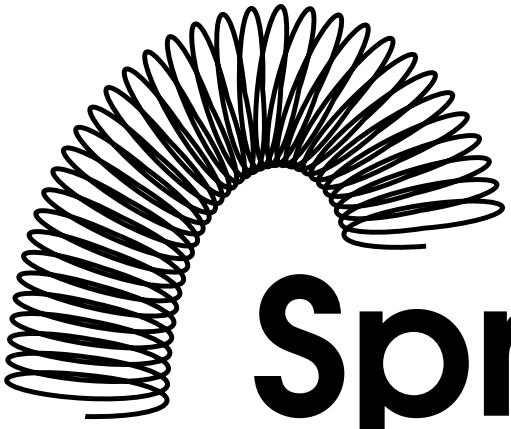


Spring School 2025

Barbora Dohnalová, Petr Chmel (eds.)

Preface

Spring school on Combinatorics has been a traditional meeting organized for more than 40 years for faculty and students participating in the Combinatorial Seminar at Faculty of Mathematics and Physics of the Charles University. It is internationally known and regularly visited by students, postdocs and teachers from our cooperating institutions in the DIMATIA network. As it has been the case for several years, this Spring School is supported by Computer Science Institute (IÚUK) of Charles University, the Department of Applied Mathematics (KAM) and by some of our grants (SVV, UNCE, Progres). This year we are glad we can also acknowledge generous support by the RSJ Foundation.

The Spring Schools are entirely organized and arranged by our students. The topics of talks are selected by supervisors from the Department of Applied Mathematics (KAM) and Computer Science Institute (IÚUK) of Charles University as well as from other participating institutions. In contrast, the talks themselves are almost exclusively given by students, both undergraduate and graduate. This leads to a unique atmosphere of the meeting, which helps the students in further studies and their scientific orientation.

This year the Spring School is organized in Pastviny (in Orlické mountains in northeastern Bohemia) with a great variety of possibilities for outdoor activities.

Robert Šámal, Pavel Veselý
Petr Chmel, Barbora Dohnalová, Júlia Križanová

KAM

CSI
IÚUK

RSJ

Contents

Series Talks

Martin Černý
cerny@kam.mff.cuni.cz

It's Bonus Time: Who Deserves What in Your Startup? A Cooperative Game Theory Approach. *as part of series* Cooperative game theory

Introduction

Startups today often try to stand out not just with their ideas, but also in how they organize their work and reward their employees. Many adopt unconventional rules, team structures, or even alternative ways of distributing pay. But when compensation doesn't feel fair, it can lead to frustration and lower motivation. In this talk, we'll take a closer look at how cooperative game theory can help us think about this problem. Specifically, we'll explore what tools and concepts it offers for designing fair payment schemes in startups—so that team members feel that their contributions are properly recognized and rewarded.

Cooperative Games: Where We Begin

We denote our set of employees as $N = \{1, 2, \ldots, n\}$. To design fair compensation, we consider what value each group of employees can generate by cooperating. This leads us to the classical notion of a *cooperative game*, as introduced by von Neumann and Morgenstern in [3].

Definition 1 (Cooperative Game) *A* cooperative game *is a pair* (N, v) *where:*

- N *is a finite set of players,*
- $v : 2^N \to \mathbb{R}$ *is a characteristic function with* $v(\emptyset) = 0$*,*
- $v(S)$ *represents the worth of coalition* $S \subseteq N$*.*

In the world of cooperative games, when the goal is to fairly divide the value of the grand coalition $v(N)$ among individual players, the most recognized solution concept is the *Shapley value* [4].

Definition 2 (Shapley Value) *For a game* (N, v)*, the* Shapley value $\varphi_i(v)$ *of player* $i \in N$ *is:*

$$\varphi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} \cdot \left[v(S \cup \{i\}) - v(S) \right].$$

Generalized Games: When Things Get More Complicated

So far, we have assumed that any group of employees (players) can freely cooperate and that the value of their cooperation is always known. Unfortunately, real-world situations are rarely that simpleMore realistic settings where additional constraints or uncertainties come into play.

1. Restricted Cooperation. In many settings, not all players can cooperate freely—perhaps due to communication limits, hierarchy, or practical constraints. Myerson [2] was the first to formalize this idea by modeling cooperation possibilities using a graph: two players can cooperate if there is a direct link between them, and larger groups can cooperate if they form a connected component. This leads to the concept of *games on graphs* and the associated *Myerson value*.

Definition 3 (Game on a Graph) *A* game on a graph *is a triple* (N, v, G) *where:*

- *N is a finite set of players,*
- *$G = (N, E)$ is an undirected graph defining feasible communication,*
- *$v : \mathcal{F} \to \mathbb{R}$ is a characteristic function defined on*

$$\mathcal{F} = \{S \subseteq N \mid G[S] \text{ is connected}\}.$$

Definition 4 (Myerson Value) *Let (N, v, G) be a game on a graph. For any coalition $S \subseteq N$, let $\mathcal{C}(S)$ denote the set of connected components of the induced subgraph $G[S]$. Define the extended characteristic function $v_G : 2^N \to \mathbb{R}$ by*

$$v_G(S) = \sum_{C \in \mathcal{C}(S)} v(C).$$

The Myerson value *of player $i \in N$ is then given by*

$$\varphi_i^M(v, G) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} \left[v_G(S \cup \{i\}) - v_G(S) \right].$$

2. Incomplete Information about Coalition Values. Sometimes, we simply don't know what certain groups of players would achieve together. This may happen if players are newly hired and haven't yet interacted, or if the company assigns tasks to fixed project teams, and we only observe the value produced by those specific teams. In such cases, we face an *incomplete game*—a setting where only some coalition values are known.

Definition 5 (Incomplete Game) *An* incomplete cooperative game *is a triple $(N, v, \mathcal{K})$ where:*

- *N is a finite set of players,*
- *$\mathcal{K} \subseteq 2^N$ is a set of known coalitions*
- *$v : \mathcal{K} \to \mathbb{R}$ is a partial characteristic function.*

When coalition values are partially unknown, we may still want to assign fair payoffs based on what we know. For this, we use a refined solution concept that extends the Shapley value to such uncertain settings. One such approach is the *UD (Uniform-dividend) value* [1], which aims to preserve fairness even under ambiguity.

Definition 6 (UD-value) *Given an incomplete game $(N, \mathcal{K}, v)$, the* uniform-dividend value *is*

$$\Phi_i^{\mathcal{K}}(v) = \sum_{\substack{S \subseteq N \\ i \in S}} \frac{\delta_v^{\mathcal{K}}(S)}{|S|},$$

where the values $\delta_v^{\mathcal{K}}(S)$ are uniquely determined by:

$$\textstyle\sum_{T \subseteq S} \delta_v^{\mathcal{K}}(T) = v(S) \text{ for all } S \in \mathcal{K}, \quad \delta_v^{\mathcal{K}}(S) = \delta_v^{\mathcal{K}}(T) \text{ if } c_{\mathcal{K}}(S) = c_{\mathcal{K}}(T).$$

Bibliography

[1] Martin Černý. A New Value for Cooperative Games on Intersection-Closed Systems. *arXiv preprint* arXiv:2501.05169 [cs.GT], 2025. https://doi.org/10.48550/arXiv.2501.05169

[2] Roger B. Myerson. Graphs and Cooperation in Games. *Mathematics of Operations Research*, 2(3):225–229, 1977.

[3] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[4] Lloyd S. Shapley. A Value for n-Person Games. In *Contributions to the Theory of Games II* (H. W. Kuhn and A. W. Tucker, eds.), Annals of Mathematics Studies, vol. 28, pp. 307–317, Princeton University Press, 1953.

Richard Mužík
richard@imuzik.cz

Integer cooperative game theory *as part of series* Cooperative game theory

Introduction

In this talk, we explore cooperative game theory in the context of integer-valued utilities and outcomes. Motivated by the realism of indivisible resources, we examine how integer constraints impact both game properties and solution concepts. We introduce and study two new classes of games—c-bounded and c-tight—and analyze their connections to established concepts like convexity and superadditivity. Additionally, we propose several new integer-valued analogues of the Shapley value and compare their properties with the classical version. The talk also includes an investigation of the nucleolus in the integer setting, highlighting key differences from the traditional, real-valued approach.

Definition 1 *The* integer cooperative game *is a pair (N, v_I), where N is a finite set of players and $v_I : 2^N \to \mathbb{Z}$ is a characteristic function. Furthermore, $v_I(\emptyset) = 0$.*

Definition 2 *For integer cooperative game $G_I \in \mathcal{G}_I^n$, the following sets are defined:*

- Integer imputation set*: $\mathcal{I}_{\mathbb{Z}}(G_I) = \mathcal{I}(G_I) \cap \mathbb{Z}^n$,*
- Integer dual imputation set*: $\mathcal{I}^{\star}_{\mathbb{Z}}(G_I) = \mathcal{I}^{\star}(G_I) \cap \mathbb{Z}^n$.*

Definition 3 *Let $c \in \mathbb{N}_0$. A cooperative game (N, v) is* c-tight *if it holds that*

$$\forall S \subseteq N : 0 \leq v(S) \leq c \wedge v(N) = c.$$

Definition 4 *Let $c \in \mathbb{N}_0$. A cooperative game (N, v) is* c-bounded *if it holds that*

$$\forall S \subseteq N : 0 \leq v(S) \leq c.$$

Theorem 5 *Let $c, n, k \in \mathbb{N}$ and $k \leq n$. The number of c-tight integer k-games is given by*

$$\binom{\binom{n}{k} - 1 + c}{c}.$$

Definition 6 *For an integer cooperative game $G_I \in \mathcal{G}_I^n$, the* Floor Shapley value $\lfloor\varphi\rfloor(G_I)$ *is given by*

$$\lfloor\varphi\rfloor(G_I) = \lfloor\varphi(G_I)\rfloor.$$

Definition 7 *For an integer cooperative game $G_I = (N, v_I) \in \mathcal{G}_I^n$, the* Efficient Floor Shapley value $\varphi^E(G_I)$ *is defined as follows:*

1. *Compute the Floor Shapley value $\lfloor\varphi\rfloor(G_I)$ and the Shapley value $\varphi(G_I)$.*
2. *Compute the weights $w_i = \varphi_i(G_I) - \lfloor\varphi\rfloor_i(G_I)$ for all $i \in N$.*
3. *Sort the weights in descending order such that if multiple players have the same weight, then their ordering is uniformly random.*
4. *Each player receives his Floor Shapley value. Additionally, the top k players, where $k = v_I(N) - \sum_{i \in N} \lfloor\varphi\rfloor_i(v_I) = w(N)$, receive one extra unit.*

Definition 8 *For an integer cooperative game $G_I \in \mathcal{G}_I^n$, the* Probabilistic Efficient Floor Shapley value $\varphi^{\mathbb{E}}(v)$ *is defined as follows:*

1. *Compute the Floor Shapley value $\lfloor\varphi\rfloor(G_I)$ and the Shapley value $\varphi(G_I)$.*
2. *Compute the remainders $\tilde{p}_i = \varphi_i(G_I) - \lfloor\varphi\rfloor_i(G_I)$ for all $i \in N$.*
3. *Compute the probabilities $p_i = \frac{\tilde{p}_i}{\sum_{j\in N}\tilde{p}_j}$ for all $i \in N$.*
4. *Each player receives his Floor Shapley value and additionally, each unit of the remainder with probability p_i, i.e., each unit of $\tilde{p}(N) = \sum_{j\in N}\tilde{p}_j$ is given to player i with probability p_i.*

Theorem 9 *The Probabilistic Efficient Floor Shapley value $\varphi^{\mathbb{E}}$ satisfies the following properties for all integer games $(N, v_I), (N, w_I) \in \mathcal{G}_I^n$:*

1. *The expected value is the same as the Shapley value:*
$$\mathbb{E}[\varphi^{\mathbb{E}}(v_I)] = \varphi(v_I),$$
2. *Axiom of efficiency:*
$$\sum_{i\in N}\varphi_i^{\mathbb{E}}(v_I) = v_I(N),$$
3. *Axiom of expected symmetry:*
$$\forall i,j \in N(\forall S \subseteq N \setminus \{i,j\} : v_I(S\cup i) = v_I(S\cup j)) \Rightarrow \mathbb{E}[\varphi_i^{\mathbb{E}}(v_I)] = \mathbb{E}[\varphi_j^{\mathbb{E}}(v_I)],$$
4. *Axiom of null player:*
$$\forall i \in N(\forall S \subseteq N : v_I(S) = v_I(S\cup i)) \implies \varphi_i^{\mathbb{E}}(v_I) = 0,$$
5. *Axiom of expected additivity:*
$$\mathbb{E}[\varphi^{\mathbb{E}}(v_I + w_I)] = \mathbb{E}[\varphi^{\mathbb{E}}(v_I)] + \mathbb{E}[\varphi^{\mathbb{E}}(w_I)].$$

Definition 10 *Let $||\bullet||$ be a vector norm. The* Closest Lattice Shapley (CLS) *value of integer cooperative game $G_I \in \mathcal{G}_I^n$ is given by $\varphi^{\mathcal{W}}(G_I) = \min_{x\in\mathcal{W}_{\mathbb{Z}}(G_I)}||\varphi(G_I) - x||$.*

Proposition 11 *The CLS value:*

- *exists for all integer games $G_I \in \mathcal{G}_I^n$,*
- *is not unique in general,*
- *depends on the choice of the norm,*
- *is different from the Efficient Floor Shapley value.*

Definition 12 *For an integer cooperative game $G_I \in \mathcal{G}_I^n$, the* integer nucleolus $\eta_{\mathbb{Z}}(G_I)$ *is defined as*
$$\eta_{\mathbb{Z}}(G_I) = \{x \in \mathcal{I}_{\mathbb{Z}}(G_I) \mid \forall y \in \mathcal{I}_{\mathbb{Z}}(G_I) : \Theta_{\mathbb{Z}}(x) \preceq_{lex} \Theta_{\mathbb{Z}}(y)\}.$$

Theorem 13 *For an integer cooperative game $G_I \in \mathcal{G}_I^n$, it holds*
$$\mathcal{I}_{\mathbb{Z}}(G_I) \neq \emptyset \implies \eta_{\mathbb{Z}}(G_I) \neq \emptyset.$$

Theorem 14 *For an integer cooperative game $G_I \in \mathcal{G}_I^n$, it holds*

$$\eta_{\mathbb{Z}}(G_I) \neq \emptyset \iff \eta(G_I) \neq \emptyset.$$

Theorem 15 *For an integer cooperative game $G_I \in \mathcal{G}_I^n$ the nucleolus is not necessarily a single point solution concept.*

David Ryzák
david.ryzak99@gmail.com

Fair shares in discrete fair division
as part of series Cooperative game theory

Introduction

What if a group of people wants to fairly divide a set of indivisible resources or objects? Discrete fair division studies such questions and in such settings, standard fairness notions like the maximin share (MMS) or envy-freeness (EF) are often considered. However, these concepts do not always guarantee the existence of a feasible allocation, even when players have additive valuations so they might be too strong in some cases.

Moreover, if players are allowed to choose which fairness concept to apply, they may have conflicting preferences, leading to disagreements.

To address this issue, we introduce the concept of a fair share: a fairness notion that is both feasible and universally acceptable.

Definition 1 (Fair division instance) *Let $N = \{1, \dots, n\}$ be a set of n players and M a finite set of indivisible goods. Each player $i \in N$ has a valuation function $v_i : 2^M \to \mathbb{R}_{\geq 0}$, which is assumed to be* additive*, i.e., $v_i(S) = \sum_{g \in S} v_i(\{g\})$ for any $S \subseteq M$.*

Definition 2 (Feasible Fairness Concept) *A fairness concept $\mathcal{F}$ is said to be* feasible *for a class of valuation functions $\mathcal{C}$ if for every instance with:*

- *a set $N = \{1, \dots, n\}$ of agents,*
- *a set M of indivisible goods, and*
- *a valuation profile $(v_1, \dots, v_n)$,*

there exists an allocation $(A_1, \dots, A_n)$ such that the allocation satisfies the fairness concept $\mathcal{F}$ with respect to the valuations $(v_1, \dots, v_n)$.

Definition 3 (Proportional Share (PROP)) *An allocation $(A_1, \dots, A_n)$, where each $A_i \subseteq M$ and $A_i \cap A_j = \emptyset$ for $i \neq j$, satisfies* proportionality *if for every player $i \in N$,*

$$v_i(A_i) \geq \frac{v_i(M)}{n}.$$

Definition 4 (Envy-Freeness (EF)) *An allocation is* envy-free *if no player prefers another player's bundle to their own. That is, for every pair $i, j \in N$,*

$$v_i(A_i) \geq v_i(A_j).$$

Definition 5 (Maximin Share (MMS)) *The* maximin share *of player i is defined as the maximum value player i can guarantee themselves by partitioning the goods into n bundles and receiving the least valuable one (according to their own valuation). Formally,*

$$\mathrm{MMS}_i = \max_{\{P_1, \dots, P_n\} \in \Pi_n(M)} \min_{j=1,\dots,n} v_i(P_j),$$

where $\Pi_n(M)$ denotes the set of all n-partitions of M. An allocation satisfies the MMS condition if

$$v_i(A_i) \geq \mathrm{MMS}_i \quad \text{for all } i \in N.$$

Definition 6 (Share function) *A* share function *(or simply* share*) s for a class of valuations $\mathcal{C}$ is a function that outputs a real number $s(v,n)$ when given an additive valuation function $v \in \mathcal{C}$ over a set of items M, and the number of agents n. The function must satisfy the following properties:*

- ***Realizability:*** *For every v and every n, it holds that*
$$s(v,n) \leq \max_{S \subseteq M} v(S).$$
- ***Name Independence:*** *Renaming the items does not change the value of the share.*

Given a share function s, a bundle $S \subseteq M$ is acceptable *for an agent with valuation v if it satisfies:*
$$v(S) \geq s(v,n).$$

Definition 7 (Self-Maximizing Share) *A share function s is* self-maximizing *if for every set M of items and number of agents n, for every true additive valuation v and every reported additive valuation v, it holds that*
$$\hat{s}^v(v) = \hat{s}(v,n) \geq \hat{s}^v(v'),$$

Definition 8 (ρ-dominating share) *Let $\rho > 0$. A share function s' ρ-*dominates *(or simply* dominates *if $\rho = 1$) another share function s if*
$$s'(v,n) \geq \rho \cdot s(v,n)$$
*for every additive valuation v. We say that s' is ρ-*dominating *if it ρ-dominates every feasible share function s for additive valuations.*

Bibliography

[1] Moshe Babaioff, Uriel Feige. *Fair Shares: Feasibility, Domination and Incentives* EC '22: Proceedings of the 23rd ACM Conference on Economics and Computation

[2] Moshe Babaioff, Uriel Feige. *Share-Based Fairness for Arbitrary Entitlements* `https://arxiv.org/abs/2405.14575`

[3] David Kurokawa, Ariel D. Procaccia, and Junxing Wang. *Fair enough: Guaranteeing approximate maximin shares* J. ACM, 65(2):8:1–8:27, 2018.

David Sychrovský
sychrovsky@kam.mff.cuni.cz

Presented paper by Sychrovský et al.

Approximating Nash Equilibria in General-Sum Games via Meta-Learning *as part of series* Cooperative game theory

(https://arxiv.org/abs/2504.18868)

Introduction

Nash equilibrium is perhaps the best-known solution concept in game theory. Such a solution assigns a strategy to each player which offers no incentive to unilaterally deviate. While a Nash equilibrium is guaranteed to always exist, the problem of finding one in general-sum games is PPAD-complete, generally considered intractable. Regret minimization is an efficient framework for approximating Nash equilibria in two-player zero-sum games. However, in general-sum games, such algorithms are only guaranteed to converge to a coarse-correlated equilibrium (CCE), a solution concept where players can correlate their strategies. In this work, we use meta-learning to minimize the correlations in strategies produced by a regret minimizer. This encourages the regret minimizer to find strategies that are closer to a Nash equilibrium. The meta-learned regret minimizer is still guaranteed to converge to a CCE, but we give a bound on the distance to Nash equilibrium in terms of our meta-loss. We evaluate our approach in general-sum imperfect information games. Our algorithms provide significantly better approximations of Nash equilibria than state-of-the-art regret minimization techniques.

Bibliography

[1] *David Sychrovský and Christopher Solinas and Revan MacQueen and Kevin Wang and James R. Wright and Nathan R. Sturtevant and Michael Bowling*, Approximating Nash Equilibria in General-Sum Games via Meta-Learning, In: *arXiv preprint*, 2025, URL: https://arxiv.org/abs/2504.18868

Anna Havelková

anna.havelka@seznam.cz

Who Can Join the Team? Hierarchy in Restricted Cooperation Games *as part of series* Cooperative game theory

Introduction

Cooperation in groups is rarely unconstrained. In organizations, projects, and decision-making bodies, certain players may only act once others have approved or participated. Such settings require generalizing traditional cooperative games to account for restrictions in coalition formation.

This talk explores how to assign fair values to players in cooperative games where not all coalitions are feasible. Specifically, we focus on games with *permission structures* and *antimatroids*.

From Classical to Structured Games

In a *transferable utility (TU)* cooperative game (N, v), any subset of players $S \subseteq N$ is allowed to form a coalition, and the function v assigns a value to each such coalition.

In structured settings, not all coalitions are feasible. The feasibility of S may depend on whether certain players precede others (e.g., hierarchically). To formalize this, we define the set of *feasibility structure* $\Phi \subseteq 2^N$.

Definition 1 (Feasibility Structure) *Let N be a finite set. A subset $\Phi \subseteq 2^N$ is a* feasibility structure *if it satisfies:*

- $\emptyset \in \Phi$,
- *For each $S \in \Phi$, if $T \subseteq S$, then $T \in \Phi$ (downward closed).*

Antimatroid Feasibility

Definition 2 (Antimatroid) *A collection $\Phi \subseteq 2^N$ is an* antimatroid *if:*

- $\emptyset \in \Phi$,
- $S, T \in \Phi \Rightarrow S \cup T \in \Phi$,
- $S \in \Phi, S \neq \emptyset \Rightarrow \exists i \in S : S \setminus \{i\} \in \Phi$,
- $\forall i \in N, \exists S \in \Phi$ *such that* $i \in S$.

These properties reflect settings where coalitions are built up incrementally with non-circular dependencies. Antimatroids are especially useful in contexts where there are multiple valid paths to achieving authorization.

Restricted Games and the Shapley Value

Let $v : \Phi \to \mathbb{R}$ be a characteristic function defined only on feasible coalitions. We define a *restricted game* v^r for every $S \subseteq N$ by:

$$v^r(S) = v(S^*),$$

where

$$S^*$$

is the largest feasible subset of S.

Definition 3 (Permission-Based Shapley Value) *The* Shapley value with restricted feasibility *is given by:*

$$\varphi_i^{\Phi}(v) = \varphi_i(v^r),$$

where φ_i is the classical Shapley value applied to the restricted game.

Axioms and Uniqueness

The value φ^{Φ} satisfies:

- **Efficiency:** $\sum_{i \in N} \varphi_i^{\Phi}(v) = v^r(N)$,
- **Dummy Player:** If $v(S \cup \{i\}) = v(S)$ whenever $S \cup \{i\} \in \Phi$, then $\varphi_i^{\Phi}(v) = 0$,
- **Fairness:** Structure-respecting versions of symmetry and monotonicity.

Theorem 4 (Uniqueness) *The value φ^{Φ} is the unique function satisfying linearity, efficiency, dummy player, and structure monotonicity on antimatroid feasibility domains.*

Applications

- **Strategic Project Teams:** Where some members must prepare groundwork before others join.
- **Knowledge Hierarchies:** Access granted only after prerequisite understanding.
- **Voting Blocks:** Where influence depends on layered approval structures.

Bibliography

[1] Q. Vandendriessche et al. Games with a permission structure – A survey on generalizations and applications. *Annals of Operations Research*, 2017.

Standalone Talks

Adam Beneš
ad.benes@gmail.com

Presented paper by Oded Regev and Noah Stephens-Davidowitz

A simple proof of a reverse Minkowski theorem for integral lattices

(https://arxiv.org/pdf/2306.03697)

Introduction

A lattice $\mathcal{L} \subset \mathbb{R}^n$ is the set of integer linear combinations of some linearly independent basis vectors. For any integral lattice $\mathcal{L} \subset \mathbb{R}^n$ (that is, a lattice $\mathcal{L}$ such that the inner product $\langle y_1, y_2 \rangle$ is an integer for all $y_1, y_2 \in \mathcal{L}$) and positive $k \in \mathbb{Z}$ it holds, that $N_{=k}(\mathcal{L}) := |\{y \in \mathcal{L} : \|y\|^2 = k\}| \leq 2\binom{n+2k-2}{2k-1}$. This is an opposite bound to Minkowski theorem. It is not hard to see that $N_{\leq 1}(\mathcal{L}) \leq 2n + 1$. However it is not true that $N_{\leq k}(\mathcal{L}) \leq N_{\leq k}(\mathbb{Z}^n)$.

Definitions

Definition 1 *Integral lattice $L \subset \mathbb{R}^d$ is a lattice that has inner product $\langle y_1, y_2 \rangle$ integer for all $y_1, y_2 \in \mathcal{L}$.*

Definition 2 $N_{\leq k}(\mathcal{L}) : |\{y \in \mathcal{L} : \|y\|^2 \leq k\}|$

Definition 3 $N_{=k}(\mathcal{L}) : |\{y \in \mathcal{L} : \|y\|^2 = k\}|$

Theorems

Observation 4 *Given lattice $\mathcal{L} \subset \mathbb{R}^n$ it holds that,*

$$N_{\leq 1}(\mathcal{L}) \leq N_{\leq 1}(\mathbb{Z}^n) = 2n + 1$$

Theorem 5 *Given integral lattice $\mathcal{L} \subset \mathbb{R}^n$,*

$$N_{\leq k}(\mathcal{L}) \leq 2\binom{n + 2k - 1}{2k - 1} - 1$$

Theorem 6 *Given integral lattice $\mathcal{L} \subset \mathbb{R}^n$,*

$$N_{=k}(\mathcal{L}) \leq 2\binom{n + 2k - 2}{2k - 1}$$

Theorem 7 *Given integral lattice $\mathcal{L} \subset \mathbb{R}^n$,*

$$N_{\leq k}(\mathcal{L}) \leq 2^k(k-1)! \cdot n^k + o(n^k)$$

Theorem 8 *Given integral lattice $\mathcal{L} \subset \mathbb{R}^n$ and constant $C > 0$,*

$$\sum_{y \in \mathcal{L}} \exp(-2\log(2n)\|y\|^2) \leq 1 + \frac{C}{n}$$

Theorem 9 *Given integral lattice* $\mathcal{L} \subset \mathbb{R}^n$*,*

$$N_{\leq 2}(\mathcal{L}) \leq f(n) + 1$$

$$\textit{Where } f(n) = \begin{cases} 126 & \textit{if } n = 7 \\ 240 + 2(n-8)^2 & \textit{if } 8 \leq n \leq 11 \\ 2n^2 & \textit{otherwise} \end{cases}$$

Theorem 10 *Given integral lattice* $\mathcal{L} \subset \mathbb{R}^n$ *with* $\det(\mathcal{L}) \leq 1$ *and* $k > 0$*,*

$$N_{\leq k}(\mathcal{L}) \geq 2^{-n} \cdot \textit{vol}(\sqrt{k}B_2^n)$$

Theorem 11 *Given integral lattice* $\mathcal{L} \subset \mathbb{R}^n$ *with* $\det(\mathcal{L}') \geq 1$ *for all sublattices* $\mathcal{L}' \subseteq \mathcal{L}$ *and* $k > 0$ *and constant* $C > 0$*,*

$$N_{\leq k}(\mathcal{L}) \leq 2\exp(C \cdot k \cdot \log^2(2n))$$

Petr Chmel
chmel@iuuk.mff.cuni.cz

Catalytic Space: Using Your Family Pictures for Computation

Motivation

Definition 1 (STCONN)
Input: *Directed graph G and two of its vertices s, t.*
Output: *Decide whether there is a path from s to t in G.*

Theorem 2 (Savitch, [1]) *STCONN can be solved in space $\mathcal{O}(\log^2 n)$ (and the algorithm uses $n^{\Theta(\log n)}$ time). As a corollary,* $\mathrm{NSPACE}(s(n)) \subseteq \mathrm{DSPACE}(s(n)^2)$.

Catalytic computation

Definition 3 (Catalytic computation, CSPACE, [3]) *Let $s, w : \mathbb{N} \to \mathbb{N}$ be non-decreasing functions. A language L is decided by a catalytic Turing machine M in space $s(n)$ using catalytic space $w(n)$ if on every input x of length n and arbitrary string a of length $w(n)$ written on the auxiliary tape the machine halts with a on its auxiliary tape, during its computation M uses (accesses) at most $s(n)$ tape cells on its work tape and $w(n)$ cells initially containing a on its auxiliary tape, and M correctly outputs whether $x \in L$.*

We define $\mathrm{CSPACE}(s(n), w(n))$ *to be the set of all languages decidable by a catalytic machine in space $s(n)$ with catalytic space $w(n)$. As a shortcut, we define* $\mathrm{CSPACE}(s(n)) = \mathrm{CSPACE}(s(n), 2^{\mathcal{O}(s(n))})$.

We define catalytic log-space to be the class $\mathrm{CL} = \mathrm{CSPACE}(\mathcal{O}(\log(n)))$.

Theorem 4 (Compress-or-random, [5], attributed to Bruno Loff) $\mathrm{BPL} \subseteq \mathrm{CL}$*, where BPL is the class of all languages recognizable in logspace using randomized Turing machines with probability of correctly answering at least* $2/3$.

In fact, the previous theorem can be made even stronger by an argument of Pyne [7]: BPL can be decided by catalytic Turing machines in time n using space $\log(n)$ and catalytic space $\log^2(n)$.

Theorem 5 (Register program arguments, [3]) $\mathrm{TC}^1 \subseteq \mathrm{CL}$*, where* TC^1 *is a class of log-depth dircuits with unbounded fan-in $g_{=\ell}$ gates for every ℓ, which output 1 iff the number of 1-inputs is exactly ℓ.*

Theorem 6 (Lower bound via average catalytic tape, [3]) $\mathrm{CL} \subseteq \mathrm{ZPP}$*, where ZPP is the class of probabilistic Turing machines running in polynomial time in expectation and* never erring.

Influence

Definition 7 (Tree evaluation problem, [2])
Input: *A complete d-ary tree of height h such that each leaf is labeled with a b-bit string and each internal node v is labeled with a function $f_v : \{0,1\}^{d \cdot b} \to \{0,1\}^b$.*
Output: *The value of the function at the root, where in each node, we compute its value using its function and the values of its children.*

Theorem 8 (TEP in almost log-space, [6]) Tree Evaluation *can be computed in $\mathcal{O}(\log n \cdot \log\log n)$ space (in terms of d, h, b, it is $\mathcal{O}(d \cdot b + h \cdot \log(d \cdot b))$).*

Theorem 9 (Simulating time in square-root space, [8]) *For multi-tape Turing machines and every function $t(n) \geq n$,* $\mathrm{TIME}[t(n)] \subseteq \mathrm{SPACE}[\sqrt{t(n) \log t(n)}]$.

Bibliography

[1] Walter J. Savitch. *Relationships between nondeterministic and deterministic tape complexities.*
Journal of Computer and System Sciences, 4(2), 1970.

[2] Stephen Cook, Pierre McKenzie, Dustin Wehr, Mark Braverman, Rahul Santhanam. *Pebbles and branching programs for tree evaluation.*
ACM Transactions on Computation Theory (TOCT), 3(2), 2012.

[3] Harry Buhrman, Richard Cleve, Michal Kouckỳ, Bruno Loff, Florian Speelman. *Computing with a full memory: catalytic space.*
In Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing, 2014.

[4] Michal Kouckỳ. *Catalytic computation.*
Bulletin of EATCS, 118(1), 2016.

[5] Ian Mertz. *Reusing space: Techniques and open problems.*
Bulletin of EATCS, 141(3), 2023.

[6] James Cook, Ian Mertz. *Tree Evaluation Is in Space $O(\log n \cdot \log \log n)$.*
Proceedings of the 56th Annual ACM Symposium on Theory of Computing, 2024.

[7] Edward Pyne. *Derandomizing logspace with a small shared hard drive.*
In 39th Computational Complexity Conference (CCC 2024), 2024.

[8] Ryan R. Williams. *Simulating Time With Square-Root Space.*
`https://arxiv.org/abs/2502.17779`. To appear in Proceedings of the 57th Annual ACM Symposium on Theory of Computing, 2025.

Ondřej Chwiedziuk
ondrachwiedziuk@gmail.com

Chromatic homology of graphs

Abstract

The *chromatic symmetric function* of a graph, defined by Stanley, is a remarkable combinatorial invariant which refines the chromatic polynomial. Recently, Sazdanovic and Yip catogorified this invariant by defining a new homological theory, called the *chromatic symmetric homology* of a graph G. This construction is obtained by assigning a graded representation of the symmetric group to every subgraph of G with the same vertices as G. Ciliberti and Moci proved that if graph G is non-planar, then its chromatic symmetric homology in bidegree $(1,0)$ contains $\mathbb{Z}_2$-torsion. In this talk, it will be explained what the chromatic symmetrc homology is.

Definitions

We will denote C_i as a i-th chain group and H_i the i-th homology group.

Definition 1 (Spanning subgraph)
Let G be a simple graph. A spanning subgraph $H \leq G$ *is a graph such that $V(H) = V(G)$ and $E(H) \subset E(G)$. A size of H, denozed as $|H|$, is a number of edges in H. Let $\mathbf{B}(G)$ be a poset of all spanning subgraphs of G ordered by reverse inclusion.*

Definition 2
Let $H \subset G$ be a spanning subgraph with connected components $B_1, \ldots, B_r$ with number of vertices in each component $b_1, \ldots, b_r$ respectively. Then the module associated to it in q-degree zero is the permutation module:

$$M_H = Ind_{\mathbf{B}_1 \times \cdots \times \mathbf{B}_r}^{\mathbf{S}_n}(S_{(b_1)} \otimes \cdots \otimes S_{(b_r)}),$$

where $\mathbf{S}_n$ is a permutation group on n elements and $S_{(i)}$ is a Specht module related to partition (i). We define a chain module $C_i(G)$ as a direct sum of all M_H with i edges.

Barbora Dohnalová

bdohnalova@kam.mff.cuni.cz

Presented paper by Noga Alon, Olivier Bousquet, Kasper Green Larsen, Shay Moran & Shlomo Moran

Diagonalization Games

(https://doi.org/10.1080/00029890.2024.2393992)

Introduction

The Cantor-Kronecker game is played as follows: Kronecker has a private list of binary vectors $v_1, \ldots, v_m$, each of length n. Cantor's goal is to produce a new vector which is not included among $v_1, \ldots, v_m$.

Cantor can only use queries in the form "What is the i-th bit of vector j?" and Kronecker must answer truthfully. Cantor's goal is to minimize the number of queries needed to produce the new vector. (Or to find out that such vector does not exist.)

There are two versions of the game:

- *Oblivious*: Cantor has to decide all the queries in advance.
- *Adaptive*: Cantor can decide the next query based on Kronecker's previous answers.

Puzzle

When $m = n$, Cantor's strategy is simple: use diagonalization. His vector's i-th bit will be 1 minus the i-th bit of the i-th Kronecker's vector, so it will difer from each vector in at least one bit.

Keep this idea in mind, and consider the case $m = n + 1$. How can Cantor find the vector in the adaptive version using $n + 2$ queries? What about the oblivious version and $n + 4$ queries?

Results

Below is the number of queries Cantor needs to produce a new vector (or to decide that there is none):

Version	**# of Queries**
$m \leq n$	m
$n < m < 2^n$, adaptive	$2m - n$
$n < m < 2^n$, oblivious	$m\left(\log\lceil\frac{m}{n}\rceil \pm O\left(\log\log\lceil\frac{m}{n}\rceil\right)\right)$
$2^n \leq m$	$m \cdot n$

Adam Džavoronok
adam.dzavoronok@gmail.com

Presented paper by N. Alon, M. Bucić,L. Gishboliner

The spanning tree spectrum

(https://arxiv.org/abs/2503.23648)

Introduction

Given a graph G we denote $\tau(G)$ the number of spanning trees of G. In this talk we will be interested how large is the range, when evaluated on n vertex graph. It is well known, that $\tau(G) \leq n^{n-2}$ for every G but if G is planar that turns out to significantly bound $\tau(G) \leq 5.29^n$ [1]. Therefore it is natural to consider this problem over certain families of graphs. To introduce some notation let $\mathcal{G}$ be a family of graphs, for which will denote $T(\mathcal{G}) := \{\tau(G) \mid G \in \mathcal{G}\}$. We will be particularly interested in size of $T(\mathcal{G})$ for $\mathcal{G}_n^{\text{all}}$ and $\mathcal{G}_n^{\text{planar}}$ as families of all graphs and all planar graphs on n vertices respectively. Our main theorem is the following lower bound.

Theorem 1 *$|T(\mathcal{G}_n^{all})| \geq |T(\mathcal{G}_n^{planar})| \geq 1.49^n$, for large enough n.*

Proof overview

Definition 2 *We say that a vector $\begin{bmatrix} t \\ u \end{bmatrix}$ is* n-planar-feasible *if there exists a planar graph G on up to n vertices and an edge $e \in E(G)$ such that $t = \tau(G \setminus e)$ and $u = \tau(G - e)$, where $G \setminus e$ and $G - e$ are graphs obtained from G by contracting / deleting e.*

Lemma 3 *If there are at least N distinct n-planar-feasible vectors, then $|T(\mathcal{G}_n^{planar})| \geq \sqrt{N}$.*

Lemma 4 *If* n-planar-feasible *vector $\begin{bmatrix} t \\ u \end{bmatrix}$ is multiplied by one of the following matrices, we obtain* $n+1$-planar-feasible *vector*

$$A := \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \qquad B := \begin{pmatrix} 2 & 0 \\ 1 & 2 \end{pmatrix} \qquad C := \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \qquad D := \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

Further results

Theorem 5 *For any fixed integer $k \geq 3$ there are at least $2^{\Omega(n)}$ different values of $\tau(G)$ among k-regular connected graphs G on n vertices, provided that kn is even.*

Theorem 6 *Let N be sufficiently large. For any coprime a, b, there exists $t \leq O(\log N)$ and $i_1, \ldots, i_t \in \{1, 2\}$ such that*

$$\begin{bmatrix} a \\ b \end{bmatrix} \equiv A^{i_1} D A^{i_2} D \ldots A^{i_t} D \begin{bmatrix} 1 \\ 0 \end{bmatrix} \bmod N.$$

In particular, there exists a $O(\log N)$-planar-feasible vector $\begin{bmatrix} x \\ y \end{bmatrix}$ such that $x \equiv a \bmod N$ and $y \equiv b \bmod N$.

Definition 7 *We say that an n-vertex graph is a c-*expander *if any subset U on up to $n/2$ vertices has its external neighborhood $N(U)$ of size at least $c|U|$.*

Definition 8 *Given a group G and a subset of its elements S, the Cayley digraph of G generated by S, denoted $\mathrm{Cay}(G, S)$, is the graph whose vertex-set is G and where an edge from a to b exists if and only if $a^{-1}b \in S$.*

Proposition 9 *(Selberg's Theorem)[2] There exists $c > 0$ so that for any large enough N, the Cayley graph $\mathrm{Cay}(\mathrm{SL}_2(\mathbb{Z}_N), S \cup S^{-1})$, with $S = \{AD, A^2D\}$, is a c-expander, moreover $\mathrm{Cay}(G, S)$ is a $\frac{c}{2|S|}$-out-expander.*

Bibliography

[1] K.Buchin and A. Schulz. On the number of spanning trees a planar graph can have, Algorithms–ESA 2010: 18th Annual European Symposium, Liverpool, UK, September 6-8, 2010. Proceedings, Part I 18, Springer, 2010, pp. 110– 121.

[2] A. Selberg, On the estimation of Fourier coefficients of modular forms, Proceedings of Symposia in Pure Mathematics, American Mathematical Society, 1965, pp. 1–15.

Filip Filipkowski

filip.filipkowski.stud@pw.edu.pl

Presented paper by Bernhard Haeupler, Richard Hladík, Václav Rozhoň, Robert E. Tarjan and Jakub Tětek

Bidirectional Dijkstra's Algorithm is Instance-Optimal

(https://arxiv.org/pdf/2410.14638)

Introduction

The bidirectional Dijkstra Algorithm works by running two simultaneous Dijkstra searches: from the source s forward and from the target t backward, alternating edge relaxations and stopping as soon as the sum of their current best distances meets or exceeds the best s–t path found so far. We can prove that this simple variant is *instance-optimal* in the adjacency-list model:

- **Weighted graphs:** No correct algorithm can explore asymptotically fewer edges on any input than our bidirectional implementation, up to a constant factor.
- **Unweighted graphs:** Bidirectional BFS is optimal up to a factor of Δ (the maximum degree), and this Δ-factor is the best possible.

Theorems

Here you can find all the theorems that will be proved, for making it easier to follow the talk.

Theorem 1 *Let G be a directed weighted graph with strictly positive edge weights, and let $s, t \in V(G)$. Suppose the only allowed query is to take a seen vertex u and ask for its next out-neighbor (in adversarial order) and the weight of that edge. If we run Dijkstra's algorithm from s and abort as soon as we close a vertex v with $\hat{d}(s,v) = \hat{d}(s,t)$, then:*

1. *It correctly computes the s-t distance.*
2. *No correct deterministic algorithm can perform fewer queries on G.*

Theorem 2 *In the adjacency-list query model on weighted multigraphs (directed or undirected) with positive weights, the following bidirectional algorithm is instance-optimal under query complexity, up to a constant factor:*

- *Alternate one edge relaxation in a forward Dijkstra from s and one in a backward Dijkstra from t.*
- *Maintain the best meeting-edge length μ and stop as soon as the current frontiers satisfy $\hat{d}(s,u_s) + \hat{d}(u_t,t) \geq \mu$.*

That is, on every input (G,s,t), no correct algorithm can in expectation query fewer edges by more than a constant factor.

Theorem 3 *In the adjacency-list model on unweighted graphs, the bidirectional BFS algorithm described above is instance-optimal up to a factor of $O(\Delta)$, where Δ is the maximum degree of G. In particular, its query (and time) complexity on any instance is within an $O(\Delta)$-factor of any correct algorithm.*

Theorem 4 *Let W be any set of allowed positive edge-weights with $\nu = \min W > 0$, and restrict inputs to graphs of maximum degree Δ. Then no algorithm can be instance-optimal (under query or time complexity) up to a factor of $o(\Delta)$. Equivalently, the $O(\Delta)$ gap in Theorem 3 is asymptotically tight.*

Antonina Frąckowiak
antoninamariafrackowiak@gmail.com
Presented paper by Ivor van der Hoog, Eva Rotenberg, Daniel Rutschmann
Simpler Optimal Sorting from a Directed Acyclic Graph
(https://epubs.siam.org/doi/10.1137/1.9781611978315.26)

Introduction

Consider the following problem: we are given a ground set X, a partial order P over X, and a comparison oracle O_L that defines a linear extension L of P. A query to O_L takes two distinct elements $x, x' \in X$ and returns whether $x <_L x'$. The task is to reconstruct the full linear order L using the minimum number of oracle queries.

This problem, proposed by Fredman in 1976, generalizes the standard sorting problem. Fredman also proved that any algorithm must make at least $O(\log e(P))$ queries in the worst case, where $e(P)$ is the number of linear extensions of P.

Over the years, various algorithms have been developed to tackle this. The authors of the presented paper propose a new algorithm that uses linear space, runs in $O(n + m + \log e(P_G))$ time, and requires only $O(\log e(P_G))$ queries to the oracle.

As input, the algorithm takes a directed acyclic graph (DAG) with vertex set $X = (x_1, \ldots, x_n)$. This DAG defines a partial order P_G where $x_i \prec x_j$ if and only if there exists a directed path from x_i to x_j in G.

Algorithm

Algorithm 1 Sort(directed acyclic graph G over a ground set X, Oracle O_L) — time

1: $\pi \leftarrow$ a longest directed path in G ▷ $O(n + m)$
2: $T_\pi \leftarrow$ a level-linked (2-4)-tree over π ▷ $O(n)$
3: $H \leftarrow G - \pi$ ▷ $O(n + m)$
4: Compute for each vertex in H its in-degree in H ▷ $O(n + m)$
5: $S \leftarrow$ sources in H ▷ $O(n)$
6: **while** $S \neq \emptyset$ **do**
7: Remove an arbitrary vertex x_i from S ▷ $O(1)$
8: $p_i \leftarrow$ a dummy vertex, which is prepended before the head of π ▷ $O(1)$
9: **for all** in-neighbors u of x_i in G **do**
10: $p_i \leftarrow$ Compare(p_i, u) ▷ $O(1)$
11: Remove x_i from H and add any new sources in H to S ▷ $O(d^*(x_i))$
12: $q_i \leftarrow$ Search(x_i, p_i, O_L) ▷ $O(1 + \log d_i)$
13: FingerInsert(q_i, x_i) ▷ $O(1)$ amortised
14: **end while**
15: **return** the leaves of T_π in order ▷ $O(n)$

We use a data structure to store the path π that supports the following operations:

FingerInsert(q_i, x_i). Given vertices $x_i \notin \pi$ and $q_i \in \pi$, insert x_i into π succeeding q_i in $O(1)$ time.

SEARCH(x_i, p_i, O_L). Given a vertex $x_i \notin \pi$ and a vertex $p_i \in \pi$ with $p_i <_L x_i$. Return the farthest vertex q_i along π where $q_i <_L x_i$. Let d_i denote the number of vertices on the subpath from p_i to q_i along π. We want to use $O(1 + \log d_i)$ time and queries to O_L.

COMPARE(p, q). Given $p, q \in \pi$, return q if it succeeds p in π in $O(1)$ time (return p otherwise).

The above algorithm runs in $O(n+m+k+\sum_{i=1}^{k} \log d_i)$ time and uses $O(k+\sum_{i=1}^{k} \log d_i)$ queries, where d_i is the length of the subpath from p_i to q_i in π ($d_i = 1$ if $p_i = q_i$).

Now we use the following lemmas to show that the number of queries to the oracle is optimal that is $\sum_{i=1}^{k}(1 + \log d_i) = k + \sum_{i=1}^{k} \log d_i \in O(\log e(P_G))$.

Lemma 1 *Let G be a directed acyclic graph, P_G be its induced partial order and π be a longest directed path in G. If π has $n - k$ vertices then $\log e(P_G) \geq k$.*

Definition 2 *Let π^* be the directed path that Algorithm 1 outputs. For any $i \in [n]$ denote by $\pi^*(x_i)$ the index of x_i in π^*. We create an embedding E of X by placing x_i at position $\pi^*(x_i)$.*

We create as set $\mathcal{R}$ of n open intervals $R_i = (a_i, b_i) \subseteq [0, n]$ as follows:

- *If $i > k$ then $(a_i, b_i) := (\pi^*(x_i) - 1, \pi^*(x_i))$.*
- *Else, $(a_i, b_i) := (\pi^*(p_i), \pi^*(x_i))$.*

Lemma 3 *Given distinct $x_i, x_j \in X$, if there exists a directed path from x_i to x_j in G then the intervals $R_i = (a_i, b_i)$ and $R_j = (a_j, b_j)$ are disjoint with $b_i \leq a_j$.*

Lemma 4 *Let $\mathcal{R} = (R_1, \ldots R_n)$ be a set of n open intervals in $[0, n]$ and let each interval have at least unit size. Let $P_{\mathcal{R}}$ be its induced partial order. Then:*

$$\sum_{i=1}^{n} \log(|R_i|) \in O(\log e(P_{\mathcal{R}})).$$

With these lemmas we can prove the main theorem:

Theorem 5 *Given a directed acyclic graph G over X, inducing a partial order P_G, and an oracle O_L whose queries specify a linear order L that extends P_G, there exists an algorithm that uses linear space, $O(n + m + \log e(P_G))$ time and $O(\log e(P_G))$ oracle queries to output the sorted order of X.*

Vojtěch Gaďurek

dlaza@kam.mff.cuni.cz

Presented paper by William Kuszmaul

A Simple and Combinatorial Approach to Proving Chernoff Bounds and Their Generalizations

(https://epubs.siam.org/doi/abs/10.1137/1.9781611978315.6)

Introduction

Chernoff bound and its variants are quite useful, especially, when using probabilistic method or analyzing randomized algorithms. However its most common statement and analysis may be hard to understand and see intuition, why it should work. The results are worse but only in constant.

Goals

Definition 1 *Be $X = \sum_{i\in[n]} X_i$, where X_i are bern. d. variables with $p \leq 1/2$ and mean $\mu = pn$.*

Theorem 2 (Chernoff Bound) *For $k \in [\sqrt{n}]$:*

small-deviation

$$P(X \geq k\sqrt{n}) \leq 2^{-\Theta(k^2)}$$

large-deviation

$$P(X \geq k\sqrt{n}) \leq \Theta(k)^{-\Theta(k)}$$

Preliminiaries

Generally, it may be beneficial to have some basic understanding of probability.

Theorem 3 (Chebychev) *Be X random variable with non-zero variance σ^2 and mean 0, then:*

$$P(X > k\sigma) \leq \frac{1}{k^2}$$

for $k > 0$

Weaker bounds

Lemma 4 (Extended Chebychev) *For $k \geq 1$:*

$$(\max_j \sum_{i\in[j]} X_i > k\sqrt{n}) \leq \frac{2}{k^2}$$

Lemma 5 (Poor man's Chernoff Bound) *For all $k \geq 1$:*

$$P(X \geq k\sqrt{n}) \leq 2^{-\Omega(k)}$$

Usefull Lemmas

Lemma 6 (Sum of Geometric Random Variables) *Let Y_i for $i \in [n]$ be independent real-valued random variables and let $p \in (0, 1)$. Suppose each Y_i satisfies for all non-negative integers j:*

$$P(Y_i \leq j) \geq p^j.$$

Then the sum $Y = \sum_i Y_i$ satisfies:

$$P[Y \leq 2n] \geq 4n$$

Goals if time left

Theorem 7 (Adaptive Version of Bennett's Inequality) *Let $n \in \mathbb{N}$ and $v \in \mathbb{R}^+$. Suppose that Alice selects $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$ where each $\mathcal{D}_i$ is a probability distribution over $[-\infty, 1]$ with mean 0 and with some variance v_i. Alice selects $\mathcal{D}_1, \mathcal{D}_2, \ldots$ one at a time, and once a given $\mathcal{D}_i$ is selected, a random variable X_i is drawn from the distribution $\mathcal{D}_i$. Alice gets to select the $\mathcal{D}_i$s (and thus also the v_is) adaptively, basing $\mathcal{D}_i$ on the outcomes of $X_1, \ldots, X_{i-1}$. The only constraint on Alice is that $\sum_{i=1}^n v_i \leq v$. Define $X = \sum_{i=1}^n X_i$. Then, for $k \in [1, \sqrt{v}]$, we have that*

- *(the small-deviation case)*
$$\Pr[X \geq k\sqrt{v}] \leq 2^{-\Omega(k^2)}.$$
- *And for $r \geq 1$, we have*
(the large-deviation case)
$$\Pr[X \geq rv] \leq O(1/r)^{\Omega(rv)}.$$

Corollary 8 (Chernoff Bound for Non-Identical Real-Valued Coin Flips) *Let $X_1, \ldots, X_n \in [0, 1]$ be independent random variables with means $p_1, \ldots, p_n$. Let $\mu = \sum_i p_i$ and let $X = \sum X_i$. Then, for any integer $k \leq \sqrt{\mu}$,*

$$\Pr[X \geq \mu + k\sqrt{\mu}] \leq 2^{-\Omega(k^2)}. \tag{5.16}$$

And for any $r \geq 1$,

$$\Pr[X \geq \mu + r\mu] \leq O(1/r)^{\Omega(r\mu)}.$$

Proof. Each X_i has variance $\mathbb{E}[X^2] - p_i^2 \leq \mathbb{E}[X] \leq p_i$. So the result follows from Theorem 5.1. □

Karolína Hylasová
khylas@kma.zcu.cz
Presented paper by Olha Silina

Covering the edges of a graph with a perfect matchings

(https://arxiv.org/html/2309.10224v2)

Introduction

Let $G = (V, E)$ be a connected nontrivial graph. A set of edges $M \subseteq E$ is a *perfect matching* if every vertex is incident with exactly one edge of M. A graph is *matching covered* if every edge belongs to a perfect matching. A *cut* is the set of edges with exactly one point in S for some $S \subseteq V$. A cut is *odd* if $|S|$ is odd. We say that a graph is an *r-graph* if every vertex has degree r and all odd cuts have size at least r. Seymour showed that an r-graph is always matching covered [2].

One of the central results in this area is due to Lovász, who got a characterization for the lattice $\mathcal{L} = \{\sum \alpha_i A_i : \alpha_i \in \mathbb{Z}\}$ generated by the incidence vectors of perfect matchings A_i of a matching covered graph G using dual lattice theory [1]. From this characterization it follows that for an r-graph, the vector $2 \cdot \mathbb{1}$ belongs to $\mathcal{L}$. Hence, the vector of all-ones $\mathbb{1}$ can be obtained as a half-integral combination of the perfect matchings of G. In this paper, the author strengten this result as follows.

Given an r-graph G with n vertices and m edges, we want to study the set of solutions to $Ax = \mathbb{1}$, where A is a $\{0, 1\}$ incidence matrix with rows corresponding to the edges of G and columns corresponding to the perfect matchings in G. We consider graphs with no loops, but parallel edges are allowed, and we refer to a vector x satisfying $Ax = \mathbb{1}$ as a *partitioning* of the edges of G into perfect matchings.

Theorem 1 (Main theorem) *Let G be an r-graph with m edges and n vertices and let A be its edge to perfect matching incident matrix. Then there is a solution x^* to $Ax = \mathbb{1}$ satisfying the following conditions:*

1. *all non-integral entries of x^* are equal to $+1/2$;*
2. *x^* has at most $m - n + 1$ non-zero entries;*
3. *all non-zero entries of x^* correspond to linearly independent set of perfect matchings of G;*
4. *the total number of $+1/2$'s in x^* is at most $6p$, where p is the number of Petersen bricks of G.*

Tight cut decomposition

Definition 2 *A* tight cut *is an odd cut C such that every perfect matching intersects C in exactly one edge. A cut is* trivial *if one of its shores is a single vertex. Otherwise, a cut is* non-trivial.

All trivial cuts of G are tight, but in general there could be other non-trivial tight cuts.

Lemma 3 *In a matching-covered graph G with $Ax = \mathbb{1}$ feasible, all tight cuts have the same size. In particular, G is regular.*

Definition 4 *If a graph G has no non-trivial tight cuts and G is bipartite, then it is a* brace. *If G is not bipartite, then it is a* brace.

Existence of tight cuts motivates the so called *tight cut decomposition* process on an r-graph G:

1. Find a tight cut C of G, say its shores are U_1 and U_2.
2. Consider two *C-contractions* of G: let $G_1 = (V \setminus U_1 \cup \{u_1\}, E[U_2] \cup C)$ and $G_2 = (V \setminus U_2 \cup \{u_2\}, E[U_1] \cup C)$, i.e. we obtain G_i from G by mapping a shore of C into a single vertex u_i and deleting any loops.
3. Consider each G_1 and G_2 separately, go back to Step 1.

The process stops when all current graphs in the decomposition are either a brick or a brace. There are several useful properties of such decomposition.

Proposition 5 *Let $G = (V, E)$ be a matching covered graph and let C be a tight cut of G. Suppose $G_i = (V_i, E_i), i = 1, 2$ are two C-contractions of G. Then*

i. for any perfect matching M of G, the set $M \cap E_i$ is a perfect matching of G_i, for $i = 1, 2$;

ii. conversely, for any perfect matchings M_1, M_2 of G_1, G_2, respectively, such that $|M_1 \cap M_2| = \{e\} \in C$, the set $M_1 \cup M_2$ is a perfect matching of G;

iii. both G_1 and G_2 are matching covered;

iv. if G is an r-graph, then so are G_1 and G_2.

We can use the tight cut decomposition to construct solutions to $Ax = \mathbb{1}$ with the desired properties. Consider the graphs G_1 and G_2 obtained from G by contracting one shore of C and let E_1, E_2 be the edge sets of G_1, G_2, respectively. Furthermore, let A_1 and A_2 be the edge to perfect matching incidence matrices for G_1 and G_2, respectively. Given y, t statisfying $A_1 y = \mathbb{1}$ and $A_2 t = \mathbb{1}$, we will construct a vector x satisfying $Ax = \mathbb{1}$.

Fix an edge $e \in C$ and let $\{M_i^e\}_{i \in I^e}$ be the set of perfect matchings of G_1 using e, and let $\{N_j^e\}_{j \in J^e}$ be the set of perfect matchings of G_2 using e. All of these matchings do not use any other edges of C. Clearly, $M_i^e \cup N_j^e =: K_{ij}^e$ is a perfect matching in G for all indices i, j. Defining $x_{ij}^e := y_i^e t_j^e$ we get that x is a partitioning of G, i.e. $Ax = \mathbb{1}$, which is equivalent to

$$\sum_{e \in C} \sum_{i \in I^e, j \in J^e} x_{K_{ij}^e} K_{ij}^e = \mathbb{1}.$$

Main theorem

The paper gives us a different way of combining two solutions for tight cut contractions, which is a crucial ingredient of the main result. First step is to prove a weaker version of Theorem 3 which would give us an algorithm that we will later analyze in more details.

Lemma 6 *There is a solution x^* to $Ax = \mathbb{1}$ with all entries being either integral, or equal to $+1/2$.*

The algorithm of combining two solutions obtained from the proof of Lemma 6 preserves several important properties:

Theorem 7 *Suppose the method from Lemma 6 receives as input partitionings y, t of the two C-contractions of G and returns a partitioning x^* of G. Then, the following properties hold:*

i. the support size of x^* satisfies $|supp(x^*)| \leq |supp(y)| + |supp(t)|$;

ii. the largest entry of x^* satisfies $\|x^*\|_\infty \leq \max(\|y\|_\infty, \|t\|_\infty)$;

iii. if y, t have all entries in $\mathbb{Z} \cup \{+1/2\}$, then so does x^* and the total number of $+1/2$'s in x^* is at most number of $+1/2$'s in y and t combined;

iv. if both y and t only use linearly independent perfect matchings, then so does x^*.

To complete the proof of Lemma 6 we will talk separately about the base cases - braces, Petersen bricks, and non-Petersen bricks:

Lemma 8 *Let G be an r-graph and a brace, let A be its edge to perfect matching incidence matrix. Then there is a solution x^* to $Ax = \mathbb{1}$ satisfying the following conditions:*

1. *all entries of x^* are in $\{0, 1\}$;*
2. *$|supp(x^*)| = r$;*
3. *all perfect matchings used in x^* are disjoint, and thus linearly independent.*

Lemma 9 *Let G be an r-graph whose underlying simple graph is a Petersen graph. Let A be its edge to perfect matching incidence matrix. Then there is a solution x^* to $Ax = \mathbb{1}$ satisfying the following conditions:*

1. *all entries of x^* are in $\{0, 1/2, 1\}$;*
2. *$|supp(x^*)| \leq m - n + 1$ where m and n are the numbers of edges and vertices in G, respectively;*
3. *x^* has at most 6 entries equal to $+1/2$;*
4. *all perfect matchings used in x^* are linearly independent.*

Lemma 10 *Let G be an r-graph and a non-Petersen brick. Let A be its edge to perfect matching incidence matrix. Then there is a solution x^* to $Ax = \mathbb{1}$ satisfying the following conditions:*

1. *all entries of x^* are in $\mathbb{Z}$;*
2. *$|supp(x^*)| \leq m - n + 1$ where m and n are the numbers of edges and vertices in G, respectively;*
3. *all perfect matchings used in x^* are linearly independent;*
4. *$\|x^*\|_\infty \leq 2^{m-n+1}$.*

Bibliography

[1] L. Lovász. Matching structure and the matching lattice. *Journal of Combinatorial Theory*, Series B 43(2), 187-222, 1987.

[2] P.D. Seymour. On multi-colourings of cubic graphs, and conjectures of Fulkerson and Tutte. *Proceedings of the London Mathematical Society* s3-38(3), 423-600,1979.

Adam Jahoda

jahodaadam79@gmail.com

Presented paper by Sepehr Assadi, Helia Yazdanyar

Simple Sublinear Algorithms for $(\Delta + 1)$ Vertex Coloring via Assymmetric Pallete Sparsification

(https://epubs.siam.org/doi/pdf/10.1137/1.9781611978315.1)

Introduction

Given graph G with maximum degree Δ its easy to show that it can be colored using $\Delta + 1$ colors in linear time. In 2019 Assadi, Chen and Khanna proved the pallete sparsification theorem, which states that in every graph G with maximum degree Δ randomly sampling $\mathcal{O}(\log n)$ colors from $\{1, \ldots, \Delta + 1\}$ for every vertex independently and uniformly allows, with high probabilty, for finding $(\Delta + 1)$ vertex coloring in G while coloring each vertex using color from its sampled colors. This leads to sublinear algorithms for $(\Delta + 1)$ coloring.

This paper shows a weaker version of the theorem which bounds only average amount of sampled colors of each vertex. This leads to a much simpler proof and simpler, but similarly fast algorithms stemming from it.

Preliminaries

Definition 1 Hypergeometric *random variable with parameters N, K, M is a discrete random variable defined like this: we have N elements, K of them are "good" and we sample M elements uniformly at random without replacement and count the numbe of good samples.*

Fact 2 *Let X be a hypergeometric random variable with parameters N, K, M, and therefore expected value $\mathbb{E}[X] = M\frac{K}{N}$. Then for any $t \geq 0$ holds*

$$Pr(X \leq \mathbb{E}[X] - t) \leq e^{\left(-\frac{t^2}{2\mathbb{E}[X]}\right)}$$

Main result

Theorem 3 *Let $G = (V, E)$ be any $n-$vertex graph with maximum degree Δ. Sample a random permutation $\pi : V \to [n]$ uniformly and define*

$$l(v) := \min\left(\Delta + 1, \frac{40n \ln n}{\pi(v)}\right)$$

for every $v \in V$ as the size of the list of colors to be sampled for vertex v. Then

- *Deterministically $\sum_{v \in V} l(v) = \mathcal{O}(n \log^2(n))$, and for any fixed $u \neq v \in V$*
$$\mathbb{E}[l(v)] = \mathcal{O}(log^2(n)) \quad and \quad \mathbb{E}[l(u) \cdot l(v)] = \mathcal{O}(\log^4(n))$$
- *If we sample a list $L(v)$ of $l(v)$ colors from $\{0, \ldots, \Delta + 1\}$ uniformly and independently for each vertex, then with high probability (over the randomness of l and sampled lists) the greedy algorithm that iterates over vertices in the increasing order of $l(v)$ finds a proper list-coloring of G from the lists $\{L(v) | v \in V\}$*

The notation $\deg_\pi^<(v)$ means the amount of neighbors u of v such that $\pi(u) < \pi(v)$, therefore they are processed later in the greedy algorithm.

Lemma 4 *For all $v \in V$ with $\pi(v) > \frac{40n \ln n}{\Delta+1}$ and $\deg(v) \geq \frac{3\Delta}{4}$*

$$Pr\left(deg_\pi^<(v) < \frac{\Delta \cdot \pi(v)}{4n}\right) \leq n^{-2.5}$$

Algorithmic use

Definition 5 *Let $G = (V, E)$ be any graph with maximum degree Δ and $\mathcal{L} := \{L(v)|v \in V\}$ be a set of sampled colors for each vertex according to the previous distribution.* **Conflict graph** *$G_\mathcal{L} = (V, E_\mathcal{L})$ is a subgraph of G consisting of all edges $(u, v) \in E$ such that $L(u)$ and $L(v)$ intersect (have some color in common).*

Igor Januszkiewicz
igor.januszkiewicz.stud@pw.edu.pl
Presented paper by Alexander Golovnev, Tom Gur, Igor Shinkar

Derandomization of Cell Sampling

(https://arxiv.org/abs/2108.05970)

Introduction

Understanding the inherent limitations of data structures is a central problem in theoretical computer science. A particularly important goal is to determine the trade-offs between the space used by a data structure and the time it takes to answer queries. For *non-adaptive static data structures*, this trade-off has been extensively studied using the **cell sampling** technique introduced by Siegel [1], which yields the best known lower bounds to date.

For an explicit problem, Siegel's original bound is $s \geq \tilde{\Omega}(n \cdot (\frac{m}{n})^{\frac{1}{t}})$. This paper develops a **deterministic variant of cell sampling** to prove new and stronger bounds for the problem.

These results are derived via a novel combinatorial argument: showing that every dense enough hypergraph contains a small subset of vertices that spans many edges. This allows for a deterministic construction that improves over the probabilistic guarantees of earlier techniques.

Theorems

Theorem 1 *Fix a finite field $\mathbb{F}$ and a parameter $m = \mathrm{poly}(n)$.*

1. *There exists an explicit problem with n inputs and m queries such that every non-adaptive static data structure solving it with query time $t = 2$ requires space*
$$s \geq m - \tilde{O}\left(\frac{m}{n}\right).$$
2. *For every $t \geq 3$, there exists an explicit problem with n inputs and m queries such that every non-adaptive static data structure solving it with query time t requires space*
$$s \geq \Omega\left(n \cdot \left(\frac{m}{n}\right)^{\frac{1}{t-1}} \cdot \frac{1}{2^t \log(n) \log(m)}\right).$$

Theorem 2 *Let $G = (V, E)$ be a multigraph with $|V| = s \geq 2$ vertices and $|E| = m \geq s(1+\varepsilon)$ edges for some $\varepsilon = \varepsilon(s) \in (0, 1]$. Then there exists a set of vertices $S \subseteq V$ of size*
$$|S| \leq 8 \log(s) \cdot \left\lceil \frac{1}{\varepsilon} \right\rceil$$
spanning at least $|S| + 1$ edges.

Let $t \geq 3$ be an integer, and let $G = (V, E)$ be a t-hypergraph with $|V| = s \geq 2$ vertices and $|E| = m$ hyperedges. Let $k \in \mathbb{N}$ be a parameter such that $2^{t+2} \log(s) \leq k \leq s$. If
$$m \geq 3s \left(\frac{2^{t+3} \cdot s \cdot \log(s)}{k}\right)^{t-2},$$
then there exists a subset $S \subseteq V$ of size $|S| \leq k$ that spans at least
$$|S| + \frac{k}{2^{t+1} \log(s)}$$
hyperedges.

Bibliography

[1] Alan Siegel. *On universal classes of extremely random constant-time hash functions* SIAM Journal on Computing, 33(3):505-543, 2004

Martin Kopřiva
mkopriva@kma.zcu.cz

Presented paper by Aysel Erey

On the average order of a dominating set of a forest

(https://arxiv.org/pdf/2104.00600)

Introduction

A subset of vertices S is called a *dominating set* of G if every vertex in $V(G) \setminus S$ is adjacent to some vertex in S. Let $\mathcal{D}(G)$ denote the family of all dominating sets of G. Recently Beaton and Brown [1] introduced the *average order* of a dominating set of G, denoted $\text{avd}(G)$, which is given by

$$avd(G) = \frac{\sum_{S \in \mathcal{D}(G)} |S|}{|\mathcal{D}(G)|}.$$

They also showed that the complete graph K_n uniquely minimizes the average order of a dominating set among all graphs on n vertices. And it is trivial that the empty graph $\bar{K}_n$ with $\text{avd}(\bar{K}_n) = n$ has the largest average order of a dominating set among graphs on n vertices. But what if we do not allow isolated vertices? Which graphs have the largest average order of a dominating set among all graphs of order n without isolated vertices? In [1] Beaton and Brown showed that $\text{avd}(G) \leq \frac{3}{4}n$ for every graph G of order n without isolated vertices. However, the factor $\frac{3}{4}$ in the upper bound is not the best possible...

Conjecture 1 *[1] If G is a graph of order n with no isolated vertices, then* $\text{avd}(G) \leq \frac{2n}{3}$.

This Conjecture was verified for all graphs up to nine vertices, all graphs with minimum degree at least 4, and all quasiregularizable graphs. Beaton and Brown [1] also proved that

$$\text{avd}(G) > \frac{n - 1 + 2^{n-2}(n+1)}{2^{n-1} + 1} = \text{avd}(K_{1,n-1})$$

for every tree graph G of order n with $G \ncong K_{1,n-1}$ and hence the star $K_{1,n-1}$ is the unique extremal graph with the minimum average order of a dominating set. However, the problem of determining extremal graphs that maximize this parameter among trees remained as an open problem. In this paper, we solve this problem by proving the following.

Theorem 2 *If G is a tree of order $n \geq 2$, then* $\text{avd}(G) \leq \frac{2n}{3}$. *Moreover, the equality holds if and only if every non-leaf vertex of G is a support vertex with one or two leaf neighbors.*

We actually prove Conjecture 1 for forests (see Theorem 7), which immediately implies Theorem 2.

Main result

The domination number $\gamma(G)$ of a graph G is the cardinality of a minimum dominating set of G. Let $d_k(G)$ be the number of dominating sets of G with cardinality k. The domination polynomial of G, denoted by $D_G(x)$, is given by

$$D_G(x) = \sum_{k=\gamma(G)}^{|V(G)|} d_k(G) x^k$$

and it is easy to see that

$$\mathrm{avd}(G) = \frac{D'_G(1)}{D_G(1)}.$$

By the notation $N_G(v)$ we mean the open neighborhood of the vertex v in G, i.e. all neighbors of v in G, the closed neighborhood is $N_G[v] = N_G(v) \cup \{v\}$. An edge containing a leaf vertex is called a *pendant* edge and $L_G(v)$ denote the set of all leaf neighbors of a vertex u in G. We say that u is a support vertex of G if u is adjacent to a leaf vertex v of G, and u is called the *support* of v in G. A graph G/u be the graph obtained from G by deleting the vertex u and adding edges between all pairs of nonadjacent neighbors of u. Given a graph G with a specified vertex $u \in V(G)$, we write $G_{(u,k)}$ to denote the graph obtained by gluing G and K_{k+1} at the vertex u. That is, $G_{(u,k)} = K_{k+1} \cup G$ and $K_{k+1} \cap G = \{u\}$.

To prove the main result (see Theorem 7) we need to be familiar with the following lemmas.

Lemma 3 *[2] Let u and v be two vertices of G such that $N_G[v] \subseteq N_G[u]$. Then*

$D_G(x) = xD_{G/u}(x) + D_{G\setminus u}(x) + xD_{G\setminus N_G[u]}(x)$.

Lemma 4 *Let G be a graph of order n and w be a support vertex of G with $L_G(w) = \{v_1, \ldots, v_t\}$ for some integer $t \geq 1$. Let also $H = G \setminus L_G[w]$. Suppose that every vertex u in $N_G(w) \setminus L_G(w)$ is a support vertex in G, and $3D'_H(1) \leq 2(n-t-1)D_H(1)$. Then $3D'_G(1) \leq 2nD_G(1)$ with equality if and only if $t \in \{1, 2\}$ and $3D'_H(1) = 2(n-t-1)D_H(1)$.*

Lemma 5 *Let G be a graph and $u \in V(G)$. Then, for every integer $k \geq 1$,*

$D_{G_{(u,k)}}(x) = (x+1)^{k-1}[D_{G_{(u,1)}}(x) + D_{G\setminus u}(x)] - D_{G\setminus u}(x)$.

Lemma 6 *Let T be a tree with $|V(T)| \geq 3$ and u be a vertex of T. Suppose that u is not a support vertex of T and u has at most one neighbor in T which is not a support vertex in T. Let G_1 be the graph obtained from T by attaching a new leaf vertex v at u. Then*

1. $D_{G_1}(1) \leq D_T(1) + 3D_{T\setminus u}(1)$,
2. $D_{G_1}(1) \leq 5D_{T\setminus u}(1)$ *and*
3. $D_T(1) \leq 3D_{T\setminus u}(1)$.

Now we are ready to prove the main result.

Theorem 7 *Let G be a forest on n vertices with no isolated vertices. Then* $\mathrm{avd}(G) \leq \frac{2n}{3}$ *and, moreover, equality holds if and only if every non-leaf vertex of G is a suppport vertex with one or two leaf neighbors.*

Every graph G without isolated vertices contains a spanning forest F without isolated vertices and F can be obtained from G by a succession of non-pendant edge removals. Hence, one can ask...

Question 8 *In every graph G (which is not a disjoint union of stars or empty graphs) does there exist a non-pendant edge e of G such that* $\mathrm{avd}(G) < \mathrm{avd}(G \setminus e)$*?*

Observe that an affirmative answer to Question 8 would yield a proof of Conjecture 1 in general since our Theorem 7 and the remark above.

Bibliography

[1] Iain Beaton, Jason I. Brown: The Average Order of Dominating Sets of a Graph. *Disc. Math.* 344 (12) (2021), 112959 – `https://arxiv.org/pdf/2008.06531`.

[2] Tomer Kotek, James Preen, Frank Simon, Peter Tittmann, Martin Trinks: Recurrence relations and splitting formulas for the domination polynomial. *Elec. J. Comb.* 19 (3) (2012) $P47$ – `https://arxiv.org/pdf/1206.5926`.

Júlia Križanová

julia.krizannova@gmail.com

Presented paper by E. Cohen, H. Kaplan, Y. Mansour, S. Moran, K. Nissim, U. Stemmer, E. Tsfadia

Data Reconstruction: When You See It and When You Don't

(https://arxiv.org/pdf/2405.15753)

Introduction

The rigorous protection of individual privacy in data analysis has given rise to differential privacy (DP), a mathematical framework that bounds the influence any single record can have on query outputs. Despite its strong stability guarantees and widespread adoption in various domains, the the DP mechanisms remain vulnerable to *reconstruction attacks*. By aggregating noisy answers over multiple queries, an adversary can recover detailed approximations of the original dataset, thus undermining privacy at scale.

In this talk, we revisit the fundamental question of formally defining what constitutes a *reconstruction attack* by "sandwiching" the notion of reconstruction resistance between two complementary questions: (i) Under what conditions can one guarantee that a mechanism is immune to reconstruction attacks? (ii) Under what circumstances does a given attack clearly indicate that a system is not protected?

To address these questions, we introduce *Narcissus Resiliency*, a self-referential definitional paradigm that not only subsumes differential privacy but also captures classical security notions, such as one-way functions, entropy measures, and encryption schemes, as special cases. Furthermore, we establish a formal link to Kolmogorov complexity, providing a metric to assess the fidelity of reconstructed data and identifying the threshold beyond which a DP guarantee can no longer ensure protection.

Definitions and Theorems

Definition 1 ((ε, δ)-Differential Privacy) *Let $\mathcal{X}$ be a data universe and $\mathcal{Y}$ an output space. A randomized algorithm $\mathcal{M} : \mathcal{X}^n \longrightarrow \mathcal{Y}$ is said to satisfy* (ε, δ)-differential privacy *if for every pair of* neighboring *datasets $X, X' \in \mathcal{X}^n$ that differ in one entry, and for all events $S \subseteq \mathcal{Y}$:*

$$\Pr\Big[\mathcal{M}(X) \in S\Big] \;\leq\; e^{\varepsilon}\, \Pr\Big[\mathcal{M}(X') \in S\Big] \;+\; \delta.$$

Definition 2 ((ε, τ, D)-R-reconstruction-robust) *Let $\mathcal{X}$ be a data domain, let $\mathcal{D}$ be a distribution over datasets containing elements from $\mathcal{X}$, and let $R : \mathcal{X}^* \times \{0,1\}^* \;\to\; \{0,1\}$. Algorithm $\mathcal{M}$ is* (ε, τ, D)-R-reconstruction-robust *if for all attackers $\mathcal{A}$ it holds that*

$$\Pr_{\substack{S \leftarrow \mathcal{D} \\ y \leftarrow \mathcal{M}(S) \\ z \leftarrow \mathcal{A}(y)}}\left[R\Big(S, z\Big) = 1 \;\wedge\; \Pr_{T \leftarrow \mathcal{D}}\Big[R(T, z) = 1\Big] \leq \tau\right] \;\leq\; \varepsilon.$$

Definition 3 (Narcissus-resiliency) *Let $\mathcal{X}$ be a data domain, let $\mathcal{F}$ be a family of distributions over datasets containing elements from $\mathcal{X}$, and let $R : \mathcal{X}^* \times \{0,1\}^* \;\to\; \{0,1\}$. Algorithm $\mathcal{M}$ is* $(\varepsilon, \delta, \mathcal{F})$-$R$-Narcissus-resilient *if for all $\mathcal{D} \in \mathcal{F}$ and for all atackers $\mathcal{A}$, it holds that*

$$\Pr_{\substack{S \leftarrow \mathcal{D} \\ y \leftarrow \mathcal{M}(S) \\ z \leftarrow \mathcal{A}(y)}}\Big[R\Big(S, z\Big) = 1\Big] \;\leq\; e^{\varepsilon} \Pr_{\substack{S \leftarrow \mathcal{D} \\ T \leftarrow \mathcal{D} \\ y \leftarrow \mathcal{M}(S) \\ z \leftarrow \mathcal{A}(y)}}\Big[R\Big(T, z\Big) = 1\Big] \;+\; \delta.$$

Definition 4 ($K_{\mathcal{L}}$-Complexity) *Let $\mathcal{L}$ be a programming language (e.g., Python). The $K_{\mathcal{L}}$-complexity of a string x, denoted by $K_{\mathcal{L}}(x)$, is the length of the shortest $\mathcal{L}$-program that outputs x and halts. Similarly, given a set of strings X, we denote by $K_{\mathcal{L}}(X)$ the length of the shortest $\mathcal{L}$-program that outputs an element in X and halts.*

Definition 5 (Extraction definition, informal) *Let R be an extraction relation and $\mathcal{L}$ a programming language. We say that a string x is $(R, \mathcal{L})$-extractable from a string y* iff *there exists an $\mathcal{L}$-program $\mathcal{A}$ such that the following holds:*

1. *$\mathcal{A}(y)$ outputs z such that $R(x, z) = 1$, and*
2. *$K_{\mathcal{L}}\Big(\{\, z : R(x, z) = 1\}\Big) \gg |\mathcal{A}|$.*

We measure the quality of the extraction by $1 \;-\; \frac{|\mathcal{A}|}{K_{\mathcal{L}}\Big(\{\, z : R(x,z)=1\}\Big)}$.

Definition 6 (Resilience to Membership Inference) *Let $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ be an algorithm operating on an n-point dataset, and let $\mathcal{D}$ be a distribution over $\mathcal{X}$. We say $\mathcal{M}$ is* (δ, D)-MI-secure *if for every adversary $\mathcal{A}$,*

$$\left| \Pr_{\substack{S \leftarrow \mathcal{D}^n \\ y \leftarrow \mathcal{M}(S) \\ z \leftarrow \mathcal{D} \\ b \leftarrow \mathcal{A}(y,z)}} [\, b = 1 \,] \;-\; \Pr_{\substack{S \leftarrow \mathcal{D}^n \\ y \leftarrow \mathcal{M}(S) \\ z \in_R S \\ b \leftarrow \mathcal{A}(y,z)}} [\, b = 1 \,] \right| \;\le\; \delta.$$

Definition 7 *Define R_{MI} to be a (randomized) binary function taking two arguments: a dataset $Z \in \mathcal{X}^n$ and a (possibly randomized) function $f : \mathcal{X} \to \{0, 1\}$. Given Z and f, to compute $R_{\mathrm{MI}}(Z, f)$, sample a point $z \in Z$ and return $f(z)$.*

Theorem 8 *Let $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ be an algorithm and let $\mathcal{D}$ be a distribution over $\mathcal{X}$. Then $\mathcal{M}$ is $(\delta, \mathcal{D})$-MI-secure if and only if it is $\Big(0, \delta, \{\mathcal{D}^n\}\Big)$-$R_{\mathrm{MI}}$-Narcissus-resilient.*

Definition 9 (Extraction quality relation) *Fix a parameter $q \le 1$ controlling the desired quality of extraction. We define the relation R_{ext} which takes a dataset S and a pair (y, A), where y is an outcome of $\mathcal{M}$ and $\mathcal{A}$ is a program, and returns 1 if and only if both of the following hold:*

1. *$R\Big(S,\, \mathcal{A}(y)\Big) = 1$.*
2. *$\frac{|\mathcal{A}|}{K_{\mathcal{L}}\Big(\{\, z : R(S,z)=1\}\Big)} \;\le\; 1 - q$.*

Lemma 10 (Narcissus-resiliency prevents non-trivial extraction) *Let $\mathcal{M}$ be an $(\varepsilon, \delta, \mathcal{F})$–$R_{\mathrm{ext}}$-Narcissus-resilient mechanism, and let $\mathcal{B}$ be an adversary that, on view y returned by $\mathcal{M}$, outputs an $\mathcal{L}$-program A. Then for every distribution $D \in \mathcal{F}$ we have*

$$\Pr_{\substack{S \leftarrow \mathcal{D} \\ y \leftarrow \mathcal{M}(S) \\ \mathcal{A} \leftarrow \mathcal{B}(y)}} \left[\begin{array}{c} \mathcal{A} \text{ is an } (R, \mathcal{L})\text{-extraction} \\ \text{evidence of } S \text{ from } y \end{array} \right] \;\le\; e^{\varepsilon} \Pr_{\substack{S \leftarrow \mathcal{D} \\ T \leftarrow \mathcal{D} \\ y \leftarrow \mathcal{M}(S) \\ \mathcal{A} \leftarrow \mathcal{B}(y)}} \left[\begin{array}{c} \mathcal{A} \text{ is an } (R, \mathcal{L})\text{-extraction} \\ \text{evidence of } T \text{ from } y \end{array} \right] \;+\; \delta.$$

Bibliography

[1] Cynthia Dwork, Aaron Roth: The Algorithmic Foundations of Differential Privacy. `https://www.cis.upenn.edu/ aaroth/Papers/privacybook.pdf`

[2] Edith Cohen, Haim Kaplan, Yishay Mansour, Shay Moran, Kobbi Nissim, Uri Stemme, Eliad Tsfadia: Data Reconstruction: When You See It and When You Don't
https://arxiv.org/pdf/2405.15753

Martin Kunst
martin.kunst2001@centrum.cz

Stochastic matrices

Introduction

Stochastic matrices are a key tool in the fields of probability and statistics, used to model systems with random behavior. In this work, we focus on interval stochastic matrices, which generalize stochastic matrices by allowing entries to be real intervals instead of fixed probabilities. Interval stochastic matrices are useful in situations where transition probabilities between states are uncertain.

We aim to generalize properties of stochastic matrices to interval stochastic matrices, examining transience, recurrence, irreducibility.

Preliminaries

Definition 1 *A sequence of integer-valued random variables $(X_n)_{n\geq 0}$ is* a Markov chain *with initial distribution λ if for all $n \in \mathbb{N} \cup \{0\}$ and $i_0, \ldots, i_{n+1} \in S$, it holds that*

$$P(X_0 = i_0) = \lambda_{i_0}$$

$$P(X_{n+1} = i_{n+1} \mid X_0 = i_0, \ldots, X_n = i_n) = P(X_{n+1} = i_{n+1} \mid X_n = i_n),$$

where S is a countable set, called the state space, and its elements are called states.

If the state space of a Markov chain is finite, the transition probabilities between states can be represented by a matrix (called the transition matrix). For a finite Markov chain with n states, we denote the state space as $N = \{1, \ldots, n\}$.

Definition 2 *A matrix $A \in \mathbb{R}^{n\times n}$ is called* a stochastic matrix *if the sum of each of its columns is equal to one and $A_{i,j} \in [0, 1]$ for any $i, j \in N$.*

We denote the set of all stochastic matrices of size n as $\mathcal{S}^{n\times n}$ (or simply $\mathcal{S}$ when the dimension is clear from context). For a given Markov chain $(X_n)_{n\geq 0}$ with transition matrix A, the entry $A_{i,j}$ represents the probability $P(X_{n+1} = i \mid X_n = j)$ (the probability of transitioning from state j to state i).

Theorem 3 *Let A be a stochastic matrix representing the transition probabilities of a Markov chain $(X_n)_{n\geq 0}$ with initial distribution λ. For any states i, j, it holds that:*

$$P(X_{n+m} = j \mid X_m = i) = {A_{j,i}}^{(n)} = \sum_{i_1,\ldots,i_{n-1}} A_{i_1,i} \ldots A_{j,i_{n-1}},$$

$$P(X_n = j) = (A^{(n)}\lambda)_j.$$

For a given stochastic matrix A, we write $i \to j$ to denote that $P(X_n = j \mid X_0 = i) = (A^n)_{j,i} > 0$, meaning state j is reachable from state i. If both $i \to j$ and $j \to i$ hold, we say that i and j communicate, denoted as $i \leftrightarrow j$. The relation $\leftrightarrow$ defines an equivalence relation.

It is useful to view stochastic matrices as directed graphs. Specifically, a stochastic matrix $A \in \mathcal{S}^{n\times n}$ can be represented as a directed graph $G = (N, E, w)$, where $(i, j) \in E$ if and only if $A_{j,i} > 0$ and the edge weight function is given by $w(i, j) = A_{j,i}$.

The set of all edges leading from a given state v is denoted as $\delta(v)$, meaning $(v, u) \in \delta(v)$ if and only if $A_{u,v} > 0$. When discussing edges in a matrix $A \in \mathcal{S}$, we refer to pairs (i, j) such that $A_{j,i} > 0$, and when referring to vertices, we mean the column indices of A (i.e., the states).

The relation $\leftrightarrow$ partitions the state space into equivalence classes. These classes correspond to sets of states that communicate with each other. We introduce a key properties of these classes.

Definition 4 *We say that a class of states $C \subseteq N$ is closed in a stochastic matrix A if for every state $i \in C$ and $j \in N$, it holds that if $i \to j$, then $j \in C$.*

Thus, for a closed class C, it holds that if we are in a state $i \in C$, it is not possible to reach any state $j \notin C$ in the future.

Definition 5 *We say that a stochastic matrix $A \in \mathbb{R}^{n\times n}$ is **irreducible** if*

$$\forall i, j \in N : i \to j.$$

In other words, $A \in \mathbb{R}^{n\times n}$ is irreducible if and only if N forms a closed class in A.

Definition 6 *A stochastic matrix $A \in \mathbb{R}^{n\times n}$ is* irreducible *if*

$$\forall i, j \in N : i \to j.$$

In other words, $A \in \mathbb{R}^{n\times n}$ is irreducible if and only if N forms a closed class in A.

In practice, we can determine whether a stochastic matrix A is irreducible by finding its strongly connected components (for example, using Tarjan's algorithm). Each strongly connected component corresponds to an equivalence class. A graph representing a stochastic matrix is strongly connected if and only if the stochastic matrix is irreducible.

Definition 7 *Let $(X_n)_{n\geq 0}$ be a Markov chain. We define*

$$T_i = \inf\{n \geq 1 : X_n = i\} = \textit{first visit time of state } i.$$

The infimum of an empty set is defined as ∞. Let $1_{\{X_n=i\}}$ be the indicator function that takes the value 1 if $X_n = i$ and 0 otherwise. Then we define:

$$V_i = \sum_{n=0}^{\infty} 1_{\{X_n=i\}} = \textit{number of visits to state } i,$$
$$f_i = P(T_i < \infty | X_0 = i) = \textit{probability of returning to state } i.$$

Definition 8 *Given a Markov chain $(X_n)_{n\geq 0}$, we say that a state $i \in N$ is* recurrent *if*

$$P(V_i = \infty | X_0 = i) = 1.$$

If the state is not recurrent, we say that it is transient.

Theorem 9 *For any stochastic matrix A, the following hold:*

- *i is recurrent $\Leftrightarrow \sum_{n=0}^{\infty}(A^n)_{ii} = \infty \Leftrightarrow f_i = 1$*
- *i is transient $\Leftrightarrow \sum_{n=0}^{\infty}(A^n)_{ii} < \infty \Leftrightarrow f_i < 1$.*

Thus, a state is recurrent if, after leaving it, we are certain that we will return to it at some point in the future. On the other hand, transient states are those from which there is a nonzero probability of escaping to a state from which return is impossible. Recurrence and transience are class properties, meaning that a state in a class is recurrent if and only if the entire class is recurrent, and a state is transient if and only if the entire class is transient.

Theorem 10 *Let $A \in \mathbb{R}^{n\times n}$ be a stochastic matrix. A class $C \subseteq N$ is recurrent if and only if it is closed in A.*

Definition 11 *An interval stochastic matrix* $[A]$ *with a lower matrix* $\underline{A} \in \mathbb{R}^{n\times n}$ *and an upper matrix* $\overline{A} \in \mathbb{R}^{n\times n}$ *is an interval matrix where, for every* $i, j \in N$*, it holds that* $0 \leq \underline{A}_{i,j} \leq \overline{A}_{i,j} \leq 1$*, and at the same time, it holds that* $[A] \cap \mathcal{S} \neq \emptyset$*.*

Bibliography

[1] Levin, D.A. and Peres, Y: Markov Chains and Mixing Times

Matúš Matok
matus.matok@fmph.uniba.sk

Presented paper by S. M. Cioabă, S. Dewar, X. Gu

Spectral conditions for graph rigidity in the Euclidean plane

(https://kam.mff.cuni.cz/ spring/media/papers/3/spectral_rigidity_JS.pdf)

Rigidity is the property of a structure that does not flex. Arising from mechanics, rigidity has been studied in discrete geometry and combinatorics and has application in material science, engineering and biological sciences. A *d-dimensional framework* is a pair (G, p), where G is a graph and p is a map from $V(G)$ to $\mathbb{R}^d$. Two frameworks (G, p) and (G, q) are *equivalent* if $||p(u) - p(v)|| = ||q(u) - q(v)$ holds for every edge $uv \in E(G)$ and are *congruent* if the same equality holds for every $u, v \in V(G)$. $||.||$ denoted the Euclidean norm in $\mathbb{R}^d$. A framework (G, p) is *generic* if the coordinates of its point are algebraically independent over the rationals. The framework is *rigid* if there exists an $\varepsilon > 0$ such that if (G, p) is equivalent to (G, q) and $||p(u) - q(u)|| < \varepsilon$ for every $u \in V(G)$, then (G, p) is congruent to (G, q). The generic rigidity can be considered as a property of the underlying graph, hence a graph is called *rigid* in $\mathbb{R}^d$ if every/some generic realization of G is rigid in $\mathbb{R}^d$.

A d-dimensional framework (G, p) is *globally rigid* if every framework that is equivalent to (G, p) is congruent to (G, p). It was proven that if there exists a generic framework (G, p) that is globally rigid, then any other framework (G, q) in $\mathbb{R}^d$ will also be globally rigid. Following from this, we define a graph G to be *globally rigid* in $\mathbb{R}^d$ if there exists a globally rigid generic framework (G, p) in $\mathbb{R}^d$. A graph G is *redundantly rigid* in $\mathbb{R}^d$ if $G - e$ is rigid in $\mathbb{R}^d$ fir every edge $e \in E(G)$.

We will study rigidity from spectral point of view. We describe the matrices and the eigenvalues of our interest below. If G is an undirected simple graph with $V(G) = \{v_1, v_2, \ldots, v_n\}$, its *adjacency matrix* is the n by n matrix $A(G)$ with entries $a_{ij} = 1$ if there is an edge between v_i and v_j and a_{ij} otherwise, for $1 \leq i, j \leq n$. Let $D(G) = (d_{ij})_{1 \leq i,j \leq n}$ be the degree matrix of G, that is, the n by n diagonal matrix with d_{ii} being the degree of vertex v_i in G for $1 \leq i \leq n$. The matrix $L(G) = D(G) - A(G)$ is called the *Laplacian matrix* of G. For $1 \leq i \leq n$, we use $\mu_i(G)$ to denoted the i-th smallest eigenvalue of $L(G)$. It is not difficult to see that $\mu_1(G) = 0$. The second smallest eigevalue of $L(G)$, $\mu_2(G)$, is known as the *algebraic connectivity* of G.

Theorem 1 *Let G be a graph with minimum degree $\delta(G) \geq 6k$. If*

$$\mu_2(G - Z) > \frac{6k - 2k|Z| - 1}{\delta(G - Z) + 1},$$

for every $Z \subset V(G)$ with $|Z| \leq 2m$ then G has at least k edge-disjoint spanning rigid subgraphs.

Corollary 2 *Let G be a graph with minimum degree $\delta(G) \geq 6k$. If*

$$\mu_2(G) > 2 + \frac{2k - 1}{\delta - 1},$$

then G contains at least k edge-disjoint spanning rigid subgraphs.

Corollary 3 *Let G be a graph with minimum degree $\delta(G) \geq 6$. If*

$$\mu_2(G) > 2 + \frac{1}{\delta + 1},$$

then G is rigid.

Theorem 4 *Let G be a graph with minimum degree $\delta(G) \geq 6$. If*

$$\mu_2(G - Z) > \frac{6 - 2|Z|}{\delta(G - Z) + 1},$$

for every $Z \subset V(G)$ with $|Z| \leq 2$, then G is redundantly rigid.

Corollary 5 *Let G be a graph with minimum degree $\delta(G) \geq 6$. If*

$$\mu_2(G) > 2 + \frac{2}{\delta + 1},$$

then G is globally rigid.

Bibliography

[1] Sebastian M. Cioabă and Sean Dewar and Xiaofeng Gu: Spectral conditions for graph rigidity in the Euclidean plane.
`https://www.sciencedirect.com/science/article/pii/S0012365X21002405`

Amit Roy
amitroy@kam.mff.cuni.cz
Presented paper by Rémi de Joannis de Verclos

Chordal graphs are easily testable

(https://arxiv.org/pdf/1902.06135)

Introduction

A graph G on n vertices is ε-far from satisfying a property $\mathcal{P}$ if one has to add or delete at least εn^2 edges to G to obtain a graph satisfying $\mathcal{P}$. A hereditary class $\mathcal{P}$ of graphs is testable if for every fixed $\varepsilon > 0$ there is a size m_ε such that the following holds. If G is ε-far from $\mathcal{P}$ then a set $X \subseteq V(G)$ sampled uniformly at random among all subsets of $V(G)$ of size m_ε induces a graph $G[X]$ that is not in $\mathcal{P}$ with probability at least $1/2$. The property $\mathcal{P}$ is easily testable if moreover m_ε is a polynomial function of ε^{-1}. Otherwise, $\mathcal{P}$ is hard to test.

Definition 1 (Chordal Graphs) *A graph G is chordal if it does not contain any induced cycle of length at least four; i.e., any (≥ 4)-cycle in G has a chord (an edge between non-consecutive vertices of the cycle).*

Theorem 2 *The class of chordal graph is testable with query complexity $\mathcal{O}(\varepsilon^{-37})$.*

In particular, the class of chordal graph is easily testable.

Bibliography

[1] Verclos, Rémi de Joannis de. "Chordal graphs are easily testable." arXiv: Combinatorics (2019): n. pag.

Hadi Zamani
hadiz@iuuk.mff.cuni.cz

Presented paper by Claire Mathieu and Hang Zhou

A simple algorithm for graph reconstruction

(https://onlinelibrary.wiley.com/doi/pdf/10.1002/rsa.21143)

Introduction

How efficiently can we find an unknown graph using shortest path queries between its vertices? This is a natural theoretical question from the standpoint of recovery of hidden information. This question is related to discovering the topology of Internet networks, which is a crucial step for building accurate network models and designing efficient algorithms for Internet applications.
There are graphs for which reconstruction requires $\Omega(n^2)$ distance queries on general graphs[1]. So, our focus here is on the bounded degree case.

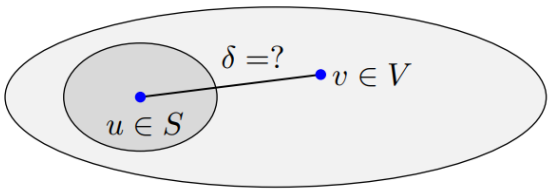
Question 1 *Is there any (randomized) algorithm to reconstruct a Δ-bounded degree graph G with $\tilde{O}(n)$[1] shortest path queries?*

The following simple algorithm provides a **partial** answer to this question.

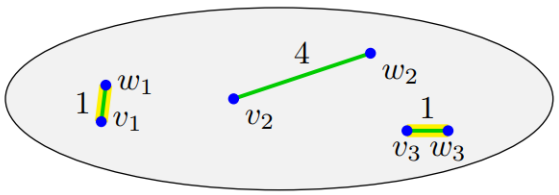
Algorithm 2 SIMPLE(V, s)

1: $S \leftarrow$ sample of s vertices selected uniformly and independently at random from V
2: **for** $u \in S$ and $v \in V$ **do**
3: QUERY $\delta(u, v)$
4: **end for**
5: $\hat{E} \leftarrow$ set of vertex pairs $\{a, b\} \subseteq V$ such that, for all $u \in S$, $|\delta(u, a) - \delta(u, b)| \leq 1$
6: **for** $\{a, b\} \in \hat{E}$ **do**
7: QUERY $\delta(a, b)$
8: **end for return** set of vertex pairs $\{a, b\} \in \hat{E}$ such that $\delta(a, b) = 1$

Observation 2 $E \subseteq \hat{E}$.



(a) Phase 1



(b) Phase 2

Random Δ-regular graphs

A random Δ-regular graph can be generated using the **configuration model**. In this model, each of the n vertices is assigned Δ stubs (half-edges), resulting in a total of Δn stubs. These stubs are

[1]The notation $\tilde{O}(f(n))$ stands for $O(f(n) \cdot \text{polylog}(f(n)))$.

then paired uniformly at random to form $\frac{\Delta n}{2}$ edges. The resulting multigraph may include self-loops and multiple edges.

Theorem 3 *Consider a uniformly random Δ-regular graph with $\Delta = O(1)$. Let $s = \log^2 n$. In the distance query model, Simple (Algorithm 1) is a reconstruction algorithm using $\tilde{O}(n)$ queries in expectation. In addition, Simple can be parallelized using 2 rounds.*

Lemma 4 *The output of Simple (Algorithm 1) equals the edge set E. The number of distance queries in Simple is $n \cdot s + |\hat{E}|$. In addition, Simple can be parallelized using 2 rounds.*

Lemma 5 *Let G be a uniformly random Δ-regular graph with $\Delta = O(1)$. Let $s = \log^2 n$. Let $S \subseteq V$ be a set of s vertices selected uniformly and independently at random from V. We have $\mathbb{E}_{G,S}[|\hat{E} \backslash E|] = o(1)$.*

Lemma 4 and 5 give us Theorem 3.

Structural lemma

Definition 6 ***(Distinguishing)***
For a vertex pair $\{a, b\} \subseteq V$, we say that a vertex $u \in V$ distinguishes a and b, or equivalently that u is a distinguisher of $\{a, b\}$, if $|\delta(u, a) - \delta(u, b)| > 1$. Let $D(a, b) \subseteq V$ denote the set of vertices $u \in V$ distinguishing a and b.

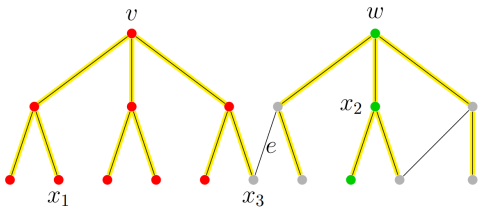
Lemma 7 ***(Structural lemma)***
Let $\Delta = O(1)$ be such that $\Delta \geq 3$. Let G' be a multigraph corresponding to a uniformly random configuration. Let $\{v, w\}$ be a vertex pair in G' such that $\delta(v, w) \geq 2$. With probability $1 - o\left(n^{-2}\right)$, we have $|D(v, w)| > 3n/\log n$.

To prove Lemma 7, we need the following.

Definition 8 ***(Interesting vertices)***
Let v, w be any non-adjacent pair of vertices. Do Breadth First Search (BFS) from v and w simultaneously.
*An edge is **exploring** if one endpoint is explored for the first time.*
*A vertex x is v-**interesting** (resp. w-interesting) if all of the following:*

- *the shortest path from v (resp. w) to x is unique*
- *all edges on that path are exploring*
- *all edges incident to that path are exploring.*

$I_k(v) \subseteq V$ denote the set of w-interesting vertices $x \in V$ such that $\delta(v, x) = k$.



BFS: exploring edges are yellow.
Red vertices are interesting with respect to v and green vertices are interesting with respect to w

Lemma 9 *For any integer $k \geq 1$, we have $I_k(v) \cup I_k(w) \subseteq D(v, w)$.*

Lemma 10 *Let $\Delta = O(1)$ be such that $\Delta \geq 3$. For positive integer $k \leq \left\lceil \log_{\Delta-1}(3n/\log n) \right\rceil + 2$, with probability $1 - o(n^{-2})$, we have $|I_k(v) \cup I_k(w)| > (\Delta - 2 - o(1))(\Delta - 1)^{k-1}$.*

This holds because of the following lemma, which says there are relatively few inexploring edges within a neighborhood of $\{v, w\}$.

Lemma 11 *Let $M = \lceil \log \log n \rceil$. We can construct two non-decreasing sequences $\{g_i\}_{1 \leq i \leq M}$ and $\{L_i\}_{1 \leq i \leq M}$, such that all of the following properties hold when n is large enough:*

1. *$g_1 = 3$; and for any $i \in [2, M]$, $g_i = o\left((\Delta - 1)^{L_{i-1}}/M\right)$;*
2. *$L_M \geq \left\lceil \log_{\Delta-1}(3n/\log n) \right\rceil + 2$;*
3. *With probability $1 - o(n^{-2})$, for all $i \in [1, M]$, strictly less than g_i edges are inexploring among the edges incident to vertices in $U_{\leq L_i} (:= \{x | \min(\delta(x, v), \delta(x, w)) \leq L_i\})$.*

Proving Lemma 5

Lemma 12 *Let $s = \omega(\log n)$ be an integer parameter. Let B be the set of vertex pairs $\{a, b\} \subseteq V$ such that $\delta(a, b) \geq 2$ and $|D(a, b)| \leq 3n \cdot (\log n)/s$. We have $\mathbb{E}_S[|\hat{E} \backslash E|] \leq |B| + o(1)$.*

From (structural) Lemma 7 and 12 we can prove Lemma 5.

Metric dimension

Definition 13 ***(Resolving set)***
A subset $S \subseteq V$ of vertices in a graph is a resolving set if for any pair of distinct vertices $v, w \in V$, there exists a vertex $u \in S$ such that $\delta(u, v) \neq \delta(u, w)$.

The **metric dimension** of a graph is the cardinality of the smallest resolving set.

Corollary 14 *A random subset S of $\log_2 n$ vertices is a resolving set for a random Δ-regular graph with high probability (w.h.p.).*

To prove it we use the following:

Lemma 15 *Let G be a uniformly random Δ-regular graph with $\Delta = O(1)$. With probability $1 - o(1)$, for any edge (a, b) of G there exists a vertex $c \in V \setminus \{a, b\}$ that is adjacent to a but not to b.*

Δ-bounded graphs

Theorem 16 *Consider a general graph of bounded degree $\Delta = O(\text{polylog}\, n)$. Let $s = n^{2/3}$. In the distance query model, Simple (Algorithm 1) is a reconstruction algorithm using $\tilde{O}\left(n^{5/3}\right)$ queries in expectation. In addition, Simple can be parallelized using 2 rounds.*

Bibliography

[1] Reyzin, Lev, and Nikhil Srivastava. "Learning and verifying graphs using queries with a focus on edge counting." International Conference on Algorithmic Learning Theory. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

Jakub Żuchowski
jakubzuchowski13@gmail.com/jakub.zuchowski.stud@pw.edu.pl
Presented paper by Jialei Song, Baogang Xu

Divisibility and coloring of some P_5-free graphs

(https://www.sciencedirect.com/science/article/pii/S0166218X24000350)

Introduction

In this talk, I present some structural and coloring properties of P_5-free graphs under additional restrictions. Here P_5 denotes the path on 5 vertices. In particular we forbid the *banner* (a 4-cycle with one pendant edge) and other small subgraphs. A graph G is *perfect* if $\chi(H) = \omega(H)$ for every induced subgraph H; by the Strong Perfect Graph Theorem [2], this means G has no induced odd cycle of length ≥ 5 (nor its complement). We denote by $\omega(G)$ the clique number (size of a largest clique) and by $\alpha(G)$ the independence number. In [1], Song and Xu defined notions of perfect divisibility and 2-divisibility. The authors shown that (P_5, banner)-free graphs are perfectly divisible, $(P_5, C_5, \text{banner}, \text{hammer})$-free graph G is $\omega^{\frac{3}{2}}(G)$-colorable, and for every P_5-free graph G with $\alpha(G) \geq 3$, G admits a 2-division if G is banner-free, and G is perfect if G is connected and $K_{1,3}$-free.

Definitions

We use the following definitions:

- P_5: the path on 5 vertices;
- C_5: the cycle on 5 vertices;
- **hammer**: the graph obtained from $P_5 = v_1v_2v_3v_4v_5$ by adding the chord v_1v_3;
- **banner**: the graph obtained from $P_5 = v_1v_2v_3v_4v_5$ by adding the chord v_1v_4;
- **claw**: the star $K_{1,3}$ (one center with 3 leaves).

A graph is $(P_5, \textit{banner})$-*free* if it contains no induced subgraph isomorphic to P_5 or a banner. Similarly we define $(P_5, C_5, \text{banner}, \text{hammer})$-free, $(P_5, K_{1,3})$-free, etc.

We also define certain decompositions of a graph G:

- A **perfect division** of G is a partition of $V(G)$ into A and B such that $G[A]$ is perfect and $\omega(G[B]) < \omega(G)$. We say G **admits a perfect division** if there exists a perfect division of $V(G)$.
- A **2-division** of G is a partition $V(G)$ into A and B such that $\max\{\omega(G[A]), \omega(G[B])\} < \omega(G)$. We say G **admits a 2-division** if $E(G) = \emptyset$ or there exists a 2-division of $V(G)$.
- G is **perfectly divisible** (resp. **2-divisible**) if every induced subgraph of G admits a perfect (resp. 2-) division.

Main results

The following are the main theorems of Song and Xu (2024) [1].

Theorem 1 *Let G be a (P_5, banner)-free graph. Then G is perfectly divisible. Moreover, if $\alpha(G) \geq 3$, then G admits a 2-division.*

Theorem 2 *If G is a connected $(P_5, K_{1,3})$-free graph with $\alpha(G) \geq 3$, then G is perfect.*

Theorem 3 *Every $(P_5, C_5, \text{banner}, \text{hammer})$-free graph G is $\omega^{\frac{3}{2}}(G)$-colorable.*

These extend earlier coloring and structural results for classes of P_5-free graphs.

Proof ideas

Here are outlined the key ideas behind the proofs of these theorems:

- **Theorem 1.** The proof assumes a minimal counterexample and performs a case analysis based on the existence of a 5-hole. If the graph contains such a cycle, its neighborhood is partitioned and analyzed structurally. Using forbidden subgraph constraints (P_5, banner), the authors prove that either a homogeneous set exists (contradicting minimality), or a perfect division can be constructed. When the independence number is at least 3, further case distinctions and clique structure arguments yield a valid 2-division.

- **Theorem 2.** Let G be a connected $(P_5, K_{1,3})$-free graph with $\alpha(G) \geq 3$. Assume G is imperfect. Then G contains an odd antihole, and Ben Rebea's Lemma implies the existence of an induced C_5. Since G is (P_5, banner)-free, the neighborhood of this C_5 is analyzed using special partition. Any stable set of size 3 in G must then intersect one of the cliques in this partition in at least two vertices, leading to a contradiction either via a P_5 or an induced claw. Thus, G must be perfect.

- **Theorem 3.** Suppose G is a minimal counterexample with clique number ω. Since G is $(P_5, C_5, \text{banner}, \text{hammer})$-free, one shows (using SPGT) that G must be perfect for $n \leq 7$. Otherwise G may have an antihole. The $(P_5, C_5, \text{banner}, \text{hammer})$-free conditions are used to control the neighborhood of the antihole. A detailed combinatorial argument shows that either the graph contains a suitable clique cutset enabling recursive coloring, or the coloring bound $\chi(G) \leq \lfloor \frac{3}{2}\omega(G) \rfloor$ holds directly via clique partitioning.

Bibliography

[1] Jialei Song, and Baogang Xu. *Divisibility and coloring of some P_5-free graphs.* Discrete Appl. Math. 348 (2024).

[2] Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. *The strong perfect graph theorem.* Ann. Math. 164 (2006).

List of participants

Adam Beneš

Martin Černý

Petr Chmel

Ondřej Chwiedziuk

Barbora Dohnalová

Adam Džavoronok

Jiří Fiala

Filip Filipkowski

Antonina Frąckowiak

Vojtěch Gaďurek

Anna Havelková

Milan Hladík

Karolína Hylasová

Adam Jahoda

Igor Januszkiewicz

Martin Kopřiva

Júlia Križanová

Martin Kunst

Matúš Matok

Richard Mužík

Ninad Rajgopal

Amit Roy

David Ryzák

David Sychrovský

Hadi Zamani

Jakub Żuchowski