal of Algorithms 43, pp. 85–105, 2002.

[4] D. F. Manlove, Algorithmics of Matching Under Preferences, World Scientific, 2013.

An efficient algorithm for the stable marriage problem with short incomplete lists under social stability

KENT KOIDE [*] TAKAHITO KUNO [†] YOSHIO SANO [‡]

1 Introduction

The stable marriage problem was first considered by Gale and Shapley [5] in 1962, and several variants and extensions of the stable marriage problem have been studied (see [10]). The problem of finding the maximum size of a socially stable matching in an instance of the stable marriage problem with incomplete lists under social stability is called MAX SMISS, which is an NP-hard problem. The problem MAX SMISS with the condition that the length of the preference list of each man is at most two is called $(2,\infty)$-MAX SMISS. Askalidis *et al.* [2] gave an algorithm for solving $(2,\infty)$-MAX SMISS with time complexity $O(n^{\frac{3}{2}}\log n)$. In this paper, we improve the time complexity and give an algorithm for solving $(2,\infty)$-MAX SMISS with time complexity $O(n^{\frac{3}{2}})$.

2 Preliminaries

In this section, we give some definitions and terminologies used in this paper. An instance of SMI (Stable Marriage with Incomplete lists) is a tuple $I = (U, W, \{\succ_p\}_{p\in U\cup W})$ consisting of a set U of men, a set W of women, and each person's *preference list* which is a linear order on a subset of the opposite set representing his/her preference, i.e., $\succ_m$ is an ordering of women in a subset W_m of W for each man $m \in M$ and $\succ_w$ is an ordering of men in a subset M_w of M for each woman $w \in W$. We assume that U and W are disjoint and that U and W have the same cardinality n. The *length* of a preference list is the number of persons on it. For $m \in U$ and $w \in W$, we say that m is *acceptable* to w if $m \in U_w$ and that w is *acceptable* to m if $w \in W_m$. A pair (m,w) is said to be *acceptable* if m and w are acceptable to each other. Let A denote the set of all acceptable pairs in an instance I, i.e., $A = \{(m,w) \in U \times W \mid m \in U_w, w \in W_m\}$. We assume that $w \in W$ is acceptable to $m \in U$ if and only if m is acceptable to w since, otherwise, (m,w) can be neither matched nor a blocking pair for any stable matching. Let $M(\subseteq A)$ be a matching on the bipartite graph $(U,W;A)$. Let $M(p)$ denote the partner of a person p in a matching M. Note that $M(m) = w$ if and only if $M(w) = m$, $(m,w) \in M$. A *single person* in a matching M is a person who have no partner in M. Let $S \subseteq U \cup W$ denote the set of single persons in M. A *blocking pair* for a matching M in an instance I of SMI is a pair $(m,w) \in A \setminus M$ which satisfies one of the following four conditions: (i) $w \succ_m M(m)$ and $m \succ_w M(w)$; (ii) $w \succ_m M(m)$ and $w \in S$; (iii) $m \in S$ and $m \succ_w M(w)$; (iv) $m \in S$ and $w \in S$. A matching M is called a *stable matching* if there is no blocking pair for M.

An *instance* of SMISS (Stable Marriage with Incomplete lists under Social Stability) is a pair (I,G) of an instance I of SMI and a social network graph $G = (U \cup W, E)$. A *social blocking pair* for a matching M in an instance (I,G) of SMISS is a pair (m,w) such that (m,w) is a blocking pair for

[*]Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba, Ibaraki 305-8573, Japan. *Email:* `koide@syou.cs.tsukuba.ac.jp`

[†]Division of Information Engineering, Faculty of Engineering, Information and Systems, University of Tsukuba, Ibaraki 305-8573, Japan. *Email:* `takahito@cs.tsukuba.ac.jp`

[‡]Division of Information Engineering, Faculty of Engineering, Information and Systems, University of Tsukuba, Ibaraki 305-8573, Japan. *Email:* `sano@cs.tsukuba.ac.jp`

M in I and that $\{m,w\} \in E$. A matching M is called a *socially stable matching* if there is no social blocking pair for M. (Note that socially stable matchings are also called *locally stable matchings* in [4] and [9].) Given an instance (I,G) of SMISS, the problem of maximizing the cardinality of a socially stable matching M is called MAX SMISS. The problem MAX SMISS is known to be an NP-hard problem. The problem MAX SMISS with the condition that the length of the preference list of each man is at most two is called $(2,\infty)$-MAX SMISS, and is solved in polynomial time.

Theorem 1 ([2])**.** *The problem $(2,\infty)$-MAX SMISS is solvable in $O(n^{\frac{3}{2}} \log n)$.*

3 Main Result

In this section, we give an algorithm for solving $(2,\infty)$-MAX SMISS with time complexity $O(n^{\frac{3}{2}})$.

The time complexity of the algorithm given by Askalidis *et al.* [2] is dominated by a step of finding a minimum weight maximum matching in a weighted bipartite graph. The idea for improving the time complexity is as follows. Instead of finding a minimum weight maximum matching, we find a maximum matching on an unweighted bipartite graph in our algorithm. Then we make the obtained matching into a socially stable matching by using a method proposed in Iwama *et al.* [8]. The following is our main result.

Theorem 2. *The problem $(2,\infty)$-MAX SMISS is solvable in $O(n^{\frac{3}{2}})$.*

References

[1] G. Askalidis, N. Immorlica, A. Kwanashie, D. F. Manlove, and E. Pountourakis: Socially stable matchings in the hospitals/residents problem, in: F. Dehne, R. Solis-Oba, and J.-R. Sack (Eds.): *Algorithms and Data Structures*, WADS 2013, *Lecture Notes in Computer Science* **8037**, (2013) 85–96.

[2] G. Askalidis, N. Immorlica, A. Kwanashie, D. F. Manlove, and E. Pountourakis: Socially stable matchings in the Hospitals / Residents problem, *Preprint*, (2013) 23 pages. `http://arxiv.org/abs/1303.2041v3`

[3] G. Askalidis, N. Immorlica, and E. Pountourakis: Socially stable matchings, *Preprint*, (2013) 12 pages. `http://arxiv.org/abs/1302.3309`

[4] C. T. Cheng and E. McDermid: Maximum locally stable matchings, *Algorithms* **6** (2013) 383–395.

[5] D. Gale and L. S. Shapley: College admissions and the stability of marriage, *American Mathematical Monthly* **69** (1962) 9–15.

[6] J. E. Hopcroft and R. M. Karp: An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs, *SIAM Journal on Computing* **2** (1973) 225–231.

[7] R. W. Irving, D. F. Manlove, and G. O'Malley: Stable marriage with ties and bounded length preference lists, *Journal of Discrete Algorithms* **7** (2009) 213–219.

[8] K. Iwama, K., S. Miyazaki, and K. Okamoto: A $(2 - c\frac{\log N}{N})$-approximation algorithm for the stable marriage problem, *Algorithm Theory - SWAT 2004*, *Lecture Notes in Computer Science* **3111** (2004) 349–361.

[9] A. Kwanashie and D. F. Manlove: The Hospitals/Residents problem with free pairs, *Technical Report*, TR-2013-340, University of Glasgow, Department of Computing Science, (2013) 19 pages. `http://ftp.dcs.glasgow.ac.uk/publications/paperdetails.cfm?id=9399`

[10] D. F. Manlove: *Algorithms of Matching Under Preference*, World Scientific Publishing (2013).

Multi-agent Cooperative Patrolling of Designated Points on Graphs

Akitoshi Kawamura Hideaki Noshiro

1 The problem

In *patrolling* problems, one or more agents move around a given terrain to defend or supervise it by visiting designated places in the terrain with sufficient frequency [2].

Here we consider the problem where the terrain is represented by an undirected graph with specified edge lengths, on which agents move with speed 1 or less. Each vertex has an *idle time*, and is said to be *guarded* by a set of moving agents if it is visited at least once in every time period of that length. We are interested in deciding whether there is a schedule for a given number of agents to guard all vertices.

Since the problem for general graphs is NP-hard even for a single agent, we consider the following special graphs: *lines* (paths), *stars* and *unit-length cliques*. A star is a graph where there is a special vertex called the *centre*, which does not need to be guarded, and all other vertices are adjacent only to the centre. A unit-length clique is a complete graph with all edges equal in length. Since all that matters for patrolling is the time it takes to travel between each pair of nodes, a unit-length clique can be regarded as a special case of a star with all edges equal in length.

2 Previous work with no cooperation

Coene et al. [1] studied a slightly different version of the problem which require that each vertex be guarded already by one agent alone. In this setting, they showed polynomial time algorithms and NP-hardness results for some graph classes. We remove this requirement and study the problem where a vertex can be guarded by several agents in cooperation.

The two versions of the problem come to the same thing when there is only one agent. For this setting, Coene et al. [1] gave polynomial time algorithms for trees with uniform idle times and for lines, and proved that the problem is NP-hard for stars with arbitrary idle times. Thus, our study of the cooperating agents will focus on the cases where the problem is not already known to be NP-hard for single agent, that is, for (1) lines, (2) unit-length cliques, (3) stars with uniform idle times.

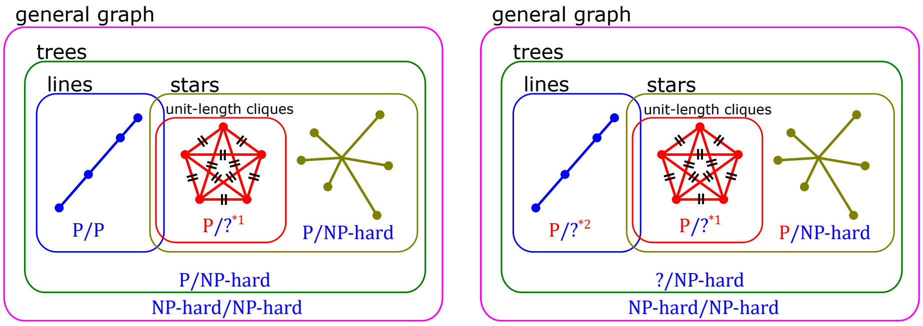


Figure 1: Classes of graphs and the complexity of the patrolling problem. The left figure is for a single agent and the right figure is for multiple agents in cooperation. The left side of the slash represents the case of uniform idle times, and the right side represents the case of arbitrary idle times.

3 Uniform idle time

We first considered the case where all vertices have the same idle time. We proved that our problem can be solved in polynomial time both for lines and for stars, because in these cases the situation is simplified in the following way: For lines, we showed that if patrolling is possible at all, then there is a patrolling schedule where each vertex is guarded by one agent alone. For stars, there is a schedule where the only form of cooperation is for several agents to periodically visit a subset of vertices in the same order at a certain time interval.

For these graphs, we can also efficiently solve the optimization problem where each vertex has a profit and we want to maximize the sum of the profit of the guarded vertices.

4 Arbitrary idle times

On the other hand, because it was difficult to determine the complexity for distinct idle times, we considered another variant of the problem where, instead of idle times, the exact times at which each point must be visited is specified.

For this setting, we found an algorithm that greedily determines the motions of the agents in lines (Figure 1-*2). Furthermore, we showed the NP-hardness of the optimization problem mentioned at the end of section 3 for unit-length cliques (Figure 1-*1) by a reduction from the maximum independent set problem.

References

[1] S. Coene, F. C. R. Spieksma, and G. J. Woeginger. Charlemagne's challenge: the periodic latency problem. *Operations research*, 59(3), pp. 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), LNCS 6942, pp. 701–712, 2011.

K_3 edge cover in a wide sense

CHIBA Kyohei*‡ ITO Hiro*‡ NAGAO Atsuki†‡
*School of Informatics and Engineering,
The University of Electro-Communications (UEC); Tokyo, Japan.
†Faculty of Science and Technology, Seikei University; Tokyo, Japan.
‡CREST, JST; Tokyo, Japan.
*{c1731106@edu.cc., itohiro@}uec.ac.jp, †a-nagao@st.seikei.ac.jp

In this study, we consider a problem for finding the minimum number of 3-cliques (K_3s) that covers all edges of a given $G = (V, E)$. Covering one edge by more than one 3-cliques is allowed. Moreover, in this problem, we allow "spilling-out," i.e., a set of three vertices $\{x, y, z\}$ can be covered by a 3-clique even if the induced subgraph of them is not a clique. We call this problem K_3 *edge cover problem in a wide sense.* This problem is an extension of the schoolgirl problem, finite projective planes, and experimental designs. Allowing spilling-out is useful for some applications: E.g., when we want to compare n items through some tries of experiments, in which at most three items can be compared simultaneously, and pairs of items that must be compared are given by a graph, finding the minimum number of tries is formalized as this problem. In the known researches, there are many results that considered problems for covering vertices or edges by minimum number of cliques [1, 2]. However, there is no theoretical result that considers spilling-out.

We obtain the following results:

1. The problem is NP-hard even if graphs are restricted to planar, cubic, and $\{C_4, C_5, \text{bowtie}\}$-free in a sense of subgraphs (i.e., not restricted to induced ones), where a *bowtie* is a graph isomorphic to a graph consisting of two 3-cliques sharing one vertex.

2. For the problem with a parameter k, which is the number of 3-cliques in G, there is an $O(2^k m)$-time algorithm.

3. For the problem, if the maximum degree is bounded by an integer d and a tree decomposition with the tree-width t is given, there is an $O(2^{t^2+t} d^{2t+2} t^2 n)$-time algorithm.

References

[1] Michael R. Garey, David S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman & Co., NY, USA, 1979.

[2] James Orlin, Contentment in graph theory: Covering graphs with cliques, Indagationes Mathematicae (Proceedings), Vol. 80, Issue 5, pp. 406–424, 1977.

On the Complexity of Finding a Largest Common Subtree of Trees (Extended Abstract)

Hiroki Katsumata*, Satoshi Tayu , Shuichi Ueno

Tokyo Institute of Technology, Tokyo 152–8550-S3-57, Japan
{katsumata,tayu,ueno}@eda.ce.titech.ac.jp

The largest common subgraph problem (LCSG) is defined as follows.

LCSG (Largest Common Subgraph of Graphs)

Instance: Graphs $G_1, G_2, \ldots, G_N$.
Question: Find a connected graph with the maximum number of edges that is a subgraph of all the G_is.

It has been known that LCSG has applications in various areas such as pattern recognition, bioinformatics, and cheminformatics [2, 5, 9]. LCSG is NP-hard even if $N = 2$, since it is a natural generalization of the subgraph isomorphism problem, which is well-known to be NP-hard [3].

We denote the vertex set and the edge set of a graph G by $V(G)$ and $E(G)$, respectively. Let $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a sequence of subsets of $V(G)$. The *width* of $\mathcal{X}$ is $\max_{1 \le i \le r} |X_i| - 1$. $\mathcal{X}$ is called a *path-decomposition* of G if the following conditions are satisfied:

(i) $\bigcup_{1 \le i \le r} X_i = V(G)$;

(ii) for any edge $(u, v) \in E(G)$, there exists an i such that $u, v \in X_i$;

(iii) for all l, m, and n with $1 \le l \le m \le n \le r$, $X_l \cap X_n \subseteq X_m$.

The *pathwidth* of G, denoted by $\mathrm{pw}(G)$, is the minimum width over all path-decompositions of G [8].

Since the subgraph isomorphism problem is NP-hard for graphs of pathwidth two [6], LCSG is NP-hard even if $N = 2$ and $\mathrm{pw}(G_1) = \mathrm{pw}(G_2) = 2$. We first show the following.

Theorem 1. *LCSG can be solved in polynomial time if* $N = O(1)$ *and* $\mathrm{pw}(G_i) = 1$ *for* $1 \le i \le N$. □

The largest common subtree problem (LCST) is a subproblem of LCSG, and defined as follows.

LCST (Largest Common Subtree of Trees)

Instance: Trees $T_1, T_2, \ldots, T_N$.
Question: Find a tree with the maximum number of edges that is a subtree of all the T_is.

LCST can be solved in polynomial time if $N = 2$, pioneered by Edmonds and Matula in the 1960s [7], and faster algorithms have been proposed in the literature [1, 4]. Akutsu showed that LCST is NP-hard if $N = 3$ [1]. It is implicit in his proof that LCST is NP-hard even if $N = 3$ and the pathwidth of every input tree is three. We next show the following.

*Presenter

Theorem 2. *LCST is NP-hard even if* $N = 3$ *and* $\text{pw}(T_i) = 2$ *for* $1 \le i \le 3$. □

We also show the following.

Theorem 3. *LCST is NP-hard even if* $\text{pw}(T_i) = 1$ *for* $1 \le i \le N$. □

It should be noted that Theorems 2 and 3 complement Theorem 1.

It is shown in [1] that LCST can be solved in polynomial time if $N = O(1)$ and $\Delta(T_i) = O(1)$ for $1 \le i \le N$, where $\Delta(G)$ is the maximum degree of a vertex of a graph G. We finally show the following.

Theorem 4. *LCST is NP-hard even if* $\text{pw}(T_i) = 1$ *and* $\Delta(T_i) = 3$ *for* $1 \le i \le N$. □

References

[1] Tatsuya Akutsu. An RNC algorithm for finding a largest common subtree of two trees. *IEICE Transactions on Information and Systems*, 75(1):95–101, 1992.

[2] Donatello Conte, Pasquale Foggia, Carlo Sansone, and Mario Vento. Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence*, 18(03):265–298, 2004.

[3] Stephen A. Cook. The complexity of theorem-proving procedures. *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, 1971.

[4] Andre Droschinsky, Nils M. Kriege, and Petra Mutzel. Faster algorithms for the maximum common subtree isomorphism problem. *Leibniz International Proceedings in Informatics*, 58:33:1–33:14, 2016.

[5] Hans-Christian Ehrlich and Matthias Rarey. Maximum common subgraph isomorphism algorithms and their applications in molecular science: a review. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1(1):68–79, 2011.

[6] Arvind Gupta and Naomi Nishimura. The complexity of subgraph isomorphism for classes of partial k-trees. *Theoretical Computer Science*, 164(1):287–298, 1996.

[7] David W. Matula. Subtree isomorphism in $O(n^{5/2})$. *Annals of Discrete Mathematics*, 2:91–106, 1978.

[8] Neil Robertson and Paul D. Seymour. Graph minors. i. excluding a forest. *Journal of Combinatorial Theory, Series B*, 35(1):39–61, 1983.

[9] Leander Schietgat, Jan Ramon, and Maurice Bruynooghe. A polynomial-time maximum common subgraph algorithm for outerplanar graphs and its application to chemoinformatics. *Annals of Mathematics and Artificial Intelligence*, 69(4):343–376, 2013.

The Odd Depth Tree Problem

Hiroshi Nishiyama[1]

[1] Graduate School of Information Science and Electrical Engineering,
Kyushu University, Fukuoka, 819-0395, Japan
hiroshi.nishiyama@inf.kyushu-u.ac.jp

July 2, 2017

Abstract

Spanning tree is a fundamental combinatorial object in many research areas, and its variants with some constraints also have been well studied. This paper studies a variant having a parity condition. The odd depth tree problem is a problem to decide whether a given undirected graph has a spanning tree such that every leaf has an odd distance from a prescribed root vertex. For bipartite graphs, we give a Hall-type characterization of graphs having an odd depth tree, and based on the characterization we give a polynomial time algorithm to construct an odd depth tree. We also extend the algorithm to a weighted version. On the other hand, we show the problem is NP-complete for non-bipartite graphs. We also deal with directed graphs, for which we give some results similar to undirected case.

1 Introduction

A spanning tree of a graph G is a spanning connected subgraph of G having no cycles. Spanning tree has been one of the topics that is payed most attentions in various research areas such as graph theory, matroid theory, and game theory for the theoretical point of view, or network designs, distributed algorithms, and data structures as application areas.

While it is easy to find a spanning tree (of minimum weight) in a graph, there are many cases it becomes intractable when some constraints are given. For example, finding a spanning tree having minimum or maximum number of leaves, bounded degrees, and bounded diameter, are all known to be NP-hard.

Meanwhile, there are plenty of graph problems with *parity* constraints studied; realizing plane graphs with prescribed parity of degrees of vertices [1], subset feedback set problem with parity condition [2], multi-way cut with parity constraint [3], etc. Signed graph is an example of a large topic concerning parity.

In this paper, we study the spanning tree problem with a parity condition. An *odd depth tree* is a spanning tree such that every leaf has an odd distance from a specified root vertex. The *Odd depth tree problem* is the problem to find an odd depth tree with respect to the root vertex in a given undirected graph.

Our results are as follows. For bipartite graphs, we show a Hall-type characterization of graphs having an odd depth tree. We also give a polynomial time algorithm to construct an odd depth tree which is based on the characterization, which is applicable for minimization version of the problem as well. For non-bipartite graphs, we show the problem is NP-complete. We also deal with directed graphs, for which we obtain similar results to the undirected case.

2 Main Results

Let $G = (V, E)$ be an undirected graph and $r \in V$. A spanning tree T of G is an *odd depth tree with respect to* r if every leaf of T (except r if it is a leaf) has an odd distance from r. The *Odd depth tree problem* is the problem to find an odd depth tree with respect to r in a given undirected graph and a root vertex r. We first consider bipartite graphs.

Lemma 1. *Let $G = (U, V; E)$ be a bipartite graph. Suppose $r \in U$. Then G has an odd depth tree with respect to r if and only if*

$$|N(X)| \geq |X| + 1 \tag{1}$$

holds for every $X \subseteq U \setminus \{r\}, X \neq \emptyset$, where $N(X)$ denotes the set of neighbors of vertices in X.

By Lemma 1, we obtain the following tractability result.

Theorem 2. *The odd depth tree problem is solved in polynomial time for bipartite graphs.*

For *weighted* version of the problem, we obtain the following result.

Theorem 3. *Let $G = (U, V; E)$ be a bipartite graph and let $r \in U$ and $w \colon E \to \mathbb{R}_{\geq 0}$ be a nonnegative weight on edges. An odd depth tree with respect to r of minimum weight is found in polynomial time.*

For non-bipartite graphs, we obtain the following hardness result. The reduction is from CNF-SAT.

Theorem 4. *Odd depth tree problem is NP-complete for two-connected non-bipartite graphs.*

We also consider the problem for directed graphs (*digraph* for short). Let $D = (V, A)$ be a digraph and $r \in V$. An *in-tree with root r* is a directed tree such that every vertex except r has out-degree one, and r has no outgoing arc. An in-tree T of D is an *odd depth in-tree with respect to r* if every leaf of T has an odd distance from r. The *odd depth in-tree problem* is the problem to find an odd depth in-tree with respect to r in a given digraph and a root vertex r.

Similar to the undirected case, we first consider bipartite graphs. The following lemma gives a characterization of bipartite digraphs which contain odd depth in-trees.

Lemma 5. *Let $D = (U, V; A)$ be a bipartite digraph which contains an in-tree with root r. Suppose $r \in U$. Then D has an odd depth in-tree with respect to r if and only if there is a perfect matching M with respect to $U \setminus \{r\}$ consisting of arcs from V to $U \setminus \{r\}$, which satisfies the following:*

(A) *For every non-empty $X \subseteq U \setminus \{r\}$, there exists a vertex $v \in N^+(X) \setminus M(X)$ such that $N^+(v) \setminus X \neq \emptyset$,*

where $N^+(X)$ denotes the set of vertices having incoming arcs from some member of X, and $M(X)$ denotes the set of vertices mated to some vertex in X in M.

For bipartite DAGs (digraphs having no directed cycles), we obtain a much simpler characterization.

Lemma 6. *Let $D = (U, V; A)$ be a bipartite DAG which has an in-tree with root r. Suppose $r \in U$. Then D has an odd depth tree with respect to r if and only if*

$$|N^-(X)| \geq |X| \tag{2}$$

holds for any $X \subseteq U \setminus \{r\}$, where $N^-(X)$ denotes the set of vertices having outgoing arcs to some member of X.

The characterization of Lemma 6 implies the following tractability result.

Theorem 7. *The odd depth in-tree problem is solved in polynomial time for bipartite DAGs.*

For non-bipartite digraphs, we obtain the following hardness result.

Theorem 8. *Odd depth in-tree problem is NP-complete for non-bipartite DAGs.*

References

[1] O. Aichholzer, T. Hackl, M. Hoffmann, A. Pilz, G. Rote, B. Speckmann, B. Vogtenhuber, Plane graphs with parity constraints, proc. WADS 2009, 13–24.

[2] N. Kakimura, K. Kawarabayashi, Fixed-parameter tractability for subset feedback set problems with parity constraints, Theoretical Computer Science, **576** (2015), 61–76.

[3] D. Lokshtanov, M. S. Ramanujan, Parameterized tractability of multiway cut with parity constraints, proc. ICALP 2012, 750–761.

Counting Hamiltonian Cycles on Quartic 4-Vertex-Connected Planar Graphs

Robert D. Barish[1] and Akira Suyama[1]

[1] Graduate School of Arts and Sciences,
University of Tokyo, Meguro-ku Komaba 3-8-1 Tokyo 153-8902, Japan
rbarish@genta.c.u-tokyo.ac.jp

Abstract. We show that counting Hamiltonian cycles on quartic 4-vertex-connected planar graphs is $\#P$-complete under many-one counting ("weakly-parsimonious") reductions, and that no Fully Polynomial-time Randomized Approximation Scheme (FPRAS) can exist for this integer counting problem unless $NP = RP$.

Discussion. Motivated by Tait's observation that the Hamiltonicity of cubic 3-connected planar graphs (polyhedral graphs by Steinitz's Theorem) would imply the 4-color theorem [1], in 1931 Hassler Whitney proved that every planar triangulation having no separating triangles is Hamiltonian [2]. While a non-Hamiltonian cubic 3-connected planar graph on 46 vertices was eventually found by Tutte [3], Whitney's proof nevertheless initiated a search for minimum set of constraints necessary to ensure the Hamiltonicity of a class of graphs. Tutte's 1956 proof of the Hamiltonicity of 4-vertex-connected planar graphs [4] was arguably a consequence of this program, and within a few decades, it was established by Thomassen that 4-vertex-connected planar graphs were also "Hamiltonian connected" [5] (i.e. that there will always exist some embedding of a Hamiltonian path in an arbitrary 4-vertex-connected planar graph such that the path's endpoints lie on any specified pair of vertices), and that Hamiltonian circuits could be found in 4-vertex-connected planar graphs in linear time [6]. From Garey, Johnson, and Tarjan's proof of the NP-completeness of the Hamiltonian cycle problem on cubic 3-connected planar graphs [7], and from Meredith's construction of a non-Hamiltonian quartic 4-vertex-connected graph [8], it was moreover established that both 4-vertex-connectivity and planarity were "near-minimal" criterion for ensuring Hamiltonicity and the algorithmically efficient discovery of Hamiltonian cycles.

With the hope of using Valiant's class $\#P$ of integer counting problems [9] as a means of obtaining a complexity theoretic metric beyond NP-completeness for determining when vertex degree, connectivity, and planarity restrictions conspire to allow efficient access to the set of Hamiltonian cycles in a graph, and noting the 2003 proof by Liśkiewicz, Ogihara, and Toda that counting Hamiltonian cycles on cubic 2-connected planar graphs is $\#P$-complete [10], we investigate the complexity of counting Hamiltonian cycles on 4-vertex-connected planar graphs. Here, we discovered that not only was the problem of counting Hamiltonian

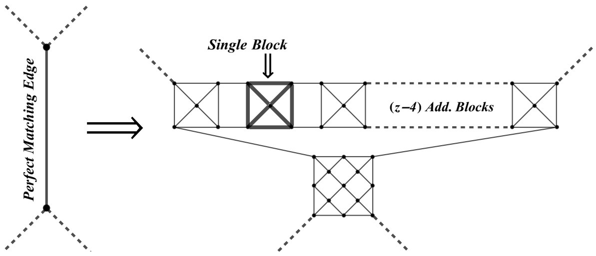


Fig. 1. Let G be an arbitrary cubic 3-connected essentially-4-(edge,vertex)-connected planar graph with n vertices. By Petersen's Theorem we have that G contains an edge set $(e_1, e_2, ...) \in P$ corresponding to a perfect matching for the graph. If we substitute all of the perfect matching edges in the manner shown, setting the number of "blocks" z equal to n^2, and call the resultant 4-vertex-connected planar graph H, then the number of Hamiltonian cycles in G will be exactly equal to $\lfloor \frac{(\#\ Hamiltonian\ Cycles\ in\ H)}{(16\times(3*2^{n^2}+5^{n^2}-1))^{n/2}} \rfloor$.

cycles on 4-vertex-connected planar graphs $\#P$-complete, but moreover, that no Fully Polynomial-time Randomized Approximation Scheme (FPRAS) could exist for the problem unless $NP = RP$. Our method of proof involves first establishing NP-completeness and $\#P$-completeness for the problems of deciding the existence of and counting Hamiltonian cycles, respectively, on cubic 3-connected essentially-4-(edge,vertex)-connected planar graphs. We then proceed via a many-one counting ("weakly-parsimonious") reduction from counting Hamiltonian cycles in cubic 3-connected essentially-4-(edge,vertex)-connected planar graphs to 4-vertex-connected planar graphs (see Fig. 1).

References

1. Tait, P. G.: Listing's topologie. *Philosophical Magazine (5th ser.)* **17**, pp. 30–46 (1884)
2. Whitney, H.: A theorem on graphs. *Ann. Math.* **32**(2), 378–390 (1931)
3. Tutte, W. T.: On Hamiltonian circuits. *J. London Math. Soc.* **21**, 98–101 (1946)
4. Tutte, W. T.: A theorem on planar graphs. *Trans. Amer. Math. Soc.* **82**, 99–116 (1956)
5. Thomassen, C.: A theorem on paths in planar graphs. *J. Graph Theory* **7**, 169–176 (1983)
6. Chiba, N.: The Hamiltonian cycle problem is linear-time solvable for 4-connected planar graphs. *J. Algorithms* **10**(2), 187–211 (1989)
7. Garey, M. R., Johnson, D. S., Tarjan, R. E.: The planar Hamiltonian circuit problem is NP-complete. *SIAM J. Comput.* **5**(4), pp. 704–714 (1976)
8. Meredith, G. H. J.: Regular n-valent n-connected nonHamiltonian non-n-edge-colorable graphs. *J. Combin. Theory Ser. B* **14**(1), 55–60 (1973)
9. Valiant, L. G.: The complexity of computing the permanent. *Theor. Comput. Sci.* **8**(2), pp. 189–201 (1979)
10. Liśkiewicz, M., Ogihara, M., Toda, S.: The complexity of counting self-avoiding walks in subgraphs of two-dimensional grids and hypercubes. *Theor. Comput. Sci.* **304**(1-3), pp. 129–156 (2003)

Complexity of Benndorf's "The Game"

Jason S. Ku* Mikhail Rudoy*

"The Game" is the title of a neat cooperative card game designed by Steffen Benndorf and published by Nürnberger-Spielkarten-Verlag [1]. In this paper, we describe The Game, provide a number of generalizations, and discuss the computational complexity of winning them, provided a given input card distribution.

The Game requires players to cooperatively discard all cards in a deck of 100 uniquely numbered cards into a set of piles that follows certain rules. In the original, cards placed on a given pile must be either monotonically increasing or decreasing, with the exception that any card exactly 10 away from the top card in the pile may also be played. These rules define a valid placement graph for each pile which we call a *transition graph*. The players win if they can place all the cards in turn onto piles in a way that is consistent with each pile's transition graph.

We consider a one-player version of The Game that is *offline*, i.e., the order of the cards in the deck is known. We will generalize this game for variable deck size n, variable hand size h, variable play number p, with a variable number of discard piles k, each having a transition graph that may be any general directed graph. We show that if either the hand size or the number of piles is non-constant, then deciding whether one can win The Game is NP-Complete.

We begin with some definitions. A *Card* is a number in $[1..n]$; cards are unique. A *Deck* is a permutation on the cards from which we draw. The *Hand* is a temporary storage in which we place cards drawn from the Deck. *Hand Size* is the max number of cards in the Hand. The *Play Number* is the min number of cards that can leave the Hand per turn. A *Pile* is a container into which we can place cards from the Hand. A *Pile Transition Graph* is a graph on vertices $[0..n]$, with 0 referencing no card, and vertex i corresponding to card i. Edge (i, j) says that card j may be placed on top of card i.

Every turn, we remove at least or exactly the Play Number of cards from the Hand and place these cards one at a time onto piles on which they can legally be placed. Then we replenish the Hand to the Hand Size from the Deck. The Game ends if no more actions can be made. The player wins if every card is in a pile and the decision problem for The Game is then: can the player win for a given input? This decision problem is in NP, certified by a play order.

*MIT, {jasonku,mrudoy}@mit.edu

Theorem 1 *The Game can be decided in polynomial time if the hand size and the number of piles are constant.*

Proof. For a constant number of Piles and a constant Hand Size, the game state graph is polynomial in size $O(n^{h+k+1})$, so we can easily determine whether a path exists from the starting state to any win state. Here state encodes the hand contents, the top card of each pile, and the progress through the deck. □

When either hand size or number of piles is not constant, and Pile Transition Graphs are allowed to encode general graphs provided as input, the problem becomes hard.

Theorem 2 *The Game is NP-Complete when hand size is a constant fraction of deck size, allowing general transition graphs for one discard pile.*

Proof. For one pile with a general directed transition graph and hand size that is a constant fraction r of deck size (or part of the input), we can reduce from Hamiltonian Path. We construct a transition graph as follows. Let the Hand Size be the size n of the input graph, and fix the Deck size to n/r. Let the transition graph on vertices $[1..n/r]$ be our Hamiltonian Path instance with vertex 1 the start vertex and vertex n the end vertex. Add edge $(0, 1)$ and edge $(i, i + 1)$ for $n \leq i < n/r$. Set the Deck to be $[1..n]$ in that order. Analysis is straightforward and not detailed here. □

Theorem 3 *The Game is NP-Complete for constant hand size and play number, allowing general transition graphs for many piles.*

This can be proved via a chain of reductions starting from the following NP-Complete problem (from [2]):

[Swap or Not Reachability] Given permutations σ_1, σ_2, …, σ_k, and T on n elements such that σ_i is a swap of some two elements and T is an arbitrary permutation, decide whether there exists a bitstring $b_1 b_2 \dots b_k$ such that $\sigma_k^{b_k} \circ \sigma_{k-1}^{b_{k-1}} \circ \dots \circ \sigma_2^{b_2} \circ \sigma_1^{b_1} = T$.

We reduce this problem to the following more general problem:

[DAG Path Cover] Given a DAG and a list of pairs of vertices (s_i, t_i), decide whether there exists disjoint paths from each s_i to its corresponding t_i such that each vertex of the DAG exists in one of the paths.

Lemma 4 DAG PATH COVER *is NP-Complete.*

Proof of this Lemma is omitted for brevity. We then use this lemma to prove the preceding theorem:

Proof. Given a DAG PATH COVER instance D, with DAG G on m vertices requiring k paths, we construct an instance I of The Game with hand size h, play number p, k piles, and $n = (h-p)+m+k$ cards.

The deck is given in order from 1 to n. We call cards $\{1,\dots,h-p\}$ *trash*, cards $\{h-p+1,\dots,m-k\}$ *cool*, and cards $\{m-k+1,\dots,m\}$ *sentinel*. Our strategy will be to construct a transition graph for each pile, each containing G as a subgraph such that the play order on pile i contains a path in G from s_i to t_i.

We construct a correspondence between cool cards and vertices of G, such that if (u,v) is an edge, then corresponding cards c_u and c_v satisfy $c_u < c_v$. This is possible using any topological sort on G. In the transition graph for every pile, we add an edge from card c_u to c_v if (u,v) is an edge of G. For pile i, we add an edge from the empty state 0 to the card corresponding to vertex s_i, and an edge from the card corresponding to vertex t_i to sentinal card $m-k+i$. Lastly, we form a directed chain through the trash cards starting at sentinal card m in the transition graph of pile k. By this construction, the trash cards can only be played after the last card in the deck.

Given a solution to D consisting of paths $\mathcal{S}_p$, we construct a solution to I. We will play all non-trash cards in order and then all the trash cards. Assuming there are piles that can accept this order of cards, this play order is possible given the deck and play number. We now show that there always exists a pile on to which we can place these cards. We place cards on to piles as follows:

C1 For each cool card c corresponding to start vertex s_i, we will place that card onto pile i. Provided the pile is empty, we will be able to place this card.

C2 For each cool card c corresponding to non-start vertex v, we will place c on to the card corresponding to the previous vertex u along the path containing v. Provided the card corresponding to u is on the top of some pile, we will be able to play c.

C3 For every sentinel card $m-k+i$, we will place it onto the card corresponding to t_i. Provided the card corresponding to t_i is on the top of pile i.

C4 For every trash card c, we will place it on pile k on top of the parent of c in the transition graph of pile k.

Now we argue that the above placements are possible.

C1 No card in [C2], [C3], [C4] is placed on an empty pile and no two cards in [C1] are placed onto the same pile, so we can place cards in [C1].

C2 Because the cards corresponding to path $\mathcal{S}_j$ are played in path order based on the topological sort, no card in [C2] will be played before the card it should be played on. Further, no two cards are placed onto the same card, so we can place cards in [C2]. Note that all cards corresponding to vertices in path $\mathcal{S}_p$ are placed in pile p by induction.

C3 Because cool cards are placed before sentinel cards, no card in [C3] will be played before the card it is played on. Since the card corresponding to t_i will be placed in pile i, we will place the sentinel card $m-k+i$ in the correct pile, so we can place cards in [C3].

C4 Since card m is in [C3] and can be played, we can play the trash cards in the order in which they appear in the transition graph of pile k.

Thus we can play all cards, solving I. Given a solution to I, we construct a solution to D. Consider pile i. Card $m-k+i$ can only be played on this pile, and only immediately after the card corresponding to t_i. The card corresponding to s_i is the only card that can be played onto pile i when it is empty. The sequence of cards played on pile i starting with the card corresponding to s_i and ending with the card corresponding to t_i will be a path in the transition graph of pile i. The only transition in the transition graph of pile i between cool cards and non-cool cards is from t_i, so this path is in G. No card can be placed in two piles, so the paths are disjoint, and every card must be placed, so the paths cover the vertices of G. These paths comprise a solution to D.

Specifying play order and location can be described in polynomial time, so the problem is in NP. □

In our generalized models, we allow for transition graphs to be general graphs, but the original game uses only very particular transition graphs. If transition graphs are not part of the input and only depend on the size of the deck, is The Game still NP-Hard? We leave this question to future work.

References

[1] S. Benndorf. The Game. Nürnberger-Spielkarten-Verlag: `https://boardgamegeek.com/boardgame/173090/game`, 2015.

[2] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

Complexity of "Goishi Hiroi"

Masanori Fukui[1] Koki Suetsugu[2] Akira Suzuki[3]

[1]Hyogo University of Teacher Education, m16195c@hyogo-u.ac.jp
[2]Kyoto University, suetsugu.koki.72r@st.kyoto-u.ac.jp
[3]Tohoku University, a.suzuki@ecei.tohoku.ac.jp

1. Introduction

In this paper, we treat a Japanese picking stone game called "Goishi Hiroi." This game has a long history so that we can find it in "Wakoku Chie Kurabe" written by Kanchusen Tagaya in 1727 [1]. Fig.1 is the page introducing "Goishi Hiroi" in [1]. While [1] is a rare book, we scaned this figure from [2]. "Goishi Hiroi" is a game to pick up all stones arranged on a lattice board in the following way: (1) pick up one stone and choose a direction from four directions (up, down, left or right); (2) trace the lattice board from the previous stone for the current direction until there is a stone, then you pick up the stone, and choose a new direction from three directions (same, right or left of the current direction); and (3) continue (2) until there are no stones. Note that once you picking up the stone, there are no stone any more, therefore if you come back again there, you cannot change the direction at there.

In this paper, we consider the deterministic problem of "Goishi Hiroi," say Goishi Hiroi. In the Goishi Hiroi we are given an arrangement of stones. Then we determine whether there exists a way to pick up all stones or not. For a stone arrangement S, we denote Goishi Hiroi for S as the instance of this problem whose given arrangement of stones is S. While this problem permits multiple passes through vertex, this problem is a variant of Hamilton path problem in a sense. In other words, this problem is a specific example of Hamilton path problem. There is a related work about the Goishi Hiroi. Miyadera and Fukui give a solution of Goishi Hiroi for *Yatsuhashi Type* (see Figs. 2 and 3) for each $x = 4$ and $y \geq 1$ [4]. Note that "Wakoku Chie Kurabe" [1] showed the solution only for Yatsuhashi Type with $x = 4$ and $y = 5$.

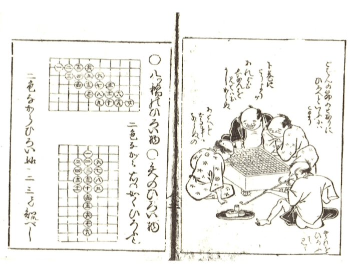


Fig. 1: The page introducing "Goishi Hiroi" in [1].

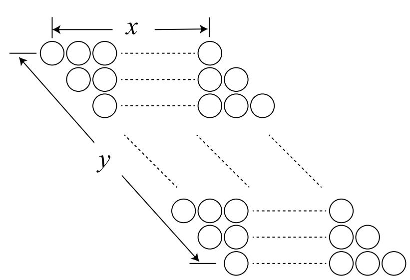


Fig. 2: The stone arrangement of Yatsuhashi Type.

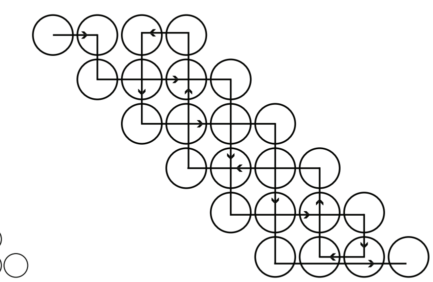


Fig. 3: A solution of Yatsuhashi Type with $x = 4$ and $y = 6$.

In this paper, we give both negative and positive results for the Goishi Hiroi. In Section 2, we show the computational hardness of the Goishi Hiroi. In Section 3, we show that we can solve the Goishi Hiroi for specific types of arrangements called *Stairs Type* and Yatsuhashi Type.

Due to page limitation, we omit proofs of some theorems.

2. Computational hardness

In this section, we prove that the Goishi Hiroi is NP-complete. We will construct a polynomial-time reduction from well known NP-complete problem, Hamiltonian Path [3] to our problem. We emphasize that our reduction is carefully designed so that each solution of the corresponding instance have the one-to-one correspondence with each solution of a given instance of Hamiltonian Path. Therefore, our reduction also proves that the counting variants and the another-solution-problem variants of the Goishi Hiroi are #P-complete and ASP-complete, respectively. Note that the another-solution-problem variant of a problem is defined as follows [5]: Given an instance of the problem and its solution s, then find a solution s' of the instance other than s if exists.

Theorem 1. The GOISHI HIROI and its counting and another-solution-problem variants are NP-complete, #P-complete and ASP-complete, respectively. □

3. Solvable cases

In the previous section, we show the computational hardness of the GOISHI HIROI. On the other hand, we show that in some special cases, we can solve the GOISHI HIROI. We treat the case where given arrangements are Stairs Type and Yatsuhashi Type.

First, we focus on the GOISHI HIROI for Stairs Type (See Fig. 4). We can immediately solve the small cases; the GOISHI HIROI is Yes-instance for Stairs Type with $n = 1$ or $n = 2$; on the other hand, it is No-instance solution for $n = 3$. In the following Theorem, we show that there is a solution for Yes-instance for each $n \geq 4$:

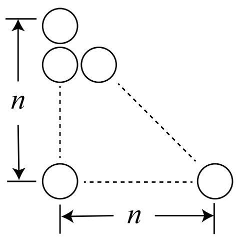


Fig. 4: The stone arrangement of Stairs Type.

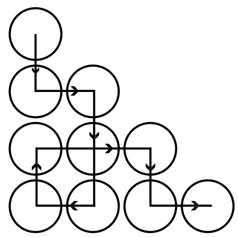


Fig. 5: Stairs Type with $n = 4$.

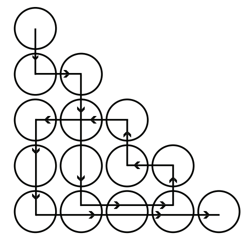


Fig. 6: Stairs Type with $n = 5$.

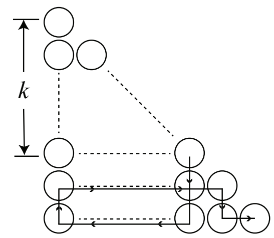


Fig. 7: The solution for the induction step in Theorem 3.

Theorem 2. The GOISHI HIROI is Yes-instance if a given arrangement is Stairs Type with $n \geq 4$.

Proof. We can show that the GOISHI HIROI is Yes-instance for Stairs Type with $n = 4$ and $n = 5$ as Figs. 5 and 6, respectively. In the other cases, we prove by induction. If there is a solution for $n = k$, then we can construct the solution for $n = k + 2$, by extending the solution for $n = k$ as Fig. 7. Since there are solutions for $n = 4$ and $n = 5$, we have solution for each $n \geq 4$. □

Next we focus on the GOISHI HIROI for Yatsuhashi Type (see Fig. 2).

Theorem 3. The GOISHI HIROI for Yatsuhashi Type is solvable for each $x, y \geq 1$. □

Using this theorem, we can obtain the result of analysis for Yatsuhashi Type. The GOISHI HIROI for Yatsuhashi Type has a solution as Table 1.

Table 1: The results of analysis for Yatsuhashi Type (Y: Yes-instance, N: No-instance).

<table>
<tr><th>$y \setminus x$</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7+ odd</th><th>7+ even</th></tr>
<tr><td>1</td><td colspan="8">Y</td></tr>
<tr><td>2</td><td>N</td><td>Y</td><td>N</td><td>Y</td><td>N</td><td>Y</td><td>N</td><td>Y</td></tr>
<tr><td>3+</td><td>N</td><td>Y</td><td>N</td><td colspan="5">Y</td></tr>
</table>

References

[1] K. Tagaya, Wakoku Chie Kurabe, 1727. (in Japanese)

[2] H. Akiyama, The Book of Board Puzzles, Shinkigensha, pp. 189–191, 2009. (in Japanese)

[3] M.R. Garey, D.S. Johnson, Some simplified NP-complete problems, *Proc. 6th ACM Symposium on Theory of Computing (STOC '74)*, pp. 47–63, 1974.

[4] R. Miyadera, M. Fukui, High School Mathematics Research using mathematical games, *The Journal of Game Amusement Society*, Game Amusement Society, **10**(1), accepted. (in Japanese)

[5] T. Yato, T. Seta, Complexity and completeness of finding another solution and its application to puzzles, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E86-A, pp. 1052–1060, 2003.

On the complexity of lattice puzzles

Yasuaki Kobayashi* Koki Suetsugu† Hideki Tsuiki†

1 Introduction

Lattice puzzle (*Koshi puzzle*, in Japanese) is a kind of assembly puzzles which is usually played on wooden pieces (as in Figure 1-(a)), and the goal is to combine them into a lattice pattern as in Figure 1-(b). The name of this puzzle may be originated from a woodwork, called *chidori-koshi*, which can be found in an old book written in the 18th century [2][1], for example. Although the puzzle itself is not particularly complicated, it is quite difficult to solve even when the number of pieces is small. Given this, it is natural to ask whether the puzzle is difficult for computers. In this research, we consider simplified versions of this puzzle and clarify the computational complexity of those.

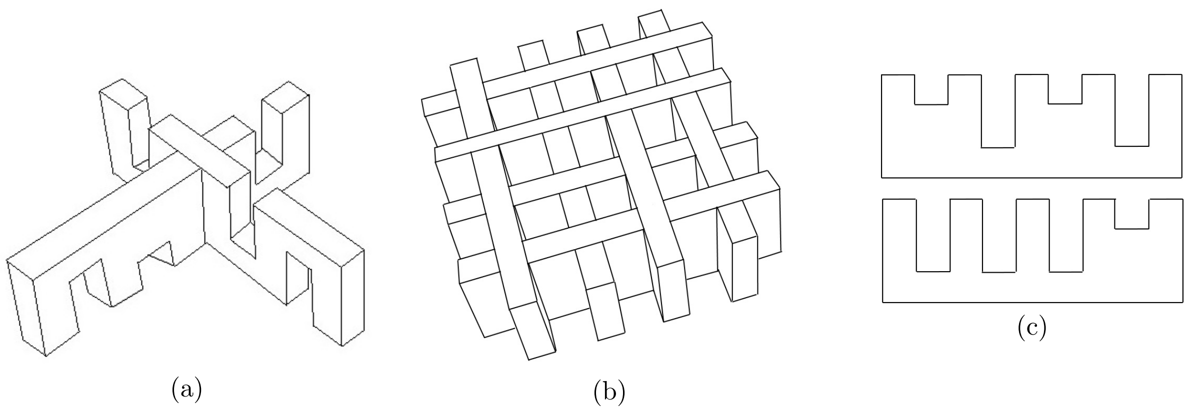


Figure 1: (a) Two pieces are combined at their slots. (b) An assembled lattice puzzle of eight pieces. (c) Each piece has two types of slots, and the slots are on one side of the piece.

The puzzle considered in this extended abstract is as follows. We have a set of n pieces, called *vertical pieces*, that are placed parallel to each other and another set of n pieces, called *horizontal pieces*, that are placed perpendicular to vertical pieces. Each piece has n evenly spaced slots on one side, and each slot is either a deep slot or a shallow slot as in Figure 1-(c). To combine them, we need to consider (1) the arrangements of vertical/horizontal pieces and (2) the direction of each piece. The vertical (resp. horizontal) arrangement is determined by a permutation on the set of vertical pieces (resp. the set of horizontal pieces) and the direction is determined by rotating 180 degrees for each piece. Let us note that since slots are only on one side of each piece, the direction of the rotation is uniquely determined. Therefore, there are $n! \cdot n! \cdot 2^n \cdot 2^n$ possibilities on the configurations of the pieces. Suppose we are given a configuration of the pieces. We say that the lattice puzzle is solved in this configuration if for every $1 \leq i, j \leq n$, the j-th slot of the i-th vertical piece is matched to the i-th slot of the j-th horizontal piece, where two slots are *matched* to each other if and only if one slot is deep and the other is shallow. The problem is, given n vertical pieces and n horizontal pieces, deciding whether the puzzle can be solved or not. We study the computational complexity on three variants of this puzzle.

*Graduate School of Informatics, Kyoto University
†Graduate School of Human and Environmental Studies, Kyoto University
[1] The book can be found in `http://www.wul.waseda.ac.jp/kotenseki/html/i16/i16_00875/index.html`.

2 Computational complexity

2.1 An easy case

Suppose here that we are given an arrangement of vertical pieces and that of horizontal pieces. The task is to solve the puzzle by only rotating each piece.

Theorem 1. *The problem of solving the lattice puzzle can be solved in polynomial time when we are only allowed to rotate pieces.*

The theorem is proved by showing that the instance can be reduced to an instance of the 2-SAT problem with $O(n^2)$ variables. It is well-known that the 2-SAT problem can be solved in linear time.

2.2 A not-so-easy case

Suppose here that we are not allowed to rotate any pieces. The task is to solve the puzzle by only arranging vertical pieces and horizontal pieces.

Theorem 2. *The problem of solving the lattice puzzle is equivalent to the graph isomorphism problem on bipartite graphs when we are only allowed to arrange pieces.*

The proof goes as follows. We construct two bipartite graphs that have n vertices for one color class and n vertices for the other color class. The construction can be done in polynomial time and reflects the relation between pieces and the position of slots. We can show that our puzzle is solved if and only if the resulting bipartite graphs are isomorphic to each other. Conversely, we can reduce the graph isomorphism problem on bipartite graphs with $2n$ vertices, which is known to be GI-complete [3], to the problem of solving a lattice puzzle of $2n$ pieces in polynomial time.

2.3 A hard case

In the original rule, two slots are matched to each other if exactly one of those is deep. Here, we relax the rule of our puzzle: two slots are matched to each other if *at least* one of those is deep. We call this a *relaxed lattice puzzle.*

Theorem 3. *The problem of solving the relaxed lattice puzzle is NP-complete.*

The theorem is given by showing a polynomial time reduction from the 3 SAT problem [1].

3 Conclusion

We have considered three variants of the lattice puzzle and clarified their computational complexities. However, it is still open for the original lattice puzzle.

Acknowledgements

We would like to thank Prof. Shuji Yamada for drawing the authors' attention to the lattice puzzle and Prof. Ryuhei Uehara for his valuable comments.

References

[1] M. R. Garey and D. S. Johnson: Computers and Intractability: A Guide to the Theory of NP-Completeness. W H Freeman & Co, 1979.

[2] Hirōka Yasunori: banjō machiya hinagata. Suharaya Mohei, 1770.

[3] V. N. Zemlyachenko, N. M. Korneenko, R. I. Tyshkevich: Graph isomorphism problem. Journal of Soviet Mathematics 29(4), pp. 1426–1481, 1985.

Matchstick Puzzles on a Grid

extended abstract

Yasuhiko Takenaga, Shohei Mishiba and Haruka Sugiyama
Department of Communication Engineering and Informatics,
Graduate School of Informatics and Engineering,
The University of Electro-Communications

1 Introduction

Matchstick puzzles are played worldwide by many people from children to adults. Though they include so many variations of puzzles, most of the matchstick puzzles take the form to move the indicated number of matchsticks in the initial placement so as to satisfy the given condition.

In this paper, we consider two kinds of matchstick puzzles of this form. In the puzzles, we assume that matchsticks can be placed only on the edges of a square grid. The first puzzle is a simple one whose objective is to make a given final placement of matchsticks. We show an algorithm to solve this problem faster than exhaustive search. It runs in $O(n^2(k \log k)^{1/2})$ time, where the size of the grid is $n \times n$ and k is the number of matchsticks to be moved. In the second puzzle, the objective is to make the indicated number of squares. It is one of the most typical style of matchstick puzzles. We show that the problem to decide if the puzzle has a solution is NP-complete. In such puzzles, there exists a tacit understanding that all the matchsticks must be included in at least one square. We also adopt this rule in this paper. In the following of this paper, we call a matchstick simply a stick.

2 Puzzles to Make the Final Placement

In this section, we consider the puzzles to make the given final placement of sticks by moving k sticks. An example of the puzzle is shown in Fig.1. We assume that both the initial and the final placements can be represented within a $n \times n$ grid. Note that the sticks can be moved outside of the $n \times n$ grid of the initial placement. That is, the final placement can be made using the region outside the initial placement. We assume that the initial and final placements include the same number of sticks and that the number of sticks is larger than k. We do not allow that the placement obtained by moving sticks is a one obtained by rotating the final placement. Even without the restriction, we have only to solve this problem four times. This problem can be solved by exhaustive search in $O(n^4)$ time.

The algorithm we propose is based on string matching with mismatches. This is the problem to decide, given two strings called the text and the pattern and a positive integer k, find the positions where the pattern matches the text within k errors. It is known that the problem can be solved in $O(n(k \log k)^{1/2})$ time, where n is the size of the text [1, 2].

In the proposed algorithm, we represent the initial and final placements as sequences T and P, respectively, by representing an edge by a character. If P matches T with $2k$ errors in legal positions, it means that the final placement can be obtained from the initial placement by moving k sticks.

Now we show how to construct T and P. In T and P, each edge is encoded by $0, 1$ or 2. It is encoded by 0 if no stick is placed, by 1 if a stick exists on a vertical edge and by 2 if a stick exists on a horizontal edge. In a grid, let a column be the series of vertical or horizontal edges arranged vertically. We encode a column by ordering the codes for edges from the top edge to the bottom edge. The sequences encoding columns are ordered from the leftmost column to the rightmost one. Both in T and P, place n 0's between the columns. In addition, add $8n^2 + 4n - 2$ 0's at the head and tail of T and add $4n^2 + 2n - 1$ 0's at the head and tail of P. The sequences T and P for Fig.1 are as follows.
T: 0^{38} 10 00 200 00 11 00 020 00 10 0^{38}
P: 0^{19} 11 00 000 00 01 00 002 00 11 0^{19}

Legal positions of P in T are $r(4n+1) \leq m \leq r(4n+1) + 2n$ $(0 \leq r \leq 2n)$, except 0 and $2n(4n+1) + 2n$. Each legal position corresponds to a position of the final placement on the initial placement. As no mismatch of 0 and 1 occurs in the positions, the number of mismatches is two times the number of sticks to be moved. As the size of T is $O(n^2)$ and the number of mismatches is $2k$, the algorithm runs in $O(n^2(k \log k)^{1/2})$ time.

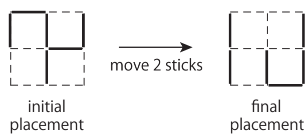


Figure 1: An example of the puzzle.

3 Puzzles to Make Squares

In this section, we consider the puzzles to make the designated number of squares by moving sticks and show that the puzzle is NP-complete. We call the following problem to decide if the problem has a solution STICK SQUARES.
Input: An initial placement of sticks on an $n \times n$ grid and positive integers k and l.
Question: Is it possible to make l squares by moving k sticks in the initial placement?
Note that we consider squares of any size and a stick may be included in more than one square. Also, as noted before, all the sticks must be included in at least one square after moving sticks.

Theorem 1. *STICK SQUARES is NP-complete.*

Sketch of proof. We prove NP-hardness by reduction from CircuitSAT. We construct the gadgets that correspond to the parts of a Boolean circuit. In addition, k vertical sticks are arranged horizontally outside the gadgets to simulate the Boolean circuit. There are three cells of the grid between these sticks. In our construction, all the sticks can be included in at least one square by moving these k sticks into the gadgets. In the following figures, sticks that are not included in any square are represented by bold lines.

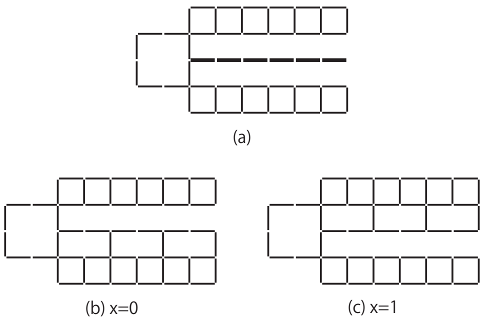


Figure 2: Input and wire gadgets.

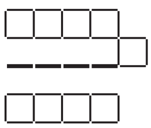


Figure 3: An output gadget.

Fig.2 (a) represents an input gadget and a wire gadget. The left part of the figure is an input gadget, in which the value of a variable is determined. By moving sticks into the gadgets, Fig.2 (b) and (c) represent the values 0 and 1 respectively. Fig.3 is an output gadget. All the sticks in the gadget can be included in squares by placing two sticks on the gadget if and only if the value from the left is 1.

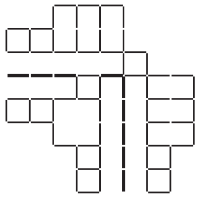


Figure 4: A bend gadget.

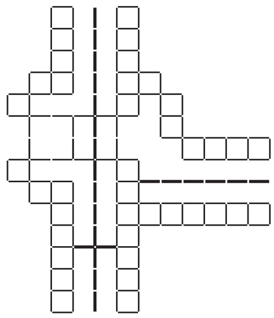


Figure 5: An AND gadget.

Bend gadgets (Fig.4) and adjustment gadgets are also used to represent wires. The former changes the direction of a wire and the latter adjusts the length of wires so that the wires can be connected to other gadgets. Logic elements are represented using NOT gadgets and AND gadgets. In an AND gadget (Fig.5) two values are given from the top and the bottom of the gadget and their logical product is output to the right.

The values of k and l are computed from the placement of the sticks simulating the Boolean circuit. It is impossible to make all the sticks included in squares if the sticks inside the gadgets are moved. It is because, if m sticks inside the gadgets are moved, at least $3m$ sticks must be moved to the outside of the gadgets. □

References

[1] A. Amir, M. Lewenstein and E. Porat, Faster algorithm for string matching with k mismatches, J. Algorithms, 50(2), pp.257-275, 2004.

[2] M. Nocolae and S. Rajasekaran, On string matching with mismatches, Algorithms, 8(2), pp.248-270, 2015.

Acknowledgement This work was partially supported by JSPS KAKENHI Grant Number 15K00505.

Facility Location Problems in Cycle Networks

Robert Benkoczi,[1] Binay Bhattacharya,[2] Ante Ćustić,[2] Sandip Das,[3] and Tiko Kameda[2]
[1]Univ. Lethbridge, Canada, [2]Simon Fraser Univ., Canada, [3]Indian Statist. Inst., Kolkata, India

***Abstract*—We present a simple $O(n \log n)$ time algorithm for finding a p-center in cycle networks. We also present an $O(n \log^2 n)$ (resp. $O(n \log^4 n)$) time algorithm for locating a k-sink in dynamic flow cycle networks with uniform (resp. non-uniform) edge capacities.**

I. Introduction

In a minmax-cost p-facility location problem, we want to place p facilities (e.g., centers, sinks) in such a way that each vertex can be serviced within the minimized cost (e.g., weighted distance, evacuation time). Recently, Wang and Zhang [10] presented an $O(n \log n)$ time algorithm for the p-center problem in tree networks, where n is the number of vertices, improving upon all the previous results, although an $O(n)$ time algorithm is known [2], [9], which is exponential in p. The fastest p-center algorithm for path networks known to date, due to Megiddo and Tamir [8], also takes $O(n \log n)$ time. This paper shows that the cycle networks also belong to this family of networks for which the p-center problem can be solved in $O(n \log n)$ time. Our simple algorithm applies the *sorted matrix method* of Frederickson [5], together with results by Chen et al. [4] and Hsu and Tsai [6].

Due to many recent disasters such as earthquakes, volcanic eruptions, typhoons, and nuclear accidents, evacuation planning is getting increasing attention. The k-sink problem is an attempt to model evacuation in such an emergency situation. Bhattacharya et al. [3] showed recently that the k-sink problem on dynamic flow path networks can be solved in $O(\min\{n \log^3 n, n \log n + k^2 \log^4 n\}$ time. When all edges have the same capacity, they show that it can be solved in $O(\min\{n \log n, n + k^2 \log^2 n\}$ time. In this paper we present an $O(n \log^2 n)$ (resp. $O(n \log^4 n)$) time algorithm for locating a k-sink in dynamic flow cycle networks with uniform (resp. non-uniform) edge capacities.

This paper shows that two similar approaches can be used to solve the center and sink problems in cycle networks.

II. Preliminaries

Let $C(V, E)$ be a cycle graph consisting of n vertices. Each vertex $v \in V$ has a nonnegative weight $w(v)$ and any portion of each edge has a nonnegative length. The clockwise (cw) path from point a to b on $C(V, E)$ is denoted by $C[a, b]$, and its distance is denoted by $d(a, b)$.

In the p-center problem, the *cost* (=weighted distance) of vertex v, at point x can be represented by a function $f_v(x) = d(v, x)w(v)$. Given a cost value λ, there is a maximal interval $I(v)$ containing v such that if a center is placed within it, v is *covered* with cost $\leq \lambda$. For point x, the cost function $f_v(x)$ thus consists of two linear half lines, if the path is represented as the horizontal axis, and the cost is represented by the vertical axis.

In the k-sink problem each edge $e \in E$ has *transit time* and *capacity*. Let λ^* denote the minimum cost to cover all vertices by a set of k facilities.

Lemma 1: (Kariv and Hakimi [7]) There exist two vertices v_a, v_b, and a facility q in an optimal facility allocation such that $f_{v_a}(q) = f_{v_b}(q) = \lambda^*$. ■

Lemma 2: (Hsu and Tsai [6]) Let $\mathcal{A}$ be a set of circular arcs along a cycle C. A minimum set of piecing points for $\mathcal{A}$ can be found in time linear in $|\mathcal{A}|$. ■

Let $\mathcal{H} = \{H_1, H_2, \ldots, H_m\}$ be a set of m upper half-planes. Given two indices i and j with $1 \leq i \leq j \leq m$, a *2D sublist LP query* [4] asks for the lowest point in the common intersection of $H[i, j] = \{H_i, H_{i+1}, \ldots, H_j\}$.

Lemma 3: (Chen et al. [4]) Assume that the intersections between the x-axis and the bounding lines of the half-planes in $\mathcal{H}$ are ordered from left to right according to the half-plane indices. Then, after $O(n \log n)$ preprocesing time, a 2D sublist LP query can be answered in $O(\log n)$ time. ■

III. Feasibility tests

Given a cost value λ, a λ-*feasibility test* decides if a given number of facilities can be located in such a way that each vertex v can be serviced by a facility within cost λ.

Given a cycle C, we remove an edge of C and construct a path P_C of length $2n - 1$ by concatenating two copies of the resulting path, and name the vertices on P_C as $v_1, \ldots, v_{2n}$ from left to right, where v_{n+i} is a copy of v_i. Let λ be a value provided by a 2D sublist LP query on P_C. In solving the p-center problem, for each vertex $v \in V$, we create a *demand interval* of length $2\lambda/w(v)$ centered at v on C. We then invoke Lemma 2 to test if all the demand intervals can be pierced by p piercing points (=centers). If so, the λ-feasibility test succeeds, otherwise it fails. Using Lemma 2, we can prove

Lemma 4: Given any value λ, we can test λ-feasibility for the p-center problem in $O(n)$ time. ■

Ben-Moshe et al. [1] have a more complicated feasibility test for weighted *cactus* networks that runs in $O(n)$ time.

In solving the k-sink problem, given cost λ, for each vertex $v \in V$, we need to find the farthest point $p_{cw}(v)$ (resp. $p_{ccw}(v)$) on the cw (resp. counter-cw (ccw)) side of v such that the evacuees from all vertices between v and $p_{cw}(v)$ (resp. $p_{ccw}(v)$) can evacuate to $p_{cw}(v)$ (resp. $p_{ccw}(v)$) within cost (=time) λ. To keep the cost within λ, we clearly need to place a sink in the demand interval $C[p_{ccw}(v), p_{cw}(v)]$, and Lemma 2 is applicable. If the edge capacities are uniform (resp. non-uniform), we can compute $p_{cw}(v)$ and $p_{ccw}(v)$ in $O(\log n)$ (resp. $O(\log^3 n)$) time using the method employed in [3], with $O(n)$ (resp. $O(n \log n)$) preprocessing time.

Lemma 5: Given any value λ, we can test λ-feasibility for the k-sink problem with uniform (resp. non-uniform) edge capacities in $O(n \log n)$ (resp. $O(n \log^3 n)$) time. ■

IV. Optimization algorithms

In the optimization phase, we generate candidate costs (λ-values). We look for the smallest λ, λ^*, for which the problem instance is λ-feasible. In a *sorted matrix* [5], the elements in each row and each column are sorted in the non-decreasing order. Frederickson proved

Lemma 6: [5] Given an $n \times n$ sorted matrix $M_{n\times n}$, suppose that $M[i,j]$, for any $1 \le i, j \le n$, can be computed in $g(n)$ time, and $M[i,j]$-feasibility can be tested in $f(n)$ time. Then we can solve the optimal k-facility problem in $O(h(n) + ng(n) + f(n)\log n)$ time, where $h(n)$ is the preprocessing time. ■

Let $M[i,j]$, $1 \le i \le j \le 2n$, be the minimum cost of a facility that services the vertices from v_i to v_j on P_C. If we set $M[i,j] = 0$ for $i \ge j$ and $M[i,j] = \lambda_\infty$ (where λ_∞ is larger than any actual cost) for $j \ge i+n$, then $M_{2n\times 2n}$ is a sorted matrix. We apply Lemma 3 to generate candidate λ values. Note that P_C gives rise to λ values that do not exist in cycle C, but it does no harm. For the p-center problem, we have $h(n) = O(n\log n)$, $g(n) = O(\log n)$ by Lemma 3, and $f(n) = O(n)$ by Lemma 4, hence $h(n) + f(n)\log n + ng(n) = O(n\log n)$.

Theorem 7: We can solve the p-center problem on cycle networks in $O(n\log n)$ time. ■

Regarding the k-sink problem, we introduce the *upper envelopes tree* (or *UE tree*), $\mathcal{T}$, with root ρ, whose leaves are the vertices of P_C, arranged from left to right. It is a balanced tree with height $O(\log n)$. For a node u of $\mathcal{T}$, let $v_L(u)$ (resp. $v_R(u)$) denote the leftmost (resp. rightmost) vertex on P_C that belongs to $\mathcal{T}(u)$ (the subtree rooted at u). We say that node u *spans* subpath $P_C[v_L(u), v_R(u)]$. At node u we store $v_L(u)$, $v_R(u)$ and four other pieces of data, as explained in the next paragraph.

For an arbitrary index pair $i \le j$, there are $O(\log n)$ nodes which together span $P_C[v_i, v_j]$. Let $\mathcal{P}[v_i, v_j]$ denote the set of maximal subpaths spanned by such nodes, which can be found in $O(\log n)$ time from $\mathcal{T}$. Let $P_C[v_L(u), v_R(u)] \in \mathcal{P}[v_i, v_j]$ for some node u. Assume that all the evacuees from $P_C[v_L(u), v_R(u)]$ move cw past $v_R(u)$. Let v_k be the vertex lying between $v_L(u)$ and $v_R(u)$, inclusive, such that there is a break in the flow of evacuees out of $v_R(u)$ just before the first evacuee from v_k is seen, after which a stream of all the evacuees from $P_C[v_L(u), v_k]$ are observed without any break. Similarly, assume that all the evacuees from $P_C[v_L(u), v_R(u)]$ move ccw past $v_L(u)$. Define v_l symmetrically to v_k. At u we store v_k, v_l, and the total weights of the vertices on $P_C[v_L(u), v_k]$ and those on $P_C[v_l, v_R(u)]$.

Let v_{k1} (resp. v_{k2}) be the v_k-vertex stored in the left (resp. right) child of a non-leaf node of $\mathcal{T}$. Then it is known in the uniform edge capacity case that either $v_k = v_{k1}$ or $v_k = v_{k2}$ [3]. Similarly, the v_l-vertex is either v_{l1} or v_{l2} of the child nodes.

Lemma 8: [3] If the edge capacities are uniform, we can construct UE tree $\mathcal{T}$, together with the four pieces of data (v_k, v_l, and two weights) at each node, in $O(n)$ time. ■

For each node $v_i \in V$, climb $\mathcal{T}$ from v_i along path $\pi(v_i, \rho)$ from v_i to ρ, and let u be the current node being visited. Based on the data stored at u and all the nodes visited so far, compute the current cost of the vertices at point x just on the cw side of $v_R(u)$ and compare it with λ. If λ is larger, we continue the ascent, otherwise we start a descent. We will eventually find the edge containing a point at which the cost equals λ and compute the exact position where they are the same. This point is the cw endpoint of the demand interval associated with vertex v_i, and it can be found in $O(\log n)$ time. The ccw endpoint of the λ-demand interval associated with v_i can be found similarly in $O(\log n)$ time. Thus the time needed to find the λ-demand intervals for all the vertices is $O(n\log n)$. We now resort to Lemma 2 to test if k piercing points can pierce all the intervals, which takes $O(n)$ time. If "Yes," then the λ-feasibility test succeeds, otherwise it fails.

We have shown that $h(n) = O(n)$ and $f(n) = O(n\log n)$ for cycle networks with uniform edge capacities. We can show $g(n) = O(\log n)$, as in [3]. If the edge capacities are non-uniform, we can show that $h(n) = O(n\log n)$ and $g(n) = O(\log^3 n)$, as in [3], and we have $f(n) = O(n\log^3 n)$, since its major component is the time required to find the $O(n)$ demand intervals, spending $O(\log^3 n)$ time per vertex. The test itself takes only $O(n)$ time by Lemma 2. Lemma 6 thus implies

Theorem 9: We can solve the k-sink problem in dynamic flow cycle networks with uniform (resp. non-uniform) edge capacities in $O(n\log^2 n)$ (resp. $O(n\log^4 n)$) time. ■

Concluding remarks

The currently most efficient p-center algorithm for cactus networks runs in $O(n^2)$ time [1]. A *unicycle* graph is a special cactus graph, which contains just one cycle. We can combine our p-center algorithm for cycle networks presented here with that for tree networks in [10] to find a p-center in unicycle networks in $O(n\log n)$ time.

References

[1] Ben-Moshe, B., Bhattacharya, B., Shi, Q., Tamir, A.: Efficient algorithms for center problems in cactus networks. Theoretical Compter Science 378(3), 237–252 (2007)

[2] Bhattacharya, B., Shi, Q.: Optimal algorithms for the weighted p-center problems on the real line for small p. In: Proc. Workshop on Algorithms and Data Structures (WADS), Springer-Verlag. vol. LNCS 7434, pp. 529–540 (2007)

[3] Bhattacharya, B., Golin, M., Higashikawa, Y., Kameda, T., Katoh, N.: Improved algorithms for computing k-sink on dynamic flow path networks. In: Proc. Algorithms and Data Structures Symp., Springer-Verlag, LNCS 10389. pp. 133–144 (2017)

[4] Chen, D.Z., Li, J., Wang, H.: Efficient algorithms for the one-dimensional k-center problem. Theor. Comp. Sci. 592, 135–142 (August 2015)

[5] Frederickson, G.: Optimal algorithms for tree partitioning. In: Proc. 2nd ACM-SIAM Symp. Discrete Algorithms. pp. 168–177 (1991)

[6] Hsu, W.L., Tsai, K.H.: Linear time algorithms on circular-arc graphs. Information Processing Letters 40, 123–129 (1991)

[7] Kariv, O., Hakimi, S.: An algorithmic approach to network location problems, Part I: The p-centers. SIAM J. Appl. Math. 37, 513–538 (1979)

[8] Megiddo, N., Tamir, A.: New results on the complexity of p-center problems. SIAM J. Comput. 12, 751–758 (1983)

[9] Shi, Q.: Efficient algorithms for network center/covering location optimization problems. Ph.D. thesis, School of Computing Science, Simon Fraser University, Canada (2008)

[10] Wang, H., Zhang, J.: An $O(n\log n)$-time algorithm for the k-center problems in trees (2017), `arXiv:1705.02752v1 [cs.DS]`

The partial sum dispersion problem on the line

Toshihiro Akagi[1,a)] Tetsuya Araki[2,b)] Shin-ichi Nakano[1,c)]
Hiroshi Ishikawa[2,d)]

Abstract:
The dispersion problem is a variant of the facility location problem. In this paper we design some algorithms for some dispersion problems.
Given a set P of n points on the horizontal line and an integer k we wish to find a subset S of P such that $|S| = k$ and maximizing the cost $\min_{x \in S}\{cost(x)\}$, where $cost(x)$ is the sum of the distances from x to the nearest c points in S. The problem is the dispersion problem with pertial c sum cost and we write it as PcS-dispersion problem. In this paper we give a simple $O(kn^2 \log n)$ time algorithm to solve the P2S-dispersion problem and $O(kn^{c+1})$ time algorithm to solve the PcS-dispersion problem.

1. Introduction

The facility location problem and many of its variants have been studied [4], [5]. A typical problem is to find a set of locations to place facilities with the designated cost minimized. By contrast, in this paper we consider the dispersion problem (or obnoxious facility location problem) ,which finds a set of locations with a certain objective function maximized.

Given a set P of n possible locations, and the distance d for each pair of locations, and an integer k with $k \leq n$, we wish to find a subset $S \subset P$ with $|S| = k$ such that some designated objective function is maximized [1], [2], [3], [7], [8], [9], [10], [11], [12].

The intuition of the problem is as follows. Assume that we are planning to open several chain stores in a city. We wish to locate the stores mutually far away from each other to avoid self-competition. So we wish to find k locations so that some objective function based on the distance is maximized. See more applications, including *result diversification*, in [8], [9], [10].

In one of basic cases the objective function to be maximized is the minimum distance between two points in S. Then papers [9], [11] show if P is a set of points on the plane then the problem is NP-hard, and if P is a set of points on the line then the problem can be solved in $O(\max\{n \log n, kn\})$ time [9] by dynamic programming approach, and in $O(n)$ time by the sorted matrix search method [6].

In this paper we consider the following problem [8]. Given a set P of n points on the horizontal line and an integer k we wish to find a subset S of P such that $|S| = k$ and maximizing the cost $\min_{x \in S}\{cost(x)\}$, where $cost(x)$ is the sum of the distances from x to the nearest c points in S. The problem is the dispersion problem with pertial c sum cost and we write it as PcS-dispersion problem. Intuitively this cost models competition to the nearest c stores. Then the cost $cost(S)$ of S is the minimum cost among the costs of the points in S, which is $\min_{x \in S}\{cost(x)\}$. In the case of $c = 2$, some experimental results (for more general problems) are known. See [8].

In this paper we design an algorithm to solve the P2S-dispersion problem by dynamic programming approach if all points of P are on a line. The running time of the algorithm is $O(kn^2 \log n)$. Similarly, we design an algorithm to solve the PcS-dispersion problem if all points of P are on a line. The running time of the algorithm is $O(kn^{c+1})$.

2. P2S-dispersion problem

In this section we design an algorithm to solve the P2S-dispersion problem, based on dynamic programming approach, if all points of P are on the horizontal line. We define the subproblem $P2S(h, i; k)$ as follows.

Let P_i be the subset of the points in P locating on the left of $p_i \in P$ including p_i, where p_i is the i-th point from left in P. Given $p_h \in P_i$ and an integer $k \geq 3$, we wish to find a subset $S \subset P_i$ such that (1) $|S| = k$ and (2) the rightmost two points in S are p_h and p_i, with $h < i$, (3) maximizing $cost(S)$. This is the subproblem $P2S(h, i; k)$. We denote by $cost(h, i; k)$ the optimal cost of a solution of $P2S(h, i; k)$. This is the P2S-dispersion problem with the rightmost two points in S are designated.

We can observe that $P2S(h, i; k)$ has a solution S containing the leftmost and rightmost points in P_i. Thus we can assume $p_1, p_i \in S$.

We have the following lemma.

Lemma 1. *Let S be a solution of $P2S(h, i; k)$, and p_h, p_i the rightmost two points in S. Then the following (a)–(c) holds. (a) The two nearest points of $p_i \in S$ are located on*

[1] Department of Computer Science, Gunma University, Kiryu, 376–8515, Japan
[2] Faculty of System Design, Tokyo Metropolitan University, Hino, Tokyo 191–0065, Japan
a) akagi@nakano-lab.cs.gunma-u.ac.jp
b) araki@tmu.ac.jp
c) nakano@cs.gunma-u.ac.jp
d) ishikawa-hiroshi@tmu.ac.jp

the left of p_i*, (b) The two nearest points of* $p_h \in S$ *are located either on the left of* p_h*, or one on the left and one on the right (it is* p_i*), (c)* $cost(p_h) < cost(p_i)$*.*

Proof. (a)(b) Immediately. (c) Let $p_{h'}$ be the 3rd rightmost point in S. Then $cost(p_h) \le d(p_{h'}, p_h) + d(p_h, p_i) < d(p_{h'}, p_i) + d(p_h, p_i) = cost(p_i)$. Note that $d(p, q)$ is the distance between points p and q in P. □

Thus when we compute $cost(h, i; k)$ which is the minimum over $cost(x)$ for $x \in S$, we can ignore p_i since $cost(p_i) > cost(p_h)$. The cost $cost(h, i; k)$ is $\max_{h'=1,2,\cdots,h-1} \min\{cost(h', h; k-1), d(p_{h'}, p_i)\}$ and we can compute it in $O(n)$ time. The number of the subproblems is at most kn^2 and we can solve each subproblem in $O(n)$ time.

The entire algorithm **find-P2S-dispersion**(P, n, k) is shown below.

Algorithm 1 find-P2S-dispersion(P, n, k)

```
% Compute P(h, i; 3)      (Case k = 3)
for i = 3, 4, ..., n do
  for h = 2, 3, ..., i − 1 do
    cost(h, i; 3) = d(p_1, p_i)
  end for
end for
% Compute P(h, i; k)      (Case k > 4)
for k' = 4, 5, ..., k do
  for i = k', k' + 1, ..., n do
    for h = k' − 1, k', ..., i − 1 do
      cost(h, i; k') = 0
      % Compute the maximum cost
      for h' = k' − 2, k' − 1, ..., h − 1 do
        cost(h, i; k') = max{cost(h, i; k'), min{cost(h', h; k' −
        1), d(p_h', p_i)}}
      end for
    end for
  end for
end for
% Compute the optimal cost
cost = 0
for h = k − 1, k, ..., n − 1 do
  if cost(h, n; k) > cost then
    cost = cost(h, n; k)
  end if
end for
Output cost
```

Theorem 1. *One can solve the P2S-dispersion problem in* $O(kn^3)$ *time.*

We can prove that $cost(h', h; k-1)$ is a non-decreasing function with respect to h'. Then $\min\{cost(h', h; k-1), d(h', i)\}$ is a non-decreasing function with respect to h' up to some points, then it is a decreasing linear function with respect to h', so we can find the maximum one by binary search in $O(\log n)$ time.

We have the following thorem.

Theorem 2. *One can solve the P2S-dispersion problem in* $O(kn^2 \log n)$ *time.*

3. PcS-dispersion problem

In this section we design an algorithm to solve the PcS-dispersion problem, based on dynamic programming approach, if all points of P are on the horizontal line. We define the subproblem $PcS(h_{c-1}, h_{c-2}, \cdots, h_1,\ i; k)$ as follows.

Let P_i be the subset of the points in P locating on the left of $p_i \in P$ including p_i, where p_i is the i-th point from left in P. Given $p_{h_{c-1}}, p_{h_{c-2}}, \cdots, p_{h_1} \in P_i$ and an integer $k \ge c$, we wish to find a subset $S \subset P_i$ such that (1) $|S| = k$ and (2) the rightmost c points in S are $p_{h_{c-1}}, p_{h_{c-2}}, \cdots, p_{h_1}$ and p_i, with $h_{c-1} < h_{c-2} < \cdots < h_1 < i$, (3) maximizing $cost(S)$. This is the subproblem $PcS(h_{c-1}, h_{c-2}, \cdots, h_1,$ $i; k)$.

The number of the subproblems is at most kn^c and we can solve each subproblem in $O(n)$ time. Thus we can solve the PcS-dispersion problem in $O(kn^{c+1})$ time.

Theorem 3. *One can solve the PcS-dispersion problem in* $O(kn^{c+1})$ *time.*

4. Conclusion

In this paper we gave an algorithm for the P2S-dispersion problem. The running time of the algorithm is $O(kn^2 \log n)$. Also we gave an algorithm to solve the PcS-dispersion problem. The running time of the algorithm is $O(kn^{c+1})$.

Acknowledgements

This work was partly supported by Tokyo Metropolitan University Grant-in -Aid for Research on Priority Areas“Research on social big data.”

References

[1] T. Akagi and S. Nakano, Dispersion on the Line, Technical Report, 2016-AL-158-3, IPSJ (2016)

[2] C. Baur and S.P. Feketee, Approximation of Geometric Dispersion Problems, Pro. of APPROX '98, Pages 63-75 (1998).

[3] B. Chandra and M. M. Halldorsson, Approximation Algorithms for Dispersion Problems, J. of Algorithms, 38, pp.438-465 (2001).

[4] Z. Drezner, Facility Location: A Survey of Applications and Methods, Springer (1995).

[5] Z. Drezner and H.W. Hamacher, Facility Location: Applications and Theory, Springer (2004).

[6] G. Frederickson, Optimal Algorithms for Tree Partitioning, Proc. of SODA '91 Pages 168-177 (1991).

[7] R. Hassin, S. Rubinstein and A. Tamir, Approximation Algorithms for Maximum Dispersion, Operation Research Letters, 21, pp.133-137 (1997).

[8] T. L. Lei and R. L. Church, On the Unified Dispersion Problem: Efficient Formulations and Exact Algorithms, European Journal of Operational Research, 241, pp.622-630 (2015).

[9] S. S. Ravi, D.J. Rosenkrantz and G. K. Tayi, Heuristic and Special Case Algorithms for Dispersion Problems, Operations Research, 42, pp.299-310 (1994).

[10] M. Sydow, Approximation Guarantees for Max Sum and Max Min Facility Dispersion with Parameterised Triangle Inequality and Applications in Result Diversification, Mathematica Applicanda, 42, pp.241-257 (2014).

[11] D. W. Wang and Yue-Sun Kuo, A study on Two Geometric Location Problems, Information Processing Letters, 28, pp.281-286 (1988).

[12] T. Akagi, T. Araki and S. Nakano, The LR-dispersion Problem, LA Symposium (2017).

On Equivalence of de Bruijn Graphs and State-minimized Finite Automata

Yoshiaki TAKAHASHI † Akira ITO ‡

† Solar, Hofu Science Museum 6-41 Kotobukichou, Hofu-shi, Yamaguchi, 747-0809 Japan

‡ Faculty of Engineering, Yamaguchi University 2-16-1 Tokiwadai, Ube-shi, Yamaguchi, 755-8611 Japan

E-mail: † solar4@minos.ocn.ne.jp, ‡ akito@yamaguchi-u.ac.jp

Abstract A de Bruijn sequence of order n on the alphabet {0,1} is a cyclic sequence in which every possible string of length n over {0,1} occurs exactly once. In order to enumerate all possible de Bruijn sequences, de Bruijn graphs were introduced.

This study investigates the relationship between de Bruijn graphs and finite automata and proves the structural equality of de Bruijn graph of order n and the state transition diagram for minimum state deterministic finite automaton which accepts regular language $(0+1)^*1(0+1)^{n-1}$. We also extend this result naturally to the *k*-ary de Bruijn graphs for arbitrary *k*'s by introducing coloring finite automata whose accepting states are refined with two or more colors.

Keywords de Bruijn sequence, de Bruijn graphs, finite automata, state-minimization

1. de Bruijn graph and de Bruijn sequence

Definition.1 A directed graphs defined as follows is called a *k*-ary order *n* de Bruijn graph and abbreviated $DB_{k,n}$.

$$V=\{\{0,1,\cdots,k-1\}^n\}=\{0,1,\cdots,k^n-1\}$$

$$E=\{(b_1b_2\cdots b_n, b_1'b_2'\cdots b_n') \mid b_i,b_i'\in\{0,1\}, i=1,\cdots,n, b_2=b_1', b_3=b_2',\cdots,b_n=b_{n-1}'\}$$

$$=\{(x,(kx+i)\bmod k^n)\mid x\in V, i=0,1,\cdots,k-1\}$$

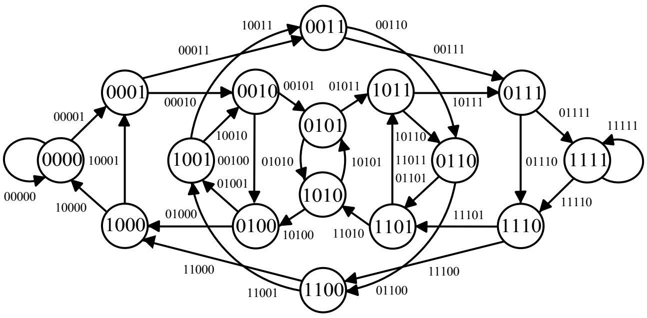


Fig.1 The de Bruijn graph $DB_{2,4}$.

An example $DB_{2,4}$ is shown in Fig.1.

2. Finite automaton which accepts regular language $(0+1)^*1(0+1)^{n-1}$.

We consider the language $L_n=\{x\in\{0,1\}^* \mid$ the nth symbol from the end of x is 1$\}$, i.e., the set of strings over {0,1} whose *n*th symbols from their right ends are 1's. Nondeterministic and deterministic finite automaton (NFA and DFA) accepting L_n are abbreviated N_n, D_n, respectively. N_4 is illustrated in Fig.2.

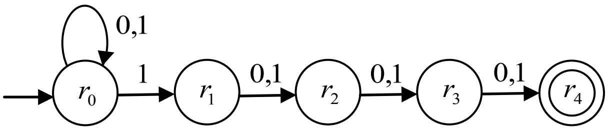


Fig.2 NFA N_4 accepting L_4.

By using the subset construction method, N_4 is converted to D_4 as depicted in Fig.3. One can tell that D_4 and $DB_{2,4}$ are structurally equivalent.

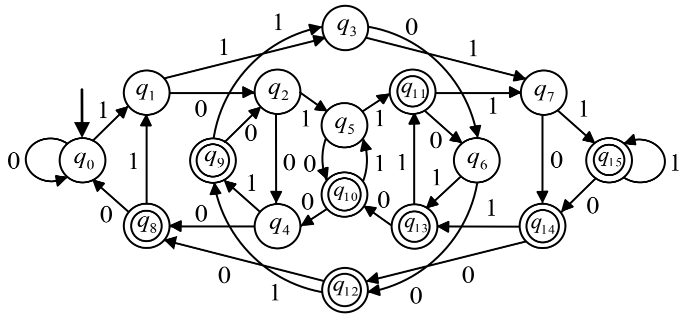


Fig.3 DFA D_4 accepting L_4.

3. Equivalence of $DB_{2,n}$ and D_n

Theorem.1 The graph structure of D_n is isomorphic to $DB_{2,n}$.

It is known that any DFA accepting L_n requires more than or equal to 2^n states.

Example.1 Fig.5 shows the final stage of N_2-D_2 conversion by using the subset construction method.

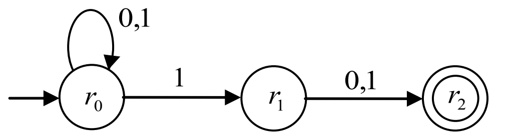


Fig.4 NFA N_2 accepting L_2.

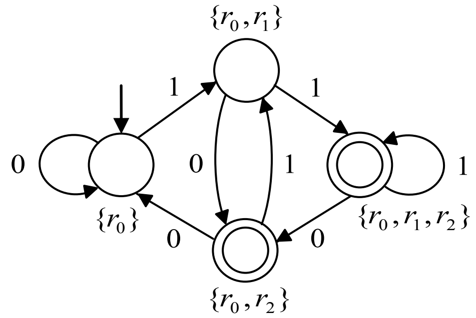


Fig.5 The final stage of N_2-D_2 conversion.

In the following, $[b_0,b_1,b_2]$ denotes 0-1 sequence (characteristic function) which represents a subset of $\{r_0,r_1,r_2\}$.

From Fig.4, the destinations from $\{r_0,r_1\}=[110]$ can be decided as follows: One is $\{r_0,r_2\}=[101]$ if input 0, because there are edges from r_0 to r_0 and from r_1 to r_2. The other is $\{r_0,r_1,r_2\}=[111]$ if input is 1, because there are edges from r_0 to r_0 and r_1 and r_1 to r_2. Doing these kinds of work for all subset of N_2, we can get the following D_2.

$$D_2=(\{[100],[101],[110],[111]\},\{0,1\},\delta',[100],\{[101],[111]\}),$$

For $b_1,b_2\in\{0,1\},a\in\{0,1\}$,

$$\delta'([1b_1b_2],a)=\begin{cases}[10b_1], \text{ if } a=0,\\ [11b_1], \text{ if } a=1.\end{cases}$$

By the left/right inversion of $[b_1,b_2]$ parts of states, the above description of D_2 can be rewritten to the following.

$$D_2=(\{00,10,01,11\},\{0,1\},\delta',00,\{01,11\}),$$

For $b_1,b_2\in\{0,1\},a\in\{0,1\}$

$$\delta'(b_2b_1,a)=\begin{cases}b_10, \text{ if } a=0,\\ b_11, \text{ if } a=1.\end{cases}$$

The above description of D_2 is identical to the description of $DB_{2,2}$ in Definition.1:

$$V=\{00,01,10,11\}$$
$$E=\{(00,00),(00,01),(01,10),(01,11),(10,00),(10,01),(11,10),(11,11)\}$$

As shown above, the state transitions of D_2 correspond to those of two-stage shift register provided with 0 or 1 sequential input (the value of register $r_i=1 \Leftrightarrow$ the state of D_2 contains r_i)

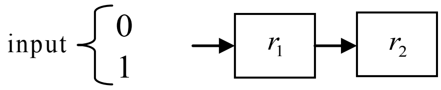


Fig.6 The two-stage shift register correspondent to N_2-D_2 conversion.

4. Generalization to *k*-ary

In this section, we investigate finite automata accepting the language over $\{0,1,\cdots,k-1\}$ $L_{k,n}=\{x\in\{0,1,\cdots,k-1\}^*\mid$ the *n*th symbol of *x* from its right end is $1,2\cdots$, or $k-1\}$.

Fig.7 is the state transition diagram of NFA $N_{3,2}$ accepting $L_{3,2}$. Fig.8 shows DFA $D_{3,2}$ converted from $N_{3,2}$ by using the subset construction method. One can tell that the de Bruijn graph $DB_{3,2}$ defined in Section.1 is identical to $D_{3,2}$ excluding its labels.

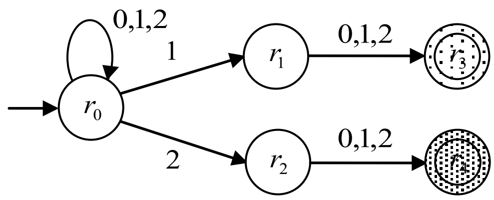


Fig.7 NFA $N_{3,2}$ accepting $L_{3,2}$.

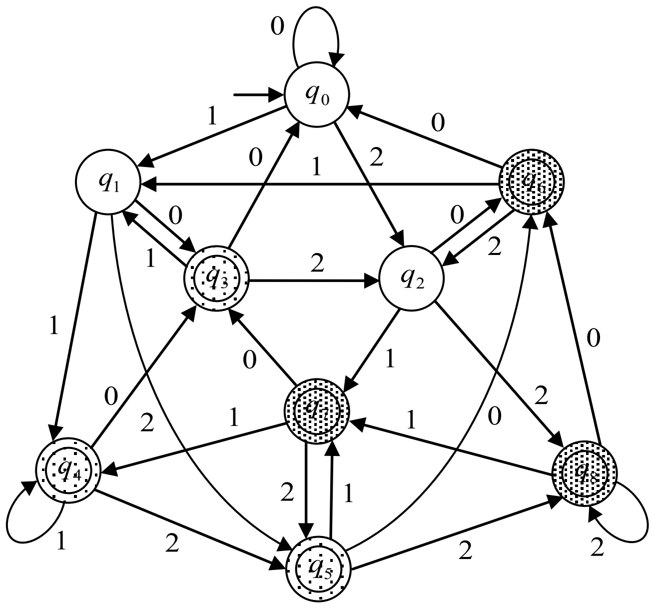


Fig.8 DFA $D_{3,2}$ accepting $L_{3,2}$.

However, the number of states of this DFA is not minimal: $D_{3,2}$ is simplified to the shape of Fig.5 by using the well-known minimization algorithm for DFAs. In order to extend the equivalence of *k*=2 case naturally to higher radix *k*, we take automata have classifying function of the features of input strings in addition to the conventional function of either accepting or non-accepting.

We call such a automaton a coloring automaton. Regarding $N_{k,n}$ depicted in Fig.7 as a nondeterministic coloring finite automaton $N'_{k,n}$ whose accepting states have two different colors (*k*=3) and applying the subset construction method to it, we get the deterministic coloring finite automaton $D'_{k,n}$ of Fig.8 whose two colors are depicted in the figure. Then, we can conclude that $DB_{k,n}$ and $D'_{k,n}$ are isomorphic for $k\geq 3$. We can also assert the minimality of $D'_{k,n}$.

Primary references

[1] D.Dn, F.Cao,and.F.Hsu, "De Bruijn digraphs,kautz digraphs, and their generalizations, in combinatorial network theory, ed. Du and D.F. Hsu, pp.65-96, Kluwer Academic , the Netherlands, 1996.

[2] G.Castinglione, A.Restivo, M.Sciortino, "Nondeterministic Moore automata and Brozozowski's minimization", Theoretical Computer Science 450 81-91, 2012.

On the Total Vertex Irregularity Strength for Trees with Many Vertices of Degree 2

Susilawati [1,2], E. T. Baskoro and Rinovia Simanjuntak[3]

[1,2,3] *Combinatorial Mathematics Research Group,*
Faculty of Mathematics and Natural Sciences,
Institut Teknologi Bandung
Bandung, Indonesia

Abstract

For a simple graph $G = (V, E)$, we define an injection $\phi : V(G) \cup E(G) \to \{1, 2, 3, \ldots, k\}$ to be a vertex irregular total k-labeling of G if for every two different vertices x and y of $G, wt(x) \neq wt(y)$, where $wt(x) = \phi(x) + \sum_{xy \in E(G)} \phi(xy)$. The minimum k for which the graph G has a vertex irregular total k-labeling is called the *total vertex irregularity strength* of G, denoted by $tvs(G)$.

Let T be a tree. Nurdin, Baskoro, Salman and Gaos (2010) conjectured that $tvs(T) = max\{t_1, t_2, t_3\}$, where $t_i = \lceil (1 + \sum_{j=1}^{i} n_i)/(i + 1) \rceil$ and n_i is the number of vertices of degree $i \in [1, 3]$. In this paper, we determine the exact value of the total vertex rregularity strength for trees with many vertices of degree two. We provide three possible values of total vertex irregularity strength of these trees. We give sufficient conditions for such trees to have the above mentioned total vertex irregularity strength. This paper adds further support to Conjecture Nurdin *et.al* (2010) by showing that such tree has total vertex irregularity strength equal to t_1, t_2, t_3.

Keywords: Irregularity strength, total vertex irregularity strength, subdivision, degree, tree.

---

[1] Ph.D student of mathematics Institut Teknologi Bandung
[2] Email: susilawati_nurdin@students.itb.ac.id
[3] Email: {ebaskoro,rino}@math.itb.ac.id

Cordial sets of Honeycomb networks

Suhadi Wido Saputro

Institut Teknologi Bandung, Indonesia

Let G be a graph. A binary labeling $f : V(G) \to \mathbb{Z}_2$ is called a *friendly labeling* if the absolute difference between the number of vertices labeled 0 and 1 is at most one. The friendly vertex labeling induces an edge labeling $f_* : E(G) \to \mathbb{Z}_2$ defined by $f_*(xy) = |f(x) - f(y)|$, $\forall xy \in E(G)$. Let $e_f(i)$ be the number of edges of G labeled i. The *cordial set* of the graph G, denoted by $C(G)$, is defined by

$$C(G) = \{|e_f(1) - e_f(0)| \mid f \text{ is a friendly vertex labeling of } G\}.$$

If 0 or 1 is belong to $C(G)$, then the graph G is called a *cordial graph.* In this paper, we show that honeycomb network graphs are cordial graph. We also determine all values in the cordial set of a honeycomb network. (*This talk is based on a joint work with Abdurrahman Shofy Adianto.*)