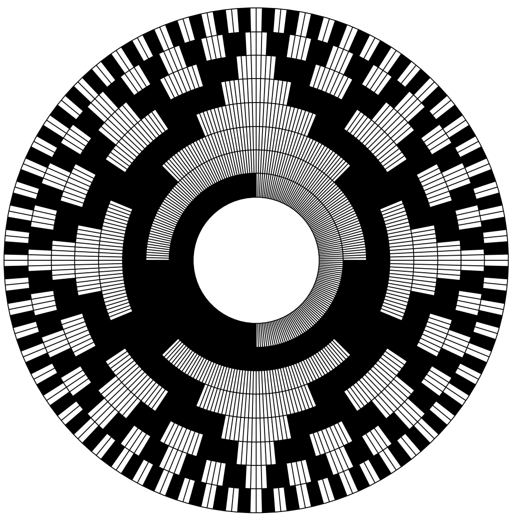
Open problems booklet

Combinatorics, Algorithms,
and Geometry Workshop 2024

Dresden



1: Triangulations of products of simplices

(suggested by Hung P. Hoang)

Products of simplices are important polytopes that have multiple applications to game theory, optimisation, and algebraic geometry. Their triangulations can be encoded by a collection of bipartite spanning trees, whose degree vectors are distinct.

Some formal detail with questionable relevance. See Figure 1 for an illustration.

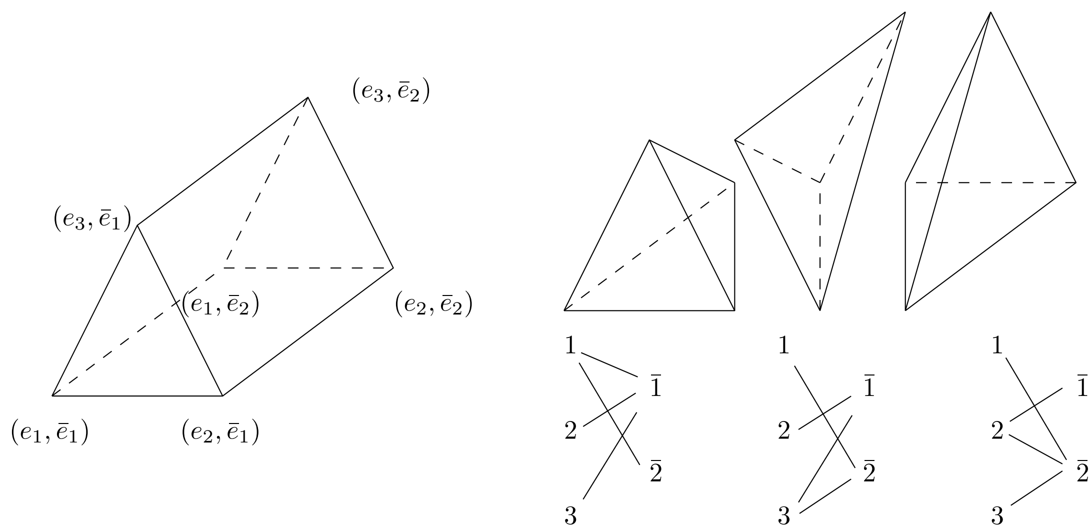


Figure 1: $\Delta_2 \times \Delta_1$ *(left) and one of its triangulations and the corresponding bipartite spanning tree collection (right)*

Let e_i denote the standard unit vector in $\mathbb{R}^n$. The simplex Δ_{n-1} is defined as the convex hull of $e_1, \dots, e_n$. The product of two simplices is defined as

$$\Delta_{n-1} \times \Delta_{d-1} := \operatorname{conv}\{(e_i, \bar{e}_j) \mid i = [n], j \in [d]\} \subset \mathbb{R}^{n+d}.$$

Note that $\dim(\Delta_{n-1} \times \Delta_{d-1}) = n + d - 2$.

For a polytope P with $\dim(P) = k$, a *triangulation* of P is a decomposition of P into copies of Δ_k such that every vertex of each copy is a vertex of P, and the intersection of any two copies is a simplex with dimension lower than k.

We can characterise a triangulation of a product of simplices combinatorially as follows.

Theorem 1 ([1]). $\{G_1, \dots, G_k\}$ *encodes a triangulation of* $\Delta_{n-1} \times \Delta_{d-1}$ *if and only if:*

- *Each* G_I *is a spanning tree of* $K_{n,d}$*;*
- *For each edge* $e \in G_i$*, either* $G_i \setminus e$ *has an isolated vertex or is a subgraph of some* G_j*;*
- *If* G_i*,* G_j *contain a perfect matching on the same vertex set, then it is the same matching.*

There is a correspondence between triangulations of $\Delta_{n-1} \times \Delta_2$ and lorenzo tilings (i.e., a tilling of an equilateral triangle with side length n with n triangles and $(n^2 - n)/2$ rhombi of side length one) [5].

Flip graph of triangulations. We can define a notion of *flip* of a triangulation of products of simplices. Very roughly speaking, we remove some face(s) and insert some other face(s) to obtain another triangulation. The flip graph for $\Delta_{n-1} \times \Delta_1$ is isomorphic to the skeleton of the n-dimensional permutahedron [2, Section 6.2]. The flip graphs for $\Delta_{n-1} \times \Delta_2$ and $\Delta_{n-1} \times \Delta_3$ are connected [5, 4], while for sufficiently large n, the flip graph for $\Delta_{n-1} \times \Delta_4$ is not [3].

> **Question:** Are the flip graphs of triangulations of $\Delta_{n-1} \times \Delta_2$ and $\Delta_{n-1} \times \Delta_3$ Hamiltonian?

References

[1] Ardila, F., and Billey, S. Flag arrangements and triangulations of products of simplices. *Adv. Math. 214*, 2 (2007), 495–524.

[2] De Loera, J. A., Rambau, J., and Santos, F. *Triangulations*, vol. 25 of *Algorithms and Computation in Mathematics*. Springer-Verlag, Berlin, 2010. Structures for algorithms and applications.

[3] Liu, G. A zonotope and a product of two simplices with disconnected flip graphs. *Discrete Comput. Geom. 59*, 4 (2018), 810–842.

[4] Liu, G. Flip-connectivity of triangulations of the product of a tetrahedron and simplex. *Discrete Comput. Geom. 63*, 1 (2020), 1–30.

[5] Santos, F. The Cayley trick and triangulations of products of simplices. In *Integer points in polyhedra—geometry, number theory, algebra, optimization*, vol. 374 of *Contemp. Math.* Amer. Math. Soc., Providence, RI, 2005, pp. 151–177.

2: Flips in colorful triangulations

(suggested by Torsten Mütze)

The well-known *associahedron* G_n is the flip graph of triangulations of a convex n-gon, where any two triangulations that differ in replacing one diagonal are connected by an edge; see Figure 2. It is well-known that G_n admits a Hamilton cycle for all $n \geqslant 5$ [2].

For this problem, we consider a subgraph of G_n, obtained by coloring the points of the n-gon red and blue alternatingly, and by only considering *colorful* triangulations, i.e., triangulations without monochromatic (all-red or all-blue) triangles. We refer to the resulting subgraph of G_n as H_n. Bruce Sagan [3] proved that H_n is a connected graph, and he asked (personal communication) whether it admits a Hamilton cycle for sufficiently large n. Interestingly, as one can see in Figure 2, H_6 neither admits a Hamilton cycle nor path, and H_7 admits a Hamilton path, but not cycle. However, one can indeed show that H_n has a Hamilton cycle for all even $n \geqslant 8$ (by combining the results from [1] with the laceability of hypercubes; see the remarks after Question 2).

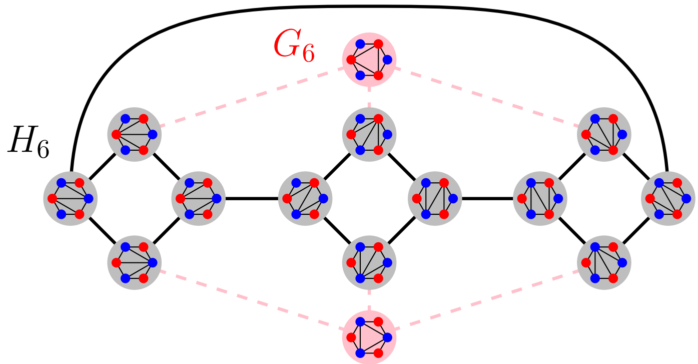


Figure 2: *The flip graph H_6 of colorful triangulations on 6 points, obtained as a subgraph of the 3-dimensional associahedron G_6 by removing two of its vertices, namely the ones with monochromatic triangles (marked red).*

Question 1: Does H_n admit Hamilton paths or cycles for odd $n \geqslant 7$? If so, do these constructions translate into efficient algorithms?

Question 2: What about generalizing the coloring pattern of the n-gon from red, blue, red, blue, etc. to any other coloring sequence? This gives a huge family of interesting subgraphs of the associahedron. What are their properties? What happens with more than two colors?

By considering the geometric duals of the triangulations, flips in triangulations correspond to rotations in the corresponding binary trees. Interestingly, in the colored setting, we can consider a reduced graph in which the flips correspond to rotations in the corresponding ternary trees, and this is where the result [1] comes into play.

References

[1] HUEMER, C., HURTADO, F., AND PFEIFLE, J. The rotation graph of k-ary trees is Hamiltonian. *Inform. Process. Lett. 109*, 2 (2008), 124–129.

[2] HURTADO, F., AND NOY, M. Graph of triangulations of a convex polygon and tree of triangulations. *Comput. Geom. 13*, 3 (1999), 179–188.

[3] SAGAN, B. E. Proper partitions of a polygon and k-Catalan numbers. *Ars Combin. 88* (2008), 109–124.

3: Chromatic number of flip graphs

(suggested by László Kozma)

The associahedron-graph A_n is the flip graph of triangulations of a convex n-gon (or equivalently the rotation graph of binary search trees on $n-2$ nodes). Its chromatic number (for large n) is known to be $\chi(A_n) \geqslant 4$ and $\chi(A_n) \in O(\log n)$. See [1]. It is conjectured that $\chi(A_n)$ is unbounded. As this seems hard to prove for now, how about showing it for more general structures?

> **Question:** How large can the chromatic number be for the flip graph of triangulations of a point set P?
>
> How large can the chromatic number be for the flip graph of triangulations of a simple polygon P?
>
> How large can the chromatic number be for the rotation graph of elimination trees of a graph G? (binary search trees correspond to $G = P_n$)

For all three questions, natural structured families may also be interesting (e.g. triangulations of a grid or elimination trees of a tree), and the question can also be asked for broader generalizations of the associahedron.

References

[1] BERRY, L. A., REED, B., SCOTT, A., AND WOOD, D. R. A logarithmic bound for the chromatic number of the associahedron. *arXiv preprint arXiv:1811.08972* (2018).

4: Relaxing cycle-plus-triangles

(suggested by László Kozma)

A *cycle-plus-triangles* graph has $3n$ vertices and is the disjoint union of a Hamiltonian cycle of length $3n$ with n vertex-disjoint triangles. Du, Hsu, and Hwang conjectured in 1986 that such a graph must have an independent set of size n, i.e. one vertex from each triangle (which is clearly the largest possible). Later Erdős conjectured more strongly that such graphs are 3-colorable.

Several proofs of the stronger claim were obtained using algebraic, topological, or combinatorial techniques (Fleischner and Stiebitz; Alon and Tarsi; Sachs; Aharoni et al.) None of these proofs yield an efficient algorithm; it is in fact open whether an independent set of size n can be found in polynomial time. The problem is also *not* known to be complete for any of the popular search–complexity classes, so perhaps there is such an algorithm. As the exact problem seems hard for now, can we solve some relaxation or approximation of it?

> **Question:** What is the largest α for which we can efficiently find an αn-size independent set of a cycle-plus-triangles graph? As the graph is 4-regular, a simple algorithm yields $\alpha \geqslant \frac{3}{5}$, and $\alpha = 1$ is the famous open question. Can we get α closer to 1?

Cycle-plus-triangles graphs can be recognized in polynomial time [1]. This paper also has references to the older relevant results.

References

[1] BÉRCZI, K., AND KOBAYASHI, Y. An algorithm for identifying cycle-plus-triangles graphs. *Discrete Applied Mathematics 226* (2017), 10–16.

5: TSP with advice

(suggested by László Kozma)

Consider a traveling salesperson (TSP) instance on n cities with arbitrary distances known to both Alice and Bob. Alice has unlimited computational power (can thus solve the TSP problem) and would like to send an "advice" to Bob, so that he can also construct the solution. (Bob can only do polynomial-time computation.) How many bits of advice are needed?

A trivial solution is to send the entire solution (a permutation) in $n \log_2 n + O(n)$ bits.

A folklore improvement is the following: Alice solves the instance, obtaining the sequence of cities $c_{\pi_1}, \dots, c_{\pi_n}$ for some permutation π. She sends the *set* $\{\pi_2, \pi_4, \pi_6, \dots\}$ (every other city in the solution), and their ordering in the solution (a permutation of $n/2$ indices). This needs $\approx \frac{1}{2} n \log_2 n$ bits. Bob just needs to find where $c_{\pi_1}, c_{\pi_3}, \dots$ go in the solution, which he can do (in poly-time) by solving a bipartite perfect matching problem with appropriate edge-weights.

> **Question:** Is there a strategy for Alice with $0.4999 \cdot n \log_2 n$ or fewer bits of advice? More ambitiously, is there a solution with $o(n \log n)$ bits?

Note that if $P = NP$, then no advice is needed, so we don't expect to show a lower bound anytime soon. On the other hand, $o(n)$ would be very surprising (Bob can "guess" the advice).

Can something stronger be shown if the distances are metric or euclidean?

Note: this setting was discussed on R.J. Lipton's blog here:
https://rjlipton.wordpress.com/2020/11/19/traveling-salesman-problem-meets-complexity-theory/

6: The 2-Opt algorithm for the Travelling Salesperson Problem

(suggested by Hung P. Hoang)

The k-Opt algorithm is a local search algorithm for the Travelling Salesperson Problem (TSP). Starting with an initial tour (i.e., an initial Hamilton cycle), it iteratively replaces at most k edges in the tour with the same number of edges to obtain a better tour. Not only is TSP NP-complete, but TSP/k-Opt is also PLS-complete [3]. In other words, it is hard to find a global optimum, and it is also hard to find a local optimum with respect to the k-Opt neighbourhood. However, the original proof of the PLS-completeness by Krentel [3] requires $k \gg 1000$. Recently, we brought the value of k down to 17 [2]. Further, we also showed that for $k \geqslant 5$, there are infinitely many pairs of a TSP instance and an initial tour, for which the k-Opt algorithm always takes exponential time, regardless of the pivot rule.

However, there has been no result in the other direction, i.e., for the following question.

> **Question 1:** For $k \leqslant 4$, is there a pivot rule such that k-Opt algorithm always terminates in polynomial time, for any pair of an edge-weighted complete graph G and an initial tour τ?

The question is still interesting (and challenging) when we restrict to $k = 2$ and the Euclidean setting. In this setting, the vertices of G are embedded on a plane, and the weight of each edge is the Euclidean distance between the two incident vertices. Observe that if two edges cross in a tour, then we can always uncross them to obtain a shorter tour. Hence, we can assume that the initial tour to have no crossing. However, the 2-Opt algorithm may still introduce crossings during its execution.

Note that an affirmative answer to the question above does not mean that k-Opt algorithm always take polynomial time. In fact, Englert, Röglin, and Vöcking [1] showed that this is not true, even for $k = 2$ and the Euclidean setting. However, in their counterexamples, it is easy to see a different pivot rule that removes the exponentially long sequences in the execution that they pointed out.

Another question is to show PLS-completeness for Euclidean TSP, since the PLS-completeness proofs so far only work in the metric setting.

> **Question 2:** Does there exist some k, for which Euclidean TSP/k-Opt is PLS-complete?

References

[1] Englert, M., Röglin, H., and Vöcking, B. Worst case and probabilistic analysis of the 2-opt algorithm for the TSP. *Algorithmica 68*, 1 (2014), 190–264.

[2] Heimann, S., Hoang, H. P., and Hougardy, S. The k-opt algorithm for the traveling salesman problem has exponential running time for $k \geqslant 5$. *arXiv:2402.07061 [cs.DS]* (2024).

[3] Krentel, M. W. Structure in locally optimal solutions. In *30th Annual Symposium on Foundations of Computer Science* (1989), IEEE Computer Society, pp. 216–221.

7: The complexity of variants of permanents and determinants

(suggested by Radu Curticapean)

The determinant of $A \in \mathbb{Q}^{n \times n}$ can be viewed as a sum over permutations, namely

$$\det(A) = \sum_{\pi \in S_n} \operatorname{sgn}(\pi) \prod_{i=1}^{n} A_{i,\pi(i)}.$$

The *permanent* of A is obtained by dropping the sign from the above expression, to obtain

$$\operatorname{per}(A) = \sum_{\pi \in S_n} \prod_{i=1}^{n} A_{i,\pi(i)}.$$

While determinants admit polynomial-time algorithms, computing permanents is NP-hard. Bürgisser [1] initiated the complexity-theoretic study of related matrix functionals, e.g. *immanants*, which are obtained by replacing $\operatorname{sgn} : S_n \to \{-1, 1\}$ in the determinant expansion with an irreducible character $\chi_\lambda : S_n \to \mathbb{Z}$ for $\lambda \vdash n$. These "re-weighted" determinants subsume the determinant and permanent, but also matrix functions of intermediate complexity. We [2] recently determined how λ governs the complexity of the associated immanant.

In the variants considered so far, the weights replacing the permutation sign depend only on the cycle format of the permutation. It is natural to ask whether other properties of permutations (e.g., pattern avoidance) can also be exploited algorithmically when computing determinant-like sums.

Question 1: Are there nontrivial permutation classes $\mathcal{C} \subseteq S_n$ such that

$$\det_{\mathcal{C}}(A) := \sum_{\pi \in \mathcal{C}} \operatorname{sgn}(\pi) \prod_{i=1}^{n} A_{i,\pi(i)},$$

and the similarly defined $\operatorname{per}_{\mathcal{C}}(A)$ can be computed efficiently? Can we even classify these classes?

These questions can also be asked for general immanants.

References

[1] BÜRGISSER, P. *Completeness and Reduction in Algebraic Complexity Theory*, vol. 7 of *Algorithms and computation in mathematics*. Springer, 2000.

[2] CURTICAPEAN, R. A full complexity dichotomy for immanant families. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021* (2021), S. Khuller and V. V. Williams, Eds., ACM, pp. 1770–1783.

8: Rotation distance of k-cut elimination trees

(suggested by Benjamin Aram Berendsohn)

An *elimination tree* T on a connected graph G is a rooted tree that is constructed as follows. Pick a node $r \in V(G)$ and make it the root of T. Then, recursively construct elimination trees on all components of $G - r$ and attach them as children to r.

Note that if G is a path, then T is a binary tree. We can define a rotation in an elimination tree analogously to a binary tree rotation: Swap some node v and its parent p and possibly move some children from v to p to make the result a valid elimination tree (there is always precisely one way to do this).

Figure 3: *A tree* G *and two elimination trees on* G*, differing by the rotation at* c, e*. The node* d *switches parents from* c *to* e*, because it is "between"* c *and* e *in* G*.*

There are some known results on the maximum rotation distance between two elimination trees, depending on the graph G. Here, we want to focus on the case when G is a tree, and restrict ourselves to a certain subset of elimination trees.

We denote a subtree rooted at v (i.e., the subtree containing v and all its descendants) by T_v. By construction, the vertex set $V(T_v)$ induces a connected subgraph of G. The *boundary* of T_v, denoted by $\partial(T_v)$, is the set of vertices in $V(G) \setminus V(T_v)$ that are adjacent to some vertex in $V(T_v)$. An elimination tree T is called k-*cut* if for each vertex v, we have $|\partial(T_v)| \leqslant k$.

The maximum rotation distance between two k-cut trees on a tree G is known to be at most $(2k-1)n$ [1]. The proof is relatively simple. For $k = 1$, just observe that 1-cut trees are *rootings* of G; repeatedly rotating at the root transforms every rooting into every other rooting with n rotations. For $k > 1$, there is a procedure of converting a k-cut tree into a $(k-1)$-cut tree with n rotations. Thus, each k-cut tree can be transformed into a 1-cut tree with $(k-1)n$ rotations.

> **Question 1:** 2-cut elimination trees are particularly interesting because they "locally" behave like binary trees. (Observe that each binary tree, i.e. elimination tree on a path, is 2-cut.) The (tight) rotation distance bound for binary trees is $2n - 6$. Can the $5n$ bound for 2-cut trees be improved?

> **Question 2:** The k-cut bound degenerates to $\Theta(n^2)$ for $k \approx n$, i.e., the general case. However, the rotation distance between elimination trees on trees is known to be only $\mathcal{O}(n \log n)$ [2]. Can the k-cut bound be tightened to $\mathcal{O}(n \log k)$?

Question 3: What about k-cut trees on general graphs? A graph admits a k-cut tree if and only if it has treewidth at most k. The best known upper bound for treewidth-t graphs is $\mathcal{O}(t \cdot n \log n)$ [3]. For k-cut elimination trees, can we get $\mathcal{O}(t \cdot k \cdot n)$, analogous to the $\mathcal{O}(k \cdot n)$ bound when G is a tree?

References

[1] BERENDSOHN, B. A., AND KOZMA, L. Splay trees on trees. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* (2022), pp. 1875–1900.

[2] CARDINAL, J., LANGERMAN, S., AND PÉREZ-LANTERO, P. On the diameter of tree associahedra. *The Electronic Journal of Combinatorics* (2018).

[3] CARDINAL, J., POURNIN, L., AND VALENCIA-PABON, M. Diameter estimates for graph associahedra. *Annals of Combinatorics* (Aug. 2022).

9: Computing the diameter of a matroid, and beyond

(suggested by Jean Cardinal)

Consider the following computational problem, which I already discussed with Arturo Merino: Given a matroid M on m elements, in the form of an independence oracle, what is the diameter $\delta(M)$ of its base polytope? The diameter is the largest distance between two bases of M, measured by the number of exchanges. The distance between two bases is always equal to half their symmetric difference.

We can show that this problem is solvable in polynomial time by reducing it to a matroid intersection problem. Indeed, the diameter of a matroid M is the largest size of a subset of its elements that is independent in both M and its dual M^*. Hence computing the diameter boils down to finding such a subset, a special case of finding the largest common independent set of two matroids. Using the best known algorithm for matroid intersection [3] yields a running time of $O(m\delta \log \delta) = O(m^2 \log m)$.

> **Question:** Can we do better in the oracle model? Is there a superlinear (conditional) lower bound? Can we design algorithms in another computation model, for matroids that are given in some specific forms, for instance graphic matroids (a graph), realizable matroids (a matrix), lattice path matroids [1], or, more generally, transversal matroids?

It is known that computing the diameter of the skeleton of a polytope is hard for polytopes defined by inequalities [4], even for half-integral fractional matching polytopes [6].

> **Question (vague):** Are there other classes of polytopes, given in a more or less implicit form, whose diameter can be computed in polynomial time?

Note that the distance between two bases of a polymatroid is hard to compute in the oracle model [5]. It would be interesting to know if there are classes of polymatroids whose diameter can be computed efficiently.

Another family of nice 0/1 polytopes are the bipartite perfect matching polytopes. They are the intersection of two matroid base polytopes. The edges correspond to flips of alternating cycles. It is known that distances are hard to compute, even approximately [2].

> **Question:** What is the complexity of computing the diameter of the bipartite perfect matching polytope?

References

[1] BONIN, J. E., AND DE MIER, A. Lattice path matroids: Structural properties. *Eur. J. Comb. 27*, 5 (2006), 701–738.

[2] Cardinal, J., and Steiner, R. Inapproximability of shortest paths on perfect matching polytopes. In *Integer Programming and Combinatorial Optimization - 24th International Conference, IPCO 2023, Madison, WI, USA, June 21-23, 2023, Proceedings* (2023), A. D. Pia and V. Kaibel, Eds., vol. 13904 of *Lecture Notes in Computer Science*, Springer, pp. 72–86.

[3] Chakrabarty, D., Lee, Y. T., Sidford, A., Singla, S., and Wong, S. C. Faster matroid intersection. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019* (2019), D. Zuckerman, Ed., IEEE Computer Society, pp. 1146–1168.

[4] Frieze, A. M., and Teng, S. On the complexity of computing the diameter of a polytope. *Comput. Complex. 4* (1994), 207–219.

[5] Ito, T., Kakimura, N., Kamiyama, N., Kobayashi, Y., Maezawa, S., Nozaki, Y., and Okamoto, Y. Hardness of finding combinatorial shortest paths on graph associahedra. In *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany* (2023), K. Etessami, U. Feige, and G. Puppis, Eds., vol. 261 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 82:1–82:17.

[6] Sanità, L. The diameter of the fractional matching polytope and its hardness implications. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018* (2018), M. Thorup, Ed., IEEE Computer Society, pp. 910–921.

10: Sorting by parallel block reversals

(suggested by Vit Jelinek)

Suppose that we are given a sequence of n distinct numbers, which we want to sort into ascending order by using the following iterative procedure: in each round, we partition the current sequence arbitrarily into disjoint blocks of entries in consecutive positions, not necessarily of the same length, and then in a single step we reverse the order of entries within each block. For example, the next figure shows how the sequence 2,1,8,6,7,3,9,5,4 can be sorted in 3 rounds.

original sequence:	2	1	8	6	7	3	9	5	4
after 1 round:	8	1	2	3	7	6	9	4	5
after 2 rounds:	3	2	1	8	7	6	5	4	9
after 3 rounds:	1	2	3	4	5	6	7	8	9

Figure 4: *Sorting the sequence 2,1,8,6,7,3,9,5,4 by parallel block reversals. The horizontal lines indicate which blocks of consecutive elements are reversed in a given round.*

The following problem comes from my recent joint paper with Michal Opler and Jakub Pekárek [2], where we look at the relative power of sorting operators derived from hereditary permutation classes:

> **Question:** What is the smallest number $K(n)$ such that any input sequence of length n can be sorted in at most $K(n)$ rounds? We may assume without loss of generality that the input is a permutation, i.e., a sequence containing each number from the set $\{1, 2, \dots, n\}$ exactly once.

The problem is also related to another sorting paradigm, called sorting by length-weighted reversals, studied previously by Bender et al. [1].

Related results:

- Since there are 2^{n-1} possibilities to choose the blocks to reverse in a single round, it follows that for any given r, there are at most $2^{r(n-1)}$ permutations of length n that can be sorted in r rounds. Since there are in total $n! = 2^{\Omega(n \log n)}$ permutations of length n, it follows that $K(n) = \Omega(\log n)$. This is the best known lower bound on $K(n)$.
- There is a (not too difficult) strategy which can sort any input sequence in $O(\log^2 n)$ rounds. This is the best known upper bound on $K(n)$.
- Apart from sorting by parallel block reversals, described above, we may also consider a modification, called sorting by parallel block transpositions, defined as follows: in every round, we partition the given sequence into an even number of blocks (which may now be empty) numbered left to right as $B_1, B_2, \dots, B_{2k}$, and then for each $i \leqslant k$, we swap the blocks B_{2i-1} and B_{2i}, without changing the order of elements within the blocks. Again, the goal is to determine the smallest number of rounds needed to sort any input of length n. The best known upper and lower bounds are the same as in the original problem.

References

[1] BENDER, M. A., GE, D., HE, S., HU, H., PINTER, R. Y., SKIENA, S., AND SWIDAN, F. Improved bounds on sorting by length-weighted reversals. *Journal of Computer and System Sciences 74*, 5 (2008), 744–774.

[2] JELÍNEK, V., OPLER, M., AND PEKÁREK, J. The hierarchy of hereditary sorting operators. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* (2024), pp. 1447–1464.

11: Chains inside skew shapes

(suggested by Vit Jelinek)

A *polyomino* is a finite collection of unit squares (called *cells*) in the plane whose vertices have integer coordinates. A *filling* of a polyomino is a mapping assigning a value 0 or 1 to each cell of the polyomino. A *transversal filling* (or just a *transversal*) is a filling with exactly one 1-cell in each row and column.

With this terminology, permutations of size n correspond bijectively to transversals of the $n \times n$ square polyomino. Such a representation of a permutation is commonly known as a permutation diagram.

I will only be interested in a special type of polyominoes known as skew shapes. A *skew shape* is a polyomino whose boundary is a union of two internally disjoint lattice paths consisting of up-steps and right-steps; see Figure 5. Clearly, a square is also a skew shape, so permutation diagrams are special cases of transversals of skew shapes.

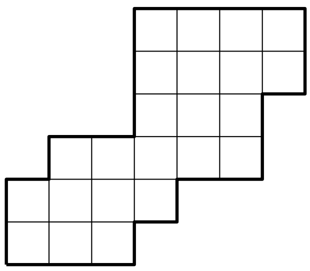


Figure 5: *A skew shape*

I am interested in two kinds of patterns in a filling of a polyomino: increasing chains and decreasing chains. Given a polyomino P and a filling $f \colon P \to \{0, 1\}$ an *increasing* k-*chain* is a sequence of k distinct 1-cells $c_1, \ldots, c_k$ of f with the following properties:

- For each $i = 1, \ldots, k-1$, the column containing the cell c_i is to the left of the column containing c_{i+1} and the row of c_i is below the row of c_{i+1} (but note that the two rows and colmns are not required to be adjacent), and
- the *bounding box* of the cells $c_1, \ldots, c_k$ (i.e., the rectangle whose bottom-left corner is c_1 and top-right corner is c_k) is completely inside the polyomino P.

A *decreasing* k-*chain* is defined analogously. Note that when we consider decreasing chains in skew shapes, the bounding-box condition is always satisfied so the second part of the definition becomes redundant.

If a transversal f does not contain any increasing k-chain, we say it *avoids* increasing k-chains (and similarly for decreasing chains). Figure 6 shows the three possible transversals of a small skew shape. The leftmost transversal avoids decreasing 2-chains (while containing two increasing 2-chains), the other two transversals avoid increasing 2-chains (while each of them contains a decreasing 2-chain).

Over a decade ago, based on computer experiments, I have made the following conjecture:

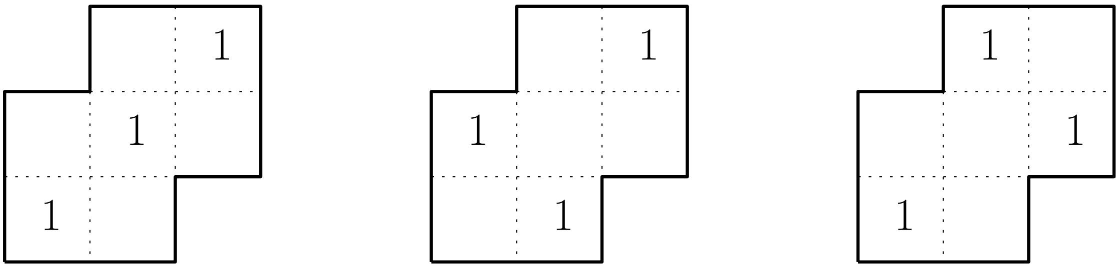


Figure 6: *Three possible transversals of a given skew shape. The 0-cells are depicted as empty boxes.*

Conjecture 1. *For any* k *and for any skew shape* S*, the number of transversals of* S *that avoid decreasing* k*-chains is smaller or equal to the number of transversals of* S *that avoid increasing* k*-chains.*

If true, the conjecture would have implications for the enumeration of pattern-avoiding permutations, by providing new criteria to determine when a permutation pattern is more restrictive than another. The connection with permutation patterns is my motivation for focusing on transversal fillings, but it is entirely possible that the conjecture also holds for other, more general classes of fillings.

The conjecture is known to hold for $k = 2$: in fact, it is not hard to show that any skew shape that has at least one transversal has exactly one transversal avoiding a decreasing 2-chain, while it has at least one (and possibly more) transversal avoiding an increasing 2-chain.

The conjecture is known to hold with equality when instead of general skew shapes we only consider Ferrers shapes, which are skew shapes whose rows are left-justified and columns top-justified. This is a result of Backelin, West and Xin [1], which has since its discovery inspired various generalizations to more general fillings [3] or more general shapes [4]. The equality can also be generalized to a subclass of skew shapes slightly more general than Ferrers shapes, as we've proved jointly with Mark Karpilovskij [2].

However, for any fixed $k \geqslant 3$ and a general skew shape S, Conjecture 1 remains open.

References

[1] BACKELIN, J., WEST, J., AND XIN, G. Wilf-equivalence for singleton classes. *Advances in Applied Mathematics 38* (2007), 133–148.

[2] JELÍNEK, V., AND KARPILOVSKIJ, M. Fillings of skew shapes avoiding diagonal patterns. *Discrete Mathematics & Theoretical Computer Science vol. 22 no. 2, Permutation Patterns 2019* (June 2021).

[3] KRATTENTHALER, C. Growth diagrams, and increasing and decreasing chains in fillings of Ferrers shapes. *Advances in Applied Mathematics 37* (2006), 404–431.

[4] RUBEY, M. Increasing and decreasing sequences in fillings of moon polyominoes. *Advances in Applied Mathematics 47* (2011), 57–87.

12: Coloring ordered graphs with an excluded ordered pattern

(suggested by Bartosz Walczak)

An *ordered graph* is a graph equipped with a linear order on the vertices. An *ordered subgraph* of an ordered graph is a subgraph in which the additional linear order on the vertices is induced from the initial linear order on the vertices of the original graph. Axenovich, Rollin, and Ueckerdt [1] studied the following general problem: for which ordered graphs H do the ordered graphs excluding H as an ordered subgraph have bounded chromatic number? (We consider the ordinary chromatic number, which disregards the additional linear order.) In particular, they provided a positive or negative answer for all but 6 graphs with up to 3 edges (up to reversal of the order). Nowadays, we also know the answer for 5 of the 6 remaining cases, leaving just one:

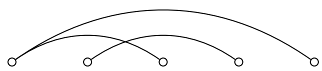


Figure 7: *The ordered graph* H_0*.*

> **Question:** Is the chromatic number of ordered graphs excluding H_0 as an ordered subgraph bounded by a constant?

Interestingly, the maximum average degree of an ordered graph on n vertices excluding H_0 as an ordered subgraph is just barely super-constant, namely, it is of the order $\Theta(\alpha(n))$, where α denotes the inverse Ackermann function. This is an easy consequence of a known bound on the length of so-called Davenport-Schinzel sequences excluding the pattern $ababa$. Therefore, if the chromatic number was unbounded, a construction driving the chromatic number arbitrarily high would need to produce graphs of Ackermann size, whereas classical constructions of triangle-free graphs with large chromatic number lead to primitive recursive bounds on the number of vertices.

References

[1] Axenovich, M., Rollin, J., and Ueckerdt, T. Chromatic number of ordered graphs with forbidden ordered subgraphs. *Combinatorica 38* (2018), 1021–1043.

13: Patterns in rectangulations

(suggested by Namrata)

A (generic) rectangulation [5] is a partition of a rectangle into finitely many interior disjoint rectangles, such that no four rectangles meet at a point; see Figure 8. We refer to every rectangle corner as a *vertex* and to every maximal line segment between two vertices as a *wall*, provided that it does not lie on the boundary of the rectangulation.

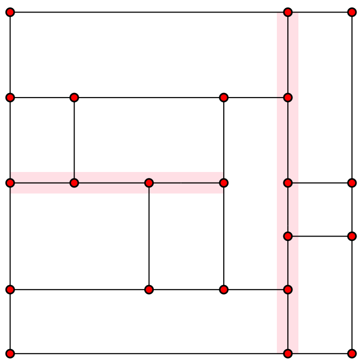
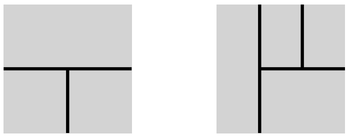


Figure 8: *A rectangulation with the vertices marked, and with one horizontal wall and one vertical wall being highlighted.*

A *pattern* in a rectangulation is a configuration of walls with prescribed directions and incidences. We say that a rectangulation R *contains the pattern* P, if R contains a subset of walls with the directions and incidences specified by P. Otherwise we say that R *avoids* P. Pattern avoidance in rectangulations was first studied in [4]. More specifically, this paper considered eight simple patterns, and counted experimentally the number of rectangulations that avoid any subset of those eight patterns, yielding many matches with the Online Encyclopedia of Integer Sequences (OEIS).

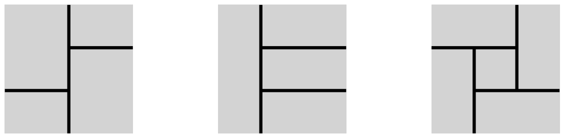
Interestingly, the authors missed several simpler patterns consisting of two or three walls. Experimentally, the number of rectangulations that avoid subsets of these new and old patterns produces some very well-known sequences, where the bijection remains elusive. We present here two of such subsets of patterns.

Question 1: The number of rectangulations that avoid the two patterns in Figure 9 simultaneously, produces the famous Catalan numbers (A000108). Is there an explanation for this phenomenon? Can we give a formal bijection?



(a) (m)

Figure 9: *Rectangulation patterns T and rotated F.*



(c) (j) (o)

Figure 10: *Rectangulation patterns left-right brick, rotated Π, and windmill.*

Question 2: The number of rectangulations that avoid the three patterns in Figure 10 simultaneously produces the Baxter numbers (A001181). Can we give a formal bijection?

There is already a known bijection between Baxter numbers and diagonal rectangulations [6]. A diagonal rectangulation is one in which every rectangle intersects the main diagonal that goes from the top-left to the bottom-right corner of the rectangulation. They are characterized by avoiding the wall patterns  and  [3]. Therefore, it is quite intriguing that we get diagonal rectangulations whether we avoid the above two patterns or the three patterns given in Figure 10.

In preparatory work, the author conducted systematic experiments with all small rectangulation patterns (many of which missed by [4]) that reveal many more such unexplained matches with the OEIS.

Question 3: Is there a general bijection between rectangulations avoiding a particular rectangulation pattern P and permutations avoiding a corresponding (mesh) permutation pattern $\mu(P)$?

Some preliminary results about the second problem have been obtained in [1] and [2].

References

[1] ASINOWSKI, A., AND BANDERIER, C. From geometry to generating functions: rectangulations and permutations. Unpublished manuscript.

[2] ASINOWSKI, A., CARDINAL, J., FELSNER, S., AND FUSY, É. Combinatorics of rectangulations: Old and new bijections. Unpublished manuscript.

[3] CARDINAL, J., SACRISTÁN, V., AND SILVEIRA, R. I. A note on flips in diagonal rectangulations. *Discrete Math. Theor. Comput. Sci. 20*, 2 (2018), Paper No. 14, 22.

[4] MERINO, A., AND MÜTZE, T. Combinatorial generation via permutation languages. III. Rectangulations. *Discrete Comput. Geom. 70*, 1 (2023), 51–122.

[5] READING, N. Generic rectangulations. *European J. Combin. 33*, 4 (2012), 610–623.

[6] YAO, B., CHEN, H., CHENG, C.-K., AND GRAHAM, R. L. Floorplan representations: Complexity and connections. *ACM Trans. Design Autom. Electr. Syst. 8*, 1 (2003), 55–80.

14: Wiggly pseudotriangulations and wiggly permutations

(suggested by Vincent Pilaud)

This problem arose from recent discussions with Asilata Bapat [1]. It was originally motivated by her joint work with Anand Deopurkar and Anthony Licata on categorical representation theory and stability conditions, but it can be stated in purely combinatorial terms.

Fix an integer $n \geqslant 1$. We consider the following two combinatorial families.

Definition 1. *A* wiggly arc *is a quadruple* (i, j, A, B) *where* $0 \leqslant i < j \leqslant n+1$ *and the sets* A *and* B *form a partition of* $\{i+1, \dots, j-1\}$*. We represent it by an* x*-monotone curve wiggling around the horizontal axis, starting at point* i*, ending at point* j*, and passing above the points of* A *and below the points of* B*. The wiggly arcs* $(0, n+1, [n], \varnothing)$ *and* $(0, n+1, \varnothing, [n])$ *are called* irrelevant*, all others are called* relevant*. Two wiggly arcs* (i, j, A, B) *and* (i', j', A', B') *are*

- crossing *if* $(A \cap B') \cup (\{i, j\} \cap B') \cup (A \cap \{i', j'\}) \neq \varnothing \neq (A' \cap B) \cup (\{i', j'\} \cap B) \cup (A' \cap \{i, j\})$*, that is, if the corresponding curves cross,*
- pointed *if* $i \neq j'$ *and* $i' \neq j$*.*

The wiggly complex WC_n *is the simplicial complex of pairwise pointed and non-crossing subsets of wiggly arcs. Note that the irrelevant wiggly arcs are pointed and non-crossing with any other wiggly arc, so that* WC_n *is the join of a segment with its subcomplex* RWC_n *induced by relevant wiggly arcs. It turns out that* RWC_n *is a pure* $(2n-1)$*-dimensional pseudomanifold [1]. A* wiggly pseudotriangulation *is a facet of* WC_n*. The* wiggly flip graph WFG_n *is the adjacency graph of the facets of* WC_n*. Note that, by construction,* WFG_n *is regular of degree* $2n-1$*. See Figure 11 for an illustration when* $n = 2$*.*

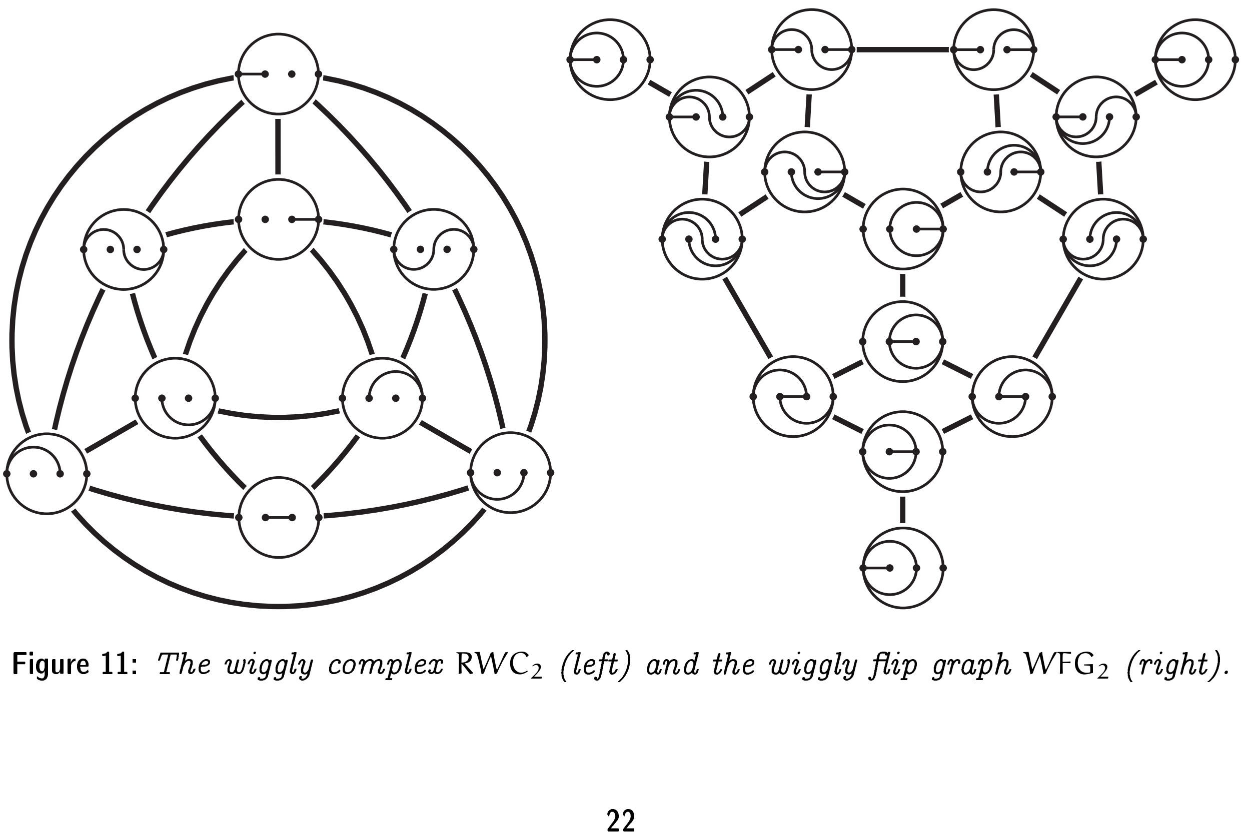


Figure 11: *The wiggly complex* RWC_2 *(left) and the wiggly flip graph* WFG_2 *(right).*

Remark 2. RWC_1 *contains two isolated points.* RWC_2 *is a simplicial 3-dimensional associahedron (this coincidence fails for* $n > 2$*). Here are the first few* f*-vectors:*

$$
\begin{aligned}
f(\mathrm{RWC}_1) &= (1, 2),\\
f(\mathrm{RWC}_2) &= (1, 9, 21, 14),\\
f(\mathrm{RWC}_3) &= (1, 24, 154, 396, 440, 176),\\
f(\mathrm{RWC}_4) &= (1, 55, 729, 4002, 10930, 15684, 11312, 3232),\\
f(\mathrm{RWC}_5) &= (1, 118, 2868, 28110, 140782, 400374, 673274, 662668, 352728, 78384),
\end{aligned}
$$

and the first few h*-vectors:*

$$
\begin{aligned}
h(\mathrm{RWC}_1) &= (1, 1),\\
h(\mathrm{RWC}_2) &= (1, 6, 6, 1),\\
h(\mathrm{RWC}_3) &= (1, 19, 68, 68, 19, 1),\\
h(\mathrm{RWC}_4) &= (1, 48, 420, 1147, 1147, 420, 48, 1),\\
h(\mathrm{RWC}_5) &= (1, 109, 1960, 11254, 25868, 25868, 11254, 1960, 109, 1).
\end{aligned}
$$

Definition 3. *A* wiggly permutation *is a permutation of* $[2n]$ *which avoids the patterns:*

- $(2j-1) \cdots s \cdots (2j)$ *for* $j \in [n]$ *and* $s < 2j$,
- $(2j) \cdots b \cdots (2j-1)$ *for* $j \in [n]$ *and* $b > 2j$.

The wiggly lattice WL_n *is the sublattice of the weak order on permutations of* $[2n]$ *induced by the set of wiggly permutations [1]. Each wiggly permutation covers (resp. is covered by) as many wiggly permutations as its number of descents (resp. ascents), so that the cover graph of* WL_n *is regular of degree* $2n-1$*. See Figure 12 for an illustration when* $n = 2$.

Remark 4. *The number of wiggly permutations are given by*

n	1	2	3	4	5	6	...
$\lvert\mathrm{WL}_n\rvert$	2	14	176	3232	78384	2366248	...

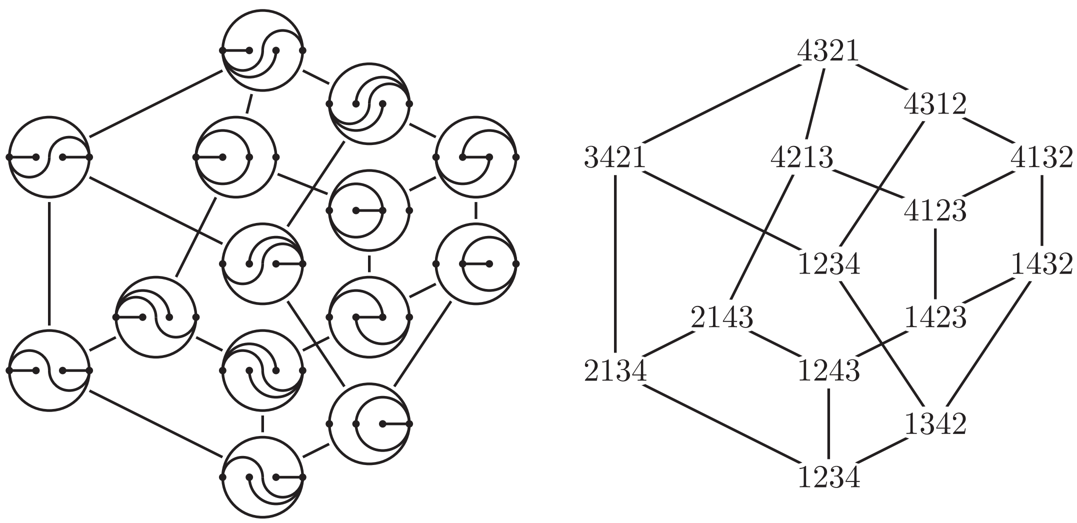


Figure 12: *The wiggly lattice* WL_2 *on wiggly pseudotriangulations (left) and on wiggly permutations (right).*

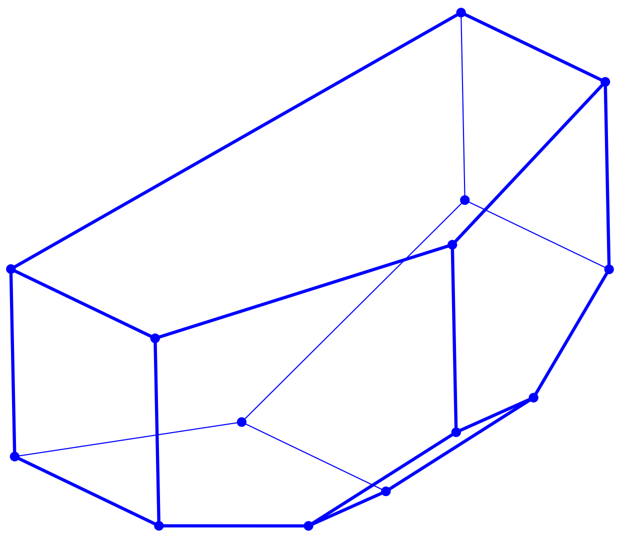


Figure 13: *The wigglyhedron W_2.*

Proposition 5 ([1]). *The wiggly flip graph* WFG_n *is isomorphic to the cover graph of the wiggly lattice* WL_n*.*

Theorem 6 ([1]). *The wiggly complex* RWC_n *is isomorphic to the boundary complex of a simplicial* $(2n-1)$*-dimensional polytope, called* wigglyhedron W_n*. In fact, the Hasse diagram of the wiggly lattice* WL_n *is isomorphic to the graph of the wigglyhedron* W_n *oriented in a linear direction. See Figure 13 for an illustration when* $n = 2$*.*

Conjecture 7. *The wiggly flip graph is Hamiltonian. (Note that unfortunately, wiggly permutations do not form a zigzag language.)*

We now extend Theorem 6 and Conjecture 7 to wiggly pseudotriangulations of arbitrary point sets in the plane (neither necessarily aligned, nor necessarily in general position).

Definition 8. *Fix a point set* P *of the plane, and an arbitrary total order* $<$ *on* P*. A* wiggly arc *is a quadruple* (p, q, R, S) *where* $p < q \in P$ *and the sets* R *and* S *form a partition of the points of* P *located in the open segment joining* p *to* q*. Two wiggly arcs* (p, q, R, S) *and* (p', q', R', S') *are* crossing *if*

- *either the segments* $[p, q]$ *and* $[p', q']$ *cross,*
- *or* $(R \cap S') \cup (\{p, q\} \cap S') \cup (R \cap \{p', q'\}) \neq \varnothing \neq (R' \cap S) \cup (\{p', q'\} \cap S) \cup (R' \cap \{p, q\})$,

A set X *of wiggly arcs is* pointed *if for any* $p \in P$*, the wiggly arcs of* X *with an endpoint at* p *generate a pointed cone. The* wiggly complex WC_P *is the simplicial complex of pairwise pointed and non-crossing subsets of wiggly arcs. Note that the boundary wiggly arcs are irrelevant, which allows us to consider a reduced wiggly complex* RWC_P *induced by relevant wiggly arcs. A* wiggly pseudotriangulation *of* P *is a facet of* WC_P*. The* wiggly flip graph WFG_P *is the adjacency graph of the facets of* WC_P*. Note that, by construction,* WFG_P *is regular. See Figure 14 for an illustration.*

Conjecture 9 ([1]). *For any point set* P *in the plane, the wiggly complex* RWC_P *is the boundary complex of a simplicial polytope.*

Note that in Conjecture 9, the case of aligned points is given by the wigglyhedron of Theorem 6, while the case of points in general position is given by the pseudotriangulation polytope of [5] (see also [6] for a nice survey on pseudotriangulations).

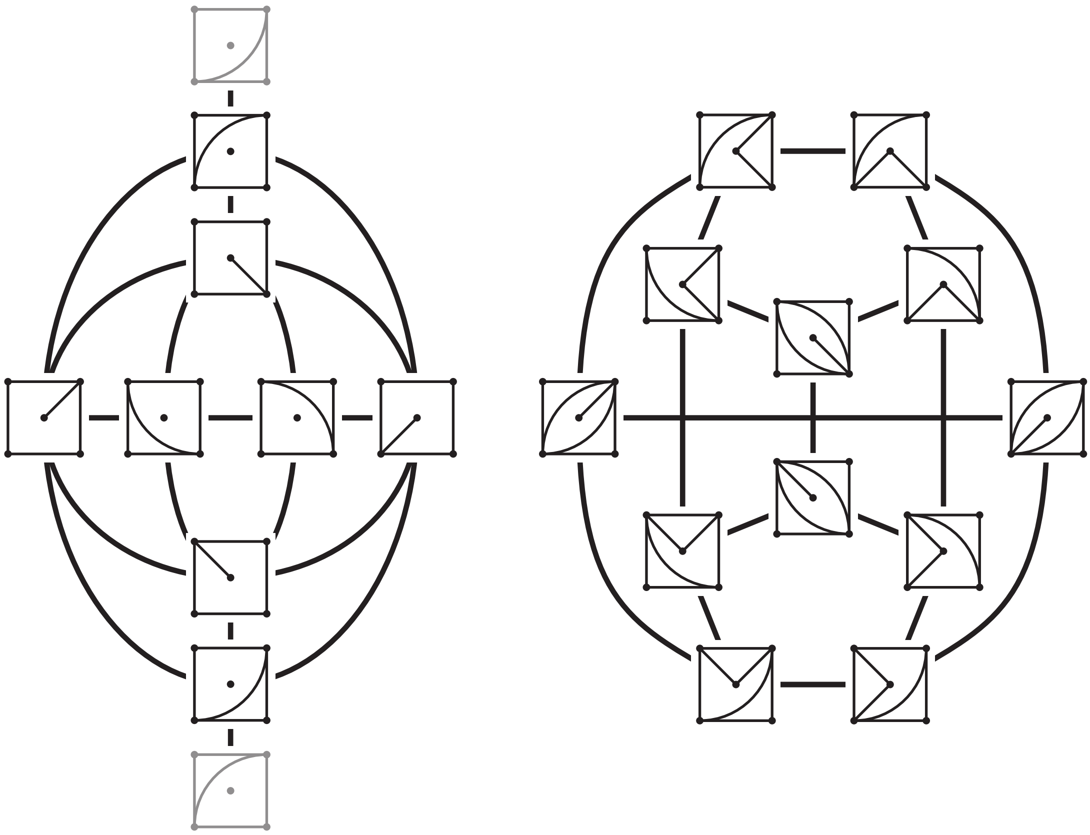


Figure 14: *The wiggly complex* $\mathrm{RWC_P}$ *(left) and the wiggly flip graph* $\mathrm{WFG_P}$ *(right) of a point set* P.

Conjecture 10. *For any point set* P *in the plane, the wiggly flip graph* $\mathrm{WFG_P}$ *is Hamiltonian.*

A few additional questions to arouse your curiosity:

1. Can the construction of Theorem 6 be adapted to provide a more combinatorial construction of the polytope of pseudotriangulations [5], that would only depend on the order type (aka oriented matroid [2]) of the point configuration?

2. What is the dual interpretation of wiggly pseudotriangulations as pseudoline arrangements? (see [4] for context) Can this interpretation be extended to other finite Coxeter groups? (see [3] for context).

3. What is the multi wiggly complex?

References

[1] BAPAT, A., AND PILAUD, V. Personal communication. 2024.

[2] BJÖRNER, A., LAS VERGNAS, M., STURMFELS, B., WHITE, N., AND ZIEGLER, G. M. *Oriented matroids*, second ed., vol. 46 of *Encyclopedia of Mathematics and its Applications.* Cambridge University Press, Cambridge, 1999.

[3] Ceballos, C., Labbé, J.-P., and Stump, C. Subword complexes, cluster complexes, and generalized multi-associahedra. *J. Algebraic Combin. 39*, 1 (2014), 17–51.

[4] Pilaud, V., and Pocchiola, M. Multitriangulations, pseudotriangulations and primitive sorting networks. *Discrete Comput. Geom. 48*, 1 (2012), 142–191.

[5] Rote, G., Santos, F., and Streinu, I. Expansive motions and the polytope of pointed pseudo-triangulations. In *Discrete and computational geometry*, vol. 25 of *Algorithms Combin.* Springer, Berlin, 2003, pp. 699–736.

[6] Rote, G., Santos, F., and Streinu, I. Pseudo-triangulations—a survey. In *Surveys on discrete and computational geometry*, vol. 453 of *Contemp. Math.* Amer. Math. Soc., Providence, RI, 2008, pp. 343–410.

15: Pancake flipping for permutations (Big 3)

(suggested by Joe Sawada and Aaron Williams)

A stack of n different sized pancakes can be modeled by a permutation. A *flip* of j pancakes corresponds to a prefix reversal of length j in the permutation. If the pancakes are burnt on one side, then they can be modeled by signed permutations and a flip also changes the signs of the reversed elements. See [1, 2, 3, 4, 5]. The following problems correspond to finding Hamilton paths (or cycles) in their corresponding underlying graphs; similar statements originally appear in [4].

Question 1: **Big 3**: Construct a flip-Gray code for permutations of order $n > 2$ where only flips of size $n-2$, $n-1$, and n are allowed.

Question 2: **Big 3 (burnt)**: Construct a flip-Gray code for signed permutations (burnt pancakes) of order n where only flips of size $n-2$, $n-1$, and n are allowed.

Question 3: Construct a flip-Gray code for permutations of order n where only flips of size 2, $n-1$, and n are allowed.

Since the corresponding graphs are connected, by showing no such Gray code exists would be contrary to a famous conjecture of Lovász stating that every connected Caley graph has a Hamilton cycle).

References

[1] CAMERON, B., SAWADA, J., THERESE, W., AND WILLIAMS, A. Hamiltonicity of k-sided pancake networks with fixed-spin: Efficient generation, ranking, and optimality. *Algorithmica 85*, 3 (Mar 2023), 717–744.

[2] DWEIGHTER, H. Problem E2569. *American Mathematical Monthly 82* (1975), 1010.

[3] SAWADA, J., AND WILLIAMS, A. Greedy flipping of pancakes and burnt pancakes. *Discrete Appl. Math. 210* (2016), 61–74.

[4] SAWADA, J., AND WILLIAMS, A. Successor rules for flipping pancakes and burnt pancakes. *Theoret. Comput. Sci. 609*, part 1 (2016), 60–75.

[5] ZAKS, S. A new algorithm for generation of permutations. *BIT 24*, 2 (1984), 196–204.

16: Lex smallest universal cycles for shorthand permutations

(suggested by Joe Sawada)

Let $\pi = p_1p_2\cdots p_n$ be a permutation of order n. A *shorthand permutation* for π is $p_1p_2\cdots p_{n-1}$, where the missing redundant symbol is p_n. Let $SP(n)$ denote the set of $n!$ shorthand permutations of order n. For example,

$$SP(4) = \begin{array}{l} 123, 124, 132, 134, 142, 143, 213, 214, 231, 234, 241, 243 \\ 312, 314, 321, 324, 341, 342, 412, 413, 421, 423, 431, 432. \end{array}$$

A *universal cycle* (UC) for a set S of length n strings is a circular string of length $|S|$ where every string in S appears exactly once as a substring. Universal cycles for permutations do not exist for $n \geqslant 3$; however, they do exist for shorthand permutations [1, 2, 3, 5, 7].

Finding the lexicographically smallest element for some combinatorial object is a natural question. Thus, we ask: *What is the lexicographically smallest UC for shorthand permutations?* Let U_n^* denote the lexicographically smallest UC for $SP(n)$. Exhaustive search reveals:

$$U_4^* = 123\ 124\ 132\ 134\ 214\ 324\ 314\ 234.$$

Notice that the universal cycle can be written as the concatenation of "necklace representatives" of each equivalence class of shorthand permutations under rotation. See Figure 15 for a corresponding cycle-joining tree; each cyclic node differs from its parent by a single symbol. Connections between the cycle-joining trees and the resulting traversal/concatenation of the cycles is detailed in recent work [6].

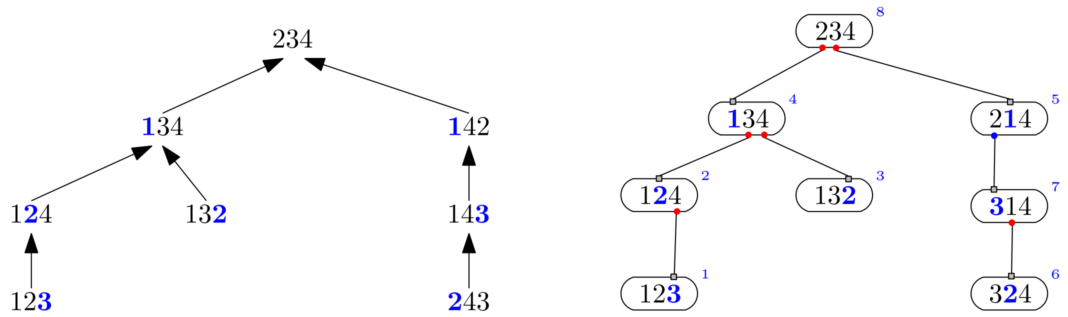


Figure 15: *(left) A cycle-joining tree for necklace cycles of* $SP(4)$ *represented by their smallest rotation. What is the parent rule? (right) A corresponding concatenation tree – visiting the nodes in the labeled RCL (right-current-left) order produces* U_4^*.

For $n = 5$, exhaustive search reveals:

$$U_5^* = \begin{array}{l} 1234\ 1235\ 1243\ 1245\ 1324\ 1325\ 1342\ 1345\ 2135\ 2143\ 2145\ 2314\ 2315\ 2415\ 3215 \\ 3425\ 1435\ 2435\ 1425\ 3145\ 3245\ 3125\ 4135\ 4215\ 4325\ 4315\ 4235\ 4125\ 3415\ 2345. \end{array}$$

A conjectured cycle-joining tree for U_5^* is shown in Figure 16. Each node is written in its lexicographically smallest rotation (though perhaps another representation is helpful). Notice when 1 is missing, we replace the 2. A 5 is never replaced. Is there enough information to articulate a general parent rule?

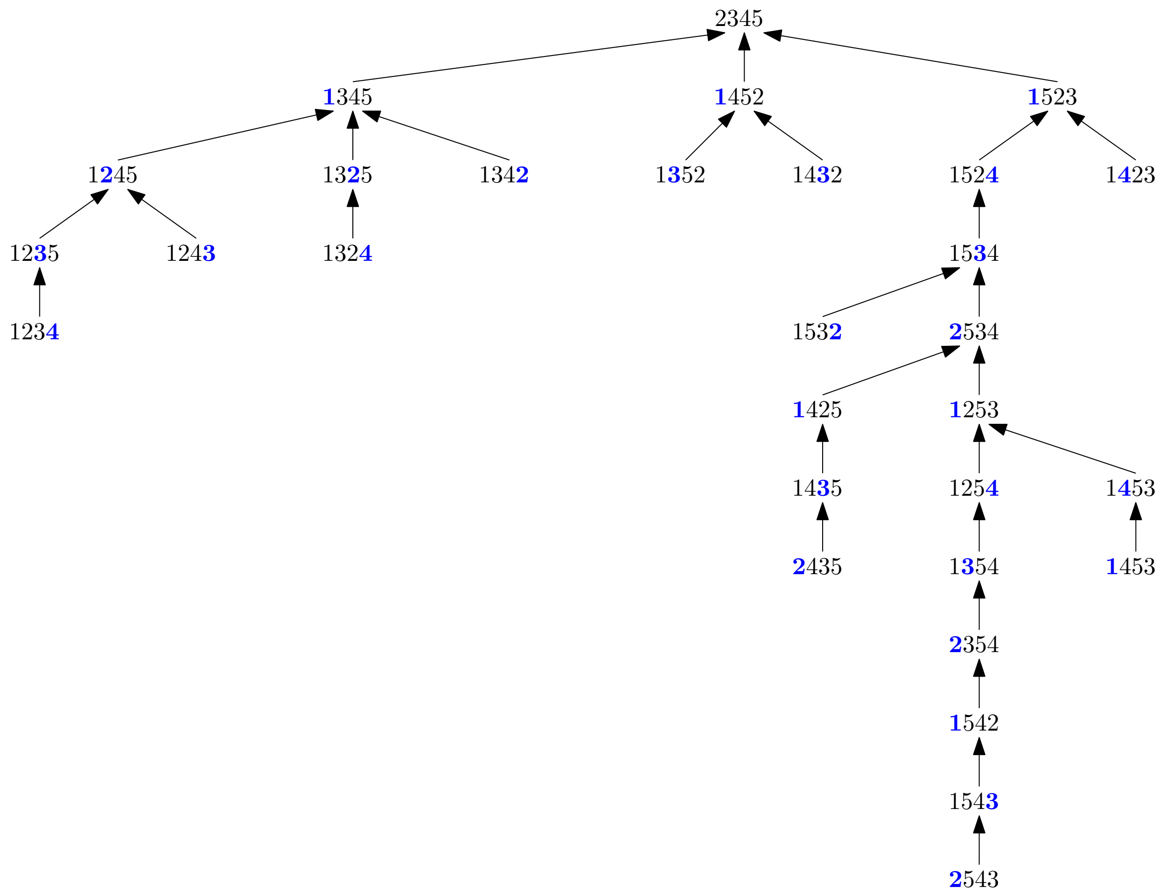


Figure 16: *A cycle-joining tree for U_5^*? What is the parent rule?*

Question 1: Is there a $O(n)$-time successor rule for determining successive symbols in U_n^*? This can be accomplished by finding a simple cycle-joining approach to construct U_n^*; in other words, is there a simple parent rule to construct a cycle-joining tree for necklace representatives of $SP(n)$? By studying U_5^* and U_6^*, there may be enough details to determine such a tree.

Question 2: What is U_6^* and U_7^*? Since U_6^* has length $6! = 720$, a brute force search already seems infeasible. However, it may be critical to answering the previous question. Thus, implementing the greedy Euler cycle algorithm by Moreno and Matamala [4] seems the next natural step.

References

[1] GABRIC, D., SAWADA, J., WILLIAMS, A., AND WONG, D. A successor rule framework for constructing k -ary de Bruijn sequences and universal cycles. *IEEE Transactions on Information Theory 66*, 1 (2020), 679–687.

[2] HOLROYD, A. E., RUSKEY, F., AND WILLIAMS, A. Shorthand universal cycles for permutations. *Algorithmica 64*, 2 (2012), 215–245.

[3] JOHNSON, J. R. Universal cycles for permutations. *Discrete Mathematics 309*, 17 (2009), 5264–5270.

[4] MORENO, E., AND MATAMALA, M. Minimal Eulerian circuit in a labeled digraph. In *LATIN 2006: Theoretical Informatics* (Berlin, Heidelberg, 2006), J. R. Correa, A. Hevia, and M. Kiwi, Eds., Springer Berlin Heidelberg, pp. 737–744.

[5] Ruskey, F., and Williams, A. An explicit universal cycle for the (n-1)-permutations of an n-set. *ACM Trans. Algorithms 6*, 3 (July 2010), 1–12.

[6] Sawada, J., Sears, J., Trautrim, A., and Williams, A. Concatenation trees: A framework for efficient universal cycle and de Bruijn sequence constructions. arxiv preprint arxiv:2308.12405, 2023.

[7] Wong, D. A new universal cycle for permutations. *Graph. Comb. 33*, 6 (Nov. 2017), 1393–1399.

17: Partial universal cycles

(suggested by Joe Sawada)

A *universal partial cycle* (UPC) of order n is a cyclic sequence that contains every k-ary string of length n as a substring – like a de Bruijn sequence, except that a wildcard symbol $\diamond$ is allowed to represent any alphabet symbol. For instance: $001\diamond110\diamond$ is a UPC for $n = 4$ and $k = 2$. The existence of UPCs with wildcard symbols are highly constrained, and only recently a UPC was discovered for $n > 4$ [1]. In particular, they discovered UPCs for $n = 8$ and $k = 2$ via computer search and extended them to larger alphabets. They state a conjecture similar to the following question:

> **Question 1:** Does there exist a (binary) UPC for all $n = 2^j$, $j \geqslant 2$, where wildcards appear at every n-th position? As a starting point, can we construct a UPC for $n = 2^4 = 16$?

Motivated by their discovery, we generated a UPC for $n = 8$ with an interesting property: it corresponds to the concatenation of smaller "co-necklace" cycles of length $2n$. In the following UPC example, every second string between wildcards is the complement of the string that precedes it:

$$0000001 \diamond 1111110 \diamond 0001001 \diamond 1110110 \diamond 0010001 \diamond 1101110 \diamond 0011011 \diamond 1100100 \diamond$$
$$0011001 \diamond 1100110 \diamond 0100001 \diamond 1011110 \diamond 0101001 \diamond 1010110 \diamond 0111001 \diamond 1000110 \diamond.$$

A *co-necklace* is an equivalence class of strings induced by the complementing cycling register (CCR) $f(a_1a_2\cdots a_n) = a_2\cdots a_n\overline{a}_1$. Each co-necklace class has at most $2n$ elements. We can also think of a co-necklace as the cycle induced by the feedback function $g(a_1a_2\cdots a_n) = \overline{a}_1$. For example, 0000000011111111 is a co-necklace cycle for $n = 8$; each substring of length n belongs to the same co-necklace equivalence class. Note that the first 16 symbols in our example UPC correspond to the joining of two co-necklace cycles:

$$0000001\,0\,1111110\,1 \cdot 0000000\,1\,1111111\,0.$$

When $n = 2^j$, there are 2^{n-j-1} co-necklace equivalence classes; when $n = 8$ there are $2^4 = 16$ such classes.

> **Question 2:** Let $n = 2^j$. Define a partition of co-necklace cycles into two equal sized sets $S_1 = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$ and $S_2 = \{\beta_1, \beta_2, \ldots, \beta_m\}$ such that α_i and β_i differ in exactly two positions n bits apart for each $1 \leqslant i \leqslant m$. The pairings in the above UPC example correspond to such a partition.

If this question can be answered, then quite likely it will lead to a solution to Question 1. More details on co-necklaces are available in [2].

References

[1] FILLMORE, D., GOECKNER, B., KIRSCH, R., MARTIN, K., AND MCGINNIS, D. The existence and structure of universal partial cycles. arxiv preprint arxiv:2310.13067, 2023.

[2] GABRIC, D., AND SAWADA, J. A de Bruijn sequence construction by concatenating cycles of the complemented cycling register. In *Combinatorics on Words* (Cham, 2017), S. Brlek, F. Dolce, C. Reutenauer, and É. Vandomme, Eds., Springer International Publishing, pp. 49–58.

18: Generating k-ary unlabeled necklaces

(suggested by Joe Sawada)

A *necklace* is an equivalence class of strings under rotation. A *bracelet* is an equivalence class of strings under rotation and reversal. An *unlabeled necklace* is an equivalence class of strings under rotation and permutation of the alphabet symbols. Constant amortized time (CAT) algorithms are known to list distinct representatives from each equivalence class of k-ary necklaces [2, 3] and k-ary bracelets [4]. However, for unlabeled necklaces, a CAT algorithm is known only for $k = 2$ [1].

Observe that 0012, 0021, 0112, 0221, 0122, 0211 are the lexicographically smallest elements in their respective necklace class; they all belong to the same unlabeled necklace class. There are six unlabeled necklaces for $n = 4$ and $k = 3$; their lexicographically smallest representatives are:

$$0000, 0001, 0011, \underline{0012}, 0101, 0102.$$

For $n = 5$ and $k = 4$ there are 11 unlabeled necklace classes represented by:

$$00000, 00001, 00011, 00012, 00101, 00102, 00112, 00121, 00123, 01012, 01023.$$

Question: Does there exist a CAT algorithm to list representatives of each k-ary unlabeled necklace class of order n? Even the special cases fixing $k = 3$ and $k = 4$ are interesting. If k is not fixed, an $O(n)$-amortized time algorithm may be a nice result.

Note: I have received multiple inquires regarding the existence of such an efficient algorithm over the years from researchers in various disciplines (including last month!)

References

[1] CATTELL, K., RUSKEY, F., SAWADA, J., SERRA, M., AND MIERS, C. Fast algorithms to generate necklaces, unlabeled necklaces, and irreducible polynomials over GF(2). *Journal of Algorithms 37*, 2 (2000), 267–282.

[2] FREDRICKSEN, H., AND KESSLER, I. J. An algorithm for generating necklaces of beads in two colors. *Discrete Math. 61*, 2-3 (1986), 181–188.

[3] FREDRICKSEN, H., AND MAIORANA, J. Necklaces of beads in k colors and k-ary de Bruijn sequences. *Discrete Math. 23*, 3 (1978), 207–210.

[4] SAWADA, J. Generating bracelets in constant amortized time. *SIAM J. Comput. 31*, 1 (2001), 259–268.

19: Bifurcated Ordered Trees (BOTs)

(suggested by Joe Sawada and Aaron Williams)

An *ordered tree* is a rooted tree in which the children of each node are given a total order. For example, a node in an ordered tree with three children has a first child, a second child, and a third (last) child. In contrast, a *cardinal tree* is a rooted tree in which the children of each node occupy specific positions. In particular, a k-*ary tree* has k positions for the children of each node. For example, each child of a node in a 3-ary tree is either a left-child, a middle child, or a right-child.

We consider a new type of tree that is both ordinal and cardinal; while ordered trees have one "type" of child, our trees will have two types of children. We refer to such a tree as a *bifurcated ordered tree* (*BOT*), with the two types of children being *left-children* and *right-children*. To illustrate bifurcated ordered trees, Figure 17 provides all BOTs with $n = 3$ nodes. This type of "ordinal-cardinal" tree seems quite natural, and it is very likely

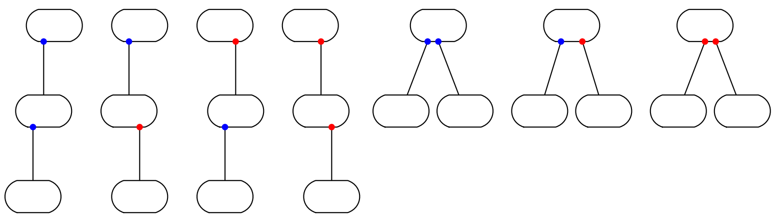


Figure 17: *All seven bifurcated ordered trees (BOTs) with n=3 nodes. Each left-child descends from a blue •, while each right-child descends from a red •.*

to have been used in previous academic investigations. Nevertheless, we have not been able to find an exact match in the literature. In particular, 2-tuplet trees use a different notion of a root, and correspond more closely to ordered forests of BOTs. A (non-trivial) computer program to enumerate all BOTs demonstrates that the total number for n from 1 to 12 are:

$$1, 2, 7, 30, 143, 728, 3876, 21318, 120175, 690690, 4032015, 23841480.$$

When extended for larger n, the sequence corresponds to all 23 entries for sequence A006013 in the Online Encyclopedia of Integer Sequence [1]; however, no obvious relationship to such a tree is noted.

> **Question 1:** Have BOTs been previously studied or applied in previous academic investigations? Is there a natural bijection between BOTs and objects with the same enumeration sequences as detailed in sequence A006013 [1]?

> **Question 2:** Is there a simple and efficient (Gray code) algorithm to generate all BOTs of order n? Can the resulting ordering be efficiently ranked/unranked?

Application: BOTs, and a non-traditional right-current-left traversal, have recently been applied in the most efficient framework to construct de Bruijn sequences and universal cycles [2]. The natural generalization to k-*furcated ordered trees* is required in extending the application.

References

[1] The On-Line Encyclopedia of Integer Sequences, published electronically at https://oeis.org, 2010, sequence A006013.

[2] SAWADA, J., SEARS, J., TRAUTRIM, A., AND WILLIAMS, A. Concatenation trees: A framework for efficient universal cycle and de Bruijn sequence constructions. arxiv preprint arxiv:2308.12405, 2023.

20: Pivot Gray codes for spanning trees

(suggested by Torsten Mütze)

In this problem, we consider the set of spanning trees of a fixed graph G, with the goal of listing them so that any two consecutive spanning trees T and T′ differ in an edge exchange, i.e., $T' = T + e - f$ for two suitable edges e and f of G. It is well-known that such listings exist for any graph G, and the corresponding listings correspond to walks along the skeleton of the corresponding base polytope.

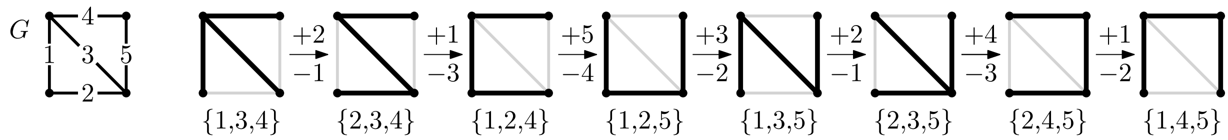


Figure 18: *Edge exchange Gray code for the spanning trees of the diamond graph on the left. Below each spanning tree is the subset of tree edges.*

In a recent paper, Cameron, Grubb and Sawada [1] asked whether any graph G admits a Gray code listing of its spanning trees subject to the more restrictive condition that the exchanged edges e and f are incident to the same vertex. They refer to such a listing as a *pivot* Gray code, and they construct such listings for the case when G is a fan graph; see Figure 19 (a). For fan graphs, an even more restrictive exchange condition, called *face pivot* exchange, was considered in [2], where the exchanged edges e and f are required to be incident to the same vertex and the same face.

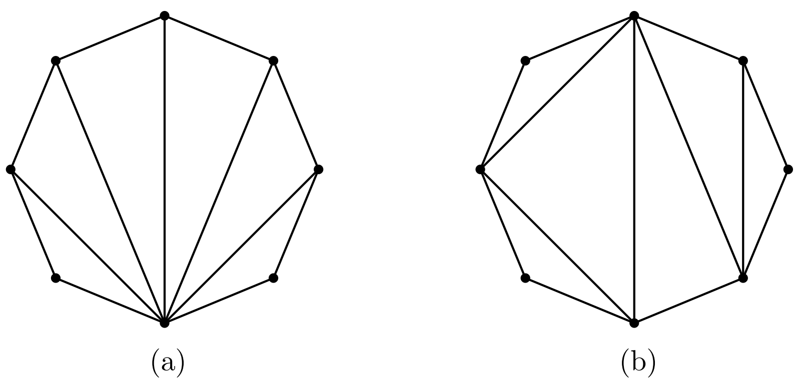


Figure 19: *(a) The fan graph; (b) a general triangulation of an n-gon.*

In preparatory work, we proved that the spanning trees of any triangulation of a convex n-gon (of which the fan graph is a very special case; see Figure 19) admit pivot Gray codes, and there is a very simple and versatile greedy algorithm to generate such a listing.

> **Question:** Can we derive pivot Gray codes for listing the spanning trees of more general classes of (planar) graphs? If so, do these constructions translate into efficient algorithms? What about another closeness condition for plane graphs, namely 'e and f are incident to the same face, but not necessarily to the same vertex' (this is a pivot exchange in the dual spanning tree)?

References

[1] CAMERON, B., GRUBB, A., AND SAWADA, J. A pivot Gray code listing for the spanning trees of the fan graph. In *Computing and Combinatorics - 27th International Conference, COCOON 2021, Tainan, Taiwan, October 24-26, 2021, Proceedings* (2021), C. Chen, W. Hon, L. Hung, and C. Lee, Eds., vol. 13025 of *Lecture Notes in Computer Science*, Springer, pp. 49–60.

[2] MERINO, A. I., MÜTZE, T., AND WILLIAMS, A. All your bases are belong to us: Listing all bases of a matroid by greedy exchanges. In *11th International Conference on Fun with Algorithms, FUN 2022, May 30 to June 3, 2022, Island of Favignana, Sicily, Italy* (2022), P. Fraigniaud and Y. Uno, Eds., vol. 226 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 22:1–22:28.

21: Flips in plane perfect matchings

(suggested by Oswin Aichholzer)

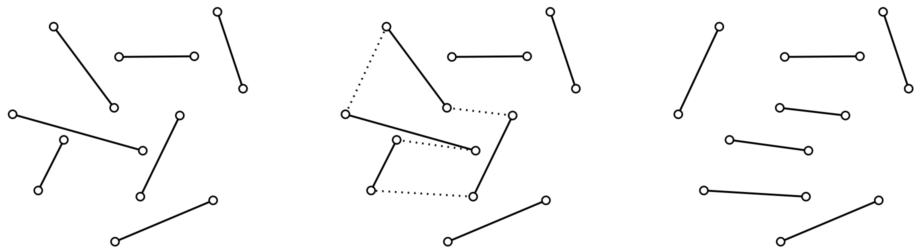


Figure 20: *A* k*-flip between two plane perfect matchings for* $k = 4$.

Let S be a set of $n = 2m$ points in the plane in general position, that is, no three points are on a common line. Two plane perfect matchings on S differ by a k-flip if their symmetric difference is a single non-crossing alternating cycle of length 2k. See for example the nice survey of Bose and Hurtado [1] for this type of flip and flips on graphs in general. Figure 20 shows an example of a 4-flip. Note that k is at least two, as removing just one edge from a plane perfect matching gives no freedom to change anything.

We are interested in the following question: Can any plane perfect matching on S be transformed into any other plane perfect matching on S by a sequence of $\leqslant$ k-flips for some fixed k? Houle et al. [2] showed that the corresponding flip-graph is connected assuming that no upper bound is prescribed on the size of k. This means, that a single flip might change a linear number of edges. However, in the special case when the points are in convex position, the flip graph is connected for 2-flips and has a Hamiltonian cycle when $m \geqslant 4$ is even and no Hamiltonian path for $m > 3$ odd [1].

> **Question 1:** For a given set S with $n = 2m$ points is the flip-graph of all plane perfect matchings on S connected via $\leqslant$ k-flips for some constant or at least sublinear k?

So far it is not even known that we can always make at least one flip of constant size. On the other hand, there is no example known that does not allow for at least one 2-flip.

> **Question 2:** How long is the **shortest** k-flip that can always be made in a plane perfect matching?

As defined above, for a k-flip the symmetric difference of the two matchings is a single non-crossing alternating cycle of length 2k. Now we allow crossings between the different edges of the alternating cycle (but not with other edges), and the two involved matchings are still plane. Is this operation more powerfull than the original flip-operation? Note that for $k = 2$ crossings are not possible.

> **Question 3:** If for a k-flip, $k \geqslant 3$, we allow crossings between the different edges of the alternating cycle (but not with other edges), how does this influence the answers to the previous two questions?

References

[1] Bose, P., and Hurtado, F. Flips in planar graphs. *Computational Geometry 42*, 1 (2009), 60–80.

[2] Houle, M., Hurtado, F., Noy, M., and Rivera-Campo, E. Graphs of triangulations and perfect matchings. *Graphs and Combinatorics 21* (2005), 325–331.

22: Hamiltonicity of Token Graphs

(suggested by Arturo Merino)

Let $G = (V, E)$ be a graph on n vertices with m edges and $\vec{a} = (a_1, \ldots, a_k)$ an integer partition of n. We say that a coloring of G is an $\vec{a}$-coloring if for every $i \in [k]$ we have a_i vertices with color i in G. The *token graph* $\mathcal{T}(G, \vec{a})$ is a graph that has as vertices all $\vec{a}$-colorings of G, and two $\vec{a}$-colorings are connected by an edge if they differ only in the colors of the end vertices of one edge. (In other words, the token graph $\mathcal{T}(G, \vec{a})$ is the flip graph of the $\vec{a}$-colorings, where the operation is to swap two different color tokens along an edge.)

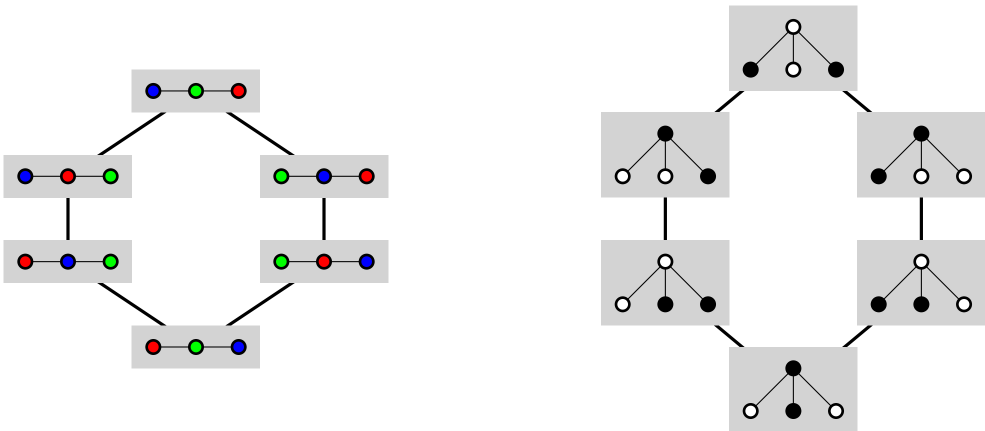


Figure 21: *The token graphs $\mathcal{T}(P_3, (1, 1, 1))$ on the left and $\mathcal{T}(K_{1,3}, (2, 2))$ on the right.*

Token graphs have amassed some popularity recently, because of their many connections to other fields in mathematics and physics. People have also studied many computational aspects of them: connectivity, shortest paths, diameter, etc. Surprisingly, not so much is known about their Hamiltonicity; neither structurally nor computationally.

These Hamiltonicity properties are very relevant for combinatorial generation problems, as many classical and new Gray-coding results can be phrased quite nicely in the language of token graphs. Take for example Tchuente's theorem: It states when we can generate permutations by a given set of transpositions. This can be nicely phrased in terms of token graphs as follows.

Theorem 1 (Tchuente's theorem [5]). *The graph $\mathcal{T}(G, (1)^n)$ is Hamiltonian if and only if G is connected.*

Surprisingly, Tchuente's theorem gives us a nice characterization for Hamiltonicity of token graphs with n colors.

With fewer colors things are also very interesting and expressive. It turns out that we can, for example, restate the Middle Levels theorem in token graph language with only two colors.

Theorem 2 (Middle levels theorem [2]). *For even n, the graph $\mathcal{T}(K_{1,n-1}, (n/2, n/2))$ is Hamiltonian.*

A natural question is to study how the hardness behaves for different numbers of colors.

Question 1: How hard is to decide wether a token graph with (fixed) k colors is Hamiltonian? It is in P for $k = 1$ (trivially) and $k = n$ (by Tchuente's result). On the other hand it is NP-complete for $k = 2$ (as $\mathcal{T}(G, (1, n-1)) \cong G$). What can we say for $k = 3, \ldots, n-1$?

We can also aim for more structural/combinatorial results by studying the Hamiltonicity of token graphs when placing restrictions on G. In particular, characterizations for Hamiltonicity of token graphs are still open for trees, paths, and cycles.

Question 2: Is there a characterization for Hamiltonicity of Token graphs (similar to Tchuente's result for n colors) when the graphs are trees/paths/cycles?

There are some partial results in this direction; see e.g. [1, 3, 4]. Inductive approaches seem promising and is what is done in [1, 5].

References

[1] Gregor, P., Merino, A., and Mütze, T. Star transposition Gray codes for multiset permutations. *arXiv:2108.07465*, 2021.

[2] Mütze, T. Proof of the middle levels conjecture. *Proc. Lond. Math. Soc. 112*, 4 (2016), 677–713.

[3] Ruskey, F. Adjacent interchange generation of combinations. *J. Algorithms 9*, 2 (1988), 162–180.

[4] Stachowiak, G. Hamilton paths in graphs of linear extensions for unions of posets. *SIAM J. Discrete Math. 5*, 2 (1992), 199–206.

[5] Tchuente, M. Generation of permutations by graphical exchanges. *Ars Combin. 14* (1982), 115–122.

23: Exponential-Time Complexity of Token Swapping

(suggested by Yoshio Okamoto)

This is an algorithmic question.

Question: Can the token swapping problem be solved in singly exponential time?

In the *token swapping* problem, we are given a connected undirected graph $G = (V, E)$ and a permutation $\pi\colon V \to V$. We treat the image $\pi(v)$ as a token placed on the vertex v. The goal is to transform π to the identity permutation by a sequence of swap operations. A *swap* is an operation that transforms a permutation π to another permutation $\pi'\colon V \to V$ that satisfies the following:

- there exists an edge $uv \in E$ such that $\pi'(u) = \pi(v), \pi'(v) = \pi(u)$ and $\pi'(w) = \pi(w)$ for all $w \in V - \{u, v\}$.

In the problem, we are asked to find a shortest sequence with this property. It is known (and easy to see) that such a sequence always exists.

The problem was introduced by Yamanaka et al. [5], and the decision version is NP-complete [4, 3], even when G is a tree [1]. Parameterized complexity has also been investigated [2].

As for the exponential-time complexity, we only know a trivial algorithm: enumerate all the permutations, construct the "configuration graph," and search for a shortest path, resulting in a running time of roughly $n!$ (or $n \cdot n!$), where $n = |V|$.

The question above asks about the existence of an algorithm with a running time $2^{O(n)}$.

References

[1] AICHHOLZER, O., DEMAINE, E. D., KORMAN, M., LUBIW, A., LYNCH, J., MASÁROVÁ, Z., RUDOY, M., WILLIAMS, V. V., AND WEIN, N. Hardness of token swapping on trees. In *30th Annual European Symposium on Algorithms, ESA 2022, September 5-9, 2022, Berlin/Potsdam, Germany* (2022), S. Chechik, G. Navarro, E. Rotenberg, and G. Herman, Eds., vol. 244 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 3:1–3:15.

[2] BONNET, É., MILTZOW, T., AND RZAZEWSKI, P. Complexity of token swapping and its variants. *Algorithmica 80*, 9 (2018), 2656–2682.

[3] KAWAHARA, J., SAITOH, T., AND YOSHINAKA, R. The time complexity of permutation routing via matching, token swapping and a variant. *J. Graph Algorithms Appl. 23*, 1 (2019), 29–70.

[4] MILTZOW, T., NARINS, L., OKAMOTO, Y., ROTE, G., THOMAS, A., AND UNO, T. Approximation and hardness of token swapping. In *24th Annual European Symposium on Algorithms, ESA 2016, August 22-24, 2016, Aarhus, Denmark* (2016), P. Sankowski and C. D. Zaroliagis, Eds., vol. 57 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 66:1–66:15.

[5] Yamanaka, K., Demaine, E. D., Ito, T., Kawahara, J., Kiyomi, M., Okamoto, Y., Saitoh, T., Suzuki, A., Uchizawa, K., and Uno, T. Swapping labeled tokens on graphs. *Theor. Comput. Sci. 586* (2015), 81–94.

24: Friends-and-Strangers Graphs

(suggested by Yoshio Okamoto)

Friends-and-strangers graphs are defined by Defant and Kravitz [2], which encompasses several combinatorial problems such as the 15-puzzle and the token swapping problem. Its combinatorial aspects have been studied, but not well understood. Less is known for algorithmic aspects.

To define a friends-and-strangers graph, we need two undirected graphs X and Y with the same number of vertices, say n. The *friends-and-strangers graph* $\mathsf{FS}(X, Y)$ is a simple graph that has all the bijections from $V(X)$ to $V(Y)$ as its vertex set, and two bijections σ and σ' are adjacent in $\mathsf{FS}(X, Y)$ if and only if there exists an edge $\{a, b\} \in E(X)$ such that $\{\sigma(a), \sigma(b)\} \in E(Y)$, $\sigma(a) = \sigma'(b)$, $\sigma(b) = \sigma'(a)$, and $\sigma(x) = \sigma'(x)$ for all $x \in V(X) - \{a, b\}$.

We interpret X as a base graph and Y as a friendship graph. The vertices of Y are a set of people each of whom sits on a vertex of X. A seat assignment corresponds to a bijection from $V(X)$ to $V(Y)$. Two people can change their seats if and only if they are friends (i.e., adjacent in Y) and joined by an edge in X.

In most of the existing work on friends-and-strangers graphs, we are concerned with connectedness. Below are samples of the results. Since $\mathsf{FS}(X, Y) \simeq \mathsf{FS}(Y, X)$ (an isomorphism can be constructed by $\sigma \mapsto \sigma^{-1}$), we concentrate on cases where X is taken from a certain class.

- When X is a complete graph K_n, the problem corresponds to the so-called token swapping problem [9]. The graph $\mathsf{FS}(K_n, Y)$ is connected if and only if Y is connected.
- When X is a star $K_{1,n-1}$, the problem corresponds to the generalized 15-puzzle. By the result of Wilson [8], we know a full characterization of the number of connected components of $\mathsf{FS}(K_{1,n-1}, Y)$.
- When X is a path P_n, the connected components of $\mathsf{FS}(P_n, Y)$ bijectively correspond to the acyclic orientations of the complement $\overline{Y}$ [2]. In particular, $\mathsf{FS}(P_n, Y)$ is connected if and only if Y is a complete graph.
- When X is a cycle C_n, the connected components of $\mathsf{FS}(C_n, Y)$ are determined by a certain equivalence relation over the family of acyclic orientations of $\overline{Y}$ [2]. In particular, $\mathsf{FS}(C_n, Y)$ is connected if and only if $\overline{Y}$ is a forest in which the orders of the connected components are (setwise) coprime.

Other classes of graphs, such as lollipops [6] and complete bipartite graphs [7], have also been investigated. I should point out that there are a few papers on extremal and probabilistic aspects [1, 3].

As an interesting special case, I would like to pose the following question.

> **Question 1:** When X is a tree, what can we say about the connected components of $\mathsf{FS}(X, Y)$? Is there a (natural) correspondence between the connected components and another (interesting) combinatorial structures?

Even when X is a spider, only a partial result is known [5].

Another direction is algorithmic. Jeong [4] posed several algorithmic questions for friends-and-strangers graphs. Along this direction, the following local connectedness problem is known to be PSPACE-complete [10]: we are given two graphs X, Y and two bijections $\sigma, \sigma' \colon V(X) \to V(Y)$, and we are asked to determine whether σ and σ' belong to the same connected component of $\mathsf{FS}(X, Y)$. The following question is a version when X is taken from the class of trees.

> **Question 2:** Is the local connectedness problem still hard (e.g. PSPACE-complete) even when X is constrained to be trees?

There can be many other interesting research directions around friends-and-strangers graphs. Other open problems can be found in the literature.

References

[1] ALON, N., DEFANT, C., AND KRAVITZ, N. Typical and extremal aspects of friends-and-strangers graphs. *J. Comb. Theory, Ser. B 158*, Part (2023), 3–42.

[2] DEFANT, C., AND KRAVITZ, N. Friends and strangers walking on graphs. *Combinatorial Theory 1* (2021).

[3] JEONG, R. Bipartite friends and strangers walking on bipartite graphs. *arXiv 2309.03848 [math.CO]* (2023).

[4] JEONG, R. On the diameters of friends-and-strangers graphs. *arXiv 2201.00665 [math.CO]* (2023).

[5] LEE, A. Connectedness in friends-and-strangers graphs of spiders and complements. *arXiv 2210.04768 [math.CO]* (2022).

[6] WANG, L., AND CHEN, Y. The connectedness of the friends-and-strangers graph of a lollipop and others. *Graphs Comb. 39*, 3 (2023), 55.

[7] WANG, L., LU, J., AND CHEN, Y. Connectedness of friends-and-strangers graphs of complete bipartite graphs and others. *Discret. Math. 346*, 8 (2023), 113499.

[8] WILSON, R. M. Graph puzzles, homotopy, and the alternating group. *Journal of Combinatorial Theory, Series B 16*, 1 (1974), 86–96.

[9] YAMANAKA, K., DEMAINE, E. D., ITO, T., KAWAHARA, J., KIYOMI, M., OKAMOTO, Y., SAITOH, T., SUZUKI, A., UCHIZAWA, K., AND UNO, T. Swapping labeled tokens on graphs. *Theor. Comput. Sci. 586* (2015), 81–94.

[10] YANG, C., AND ZHANG, Z. Friends-and-strangers is PSPACE-complete. *arXiv 2402.03685 [math.CO]* (2024).

25: Matchings extending cycles in hypercubes

(suggested by Jirka Fink)

The d-*dimensional hypercube* Q_d is the graph whose vertices are all subsets of $[d] := \{1, \dots, d\}$ and whose edges connect sets that differ in a single element. It is well-known and easy to show that Q_d, $d \geqslant 2$, admits a *Hamilton cycle*, i.e., a cycle that visits every vertex exactly once. Clearly, any Hamilton cycle in Q_d is the union of two perfect matchings. A *matching* in a graph is a set of edges that are pairwise disjoint, and a matching is *perfect* if it includes every vertex of the graph.

30 years ago, Ruskey and Savage [7] conjectured that every matching of Q_d can be extended to a Hamilton cycle.

Question 1: Can every matching of Q_d, $d \geqslant 2$, be extended to a Hamilton cycle?

This problem received considerable attention, and several natural relaxations have been proved. In particular, Fink [2] settled the conjecture affirmatively for the case when the prescribed matching is perfect, thereby answering a problem due to Kreweras [6]. In fact, Fink established a considerable strengthening, obtained by considering the graph $K(Q_d)$, which is the complete graph on the vertex set of Q_d. In this context, we say that a matching M of $K(Q_d)$ *extends to* a Hamilton cycle C (or some other structure) if all edges in $C \setminus M$ belong to Q_d.

Dvořák and Fink [1] proved that every matching of Q_d, $d \geqslant 2$, with at most $d^2/16 + d/4$ edges can be extended to a Hamilton cycle. Another relaxation of the Ruskey-Savage conjecture, proposed by Vandenbussche and West [9], is to consider extensions to a *cycle factor*, i.e., a collection of disjoint cycles that together visit all vertices of the graph. This variant of the problem was also settled by Fink [3].

Fink and Mütze [4] recently proved that every matching of Q_d can be extended to a cycle of length at least $\frac{2}{3}|V(Q_d)|$. Furthermore, if M is a matching of $K(Q_d)$ the length of the extending cycle is at least $\frac{1}{2}|V(Q_d)|$. Their proof starts by the assumption that M is maximal, which has the effect that even for very small matchings M, the cycle C might be very long.

Question 2: Can we instead prove a theorem about a cycle C extending a given matching M, such that the length of C is relatively short compared to the size of M, i.e., such that $|C| \leqslant f(|M|)$ for some reasonable function f? Obviously, we will always have $|C| \geqslant 2|M|$. More generally, given a matching M of Q_d or $K(Q_d)$, for which integers ℓ is there a cycle C with $|C| = \ell$ that extends M?

This includes the Ruskey-Savage conjecture as a special case (when $\ell = 2^d$). This can also be stated as a decision problem: Given M and ℓ, does C exist, yes or no?

We also note that not every matching of $K(Q_d)$ can be extended to a Hamilton cycle, nor a cycle factor. In fact, any matching in $K(Q_d)$ between all vertices of the same parity has the property that any cycle extending it includes only half of the vertices from the other parity, so it has only length $\frac{3}{4}|V(Q_d)|$. The *parity* of a vertex u of Q_n is the parity of $|u|$, i.e., of the size of the set u. In particular, in $K(Q_2)$, the matching with the single edge $(\emptyset, \{1,2\})$ can only be extended to a cycle of length 3.

Clearly, if a matching M of $K(Q_d)$ is extendable to a Hamilton cycle, then M has the same number of vertices of each parity. Note that every matching of $B(Q_d)$ satisfies this condition, where $B(Q_d)$ is the complete bipartite graph obtained from Q_d by adding all edges between vertices of opposite parity. However, Dvořák and Fink [1] showed that this condition is not sufficient, by constructing a matching of $B(Q_d)$ for $d \geqslant 9$ which cannot be extended to a Hamilton cycle, nor a cycle factor. The extendibility of a matching of $K(Q_d)$ to cycle factor can be determined in polynomial time. So, are there other obstacles for a matching of $K(Q_d)$ to be extendable to Hamilton cycle?

> **Question 3:** Is there a matching of $K(Q_d)$ or $B(Q_d)$ that is extendable to a cycle factor but not to a Hamilton cycle?

The hypercube Q_d is *Hamilton-laceable*, i.e., it admits a Hamilton path between any two prescribed end vertices of opposite parity [8]. Gregor, Novotný, and Škrekovski [5] considered laceability combined with matching extensions. Specifically, they considered the problem of extending a perfect matching of Q_d to a Hamilton path between two prescribed end vertices with opposite parity. Their proof works in the more general setting of the complete bipartite graph $B(Q_d)$. For a matching M of Q_d and one of its vertices x, we write x^M for the other end vertex of the edge of M incident with x. They proved that for two vertices x, y of opposite parity in Q_d, a perfect matching M of $B(Q_d)$ with $xy \notin M$ can be extended to a Hamilton path with end vertices x and y if and only if and $M \setminus \{xx^M, yy^M\} \cup \{x^M y^M\}$ contains no so-called half-layers. They conjectured that the statement can be generalized for matchings of $K(Q_d)$, which was proved by Fink and Mütze [4].

> **Question 4:** Which matchings of Q_d or $K(Q_d)$ can be extended to Hamilton paths with prescribed ends?

References

[1] DVOŘÁK, T., AND FINK, J. Gray codes extending quadratic matchings. *J. Graph Theory 90*, 2 (2019), 123–136.

[2] FINK, J. Perfect matchings extend to Hamilton cycles in hypercubes. *J. Combin. Theory Ser. B 97*, 6 (2007), 1074–1076.

[3] FINK, J. Matchings extend into 2-factors in hypercubes. *Combinatorica 39*, 1 (2019), 77–84.

[4] FINK, J., AND MÜTZE, T. Matchings in hypercubes extend to long cycles. *arXiv preprint arXiv:2401.01769* (2024).

[5] GREGOR, P., NOVOTNÝ, T., AND ŠKREKOVSKI, R. Extending perfect matchings to Gray codes with prescribed ends. *Electron. J. Combin. 25*, 2 (2018), Paper No. 2.56, 18 pp.

[6] KREWERAS, G. Matchings and Hamiltonian cycles on hypercubes. *Bull. Inst. Combin. Appl. 16* (1996), 87–91.

[7] Ruskey, F., and Savage, C. Hamilton cycles that extend transposition matchings in Cayley graphs of S_n. *SIAM J. Discrete Math. 6*, 1 (1993), 152–166.

[8] Simmons, G. J. Almost all n-dimensional rectangular lattices are Hamilton-laceable. *Congr. Numer. XXI* (1978), 649–661. Proceedings of the Ninth Southeastern Conference on Combinatorics, Graph Theory and Computing (Boca Raton, FL, 1978).

[9] Vandenbussche, J., and West, D. B. Extensions to 2-factors in bipartite graphs. *Electron. J. Combin. 20*, 3 (2013), Paper 11, 10 pp.

26: Cup stacking of minimal weight on Hamilton paths

(suggested by Petr Gregor)

Cup stacking is a one-player game on a connected graph introduced by Fay, Hurlbert and Tennant [1]. Initially, each vertex has a single cup placed on it, i.e. a stack of weight 1. In each move, an entire stack of k cups from one vertex is added to a nonempty stack on another vertex at distance k. Thus a stack of weight k moves exactly by distance k. The goal is to end up with all cups stacked on a single target vertex. If there exists cup stacking to any target vertex, the graph is called *stackable*. Very recently, we have shown that every traceable graph is stackable [2].

Here we suggest to study the following associated optimization problem. The *weight* of stacking is the sum of weights (i.e. traversed distances) in all moves. Equivalently, it is the total depth of vertices in a corresponding tree of stacking moves. Ideally, we want to determine the minimum weight of stacking in a graph G to a target t, denoted by $\mu(G,t)$.

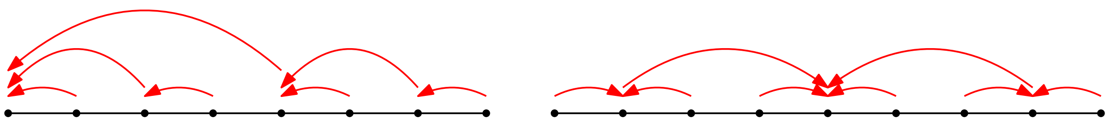


Figure 22: *Cup stacking of minimal weight* 12 *on paths* P_8 *(to the left end vertex) and* P_9 *(to the middle vertex), i.e.* $\mu(P_8,1) = \mu(P_9,5) = 12$.

> **Question 1:** What is the value of $\mu(P_n,i)$ for a path P_n, $n \geqslant 1$, and vertex $i \in [n]$?

It is easy to show that $\mu(P_n,1) \leqslant \frac{1}{2} n \log_2 n$ if $n = 2^k$, and $\mu(P_n,\lceil n/2 \rceil) \leqslant \frac{2}{3\log_2 3} n \log_2 n \doteq 0.42 n \log_2 n$ if $n = 3^k$. Here are the exact values for all $n \leqslant 12$ and $i \in [n]$:

1	$\{0\}$
2	$\{1,1\}$
3	$\{3,2,3\}$
4	$(4,4,4,4)$
5	$(6,5,6,5,6)$
6	$(9,7,7,7,7,9)$
7	$(11,10,9,8,9,10,11)$
8	$(12,12,12,10,10,12,12,12)$
9	$(14,13,14,13,12,13,14,13,14)$
10	$(17,15,15,15,15,15,15,15,15,17)$
11	$(19,18,17,16,17,18,17,16,17,18,19)$
12	$(22,20,20,18,18,20,20,18,18,20,20,22)$

> **Question 2:** Given a Hamilton path P in a graph, which strategy of stacking along the path P to a target t has smallest weight?

The strategy described in [2] always picks the first vertex s of P and inductively stacks the cups from the first $d(s,t)$ vertices of P to the vertex s, which are then moved from s to the target t. Then it proceeds similarly with the next available vertex until all vertices on P before the target t are empty. This is then repeated on the other side of P from the last vertex of P.

Question 3: Given a traceable graph G, which Hamilton path in G allows a stacking strategy of smallest weight?

This question is interesting for "highly" Hamilton graphs, for example hypercubes.

References

[1] Fay, P., Hurlbert, G., and Tennant, M. Cup stacking in graphs. *arXiv:2310.06192*, 2023.

[2] Gregor, P., Merino, A., and Mütze, T. Hamiltonian graphs are cup-stackable. *arXiv:2401.06189*, 2024.