Spring School 2024

Petr Chmel, Barbora Dohnalová (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 Vysoká Lípa (in Bohemian Switzerland in northern Bohemia) with a great variety of possibilities for outdoor activities.

Robert Šámal, Pavel Veselý
Petr Chmel, Barbora Dohnalová

KAM CSI IÚUK RSJ

Contents

Series Talks

Martin Černý

cerny@kam.mff.cuni.cz

Introduction to Cooperative Game Theory
as part of series Cooperative Game Theory

Introduction

Cooperative game theory, as introduced by von Neumann and Morgenstern in order to model the strength and power of coalitions, has been largely developed in the last fifty years. Applications of cooperative games are numerous, mainly in economics (fair division of a benefit/cost, bankruptcy problems, etc.), social choice (voting games), but also more recently in artificial intelligence and machine learning.

Cooperative games

Definition 1 (Cooperative game) *A* cooperative game *is an ordered pair* (N, v)*, where* N *is the set of players and* $v\colon 2^N \to \mathbb{R}$ *is its characteristic function. Further,* $v(\emptyset) = 0$*.*

What distinguishes cooperative games from other game-theoretical models is the absence of player actions. These are hidden behind the values of coalitions; the value $v(S)$ might be interpreted as the best joint utility of the players in S if they decide to act according to their best strategies.

The notion of *bigger coalition is a better coalition* can be captured by different properties of cooperative games. The following classes of games reflect this notion and form a hierarchy.

Definition 2 (Classes of games) *A cooperative game* (N, v) *is*

1. *monotone if it satisfies* $v(S) \leq v(T)$, $\quad S \subseteq T \subseteq N$*;*
2. *superadditive if* $v(S) + v(T) \leq v(T \cup S)$, $\quad S, T \subseteq N, S \cap T = \emptyset$*;*
3. *convex if* $v(S) + v(T) \leq v(T \cup S) + v(T \cap S)$, $\quad S, T \subseteq N$*.*

When the coalition of all players N cooperates, an important problem is how to allocate $v(N)$ between the players. Allocation $x \in \mathbb{R}^n$ is called *payoff vectors*. For simplicity, let $x(S) = \sum_{i \in S} x_i$. *Solution concepts* are formally subsets of payoff vectors following different goals. The following solution concepts captures the notion of *stability*.

Definition 3 (Core) *The* core $\mathcal{C}(v)$ *of a cooperative game* (N, v) *is defined as*

$$\mathcal{C}(v) = \{x \in \mathbb{R}^n \mid x(N) = v(N) \text{ and } x(S) \geq v(S) \text{ for every } S \subseteq N\}.$$

The next one-point solution concept satisfies several properties, which make it considered a *fair* allocation.

Definition 4 (Shapley value) *The* Shapley value $\phi\colon \Gamma^n \to \mathbb{R}^n$ *is defined as*

$$\phi_i(v) = \sum_{S \subseteq N \setminus i} \frac{s!(n-s-1)!}{n!}(v(S \cup i) - v(S)).$$

Bibliography

[1] Hans Peters, *Game Theory: A Multi-Leveled Approach*. Springer-Verlag Berlin AND Heidelberg Gmbh & Co. KG, 2015, ISBN 9783662469491.

Júlia Križanová
julia.krizannova@gmail.com
Presented paper by L. H. B. Olsen, I. K. Glad1, M. Jullum, Kjersti Aas

Applications of cooperative games in machine learning *as part of series* Cooperative Game Theory

(https://link.springer.com/article/10.1007/s10618-024-01016-z)

Introduction

Shapley values originated in cooperative game theory but are extensively used today as a model-agnostic explanation framework to explain predictions made by complex machine learning models in the industry and academia. There are several algorithmic approaches for computing different versions of Shapley value explanations.

Here, we consider Shapley values incorporating feature dependencies, referred to as conditional Shapley values, for predictive models fitted to tabular data. Estimating precise conditional Shapley values is difficult as they require the estimation of non-trivial conditional expectations. In this article, we develop new methods, extend earlier proposed approaches, and systematize the new refined and existing methods into different method classes for comparison and evaluation.

Terms and definitions

Shapley values assigned to each player j, for $j = 1...M$ uniquely satisfy following properties:

Fact 1 (Efficiency) *Their payoff sum corresponds to the value of the grand coalition $\mathcal{M}$ over the empty set $\emptyset$, i.e. $\sum_{j=1}^{M} \phi_j = v(\mathcal{M}) - v(\emptyset)$.*

Fact 2 (Symmetry) *Two equally contributing players j, k, i.e. $v(\mathcal{S} \bigcup \{j\}) = v(\mathcal{S} \bigcup \{k\})$ for all $\mathcal{S}$, receive equal payouts $\phi_j = \phi_k$.*

Fact 3 (Dummy) *A non-contributing player j, i.e. $v(\mathcal{S}) = v(\mathcal{S} \bigcup \{j\})$ for all $\mathcal{S}$, receives $\phi_j = 0$.*

Fact 4 (Linearity) *A linear combination of n games $\{v_1, ..., v_n\}$, that is $v(\mathcal{S}) = \sum_{k=1}^{n} c_k v_k(\mathcal{S})$, has Shapley values given by $\phi_j(v) = \sum_{k=1}^{n} c_k \phi_j(v_k)$.*

Shapley showed that the values which satisfy the axioms above are given by

$$\phi_j = \sum_{\mathcal{S} \in \mathcal{P}(\mathcal{M}) \setminus \{j\}} \frac{|\mathcal{S}|!(M - |\mathcal{S}| - 1)!}{M!} (v(\mathcal{S} \bigcup \{j\}) - v(\mathcal{S})),$$

where $|\mathcal{S}|$ is the number of players in coalition S and $\mathcal{P}(\mathcal{M})$ is the power set of M.

Filip Úradník
uradnik@kam.mff.cuni.cz

Learnability of Cooperative Games
as part of series Cooperative Game Theory

Introduction

Many problems, ranging from cost sharing to AI explainability, are modelled using cooperative game theory. The goal is to decide if cooperation is rational in a given situation and how the profit should be divided. However, the applications are limited to only selected cases with relatively low number of players. This is because to distribute the profit, values of all sub-coalitions of players is required — this being exponential in the number of players n. Incomplete cooperative game theory offers a way to describe the game using only selected coalition values. Additional structure of the game can be leveraged to obtain more information about the missing values.

However, the lack of full information can be used by each player in order to bargain with the others about his payoff. This in turn makes it difficult to agree on the profit distribution. A central *principal* may want to limit such exploitation by gathering more information about the game, i.e. obtaining values of coalitions which were previously unknown. The principal wants to choose such coalitions that, after their values are revealed, the strategic behaviour is limited as much as possible. This "coalition revealing strategy" offers insights into which coalitions are most important for a given class of games.

Background

Definition 1 (Cooperative game) *A* cooperative game *is an ordered pair* (N, v) *where* $N = \{1, \dots n\}$ *is the set of* players *and* $v\colon 2^N \to \mathbb{R}$ *is the* characteristic function *of the cooperative game. Further,* $v(\emptyset) = 0$.

We will refer to a subset of players $S \subseteq N$ as a *coalition.* Coalitions of size 1 are called *singletons* and N is the *grand coalition.* Further, we denote the set of all games on the set $N = \{1, \dots, n\}$ as Γ^n.

Definition 2 *The game* (N, v) *is said to be*

- monotone $\equiv (\forall S \subseteq T \subseteq N)(v(S) \leq v(T))$,
- super-additive $\equiv (\forall S, T \subseteq N : S \cap T = \emptyset)(v(S) + v(T) \leq v(S \cup T))$,
- convex $\equiv (\forall S, T \subseteq N)(v(S) + v(T) \leq v(S \cap T) + v(S \cup T))$.

The classes of *monotone, super-additive,* and *convex* games with n players are denoted $\mathbb{M}^n$, $\mathbb{S}^n$ and $\mathbb{C}^n$, respectively.

One of the goals of cooperative game theory is to find a way to distribute the payoff of the grand coalition to the players. This is done using so-called *payoff vectors* $x \in \mathbb{R}^n$, where x_i represents the payoff of player i. We say that a payoff vector is

- *efficient* $\equiv x(N) := \sum_{i \in N} x_i = v(N)$,
- *individually rational* $\equiv (\forall i)(x_i \geq v(i))$.

More generally, we will talk about *solution concepts*, is a function $\mathcal{S} : \Gamma^n \to 2^{\mathbb{R}^n}$, that assigns each game a payoff vector, or a set of payoff vectors, with given properties. One of the most studied solution concepts is the *Shapley value.*

Definition 3 (Shapley value) *Let (N, v) be a cooperative game. The* Shapley value *of player i is*

$$\phi_i(v) := \sum_{S \subseteq N \setminus \{i\}} \frac{s!\,(n-s-1)!}{n!} \left(v(S \cup \{i\}) - v(S)\right),$$

where $s := |S|$.

Incomplete Cooperative Games

Definition 4 (Incomplete game) *An* incomplete game *is a cooperative game, in which we do not know the values of v for all coalitions. Formally, it is a tuple $(N, \mathcal{K}, v)$, where*

- *$N = \{1, \ldots, n\}$ is the set of* players*,*
- *$\mathcal{K} \subseteq 2^N$ is the set of* coalitions with a known value*, $\emptyset \in \mathcal{K}$,*
- *$v : 2^N \to \mathbb{R}$ is the* characteristic function*, $v(\emptyset) = 0$.*

Further, we usually want for at least the so-called minimal information *to be present, which is $\mathcal{K}_0 := \{\emptyset, N\} \cup \{\{i\} \mid i \in N\}$. An incomplete game is said to* include minimal information *$\equiv \mathcal{K}_0 \subseteq \mathcal{K}$. We say that (N, v) is the* underlying full game *of $(N, \mathcal{K}, v)$.*

Definition 5 ($\mathbb{S}^n$-extension) *A game $(N, w) \in \mathbb{S}^n$ is a* $\mathbb{S}^n$-extension *of the incomplete game $(N, \mathcal{K}, v) \equiv \forall S \in \mathcal{K} : v(S) = w(S)$.*

The class of all $\mathbb{S}^n$-extensions of $(N, \mathcal{K}, v)$ is denoted by $\mathbb{S}^n(\mathcal{K}, v)$. If $\mathbb{S}^n(\mathcal{K}, v)$ is non-empty, then we say that $(N, \mathcal{K}, v)$ is $\mathbb{S}^n$-extensible*.*

As a reminder, $\mathbb{S}^n$ is the class of all super-additive games on n players. The extensions can be defined for a general class C, but in this talk, we will only discuss $\mathbb{S}^n$-extensions.

Principal's Problems

From now, we will assume that the underlying game $(N, v) \in \mathbb{S}^n$. Each player in the game can leverage the lack of knowledge of v to increase his profit as much as possible.

Definition 6 (Utopian game) *A player i's* utopian game *is (N, v_i), where*

$$v_i = \underset{v \in \mathbb{S}^n(\mathcal{K}, v)}{\arg\max}\, \phi_i(v)\,.$$

The utopian game represents the "dream game" of the player (the one where they receive the most money, of course). We call the difference between the "collective optimism" of the players, and reality, the *utopian gap.*

Definition 7 (Utopian gap) *The* utopian gap *in $(N, \mathcal{K}, v)$ is*

$$\mathcal{G}_{(N,v)}(\mathcal{K}) := \sum_{i \in N} \phi_i(v_i) - v(N)\,.$$

We imagine the utopian gap is the function the principal is trying to minimize. However, she has a fixed budget of τ coalitions she can reveal, not more. We establish two approaches, an *online* approach and an *offline* approach.

Definition 8 (Offline Principal's Problem) *Let $\mathcal{F}$ be a known distribution of super-additive cooperative games. The optimal solution of the* Offline Principal's Problem *is to find $\mathcal{S} \subseteq 2^{2^N \setminus \mathcal{K}}$, where*

$$\mathcal{S} = \underset{\mathcal{S} \subseteq 2^{2^N \setminus \mathcal{K}}, |\mathcal{S}| \leq \tau}{\arg\min} \mathop{\mathbb{E}}_{v \sim \mathcal{F}} \left[\mathcal{G}_{(N,v)}(\mathcal{K} \cup \mathcal{S}) \right].$$

Definition 9 (Online Principal's Problem) *Let $\mathcal{F}$ be a known distribution of super-additive cooperative games. The optimal solution of the* Online Principal's Problem *is to find a policy $\pi : (2^N \times \mathbb{R}) \times \mathbb{N} \to \Delta^{2^N \setminus \mathcal{K}}$, which receives the already revealed coalitions, along with their values, and the remaining budget τ, and returns a distribution on the rest of the coalitions, such that it minimizes the following quantity*

$$\mathop{\mathbb{E}}_{v \sim \mathcal{F}} \mathcal{G}_{(N,v)}(\mathcal{K}_F),$$

where $\mathcal{K}_F$ is $\mathcal{K}$ with τ extra coalitions revealed in sequence according to π.

PPO

Let us also give a brief introduction into the realm of *Reinforcement learning*, in the scope required for the talk. The general setup is as follows:

We have an agent, and an environment. There is a set of *states* $\mathcal{S}$, with a specified starting state s_0. The agent has a set of *actions* $\mathcal{A}$, and using these actions, interacts with the environment. The environment itself is represented by two black box functions: the *transition function* $\tau : \mathcal{S} \times \mathcal{A} \to \Delta^{\mathcal{S}}$, specifying how the state the agent is in changes, and the *reward function* $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The agent's aim is to maximize the cumulative reward $\mathbb{E}_{s \sim \tau} \sum_{t=0}^{T} r(s_t, a_t)$.

The agent is usually formalised as a *policy*, which is a function $\pi : \mathcal{S} \to \Delta^{\mathcal{A}}$, which for a given state s gives a distribution on which actions to take next. If the agent is represented by a neural network (or networks), the policy is usually parametrized by θ, which represents the specific weights of the neural network.

Definition 10 (Return) *Let $r(a, s)$ be the reward gained by taking action a in state s. Then for a sequence $s_0, a_0, s_1, \dots s_T$ the return is*

$$R_T = \sum_{t=1}^{T} r(a_t, s_t).$$

Definition 11 (q-value) *Let $\pi : \mathcal{S} \to \Delta^{|\mathcal{A}|}$ be the policy and $\tau : \mathcal{S} \times \mathcal{A} \to \Delta^{\mathcal{S}}$ be the transition function. Then*

$$v(s) := \mathbb{E}_{a \sim \pi} \left[\sum_{t=1}^{T} r(a_t, s_t) \right],$$

$$q(a, s) := \mathbb{E}_{a \sim \pi} \left[r(a, s) + \sum_{s' \in \mathcal{S}} \tau_{s'}(s, a) v(s') \right].$$

Definition 12 (PPO) *The PPO algorithm minimizes the following two losses*

$$L^{\pi}(\theta|\varphi) := -\mathbb{E}_{t \sim \mathcal{T}(\pi)} \left[\left[(R_t - v(s_t|\varphi)) \frac{\pi(a_t|\theta)}{\pi(a_t|\theta_0)} \right]_{1-\varepsilon}^{1+\varepsilon} \right],$$

$$L^{v}(\varphi|\theta) := \mathbb{E}_{t \sim \mathcal{T}(\pi)} \left[(R_t - v(s_t|\varphi))^2 \right],$$

along a trajectory $t \sim \mathcal{T}$ sampled under policy π.

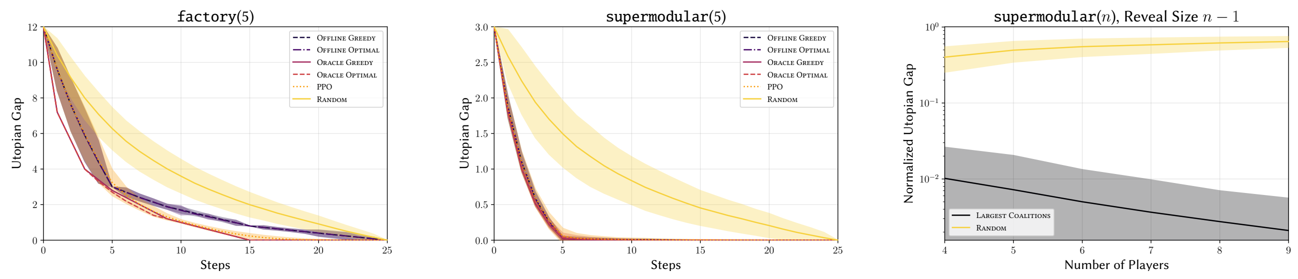


Figure 1: Plots from figures 1 and 3 of [1].

PPO has two parts, the *actor* and the *critic*. The critic tries to approximate the value of an action in a state (φ in the above definition). The actor (represented by θ) tries to approximate the best action, based on the current approximations via φ.

Bibliography

[1] Filip Úradník, David Sychrovský, Jakub Černý, and Martin Černý. 2024. *Reducing Optimism Bias in Incomplete Cooperative Games.* In Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024).

David Ryzák
david.ryzak99@gmail.com

Stochastic cooperative games *as part of series* Cooperative Game Theory

Introduction

Stochastic cooperative games, in this talk viewed as cooperative games with a stochastic characteristic function, are generalization of the classical model of von Neumann and Morgenstern. Since the stochastic model makes less assumptions about the characteristic function (values of coalitions are random variables) than the classical model we need to add some other information to be able to define solution concepts or in general to solve a cooperative game. One way to approach decision problems where randomness is present is to assume preferences of players to be defined over the set of outcomes. In this talk we assume all the players to be risk averse, i.e., second order stochastic dominance is used. We study so called SSD-dominating core. Especially, uniform distribution of characteristic function is assumed.

Stochastic cooperative games

Definition 1 (Stochastic cooperative game) *Stochastic cooperative game is a pair* (N, v), *where* $N = \{1, 2, \ldots, n\}$ *is a set of players and for any* $S \subseteq N$, $v(S) : (\Omega, \mathcal{F}, P) \longrightarrow \mathbb{R}$ *is a random variable on a probability space* $(\Omega, \mathcal{F}, P)$.

Definition 2 (Random payoff of a player under allocation type (d, r_+)**)** *Let* (N, v) *be a stochastic cooperative game. Payoff* x_i *of a player in a coalition* S *in* (N, v) *under allocation type* (d, r) *is defined as:*

$$x_i = d_i + r_i(v(S) - \mathbb{E}[v(S)])$$

Definition 3 (Feasible random payoff) *Let* (N, v) *be a stochastic cooperative game. Feasible payoff for a coalition* $S \subseteq N$ *is a random payoff* $(d, r_+) \in \mathbb{R}^{2n}$ *satisfying:*

- $d(S) = \mathbb{E}[v(S)]$,
- $r(S) = 1$,
- $r_i \geq 0, \ \forall i \in S$.

Definition 4 (Second order stochastic dominance) *Let* X, Y *be a random variables. A variable* X *stochastically dominates* Y *if for all concave nondecreasing utilitity function* u *it holds*

$$\mathbb{E}[u(X)] \geq \mathbb{E}[u(Y)].$$

Definition 5 (SSD- dominating core) *Let* (N, v) *be a stochastic cooperative game. SSD-dominating core is set of feasible random payoffs for the grand coalition* N *for which it holds that* $x(S) \succeq_{SSD} v(S)$ $\forall S \subseteq N$ *and* $x(N)$ *has the same distribution as* $v(N)$. *SSD-dominating core is denoted by* $\boldsymbol{DC}_{SSD}(v)$

Theorem 6 (Nonemptiness of SSD core under uniform distribution) *Let* (N, v) *be a stochastic cooperative game. Suppose* $\forall S \subseteq N, v(S)$ *has uniform distribution* $U[a_S, b_S]$, *where parameters* $a_S, b_S \in \mathbb{R}$ *are known. Let* (N, a) *and* (N, μ) *be a cooperative games with characteristic functions* *defined as* $a(S) = a_S$ *and* $\mu(S) = \mathbb{E}[v(S)]$ *respectively. Then the following implications hold:*

$$\begin{aligned}
&(N,a) \text{ is a convex game} \quad \& \;\; C(\mu) \neq \emptyset \implies \boldsymbol{DC}_{SSD}(v) \neq \emptyset,\\
&\boldsymbol{DC}_{SSD}(v) \neq \emptyset \implies C(\mu) \neq \emptyset \;\; \& \;\; C(a) \neq \emptyset.
\end{aligned}$$

Richard Mužík
richard@imuzik.cz
Presented paper by Alexandra B. Zinchenko

Set-valued Solutions for Cooperative Game with Integer Side Payments *as part of series* Cooperative Game Theory

(https://m-hikari.com/ams/ams-2014/ams-9-12-2014/zinchenkoAMS9-12-2014.pdf)

Introduction

The design of a cooperative game requires the assumptions of transferable utility and side payments. However, the utility can consist of indivisible units (e.g., one stock of something). This paper studies cooperative games with side payments and integer outcomes, demonstrating the effects of integer requirements. J. von Neumann and O. Morgenstern pointed out that such a modification of the utility concept 'would make our theory more realistic,' but 'it is clear that definite difficulties must be overcome in order to carry out this program.'

Definition 1 (Discrete game) *A discrete game is a game $G_D = (N, v)$, where $v(S) \in \mathbb{Z}$ and $x(S) \in \mathbb{Z}$ for all $S \subseteq N$. A set of all n player discrete games is denoted by $\mathcal{G}_D^N$.*

Definition 2 (Dual game) *Dual game to $G = (N, v)$ is $G^* = (N, v^*)$, where $v^*(S) = v(N) - v(N \setminus S)$ for $S \subseteq N$.*

Definition 3 (Imputation set) *Imputation set of $G = (N, v)$ is $I(G) = \{x \in X(G) | \forall S \subseteq N : x(S) \geq v(S)\}$, where $X(G) = \{x \in \mathbb{R} | x(N) = v(N)\}$.*

Definition 4 (Core of a game) *Core of $G = (N, v)$ is $C(G) = \{x \in X(G) \mid x(S) \geq v(S), \forall S \subseteq N\}$.*

Definition 5 (D-core of a game) *D-core of $G = (N, v)$ is $DC(G) = I(G) \setminus dom(I(G))$. $dom(I(G))$ are all dominated imputations. I.e. $dom(I(G)) = \{x \in I(G) | \exists y \in I(G) : (\forall i \in 1 \dots n : y_i \geq x_i) \wedge (\exists j \in 1 \dots n : y_j > x_j)\}$.*

Lemma 6 *Let $G_D \in \mathcal{G}_D^N$, $x \in I(G_D)$, $y \in I(G_D)$ such that $x \neq y$. Then there exists nonzero $\delta \in \mathbb{Z}^N$, such that $x = y + \delta$ and $\delta(N) = 0$.*

Lemma 7 *Let $G_D \in \mathcal{G}_D^N$, $x \in I(G_D)$, $y \in I(G_D)$ such that $y \succ_S x$ via some coalition $\emptyset \neq S \subseteq N$. Then $x(S) \leq v(S) - |S|$ and $2 \leq |S| \leq |N| - 1$.*

Theorem 8 *Let $G_D \in \mathcal{G}_D^N$. Then $C(G_D) = DC(G_D) \iff C(G_D) = I(G_D)$.*

Theorem 9 *Let $G_D \in \mathcal{G}_D^N$. Fix $k \in N$, $H \subseteq N \setminus k$ and let $\Omega' = \{S \in \Omega \mid S \neq \{i\}$ for $i \in H$ and $S \neq N \setminus i$ for $i \in (N \setminus H) \setminus k\}$, where $\Omega = 2^N \setminus \{N, \emptyset\}$. If the following condition holds, then $C(G_D) \neq \emptyset$:*

$$\begin{cases} v(S) \leq \sum_{i \in S \cap H} v(i) + \sum_{i \in S \setminus H} v^\star(i), & S \in \Omega', k \notin S \\ v(S) \leq v(N) - \sum_{i \in H \setminus S} v(i) - \sum_{i \in (N \setminus H) \setminus S} v^\star(i), & S \in \Omega', k \in S \end{cases}$$

Definition 10 (Stable set) *The stable set $NM(G)$ of game $G = (N, v)$ is defined by conditions:*

$$NM(G) \cap dom(NM(G)) = \emptyset \text{ (internal stability)}$$

$$I(G) \setminus NM(G) \subset dom(NM(G)) \text{ (external stability)}$$

The family of all stable sets of G is denoted by $\mathcal{NM}(G)$.

Theorem 11 *Let $G_D \in \mathcal{G}_D^N$ be a convex game. Then $\mathcal{NM}(G_D) = \{DC(G_D)\}$.*

Theorem 12 *Let $G_D \in (G)_D^N$. Then $C(G_D) \in \mathcal{NM}(G_D) \iff C(G_D) = I(G_D)$.*

Petr Vincena
vincena.petr@gmail.com

Cooperative games with skills in Open Anonymous Environments *as part of series* Cooperative Game Theory

Introduction

Traditional models of cooperative game theory consider agents as entities without more subtle differentiation. Agent as a whole brings some value to already existing coalition and this value is represented in the characteristic function. Instead, we can see agents not as "magical" units but as entities with some sets of skills and these skills they bring to the coalitions. This model allows us to represent more fine-grained picture of reality.

In open anonymous environments (such as internet), it is very easy for agents to collude (and create a bigger, non-existing agent) which has both their capabilities, separate themselves (and create smaller agents) or hide some of their skills in order to increase their profits. Traditional solution concepts are vulnerable to these 3 ways of manipulation and new concepts based on traditional ones are proposed.

Basic definitions

Definition 1 (Skills and agents) *Let T be the set of all possible skills. Each agent t has a subset of skills $S_t \subseteq T$. We assume that the skills are unique: $\forall t \neq u, S_t \cap S_u = \emptyset$.*

Definition 2 (Characteristic function over skills) *A characteristic function $v : 2^T \to \mathbb{R}$ assigns a value to each set of skills.*

Definition 3 (Hiding skills) *If agent i has a set of skills S_i, for any $S'_i \subseteq S_i$, it can declare that it has only S'_i.*

Definition 4 (False names or separation) *Agent i can use multiple identifiers and declare that each identifier has a subset of skills S_i. Since we assume each skill is unique, two different identifiers cannot declare they have the same skill. Thus, a false-name manipulation by agent i corresponds to a partition of S_i into multiple identifiers. (If the manipulation is combined with a skill-hiding manipulation, only a subset of S_i is partitioned.)*

Definition 5 (Collusion) *Multiple agents can collude and pretend to be a single agent. They can declare the skills of this agent to be the union of their skills (or a subset of this union, in case we combine the manipulation with a skillhiding manipulation).*

Definition 6 (Solution concept - Shapley value) *The Shapley value represents the idea that when you take all possible permutations of gradually joining the coalition, the mean of your contribution in all of these permutations should be your payback.*

Formally, give an ordering o of the set of agents W in the coalition, let $X(o, i)$ be the set of agents in W that appear before i in ordering o. Then the Shapley value for agent i is defined as

$$Sh(W, i) = \frac{1}{|W|!} \sum_{o} \left(w(X(o,i) \cup \{i\}) - w(X(o,i)) \right)$$

Definition 7 (Solution concept - core) *The core represents all possible paybacks with the property that no subgroup of agents have an intention to leave the coalition and create a separate coalition.*

Formally, for a set of agents W in the coalition and payback function w, the payback p is coalitionaly rational if:

$$\forall C \subseteq W : \sum_{i \in C} p_i \geq w(C)$$

.

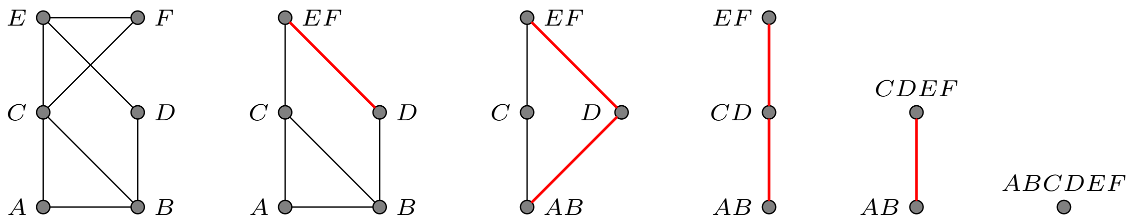
The core is then set of all such paybacks.

Jakub Balabán
balaban.jakub@gmail.com

Introduction to Twin-Width *as part of series* Twin-Width

Introduction

Twin-width is a graph parameter introduced in 2020 [1], which has attracted a lot of attention since then. Let us start with an informal description of twin-width (formal definition follows). Two vertices u, v in a graph are called *twins* if they have the same neighborhood (it does not matter whether uv is an edge). To show that the twin-width of a graph is small, we must show how the graph can be reduced to a single vertex by successively merging pairs of vertices that are "almost" twins. If the two merged vertices are twins, then the merging is equivalent to deleting one of them. Otherwise, there is a vertex which is adjacent to only one of them: this discrepancy is recorded via a so-called *red edge*. To show that the twin-width is at most k, we must avoid creating a vertex incident to more than k red edges. The following figure shows that the twin-width of the graph on the left is at most 2.



Formal definition

Definition 1 (Trigraph, contraction) *A* trigraph *G is a clique with the edge-set partitioned into three sets $B(G)$, $R(G)$, and $W(G)$: black, red, and white*[1] *edges. Let $G - u, v$ be the trigraph obtained from G by removing vertices u and v. A* contraction *of distinct vertices $u, v \in V(G)$ is the operation which produces a new trigraph G' defined as follows:*

- $V(G') = V(G - u, v) \sqcup \{w\}$;
- $B(G') = B(G - u, v) \cup \{wx \mid ux, vx \in B(G)\}$;
- $W(G') = W(G - u, v) \cup \{wx \mid ux, vx \in W(G)\}$;
- *$R(G')$ contains all remaining edges of G'.*

Definition 2 (Contraction sequence, Twin-width) *A sequence $(G_1, G_2, \ldots, G_n)$ of trigraphs is a* contraction sequence *of G_1 if each G_{i+1} for $i \geq 1$ is obtained from G_i by contracting two vertices, and G_n contains a single vertex. The* width *of a contraction sequence C is the maximum red degree over all vertices in all trigraphs in C (the red degree of u is the number of red edges incident to u). The* twin-width *of G, denoted* $\operatorname{tww}(G)$*, is the minimum width of any contraction sequence of G.*

[1]In the literature, white edges are usually called non-edges (white lines cannot be seen on paper), and a trigraph is defined as a graph with red and black edges.

An undirected simple graph G can be viewed as a trigraph G' such that $V(G') = V(G)$, $B(G') = E(G)$, $R(G') = \emptyset$, and $W(G')$ are the non-edges of G. Hence, we can talk about the twin-width of graphs.

Why should we care about twin-width, though? In the paper introducing twin-width [1], it was proven that if a contraction sequence of G of width d is provided, then any first-order (FO) formula[2] φ can be evaluated in FPT time, i.e., in time $f(|\varphi|, d) \cdot |G|$ for some computable function f. There are also more "practical" applications: for example, graphs of small twin-width can be compressed to size linear in the number of vertices (which allows BFS or DFS search in linear time) [5].

Basic properties

Observation 3 *The twin-width of a graph equals the twin-width of its complement.*

Observation 4 *The twin-width of a disconnected graph is the maximum twin-width over all its connected components.*

Observation 5 *If H is an induced subtrigraph of G, then $\operatorname{tww}(H) \le \operatorname{tww}(G)$. Moreover, replacing a red edge with a black or white edge does not increase twin-width.*

Observation 6 *Cliques, edgeless graphs, and stars have twin-width 0, paths have twin-width at most 1, and trees have twin-width at most 2.*

Proof For paths, it suffices to start contracting from one end of the path. For trees, fix an arbitrary root and repeat the following. If there are two leaves with the same parent, contract them. Otherwise, contract the deepest leaf with its parent. $\square$

Observation 7 *Graphs of twin-width 0 are exactly cographs, i.e., graphs that can be recursively built from single vertices by disjoint union and complementation.*

Computing twin-width

Observation 8 *The twin-width of an n-vertex graph G can be computed in time $2^{\mathcal{O}(n \cdot \log n)}$.*

Proof Try all possible contraction sequences of G. $\square$

Theorem 9 ([3]) *Graphs of twin-width 1 (and 0) can be recognized in polynomial time.*

Theorem 10 ([4]) *Recognizing graphs of twin-width at most 4 is NP-complete.*

Hence, an interesting open question is whether graphs of twin-width 2 and 3 can be efficiently recognized.

Other results

Theorem 11 ([1]) *Let G' be a trigraph obtained from a trigraph G by adding one vertex v and linking it with black edges to an arbitrary subset $X \subseteq V(G)$. Then $\operatorname{tww}(G') \le 2(\operatorname{tww}(G) + 1)$.*

Theorem 12 ([1]) *Any two-dimensional grid graph has twin-width at most 4. More generally, a d-dimensional grid graph has twin width at most $3d - 2$.*

The following two theorems are related: informally, one says that subdividing edges many times makes twin-width small, and the other one says that subdividing edges few times makes twin-width large.

[2]This means that we are only allowed to quantify over vertices, not over sets of vertices. An example of an FO formula is: $\exists u \forall v : u = v \vee E(u, v)$. It says that there is a vertex adjacent to all other vertices.

Theorem 13 ([4]) *If G is a graph obtained from an n-vertex graph by subdividing each edge at least $2\lceil \log n \rceil - 1$ times, then* $\operatorname{tww}(G) \le 4$.

Theorem 14 ([2]) *If $d \ge 0$ and G is a graph obtained from the clique K_n by subdividing each edge k-times for $1 \le k < \log_{d+1}(n-1) - 1$, then* $\operatorname{tww}(G) > d$.

Bibliography

[1] Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. *Twin-width I: Tractable FO Model Checking.* FOCS 2020, J. ACM 2022.

[2] Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. *Twin-width II: small classes.* SODA 2021.

[3] Édouard Bonnet, Eun Jung Kim, Amadeus Reinald, Stéphan Thomassé, and Rémi Watrigant. *Twin-width and Polynomial Kernels.* Algorithmica 2022.

[4] Pierre Bergé, Édouard Bonnet, and Hugues Déprés. *Deciding Twin-Width at Most 4 Is NP-Complete.* ICALP 2022.

[5] Max Bannach, Florian Andreas Marwitz, and Till Tantau. *Faster Graph Algorithms Through DAG Compression.* STACS 2024.

Jan Jedelský
xjedelsk@fi.muni.cz
Presented paper by Petr Hliněný and Jan Jedelský

Twin-width of Planar Graphs is at most 8, and some Related Bounds *as part of series* Twin-Width

(http://dx.doi.org/10.4230/LIPIcs.ICALP.2023.75)

Introduction

Twin-width is a graph parameter introduced by Bonnet et al. [1]. Informally speaking, it measures "similarity to cographs".

A graph is a cograph if it can be reduced to a single vertex by iterative identification of twins. We obtain twin-width by allowing identifications of near twins, while keeping track of the resulting "errors" by coloring edges red. We defer its formal definition for later.

A graph is planar if it can be drawn onto a plane without any crossed edges. In 2022, there has been a stream of upper bounds on twin-width of planar graph. So far, the best published upper bound is 8. Additionally, Kráľ and Lamaison [3] have found a planar graph of twin-width exactly 7.

In this talk, we show that twin-width of planar graphs is at most 8.

Outline of the presentation

We begin by showing that twin-width of planar graphs is at most 12. Then, we improve the upper bound to 9 by replacing arbitrary BFS tree with *left-aligned* BFS tree [2]. Finally, we improve the upper bound to 8 by avoiding contractions across "horizontal separators" during the contraction sequence.

Definition of twin-width

Definition 1 (Trigraph) *Let G be a (simple) graph and let c be its 2-edge-coloring (using colors* red *and* black*). Then, we say that $H := (G, c)$ is a trigraph.*

Red degree *of a vertex $v \in V(H) := V(G)$ is the number of red edges incident to v. We say that H is a d-trigraph, if its maximum red degree is at most d.*

We denote by $N_H(v)$ the set of neighbors of v in G. We denote by $N_H^R(v) \subseteq N_H(v)$ the set of *red neighbor* of v, that is, the set of vertices adjacent to v by a red edge.

We can also see graphs as a special case of trigraphs where all the edges are black.

Definition 2 (Contraction) *Let G and H be trigraphs, and let $\{u, v\} = V(G) \setminus V(H)$ and $\{w\} = V(H) \setminus V(G)$ be vertices.*

We say that H is created by a contraction *of u and v in G if the following holds:*

- $N_G(u) \cup N_G(v) \setminus \{u, v\} = N_H(w)$,
- $N_G^R(u) \cup N_G^R(v) \cup (N_G(u) \triangle N_G(v)) = N_H^R(w)$*, where $\triangle$ denotes symmetric difference, and*
- *for every $x \in V(G) \setminus \{u, v\} = V(H) \setminus \{w\}$, it holds that $N_G(x) \setminus \{u, v\} = N_H(x) \setminus \{w\}$ and $N_G^R(x) \setminus \{u, v\} = N_H^R(x) \setminus \{w\}$*

Definition 3 (Contraction sequence) *Let G be a (tri)graph, let $k \in \mathbb{N}_0$ be an integer, and let $G = G_n, G_{n-1}, \ldots, G_1 = K_1$ be a sequence of k-trigraphs starting with G and ending with a single vertex graph.*

Assume that, for all $i = 1, 2, \ldots, n-1$, it holds that G_i is created by a contraction from G_{i+1}.

*Then, we say that $G_n, G_{n-1}, \ldots, G_1$ is a k-*contraction sequence *for G.*

Definition 4 (Twin-width) *Let G be a (tri)graph. Twin-width of G is the smallest $k \in \mathbb{N}_0$ such that there is a k-contraction sequence $G = G_n, G_{n-1}, \ldots, G_1 = K_1$ for G.*

Bibliography

[1] Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. *Twin-width I: Tractable FO Model Checking.* Journal of the ACM, Volume 69, Issue 1, Article No.: 3pp 1–46.

[2] Petr Hliněný. *Twin-width of Planar Graphs is at most 9, and at most 6 when Bipartite Planar.* `https://arxiv.org/abs/2205.05378`

[3] Daniel Kral, and Ander Lamaison. *Planar graph with twin-width seven.* European Journal of Combinatorics, In Press.

Petr Chmel
chmel@iuuk.mff.cuni.cz

Introduction to Zero-Knowledge and zk-SNARKs *as part of series* Zero Knowledge

Zero Knowledge

Definition 1 (Interactive proof system) *For a function $f : \{0,1\}^n \to \mathcal{R}$, where $\mathcal{R}$ is a finite set, a k-message* interactive proof system *for f is a pair of algorithms $(\mathcal{V}, \mathcal{P})$, where $\mathcal{V}$ (the verifier) is a probabilistic algorithm running in time* $\text{poly}(n)$*, and a prescribed ("honest") deterministic algorithm $\mathcal{P}$ (the prover) that satisfy the following. Both $\mathcal{V}, \mathcal{P}$ are given a common input $x \in \{0,1\}^n$, and the prover then provides a value y claimed to be the value of $f(x)$. The prover and verifier then exchange a sequence of messages $m_1, \ldots, m_k$. The algorithm send their messages in alternating order, and in order to produce the message m_i, the currently sending algorithm run on the the input $(x, m_1, \ldots, m_{i-1})$.*

The sequence of messages $(m_1, \ldots, m_k)$, along with the claimed answer y is called the transcript.

In the end, the verifier $\mathcal{V}$ outputs a single bit b, where $b = 1$ indicates that the verifier accepts the claim $y = f(x)$, and $b = 0$ indicates that the verifier rejects the claim. We use $\text{out}(\mathcal{V}, x, r, \mathcal{P})$ *to denote the result of the protocol (whether the verifier accepts or rejects) for the verifier $\mathcal{V}$ with its internal randomness r, and the prover $\mathcal{P}$ with the common input x.*

We then say that the interactive proof system is valid, if it satisfies the two following conditions:

1. *(Completeness) For every $x \in \{0,1\}^n, \Pr_r[\text{out}(\mathcal{V}, x, r, \mathcal{P})] \geq 2/3$.*
2. *(Soundness) For every $x \in \{0,1\}^n$ and* every *deterministic prover strategy $\mathcal{P}'$, if $\mathcal{P}'$ sends a value $y \neq f(x)$ at the start of the protocol, then* $\Pr_r[\text{out}(\mathcal{V}, x, r, \mathcal{P})] \leq 1/3$.

Definition 2 (Interactive argument system) *An interactive proof system $(\mathcal{V}, \mathcal{P})$ is a valid interactive argument system, if it satisfies the same completeness condition as a valid interactive proof system, but the soundess condition is required to only hold for prover strategies running in polynomial time.*

We can also create these interactive proofs for languages, where the interaction is the same as in the function version, and the output of the verifier indicates whether the public input is in the language or not. Completeness then amounts to requiring that every string in the language is accepted with high probability, and every string not in the language is rejected with high probability.

Our goal with defining zero-knowledge is to capture the property that a correct proof does not tell us anything about a potential witness. We do this by saying that we can generate a distribution that is difficult to distinguish from the distribution of the actual transcripts.

Definition 3 (Zero-knowledge) *A proof or argument system with prescribed prover $\mathcal{P}$ and prescribed verifier $\mathcal{V}$ for a language L is said to be zero-knowledge if for any probabilistic polynomial time verifier strategy $\hat{\mathcal{V}}$, there exists a probabilistic polynomial time algorithm $\mathcal{S}$ (which can depend on $\hat{\mathcal{V}}$), called the simulator, such that for all $x \in L$, the distribution of the output $\mathcal{S}(x)$ of the simulator is "indistinguishable" from* $\text{View}_{\hat{\mathcal{V}}}(\mathcal{P}, \hat{\mathcal{V}})$*. Here,* $\text{View}_{\hat{\mathcal{V}}}(\mathcal{P}, \hat{\mathcal{V}})$ *denotes the distribution over transcripts generated by the interaction of prover strategy $\mathcal{P}$ and verifier strategy $\hat{\mathcal{V}}$ within the proof or argument system.*

We can interpret "indistinguishable" in multiple ways. We could require the two distributions to be

- identical, this would yield a *perfect zero-knowlege* system.
- statistically close, this would yield a *statistical zero-knowledge* system.
- computationally indistinguishable: all polynomial-time algorithms should not distinguish between the two distributions except with negligible probability. This would yield a *computational zero-knowledge* system.

We can also weaken the assumption in zero-knowledge so that the efficient simulator exists only for the prescribed verifier. This is still interesting, and we then say the systems with the weaker assumption are *honest-verifier zero-knowledge.*

Discrete logarithm and Schnorr's protocol

Definition 4 (Negligible function) *A function $f\colon \mathbb{N} \to \mathbb{R}_0^+$ is negligible, if for every polynomial p there exists an N such that $\forall n > N : f(n) < \frac{1}{p(n)}$.*

Definition 5 (DLOG experiment) *The discrete-logarithm experiment $\mathrm{DLog}_{\mathcal{A},\mathcal{G}}(n)$:*

1. *$\mathcal{G}(1^n)$ generates a group (G, q, g), where $q = \mathrm{ord}(G), \langle g \rangle = G$.*
2. *Choose a uniformly random $h \in G$.*
3. *$\mathcal{A}$ is given G, q, g, h, and outputs $x \in \mathbb{Z}_q$.*
4. *Return 1 if $g^x = h$.*

Definition 6 (DLOG assumption) *We say that the discrete logarithm problem is hard relative to $\mathcal{G}$ if all PPT $\mathcal{A}$ have $\Pr[\mathrm{DLog}_{\mathcal{A},\mathcal{G}}(n) = 1] \leq \mathrm{negl}(n)$.*

Protocol 7 (Schnorr's protocol) *Let G be a multiplicative cyclic group of prime order q with a generator g. The public input is $h = g^w$, where only the prover knows a private witness w.*
The prover picks a uniformly random r from $\mathbb{Z}_q$ and sends $a \leftarrow g^r$ to the verifier.
The verifier responds with a uniformly random element e from $\mathbb{Z}_q$.
The prover sends the value $z \leftarrow (we + r) \bmod q$.
The verifier finally checks that $a \cdot h^e = g^z$.

Theorem 8 *Schnorr's protocol is perfect honest-verifier zero-knowledge. Moreover, if the prover follows the protocol, then the verifier will accept with probability 1, and if, after sending the first message a, the prover can answer correctly to more than one challenge e, then the prover must know a witness.*

Fiat-Shamir transformation. If we have an interactive protocol where all verifier's responses are uniformly random elements, we can transform the protocol into a non-interactive version in the presence of a hash function H as a random oracle that both the prover and verifier have access to. The prover then runs as usual and pretends to run the interactive protocol. A message m_i that should be generated by the verifier in the interactive protocol is then instead generated as the result of $H(x, m_1, \ldots, m_{i-1})$. In the end, the prover sends a single message which consists of the transcript of the simulated protocol. The verifier can then check as if it was running the interactive protocol, except it also needs to verify that all the messages created by hashing were generated correctly.

zk-SNARKs

Definition 9 (Non-interactive argument system) *A non-interactive argument for a relation $\mathcal{R}$ of statements and proofs is a quadruple of probabilistic algorithms* $(\mathrm{Gen}, \mathrm{Prove}, \mathrm{Verify}, \mathrm{Sim})$*, where:*

- $(\mathrm{crs}, \mathrm{td}) \leftarrow \mathrm{Gen}(1^n, \mathcal{R})$*: the generator is given the security parameter n and a relation $\mathcal{R}$ and outputs a common reference string crs, and a trapdoor td.*

- $\pi \leftarrow \mathrm{Prove}(\mathrm{crs}, u, w)$*: the prover takes as input the crs, a statement u and a witness w, and outputs an argument π.*

- $b \leftarrow \mathrm{Verify}(\mathrm{crs}, u, \pi)$*: the verifier takes as input the crs, a statement u, and an argument π, and outputs $b = 1$ if the argumetn was accepted, and $b = 0$ if it was rejected.*

- $\pi \leftarrow \mathrm{Sim}(\mathrm{crs}, \mathrm{td}, u)$*: the simulator takes as input the crs, the simulation trapdoor td, and a statement u, and it produces an argument π.*

We note that zk-SNARK stands for “zero-knowledge succinct non-interactive argument of knowledge”.

Definition 10 (zk-SNARK) *We say that a non-interactive argument* $(\mathrm{Gen}, \mathrm{Prove}, \mathrm{Verify}, \mathrm{Sim})$ *is a zk-SNARK, if the following properties hold:*

- *Completeness: given a true statement for the relation, the prover should always convince the verifier.*

- *Knowledge soundess: for every PPT adversary $\mathcal{A}$ playing as the prover, there exists an extractor algorithm $\mathcal{E}_{\mathcal{A}}$ running in polynomial time such that whenever $\mathcal{A}$ produces a valid argument, then the extractor, when given full access to the state of the adversary, outputs a valid witness except with negligible probability.*

- *Succinctness: if the verifier runs in polynomial time in $n + |u|$, where u is the statement and n is the security parameter and the proof size is polynomial in n, we say that it is a preprocessing SNARK. If, moreover, the common reference string is polynomial in λ, we say the SNARK is fully succinct.*

- *Statistical zero knowledge: the distributions generated by the prover and the simulator are statistically close.*

Bibliography

[1] Anca Nitulescu, “zk-SNARKs: a gentle introduction”.
`https://www.di.ens.fr/ nitulesc/files/Survey-SNARKs.pdf`

[2] Justin Thaler, “Proofs, Arguments, and Zero-Knowledge”, draft version.
`https://people.cs.georgetown.edu/jthaler/ProofsArgsAndZK.html`

Dominik Stejskal
2dominik.stejskal2@gmail.com

The KZG polynomial commitment scheme *as part of series* Zero Knowledge

Introduction

A polynomial commitment scheme is a cryptographic primitive. It allows one party — a *prover* — to create a short *commitment* to a polynomial f. The commitment is then sent to another party — a *verifier* — and used to verify claimed evaluations of f. In this talk I will introduce the KZG polynomial commitment scheme [1] and prove some of its security properties — *evaluation binding* and *extractability*, under suitable cryptographic assumptions. The talk will roughly follow sections 15.1-2 in Justin Thaler's manuscript [2].

Bibliography

[1] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In Masayuki Abe, editor, ASIACRYPT 2010, volume 6477 of LNCS, pages 177–194. Springer, Heidelberg, December 2010.

[2] Justin Thaler (2022), "Proofs, Arguments, and Zero-Knowledge", Foundations and Trends® in Privacy and Security: Vol. 4: No. 2–4, pp 117-660.
https://people.cs.georgetown.edu/jthaler/ProofsArgsAndZK.html

Benjamin Bencik
bencikben@gmail.com

The PlonK protocol *as part of series* Zero Knowledge

Introduction

In this talk, I will describe the PlonK protocol, which enables to provide of arguments of knowledge with small proof size, fast verification and one-time trusted setup. This construction will rely on the polynomial commitment scheme KZG from the previous lecture.

Definition 1 (Arithmetic Circuit) *An arithmetic circuit $C(x, w) \to \mathbb{F}_p$ over the field $\mathbb{F}_p$ is a directed acyclic graph. Every node in it with in-degree zero is called an input gate and is labeled by either a variable or a field element in $\mathbb{F}_p$. Every other gate is labeled by either $+$ or $\times$ in the first case it is a sum gate and in the second a product gate. The argument $x \in \mathbb{F}_p^r$ is a public statement and $w \in \mathbb{F}_p^s$ is secret witness.*

Definition 2 (Argument System) *An argument system is a triple $(Setup, Prove, Verify)$ where:*

- *$Setup(C) \to$ public parameters pp*
- *$Prove(pp, x, w) \to$ proof π*
- *$Verify(pp, x, \pi) \to 0/1$*

Definition 3 (Completeness of an argument system) *An argument system is complete with respect to a negligible function $\varepsilon(x)$ if*

$$\forall x, w : C(x, w) = 0 \iff Pr[Verify(pp, x, Prove(pp, x, w)) = 1] = 1 - \varepsilon(|C|)$$

Definition 4 (Knowledge soundness of an argument system) *An argument system is knowledge sound with respect to a negligible function $\varepsilon(x)$ if*

$$Pr[pp \leftarrow Setup(C); \pi \leftarrow Prove(pp, x) : Verify(pp, x, \pi) = 1] < \varepsilon(|C|)$$

Definition 5 (Evaluation domain from root of unity) *For field $\mathbb{F}_p$ and the n-th root of unity $\omega \in \mathbb{F}_p : \omega^n = 1$ we define evaluation domain H as*

$$H = \{1, \omega, \omega^2, \omega^3, \ldots, \omega^{n-1}\}$$

Definition 6 (Lagrange basis)

$$L_i(x) = \begin{cases} 1 & x = \omega^i \\ 0 & otherwise \end{cases}$$

Lemma 7 *Schwartz-Zippel Lemma: Let $f(x_1, x_2, x_3, \ldots, x_n)$ be a non-zero polynomial over $\mathbb{F}_p$ with degree bound d and $(r_1, r_2, r_3, \ldots, r_n)$ are uniformly randomly and independently sampled from $\mathbb{F}_p$.*

$$Pr[f(r_1, r_2, r_3, \ldots, r_n) = 0] \leq \frac{d}{p}$$

Corollary 8 *Let $f(x_1, x_2, x_3, \ldots, x_n), g(x_1, x_2, \ldots, x_n)$ be a non-zero polynomials over $\mathbb{F}_p$ with degree bound d and $(r_1, r_2, \ldots, r_n)$ are uniformly randomly and independently sampled from $\mathbb{F}_p$ and $\frac{d}{p}$ is negligible. If $f(r_1, r_2, \ldots, r_n) = g(r_1, r_2, \ldots, r_n)$ then with high probability $f = g$.*

Theorem 9 *The PlonK interactive oracle proof is complete and knowledge-sound.*

Martin Pastyřík
martin.pastyrik@seznam.cz

Presented paper by Tianyi Liu et al.

Pianist: Scalable zkRollups via Fully Distributed Zero-Knowledge Proofs *as part of series* Zero Knowledge

(https://eprint.iacr.org/2023/1271.pdf)

Introduction

In the previous talks, we have seen the use of Polynomial Commitments, KZG and PLONK. These protocols are great but often present immense computation overhead for the prover. Although they may be computationally very strong, these players still seek a solution that would be more scalable. One of the first solutions that comes to mind is to distribute the computation to a group of machines and combine their partial solutions. The authors of the Pianist seem to achieve that by implementing the KZG with multivariate polynomials.

Definitions

Definition 1 (SCC scheme [2]) *An* SCC scheme *(signatures of correct computation) for a function family $\mathcal{F}$ is a tuple* $(\mathrm{KeyGen}, \mathrm{Setup}, \mathrm{Compute}, \mathrm{Verify}, \mathrm{Update})$ *of five* PPT *algorithms with the following specification:*

1. $(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathrm{KeyGen}(\lambda, \mathcal{F})$*: Algorithm* KeyGen *takes as input the security parameter λ and a function family $\mathcal{F}$. It outputs a public/secret key pair* $(\mathrm{PK}, \mathrm{SK})$*.* KeyGen *is run only once at system initialization by a trusted source;*
2. $\mathrm{FK}(f) \leftarrow \mathrm{Setup}(\mathrm{SK}, \mathrm{PK}, f)$*: Algorithm* Setup *(run by a trusted source) takes as input the secret key* SK*, the public key* PK*, and a function $f \in \mathcal{F}$. It outputs the function public key* $\mathrm{FK}(f)$ *for the function f;*
3. $(v, w) \leftarrow \mathrm{Compute}(\mathrm{PK}, f, \mathrm{a})$*: Algorithm* Compute *(run by an untrusted server) takes as input the public key* PK*, a function $f \in \mathcal{F}$ and a value $a \in \mathrm{domain}(f)$. It outputs a pair (v, w), where $v = f(a)$ and w is a signature;*
4. $\{0, 1\} \leftarrow \mathrm{Verify}(\mathrm{PK}, \mathrm{FK}(f), \mathrm{a}, v, w)$*: Algorithm* Verify *(run by any verifier) takes as input the public key* PK*, the function public key* $\mathrm{FK}(f)$*, value* $\mathrm{a} \in \mathrm{domain}(f)$*, a claimed result v and a signature w. It outputs* 0 *or* 1*;*
5. $\mathrm{FK}(f') \leftarrow \mathrm{Update}(\mathrm{SK}, \mathrm{PK}, \mathrm{FK}(f), f')$*: Algorithm* Update *(run by a trusted source) takes as input the secret key* SK*, the public key* PK*, the function public key* $\mathrm{FK}(f)$ *for the old function f and the updated function description f'. It outputs the updated function public key* $\mathrm{FK}(f0)$.

Definition 2 (Adaptive security of an SCC scheme [2]) *Let λ be the security parameter and let P be an SCC scheme* $(\mathrm{KeyGen}, \mathrm{Setup}, \mathrm{Compute}, \mathrm{Verify}, \mathrm{Update})$ *for a function family $\mathcal{F}$. We say that P is* adaptively secure *if no* PPT *adversary $\mathcal{A}$ has more than negligible probability* $\mathrm{negl}(\lambda)$ *in winning the following security game, played between the adversary $\mathcal{A}$ and a challenger:*

1. ***Initialization****. The challenger runs algorithm* KeyGen *which outputs* $(\mathrm{PK}, \mathrm{SK})$ *and then gives* PK *to the adversary but maintains* SK *secret;*

2. ***Setup and update***. *The adversary makes an oracle query to the* $\mathrm{Setup}(\mathrm{SK}, \mathrm{PK}, f_0)$ *algorithm, specifying an initial function* $f_0 \in \mathcal{F}$*, outputting* $\mathrm{FK}(f_0)$*. Then, for* $i = 1, \dots, k$*, where* $k = \mathrm{poly}(\lambda)$*, he makes a polynomial number of oracle queries to the* $\mathrm{Update}(\mathrm{SK}, \mathrm{PK}, \mathrm{FK}(f_{i-1}), f_i)$ *algorithm, each time specifying* $f_i \in \mathcal{F}$*. The challenger answers the queries by returning the resulting* $\mathrm{FK}(f_i)$*;*

3. ***Forgery***. *The adversary* $\mathcal{A}$ *outputs a point* $\mathrm{b} \in \mathrm{domain}(f_i)$ *for some* $0 \leq i \leq k$*, and the forgery* (b, v, w)*.*

The adversary $\mathcal{A}$ *wins the game if* $1 \leftarrow \mathrm{Verify}(\mathrm{PK}, \mathrm{FK}(f_i), \mathrm{b}, v, w)$ *and* $f_i(\mathrm{b}) \neq v$*.*

Definition 3 (Pianist: Distributed Bivariate Polynomial Commitment [1]) *Suppose* P *has* M *machines of* $P_0, \dots\ , P_{M-1}$ *and suppose* P_0 *is the master node. Given the bivariate polynomial* $f(X, Y) = \sum_{i=0}^{M-1} \sum_{j=0}^{T-1} f_{i,j} R_i(Y) L_j(X)$*, each machine holds* $f_i(X) = \sum_{j=0}^{T-1} f_{i,j} L_j(X)$*. The protocol proceeds as follows:*

- $\mathrm{DKZG.KeyGen}(1^{\lambda, M, T})$: *Generate* $\mathrm{pp} = \left(g, g^{\tau_X}, g^{\tau_Y}, (U_{i,j})_{\substack{0 \leq i < M, \\ 0 \leq j < T}} = \left(g^{R_i(\tau_Y) L_j(\tau_X)}\right)_{\substack{0 \leq i < M, \\ 0 \leq j < T}}\right)$, *with trapdoor* τ_Y *and* τ_X*. Let* P, V *hold* pp.
- $\mathrm{DKZG.Commit}(f, \mathrm{pp})$*: In the commitment phase, each* P_i *computes the commitment* $\mathrm{com}_{f_i} = \prod_{j=0}^{T-1} U_{i,j}^{f_{i,j}}$ *and sends it to* P_0*, where* $f_{i,j}$ *is the* j*-th entry in the evaluation representation of* $f_i(X)$*. After receiving commitments from others,* P_0 *computes* $\mathrm{com}_f = \prod_{i=0}^{M-1} \mathrm{com}_{f_i}$
- $\mathrm{DKZG.Open}(f, \beta, \alpha, \mathrm{pp})$*:*
  1. *Each* P_i *computes* $f_i(\alpha)$ *and* $q_0^{(i)}(X) = \frac{f_i(X) - f_i(\alpha)}{X - \alpha}$*.* P_i *computes* $\pi_0^{(i)} = g^{R_i(\tau_Y) q_0^{(i)}(\tau_X)}$ *using the public parameters and sends* $f_i(\alpha), \pi_0^{(i)}$ *to* P_0*.*
  2. *After receiving* $\left\{\left(f_i(\alpha), \pi_0^{(i)}\right)\right\}_{0 \leq i < M}$*,* P_0 *computes* $\pi_0 = \prod_{i=0}^{M-1} \pi_0^{(i)}$*, and also recover* $f(Y, \alpha) = \sum_{i=0}^{M-1} R_i(Y) f_i(\alpha)$*.*
  3. P_0 *computes* $f(\beta, \alpha)$ *and* $q_1(Y) = \frac{f(Y,\alpha) - f(\beta,\alpha)}{Y - \beta}$*.* P_0 *computes* $\pi_1 = g^{q_1(\tau_Y)}$ *and sends* $z = f(\beta, \alpha)$ *and* $\pi_f = (\pi_0, \pi_1)$ *to* V*.*
- $\mathrm{DKZG.Verify}(\mathrm{com}_f, \beta, \alpha, z, \pi_f, \mathrm{pp})$*:* V *parses* $\pi_f = (\pi_0, \pi_1)$*, and checks if* $e(\mathrm{com}_f / g^z, g) \overset{?}{=} e(\pi_0, g^{\tau_X - \alpha}) \cdot e(\pi_1, g^{\tau_Y - \beta})$*. It outputs* 1 *if the check passes, and* 0 *otherwise.*

Bibliography

[1] Tianyi Liu, Tiancheng Xie, Jiaheng Zhang, Dawn Song, Yupeng Zhang:
Pianist: Scalable zkRollups via Fully Distributed Zero-Knowledge Proofs
`https://eprint.iacr.org/2023/1271.pdf`

[2] Charalampos Papamanthou, Elaine Shi, Roberto Tamassia:
Signatures of Correct Computation
`https://eprint.iacr.org/2011/587.pdf`

[3] Helger Lipmaa, Roberto Parisella, and Janno Siim:
Constant-Size zk-SNARKs in ROM from Falsifiable Assumptions
`https://eprint.iacr.org/2024/173.pdf`

[4] Dan Boneh, Justin Drake, Ben Fisch, Ariel Gabizon:
Efficient polynomial commitment schemes for multiple points and polynomials
https://eprint.iacr.org/2020/081.pdf

Kristýna Mašková
maskovakristyna25@gmail.com

Algebraic Cryptanalysis of Poseidon *as part of series* Zero Knowledge

Introduction

In recent years there has been a rapid surge of interest in applications that build on cryptographic protocols relying on arithmetization. Some examples of such protocols include zero-knowledge (ZK) proofs and multiparty protocols. A recurring task in these applications consists of evaluating a hash function. However modern cryptographic hash functions such as SHA2 and SHA3 were built to operate over $\mathbb{F}_2$, while ZK protocols operate over $\mathbb{F}_q$. Therefore there has been a demand for hash functions designed to be natively efficient in $\mathbb{F}_q$.

Thus recent years have seen several new proposals for primitives designed for arithmetic protocols, usually referred to as arithmetization-oriented ciphers (AOC). However, their design allows for a concise description in terms of operations over $\mathbb{F}_q$, which in turn makes them vulnerable to algebraic attacks [3].

In this talk, we will present one such family of hash functions named Poseidon [1]. We will show how Poseidon can be described by a system of polynomial equations and how these equations can be used in a Gröbner basis attack [2].

Bibliography

[1] Lorenzo Grassi, Dmitry Khovratovich, Christian Rechberger, Arnab Roy and Markus Schofnegger. *Poseidon: A New Hash Function for Zero-Knowledge Proof Systems.* USENIX Association (Aug 2021).

[2] Tomer Ashur, Thomas Buschman and Mohammad Mahzoun. *Algebraic Cryptanalysis of HADES Design Strategy: Application to POSEIDON and Poseidon2.* Cryptology ePrint Archive, Paper 2023/537.

[3] Jan Ferdinand Sauer and Alan Szepieniec. *SoK: Gröbner Basis Algorithms for Arithmetization Oriented Ciphers.* Cryptology ePrint Archive, Paper 2021/870.

Standalone Talks

Adam Beneš

ad.benes@gmail.com

Presented paper by Sourav Chakraborty, N. V. Vinodchandran, Kuldeep S. Meel

Old and new approaches to counting distinct elements

(https://arxiv.org/pdf/2301.10191.pdf)

Introduction

In this talk, I will show you both old and new methods for counting distinct elements. The old methods are based on hash functions, while the new method is based on sampling. It is one of the fundamental problems in streaming algorithms.

Definition 1 (Number of distinct elements) *$F_0(A)$ is the number of distinct elements in A.*

Definition 2 (The problem) *Given a stream $A = (a_1, a_2, ..., a_m)$ of m elements where each $a_i \in [n]$, parameters ε, δ, output an (ε, δ)-approximation of $F_0(A)$. That is, output c such that $Pr[(1-\varepsilon) \cdot F_0(A) \leq c \leq (1+\varepsilon) \cdot F_0(A)] \geq 1 - \delta$.*

Algorithm 1: Counting Distinct Elements

```
Input : Stream $A = (a_1, a_2, \ldots, a_m)$, $\varepsilon$, $\delta$
Initialize $p \leftarrow 1$; $X = \emptyset$; $thresh = \left\lceil \frac{12}{\varepsilon^2} \log\left(\frac{8m}{\delta}\right) \right\rceil$;
for $i = 1$ to $m$ do
    $X \leftarrow X \setminus a_i$;
    With probability $p$, $X \leftarrow X \cup \{a_i\}$;
    if $|X| = thresh$ then
        Throw away each item with probability of $\frac{1}{2}$;
        $p \leftarrow \frac{p}{2}$;
        if $|X| = thresh$ then
            Output $\perp$;
        end
    end
end
Output $\frac{|X|}{p}$;
```

Fact 3 (Chernoff's Bound) *Let $v_1, v_2, ..., v_k$ be independent random variables taking values in $\{0, 1\}$. Let $V = \sum_{i=1}^{k} v_i$. Then, for $\beta > 0$, $Pr(|V - \mathbb{E}(V)| \geq \beta \cdot \mathbb{E}(V)) \leq 2e^{-\frac{\beta^2 \cdot \mathbb{E}(V)}{2+\beta}}$.*

Theorem 4 (Space complexity) *For any data stream A and any $0 < \varepsilon, \delta < 1$, the algorithm 1 outputs an (ε, δ)-approximation of $F_0(A)$. The algorithm uses $\mathcal{O}(\frac{1}{\varepsilon^2} \cdot \log n \cdot (\log m + \log \frac{1}{\delta}))$ space in the worst case.*

Definition 5 (Error) *Error : 'The algorithm algorithm 1 does not return a value in the range $[(1-\varepsilon)F0, (1+\varepsilon)F_0]$.'*

Definition 6 (Fail) *Fail : 'The algorithm 1 outputs $\perp$.'*

Theorem 7 $Pr(Fail) \leq \frac{\delta}{8}$.

Theorem 8 $Pr(Error \cap \overline{Fail}) \leq \frac{\delta}{2}$.

Observation 9 $Pr(Error) \leq Pr(Fail) + Pr(Error \cap \overline{Fail})$.

Bibliography

[1] Main paper,
`https://arxiv.org/pdf/2301.10191.pdf`

[2] Algorithms based on hashing (unit 2),
`https://www.cs.dartmouth.edu/ ac/Teach/data-streams-lecnotes.pdf`

[3] K minimum values sketch (section 2.5),
`http://dimacs.rutgers.edu/ graham/ssbd/ssbd2.pdf`

Fernando Cortés Kühnast
cortes@math.tu-berlin.de

Presented paper by Fernando Cortés Kühnast, Justin Dallant, Stefan Felsner, Manfred Scheucher

An Improved Lower Bound on the Number of Pseudoline Arrangements

(https://doi.org/10.48550/arXiv.2402.13107)

Introduction

Arrangements of pseudolines are classic objects in discrete and computational geometry. They have been studied with increasing intensity since their introduction almost 100 years ago. The study of the number B_n of non-isomorphic simple arrangements of n pseudolines goes back to Goodman and Pollack, Knuth, and others. It is known that B_n is in the order of $2^{\Theta(n^2)}$ and finding asymptotic bounds on $b_n = \frac{\log_2(B_n)}{n^2}$ remains a challenging task. In 2011, Felsner and Valtr showed that $0.1887 \leq b_n \leq 0.6571$ for sufficiently large n. The upper bound remains untouched but in 2020 Dumitrescu and Mandal improved the lower bound constant to 0.2083. Their approach utilizes the known values of B_n for up to $n = 12$.

We tackle the lower bound with a dynamic programming scheme. Our new bound is $b_n \geq 0.2721$ for sufficiently large n.

Preliminaries

An *arrangement of pseudolines* in the Euclidean plane $\mathbb{R}^2$ is a finite family of simple curves, called pseudolines, such that each curve approaches infinity in both directions and every pair intersects in exactly one point where the two curves cross. More generally, we call a collection of pseudolines *partial arrangement* if every pair intersects in at most one crossing-point.

We will focus be on *simple* arrangements, that is, no three or more pseudolines intersect in a common point (called *multicrossing*). Moreover, we consider all arrangements to be *marked*, that is, they have a unique marked unbounded cell, which is called *north-cell*. Two arrangements are *isomorphic* if one can be mapped to the other by an orientation preserving homeomorphism of the plane that also preserves the north-cell.

Main Result

A known strategy for constructing many inequivalent arrangements is the following.

- Start with a partial arrangement $\mathcal{L}$ of n lines consisting of k *bundles* $\mathcal{L}_1, \ldots, \mathcal{L}_k$ of parallel lines.
- Bound from below the number $F_k(n)$ of ways the lines in $\mathcal{L}$ can be rerouted locally.
- Apply Lemma 1.

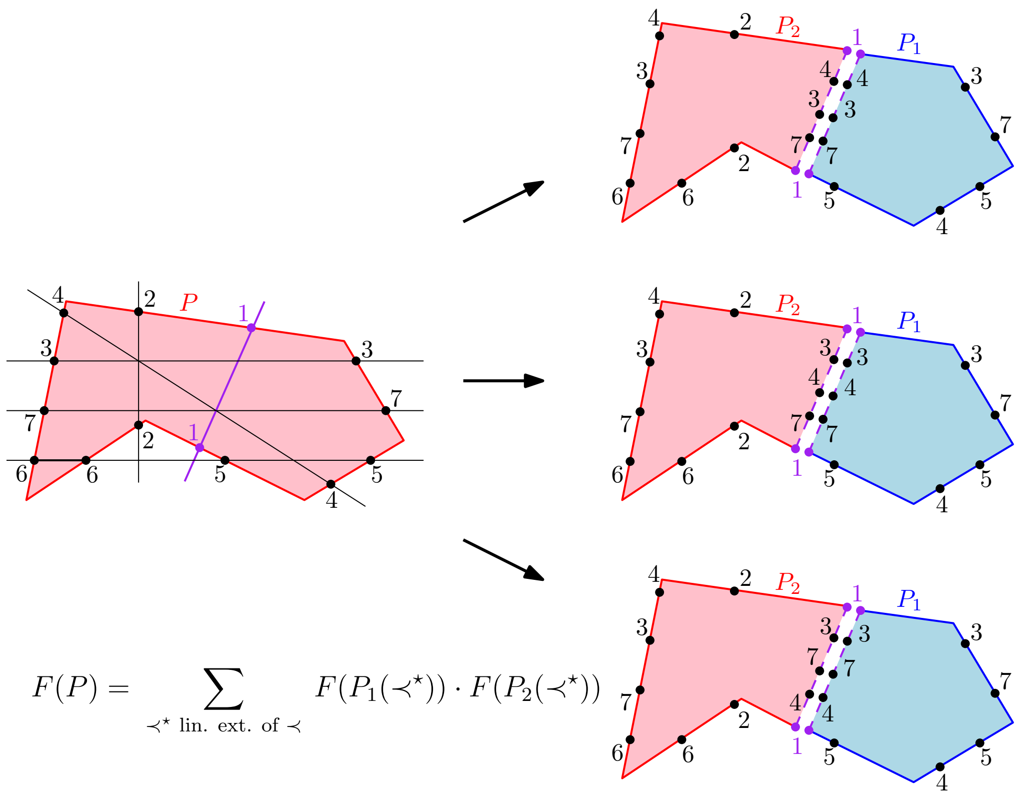
Lemma 1 *If $F_k(n) \geq 2^{cn^2 - O(n)}$ for some $c > 0$ then $B_n \geq 2^{\frac{k}{k-1}cn^2 - O(n \log n)}$.*

Our approach combines higher values of k with an increased locality for the perturbations. Instead of only locally resolving multicrossings, we allow reroutings of the arrangement within designated regions, which we call *patches*. By computing the number of reroutingss within the patches, and taking the product over all patches we obtain an improved lower bound on the number $F_k(n)$ of partial arrangements. This yields

Theorem 2 *The number B_n of non-isomorphic simple arrangements of n pseudolines satisfies the inequality $B_n \geq 2^{cn^2 - O(n\log n)}$ with $c > 0.2721$.*

Algorithms

We use two algorithms to compute the number $F(P)$ of reroutings of the arrangement within a patch P. The dynamic program, illustrated in Figure 2 can deal with any patch. In some cases we can compute the number of reroutings much more efficiently using the Lindström-Gessel-Viennot lemma.



$$F(P) = \sum_{\prec^\star \text{ lin. ext. of } \prec} F(P_1(\prec^\star)) \cdot F(P_2(\prec^\star))$$

Figure 2: An illustration of how to recurse on a patch P. When cutting along segment 1, highlighted purple, there are intersections with the segments 3, 4, and 7. As the segments 3 and 7 do no cross within P, there are only three possibilities for placing the three crossings along the segment 1, namely 4–3–7 (right top), 3–4–7 (right center) and 3–7–4 (right bottom).

Lemma 3 (Lindström, Gessel & Viennot) *Let G be a finite directed acyclic graph. Consider starting vertices $S = \{s_1, \ldots, s_k\}$ and destination vertices $E = \{e_1, \ldots, e_k\}$. For any two vertices u and v, let $p(u, v)$ be the number of paths from u to v. Assume that for any tuple of k vertex-disjoint paths starting in S and ending in E, the path starting at s_i necessarily ends at e_i, for all $1 \leq i \leq k$. Then the number of distinct such tuples is the determinant of the matrix*

$$M = \begin{pmatrix} p(s_1,e_1) & p(s_1,e_2) & \ldots & p(s_1,e_k) \\ p(s_2,e_1) & p(s_2,e_2) & \ldots & p(s_2,e_k) \\ \vdots & \vdots & \vdots & \vdots \\ p(s_k,e_1) & p(s_k,e_2) & \ldots & p(s_k,e_k) \end{pmatrix}.$$

Barbora Dohnalová

barca.dohnalova@seznam.cz

Presented paper by Greg Bodwin

An Alternate Proof of Near-Optimal Light Spanners

(https://arxiv.org/pdf/2305.18647.pdf)

Introduction

Given a graph G, $t-$spanner is a subset of edges such that the distance between any pair of vertices in the subset is at most t-times larger than the distance between them in G. It is well known that every graph has a $(2k-1)$-spanner on $O(n^{1+\frac{1}{k}})$ edges.

Instead of asking for the number of edges needed to create such a spanner, we can work with a weighted graph and ask about total weight of a t-spanner. Since this weight can be arbitrarily high, we instead divide this value by the total weight of a minimum spanning tree of G.

One of the possible algorithms for finding a spanner is the greedy algorithm, which simply tries to add edges in the order of nondecreasing weight. Perhaps suprisingly, this algorithm works quite well – in fact, it constructs an asymptotically optimal spanner.

In this paper, the author has shown the known bound: Every n-node graph G has a $(1+\varepsilon)(2k-1)$-spanner H of lightness $O_\varepsilon(n^{\frac{1}{k}})$. However, he shows it directly analysing the greedy algorithm, in contrast to the previous researchers, who used analysis of different algorithms.

In this talk we will show how to prove properties of spanners given by the greedy algorithm, using observations about graphs with high (weighted) girth.

Preliminaries

Definition 1 *Given a graph G, a t-spanner is an edge-subgraph H that satisfies $dist_H(u,v) \leq t \cdot dist_G(u,v)$ for all vertices u, v.*

Definition 2 *The lightness of a subgraph H of a graph G is the quantity*

$$l(H|G) := \frac{w(H)}{w(MST(G))}$$

where $MST(G)$ is a minimum spanning tree of G.

Theorem 3 *For all $\varepsilon > 0$ and positive integers k, n, every n-node graph G has a $(1+\varepsilon)(2k-1)$-spanner H of lightness $l(H|G) = O_\varepsilon(n^{\frac{1}{k}})$.*

Warmup 1: Spanner Sparsity

Theorem 4 *For all positive integers k, n, every n-node graph G has a $(2k-1)$-spanner H on $|E(H)| = O(n^{1+\frac{1}{k}})$ edges. This tradeoff is best possible, assuming the girth conjecture.*

Theorem 5 (Moore Bounds) *For any positive integers n, k, every n-node graph H with girth $> 2k$ has $O(n^{1+\frac{1}{k}})$edges.*

Warmup 2: Weaker Lightness

Definition 6 *For a cycle C in G, we define its normalized weight to be*

$$w^*(C) := \frac{w(C)}{\max_{e \in C} w(e)}$$

. The weighted girth of G is the minimum value of $w^*(C)$ over all cycles C in G.

Theorem 7 *Let $\varepsilon > 0$, let k, n be positive integers, and let H be an n-node graph with a unit-weight cycle C and weighted girth $> (1+2\varepsilon) \cdot 2k$. Then*

$$w(H) = O(\varepsilon^{-1} k n^{1+\frac{1}{k}})$$

Definition 8 *A path π is safe for an edge (u,v) if, for some integer $0 \leq s \leq \varepsilon w(u,v)$, it has the following structure: it starts with a prefix of exactly s forward spanning cycle edges, then it uses the edge (u,v), and then it ends with a suffix of exactly s backward spanning cycle edges. We will say that π is extra safe for (u,v) if in fact $s \leq \varepsilon w(u,v)/2$.*

Definition 9 *A path π in H is a safe k-path if it can be partitioned into k subpaths $\pi = q_1 \circ \ldots \circ q_k$ where each path q_i is safe for an edge e_i. We say that π is an extra-safe k-path if each path q_i is extra-safe for e_i. We say that π is monotone if $w(e_1) < \ldots < w(e_k)$.*

Main Result

Theorem 10 *Let $\varepsilon > 0$, let k,n be positive integers, and let H be an n-node graph with a unit-weight spanning cycle C and weighted girth $(1+4\varepsilon) \cdot 2k$. Then*

$$w(H) = O(\varepsilon^{-1} n^{1+\frac{1}{k}})$$

Bucket B_i is the set of edges in $H \setminus C$ with weights in the range $[2^i, 2^{i+1})$.

Definition 11 *A path π in H is safe for bucket B_i if it is non-backtracking (meaning that it does not repeat any edge twice in a row), all of its non-spanning-cycle edges are in B_i, and for some integer $0 \leq \varepsilon k 2^i$ it contains exactly $2s$ spanning cycle edges, where the first s are in the forward direction and the last s are in the backward direction. We say that π is extra-safe if in fact $s \leq \varepsilon k 2^{i-1}$.*

Definition 12 *A path π in H is a bucket-monotone safe k-path if it has exactly k non-spanning-cycle edges in total, and it can be partitioned into (possibly empty) subpaths $\pi = q_0 \circ \ldots \circ q_j$, where each subpath q_i is safe for bucket B_i (and so these bucket weights are increasing along π). We say that π is extra-safe if each subpath q_i is extra-safe for B_i.*

Karolina Drabik

kd417818@students.mimuw.edu.pl

Presented paper by Tereza Klimošová, Maya Stein

Antipaths in oriented graphs

(https://doi.org/10.1016/j.disc.2023.113515)

Introduction

In undirected graphs a large minimum degree is helpful for finding long paths. For example if G has minimum degree at least k, a simple greedy embedding shows that G contains a path on k vertices. If an n-vertex graph has minimal degree at least $n/2$, it contains a Hamiltonian cycle and if it has minimal degree greater than $(n-2)/2$, it contains a Hamiltonian path. One can also show that any connected graph on at least $k+1$ vertices that has minimal degree greater than $(k-1)/2$ contains a path of length k. It would be interesting to find extensions of these results to directed graphs, or particularly to oriented graphs. The natural parameter in this case will be the *minimal semidegree*, which is defined as minimum over all in and out-degrees over all vertices. For example if an n-vertex directed graph has minimal semidegree $n/2$, it contains a directed Hamiltonian cycle. In this talk we will focus on oriented graphs and, so called, *antipaths*, that means oriented paths with alternating edge directions. We will prove that for any natural number $k \geq 1$, any oriented graph D of minimum semidegree at least $(3k-2)/4$ contains an antidirected path of length k.

Definitions and notation

Definition 1 (Digraph) *A directed graph (digraph) is a graph D with oriented edges (at most one for each direction for each pair of vertices). Formally, $D = (V, E)$ where $E \subseteq V \times V$.*

Let denote uv to be a directed edge from u to v and say u is an *in-neighbor* of v and v is an *out-neighbor* of u. Let $d^-(v) = |\{u : uv \in E\}|$ denote the *in-degree* of v and $d^+(v) = |\{u : vu \in E\}|$ denote the *out-degree* of v.

Definition 2 (Oriented graph) *A digraph D is called oriented graph if for each pair $u, v \in V(D)$ at most one of the edges uv, vu is present.*

Definition 3 (Minimal semidegree) *Let D be a directed graph. The minimal semigegree of D is defined as minimum over all in- and out-degrees over all vertices of D. Formally,*

$$\delta^0(D) = min\{d^+(v), d^-(v) : v \in V(D)\}.$$

Definition 4 (Minimal pseudo semidegree) *Let D be a directed graph. If D is not edgeless, the minimal pseudo semidegree of D is defined as*

$$\bar{\delta}^0(D) = min \begin{cases} min\{d^-(v) : v \in V(D), d^-(v) > 0\} \\ min\{d^+(v) : v \in V(D), d^+(v) > 0\}. \end{cases}.$$

Otherwise, $\bar{\delta}^0(D) = 0$.

Definition 5 (Antipath/anticycle) *Antipath (anticycle) is an oriented path (cycle) such that each vertex has either out-degree 0 or in-degree 0.*

Main results

Theorem 6 *Let $k \in \mathbb{N}$ with $k \geq 3$ and let D be an oriented graph with $\bar{\delta}^0(D) \geq (3k-1)/4$. Then D contains each antipath of length k.*

Theorem 7 *For each $k \in \mathbb{N}^+$, every oriented graph D with more than $(3k-4)|V(D)|/2$ edges contains each antipath of length k.*

Tools used in the proofs

Fact 8 *Let $m \in \mathbb{N}^+$, let $1 \le \ell \le m$ and let G be a graph. Let $X, Y \subseteq V(G)$, with $X = \{x_0, \ldots, x_{m-1}\}$ and $Y = \{y_1, \ldots, y_m\}$, and let $F_0, F_m \subseteq E(G)$. If $d_{F_0}(x_0, Y) + d_{F_m}(y_m, X) \ge m + \ell$, then there is an index i with $\ell \le i \le m$ such that $x_0 y_i \in F_0$ and $x_{i-\ell} y_m \in F_m$.*

Lemma 9 *Let $k \in \mathbb{N}$ and let D be an oriented graph of minimum pseudo semidegree $\overline{\delta}^0(D) \ge k/2$. Let $P = v_0, \ldots, v_m$ be the longest antipath in D. If $m < k$ then m is odd.*

Lemma 10 *Let $k \in \mathbb{N}$, let D be an oriented graph of minimum pseudo semidegree $\overline{\delta}^0(D) \ge (3k-4)/2$ and let m be the maximum length of an antipath in D. If $1 < m < k$, then D contains an anticycle of length $m+1$.*

Lemma 11 *Let $k \in \mathbb{N}$, let D be an oriented graph of minimum pseudo semidegree $\overline{\delta}^0(D) > k/2$, and let C be an anticycle of length $m+1$ in D. If $m < k$, then D has an antipath of length $m+1$.*

Lemma 12 *Let $\ell \in \mathbb{N}$. If a digraph D has more than $\ell|V(D)|$ edges, then it contains a digraph D' with of minimum pseudo semidegree $\overline{\delta}^0(D') \ge (\ell+1)/2$.*

Filip Filipkowski
filip.filipkowski6@gmail.com

Presented paper by Tung Nguyen, Alex Scott and Paul Seymour

A note on the Gyárfás-Sumner conjecture

(https://link.springer.com/article/10.1007/s00373-024-02754-z)

Introduction

The Gyárfás-Sumner conjecture says that for any given tree T, graphs not containing a copy of T as an induced subgraph are χ-bounded. While it remains open and has only been proved for some certain families of trees, we will show that the conjecture is true when we change the word "induced" to "path-induced".

Definitions

Definition 1 - a path induced copy: Let T be a tree and let r be a vertex of a tree. Let φ be the isomorphism from T to subgraph of G (not necessarily induced). We say that φ is a path-induced copy of (T, r) if for every path P in T with one end being r, $\varphi(P)$ is an induced path in G.

Definition 2 - a c-creature: For a given number c we say that $X \subseteq V(G)$ is a $c-creature$ of G if for every vertex $v \in X$, v has less than c neighbours in $V(G) \setminus X$.

Definition 3 - χ-bounded graphs: A class $\mathcal{G}$ of graphs is χ-bounded if there is some function f such that, for every graph $G \in \mathcal{G}$ $\chi(G) <= f(\omega(G))$. The function f is called a χ-binding function for $\mathcal{G}$.

Definition 4 - a (T^k_d, r) tree: For given integers $k >= 1, d >= 2$ a (T^k_d, r) tree is a tree in which the root r has degree d, every vertex has degree either d or 1 and every path from root to a leaf has length k.

Definition 5 - a level-stable copy: Let φ be a path-induced copy of (T^k_d, r) in a graph G for some given integers $k >= 1, d >= 2$. We say that φ is level-stable if for every $i <= k$, vertices at distance i from the root r create a stable set.

Definition 6 - a type-uniform copy: Let φ be a path-induced copy of (T^k_d, r) in a graph G for some given integers $k >= 1, d >= 2$. We say that two vertices $u, v \in \varphi((T^k_d, r))$ are incomparable when neither of them is an ancestor of one another. Let w be the common ancestor of u and v that is the furthest from the root. We denote $d(x, y)$ as distance between two given vertices in the path-induced tree. We define type of a pair (u, v) as the triple (a, b, c) where $a = d(u, r), b = d(v, r)$ and $c = d(w, r)$. We call φ a type-uniform copy of (T^k_d, r) if for every two pairs of incomparable vertices (u, v) and (u', v') having the same type, the pairs are either both adjacent or both non-adjacent.

Structure of the proof

Lemma 1: For all integers $a, c >= 0$ if G is a graph with $\chi(G) > ac$ and X is a c-creature of G with $\chi(G[X]) <= a$, then $\chi(G \setminus X) = \chi(G)$.

Lemma 2: For all integers $k >= 1, d >= 2$ and $\tau >= 0$ there exist an integer K with following property: let G be a graph in which for every vertex $v \in G$ $\chi(N(v)) < \tau$. Then for every $v \in V(G)$ either G admits a path-induced copy φ of (T^k_d, r) with $\varphi(r) = v$, or there exists a $(1+d+d^2+...+d^k)$-creature X of G with $v \in X$ such that $\chi(X) <= K$.

Main result: For every rooted tree (T, r) and every integer $t >= 1$, if G is a graph with no clique of size t and does not contain a path-induced copy of (T, r), then its chromatic number is bounded.

Strengthenings

Strenthening 1: For all $k, t >= 1$ and $d >= 2$, if G is a graph with no clique of size t and with sufficiently large chromatic number, then G admits a path-induced, level-stable, type-uniform copy of (T_d^k, r).

Strenthening 2: For every tree T, there is a polynomial $f(t)$ such that for every integer $t >= 1$, if G has no induced subgraph isomorphic to T and no subgraph isomorphic to K_t, t, then G has average degree at most $f(t)$.

Strenthening 3: If G does not contain the five-vertex path P_5 as an induced subgraph, and has clique number t, then $\chi(G) <= t^{log_2 t}$

Antonina Frąckowiak
antoninamariafrackowiak@gmail.com
Presented paper by Vjekoslav Kovač

Monochromatic Boxes of Unit Volume

(https://kam.mff.cuni.cz/ spring/media/papers/4/2309.09973.pdf)

Introduction

The paper I will present negatively answers the question if for every finite colouring of the Eucledian space there exists colour class that contains vertices of a rectangle in every given area. Interestingly, this statement is correct if we consider triangles instead of rectangles. Paper's author provide an example of colouring that does not contain a rectangle of area 1 and also a more generic example of $\mathbb{R}^n$ colouring that does not contain monochromatic boxes of unit volume.

Two-dimensional case

Theorem 1 *It is possible to partition $\mathbb{R}^2$ into 25 color classes such that none of them contains the vertices of a rectangle of area 2.*

$\mathbb{R}^2$ is treated as $\mathbb{C}$ and color classes in proposed colouring are defined as

$$\ell_{j,k} := \{z \in \mathbb{C} : z^2 \in \frac{10}{3}(\mathbb{Z} + i\mathbb{Z} + \frac{j+ik}{5}) + [0, \frac{1}{5}) + i[0, \frac{1}{5}))\}.$$

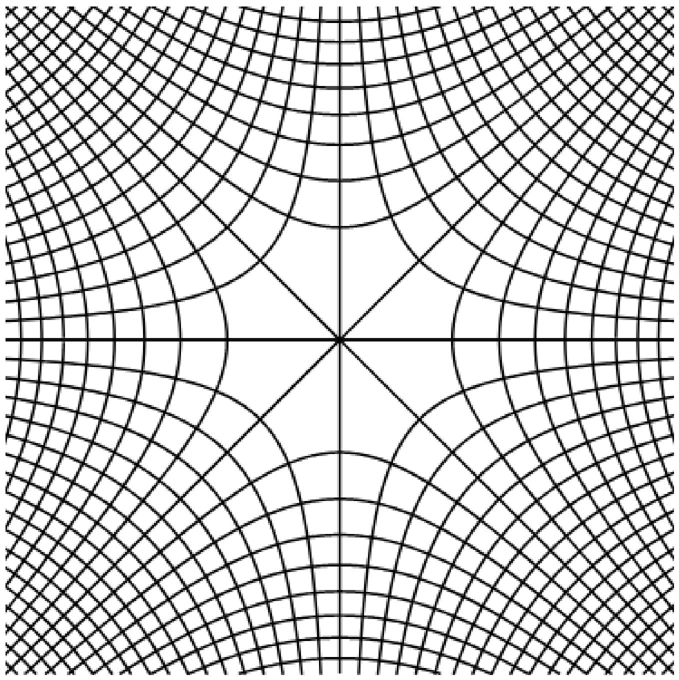


Figure 3: Boundaries of $\ell_{l,k}$ colouring

Theorem 2 *For every integer $n \geq 2$ there exists a finite colouring of the Eucledian space $\mathbb{R}^n$ such that there is no rectangular box of volume 1 with all 2^n vertices coloured the same.*

First, we partition $\mathbb{R}^n$ into sets

$$\ell_l := \{(x_1, x_2, ..., x_n) \in \mathbb{R}^n : x_1 x_2 ... x_n \in \frac{3}{2}(\mathbb{Z} + [\frac{l}{3 \cdot 2^n}, \frac{l+1}{3 \cdot 2^n}))\}$$

for $0 \leq l \leq 3 \cdot s^n - 1$. Then, we find finite subcover of $SO(n)$:

$$\{U_1\mathcal{O}, U_2\mathcal{O}, ..., U_m\mathcal{O}\}$$

where $U_i \in SO(n)$ and $\mathcal{O} = \{V \in SO(n) : \|V - I\| < \frac{1}{2^{n+2}n!}\}$. Finally, we have color classes of the desired colouring given by

$$\ell_{l_1, l_2, ... l_m} := (U_1\ell_1) \cap (U_2\ell_2) \cap ... \cap (U_m\ell_m)$$

where $l_1, l_2, ..., l_m$ run over all m-tuples of elements from $\{1, 2, ..., 3 \cdot 2^n - 1\}$.

Vojtěch Gadurek
vojtech@gadurek.cz

Simple Set Sketching

Introduction

Imagine we have two instances of one database and we would like to ensure they are the same. If they are not identical, we aim to correct the differences. The obvious way is to send all data from one instance to the other. This may be very wasteful, as with the right algorithm we may send just the data in the size of the symmetric difference of these two instances. In this talk, we will explore one, at first glance, somewhat magical algorithm and explain why it is not that surprising that it works.

Background

Exercise 1 *You are given a set of n integers. You may then choose any number of integers as your recovery key. The enemy comes and picks any integer from the set and removes it. Find a deterministic algorithm that recovers the lost integer and minimizes the memory used.*

Sometimes, we are not able to store all data given to us in memory; thus, we introduce a concept of **Data Stream**.

We may imagine a Data Stream as a vector v of unknown size holding some data and a counter i. We have two possible actions: get the value of v_i or increase the value of the counter. So when we set i to $x+1$, we may never again get the value of v_x.

Exercise 2 *You are given a Data Stream containing some numbers. Every number but one is contained twice in the Data Stream. Find a deterministic algorithm that recovers the lost integer and minimizes the memory used.*

Definition 3 *A family F of hashing functions from $X \to Y$, where $|Y| = n$, is **c-universal** if*

$$\forall (x, y \in X) \left(P\left(f(x) = f(y)\right) \leq \frac{c}{n} \right),$$

where f is chosen uniformly randomly from F and $x \neq y$.

Dictionaries may be implemented using set buckets and one hashing function from a universal hashing family that assigns every key to some bucket. This may be helpful in the next exercise.

Exercise 4 *You are given a Data Stream containing some numbers. Every number but c is contained twice (these we call singles) in the Data Stream. Find an algorithm that returns as many singles as possible in as little memory as possible. Memory used should not depend on the size of the Data Stream.*

Definition 5 ***Hypergraph*** *is a tuple (V, E), where V is a set of vertices and E is a set of subsets of V (these are called edges). We may imagine a hypergraph as a generalization of a graph, where edges may connect multiple vertices.*

Definition 6 ***k-core*** *is the largest subgraph H of graph G such that all vertices have a degree of at least k.*

Exercise 7 *You have a random hypergraph $G := (V, E)$ with edges of size 3, $|E| = n$. Determine the size of V such that G does not contain a 2-core with high probability.*

Exercise 8 *There is dice with n sides. You throw the dice n times; then, we count the side that has fallen the most, which is $O(n)$ with high probability.*

Definition 9 *A family F of hashing functions from $X \to Y$, where $|Y| = n$, is **(c,k)-independent** if*

$$\forall (x_1, \dots, x_k \in X, y_1, \dots, y_k \in Y) \left(P\left(f(x_1) = y_1 \wedge \dots \wedge f(x_k) = y_k \right) \leq \frac{c}{n^k} \right),$$

where f is chosen uniformly randomly from F and all x_i are distinct.

Now, you know nearly enough to build your own algorithm finding the symmetric difference in small memory with high probability.

Bibliography

[1] Simple Set Sketching, *arXiv:2211.03683.* Jakob Bæk Tejs Houen, Rasmus Pagh, Stefan Walzer, 2023.

Karolína Hylasová
khylas@kma.zcu.cz
Presented paper by Yangyang Cheng, Peter Keevash

On the length of directed paths in digraphs

(http://https://arxiv.org/pdf/2402.16776.pdf)

Introduction

The Caccetta-Haggkvist conjecture [1] states that any digraph on n vertices with minimum out-degree δ contains a directed cycle of length at most $\lceil n/\delta \rceil$. A stronger conjecture proposed by Thomassé [3, 2] states that any digraph with minimum out-degree δ and girth g contains a directed path of length $\delta(g-1)$ which was later disproved for every even $g \geq 4$ thanks to counterexamples given by Bai and Manoussakis.

Conjecture 1 *Any oriented graph with minimum out-degree δ contains a directed path of length 2δ.*

First we construct counterexamples to Thomassé's conjecture for every $g \geq 4$.

Proposition 2 *For every $g \geq 2$ and $\delta \geq 1$ there exists a digraph D with girth g and $\delta^+(D) \geq \delta$ such that any directed path has length at most $\frac{g\delta}{2}$ if g is even or $\frac{(g+1)\delta}{2}$ if g is odd.*

When g is large we can find a directed path of length close to 2δ.

Theorem 3 *Every digraph D with girth g and $\delta^+(D) \geq \delta$ contains a directed path of length $2\delta(1 - \frac{1}{g})$.*

For $g = 3$ or $g = 4$ we have better bounds.

Theorem 4 *Every oriented graph D with $\delta^+(D) \geq \delta$ contains a directed path of length 1.5δ. Every digraph D with $\delta^+(D) \geq \delta$ and girth $g \geq 4$ contains a directed path of length 1.6535δ.*

A digraph is called (C, d)-*regular* if $d^+(v) \geq d$ and $d^-(v) \leq Cd$ for each vertex v.

Theorem 5 *For every $C > 0$ there exists $c > 0$ such that if D is a (C, d)-regular digraph with girth g then D contains a directed path of length at least $cdg/\log d$.*

Construction

Suppose that D is a digraph with $d^+(v) = \delta$ for every $v \in V(D)$ and for each $k \geq 1$ we define the *k-lift* operation on some fixed vertex v : delete all arcs with tail v, add $k-1$ disjoint sets of δ new vertices $U_{v,1}, \ldots, U_{v,k-1}$ to D, write $U_{v,0} := \{v\}, U_{v,k} := N^+(v)$ and add arcs so that $U_{v,i-1}$ is completely directed to $U_{v,i}$ for $1 \leq i \leq k$.

We construct $D_{a,b} := \vec{K}_{\delta+1}^{\uparrow}$ for some integer $1 \leq a \leq b$, which means that we start with complete distected graph on $\delta + 1$ vertices $\vec{K}_{\delta+1}$ and we a-lift some vertex v_1 and b-lift all other vertices.

Claim 6 *The girth of $D_{a,b}$ is $a + b$ and the longest path has length δb.*

The key lemma

The following key lemma can be used together with known results on Caccetta-Häggkvist conjecture to prove Theorems 3 and 4.

Lemma 7 *If D is an oriented graph with $\delta^+(D) \geq \delta$ then D either contains a directed path of length 2δ or an induced subgraph S such that $|V(S)| \leq \delta$ and $\delta^+ \geq 2\delta - \ell(D)$.*

Theorem 8 ([5]) *Every digraph D with order n and $\delta^+(D) \geq \delta$ contains a directed cycle of length at most $\lceil \frac{2n}{\delta+1} \rceil$.*

Theorem 9 ([6]) *Every oriented graph with order n and minimum out-degree $0.3465n$ contains a directed triangle.*

Proof of the key lemma

Claim 10 *D does not contain two disjoint directed cycles of length at least $\delta + 1$.*

Claim 11 *Every vertex in $N^+(v_{a-1})$ must be on P.*

Claim 12 *$N^+(B^-) \subseteq V(C)$.*

Long directed paths in almost-regular digraphs

To prove Theorem 5 we will use following probabilistic tools [7]:

- Chernoff's inequality

  Lemma 13 *Let $X_1, \ldots, X_n$ be independent Bernoulli random variables with $\mathbb{P}[X_i = 1] = p_i$ and $\mathbb{P}[X_i = 0] = 1 - p_i$ for all $i \in [n]$. Let $X = \sum_{i=1}^n X_i$ and $E[X] = \mu$. Then for every $0 < a < 1$, we have*
  $$\mathbb{P}[|X - \mu| \geq a\mu] \leq 2^{-a^2\mu/3}.$$

- Lovász Local Lemma

  Lemma 14 *Let $A_1, \ldots, A_n$ be a collection of events in some probability space. Suppose that each $\mathbb{P}[A_i] \leq p$ and each A_i is mutually independent of a set of all other events A_j but at most d, where $ep(d+1) < 1$. Then $\mathbb{P}[\bigcap_{i=1}^n \overline{A_i}] > 0$.*

- partitioning lemma

  Lemma 15 *For every $C > 0$ there exists $c > 0$ such that for every positive integer d with $t := \lfloor cd/\log d \rfloor \geq 1$, for every (C, d)-regular digraph D there exists a partition of $V(D)$ into $V_1 \cup \cdots \cup V_t$ such that $||V_i| - |V_j|| \leq 1$ and $d^+(v, V_j) \geq \frac{\log d}{2c}$ for each $i, j \in [n]$ and $v \in V_i$.*

Concluding remarks

Conjecture 16 *There is some $c > 0$ such that $\ell(D) \geq cg(D)\delta^+(D)$ for any digraph D.*

Bibliography

[1] Louis Caccetta and Roland Haggkvist. *On minimal digraphs with given girth.* Department of Combinatorics and Optimization, University of Waterloo, 1978.

[2] B. D. Sullivan. A summary of problems and results related to the Caccetta-Häggkvist conjecture, `arXiv:math/0605646`, 2006

[3] J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications.* Springer-Verlag, London, 2008.

[4] J.A. Bondy and U.S.R. Murty. *Graph Theory.* Springer, Berlin, 2008.

[5] V. Chvátal and E. Szemerédi. Short cycles in directed graphs. *Journal of Combinatorial Theory, Series B*, 35(3):323-327, 1983.

[6] J. Hladký, D. Král, and S. Norin. Counting flags in triangle-free digraphs. *Combinatorica*, 3(1):49-76, 2017.

[7] N. Alon and J.H. Spencer. *The Probabilistic Method.* John Wiley&Sons, Inc., 1992.

Igor Januszkiewicz
igor.januszkiewicz.stud@pw.edu.pl
Presented paper by Torsten Mütze

A Book Proof of the Middle Levels Theorem

(https://link.springer.com/article/10.1007/s00493-023-00070-3)

Introduction

The middle levels conjecture had been an open problem for about 30 years until the first proof by Torsten Mütze in 2016. It was long and very technical, but in 2023 the original author of the proof published another one, this time much shorter and from 'the book'. In my talk I will present this proof.

Definitions and the theorem

Definition 1 (Hypercube) *The n-dimensional hypercube Q_n is the graph that has as vertices all bitstrings of length n and edge between any two bitstrings that differ in a single bit. The weight of a vertex x is number of 1s in x.*

Definition 2 *The kth level of Q_n is the set of vertices with weight k.*

Theorem 3 (The middle levels theorem) *For all $n \geq 1$, the subgraph of Q_{2n+1} induced by levels n and $n+1$ has a Hamilton cycle.*

Definition 4 (Dyck word) *Dyck word is a bitstring with the same number of 1s and 0s, in which every prefix contains at least as many 1s as 0s. We define D_n as the set of all Dyck words of length n and $D := \bigcup_{n\geq 0} D_n$.*

Definition 5 (Ordered three) *Ordered three is a rooted three in which the set of children of each vertex is assigned a total order.*

Lemma 6 *Every Dyck word of length $2n$ can be identified by a ordered rooted three with n edges.*

Definition 7 (Tree Rotation) *Given $x = 1u0v \in D_n$ we define a function $\rho(x) = u1v0$. We refer to the function as three rotation.*

Lemma 8 *There is a bijection between M_n and triples $\langle x, b, s\rangle$ where $x \in D_n, b \in \{0, 1\}, s \in \{0, ..., 2n\}$,*

Lemma 9 *Function f with formula: $f(\langle x, 0, s\rangle) = \langle \rho(x), 1, s+1\rangle$ and $f(\langle x, 1, s\rangle) = \langle x, 0, s\rangle$ changes only a single bit and is a bijection.*

Definition 10 (Pullable tree) *We call an ordered rooted tree $x \in D_n$ pullable if $x = 110u0v, u, v \in D$. We define $p(x) = 101u0v$ and call it pull operation.*

Lemma 11 *Any ordered rooted tree $x \in D_n$ can be transformed to the star $(10)^n$ via a sequence of tree rotations and/or pulls.*

Notations

D_n	- Set of all Dyck words of length $2n$
D	- $\bigcup_{n\geq 0} D_n$
Q_n	- n-dimensional hypercube
A_n	- Set of vertices in level n of Q_{2n+1}
B_n	- Set of vertices in level $n+1$ of Q_{2n+1}
M_n	- Subgraph of Q_{2n+1} induced by $A_n \cup B_n$
$\sigma^s(x)$	- Cyclic right rotation of a bitstring x, by s steps
$\langle x, b, s\rangle$	- $\sigma^s(xb)$, where $x \in D_n, b \in \{0,1\}, s \in \{0, ..., 2n\}$
f	- $f(\langle x, 0, s\rangle) = \langle \rho(x), 1, s+1\rangle$ and $f(\langle x, 1, s\rangle) = \langle x, 0, s\rangle$
$C(y)$	- $(y, f(y), f^2y, ...)$
F_n	- $\{C(y) \vert y \in A_n \cup B_n\}$
p	- Pull operation

Volodymyr Kuznietsov
kuzvladim7@gmail.com

Presented paper by Matt DeVos, Jessica McDonald, Kathryn Nurse

Another proof of Seymour's 6-flow theorem

(https://kam.mff.cuni.cz/ spring/media/papers/4/2302.08625.pdf)

Abstract

In 1981 Seymour proved his famous 6-flow theorem asserting that every 2-edge-connected graph has a nowhere-zero flow in the group $\mathbb{Z}_2 \times \mathbb{Z}_3$ (in fact, he offers two proofs of this result). In this note we give a new short proof of a generalization of this theorem where $\mathbb{Z}_2 \times \mathbb{Z}_3$-valued functions are found subject to certain boundary constraints.

Theorem 1 *Every 2-edge-connected digraph has a nowhere-zero $\mathbb{Z}_6$-flow*

Theorem 2 *Let $G = (V, E)$ be a connected digraph and let $T \subseteq U \subseteq V$ have $|T|$ even and $|U| \neq 1$. Assume further that every $\emptyset \neq V' \subset V$ with $V' \cap U = \emptyset$ satisfies $d(V') \geq 2$. Then for $k = 2, 3$ there exist functions $\varphi_k : E(G) \to \mathbb{Z}_k$ satisfying the following properies:*

- $(\varphi_2(e), \varphi_3(e)) \neq (0, 0)$ *for every* $e \in E$
- *supp*$(\partial\varphi_2) = T$, *and*
- *supp*$(\partial\varphi_3) = U$.

Used notations:

- We define $\delta^+(v)$, $\delta^-(v)$ to be the set of edges with tail (head) v. Let Γ be an abelian group written additevely and let $\varphi :\to \Gamma$. The *boundary* of φ is the function $\partial\varphi : V \to \Gamma$ given by the rule:

$$\partial\varphi(v) = \textstyle\sum_{e\in\delta^+(v)} \varphi(e) - \sum_{e\in\delta^-(v)} \varphi(e)$$

- We say that φ is *nowhere-zero* if $0 \notin \varphi(E)$
- supp(f) $= \{x \in X : f(x) \neq 0\}$ if X is a domain of f.
- For a graph G and a set $X \subseteq V(G)$ we use $d(X)$ to denote the number of edges with exactly one end in X.

Bibliography

[1] F. Jaeger, N. Linial, C. Payan, M. Tarsi, *Group connectivity of graphs-A nonhomogeneous analogue of nowhere-zero flow properties*, J. Combin. Theory Ser. B 56 (1992), no.2, 165-182.

[2] P.D. Seymour, *Nowhere-zero* 6*-flows*, J. Combin. Theory Ser. B 30 (1981), 130-135.

[3] W.T. Tutte, *A contribution to the theory of chromatic polynomials*, Canad. J. Math. 6 (1954), 80-91.

Cyril Kotecký
c.kotecky@gmail.com

Geometry of interval linear systems

Introduction

The sets of so-called weak solutions of interval linear systems of equalities and inequalities are unions of exponentially many convex polyhedra. As such, they are generally non-convex, but they take the form of a convex polyhedron in each orthant. This makes many problems concerning these sets NP or co-NP hard, but on the other hand, it gives them a constrained form that is interesting to study.

We will cover some geometric properties of these solution sets, mostly focusing on characterizing convexity, and describing the convex hull and its properties, such as pointedness. In some cases, we can even give a simple description of the convex hull, but it still remains hard to calculate its exact form.

Background

Definition 1

- *A convex polyhedron is* ***pointed****, if it does not contain a line.*
- *The* ***recession*** *or* ***characteristic*** *cone of a set S is the cone of all unbounded directions of S, that is*

$$R := \{r \in \mathbb{R}^n |\ \exists y \in S\ \forall \rho \geq 0 :\ y + \rho r \in P\}.$$

Theorem 2 ([1] Minkowski-Weyl)

$$P \text{ is a convex polyhedron } \iff P = Q + R \text{ where:}$$

1. *Q is the convex hull of arbitrary representative points, one from each minimal face of P*
2. *R is the recession cone of P*

Definition 3 (Interval systems)

- *An* ***interval matrix*** *is $\boldsymbol{A} := [\underline{A}, \overline{A}] = \{A \in \mathbb{R}^{m\times n} |\ \underline{A} \leq A \leq \overline{A}\}$ for $\underline{A}, \overline{A} \in \mathbb{R}^{m\times n}$, $\underline{A} \leq \overline{A}$.*
- *Its* ***center*** *and* ***radius*** *are $A_c := {}^{\underline{A}+\overline{A}}/_2$ and $A_\Delta := {}^{\overline{A}-\underline{A}}/_2$.*
- *The* ***set of interval matrices*** *of dimensions $m \times n$ is denoted $\mathbb{IR}^{m\times n}$.*
- *The* ***corner matrices*** *are $A_{es} := A_c - A_\Delta D_s$ for $s \in \{\pm 1\}^n$, where D_s denotes the diagonal matrix of s.*
- *An* ***interval linear system of equalities*** *is the system $\boldsymbol{A}x = \boldsymbol{b}$ for $\boldsymbol{A} \in \mathbb{IR}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{IR}^m$.*
  - *Its* ***set of (weak) solutions*** *is the set $\Sigma^= := \{x \in \mathbb{R}^n |\ \exists A \in \boldsymbol{A}\ \exists b \in \boldsymbol{b} :\ Ax = b\}$.*
  - *The* ***set of strong solutions*** *is the set $\Sigma^=_S := \{x \in \mathbb{R}^n |\ \forall A \in \boldsymbol{A}\ \forall b \in \boldsymbol{b} :\ Ax = b\}$.*

- *Equivalently, an* ***interval linear system of inequalities*** *is the system* $\boldsymbol{A}x \leq \boldsymbol{b}$*.*
  - *The* ***set of (weak) solutions*** *is* $\Sigma := \{x \in \mathbb{R}^n |\ \exists A \in \boldsymbol{A}\ \exists b \in \boldsymbol{b} :\ Ax \leq b\}$*.*
  - *The* ***set of strong solutions*** *is* $\Sigma_S := \{x \in \mathbb{R}^n |\ \forall A \in \boldsymbol{A}\ \forall b \in \boldsymbol{b} :\ Ax \leq b\}$*.*
- *The matrix* $\boldsymbol{A}$ *is* ***invertible****, if all the matrices contained are invertible. We define properties such as being* ***singular*** *or having* ***full column rank*** *analogously.*
- *An* ***orthant*** *of signature* $s \in \{\pm 1\}^n$ *is* $\mathbb{R}^n_s := \{x \in \mathbb{R}^n |\ D_s x \geq 0\}$*.*

Solution sets

Theorem 4 ([2] Oettli-Prager)

$$\Sigma^= = \{x \in \mathbb{R}^n |\ |A_c x - b_c| \leq A_\Delta |x| + b_\Delta\}$$

Theorem 5 ([3] Gerlach)

$$\Sigma = \{x \in \mathbb{R}^n |\ A_c x \leq A_\Delta |x| + \overline{b}\}$$

Corollary 6

$\Sigma^=$ *and* Σ *are convex in each orthant, and they are unions of up to* 2^n *convex polyhedra.*

Definition 7 (Notation)

- *Inequalities:*
  - $P_s := \{x \in \mathbb{R}^n |\ A_{es} x \leq \overline{b}\}$
  - $R_s := \{x \in \mathbb{R}^n |\ A_{es} x \leq 0\}$
  - $\Sigma_R := \{r \in \mathbb{R}^n :\ \exists x \in \Sigma\ \forall 0 \leq \rho \in \mathbb{R} :\ x + \rho r \in \Sigma\}$
- *Equalities:*
  - $P^=_s := \{x \in \mathbb{R}^n |\ A_{es} x \leq \overline{b},\ -A_{-es} x \leq -\underline{b}\}$
  - $R^=_s := \{x \in \mathbb{R}^n |\ A_{es} x \leq 0,\ -A_{-es} x \leq 0\}$
  - $\Sigma^=_R := \{r \in \mathbb{R}^n |\ \exists x \in \Sigma\ \forall 0 \leq \rho \in \mathbb{R} :\ x + \rho r \in \Sigma\}$

Corollary 8

- *Inequalities*
  1. $\Sigma = \bigcup_{s \in \{\pm 1\}^n} P_s = \bigcup_{s \in \{\pm 1\}^n} (P_s \cap \mathbb{R}^n_s)$
  2. $\Sigma_R = \bigcup_{s \in \pm 1\}^n} R_s = \bigcup_{s \in \{\pm 1\}^n} (R_s \cap \mathbb{R}^n_s)$
  3. R_s *is the recession cone of* P_s
  4. Σ_R *is the recession cone of* Σ*.*

- *Equalities*
    1. $\Sigma^{=} = \bigcup_{s\in\{\pm1\}^n} P_s^{=} = \bigcup_{s\in\{\pm1\}^n} (P_s^{=} \cap \mathbb{R}_s^n)$
    2. $\Sigma_R^{=} = \bigcup_{s\in\{\pm1\}^n} R_s^{=} = \bigcup_{s\in\{\pm1\}^n} (R_s^{=} \cap \mathbb{R}_s^n)$
    3. $R_s^{=}$ *is the recession cone of* $P_s^{=}$
    4. $\Sigma_R^{=}$ *is the recession cone of* $\Sigma^{=}$
    5. $\Sigma_R^{=} = Ker(\boldsymbol{A}) := \{x \in \mathbb{R}^n : \exists A \in \boldsymbol{A} : Ax = 0\}$
    6. $\Sigma_R^{=}$ *is centrally symmetric around the origin.*
- *The solution set* $\Sigma^{=}$ *of the system* $\boldsymbol{A}x = \boldsymbol{b}$ *is the same as the solution set* Σ *of the system* $\boldsymbol{A}x \le \boldsymbol{b},\ \boldsymbol{A}x \ge \boldsymbol{b}$.

Theorem 9 ([4] Rohn, Kreslová)

$$\Sigma_S = \{x \in \mathbb{R}^n |\ x = x_1 - x_2,\ x_1, x_2 \ge 0,\ \overline{A}x_1 - \underline{A}x_2 \le \underline{b}\} \text{ is a convex polyhedron.}$$

Square systems of equalities

Theorem 10 ([5] Jansson)

If $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$ *and* $\Sigma^{=} \neq \emptyset$*, then:*

1. *If* $\boldsymbol{A}$ *is invertible, then* $\Sigma^{=}$ *is compact and connected.*
2. *If* $\boldsymbol{A}$ *is not invertible, then each connected component of* $\Sigma^{=}$ *is unbounded.*

Theorem 11 ([6] Rohn)

Let $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$ *be invertible. Then:*

1. $\Sigma^{=}$ *has* 2^n *unique vertices* x_s*, solutions of* $A^c x - D_s A^{\Delta}|x| = b_s$ *for each* $s \in \{\pm1\}^n$.
2. $\text{conv}(\Sigma^{=}) = \text{conv}\{x_s |\ s \in \{\pm1\}^n\}$.

Convexity conditions

Theorem 12 ([6] Rohn)

For $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$ *invertible,* $\Sigma^{=}$ *is non-convex* $\iff \exists s_1, s_2 \in \Sigma$ *vertices and* $i, j \in [n]$ *such that:*

a) $A_{ij}^{\Delta} > 0$

b) $(s_1)_i = (s_2)_i$

c) $(x_{s_1})_j (x_{s_2})_j < 0$

Theorem 13 (General inequalities)

For $\boldsymbol{A} \in \mathbb{IR}^{m\times n}$*, the set* Σ *is non-convex* $\iff \exists x_1, x_2 \in \Sigma\ \exists i \in [m] :\ (A_{es}x_1)_i = \overline{b}_i < (A_{es}x_2)_i$,

$$\text{with } s \text{ is defined as } s_k := \begin{cases} \text{sgn}(x_1)_k & (x_1)_k \neq 0 \\ \text{sgn}(x_2)_k & (x_1)_k = 0 \end{cases}.$$

Square systems of inequalities

Definition 14 (Notation)

- $C = \operatorname{conv}(\Sigma)$
- $\overline{C} = \operatorname{cl}(\operatorname{conv}(\Sigma))$
- $R = \operatorname{conv}(\Sigma_R)$

Observation 15 (Properties)

- *$\overline{C}$ is a convex polyhedron while C may not be, if it is unbounded.*
- *R is the recession cone of C and $\overline{C}$*
- $\overline{C} = \mathbb{R}^n \iff R = \mathbb{R}^n$
- *Σ is bounded* $\iff \Sigma_R = \{0\} \iff R = \{0\}$
- *If $\boldsymbol{A}$ is invertible, then Σ is unbounded and connected.*

Theorem 16 (Convex hull properties)

$\overline{C}$ of $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$ invertible is:

1. *Pointed, if $\exists \boldsymbol{B} \in \mathbb{IR}^{n\times n}: \ \boldsymbol{A}^{-1} := \{A^{-1} | \ A \in \boldsymbol{A}\} \subseteq \boldsymbol{B}$ and the system $\boldsymbol{B}^T x \geq 0$ has n linearly independent strong solutions.*
2. *Equal to $\mathbb{R}^n$, if $\exists \boldsymbol{B} \in \mathbb{IR}^{n\times n}: \ \boldsymbol{A}^{-1} := \{A^{-1} | \ A \in \boldsymbol{A}\} \supseteq \boldsymbol{B}$ and the system $\boldsymbol{B}^T x \geq 0$ only has the trivial strong solution $x = 0$.*
3. *If it is pointed, then $\overline{C}$ is an offset cone with a unique vertex and n generating rays.*

Corollary 17 (Pointedness)

If $\boldsymbol{A}$ is an inverse M-matrix or its negative, then $\overline{C}$ is pointed.

Observation 18 (Non-pointedness)

If $\boldsymbol{A} \in \mathbb{IR}^{m\times n}$ contains a matrix without full column rank, then $\overline{C}$ is not pointed.

Bibliography

[1] Schrijver, A. *Theory of Linear and Integer Programming. Repr.* Wiley, Chichester. ISBN 0-471-98232-6, 1998

[2] Oettli, W. and Prager, W. *Compatibility of approximate solution of linear equations with given error bounds for coefficients and right-hand sides. Numer. Math.,*6, 405–409. doi: 10.1007/BF01386090, 1964.

[3] Gerlach, W. *Zur lösung linearer ungleichungssysteme bei störung der rechten seite und der koeffizientenmatrix. Math. Operationsforsch. Stat., Ser. Optimization, 12.* 41–43. doi: 10.1080/02331938108842705, 1981.

[4] J. Rohn and J. Kreslová. *Linear interval inequalities. Linear Multilinear Algebra*, 38(1-2):79–82, 1994.

[5] Jansson, C. *Calculation of exact bounds for the solution set of linear interval systems. Linear Algebra Appl.,* 251, 321–340, 1997.

[6] Rohn, J. *A manual of results on interval linear problems. Technical Report 1164, Institute of Computer Science*, Academy of Sciences of the Czech Republic, Prague.
`http://hdl.handle.net/11104/0212115`, 2012.

Sofiia Kotsiubynska

sofiak0423@gmail.com

Presented paper by Johannes Pardey, Dieter Rautenbach

Vertex degrees close to the average degree

(https://www.sciencedirect.com/science/article/pii/S0012365X23002856)

Abstract

The study "Vertex degrees close to the average degree", authored by Johannes Pardey and Dieter Rautenbach, delves into the investigation of how the degrees of individual vertices in a graph compare to the graph's average degree. By characterizing the minimal intervals around the average degree that must contain at least one vertex's degree, provided insights into the structural properties of graphs.

Theorem 1 provides a detailed analysis of the distribution of vertex degrees around the average degree. Specifically, it asserts that for a graph with n vertices and an average degree d, there always exists at least one vertex whose degree falls within a precisely defined interval around d. This interval is $[d - \frac{n-2}{2(n-1)}d, d + \frac{n-2}{2(n-1)}\overline{d}]$ for $d = \frac{2m}{n}$ and $\overline{d} = n - 1 - d$. Furthermore, if G has no vertex whose degree is in the interval $(d - \frac{n-2}{2(n-1)}d, d + \frac{n-2}{2(n-1)}\overline{d})$, the structure of G is very restricted: it is a disjoint union of a clique of order $d = \frac{dn}{n-1}$ and an independent set of order $d = \frac{\overline{d}n}{n-1}$ and every vertex in V_+ is adjacent to exactly half the vertices in V_-, and every vertex in V_- is adjacent to exactly half the vertices in V_+.

Theorem 2 offers a precise description, up to terms of smaller order, of the intervals around a chosen degree d_+ within which there must exist a vertex degree. If d_+ is in the interval $(\sqrt{dn}, n-1]$, then there is a vertex u in G with a degree $dG(u)$ satisfying the bounds provided by the theorem, sharpening the estimate provided by Theorem 1.

Applications and Implications

By repeatedly applying the results to a given graph and removing a vertex of degree close to the current average degree, one can identify several vertices whose degrees are confined within slowly changing intervals around the original average degree. This is useful for understanding the degree sequences of graphs and has applications in various graph problems and algorithms.

Bibliography

[1] P. Erdős, T. Gallai, Graphs with prescribed degrees of vertices (in Hungarian), Mat. Lapok 11 (1960) 264–274.

[2] S.L. Hakimi, E.F. Schmeichel, Graphs and their degree sequences: a survey, Lect. Notes Math. 642 (1978) 225–235.

[3] E. Mohr, J. Pardey, D. Rautenbach, Zero-sum copies of spanning forests in zero-sum complete graphs, Graphs Comb. 38 (2022) 132.

[4] J. Pardey, D. Rautenbach, Efficiently finding low-sum copies of spanning forests in zero-sum complete graphs via conditional expectation, Discrete Appl. Math. 328 (2023) 108–116.

Václav Lepič
vaclav@lepic.me

Presented paper by Alexandr Grebennikov, João Pedro Marciano

C_{10} has positive Turán density in the hypercube

(https://arxiv.org/abs/2402.19409v1)

Introduction

We denote the number of edges in a graph G to be $e(G)$. We denote $ex(Q_n, H)$ to be maximum number of edges in a subgraph of n-dimensional hypercube Q_n that does not contain H. We say that H has a positive Turán density in hypercube if there is a constant $\alpha > 0$ such that for every $n \in \mathbb{N}$ $ex(Q_n, H) \geq \alpha \cdot e(Q_n)$. We show that C_{10} has positive Turán density in Hypercube.

Main result

Theorem 1 *For all $n \in \mathbb{N} : ex(Q_n, C_{10}) > 0.024 \cdot e(Q_n)$*

The tools

Theorem 2 *(originally Lemma 19 in [1]) $ex(Q_n, C_{10}) > \frac{1}{3} \cdot ex^*(Q_n, C_6^-)$*

Where $ex^*(Q_n, C_6^-)$ is defined as the maximum number of edges of a subgraph of Q_n such that there is no embedding of C_6 minus one edge that can be extended to a C_6 in Q_n

Theorem 3 *For any $r, n \in \mathbb{N}$ with $r \leq n$, there exists a C_6-free induced subgraph G_r of $L_r(n)$ with $e(G_r) > \frac{c}{2} \cdot e(L_r(n))$*

Where L_r is rth edge layer of Q_n

We then combine G_r for every odd r to construct G a C_6-free subgraph of Q_n to get

$ex^*(Q_n, C_6^-) \geq e(G) > \frac{c}{4} \cdot e(Q_n)$, where $c > 0.288$

Bibliography

[1] Axenovich, M., Martin, R. & Winter, C. On graphs embeddable in a layer of a hypercube and their extremal numbers. (2023)

Matúš Matok

matus.matok@fmph.uniba.sk

Presented paper by Robert E. Tarjan, Uri Zwick

Optimal resizable arrays

(https://arxiv.org/abs/2211.11009)

Introduction

In this talk we will discuss resizable arrays. Usually, we optimise for time complexity which leaves the space complexity overlooked. The paper provides a solution which decreases the data structure's overhead to $O(rN^{1/r})$ when storing an array of size N and $O(N^{1-1/r})$ while resizing, where r is a parameter of the data structure. This comes at a cost of decreased efficiency of grow and shrink operations, but only by a constant factor of r, which makes the amortized complexity of each such operation $O(r)$. To prove that the complexity cannot be improved for any data structure only using $O(rN^{1/r})$ extra space to store an array, we will analyse a so-called *growth game*; a solitaire like single player game, which mimics the behavior of such data structures.

Definition 1 *A resizable array is an abstract data type that supports the following operations:*

1. *$A \leftarrow Array()$ - Create and return an initially empty array.*
2. *$A.Length()$ - Return the current length of the array A.*
3. *$A.Get(i)$ - Return the i-th item a_i in the array A. It is assumed that $0 \leq i < A.Length()$.*
4. *$A.Set(i, a)$ - Change the i-th item in the array A to a. It is assumed that $0 \leq i < A.Length()$.*
5. *$A.Grow(a)$ - Increase the length of array A by 1 and set the new and last item in it to a.*
6. *$A.Shrink()$ - Decrease the length of array A by 1, discarding its last item.*

Definition 2 (($s(N), t(N)$)-implementation) *Lets $s(N), t(N)$ be two non-decreasing functions. A resizable array data structure is said to be an $(s(N), t(N))$-implementation if it uses at most $N + s(N)$ space to store an array of size N and at most $N + t(N)$ space during a grow or shrink operation on an array of size N.*

Theorem 3 *Any data structure for maintaining a resizable array must at certain times use $N + \Omega(\sqrt{N})$ space where N is the current length of the array, even if only grow and access operations are performed.*

Theorem 4 *Any $(s(N), t(N))$-implementation of resizable arrays must have $s(N)t(N) \geq N$, even if only grow and access operations are supported.*

Definition 5 *Total and amortized cost Let $C_{N,k,l}$ be the minimum total cost required to play the (N, k, l)-growth game. Let $A_{N,k,l} = \frac{C_{N,k,l}}{N}$ be the corresponding amortized cost of a single grow operation.*

Definition 6 (States and their cost) *Let*

$$P_{N,k} = \left\{ a = (a_1, a_2, \ldots, a_k) \in \mathbb{N}^k \middle| \sum_{i=1}^{k} a_1 = N \text{ and } a_1 = 0 => a_{i-1} = 0, i = 2, \ldots, k \right\},$$

be the set of all states of total size N in the (N,k)-growth game. For $a \in P_{N,k}$, let $C(a) = C_k(a)$ be the minimum total cost needed to reach state $a = (a_1, a_2, /dots, a_k$, i.e., $|A_i| = a_i$, for $i \in [k]$, starting from state (0,0,..., 0). Clearly $C_{N,k} = min\{C(a) \mid a \in P_{N,k}\}$

Lemma 7 *For every $a = (a_1, a_2, \dots, a_k) \in P_{N,k}$*

- $C_k(a_1, a_2, \dots, a_k) = C_k(0, \dots, 0, a_k) + C_{k-1}(a_1, a_2, \dots, a_{k-1})$.
- $C_k(a_1, a_2, \dots, a_k) = \sum_{j=1}^{k} C_j(0, \dots, 0, a_j)$.
- $C_k(0, \dots, 0, a_k) = C_{a_k-1,k} + a_k$.
- $C(a) = N + \sum_{j=i}^{k} C_{a_j-1,j}$, *if* $0 = a_{i-1} < a_i$

Theorem 8

- *If* $\binom{n+k-1}{k} - 1 \le N \le \binom{n+k}{k} - 1$, *for some* $n \ge 0$, *then*
$$C_{N,k} = (N+1)n - \binom{n+k}{k+1}.$$
- *If* $\binom{n+k-1}{k} \le N < \binom{n+k}{k} - 1$, *for some* $n \ge 1$, *then*
$$C_{N,k} - CN - 1, k = n.$$
- *If* $\binom{n+k-1}{k} - 1 \le N < \binom{n+k}{k} - 1$, *for some* $n \ge 0$, *and* $a \in P_{N,k}$, *then*
$$C_{N,k} = C(a) \Longleftrightarrow \binom{n+i-2}{i} \le a_i \le \binom{n+i-1}{i}, \textit{for every} i \in [k].$$

Theorem 9 *Any standart resizable array data structure that uses only $N + O(rN^{1/r})$ space to store an array of size N, where $r = O(logN)$, must have an amortized cost of $\Omega(r)$ for grow operations.*

Bibliography

[1] Robert E. Tarjan, Uri Zwick: Optimal resizable arrays.
`https://arxiv.org/abs/2211.11009`

David Mikšaník
miksanik@iuuk.mff.cuni.cz

Characterization of graphs with k edge-disjoint spanning trees and its vertex analogue

Introduction

In this talk, we characterize graphs with k edge-disjoint spanning trees. This is a well-known theorem proved by Tutte [1] and Nash-Williams [2] in 1961. We will present a proof by Tomáš Kaiser [6], which directly translates into an efficient algorithm. We conclude the talk by stating two conjectures related to the theorem.

Characterization of graphs with k edge-disjoint spanning trees

Unless otherwise is stated, G is an arbitrary graph.

Definition 1 *A subgraph T of G is called a* spanning tree *if T is a tree and $V(T) = V(G)$.*

Definition 2 *For a partition $\mathcal{P}$ of $V(G)$, let $G/\mathcal{P}$ denote the multigraph obtained from G by identifying each class of $\mathcal{P}$ into single vertex and removing all loops.*

Theorem 3 ([1, 2]) *A graph G contains k pairwise edge-disjoint spanning trees if and only if, for every partition $\mathcal{P}$ of $V(G)$, the multigraph $G/\mathcal{P}$ has at least $k(|\mathcal{P}-1|)$ edges.*

Corollary 4 *Every $2k$-edge connected graph G contains k edge-disjoint spanning trees.*

Proof of Theorem 3

The proof is by Tomáš Kaiser [6]. We list here only the key definitions for the proof.

Definition 5 *A k-decomposition $\mathcal{T}$ of G is a k-tuple $(T_1, \dots, T_k)$ such that $\{E(T_i) \mid 1 \le i \le k\}$ is a partition of $E(G)$.*

Definition 6 *For a k-decomposition $\mathcal{T} = (T_1, \dots, T_k)$ of G, let $(\mathcal{P}_0, \mathcal{P}_1, \dots, \mathcal{P}_\infty)$ be the sequence of partitions of $V(G)$ defined inductively as follows*

(i) $\mathcal{P}_0 := \{V(G)\}$,

(ii) for $i \ge 0$, let c_i be the least number such that $T_{c_i}[X]$ is disconnected for some $X \in \mathcal{P}_i$ and define $\mathcal{P}_{i+1}$ to be the vertex sets of all components in $T_{c_i}[X]$, where X ranges over all the classes $X \in \mathcal{P}_i$ (if there is no such c_i, let $\infty := i$ and $c_i := k+1$).

Definition 7 *Given two k-decomposition $\mathcal{T}$ and $\mathcal{T}'$ of G, let $\mathcal{T} \prec \mathcal{T}'$ if there is some $j \ge 0$ such that*

(i) for $0 \le i < j$, $\mathcal{P}_i = \mathcal{P}'_i$ and $c_i = c'_i$,

(ii) either $\mathcal{P}_i$ refines $\mathcal{P}'_i$, or $\mathcal{P}_i = \mathcal{P}'_i$ and $c_i < c'_i$.

Conjectures

Consider the following question: Given a graph G and a parameter k, how fast can we find k pairwise edge-disjoint paths between two vertices in G? If G is $2k$-edge connected, then by Corollary 4 we can build a data structure of size $O(k|V(G)|)$ that, given two vertices $x, y \in V(G)$, outputs k pairwise edge-disjoint paths between x and y in G in time $O(k|V(G)|)$. The data structure is simply a list of k pairwise edge-disjoint spanning trees of G.

It is not hard to see that this approach does not work for k-edge connected graphs. Nevertheless, a similar approach might work for k-edge connected graphs in the following special situation: Given a special vertex $r \in V(G)$, we would like to find k pairwise edge-disjoint paths only between r and any other vertex in G. In this case, it is sufficient to have k spanning trees (not necessarily edge-disjoint) such that, for every $v \in V(G)$, the paths between r and any other vertex in the spanning trees are pairwise edge-disjoint. It is conjectured that k-edge connectivity guarantees the existence of such k spanning trees. To state the conjecture precisely, we need one definition.

Definition 8 *Given a tree T, the unique path between vertices u and v in T is denoted by $T[u, v]$.*

Conjecture 9 ([3]) *For every $\underline{\text{k-edge connected}}$ graph G and vertex $r \in V(G)$, the graph G contains k spanning trees $T_1, \ldots, T_k$ (not necessarily edge-disjoint) such that, for every $v \in V(G)$, the paths $T_1[r, v], \ldots, T_k[r, v]$ are pairwise $\underline{\text{edge}}$ disjoint.*

We can ask for paths to be vertex disjoint:

Conjecture 10 ([3]) *For every $\underline{\text{k-vertex connected}}$ graph G and vertex $r \in V(G)$, the graph G contains k spanning trees $T_1, \ldots, T_k$ (not necessarily edge-disjoint) such that, for every $v \in V(G)$, the paths $T_1[r, v], \ldots, T_k[r, v]$ are pairwise $\underline{\text{vertex}}$ disjoint.*

Both conjecture are true for $k = 2, 3, 4$ (cf. [3, 4, 5]).

Bibliography

[1] William Thomas Tutte: *On the problem of decomposing a graph into n connected factors.* Journal of the London Mathematical Society, Vol. s1-36, pp. 221-230 (1961)

[2] Crispin St John Alvah Nash-Williams: *Edge-Disjoint Spanning Trees of Finite Graphs.* Journal of the London Mathematical Society, Vol. s1-36,pp. 45-50 (1961)

[3] Alon Itai and Michael Rodeh: *The Multi-Tree Approach to Reliability in Distributed Networks.* Information and Computation, Vol. 79, pp. 43-59 (1988)

[4] Alon Itai and Avram Zehavi: *Three Tree-Paths.* Journal of Graph Theory, Vol. 3, pp. 175-188 (1989)

[5] Sean Curran, Orlando Lee, and Xingxing Yu: *Chain Decompositions of 4-Connected Graphs.* SIAM Journal on Discrete Mathematics, Vol. 19, pp. 848-880 (2005)

[6] Tomáš Kaiser: *A Short Proof of The Tree-packing Theorem.* Discrete Mathematics, Vol. 312, pp. 1689-1691 (2012)

Jędrzej Olkowski

jo417777@students.mimuw.edu.pl

Presented paper by D.Dorfman, H.Kaplan, R.Tarjan, M.Thorup, U.Zwick

Minimum-cost paths for electric cars

(https://arxiv.org/abs/2403.16936)

Introduction

We consider an electric car, which travels on a road network, consisting of a junctions V and arcs $A \subseteq V \times V$. In every junction v, there is a charging station, where unit of energy cost $r(v)$. For every arc $(u, v) \in A$, we know the cost $r(u, v)$ of traversing this arc. The cost might be negative, which might correspond to a downhill segment. At every moment, the charging level of a car needs to be within $[0, B]$. We show the algorithm for finding minimal cost of traversing between any two pairs of junctions. It turns out that we can reduce this problem to two subtasks. First one is calculation of a minimal cost between any two junctions if no chargings are allowed and second one is standard APSP (all pairs shortest paths).

Definitions

Let us define the following values to describe various distances within a road network:

1. $\rho_B(s, t)$ is a minimum cost of a travel from s to t, starting at s with an empty battery of capacity B and ending in t.
2. $\alpha_{B,a}(s, t)$ is the Maximum Final Charge (MFC) with which the car can reach t if it starts in s with charge a, where no chargings are allowed.
3. $\beta_{B,b}(s, t)$ is the (MIC) Minimum Initial Charge needed in s in order to reach t with a charge of at least b, where no chargings are allowed.

We use shortend notation for the following problems:

- Let MCP(m, n) denote the the problem of the computing of $\rho_B(s, t)$ for every $s, t \in V$.
- Let MFC(m, n) be the problem of computing of $\alpha_{B,a}(s, t)$ for every $s, t \in V$ and for a given $0 \leq a \leq B$.
- Let MIC(m, n) be the problem of computing of $\beta_{B,b}(s, t)$ for every $s, t \in V$ and for a given $0 \leq b \leq B$.
- Let MPP(n, p) be the problem of computing a Min-Plus Product of an $n \times p$ matrix and a $p \times n$ matrix.
- Let APSP(m, n) be the problem of computing the All-Pairs Shortest Paths.

Results

Theorem 1 *MCP(m, n) can be reduced to two instances of MFC(m, n) and MIC(m, n) each, an instance of MPP$(2n, p)$, where $p \leq n$ is the number of charging stations, and an instance of APSP$(4n^2, 2n)$.*

Theorem 2 *The all-pairs version of the Minimum-Cost Plans (MCP) problem in a graph with no negative cycles can be solved in* $\mathcal{O}(\frac{n^3}{2^{c\sqrt{\log n}}} + mn)$ *time, for some* $c > 0$*. The same time bound, with a different constant* c*, applies when at most* $\Delta \leq n$ *rechargings can be used on each path.*

Theorem 3 *The all-pairs version of the Minimum-Cost Plan (MCP) problem in a graph that may contain negative cycles can be solved in* $\mathcal{O}(mn^2 + n^3)$ *time. The same time bound applies when at most* Δ *rechargings can be used on each path.*

Bibliography

[1] D.Dorfman, H.Kaplan, R.Tarjan, M.Thorup, U.Zwick: Minimum-cost paths for electric cars. `https://arxiv.org/abs/2403.16936`

Lluís Sabater Rojas

llsabater@iuuk.mff.cuni.cz

Presented paper by Kempe, D., Kleinberg, J., & Tardos, É.

Maximizing the Spread of Influence through a Social Network

(https://dl.acm.org/doi/pdf/10.1145/956750.956769)

Introduction

If we want a new product to become popular by making some people to use it (and convince the largest possible number of users), who should we focus on? The optimization problem of selecting the most influential nodes is NP-hard.

Using sub-modular functions, this paper shows that it's possible to create a greedy strategy to find a solution that is provably within 63% of optimal for several classes of models.

Definitions

Definition 1 (Influence Maximization Problem) *Given a graph $G = (V, E)$, we aim to find a seed set $S \subseteq V$ of size at most k such that it maximizes the influence spread under a specific diffusion model.*

Definition 2 (Influence) *The influence of a set of nodes A, denoted $\sigma(A)$, is the expected number of active nodes at the end of the process.*

Definition 3 (Linear Threshold Model) *Every node v is influenced by each neighbour w according to a weight $b_{v,w}$ s.t.*

$$\sum_{w \in N(v)} b_{v,w} \leq 1$$

each node v chooses a threshold $\theta_v \in [0, 1]$ uniformly at random. A node is activated if

$$\sum_{w \in \text{ active } N(v)} b_{v,w} \geq \theta_v$$

Definition 4 (Independent Cascade Model) *When node v first becomes active in step t, it is given a single chance to activate each inactive neighbour (succeeds with probability $p_{v,w}$). If v succeeds, w will become active in step $t+1$. Whether or not v succeeds, it cannot make any further attempts to activate w in subsequent rounds.*

Definition 5 (Sub-modular function) *A function that satisfies*

$$f(S \cup \{v\}) \; - \; f(S) \quad \geq \quad f(T \cup \{v\}) \; - \; f(T)$$

for all elements v and all pairs of sets $S \subseteq T$.

Theorem 6 *For a non-negative, monotone sub-modular function f, let S be a set of size k obtained by selecting elements one at a time, each time choosing an element that provides the largest marginal increase in the function value. Let S^* be a set that maximizes the value of f over all k-element sets. Then $f(S) \geq (1 - 1/e) \cdot f(S^*)$; in other words, S provides a $(1 - 1/e)$-approximation.*

Results

Theorem 7 *For an arbitrary instance of the (Independent Cascade Model / Linear Threshold Model), the resulting influence function $\sigma(\cdot)$ is sub-modular.*

Theorem 8 *The influence maximization problem is* NP*-hard for the (Independent Cascade Model / Linear Threshold Model).*

Definition 9 (Triggering Model) *Each node v independently chooses a random "triggering set" T_v according to some distribution over subsets of its neighbors. To start the process, we target a set A for initial activation. After this initial iteration, an inactive node v becomes active in step t if it has a neighbor in its chosen triggering set T_v that is active at time $t+1$.*

Fact 10 *The Independent Cascade Model and the Linear Threshold Model are special cases of the Triggering Model.*

Theorem 11 *In every instance of the Triggering Model, the influence function $\sigma(\cdot)$ is sub-modular.*

Ondřej Sladký

sladky@iuuk.mff.cuni.cz

Masked superstrings framework for representing k-mer sets

Introduction

The exponentially increasing amount of sequenced DNA data [3] has led to the development of methods which instead of raw genomic data analyse corresponding sets of k-mers, which are substrings of length k from the data. This calls for efficient representations of k-mer sets, however, although this is a combinatorial problem, the general information-theory-based approaches do not provide satisfactory results as they do not take into account the structure of the genomic data. As a result textual representations which benefit from non-independence of k-mers have emerged [4]. These aim to represent a set of k-mers K as a set of strings with the following properties:

(P1) Each k-mer from K appears in at least one string.

(P2) Each substring of length k of any of the strings is in K.

We show that, in fact, only (P1) is needed and based on this we devise a new framework for representing k-mer sets called *masked superstrings*.

The talk is based on the following papers:

Paper I [1]	Ondřej Sladký, Pavel Veselý and Karel Břinda. *Masked superstrings as a unified framework for textual k-mer set representations.* bioRxiv, 2023.
Paper II [2]	Ondřej Sladký, Pavel Veselý and Karel Břinda. *Function-Assigned Masked Superstrings as a Versatile and Compact Data Type for k-Mer Sets.* bioRxiv, 2024.

Masked Superstrings

Definition 1 *Given K a set of k-mers, a pair (S, M) where S is a superstring of all k-mers in K and M is a binary mask of the same length L is called a* masked superstring *representing K if it satisfies*

$$K = \{S_i \dots S_{i+k-1} \mid M_i = 1, i \in \{0, \dots, L-k\}\}$$

We consider computation of optimal masked superstrings maximizing a general objective function $g(S, M)$.

Theorem 2 *Given a objective g and a set of k-mers K, computing a masked superstring (S, M) representing K and maximizing $g(S, M)$ is generally NP-hard, and even if $g(S, M) = |S|$.*

We thus propose a two-step optimization protocol: we first compute a superstring and then optimize the mask. In fact, each step of this protocol is in general NP-hard, even for simple respective objective. Despite this, we provide algorithms for the individual steps and further show that this protocol yields near-optimal solutions for many objectives.

f-Masked Superstrings

We further generalize the framework to be able to seamlessly perform set operations on k-mers.

Definition 3 *For a superstring S, a mask M, and a k-mer Q, the* occurrence function $\lambda(S, M, Q) \to \{0, 1\}^*$ *is a function returning a finite binary sequence with the mask symbols of the corresponding occurrences, i.e.,*

$$\lambda(S, M, Q) := \left(M_i \,\middle|\, S_i \cdots S_{i+k-1} = Q\right) .$$

Definition 4 *We call a symmetric function $f : \{0, 1\}^* \to \{0, 1, \texttt{invalid}\}$ a k*-mer demasking function.

Definition 5 *Given K a set of k-mers, a pair (f, S, M) where f is a demasking function, S is a superstring of all k-mers in K and M is a binary mask of the same length L is called a f*-masked superstring *representing K if it satisfies*

$$K = \{Q \in \{\texttt{A}, \texttt{C}, \texttt{G}, \texttt{T}\}^k \mid f(\lambda(S, M, Q)) = 1\}.$$

We prove that if we choose a suitable demasking function f, it possible to perform any symmetric set operation on k-mers using a simple concatenation of the string and the mask.

We conclude with a remark that the masked superstring framework has many practical benefits, such as improving the compressibility of k-mer sets over the currently best approaches and enabling the design of single k-mer set indexes that require substantially less memory than the state-of-the-art.

Bibliography

[1] Ondřej Sladký, Pavel Veselý and Karel Břinda. *Masked superstrings as a unified framework for textual k-mer set representations.* bioRxiv, 2023.
`https://doi.org/10.1101/2023.02.01.526717`

[2] Ondřej Sladký, Pavel Veselý and Karel Břinda. *Function-Assigned Masked Superstrings as a Versatile and Compact Data Type for k-Mer Sets.* bioRxiv, 2024.
`https://doi.org/10.1101/2024.03.06.583483`

[3] Zachary D Stephens, Skylar Y Lee, Faraz Faghri, Roy H Campbell, Chengxiang Zhai, Miles J Efron, Ravishankar Iyer, Michael C Schatz, Saurabh Sinha and Gene E Robinson. *Big Data: Astronomical or Genomical?.* PLoS Biology, 2015.
`https://doi.org/10.1371/journal.pbio.1002195`

[4] Rayan Chikhi. *K-mer Data Structures in Sequence Bioinformatics.* HDR thesis, Institut Pasteur Ecole Doctorale “EDITE”, 2021.

Diana Švecová

diana.svecova@gmail.com

Presented paper by Maria Chudnovsky, Alex Scott, Paul Seymour, Sophie Spirkl

Detecting an Odd Hole

(https://dl.acm.org/doi/pdf/10.1145/3375720)

Introduction

A *hole* of a graph G is an induced subgraph of G that is a cycle of length at least four. Scott and Seymour [2] proved that if a graph has no odd holes, then its chromatic number is bounded by a function of its clique number. In the talk, we will present an algorithm testing for the presence of an odd hole in a graph, running in polynomial time.
There are two special subgraphs we will need.

First, let $v_0 \in V(G)$, and for $i = 1, 2, 3$, let P_i be an induced path of G between v_0 and v_i such that

- P_1, P_2, P_3 are pairwise vertex-disjoint except for v_0;
- $v_1, v_2, v_3 \neq v_0$, and at least two of P_1, P_2, P_3 have length at least two;
- v_1, v_2, v_3 are pairwise adjacent; and
- for $1 \leq i < j \leq 3$, the only edge between $V(P_i) \setminus \{v_0\}$ and $V(P_j) \setminus v_0$ is the edge v_iv_j.

We call the subgraph induced on $V(P_1 \cup P_2 \cup P_3)$ a *pyramid*, with *apex* v_0 and *base* $\{v_1, v_2, v_3\}$ (see Figure 4).

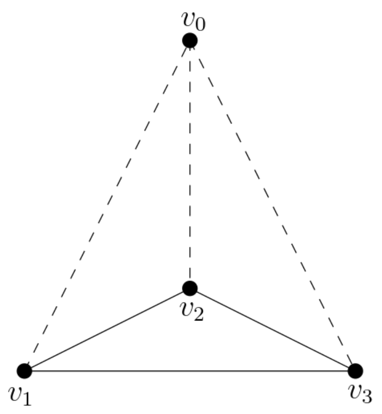


Figure 4: A pyramid

Theorem 1 *[1] There is an algorithm with the following specifications:*

Input: *A graph G.*
Output: *Determines whether there is a pyramid in G.*
Running time: *$O(|G|^9)$.*

Second, we say that $G[V(P) \cup \{v_1, \ldots, v_5\}]$ is a *jewel* in G if $v_1, \ldots, v_5$ are distinct vertices, v_1v_2, v_2v_3, v_3v_4, v_4v_5, v_5v_1 are edges, v_1v_3, v_2v_4, v_1v_4 are nonedges, and P is a path of G between v_1, v_4 such that v_2, v_3, v_5 have no neighbors in the interior of P (see Figure 5).

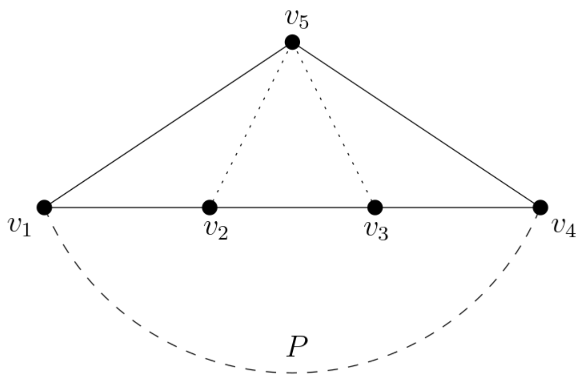


Figure 5: A jewel

Theorem 2 *[1] There is an algorithm with the following specifications:*

Input: *A graph G.*
Output: *Determines whether there is a jewel in G.*
Running time: $O(|G|^6)$.

It can be shown that if a graph G contains either a pyramid or a jewel, then G has an odd hole.

Main result

The main result of the article is the following algorithm:

Theorem 3 *There is an algorithm with the following specifications:*

Input: *A graph G.*
Output: *Determines whether G has an odd hole.*
Running time: $O(|G|^9)$.

This algorithm has three stages.

1. At first, the algorithm tests for "easily-detected configurations" (a pyramid or a jewel), which can be efficiently detected and whose presence guarantees that G contains an odd hole.

2. The second step is called "cleaning", and it is the most challenging part. Let C be a shortest odd hole of a graph G. A vertex $v \in V(G)$ is *C-major* if there is no three-vertex path of C containing all the neighbours of v in $V(C)$; and C is *clean* (in G) if no vertices of G are C-major. If G has a shortest odd hole that is clean, then it is easy to detect that G has an odd hole. In this step, the algorithm generates a "cleaning list", polynomially many subsets $X_1, \ldots, X_k$ of the vertex set of the input graph G, with the property that if G contains an odd hole, then for some $i \in \{1, \ldots, k\}$, an odd hole can be found in $G \setminus X_i$ using the method of step 3.

3. The third step is an algorithm that tries to find an odd hole directly. It would not be expected to work on a general input graph, and this is why the second step is essential.

In the paper, the authors give a new, simpler way of cleaning a shortest odd hole C in a graph G with no pyramid or jewel, which we will present in the talk.

Bibliography

[1] Maria Chudnovsky, Gérard Cornuéjols, Xinming Liu, Paul Seymour, and Kristina Vušković. 2005. Recognizing Berge graphs. *Combinatorica* 25, 2 (2005), 143–186.

[2] Alex Scott and Paul Seymour. 2016. Induced subgraphs of graphs with large chromatic number. I. Odd holes. *J. Comb. Theory, Ser. B* 121 (2016), 68-84.

Hadi Zamani

hadiz@iuuk.mff.cuni.cz

Counting Perfect Matchings, Permanent, Real Stability and Beyond

Introduction

Counting perfect matchings in a graph, even for a bipartite graph, is recognized as a #P-complete problem, equivalent to computing the permanent of the biadjacency matrix. However, what if we narrow down the class of graphs? For instance, in the case of planar graphs, the FKT algorithm operates in polynomial time, which I will elaborate on in this talk.

For the hard problem, we should be content with a good approximation, and fortunately, there exists a straightforward algorithm for this purpose. Demonstrating the effectiveness of this approximation opens a new chapter on Real Stable Polynomials and Gurvits's machinery.

To expand the application of this machinery to more general counting problems, we will explore the main reference[1].

Background

"The notion of the capacity of a polynomial was introduced by Gurvits around 2005, originally to give drastically simplified proofs of the Van der Waerden lower bound for permanents of doubly stochastic matrices and Schrijver's inequality for perfect matchings of regular bipartite graphs. Since this seminal work, the notion of capacity has been utilized to bound various combinatorial quantities and to give polynomial-time algorithms to approximate such quantities (e.g., the number of bases of a matroid). These types of results are often proven by giving bounds on how much a particular differential operator can change the capacity of a given polynomial."[2]

Definition 1 ***Real Stable Polynomials***
A multivariate polynomial p is called real stable if all coefficients are real numbers and $p\,(x_1, \ldots, x_n) \neq 0$ whenever $(x_1, \ldots, x_n) \in \mathcal{H}^n$ where $\mathcal{H} = \{y \in \mathbb{C} \mid \Im(y) > 0\}$ is the upper-half of the complex plane.

Definition 2 ***Capacity***
Given a polynomial $p \in \mathbb{R}\,[x_1, \ldots, x_n]$ with non-negative coefficients and a vector $\alpha \in \mathbb{R}^n$ with non-negative entries, we define the α-capacity of p as:

$$\operatorname{Cap}_\alpha(p) := \inf_{x>0} \frac{p(x)}{x^\alpha} = \inf_{x_1,\ldots,x_n>0} \frac{p\,(x_1, \ldots, x_n)}{x_1^{\alpha_1} \cdots x_n^{\alpha_n}}.$$

$\alpha = 1$, when there is no α.

Theorem 3 ***Gurvits's Machinery***
Given a polynomial $p \in \mathbb{R}\,[x_1, \ldots, x_n]$ with non-negative coefficients and the sum of the degrees of each term bounded by n, we have

$$\frac{\partial^n p}{\partial z_1 \ldots \partial z_n}(0, \ldots, 0) \leq \operatorname{Cap}(p) \leq \frac{n^n}{n!} \frac{\partial^n p}{\partial z_1 \ldots \partial z_n}(0, \ldots, 0).$$

Bibliography

[1] D. Straszak and N. K. Vishnoi. Real stable polynomials and matroids: Optimization and counting. In Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, pages 370–383, 2017.

[2] L. Gurvits and J. Leake. Counting matchings via capacity-preserving operators. Combinatorics, Probability and Computing, 30(6):956–981, 2021

Patrik Zavoral
patrik.zavoral@gmail.com

Presented paper by Richard Montgomery, Rajko Nenadov, Tibor Szabó

Global rigidity of random graphs in $\mathbb{R}$

(https://arxiv.org/abs/2401.10803)

Introduction

Consider a finite set of vertices V and suppose that we only know distances between some of them. In other words, consider a random graph G over V and a function $d : E(G) \to \mathbb{R}^+ \cup \{0\}$. What properties of G make it possible to uniquely reconstruct the configuration of V (up to isometry)? We will show that the minimum degree ≥ 2 is alone sufficient with high probability.

Definition 1 *Erdős-Rényi random graph process on the vertex set $[n]$ is the sequence $\{G_m\}_{m\geq 0}$, where G_0 is the empty graph and all following G_i are formed by adding a new edge uniformly at random into G_{i-1}.*

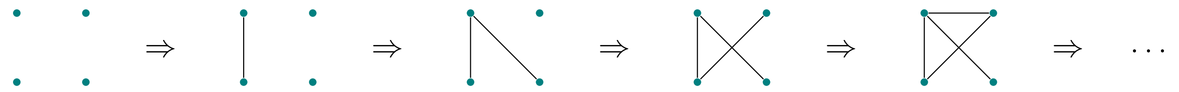


Figure 6: Erdős-Rényi random graph process for $n = 4$.

Of particular importance will be G_τ, where $\tau = \min\{m \mid \delta(G_m) \geq 2\}$. Note that a single $G_m \sim \mathrm{G}(n, m)$, where $\mathrm{G}(n, m)$ is the uniform distribution over all graphs with exactly n vertices and m edges.

Definition 2 *Let $f : V(G) \to \mathbb{R}$. We define the G-distance function of f:*

$$\begin{aligned} d_{f,G}: \quad E(G) &\longrightarrow \mathbb{R}^+ \cup \{0\} \\ ij &\longmapsto |f(i) - f(j)| \end{aligned}$$

Further, we say that a function $f : V(G) \to \mathbb{R}$ realizes $d : E(G) \to \mathbb{R}^+ \cup \{0\}$ if $d = d_{f,G}$.

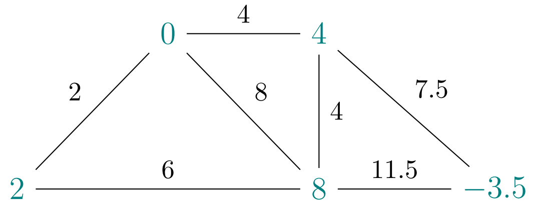


Figure 7: G-distance function and its realization.

Notice that $f_1, f_2 : V(G) \to \mathbb{R}$ are isometric if $f_1 = af_2 + b$ for some $a \in \{\pm 1\}$ and $b \in \mathbb{R}$. There can be possibly multiple non-isometric realizations of a given $d : E(G) \to \mathbb{R}^+ \cup \{0\}$ or no satisfying realization may exist at all. Given a graph G and a G-distance function $d : E(G) \to \mathbb{R}^+ \cup \{0\}$, it is possible to find the satisfying realizations $f : V(G) \to \mathbb{R}$ in non-deterministic linear time.

Definition 3 *We will call the graph G globally rigid if for every G-distance function there exists at most one injective realization up to isometry.*

Goal

Theorem 4 *Let $\{G_m\}_{m\geq 0}$ be a random graph process over $[n]$. G_τ is w.h.p. globally rigid.*

Theorem 5 $\forall \varepsilon > 0 \;\; \exists C > 0 : m \geq Cn \implies$ *w.h.p. there exists a globally rigid induced subgraph G' of $G \sim \mathrm{G}(n, m)$ with $|V(G')| \geq (1-\varepsilon)n$.*

Proof

In the talk, we will demonstrate the proof of the two theorems given in the paper. The paper builds the proof around the following two lemmas:

Lemma 6 *The graph G on the vertex set $[n]$ is globally rigid if the following properties are satisfied:*

(P1) For every disjoint $U, W \subseteq V(G)$ of size $|U|, |W| \geq \frac{n}{15}$ there is an edge between U and W.

(P2) For every $U \subseteq V(G)$ of size $\frac{n}{15} \leq |U| < n$ there exists a vertex $v \in V(G) \setminus U$ with at least two neighbors in U.

Lemma 7 $\forall \varepsilon > 0 \;\; \exists C > 0 : m \geq Cn \implies G \sim \mathrm{G}(n, m)$ *w.h.p. satisfies the following property:*

(P3) For every disjoint $X, Y \subseteq V(G)$ of size $|X|, |Y| \geq \varepsilon n$, there is an edge between X and Y.

Lemma 6 guarantees global rigidity if the graph G is in some way sufficiently connected. We will prove it by finding a correspondence between the highest-ranked vertices of f and the highest-ranked vertices f', their lowest-ranked vertices, and finally employing a type of triangulation. Lemma 7 in turn guarantees that a dense enough random graph G *w.h.p.* contains a large subgraph G' that is sufficiently connected in a related way. We will prove it holds for $C > \frac{2}{\varepsilon^2}$ by lower-bounding the probability and showing it goes to 1 as n approaches ∞. Finally, by proving these two lemmas, and by using some previous results [2][3], we will straightforwardly show that Theorems 4,5 hold.

Bibliography

[1] R. Montgomery, R. Nenadov, T. Szabó. *Global rigidity of random graphs in* $\mathbb{R}$. arXiv preprint arXiv:2401.10803, 2024.

[2] B. Bollobás. *Random graphs.*, volume 73 of Camb. Stud. Adv. Math. Cambridge: Cambridge University Press, 2nd ed. edition, 2001.

[3] A. Lew, E. Nevo, Y. Peled, and O. E. Raz. *Sharp threshold for rigidity of random graphs.* Bull. Lond. Math. Soc., 55(1):490–501, 2023.

List of participants

Jakub Balabán

Juraj Belohorec

Benjamin Benčík

Adam Beneš

Martin Černý

Petr Chmel

Fernando Cortés Kühnast

Barbora Dohnalová

Karolina Drabik

Filip Filipkowski

Antonina Frąckowiak

Vojtěch Gaďurek

Milan Hladík

Pavel Hubáček

Karolína Hylasová

Igor Januszkiewicz

Jan Jedelský

Cyril Kotecký

Sofiia Kotsiubynska

Júlia Križanová

Volodymyr Kuznietsov

Václav Lepič

Kristýna Mašková

Matúš Matok

David Mikšaník

Richard Mužík

Jędrzej Olkowski

Martin Pastyřík

David Ryzák

Lluís Sabater Rojas

Ondřej Sladký

Dominik Stejskal

Robert Šámal

Diana Švecová

Martin Tancer

Mykhaylo Tyomkyn

Filip Úradník

Pavel Veselý

Petr Vincena

Hadi Zamani

Patrik Zavoral