KAMAK 2023

Pension Popelka, Špindlerův Mlýn

September 10 – 15

Charles University, Prague

Organizers:

David Mikšaník
Aneta Pokorná
Robert Šámal

Brochure of open problems,
Prague, 2023.

Contents

Program

8:00 breakfast

9:00 morning session I

10:30 break

11:00 morning session II

12:30 lunch

15:00 afternoon session I

16:30 break

17:00 afternoon session II

18:30 progress reports

19:00 dinner

Wednesday is planned to be free to make an excursion in the neighborhood with everyone who would like to come.

OPEN PROBLEMS

Problem 1. *Do directed graphs constructed from $\overrightarrow{P_2}$ by edgeless cut joins and creating twins have bounded dichromatic number? (suggested by Guillaume Aubian)*

Source: Proposed by Bonnet, Bourneuf, Geniet, Thomassé and Trotignon.

Definitions.
• *Let D, D' be two directed graphs.*
• *A k-dicolouring of D is a colouring of its vertices so that there is no monochromatic directed cycle.*
• *The dichromatic number of D is the minimum integer k such that D admits a k-dicolouring.*
• *$\overrightarrow{P_2}$ is the directed path on two vertices.*
• *Creating a twin v' of $v \in V(D)$ consists in adding a new vertex v' with the same inneighbours and outneighbours as v (we do not add arcs vv' nor $v'v$).*
• *An edgeless cut join of D and D' is a directed graph obtained by glueing D and D' along a stable set of size at most two.*

Question: Does the (smallest) class of directed graphs containing $\overrightarrow{P_2}$ and preserved by edgeless cut joins and creating twins have bounded dichromatic number?

Related results:
• The directed graphs that are constructed are triangle-free.
• Creating a twin cannot increase the dichromatic number.
• If we don't use twins, the dichromatic number is bounded (by degeneracy).
• Finding a construction with dichromatic number > 2 would be nice...
• In the undirected case, the chromatic number is unbounded.

Reference:

Bonnet, Bourneuf, Geniet, Thomassé and Trotignon: *A tamed family of triangle-free graphs with unbounded chromatic number*. 2023, arXiv.

Problem 2. *Erdos-Gallai conjecture (suggested by Sudatta Bhattacharya)*

Source: Proposed by Erdos and Gallai in 1960's.

Question: Prove that any n-vertex graph can be decomposed into $\mathcal{O}(n)$ cycles and edges.

Related results:
• In 2014, Fox, Conlon and Sudakov [2] made the first major breakthrough on this problem, showing that such a decomposition with only $\mathcal{O}(n \log\log n)$ cycles and edges always exists.
• Recently, Bucic and Montegamory [1] in 2023 showed that any n-vertex graph can be decomposed into $\mathcal{O}(n \log^* n)$ cycles and edges.

References:

[1] Matija Bucić and Richard Montgomery: *Towards the Erdős-Gallai cycle decomposition conjecture.* In Proceedings of the 55th Annual ACM Symposium on Theory of Computing, pages 839–852, 2023.

[2] David Conlon, Jacob Fox and Benny Sudakov: *Cycle packing.* Random Structures & Algorithms, 45(4):608–626, 2014.

Problem 3. *Kernel for Travelling Salesman Problem with respect to Feedback Vertex Set number (suggested by Václav Blažej)*

Definitions.
• Travelling Salesman Problem (TSP) – *Given a graph G with edge costs and a budget b. Walk pays for every edge traversal its cost. Does there exist a closed walk that visits every vertex of G but pays at most b?*
• Feedback Vertex Set (FVS) – *A set of vertices X that hits all cycles, i.e., $G \setminus X$ is a forest.* Feedback Vertex Set number k *is size of the minimum cardinality FVS.*
• Kernelization – *A preprocessing algorithm that runs in polynomial time and reduces the input instance to size $f(k)$ (for some computable function f).*
• Modulator to $\mathcal{C}$ – *A set of vertices X of G such that $G \setminus X$ is in $\mathcal{C}$.*

Example:
Modulator to isolated vertices is the Vertex Cover. Modulator to trees is the Feedback Vertex Set.

Question: For TSP of input size n parameterized by FVS k, is there a polynomial preprocessing algorithm (running in poly(n)) that reduces the input instance to size poly(k) while preserving the answer?

Related results:
• TSP is NP-complete.
• TSP is FPT with respect to Treewidth; hence some kernel exists.
• TSP has no polynomial kernel with respect to Treewidth.
• TSP has a polynomial kernel with respect to Modulator to Stars.
• TSP has a polynomial kernel with respect to Modulator to Constant Trees.
• TSP has a polynomial kernel with respect to Feedback Edge Set number.

References:

B., Choudhary, Knop, Schierreich, Suchý, Valla: *On Polynomial Kernels for Traveling Salesperson Problem and its Generalizations.*

Problem 4. *Weak saturation of $K_{2,2,2}$ (suggested by Denys Bulavka)*

Source: Proposed by Gal Kronenberg, Taísa Martins, Natasha Morrison in *Weak saturation numbers of complete bipartite graphs in the clique.* J. Combin. Theory Ser. A, 178:105357, 15, 2021.

Definitions.
• *Given a host graph H and a pattern graph P, a graph $G \subseteq H$ is P* **weakly saturated in** *H if there exists an order of the edge set $H \setminus G = \{e_1, \ldots, e_k\}$ such that for each $i \in [k]$ there exists a graph $P_i \subseteq G \cup \{e_1, \ldots, e_i\}$ isomorphic to P and $e_i \in P_i$.*
• *The* **weak saturation number of** *P* **in** *H is the minimum number of edges in a P wealy saturated subgraph of H can have. We denote this quantity by $\mathrm{wsat}(H, P)$.*

Example: Upper bound for $\mathrm{wsat}(K_n, K_{2,2,2})$.

Question: Determine $\mathrm{wsat}(K_n, K_{2,2,2})$.

Related results:
The following are known with the host graph being the clique:
• $\mathrm{wsat}(K_n, K_{r+2}) = rn - \binom{r+1}{2}$.
• $\mathrm{wsat}(K_n, K_{r+1,r+1}) = rn - \binom{r+1}{2}$.
• $\mathrm{wsat}(K_n, K_{r+1,r+2}) = rn - \binom{r+1}{2} + 1$.
• If $n \geq 2r - 1$, $\mathrm{wsat}(K_n, K_{2,r}) = n - 2 + \binom{r}{2}$.
• $\mathrm{wsat}(K_n, K_{s+1,t+1}) = sn - c(s, t)$.

References:
The weak saturation number of $K_{2,t}$ by Meysam Miralaei, Ali Mohammadian, Behruz Tayfeh-Rezaie. Preprint: arXiv:2211.10939.

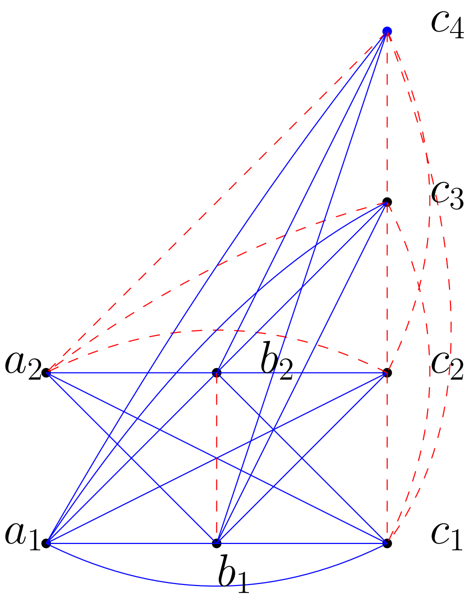

Figure 1: Upper bound example for $\mathrm{wsat}(K_n, K_{2,2,2})$.

Problem 5. *Characterize all graphs with five boundary vertices (suggested by Guillermo Gamboa)*

Source: Proposed by Stefan Steinerberger in 2022.

Definitions.
• *Let $G = (V, E)$ be a graph. The boundary ∂G of G is defined as*

$$\partial G = \left\{ u \in V \middle| \exists u \in V : \frac{1}{\deg(u)} \sum_{(u,w)\in E} d(w, v) < d(u, v) \right\}.$$

Related results:
• In [2], Steinerberger extablishes the following isoperimetric inequality: if G is a connected graph with maximal degree Δ, then for all $v \in V$

$$\left| \left\{ u \in V \middle| \frac{1}{\deg(u)} \sum_{(u,w)\in E} d(w, v) < d(u, v) \right\} \right| \geq \frac{|V|}{2\Delta \operatorname{diam}(G)}.$$

This implies that

$$|\partial G| \geq \frac{|V|}{2\Delta \operatorname{diam}(G)}.$$

• In [1], Chiem, Dudarov, Lee, Lee and Liu characterize graphs with at most four boundary vertices.

References:
[1] Chiem N., Dudarov W., Lee C., Lee S., Liu K: *A characterization of graphs with at most four boundary vertices.* 2023. arXiv: 2209.04438 [math.CO].

[2] Steinerberger S: *The boundary of a graph and its isoperimetric inequality.* Discrete Applied Math., 338 (2023), pp. 125-134.

Problem 6. *Number of independent sets in subgraphs of a random graph (suggested by Pavel Koblich Dvořák)*

Definitions.

• *A* random graph $G(n, p)$ *is a graph on n vertices where each edge is present with a probability p independently on the other edges.*

Let $G = G(n, p)$ be a random graph. Then, the expected number of independent sets of size k in G is $\mathrm{IS}(n, p, k) = \binom{n}{k} \cdot (1-p)^{\binom{k}{2}}$. Analogous formula holds for any subgraph of G, i.e., a subgraph H of G with n' vertices contains $\mathrm{IS}(n', p, k)$ independent sets of size k in expectation. However, what about all subgraphs of given size at once?

Question: Is it true that with high probability all subgraphs of G with n' vertices contains roughly $\mathrm{IS}(n', p, k)$ independent sets of size k? Or there is always a subgraph that contains significantly more k-size independent sets?

These questions arised from streaming algorithm for the independent set problem. Sufficiently strong answer should improve either the algorithm or the lower bound.

Problem 7. *Is Network Coding conjecture true for expanders? (suggested by Pavel Koblich Dvořák)*

Definitions.

- *A* network *consists of a graph $G = (V, E)$, positive capacities of edges $c : E \to \mathbb{R}^+$ and k pairs of vertices $(s_0, t_0), \ldots, (s_{k-1}, t_{k-1})$.*
- *A* coding scheme *for a network R is a collection of function that specifies messages sent from each vertex v to its neighbors based on the messages sent to v. The goal of the scheme is that in each target t_i of R we can reconstruct an input message w_i that is received at the source s_i. Moreover, the lengths of the messages sent along the edges have to respect the edge capacities.*
- *A* coding rate $r_c(R)$ *of a network R is the maximum r such that there is a coding scheme for R that sends a message of length at least r from each source to its corresponding target.*
- *A* multicommodity flow *for a network R specifies flows for each commodity i such that they transport as many units of commodity from s_i to t_i as possible and not exceeding capacity of any edge.*
- *A* flow rate $r_f(R)$ *of a network R is the maximum r such that there is a multicommodity flow that transport r units of each commodity.*

Network Coding Conjecture:
Let R be a directed network, and $\bar{R}$ be an undirected network arising from R by removing the direction of all edges. Then, $r_c(R) = r_f(\bar{R})$.

The conjecture is proved for only small graphs or somehow easy simple classes. A natural candidate of a graph class for disproving the conjecture is any class of graph expanders. Either of disproving the conjecture or proving the conjecture for a class of expanders would make an interesting result.

Problem 8. *Sorting by block reversals (suggested by Vít Jelínek)*

Source: This turned up in my joint research with Michal Opler and Jakub Pekárek

Question: 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.

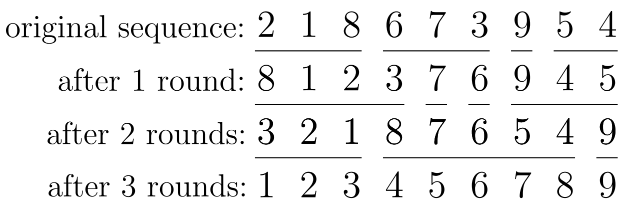

Figure 2: Sorting the sequence 2,1,8,6,7,3,9,5,4 by parallel block reversals. The horizontal lines indicate which blocks of consecutive elements are reversed in a given round.

The problem is to determine 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 $\{1, 2, \ldots, n\}$ exactly once.

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 $B_1, B_2, \ldots, B_{2k}$, and then for each $i \leq k$, 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:

This problem occurred within research of a more general concept of sorting-time of hereditary permutation classes (joint work of Vít Jelínek, Michal Opler and Jakub Pekárek), which has been submitted for publication but has not appeared yet.

Problem 9. *Reversed graph coloring game (suggested by David Mikšaník)*

Given a (simple) graph G and positive integer k, a graph coloring game on G with k colors consists of two players, Alice (the maker) and Bob (the breaker). Alice and Bob alternately color uncolored vertices in G (in the beginning, all vertices in G are uncolored) from the set $\{1, 2, \ldots, k\}$ so that at any time no adjacent vertices have the same color. The game ends when all vertices in G are colored (in this case Alice wins) or there exists an uncolored vertex in G such that its neighborhood contains all colors from $\{1, 2, \ldots, k\}$ (in this case Alice loses).

In the literature, Alice usually starts the game and we are interested in the least number k such that Alice has a winning strategy. In this problem, we are interested under which conditions it is better for Bob to start the game.

Definitions.

• *Alice game-chromatic number of G, denoted $\chi_g^A(G)$, is the least number k for which Alice has a winning strategy provided that Alice starts the game.*

• *Bob game-chromatic number of G, denoted $\chi_g^B(G)$, is the least number k for which Alice has a winning strategy provided that Bob starts the game.*

Example:

• $\chi_g^A(K_{n,n}) = 3$ and $\chi_g^B(K_{n,n}) = 2$.

• Let $K_{n,n} - nK_2$ be the graph obtained from $K_{n,n}$ by removing a perfect matching. Then

$$\chi_g^A(K_{n,n} - nK_2) = n \quad \text{and} \quad \chi_g^B(K_{n,n} - nK_2) = 2.$$

On the other hand, let G be $K_{n,n} - nK_2$ plus an isolated vertex. Then

$$\chi_g^A(G) = 2 \quad \text{and} \quad \chi_g^B(G) = n.$$

• The class of planar graphs has bounded χ_g^A.

Question 1: Determine sufficient conditions for a graph G to satisfy

$$\chi_g^B(G) > \chi_g^A(G).$$

Question 2: Suppose that Alice has a winning strategy for the graph coloring game on G with k colors. Does Alice have a winning strategy with $k+1$ colors? Or at least, with $f(k)$ colors for some function $f(k) > k$? [2]

Related results:
• Given a graph G, it is PSPACE-hard to determine $\chi_g^A(G)$. [1]

References:
[1] Eurinardo Costa, Victor Pessoa, Rudini Sampaio, Ronan, Soares: *PSPACE-completeness of two graph coloring games.* Theoretical Computer Science, Vol. 824-825, pp. 36-45 (2020) doi: 10.1016/j.tcs.2020.03.022.

[2] Xuding Zhu: *The Game Coloring Number of Planar Graphs.* Journal of Combinatorial Theory, Series B, Vol. 75, Issue 2, pp. 245-258 (1999) doi: https://doi.org/10.1006/jctb.1998.1878.

Problem 10. *Max-Min Odd and Even Cycle Transversals (suggested by Nikolaos Melissinos)*

Source: Proposed by Michael Lampis in 2020.

Definitions.
- *Given a graph $G = (V, E)$, a sub $S \subseteq V$ is a minimal odd cycle transversal if $G[V \setminus S]$ has no odd cycles (i.e. it is bipartite) and there is no subset of S with the same property.*
- *Given a graph $G = (V, E)$, a sub $S \subseteq V$ is a minimal even cycle transversal if $G[V \setminus S]$ has no even cycles and there is no subset of S with the same property.*
- *In Max-Min Odd (Even) Cycle Transversal we are searching for a minimal odd (even) cycle transversal, of a given graph, of maximum order.*

Example:

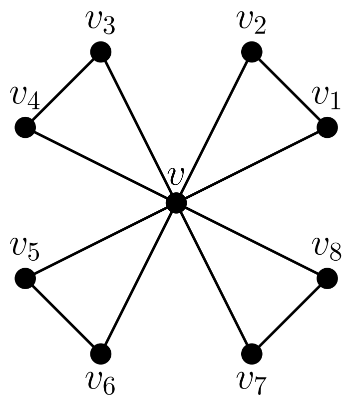

Figure 3: Example graph.

We consider the graph given in Figure 3. The sets $S_1 = \{v\}$, $S_2 = \{v, v_1\}$ and $S_3 = \{v_1, v_3, v_5, v_7\}$ are odd cycle transversals.

Notice that S_1 and S_3 are minimal while S_2 is not. Furthermore, S_1 is a minimum odd cycle transversal and S_3 is a minimal odd cycle transversal of maximum order. In Max-Min Odd Cycle Transversal we are searching for S_3 (or any other minimal odd cycle transversal of maximum order).

Question: Can we found an $n^{2/3}$-approximation algorithm for Max-Min Odd (Even) Cycle Transversal?

Related results and relations with the proposed problems:

• The Max-Min Feedback vertex set problem (Max-Min FVS) is inapproximable within a factor of $n^{2/3-\varepsilon}$ unless $P = NP$ [1]. This result should be easily extendable to the Max-Min Odd (Even) Cycle Transversal. The same holds for the NP-hardness of the problem on planar graphs of maximum degree 6.

• In [1] were also presented two approximation algorithms for Max-Min FVS. These algorithms seem to be more challenging to adapt for Max-Min Odd (Even) Cycle Transversal.

• In [2,3] someone can find several results related to the parameterized version of Max-Min FVS. Once again, the negative results seem to be easily easily extendable. It would be nice to see whether the same holds for the positive results.

References:

[1] Louis Dublois, Tesshu Hanaka, Mehdi K. Ghadikolaei, Michael Lampis, and Nikolaos Melissinosi: *(In)approximability of maximum minimal FVS.* Journal of Computer and System Sciences, 124:26–40, 2022.

[2] Ajinkya Gaikwad, Hitendra Kumar, Soumen Maity, Saket Saurabh, and Shuvam Kant Tripathi: *Maximum minimal feedback vertex set: A parameterized perspective.* CoRR, abs/2208.01953, 2022.

[3] Michael Lampis, Nikolaos Melissinos and Manolis Vasilakis: *Parameterized Max Min Feedback Vertex Set.* 48th International Symposium on Mathematical Foundations of Computer Science. In MFCS 2023, volume 272 of LIPIcs, pages 62:1–62:15.

Problem 11. *Construction of constant gap sequences (suggested by Daniela Opočenská)*

Source: An equivalent problem was introduced by Paul Erdős in the early 1930s under the name *exact covering systems*, more info can be found in [3]. We met constant gap sequences when solving problems in combinatorics on words [1]. An efficient way of construction of constant gap sequences plays an important role there and seems to be an unsolved problem.

Definitions.

• *An* alphabet $\mathcal{A}$ *is a finite set of symbols that are called* letters*.*

• *A* sequence *over an alphabet* $\mathcal{A}$ *is an infinite sequence of letters from the alphabet, we write* $\mathbf{y} = y_0 y_1 y_2 \cdots$*, where* $y_n \in \mathcal{A}$ *for all* $n \in \mathbb{N}_0$*.*

• *A sequence* $\mathbf{y} = y_0 y_1 y_2 \cdots$ *is* periodic*, if there exists* $P \in \mathbb{N}$ *such that for all* $n \in \mathbb{N}_0$*,* $y_n = y_{n+P}$*. The smallest possible* P *fulfilling this condition is called the* period*. In this case, we often write* $\mathbf{y} = (y_0 y_1 y_2 \cdots y_{P-1})^\omega$ *to symbolize infinite repetition of the first* P *letters.*

• *A sequence* $\mathbf{y}$ *over an alphabet* $\mathcal{A}$ *is a* constant gap sequence *if for each letter* $i \in \mathcal{A}$ *there is a positive integer denoted by* p_i *such that the distance between any two consecutive occurrences of* i *in* $\mathbf{y}$ *is* p_i*.*

• *We denote* n_i *the first occurrence of the letter* i *in* $\mathbf{y}$*.*

• *Let us consider* k *sequences over disjoint alphabets. Their* shuffling *is a sequence obtained when reading step by step their first letters, second letters, etc.*

Example:

• All constant gap sequences are periodic.

• $\mathbf{y} = (0102)^\omega = 01020102\cdots$ is a constant gap sequence, where the letter periods are $p_0 = 2$, $p_1 = 4 = p_2$, the first occurrences are $n_0 = 0$, $n_1 = 1$, $n_2 = 3$ and the period is 4.

• $\mathbf{v} = (0122)^\omega = 012201220122\cdots$ is periodic but not constant gap because of the occurences of 2.

• The constant gap sequence can be described by a list of pairs (n_i, p_i) for all $i \in \mathcal{A}$. For example, $\mathbf{y} = (010203)^\omega$ can be rewritten as $(0,2)$, $(1,6)$, $(3,6)$, $(5,6)$.

• We can shuffle the constant gap sequences $\mathbf{u} = (0)^\omega$, $\mathbf{v} = (12)^\omega$, $\mathbf{w} = (3435)^\omega$ to obtain a sequence over the alphabet $\mathcal{A} = \{0,1,2,3,4,5\}$ in the form

$$\mathbf{y} = (013024013025)^\omega.$$

Question: Find an algorithm to generate all constant-gap sequences over a given alphabet such that it works well for alphabets with at least 13 letters.

Related results:

• Let $\mathbf{y}$ be a constant gap sequence over an alphabet $\mathcal{A} = \{0,1,2,3, \ldots, d-1\}$. Then

$$\sum_{i=0}^{d-1} \frac{1}{p_i} = 1.$$

• Let $n, n' \in \mathbb{N}_0$, and $p, p' \in \mathbb{N}$. We say that (n,p) and (n',p') are *in collision* if $\gcd(p,p')$ divides $n - n'$. Let the pairs $(n_0,p_0), (n_1,p_1), \ldots, (n_{d-1},p_{d-1})$ satisfy $n_i < p_i$ for all $i \in \mathcal{A}$, Then the pairs $(n_0,p_0), (n_1,p_1), \ldots, (n_{d-1},p_{d-1})$ give rise to a constant gap sequence iff no two of them are in collision and

$$\sum_{i=0}^{d-1} \frac{1}{p_i} = 1.$$

• The period of the letter i in the shuffling of k sequences equals $k \cdot \hat{p}_i$, where $\hat{p}_i$ is the period of the same letter in its original sequence.

Moreover, if there exists $k \in \mathbb{N}, k > 1$, such that k divides p_i for all $i \in \mathcal{A}$, then the constant gap sequence $\mathbf{y} = y_0 y_1 y_2 \cdots$ is the shuffling of k constant gap sequences in the form

$$\begin{aligned}
&y_0 y_k y_{2k} y_{3k} \cdots \\
&y_1 y_{k+1} y_{2k+1} y_{3k+1} \cdots \\
&y_2 y_{k+2} y_{2k+2} y_{3k+2} \cdots \\
&\vdots \\
&y_{k-1} y_{2k-1} y_{3k-1} y_{4k-1} \cdots
\end{aligned}$$

Therefore a constant gap sequence over a d-letter alphabet that cannot be generated by shuffling has $\gcd(p_0, p_1, \ldots, p_{d-1}) = 1$. The smallest d such that there exists a d-letter constant gap sequence, which is not obtained by shuffling, is 13.

• A brute force search algorithm to find all constant gap sequences over a d-letter alphabet, where d is user input, is described in [4]. The algorithm runs well on standard user computer up to $d = 12$, but the computational complexity becomes too much for bigger alphabets.

• This problem is equivalent to a problem of finding an exact covering system with a given number of equivalence classes. The *exact covering system* is a system of congruence classes in the form

$$n \equiv n_i \mod p_i, \qquad i = 1, 2, 3, \ldots, k$$

such that any $n \in \mathbb{N}_0$ belongs to exactly one congruence class.

The duality is as follows: There exists a constant gap sequence defined by pairs $(n_0, p_0), (n_1, p_1), \ldots, (n_{d-1}, p_{d-1})$, if and only if

$$n \equiv n_i \mod p_i, \qquad i = 0, 1, \ldots, d-1$$

is an exact covering system. There are several theoretical results concerning covering systems, some of them can be found in [2, 5, 6].

References:

[1] L. Dvořáková, D. Opočenská, E. Pelantová: *Asymptotic repetitive threshold of balanced sequences*. Mathematics of Computation, Vol. 92, (2023), pp . 1403–1429.

[2] J. Fabrykowski, T. Smotzer: *Covering Systems of Congruences.* Mathematics Magazine, (2005), 78:3, pp. 228-231.

[3] R. K. Guy: *Unsolved Problems in Number Theory.* Springer Science & Business Media, (2004), pp. 386–390.

[4] A. Kasalová: *Konstrukce slov s konstantními mezerami* (Czech). Rozhledy matematicko-fyzikální, Vol. 97 (2022), issue 3, pp. 1-12.

[5] Š. Porubský: *Generalization of Some Results for Exactly Covering Systems.* Matematický časopis, Vol. 22 (1972), No. 3, 208–214.

[6] Š. Porubský, J. Schönheim: *Old and new necessary and sufficient conditions on* (a_i, m_i) *in order that* $n \equiv a_i (\text{mod } m_i)$ *be a covering system.* Mathematica Slovaca, Vol. 53 (2003), No. 4, 341–349.

Problem 12. *Distinguishing pairs of words using finite automata (suggested by Robert Šámal)*

Source: Goralčík and Koubek [GK86].

Question: Given two words, each with at most n letters, how large finite automaton is needed to distigush them? (We measure automata by the number of states.)

Related results:
• It is known that the size needs to be $\Omega(\log n)$. [Dem11]
• It is also known that the answer does not depend on the input alphabet. Some cases where $O(\log n)$ is known to suffice:
- words of different length,
- words of same length, but different number of 1s,
- words where the number of 1s at odd positions differs,
- words where the first/last difference is close to the start/end.

• Upper bounds known for a general pair of words:
- words of different length,
- $o(n)$ [GK86],
- $\tilde{O}(n^{2/5})$ [Robson89],
- $\tilde{O}(n^{1/3})$ [Chase21].

• Most of the results are also presented in the diploma thesis by Bilan [B23].

References:

[B23] Daria Bilan: *Distinguishing Pairs of Words Using Finite Automata.* Diploma thesis, Charles University, 2023.

[Chase21] Zachary Chase: *Separating words and trace reconstruction.* STOC, 2021, doi: 10.1145/3406325.3451118.

[Dem11] Erik D. Demaine, Sarah Eisenstat, Jeffrey Shallit, and David A. Wilson: *Remarks on separating words.* In Descriptional Complexity of Formal Systems, pages 147–157. Springer Berlin Heidelberg, 2011. doi: 10.1007/978-3-642-22600-7_12.

[GK86] Goralčík, P., Koubek, V.: *On discerning words by automata.* ICALP 1986. LNCS, vol. 226, pp. 116–122. Springer, Heidelberg (1986) doi: 10.1007/3-540-16761-7_61.

[Robson89] Robson, J.M.: *Separating strings with small automata.* Inform. Process. Lett. 30, 209–214 (1989) doi: 10.1016/0020-0190(89)90215-9.

Problem 13. *Minors and quasiminors (suggested by Robert Šámal)*

Source: Mathe Bonamy et al.

Question 1: What is the infimum c such that for any large enough t there is a graph G that admits a quasi-K_t-minor, but no K_{ct}-minor?

To remind, a graph G has K_p-minor if it has nonempty pairwise disjoint and connected bags $B_1, \dots, B_p$ of vertices such that for any $i \neq j$ there is an edge between some vertex in B_i and some in B_j. A quasiminor is a weaker notion – we do not insist the bags are connected but only that $B_i \cup B_j$ (for $i \neq j$) induce a connected subgraph.

The notion of quasiminors is related to changing a coloring by means of Kempe-chain changes. This leads to a related question, a version of Hadwiger's conjecture.

Question 2: Is there a constant c' such that for every t all the $c' \cdot t$-colorings of a graph with no K_t-minor form a single equivalence class?

Related results:

- It is known that $1/2 \leq c \leq 23$ and $c' \geq 3/2$.

References:

[B] Marthe Bonamy, Marc Heinrich, Clément Legrand-Duchesne, Jonathan Narboni: *On a recolouring version of Hadwiger's conjecture.* `https://arxiv.org/abs/2103.10684`.

Problem 14. *Non-nested matching (suggested by Robert Šámal)*

Source: János Barát et al.

Question: For what m does a 2-edge-colored K_m contain a monochromatic non-nested matching of n edges?

If $a < b < c < d$ then we call edges ad and bc *nested*. We are looking for a matching where no two edges are nested.

Related results:
• It is known, that optimal m satisfies $3n - 1 \leq m \leq 4n - 2$.
• Without the non-nested condition, the optimal m is $3n - 1$. The same is true if "non-nested" is replaced by "non-crossing".

References:
János Barát, András Gyárfás, Géza Tóth: *Monochromatic spanning trees and matchings in ordered complete graphs.* `https://arxiv.org/abs/2210.10135`.