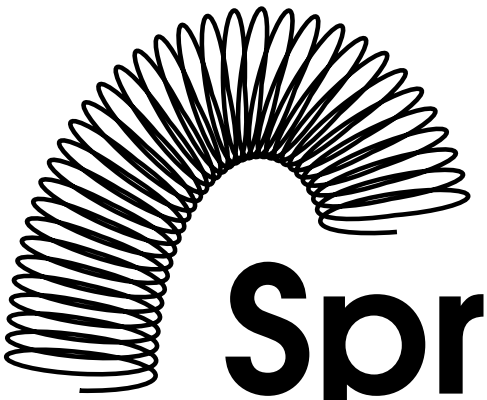


# Spring School 2016

Dušan Knop, Tomáš Masařík, Veronika Slívová (eds.)

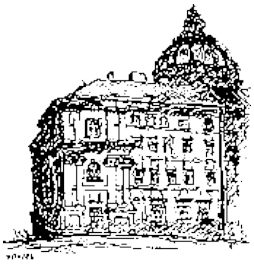
# Preface

Spring school on Combinatorics has been a traditional meeting organized for faculty and students participating in the Combinatorial Seminar at Charles University for over 30 years. 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 generously supported by Computer Science Institute (IÚUK) of Charles University and the Department of Applied Mathematics (KAM) of Charles University.

The Spring Schools are entirely organized and arranged by our students (mostly undergraduates). The lecture subjects 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 lectures 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 Nová Ves, a mountain village in Orlické hory in Northern part of Czech republic with a great variety of possibilities for outdoor activities.

Ondřej Pangrác, Robert Šámal, Martin Tancer

CSI

IÚUK

ITI


JAGUARS


SPRING
2016

# Index of Talk Titles

# Typical day

| | |
|---|---|
| 8:30 - 9:30 | Breakfast |
| 9:30 - 11:00 | Talk 1 |
| 11:00 - 12:30 | Talk 2 |
| 12:30 - 13:30 | Lunch |
| 13:30 - 18:00 | Free time |
| 18:00 - 19:00 | Dinner |
| 19:00 - 20:30 | Talk 3 |

# Schedule of talks

**Friday**

**Evening sesion** chaired by Ondřej Pangrác

Dušan Knop *Applying simultaneous diofantine approximation* [19:15–20:15]

Matas Šileikis *Half–sandwich of two random graphs* [20:15–20:45]

Tomáš Masařík *Backbone coloring* [20:45–21:00]

**Saturday** *Series: Interval methods*

**Morning sesion** chaired by Matas Šileikis

Jarda Horáček *Introduction to Interval Computations* [9:30–10:15]

Milan Hladík *Verification by Interval Approach* [10:30–11:15]

Michal Černý *Interval & Data Analysis* [11:15–12:00]

**Evening sesion** chaired by Karel Ha

Elif Garajová *Optimality in Interval Linear Programming* [18:50–19:40]

Mirek Rada *Intervals* $\rightarrow$ *Zonotopes* $\rightarrow$ *Ellipsoids* [20:00–20:45]

Pavel Klavík *Visual Complex Analysis* [21:00–22:30]

**Sunday**

**Group 1** chaired by Milan Hladík
*Interval methods — completion*

Jana Novotná *Feasibility in interval linear programming* [9:30–10:30]

Petra Pelikánová *Interval linear regression* [11:00–12:00]

**Group 2** chaired by Dušan Knop

Peter Zeman *Isomorphism of graphs of bounded valence can be tested in polynomial time* [9:30–10:30]

Honza Voborník *How Robust is the Wisdom of the Crowds?* [10:30–12:00]

**Evening sesion** chaired by Jan Voborník

Andreas Emil Feldmann *Fixed Parameter Approximations for k-Center Problems in Low Highway Dimension Graphs* [19:00–20:00]

**Monday** *Series: Nonstandard Methods in Combinatorics*

**Morning sesion** chaired by Robert Šámal

Petr Glivický *Nonstandard methods in Ramsey combinatorics I* [9:00–10:30]

Petr Glivický *Exercise: Nonstandard methods in Ramsey combinatorics* [10:30–12:00]

**Group 1** chaired by Petr Glivický

*Nonstandard Methods — completion*

Tomáš Toufar *Nonstandard methods in Ramsey combinatorics II* [18:00–19:30]

Karel Král *Nonstandard methods in Ramsey combinatorics III* [19:30–21:00]

**Group 2** chaired by Tomáš Valla

Radovan Červený *On interval representations of graphs* [18:30–20:00]

Václav Blažej *Fast Algorithms for Exact String Matching* [20:00–21:30]

**Tuesday**

**Group 1** chaired by Tomáš Toufar

Jakub Sosnovec *Every graph is (2,3)-choosable* [9:30–11:00]

Jakub Svoboda *Games on interval and permutation graph representations* [11:00–12:30]

**Group 2** chaired by Pavel Dvořák

Peter Korcsok *Drawing Planar Cubic 3-Connected Graphs with Few Segments: Algorithms and Experiments* [9:30–11:00]

Veronika Slívová *Chess: Retrograde analysis* [11:00–12:30]

**Group 1** chaired by Mark Karpilovskij

Pavel Veselý *Improved Deterministic Algorithms for Linear Programming in Low Dimensions* [19:30–21:00]

**Group 2** chaired by Robert Šámal

Karel Ha *Mastering the Game of Go with Deep Neural Networks and Tree Search* [19:30–21:00]

**Wednesday** — *trip day*

**Group 1** chaired by Jan Musílek

Martin Töpfer *Mapping planar graphs into the Coxeter graph* [19:30–21:00]

**Group 2** chaired by Tomáš Masařík

Jana Syrovátková *Iterated Prisoners Dilema* [19:30–21:00]

**Thursday**

**Group 1** chaired by Jana Novotná

Stanislav Kučera *Another exploration problem* [9:30–11:00]

Robert Lukotka *Generating Snarks* [11:00–12:00]

**Group 2** chaired by Jan Musílek

Mark Karpilovskij *The Erdős–Hajnal conjecture for paths and antipaths* [9:30–10:30]

Adam Kabela *Combinatorial proofs of addition formulas* [11:00–12:30]

**Group 1** chaired by Mark Karpilovskij

Jan Musílek *Hollow heaps* [19:30–21:00]

**Group 2** chaired by Michal Opler

Filip Mišún *Enumeration of antisymmetric monotone triangles and domino tilings of quartered Aztec rectangles* [19:30–21:00]

**Friday**

**Group 1** chaired by Robert Lukotka

Jarda Hančl *The maximum number of subset divisors of a given size* [9:30–11:00]

Pavel "Koblich" Dvořák *The diameter of Chess Grows Exponentially* [11:00–12:30]

**Group 2** chaired by Veronika Slívová

Michal Opler *Increasing paths in edge–ordered graphs: the hypercube and random graph* [9:30–10:30]

Tereza Hulcová *Independent Domination versus Weighted Independent Domination* [11:00–12:30]

# Interval Methods

Michal Černý

cernym@vse.cz

Intervals & Data Analysis (Ser: Interval Methods)

### Introduction

An interval is a natural model for a real number which is not known exactly. We can think of numbers computed by inexact numerical methods, or inexactly observed data in statistical problems. We consider the following model: let $x = (x_1, \ldots, x_n)$ be an unobservable data sample (formally, a random vector) and let $x = (x_1, \ldots, x_n)$ be a vector of observable intervals such that $x_i \in x_i$ a.s., $i = 1, \ldots, n$. We do not assume any further knowledge about the joint distribution of $(x, x)$.

Let a statistic (= continuous function) $S(\xi)$ be given. Under such weak assumptions on the distribution of $(x, x)$, the only information we can infer about $S(x)$ from the observable data $x$ is the pair of bounds $\underline{S} = \min_{\xi \in x} S(\xi)$ and $\overline{S} = \max_{\xi \in x} S(\xi)$, clearly satisfying $S(x) \in [\underline{S}, \overline{S}]$ a.s.

Although the construction of the interval $[\underline{S}, \overline{S}]$ is motivated by statistical analysis of interval data, from the optimization-theoretic viewpoint the computation of $\underline{S}, \overline{S}$ reduces to a pair of box-constrained optimization problems. Now there is an interesting question: what are the computational properties of these optimization problems for particular choices of $S$, namely for those often used in practical data analysis (such as sample mean, variance, coefficient of variation, median, other quantiles, higher moments, test statistics for various hypotheses etc.)?

### Sample variance

For sample variance $S(x) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \frac{1}{n} \sum_{j=1}^{n} x_j)^2$ many results have been proved in [2]. The computation of $\overline{S}$ is NP-hard; even worse, $\overline{S}$ is inapproximable with an arbitrary absolute error. But, conversely, it is computable pseudopolynomially. It is immediate that the computation of $\underline{S}$ is reducible to CQP and thus solvable in weakly polynomial time. It is interesting that in [2] a strongly polynomial algorithm is designed.

In our talk we show that the complexity of computation of $\overline{S}$ is related to an interesting question in the theory of interval graphs. Namely, the algorithm from [2] for computation of $\overline{S}$ has complexity $O(p(n) \cdot 2^{\omega_n})$, where $p(n)$ is a polynomial and $\omega_n$ is the size of the largest clique of the graph $G(V, E)$ with $V = \{1, \ldots, n\}$ and $E = \{\{i, j\} : |x_i^C - x_j^C| \leq \frac{1}{n},\ 1 \leq i < j \leq n\}$ (here, $x^C$ is the center of the interval $x$). In the worst case we have clearly $\omega_n = n$. But our statistical motivation leads us to the natural assumption that the intervals $x_1, \ldots, x_n$ are *random* (and then $G$ is a random graph). We assume the following model: the centers of the intervals are sampled from a distribution $\Phi$ and the radii are sampled from a distribution $\Psi$ (and the samples are independent). For a discrete distribution $\Phi$ it is easy to show $\mathsf{E}\omega_n = \Theta(n)$ by pigeonholing. But we conjecture (with O. Sokol and M. Rada) that for a certain class of continuous distributions $\Phi$ and 'reasonable' distributions $\Psi$ it holds $\mathsf{E}\omega_n = O(\log n)$ and $\operatorname{var} \omega_n = O(1)$. Then, $\overline{S}$ is computable polynomially on average. And, moreover, it is computable polynomially "in most cases" by the Chebyshev inequality. Thus we have an interesting example of an problem which is NP-hard and inapproximable in the worst case, but "efficiently" computable very often, at least in our probabilistic setup. (Recall that e.g. KNAPSACK behaves similarly: it is indeed difficult to meet a hard instance.) Moreover, the problem of computing $\overline{S}$ leads to a certain class of random interval graphs $G$ the properties of which deserve further investigation; in particular, limit properties of $G$ for $n \to \infty$ are of interest.

Similar considerations hold also for other statistics, such as the $t$-ratio (coefficient of variation) studied in [1].

## Regression

Interval methods are also useful in regression. We discuss some recent results from [3, 4]. Our setup is entirely probabilistic: we need interval-theoretic tools neither for the *formulation of the problem* nor for the *statements of results and algorithms*. The interval toolbox is a powerful technique proving that things are right, but it can be fully hidden inside proofs.

Assume the linear regression relationship $y = X\theta + \varepsilon$, where the matrix $X$ of (stochastic) regressors is unobservable. The observable data are $(Z, y)$; here, $Z$ is a contaminated form of $X$ assumed to be in the form $Z = X + \Xi$, where $\Xi$ is a matrix of random errors. This model is known as Errors-in-Variables Regression. Traditional estimation theory tells us that under certain assumptions on the distribution of $(X, \Xi, \varepsilon)$, the 'right' estimator $\widehat{\theta}$ of the unknown vector $\theta$ of regression parameters is Total Least Squares (TLS). Recall that TLS, also known as 'orthogonal regression', is a solution to the optimization problem $\min_{\widehat{\theta},\Delta Z,\Delta y} \|(\Delta Z, \Delta y)\|_F$ s.t. $(Z + \Delta Z)\widehat{\theta} = y + \Delta y$, where $\|\cdot\|_F$ stands for the Frobenius norm.

In [3, 4] we asked the question what happens when we replace the Frobenius norm by the Chebyshev norm $\|A\|_{\max} = \max_{i,j} |A_{ij}|$, obtaining the optimization problem

$$\min_{\widehat{\theta},\Delta Z,\Delta y} \|(\Delta Z, \Delta y)\|_{\max} \text{ s.t. } (Z + \Delta Z)\widehat{\theta} = y + \Delta y. \tag{1}$$

It turns out that the replacement is fruitful. First, (1) yields a consistent estimator ($\widehat{\theta} \overset{n\to\infty}{\longrightarrow} \theta$ in probability, where $n$ stands for the number of observations) under some interesting assumptions on $(X, \Xi, \varepsilon)$. Namely, it is a consistent estimator in case when all errors $(\varepsilon, \Xi)$ are uniformly bounded a.s. And, as a bonus, the bound is consistently estimable. Here we skip details of this result (important for the theory of data analysis) and briefly sketch the second one: the computation of (1) can be reduced to a family of generalized linear-fractional programming problems (GLFPs). Recall that GLFPs can be solved in polynomial time by interior-point methods.

The main idea is to use the Oettli-Prager Theorem: solving (1) is equivalent to finding the minimum $\delta$ such that the interval-valued linear system $[Z \pm \delta E]\widehat{\theta} = [y \pm e]$ is (weakly) solvable, where $E$ is the all-one matrix and $e$ is the all-one vector. The characterization given by the Oettli-Prager Theorem translates the problem to polyhedral geometry: given a certain class of convex polyhedra with a specific structure and parametrized by $\delta$, the task is to find the minimum $\delta$ such that at least one of the polyhedra is nonempty. And this question can be reduced to a family of GLFPs.

To conclude, we have a nice example of a problem motivated by a classical regression setup, where interval methods serve as a 'bridge' to a reformulation as a problem in polyhedral geometry, the solution of which then leads to interior-point methods for generalized linear-fractional programming. Except for that, we should emphasize that the polyhedral-geometric characterization of (1) given by the Oettli-Prager Theorem also plays a central role in the proof of consistency of the estimator $\widehat{\theta}$.

## References


[1] M. Černý and M. Hladík: The complexity of computation and approximation of the $t$-ratio over one-dimensional interval data. *Computational Statistics and Data Analysis* **80**, 2014, 26–43.

[2] S. Ferson, L. Ginzburg, V. Kreinovich, L. Longpré and M. Aviles: Exact bounds on sample variance of interval data. 2002. Available from: `http://citeseerx.ist.psu.edu/viewdoc/ summary?doi=10.1.1.16.2764`.

[3] M. Hladík and M. Černý: Total Least Squares and Chebyshev norm. *Procedia Computer Science* **51**, 2015, 1791–1800.

[4] M. Hladík, M. Černý and J. Antoch: Linear regression with bounded errors in data: Total "Least Squares" with Chebyshev norm. *Statistica Sinica*, submitted, 2015. Preprint: `http://nb. vse.cz/~cernym/tls.pdf`.

Elif Garajová

elif.garajova@gmail.com

# Optimality in Interval Linear Programming (Ser: Interval Methods)

## Introduction

In a linear programming problem, we usually treat coefficients as crisp real values. However, in practical applications, there is often a need to include some sort of uncertainty or inexactness in the model. In this talk, we will address the problem of determining the range of possible optimal values and the set of all optimal solutions of a linear program with coefficients perturbing independently in given intervals. A detailed discussion of known results in interval linear programming can be found in a survey by Hladík [2].

## Optimal value range

When dealing with interval linear programs (ILP), one of the following (in general non-equivalent) forms of the feasible set $\mathscr{M}(A,b)$ is usually assumed:

(A) $\mathscr{M}(A,b) = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$,

(B) $\mathscr{M}(A,b) = \{x \in \mathbb{R}^n : Ax \leq b\}$,

(C) $\mathscr{M}(A,b) = \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}$.

The first theorem by Oettli and Prager provides a useful characterization of the weak solution set in a system of interval linear equations. It is followed by an analogical characterization for systems of inequalities due to Gerlach. The corresponding proofs, as well as further results on feasibility and optimal value range in ILP, are presented in book [1].

**Theorem 1.** *Let $A \in \mathbb{IR}^{m\times n}, b \in \mathbb{IR}^m$ be given. A vector $x \in \mathbb{R}^n$ is a weak solution to the interval system $Ax = b$ if and only if it satisfies*

$$|A_c x - b_c| \leq A_\Delta |x| + b_\Delta.$$

**Theorem 2.** *Let $A \in \mathbb{IR}^{m\times n}, b \in \mathbb{IR}^m$ be given. A vector $x \in \mathbb{R}^n$ is a weak solution to the interval system $Ax \leq b$ if and only if it satisfies*

$$A_c x - A_\Delta |x| \leq \overline{b}.$$

From now on, we will consider an ILP in the form $\min c^T x$ subject to $x \in \mathscr{M}(A,b)$. An important and well-studied problem in interval linear programming is the computation of the range of optimal values based on the possible values of the coefficients:

**Definition 3.** Denote by $f(A,b,c) = \inf\{c^T x : x \in \mathscr{M}(A,b)\}$ the optimal value of a linear program. For an ILP given by the triplet $(A,b,c)$, we define the *optimal value range* as the interval $\left[\underline{f}(A,b,c), \overline{f}(A,b,c)\right]$, where

$$\underline{f}(A,b,c) = \inf\{f(A,b,c) : A \in A, b \in b, c \in c\},$$
$$\overline{f}(A,b,c) = \sup\{f(A,b,c) : A \in A, b \in b, c \in c\}.$$

**Theorem 4.** *For an ILP of type (A), we have*

$$\underline{f}(A,b,c) = \inf\{\underline{c}^T x : \underline{A}x \le \overline{b}, \overline{A}x \ge \underline{b}, x \ge 0\},$$
$$\overline{f}(A,b,c) = \sup\{f(A_c - \mathrm{diag}(p)A_\Delta, b_c + \mathrm{diag}(p)b_\Delta, \overline{c}) : p \in \{\pm 1\}^m\}.$$

**Theorem 5.** *For an ILP of type (C), we have*

$$\underline{f}(A,b,c) = \inf\{\underline{c}^T x : \underline{A}x \le \overline{b}, x \ge 0\},$$
$$\overline{f}(A,b,c) = \inf\{\overline{c}^T x : \overline{A}x \le \underline{b}, x \ge 0\}.$$

## Set of optimal solutions

Another interesting problem is finding a description and analyzing the properties of the set of all possible optimal solutions. This set is in general difficult to determine, however, characterizations for some special cases are known.

**Lemma 6.** *For an ILP of type (A), the optimal solution set can be described by the parametric interval linear system*

$$Ax = b, x \ge 0, A^T y \le c, c^T x = b^T y,$$

*with $A \in A, b \in b$ and $c \in c$.*

**Definition 7.** Let a basis $B \subseteq \{1,\dots,n\}$ be given. An ILP is said to be *B-stable*, if $B$ is an optimal basis for each scenario of the ILP. Furthermore, it is said to be *unique B-stable*, if each scenario has a unique optimal basic solution with the basis B.

**Theorem 8.** *If an ILP of type (A) is unique B-stable, then the optimal solution set can be described by the linear system*

$$A_B x_B \le \overline{b}, -\overline{A}_B x_B \le -\underline{b}, x_B \ge 0, x_N = 0. \tag{2}$$

*If the ILP is B-stable, then each solution of (1) belongs to the optimal solution set and each scenario of the ILP has an optimal solution satisfying (1).*

## References


[1] M. Fiedler, J. Nedoma, J. Ramík, J. Rohn, K. Zimmermann: *Linear Optimization Problems with Inexact Data*, Springer, New York, 2006.

[2] M. Hladík: Interval linear programming: A survey, *Linear Programming – New Frontiers in Theory and Applications*, pp. 85–120, Nova Science Publishers, New York, 2012.

Milan Hladík

hladik@kam.mff.cuni.cz

# Verification by Interval Approach (Ser: Interval Methods)

## Introduction

By **verification** we mean to compute a solution by floating-point arithmetic, and then to verify that the result is correct or to determine rigorous distance to a true solution.

**Problem statement.** As an example, we show a verification method for the problem of finding a root of a function $f\colon \mathbb{R}^n \to \mathbb{R}^n$.

Formally, given $x^*$ a numerically computed (=approximate) solution of $f(x) = 0$, find a small interval $0 \in y \in \mathbb{IR}^n$ such that the true solution lies in $x^* + y$.

## Ingredients

**Brouwer fixed-point theorem**: Let $U$ be a convex compact set in $\mathbb{R}^n$ and $g\colon U \to U$ a continuous function. Then there is a fixed point, i.e., $\exists x \in U : g(x) = x$.

**Observation**: Finding a root of $f(x)$ is equivalent to finding a fixed-point of the function $g(y) \equiv y - C \cdot f(x^* + y)$, where $C$ is any nonsingular matrix of order $n$.

**Perron theory of nonnegative matrices**:

- If $|A| \le B$, then $\rho(A) \le \rho(B)$. (herein, $\le$ is meant entrywise and $\rho(\cdot)$ is the spectral radius)
- If $A \ge 0$, $x > 0$ and $Ax < \alpha x$, then $\rho(A) < \alpha$.

**Mean value theorem**: Everybody knows ... I hope ...

## Cooking

**Lemma 1.** *If $z + Ry \subseteq \operatorname{int} y$, then $\rho(R) < 1$ for every $R \in R$.*

**Theorem 2.** *Suppose $0 \in y$. Now if $-C \cdot f(x^*) + (I - C \cdot \nabla f(x^* + y)) \cdot y \subseteq \operatorname{int} y$, then:*

- *$C$ is nonsingular,*
- *every matrix in $\nabla f(x^* + y)$ is nonsingular, and*
- *there is a unique root of $f(x)$ in $x^* + y$.*

**Proof** By the mean value theorem,

$$f(x^* + y) \in f(x^*) + \nabla f(x^* + y)y.$$

By the assumptions, the function

$$g(y) = y - C \cdot f(x^* + y) \in -C \cdot f(x^*) + (I - C \cdot \nabla f(x^* + y)) \cdot y \subseteq \operatorname{int} y$$

has a fixed point, which shows "existence". By Lemma 1, $C$ and $\nabla f(x^* + y)$ are nonsingular; the latter shows "uniqueness". □

## Recipes

**Hamburg steak**:

- take $C \approx \nabla f(x^*)^{-1}$ (numerically computed inverse),
- take $y := C \cdot f(x^*)$ and repeat inflation

$$y := \Big( -C \cdot f(x^*) + (I - C \cdot \nabla f(x^* + y)) \cdot y \Big) \cdot [0.9, 1.1] + 10^{-20}[-1, 1]$$

  until the assumption of Theorem 2 are satisfied.

## References

[1] S.M. Rump: Verification methods: Rigorous results using floating-point arithmetic, *Acta Numerica 19*, pp. 187–449, 2010

Jaroslav Horáček

horacek@kam.mff.cuni.cz

# Introduction to Interval Computations (Ser: Interval Methods)

## Introduction

The purpose of this talk is to show the basics of interval computations. We will introduce a basis on which the other talks in Interval section will build. In this talk we enclose numbers in our problems with closed real intervals

$$x = [\underline{x}, \overline{x}] = \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \overline{x}\},$$

and show why this description can be useful.

## More formally

With intervals we can define interval arithmetic that works quite similarly to the classical one with real numbers.

**Definition 1.** Interval arithmetic is defined as follows

$$\begin{aligned}
x + y &= [\underline{x} + \underline{y}, \overline{x} + \overline{y}], \\
x - y &= [\underline{x} - \overline{y}, \overline{x} - \underline{y}], \\
x * y &= [\min(S), \max(S)], \quad \text{where } S = \{\underline{xy}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{xy}\}, \\
x \,/\, y &= x * (1/y), \quad \text{where } 1/y = [1/\overline{y}, 1/\underline{y}], 0 \notin y.
\end{aligned}$$

These interval operations are defined to return the tightest interval containing all possible results emerging by picking any real number from the first interval and any real number from the second interval and providing a classical real operation on them. The only problem is that some properties of arithmetic do not hold (e.g., existence of inverse or opposite element, distributivity). However, if necessary, in many formulas we can just replace real numbers and operations with their interval versions.

**Theorem 2.** *Let us have a formula formed only by using $+, -, *, /$ and with each variable occurring only once. When we substitute variables with intervals and continue the computation with interval arithmetic, we also get the tightest possible interval enclosing all possible results of formula applied to all possible real numbers coming from corresponding intervals.*

We wish to build intervals in more complex structures – functions, matrices, linear and nonlinear systems, constraint programming. There might be various reasons why to do that – catching rounding errors, representation of measurement errors, verified intervals containing all possible values or cases. The ideas will be illustrated on some basic interval problems. We will also show practical examples – two famous computer assisted proofs – Kepler's conjecture and Lorentz attractor.

We will introduce the recent projects and research directions and the applications we are interested in – localization of robots with poor sensors, handling clinical interval data in children lung function diagnosis.

For more information about our research and projects please visit *kam.mff.cuni.cz/∼horacek*. For additional basic information about the interval computation do not hesitate to take a look into the references.

## References

[1] J. Horáček, M. Hladík, M. Černý, *Interval Linear Algebra and Computational Complexity*, `http://arxiv.org/abs/1602.00349`, accepted to Applied and Computational Matrix Analysis, Springer Verlag, 2016.

[2] J. Horáček, *Přeurčené soustavy intervalových lineárních rovnic*, Diploma thesis, MFF UK, 2011.

Jana Novotná

jamafyna@gmail.com

# Feasibility in interval linear programming (Ser: Interval Methods)

## Introduction

In linear programming we often want to know if a given problem is feasible or not. We extend the feasibility concept into interval linear programming and the basic questions we ask are: "Is every scenario feasible?" or "Is at least one scenario feasible?". We show that the answer depends on the form of the linear constraint system and that some problems are easy to compute while others are NP-hard.

## More formally

**Definition 1.** Let $\mathbf{A} \in \mathbb{IR}^{m\times n}, \mathbf{b} \in \mathbb{IR}^m, \mathbf{c} \in \mathbb{IR}^n$ be given. By an *interval linear programming (ILP)* problem we mean a family of linear problems

$$\min c^\top x \text{ subject to } x \in \mathscr{M}(A,b) \text{ where } A \in \mathbf{A}, b \in \mathbf{b}, c \in \mathbf{c}$$

where $\mathscr{M}(A,b)$ is a feasible set characterized by a linear system of equations and inequalities. We write it shortly as

$$\min \mathbf{c}^\top x \text{ subject to } x \in \mathscr{M}(\mathbf{A},\mathbf{b}).$$

**Definition 2.** We call *a scenario* a realization of interval values, i.e. $A \in \mathbf{A}, b \in \mathbf{b}, c \in \mathbf{c}$.

We focus on three types of a feasible set description:

(A) $\mathscr{M}(A,b) = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$

(B) $\mathscr{M}(A,b) = \{x \in \mathbb{R}^n \mid Ax \leq b\}$

(C) $\mathscr{M}(A,b) = \{x \in \mathbb{R}^n \mid Ax \leq b, x \geq 0\}$

and we discuss these systems separately because the types cannot be straightforwardly rewritten to each other in interval setting unlike in real value setting.

**Definition 3.** An interval linear system is *strongly feasible* if it is feasible for all scenarios and it is called *weakly feasible* if it is feasible for at least one scenario.

Testing of weak or strong feasibility in type (C) is polynomial whereas one of those problems is NP-hard in types (A) and (B).

**Theorem 4.** *An interval linear system* $\mathbf{A}x \leq \mathbf{b}, x \geq 0$ *is strongly feasible if and only if the system* $\overline{A}x \leq \underline{b}, x \geq 0$ *is feasible.*

**Theorem 5.** *An interval linear system* $\mathbf{A}x \leq \mathbf{b}, x \geq 0$ *is weakly feasible if and only if the system* $\underline{A}x \leq \overline{b}, x \geq 0$ *is feasible.*

Strong feasibility of the type (B) gives us a surprising result:

**Theorem 6.** *The interval linear system* $\mathbf{A}x \leq \mathbf{b}$ *is strongly feasible if and only it has a strong solution, i.e.* $\exists x\ \forall A \in \mathbf{A}\ \forall b \in \mathbf{b} : Ax \leq b$.

Next theorems show how important is the non-negativity of variable $x$. The testing weak feasibility of an ILP $\mathbf{A}x = \mathbf{b}, x \geq 0$ is polynomial whereas of an ILP $\mathbf{A}x = \mathbf{b}$ is NP-hard.

**Theorem 7.** *An interval linear system* $\mathbf{A}x = \mathbf{b}, x \geq 0$ *is weakly feasible if and only if the system* $\underline{A}x \leq \overline{b}, -\overline{A}x \leq -\underline{b}, x \geq 0$ *is feasible.*

**Theorem 8.** *Testing weak feasibility of an interval linear system* $\mathbf{A}x = \mathbf{b}$ *is an NP-hard problem.*

| | Type (A): $\mathbf{A}x = \mathbf{b}, x \geq 0$ | Type (B): $\mathbf{A}x \leq \mathbf{b}$ | Type (C): $\mathbf{A}x \leq \mathbf{b}, x \geq 0$ |
|---|---|---|---|
| strong feasibility | NP-hard | polynomial | polynomial |
| week feasibility | polynomial | NP-hard | polynomial |

## References


[1] M. Hladík: Interval linear programming: A survey, *Linear Programming – New Frontiers in Theory and Applications*, Nova Science Publishers, 2012.

[2] J. Rohn: Interval linear programming, *Linear Optimization Problems with Inexact Data*, Springer US, 2006.

Petra Pelikánová

ppelikanova@gmail.com

# Interval linear regression (Ser: Interval Methods)

## Introduction

Common linear regression can be extended to interval linear regression. There are many approaches. This talk presents two a little bit oposite models of interval linear regression, possibilistic model and necessity model. Tolerance approach is described and also it's adaptation to a data affected by outliers.

## Notation

**Definition 1.** We have $p$ measurements. Matrix of independent variables $X$ represent *input data* of measurements. Vector of dependent variables $y$ represents *output data*.

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_p \end{pmatrix} \quad X = \begin{pmatrix} x_{11} & \dots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{p1} & \dots & x_{pn} \end{pmatrix}$$

**Definition 2.** Basic model of interval linear regression can be represented as

$$\boldsymbol{Y}(X) = X\boldsymbol{a}$$

where $X$ is a $p \times (n+1)$ matrix (where first column is $X_{*1} = (1,\dots,1)^T$ and others are input data), $Y(X)$ is a vector from $\mathbb{IR}^p$ and $\boldsymbol{a} \in \mathbb{IR}^{n+1}$ is a vector of regression parameters. Every $j$-th observation corresponds to one equation: $Y(x_j) = X_{j*}\boldsymbol{a} = X_{j1}\boldsymbol{a} + X_{j2}\boldsymbol{a} + \dots + X_{jn}\boldsymbol{a}$. When $y$ is a vector of output data, $y_j \in Y(x_j)$. There exists $a \in \boldsymbol{a}$ such that it holds

$$y_j = X_{j*}a = a_0 + X_{j1}a_1 + X_{j2}a_2 + \dots + X_{jn}a_n.$$

## Possibilistic and Necessity models

| Possibilistic model | Necessity model |
|---|---|
| $\forall j = 1,\dots,p : \boldsymbol{y}_j \subseteq X_{j*}\boldsymbol{a}$ | $\forall j = 1,\dots,p : \boldsymbol{y}_j \supseteq X_{j*}\boldsymbol{a}$ |
| $\min_{a^c,a^\Delta} \sum_{j=1}^p \lvert X\rvert_{j*}a^\Delta$ | $\max_{a^c,a^\Delta} \sum_{j=1}^p \lvert X\rvert_{j*}a^\Delta$ |
| $\forall j = 1,\dots,p$ | $\forall j = 1,\dots,p$ |
| $\underline{y}_j \geq X_{j*}a^c - \lvert X\rvert_{j*}a^\Delta$ | $\underline{y}_j \leq X_{j*}a^c - \lvert X\rvert_{j*}a^\Delta$ |
| $\overline{y}_j \leq X_{j*}a^c + \lvert X\rvert_{j*}a^\Delta$ | $\overline{y}_j \geq X_{j*}a^c + \lvert X\rvert_{j*}a^\Delta$ |

## Tolerance approach

In a regression model $\boldsymbol{Y}(x) = X\boldsymbol{a}$ the parameter vector is given as

$$\boldsymbol{a} = [a^c - \delta r, a^c + \delta r].$$

**Theorem 3.** *If there is $j \in \{1,\dots,p\}$ such that $|X|_{j*}r = 0$ and $y_j \neq X_{j*}a^c$ then there exists no $\delta$ satisfying*

$$\forall j \in \{1,\dots,p\}\ \exists a' \in [a^c - \delta r, a^c + \delta r] : y_j = X_{j*}a'.$$

*Otherwise let*

$$\delta^* := \max_{j:|X|_{j*}r>0} \frac{|y_j - X_{j*}a^c|}{|X|_{j*}r}.$$

*Then $\delta^*$ is minimal tolerance quotient.*

## References

[1] M. Hladík, M. Černý: Interval regression by tolerance analysis approach, *Fuzzy Sets and Systems*, 2011.

Miroslav Rada

miroslav.rada@vse.cz

# Intervals → Zonotopes → Ellipsoids (Ser: Interval Methods)

### Intervals – linear regression model with interval outputs

First, consider linear regression model in form

$$y = X\beta + \varepsilon,$$

where $y \in \mathbb{R}^m$ are observations of dependent variable, $X \in \mathbb{R}^{m\times p}$ are observations of independent variables and $\varepsilon \in \mathbb{R}^m$ are disturbances. Our goal is to estimate unknown regression parameters $\beta \in \mathbb{R}^p$.

Here, we assume that regression parameters can be estimated with a linear estimator, e.g. with ordinary least squares (hereinafter *OLS*) or, more generally, with generalized least squares. Using OLS, the estimates of regression parameters read $\widehat{\beta} = Qy$ with $Q = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}$.

Now, assume that we don't know the crisp values of $y$. Instead of them, we are given intervals $[\underline{y},\overline{y}] = y \in \mathbb{IR}^m$ for $\underline{y},\overline{y} \in \mathbb{R}^m$, and we know that the true values of $y$ belong to $y$. We call this extension *linear regression model with interval outputs* (formally, it can be viewed as collection of regression models).

What can one say under this setup? Traditional approaches construct an estimator of $\beta$ that is "good" in some sense. However, these approaches usually require some additional assumptions on the distribution of $y$ over $y$. We take another, *possibilistic* approach: we will examine the set of all possible OLS estimates $\widehat{\beta}$ for some $y \in y$.

**Definition 1.** The *OLS-set* of linear regression model with data $y, X$ is the set

$$B(y,X) = \{b \in \mathbb{R}^p : b = Qy, y \in y\}. \tag{3}$$

### Zonotopes – OLS-sets of regression model with interval outputs

The OLS-set in the form (3) is not very user-friendly. It is easy to observe that the OLS-set is linear image of $m$-dimensional hypercube (living in the space of observations of output variable) in the $p$-dimensional space of parameters. Such images are polytopes of a special type with some nice properties, so called *zonotopes*.

Viewing OLS-set as a polytope (zonotope) allows for characterization of the OLS-set in terms of usual representations and/or approximations of polytopes. One might be interested in interval envelope (which gives bounds on individual regression parameters), in list of vertices and facets of the OLS-set, in volume of the OLS-set (as a measure of amount of uncertainty introduced into regression by intervals) or in ellipsoidal approximation of the OLS-set.

The approximative characteristics of the OLS-set are of a high importance, since the exact ones are expensive to compute. In particular, the problem of volume computation of zonotope is in $\#P$, and the vertices and facets of the OLS-set can not be enumerated efficiently just because of the fact that their numbers are superpolynomial in $p$:

**Lemma 2.** *Let $f_0$ resp. $f_{p-1}$ denote the number of vertices resp. facets of the OLS-set. Then $f_0 \leq 2\sum_{i=0}^{p-1}\binom{m}{i}$ and $f_{p-1} \leq 2\binom{m}{p-1}$. Furthermore, the bounds are the tightest possible.*

The next section is focused on ellipsoidal approximation of the OLS-set.

### Ellipsoids – finding tight ellipsoidal approximations of OLS-sets

**Definition 3.** Given a positive definite matrix $E \in \mathbb{R}^{p\times p}$ and a point $c \in \mathbb{R}^p$, the *ellipsoid* is the set

$$\mathscr{E}(E,c) = \{x \in \mathbb{R}^p : (x-p)^{\mathrm{T}}E^{-1}(x-p) \leq 1\}.$$

The following theorem guarantees existence of tight ellipsoidal approximation of the OLS-set:

**Theorem 4.** *Given a convex body $K \subseteq \mathbb{R}^p$, there exists an ellipsoid $\mathscr{E}(E,c)$, called* Löwner-John's ellipsoid for K, *such that*

$$\mathscr{E}(r^2E,c) \subseteq K \subseteq \mathscr{E}(E,c),$$

*where the* shrunk factor *r equals to p.*

*If K is centrally symmetric, the shrunk factor can be increased to $\sqrt{p^{-1}}$.*

In other words, for every convex set $K$ in $\mathbb{R}^p$, there is a circumscribing ellipsoid, whose $p$-times-shrunk copy (resp. $\sqrt{p}$-times-shrunk copy for symmetric bodies) is inscribed into $K$. Note that the stronger version holds for the OLS-set, since it is image of centrally symmetric hypercube.

Given an $H$-polytope, Löwner-John's ellipsoid can be computed (with given precision) in polynomial time using Goffin's algorithm (a variant of shallow-cut ellipsoid method). Unfortunately, Goffin's algorithm can't be used for OLS-set effectively, since its facet representation is not known and can't be computed in reasonable time due to Lemma 2.

Hence, we adapt Goffin's algorithm to handle the input data $y,X$ of the OLS-set. We are able to find Löwner-John's ellipsoid with shrunk factor $p$ with arbitrary precision in time polynomial in size of $y,X$. This shrunk factor is not the best possible. The improvement of the shrunk factor is related to the ability to solve the following problem:

$$\text{Assume } B(y,X) \subseteq \mathscr{E}(I_p,0). \text{ Does } \mathscr{E}(r^2I_p,0) \subseteq B(y,X) \text{ hold for } r = p^{-1/2}? \tag{4}$$

Currently, we are able to test (4) for $r = p^{-1}$. Of course, every result that enables to perform the test (4) for any $r \in (p^{-1}, p^{-1/2}]$ would be of interest.

The Goffin's algorithm for OLS-set is described in more detail in [1].

## References

[1] M. Černý, M. and M. Rada: Polynomial Time Construction of Ellipsoidal Approximations of Zonotopes Given by Generator Descriptions. In M. Agrawal and B. Cooper and A. Li: *Theory and Applications of Models of Computation*, 7287 / *Lecture Notes in Computer Science*. Springer, 2012. pp. 156–163.

# Nonstandard Methods in Combinatorics

Petr Glivický

petrglivicky@gmail.com

## Nonstandard methods in Ramsey combinatorics I

### Introduction

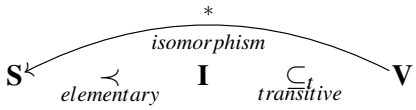
Recently, nonstandard methods have been successfully applied in many areas of combinatorics. The nonstandard methodology provides an extension of the universe of mathematics by new ideal (nonstandard) objects such as "an infinitely large natural number", "an infinitely small neighborhood of a point", and many more. The rich structure of relations between the original (standard) and the new (nonstandard) objects enables the standard objects and their standard properties to be described and studied by means of nonstandard concepts. It turns out that this nonstandard description is in many cases more elegant and the nonstandard proofs clearer and shorter than their standard alternatives.

In this series, we outline a nonstandard approach to Ramsey-type combinatorics. We prove two nonstandard Ramsey-type principles of the following common form (vaguely): "If, in a coloring of finite subsets of $\mathbb{N}$, certain nonstandard object (a witness) has a color $C$, then there is an infinite subset of $\mathbb{N}$ homogeneously having the color $C$."

As an application of these principles we give very short and simple nonstandard proofs of several well-known Ramsey-type combinatorial theorems, including Ramsey's, Hilbert's and Hindman's theorems.

### Nonstandard universes

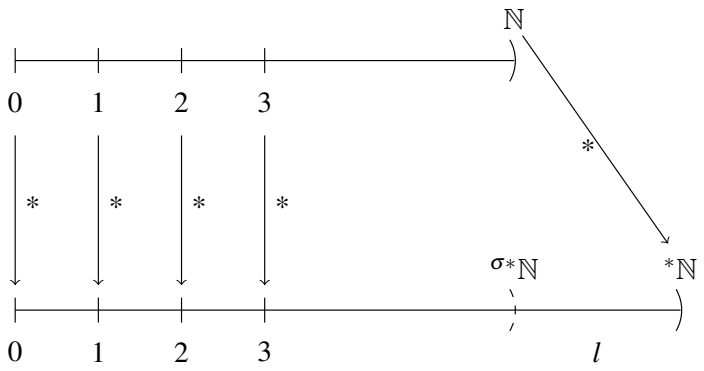
The nonstandard world consists of three interrelated universes (instead of just one universe of the classical mathematics). The universe $\mathbf{V}$ of all sets (= all objects) contains the isomorphic copy $\mathbf{S}$ of itself. The universe $\mathbf{S}$ is extended by the universe $\mathbf{I} \subseteq \mathbf{V}$ which contains new ideal (nonstandard) elements. The whole picture is as follows:




| | | |
|---|---|---|
| $^*\!:\mathbf{V}\to\mathbf{S}$ isomorphism | $\Leftrightarrow$ | $\varphi^{\mathbf{V}}(\overline{a}) \leftrightarrow \varphi^{\mathbf{S}}(^*\overline{a})$ for $\in$-formula $\varphi$, $\overline{a}\in\mathbf{V}$, $^*$ is injective and onto $\mathbf{S} = \mathrm{rng}(^*) = {}^*[\mathbf{V}]$, |
| $\mathbf{S}\prec\mathbf{I}$ (elementary) | $\Leftrightarrow$ | $\varphi^{\mathbf{S}}(\overline{a}) \leftrightarrow \varphi^{\mathbf{I}}(\overline{a})$ for $\in$-formula $\varphi$ and $\overline{a}\in\mathbf{S}$, |
| $\mathbf{I}\subseteq_t\mathbf{V}$ (transitive) | $\Leftrightarrow$ | $y\in x\in\mathbf{I}\to y\in\mathbf{I}$, |
| $\mathbf{I}$ almost universal | $\Leftrightarrow$ | $x\subseteq\mathbf{I}\to(\exists y\in\mathbf{I})(x\subseteq y)$, |
| $\mathbf{I}$ $\kappa$-saturated | $\Leftrightarrow$ | $(\mathscr{C}\subseteq\mathbf{I}$ centered & $\lvert\mathscr{C}\rvert<\kappa)\to\bigcap\mathscr{C}\neq\emptyset$ where a system $\mathscr{C}$ is centered if for any finite $\mathscr{C}'\subseteq\mathscr{C}$ it is $\bigcap\mathscr{C}'\neq\emptyset$ |

### Nonstandard natural numbers

This is what happens with the set $\mathbb{N}$ of all natural numbers:




$^*n = n \in \mathbf{S}$ for $n \in \mathbb{N}$

$l \in \mathbf{I} - \mathbf{S}$ infinite natural number

$^*\mathbb{N} \in \mathbf{S}$ standard set of nat. numbers

$\sigma^*\mathbb{N} = \mathbb{N} \notin \mathbf{I}$ set of standard natural numbers

$^*\mathbb{N} - \mathbb{N} \notin \mathbf{I}$ set of non-standard natural numbers

Whatever property of numbers $\overline{n} \in \mathbb{N}$ is true in $\underline{\mathbb{N}}$, is true in $^*\underline{\mathbb{N}}$ and vice versa.

### Basic nonstandard principles

We list some handy "nonstandard principles":

**Proposition 1.** *(Transfer principle). Let $\varphi$ be a **bounded** formula (i.e. with all quantifiers bounded in a set: $\forall x \in y$, $\exists x \in y$). Then*

$$\varphi(\overline{x}) \leftrightarrow \varphi(^*\overline{x}).$$

**Corollary 2.** *Let $\varphi$ be a **bounded** formula. Then*

$$^*\{x \in B\,;\, \varphi(x,\overline{z})\} = \{x \in {}^*B\,;\, \varphi(x,{}^*\overline{z})\}$$

*for all $B,\overline{z}$.*

**Proposition 3.** *Let operation $o(\overline{x})$ is defined by a **bounded** formula $\varphi(\overline{x},z)$.*

1. *Then $^*o(\overline{x}) \subseteq o(^*\overline{x})$ for every $\overline{x}$.*
2. *If, moreover, $o(^*\overline{x}) \subseteq A \cup \bigcup B$ for some $A,B \subseteq \mathbf{S}$, then $^*o(\overline{x}) = o(^*\overline{x})$.*

### Iterated star

The mapping $^*:\mathbf{V} \to \mathbf{S}$ can be iterated: For $n \in \mathbb{N}$ the mapping $^{n*} : \mathbf{V} \to \mathbf{V}$ is $^*$ applied $n$-times.

We also define the mapping $\dot{} : \mathbf{V} \to \mathbf{V}$ by

$$\dot{x} = \bigcup_{n\in\mathbb{N}} {}^{n*}x.$$

We have the following chain of nonstandard extensions of $\mathbb{N}$: $\mathbb{N} \subsetneq {}^*\mathbb{N} \subsetneq {}^{**}\mathbb{N} \subsetneq \cdots \subsetneq \dot{\mathbb{N}}$.

The sets $^{n*}\mathbb{N}$ are transitive, i.e. they are initial segments of $\dot{\mathbb{N}}$ with respect to the canonical ordering $\in$. Letters $\alpha,\beta,\nu$ denote elements of $\dot{\mathbb{N}}$, $\overline{\alpha},\overline{\beta},\overline{\nu}$ tuples of elements from $\dot{\mathbb{N}}$.

### References


[1] Glivický, Petr and Mlček, Josef: Nonstandard numbers and Ramsey-type combinatorics, *preprint*, 2016.

[2] di Nasso, Mauro: A taste of nonstandard methods in combinatorics of numbers, *arXiv:1312.5059*, 2013.

[3] Robinson, Abraham: Non-standard analysis, *Princeton Landmarks in Mathematics (2nd ed.), Princeton University Press*, 1996.

[4] Hindman, Neil and Strauss, Dona: Algebra in the Stone-Čech compactification. Theory and applications, *de Gruyter textbook (2nd ed.), de Gruyter, Berlin, Boston*, 2011.

Tomáš Toufar

toufi@iuuk.mff.cuni.cz

# Nonstandard methods in Ramsey combinatorics II

## Introduction

In this part of nonstandard series, we introduce the tools required to work with iterated $^*$ mapping. We also sketch the proof of an existence of idempotent elements, as these play a crucial role in arithmetic Ramsey theorems.

## Iterated $^*$ and transfer principle

**Definition 1.** For $\alpha \in {}^\cdot\mathbb{N}$, the *rank* of $\alpha$ is defined as $r(\alpha) = \min\{n \in \mathbb{N}\,;\, \alpha \in {}^{n*}\mathbb{N}\}$. For a vector $\overline{\alpha}$, its rank is defined as $r(\overline{\alpha}) = \max_{i<l(\overline{\alpha})} r(\alpha_i)$.

**Lemma 2.** *(Variant of transfer principle). Let $\varphi$ be a bounded formula, $\overline{\alpha} \in {}^\cdot\mathbb{N}$, and $m,n \in N$ such that $m,n \geq r(\overline{\alpha})$. Then*

$$\varphi(\overline{\alpha}, {}^{n*}\overline{y}) \leftrightarrow \varphi(\overline{\alpha}, {}^{m*}\overline{y}).$$

**Lemma 3.** *Let $\mathscr{S}$ denote the set of all functions $g\colon \mathbb{N}^m \to \mathbb{N}$ and relations $R \subseteq \mathbb{N}^m$ of all arities $m$. The structures $\langle {}^{n*}\mathbb{N}, {}^{n*}s\rangle_{s\in\mathscr{S}}$ form the elementary chain*

$$\langle \mathbb{N}, s\rangle_{s\in\mathscr{S}} \prec \langle {}^*\mathbb{N}, {}^*s\rangle_{s\in\mathscr{S}} \prec \langle {}^{**}\mathbb{N}, {}^{**}s\rangle_{s\in\mathscr{S}} \prec \cdots \prec \langle {}^\cdot\mathbb{N}, {}^\cdot s\rangle_{s\in\mathscr{S}}$$

*with the limit $\langle {}^\cdot\mathbb{N}, {}^\cdot s\rangle_{s\in\mathscr{S}}$.*

**Lemma 4.** *For any relation $R \subseteq \mathbb{N}^{<\omega}$, $B \subseteq \mathbb{N}$, and $\overline{\alpha} \in {}^\cdot\mathbb{N}$,*

$$\{x \in {}^{m*}B\,;\, {}^\cdot R(x, {}^{m*}\overline{\alpha})\} = {}^{m*}\{x \in B\,;\, {}^\cdot R(x,\overline{\alpha})\}.$$

## Grading

**Definition 5.** We define the *grading* transformation $^\uparrow : {}^\cdot\mathbb{N}^{<\omega} \to {}^\cdot\mathbb{N}^{<\omega}$ by

$$\overline{\alpha}^\uparrow = (\alpha_0, {}^{r(\alpha_0)*}\alpha_1, {}^{(r(\alpha_0)+r(\alpha_1))*}\alpha_2, \ldots, {}^{(\Sigma_{i<l(\overline{\alpha})-1} r(\alpha_i))*}\alpha_{l(\overline{\alpha})-1})$$

We set $r^\uparrow(\overline{\alpha}) = r(\overline{\alpha}^\uparrow) = \sum r(\alpha_i)$. For every (partial) function $g$ from $\mathbb{N}^{<\omega}$ into $\mathbb{N}$ and every relation $R \subseteq \mathbb{N}^{<\omega}$ we set

$$g^\uparrow(\overline{\alpha}) = {}^\cdot g(\overline{\alpha}^\uparrow), \qquad\qquad R^\uparrow(\overline{\alpha}) \leftrightarrow {}^\cdot R(\overline{\alpha}^\uparrow),$$

## Indistinguishability

**Definition 6.** For $\alpha,\beta \in {}^\cdot\mathbb{N}$ we define the equivalence relation $\sim$ (the *indistinguishability equivalence*) by

$$\alpha \sim \beta \leftrightarrow (\forall A \subseteq \mathbb{N})(\alpha \in {}^\cdot A \leftrightarrow \beta \in {}^\cdot A).$$

For tuples $\overline{\alpha},\overline{\beta}$, the equivalence is defined entry-wise: $\overline{\alpha} \sim \overline{\beta}$ if $\alpha_i \sim \beta_i$ for every $i < l(\overline{\alpha}) = l(\overline{\beta})$.

**Lemma 7.** *Let $\alpha, \beta \in {}^{\cdot}\mathbb{N}$. Then*

1. $\alpha \sim {}^{n*}\alpha$ *for every* $n \in \mathbb{N}$,
2. $\alpha \sim \beta \to {}^{m*}\alpha \sim {}^{n*}\beta$ *for every* $m, n \in \mathbb{N}$,
3. $\overline{\alpha} \sim \overline{\beta} \to \overline{\alpha}^{\uparrow} \sim \overline{\beta}^{\uparrow}$.

**Definition 8.** Let $R \subseteq \mathbb{N}^{<\omega}$ be a relation, we denote

$$R[\overline{\gamma}_{\smile} . {}_{\smile}\overline{\delta}] = \{\overline{x} \in \mathbb{N}^{<\omega} ; R(\overline{\gamma}_{\smile}\overline{x}_{\smile}\overline{\delta})\}, \qquad R[. {}_{\smile}\overline{\delta}] = R[\emptyset_{\smile} . {}_{\smile}\overline{\delta}].$$

**Lemma 9.** *Let $\overline{\alpha} \in {}^{\cdot}\mathbb{N}$, $\overline{m}, \overline{k} \in \mathbb{N}$. Then:*

1. $R^{\uparrow}(\overline{\alpha}) \leftrightarrow \emptyset \in R[. {}_{\smile}\overline{\alpha}^{\uparrow}]$.
2. $\overline{m} \in R[. {}_{\smile}\overline{\alpha}^{\uparrow}] \leftrightarrow \alpha_0 \in {}^{r(\alpha_0)*}R[\overline{m}_{\smile} . {}_{\smile}({}^{\frown}\overline{\alpha})^{\uparrow}]$.
3. $\overline{\alpha} \sim \overline{\beta} \to R[\overline{k}_{\smile} . {}_{\smile}\overline{\alpha}^{\uparrow}] = R[\overline{k}_{\smile} . {}_{\smile}\overline{\beta}^{\uparrow}]$,

*where ${}^{\frown}\overline{\alpha}$ denotes the tuple $\overline{\alpha}$ without its first element.*

**Theorem 10.** *The equivalence $\sim$ is a congruence with respect to $s^{\uparrow}$ whenever $s$ is a (partial) function from $\mathbb{N}^{<\omega}$ into $\mathbb{N}$ or a relation $s \subseteq \mathbb{N}^{<\omega}$.*

**Corollary 11.** *Let $g\colon \mathbb{N} \to \mathbb{N}$ be an unary function. Then $\sim$ is congruence with respect to $g$.*

### Topology and idempotent elements

**Definition 12.** We define

$$\tilde{\mathbb{N}} = {}^{\cdot}\mathbb{N}/_{\sim},\ \tilde{g} = g^{\uparrow}/_{\sim},\ \tilde{R} = R^{\uparrow}/_{\sim},\ \tilde{\alpha} = \alpha/_{\sim}$$

for $g : \mathbb{N}^{<\omega} \to \mathbb{N}$, $R \subseteq \mathbb{N}^{<\omega}$, $\alpha \in {}^{\cdot}\mathbb{N}$, and $\Upsilon : \mathscr{P}(\mathbb{N}) \to \mathscr{P}(\tilde{\mathbb{N}})$ by

$$\Upsilon(A) = \{\tilde{\alpha} ; \alpha \in {}^{\cdot}A\}.$$

The set $\mathscr{B} = \{\Upsilon(A) ; A \subseteq \mathbb{N}\}$ is a basis of a topology on $\tilde{\mathbb{N}}$. We call this topology the *canonical topology* on $\tilde{\mathbb{N}}$.

**Lemma 13.** *If $\textcircled{k} > 2^{\aleph_0}$, then*

1. $(\tilde{\mathbb{N}}, \tau(\mathscr{B}))$ *is a compact Hausdorff space.*
2. *All functions $\tilde{g}$ with $g : \mathbb{N}^{<\omega} \to \mathbb{N}$ are continuous in the first coordinate in $(\tilde{\mathbb{N}}, \tilde{\mathbb{W}})$.*

**Definition 14.** Let $g\colon \mathbb{N}^n \to \mathbb{N}$ be a function. An element $\nu \in {}^{\cdot}\mathbb{N}$ is *$g$-idempotent*, if $g^{\uparrow}(\overline{\nu}^n) \sim \nu$.

**Lemma 15.** *(Ellis-Numakura). Let $S$ be a semigroup with a compact Hausdorff topology and such that the group operation is left-continuous. Then $S$ contains an idempotent element.*

*Moreover, every element of the minimal compact subsemigroup of $S$ is idempotent.*

**Corollary 16.** *There exist $\nu, \nu' \in {}^{*}\mathbb{N} - \mathbb{N}$ such that $\nu$ is $+$-idempotent and $\nu'$ is $\cdot$-idempotent.*

Karel Král

kralka@kam.mff.cuni.cz

# Nonstandard methods in Ramsey combinatorics III

## Nonstandard Ramsey-Type Principles

For operations $f, g$ on $\mathbb{N}$ and $n \leq ar(f)$ we define

$$(f \circ_n g)(\overline{\alpha}, \overline{\beta}, \overline{\gamma}) = f(\overline{\alpha}, g(\overline{\beta}), \overline{\gamma})$$

for all $\overline{\alpha} \in \mathbb{N}^{n-1}$, $\overline{\beta} \in \mathbb{N}^{ar(g)}$, $\overline{\gamma} \in \mathbb{N}^{ar(f)-n}$.

**Lemma 1.** $(f \circ_n g)^{\uparrow} = f^{\uparrow} \circ_n g^{\uparrow}$.

**Lemma 2.** *If $\nu$ is a common idempotent of $\mathscr{G}$, then it is also a common idempotent of $\overline{\mathscr{G}}^{op}$.*

**Definition 3.** $X \subseteq \langle {}^{\cdot}\mathbb{N}\rangle^{<\omega}$ is $\nu$-$N$-inductive if $(\forall F \in \langle \mathbb{N}\rangle^{<\omega})(\wedge_{i=1}^{N-1} F \frown \overline{\nu}^i \in X \to F \frown \overline{\nu}^N \in X)$

**Theorem 4.** *Let $C \subseteq \langle \mathbb{N}\rangle^{<\omega}$, $S \subseteq \mathbb{N}$, $\nu \in {}^*S - \mathbb{N}$.*

1. *Let $I \subseteq \mathbb{N}$ be finite and suppose that $\overline{\nu}^n \in C^{\uparrow}$ for all $n \in I$. Then there is an infinite set $A \subseteq S$ such that for every $n \in I$ and $m \leq n$*
$$\langle A, \nu\rangle^{m,n-m} \subseteq C^{\uparrow} \text{ and thus in particular } \langle A\rangle^n \subseteq C.$$
2. *Suppose that $C^{\uparrow}$ is $\nu$-$N$-inductive, $N > 1$, and $\overline{\nu}^n \in C^{\uparrow}$ for all $0 < n \leq N-1$. Then there is an infinite set $A \subseteq S$ such that*
$$\langle A, \nu\rangle^{<\omega,\leq N} - \{\emptyset\} \subseteq C^{\uparrow} \text{ and thus in particular } \langle A\rangle^{<\omega} - \{\emptyset\} \subseteq C.$$

## Applications

A coloring of a set $X$ is any family $\mathscr{C}$ of mutually disjoint sets such that $X \subseteq \bigcup \mathscr{C}$. It can be easily seen that if $\mathscr{C}$ is a coloring of $X \subseteq \langle \mathbb{N}\rangle^{<\omega}$, then $\mathscr{C}^{\uparrow} = \{C^{\uparrow}\,;\, C \in \mathscr{C}\}$ is a coloring of $X^{\uparrow}$. In particular, if $X$ is some $\langle \mathbb{N}\rangle^n$ or $\langle \mathbb{N}\rangle^{<\omega}$, then $\mathscr{C}^{\uparrow}$ is a coloring of $X$ (as $X \subseteq X^{\uparrow}$).

**Theorem 5.***[Ramsey] Let $\mathscr{C}$ be a finite coloring of $\langle \mathbb{N}\rangle^n$, $n \in \mathbb{N}$. Then there is an infinite $A \subseteq \mathbb{N}$ such that $\langle A\rangle^n \subseteq C$ for some $C \in \mathscr{C}$.*

We can even prove the following stronger statement directly:

**Theorem 6.***[Ramsey II] Let $I \subseteq \mathbb{N}$ be finite and let $\mathscr{C}_n$ be a finite coloring of $\langle \mathbb{N}\rangle^n$ for every $n \in I$. Then there are an infinite set $A \subseteq \mathbb{N}$ and colors $C_n \in \mathscr{C}_n$ such that $\langle A\rangle^n \subseteq C_n$ for every $n \in I$.*

**Proof** Let us take arbitrary $\nu \in {}^*\mathbb{N} - \mathbb{N}$ and denote $C_n^{\uparrow} \in \mathscr{C}_n^{\uparrow}$ the colors of $\overline{\nu}^n$ (that is $\overline{\nu}^n \in C_n^{\uparrow}$) for $n \in I$. The existence of $A$ follows directly from Theorem 4.1 for $C = \bigcup_{n\in I} C_n$. □

**Theorem 7.***[Hilbert] Let $\mathscr{C}'$ be a finite coloring of $\mathbb{N}$ and $m \in \mathbb{N}$. Then there are an infinite set $A \subseteq \mathbb{N}$ and $C' \in \mathscr{C}'$ such that $\sum F \in C'$ for all $F \in \langle A\rangle^{\leq m}$.*

**Theorem 8.***[Hindman] Let $\mathscr{C}$ be a finite coloring of $\mathbb{N}$. Then there are an infinite set $A \subseteq \mathbb{N}$ and $C \in \mathscr{C}$ such that $\sum F \in C$ for all $F \in \langle A\rangle^{<\omega}$.*

**Proof** Let us take $\nu \in {}^*\mathbb{N} - \mathbb{N}$ a $+$-idempotent, denote $C'$ its color (that is $\nu \in {}^{\cdot}C'$), and apply Theorem 4.2 for $C = \{F \in \langle \mathbb{N}\rangle^{<\omega}\,;\, \sum F \in C'\}$. To verify that $C^{\uparrow}$ is $\nu$-2-inductive, it is enough to observe that $\sum F + \nu \sim \sum F + \nu + {}^*\nu$

for every $F \in \langle \mathbb{N} \rangle^{<\omega}$, which follows directly from $+$-idempotence of $\nu$. $\square$

### Partition Regularity and van der Waerden's Theorem for Length 3

We say that a tuple $\bar{x}$ is injective if its elements are mutually distinct. A set $X \subseteq \mathbb{N}^n$ is called injectively partition regular if for every finite coloring $\mathscr{C}$ of $\mathbb{N}$ there is an injective $\bar{x} \in X$ that is monochromatic (i.e. $\bar{x} \in C$ for some $C \in \mathscr{C}$). An equation $f(\bar{x}) = 0$ over $\mathbb{N}$ is called injectively partition regular if the set of all its solutions is.

**Proposition 9.** *Let $X \subseteq \mathbb{N}^n$. The following statements are equivalent:*

1. *$X$ is injectively partition regular.*
2. *There is injective $\overline{\nu} \in {}^*X$ such that $\nu_0 \sim \cdots \sim \nu_{n-1}$.*
3. *There is injective $\overline{\nu} \in {}^\cdot X$ such that $\nu_0 \sim \cdots \sim \nu_{n-1}$.*

**Proposition 10.** *The equation $x + y = 2z$ over $\mathbb{N}$ is injectively partition regular.*

**Corollary 11.** *[van der Waerden's Theorem for length 3] Let $\mathscr{C}$ be a finite coloring of $\mathbb{N}$. Then there is $C \in \mathscr{C}$ and an arithmetic progression $a, a+d, a+2d \in C$.*

# Standalone Talks

Vašek Blažej

blazeva1@fit.cvut.cz

Presented paper by Srikrishnan Divakaran

## Fast Algorithms for Exact String Matching

Given a pattern string $P$ of length $n$ and a query string $T$ of length $m$, where the characters of $P$ and $T$ are drawn from an alphabet of size $\Delta$, the exact string matching problem consists of finding all occurrences of $P$ in $T$.

We can solve this problem using: (1) ***character based comparison algorithms***, (2) automata based algorithms, (3) algorithms based on bit-parallelism and (4) constant-space algorithms.

The character based comparison algorithms are ofter based on ***sliding window mechanism*** which does the following.

(1) Align the $n$ characters of the pattern string $P$ with the first $n$ characters of $T$ – the search window.

(2) Repeat the following until the search window is no longer contained within the query string $T$: inspect the aligned pairs in some order until there is either a mismatch in an aligned pair or there is a complete match among all the n aligned pairs. Then shift the search window to the right.

The order in which the aligned pairs are inspected and the length by which the search window is shifted differs from one algorithm to another. The main approaches to the inspections are: (1) *left to right scan;* (2) *right to left scan;* (3) scan in specific order, and (4) scan in random order or scan order is not relevant.

1) Morris and Pratt; ***Knuth, Morris and Pratt***

When shifting, we expect that a prefix $v$ of the pattern matches some suffix of the portion $u$ of the text. The longest such prefix $v$ is called the ***border*** of $u$ (it occurs at both ends of $u$). Borders can be computed in preprocessing in $O(m)$ and using this principle, we reach $O(m+n)$ search time complexity.

2) ***Boyer-Moore-Horspool***

Compare aligned pairs from right to left. Preprocess pattern to find distance from end of the rightmost occurrence of each character (excluding last symbol). When shifting, move sliding window by preprocessed value for the first compared character in the text. This assures that it is aligned to matching character in the pattern. Runs in $O(mn)$.

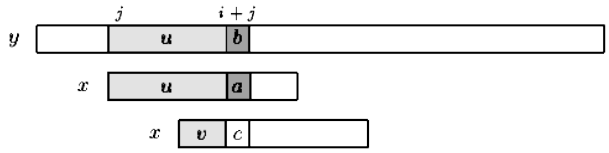
3) ***Boyer-Moore***

Extends *Boyer-Moore-Horspool* and adds notion of ***good-suffix shift***. We create array *suff* where we store how long suffix of prefix ending at position $j$ matches the suffix of the whole pattern. This allows us to shift so that the found suffix aligns with its rightmost occurrence in the pattern. When no occurrence is found, we can align it with prefix (principle of border). Since we can still use principle from *Boyer-Moore-Horspool,* we use maximum value to shift. Search worst case complexity is $O(mn)$.

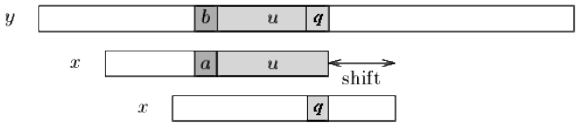
4) ***Apostolico and Giancarlo***

Extends *Boyer-Moore* with memory – it stores information about matched suffixes, so it doesn't have to check same pair twice. We keep this information in *skip* array. Lets denote length of matched suffix with $k$ and see what happens when window is shifted. We have to compare all new characters from right. When we reach known suffix three cases can arise: When $k > suff[i]$ and $suff[i] = i+1$ we found a match. If $k \neq suff[i]$ we know there will be mismatch, because either this part matched before but is different now or it did not match before but stayed the same. Third case is when $k = suff[i]$, this means that we skipped exactly and now we compare further pairs.

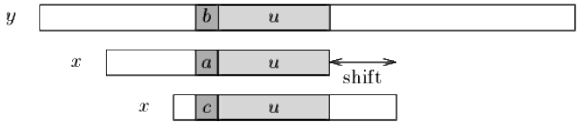
This algorithm performs in the worst case at most 3/2n text character comparisons.



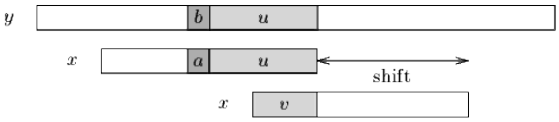

(a) Knuth, Morris and Pratt



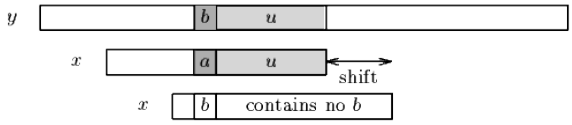

(b) Boyer-Moore-Horspool bad character shift



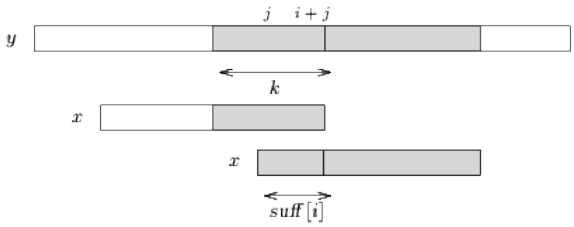

(c) Boyer-Moore good suffix shift



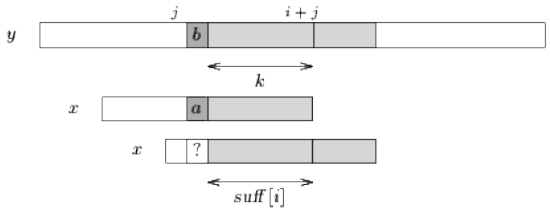

(d) Boyer-Moore matching prefix




(e) Boyer-Moore bad character shift




(f) Apostolico and Giancarlo, when $k > suff[i]$ and $suff[i] = i+1$




(g) Apostolico and Giancarlo, when $k = suff[i]$

Lets now expand on those simple algorithms and devise algorithms A and B.

Definitions 2.1

Given a pattern string $P$ of length $n$ and a query string $T$ of length $m$, we define $N_i(P)$, $i \in [1..n]$, denote the length of the longest suffix of $P[1..i]$ that matches a suffix of $P$, and $M$ to be a $m$ length vector whose $j$-th entry $M[j] = k$ indicates that a suffix of $P$ of length at least $k$ occurs in $T$ and ends at position $j$.

Definitions 2.2

Given a pattern string $P$, and an ordered pair of characters $u, v \in \Sigma$ (not necessarily distinct), we define $\texttt{sparse}^{(\mathrm{u,v})}(\mathrm{P})$, the 2sparse pattern of $P$ with respect to $u$ and $v$, to be the 3 rightmost occurrence of a substring of $P$ of longest length that starts with $u$, ends with $v$, but does not contain $u$ or $v$ within it. We define $\texttt{sparse}(\mathrm{P})$ to be the longest among the 2sparse patterns of $P$.

Definitions 2.3

Given $\texttt{sparse}(\mathrm{P})$, we define $\texttt{startc}(\mathrm{P})$ and $\texttt{endc}(\mathrm{P})$ to be the respective first and last characters of $\texttt{sparse}(\mathrm{P})$, and $\texttt{startpos}(\mathrm{P})$ and $\texttt{endpos}(\mathrm{P})$ be the respective indices of the first and last characters of $\texttt{sparse}(\mathrm{P})$ in $P$. For $c \in \Sigma$, if $c \in \texttt{sparse}(\mathrm{P})$, $\texttt{shift}_\mathrm{c}(\mathrm{P})$ is the distance between the rightmost occurrence of $c$ in $\texttt{sparse}(\mathrm{P})$ and the last character of $\texttt{sparse}(\mathrm{P})$. If $c$ is not present in $P$ then $\texttt{shift}_\mathrm{c}(\mathrm{P})$ is set to $n$, the length of $P$. If $c$ is present in $P$ but not in $\texttt{sparse}(\mathrm{P})$ then $\texttt{shift}_\mathrm{c}(\mathrm{P})$ is set to $|\texttt{sparse}(\mathrm{P})| + 1$.

Theorem 1

Given any pattern string $P$ of length $n$ and a query string $T$ of length $m$, Algorithm A finds all occurrences of $P$ in $T$ in

$O(m)$ time.

Theorem 2

Given any pattern string $P$ of length $n$ and a query string $T$ of length $m$ where each character is drawn uniformly at random, Algorithms A and B find all occurrences of $P$ in $T$ in $O(m/min(|\texttt{sparse(P)}|, \Delta))$ expected time, where $|\texttt{sparse(P)}|$ is at least $\delta$ (i.e the number of distinct characters in $P$).

Lemma 3

For any pattern string $P$, the length of $\texttt{sparse(P)}$, the longest $2-\texttt{sparse}$ pattern of $P$, is at least $\delta$, where $\delta$ is the number of distinct characters in $P$.

Lemma 4

For any pattern string $P$, Algorithm A preprocesses $P$ in $O(n\Delta^2$ time to determine (i) $N_i(P)$, for $i \in [1..n]$, (ii) $\texttt{sparse(P)}$, and (iii) $\texttt{shift}_\texttt{c}\texttt{(P)}$, for $c \in \Sigma$, where $\Delta$ is the number of characters in its alphabet $\Sigma$.

Lemma 5

For any pattern string $P$, during the search phase of Algorithm A, the expected length of shift of $P$ after a Type-1, Type-2 or Type-3 event is at least $O(min(|\texttt{sparse(P)}|, \Delta))$.

Lemma 6

For any given pattern string $P$ of length $n$ from $\Sigma$ and a query string $T$ whose characters are drawn independently and uniformly from $\Sigma$, the expected number of matches before a mismatch when invoking Apostolico-Giancarlo or Random-Match Algorithm is $O(1)$.

Radovan Červený

cervera3@fit.cvut.cz

Presented paper by On interval representations of graphs

# On interval representations of graphs

(http://www.tau.ac.il/~nogaa/PDFS/ijcai15.pdf)

## Preliminary

$k$-vertex is a vertex of degree $k$. *true twin* – vertices $u$, $v$ are true twins only if $N[u] = N[v]$. $P_g$ is a class of planar graphs with girth atleast $g$.

## General

- *d-interval representation* of $G$ is an assignement of at most $d$ intervals of the real line to every vertex of $G$. Two vertices of $G$ are adjacent only if some intervals of respective vertices intersect.
- *interval component* of an $d$-interval representation is a maximal subset $S$ of the real line such that every point is contained in an interval in the representation.
- *d-local representation* of $G$ is a $d$-interval representation of $G$ with the additional requirement that two intervals for the same vertex belong to distinct interval components.
- *d-track representation* of $G$ is given by the union of $d$ (1-interval) representations of the graphs $G_j$ for $1 \leq j \leq d$ such that $G_j$ has the same vertex set as $G$ and $E(G) = \bigcup_{1 \leq j \leq t} E(G_j)$
- *interval number i(G)* of $G$ is the least integer $d$ such that $G$ has a $d$-interval representation.
- *local track number l(G)* of $G$ is the least integer $d$ such that $G$ has a $d$-local representation.
- *track number t(G)* of $G$ is the least integer $d$ such that $G$ has a $d$-track representation.

## Article specific

- interval is *displayed* if some part of that interval does not intersect any other interval.
- extremity (one of the two ends) of an interval is *displayed* if it does not intersect any other interval.
- vertex is *displayed* if it is represented by strictly less intervals than what is allowed by considered representation or if one of its intervals is displayed.
- interval *a covers* interval *b* if *b* is contained in *a*.

## Theorems

**Theorem 1.** *The local track number of a graph in* $\mathrm{P}_7$ *is at most 2.*

**Theorem 2.** *Given a graph G, determining whether* $t(G) \leq d$ *is NP-complete, even if G is* $(K_4, 2K_3)$*-free, alternately orientable, a Meyniel graph, and a string graph.*

**Theorem 3.** *Given a 2-degenerate planar graph G with maximum degree 5, determining whether* $l(G) \leq 2$ *is NP-complete.*

Answer to the question: Is it possible to construct graph $W$ that is $d$-track and not the union of $d$ pairwise edge-disjoin interval graphs? We construct such graph $W$.

**Theorem 4.** *The graph W is a 2-track graph and the edge* $AA'$ *is represented twice in every 2-interval representation.*

Pavel Dvořák

jajsem@koblich.cz

Presented paper by Yaroslav Shitov

# The Diameter of Chess Grows Exponentially

(http://www.jstor.org/stable/10.4169/amer.math.monthly.123.1.71)

## Introduction

The paper contributes to the complexity theory of *puzzles* (games for one player). The puzzle is a directed graph whose vertices are called *position* and arcs are called *moves*. An initially position is given to the player and he has to find an oriented path to the final position. Examples of puzzles are *Rubik's cube, Game of Fifteen, Sokoban* etc.

The important parameter of puzzles is its *diameter* which is a length of the longest directed path in the graph of the puzzle. Some of the puzzles have diameter bounded by a polynomial in the size of puzzle. This is the case for generalizations of Rubik's cube [1] and Game of Fifteen [2] such that the puzzle has arbitrary size. Therefore, these puzzles can be solved in NP. However, there is no such known bound for Sokoban and this puzzle is PSPACE-complete [3].

## Chess Puzzle

We will study a diameter of puzzle related to chess. There is an $n \times n$ chess board and the player need to construct a legal series of chess moves from the initial position and the final position.

**Theorem 1.** *There is an infinite sequence $(A_n, B_n)$ of pairs of chess positions on an $n \times n$ chess board such that the minimum number of legal moves required to get from $A_n$ to $B_n$ is exponential in $n$.*

The main idea of the proof is depicted on Figure 2. The position $A_n$ is on the figure, the position $B_n$ is same as $A_n$ except the bishops in circles are switched. If we want to move the white bishops in the left most cycle to the right we have to move all white bishops in the same direction. The $i$-th bottom cycles have length $2p_i + 2$ where $p_i$ is a prime. Moreover, every bottom cycle has distinct length. To reach the position $B_n$ we have to rotate each bottom cycle at least $\prod_{i=2}^{m} p_i$ where $m$ is the number of the bottom cycles. This gives us the required lower bound.

## References

[1] E. D. Demaine, M. L. Demaine, S. Eisenstat, A. Lubiw, A. Winslow, Algorithms for solving Rubik's cubes, in *Algorithms–ESA*. Vol. 6942. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011. 689–700.

[2] D. Ratner, M. Warmuth, Finding a shortest solution for the $N \times N$ extension of the 15-puzzle is intractable, *J. Symb. Comput.* **10** (1990) 111–137.

[3] R. A. Hearn, E. D. Demaine, PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation, *Theor. Comput. Sci.* **343** (2005) 72–96.

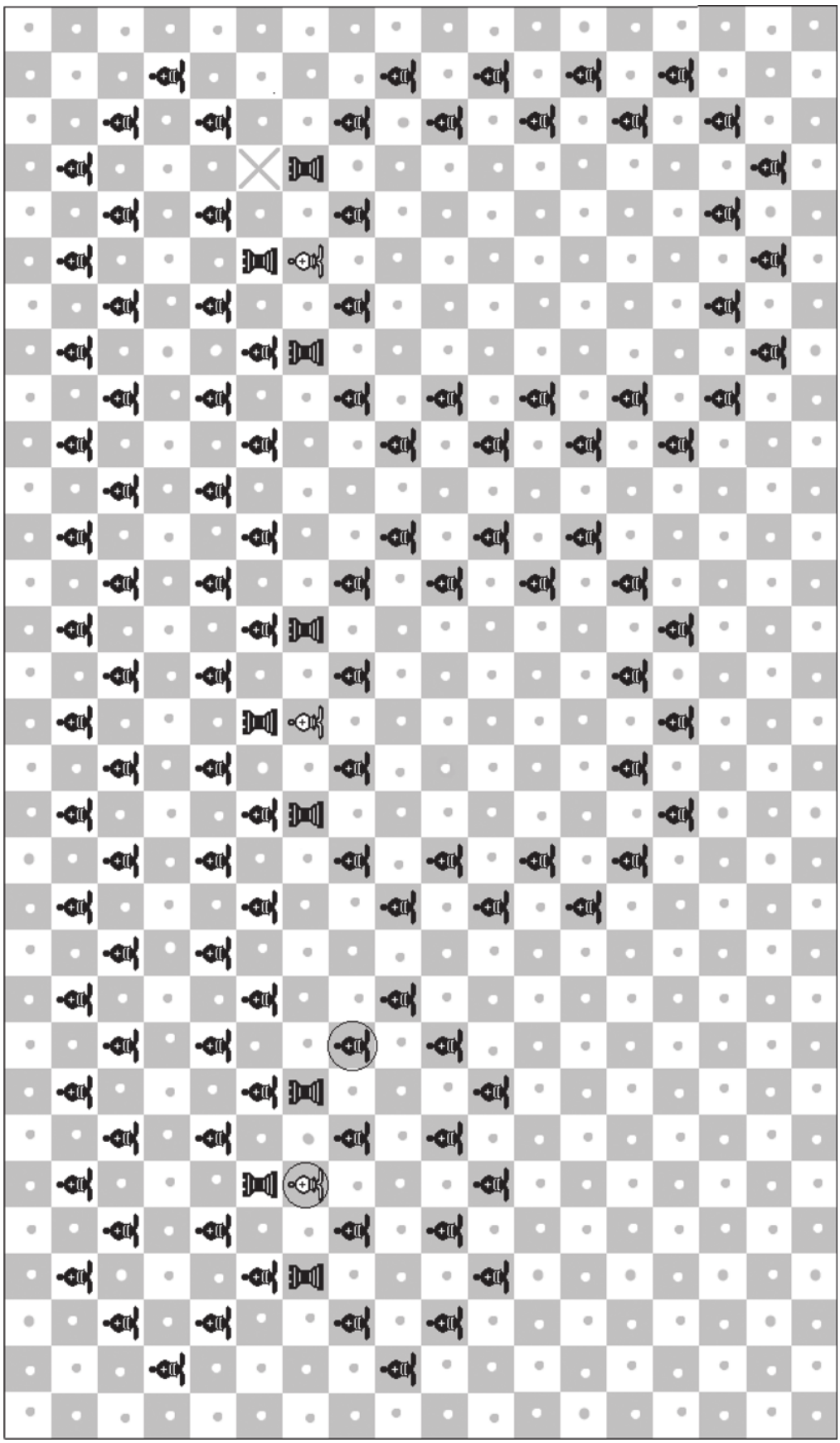


Figure 2: Initial position for the puzzle.

Andreas Emil Feldmann

andreas.feldmann@uwaterloo.ca

# Fixed Parameter Approximations for $k$-Center Problems in Low Highway Dimension Graphs

**Abstract**

For the $k$-Center problem, a set of $k$ center vertices needs to be found in a graph $G$ with edge lengths, such that the distance from any vertex of $G$ to its nearest center is minimized. This problem naturally occurs in transportation networks, and therefore we model the inputs as graphs with bounded highway dimension, as proposed by Abraham et al. [SODA 2010].

We show both approximation and fixed-parameter hardness results, and how to overcome them using fixed-parameter approximations, where the two paradigms are combined. In particular, we prove that for any $\varepsilon > 0$ computing a $(2-\varepsilon)$-approximation is W[2]-hard for parameter $k$, and NP-hard for graphs with highway dimension $O(\log^2 n)$. The latter does not rule out fixed-parameter $(2-\varepsilon)$-approximations for the highway dimension parameter $h$, but implies that such an algorithm must have at least doubly exponential running time in $h$ if it exists, unless the ETH fails. On the positive side, we show how to get below the approximation factor of 2 by combining the parameters $k$ and $h$: we develop a fixed-parameter $3/2$-approximation with running time $2^{O(kh\log h)} \cdot n^{O(1)}$.

Karel Ha

mathemage@gmail.com

Presented paper by David Silver, Aja Huang, Demis Hassabis et al.

# Mastering the game of Go with deep neural networks and tree search

(http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html )
(a copy available at http://kam.mff.cuni.cz/ spring/2016/papers/go.pdf)

## Introduction

The game of Go has long been viewed as the most challenging of classic games for artificial intelligence owing to its enormous search space and the difficulty of evaluating board positions and moves. A new approach to computer Go introduces *value networks* to evaluate board positions and *policy networks* to select moves. These deep neural networks are trained by a novel combination of supervised learning from human expert games, and reinforcement learning from games of self-play.

Furthermore, a new search algorithm is introduced: it combines Monte Carlo simulation with value and policy networks. Using this search algorithm, the computer program AlphaGo developed by Google DeepMind achieved a 99.8 % winning rate against other Go programs.

## Supervised learning

A (machine learning) program is "trained" on a pre-defined dataset. Based on its training data, the program can make accurate decisions when given new data.

**Training** is the phase, when the model "learns", i.e. it optimizes the error function by adjusting inner parameters. The portion of the dataset used for training is called the *training set*.

**Testing** is the phase evaluating, how well the model can make predictions about unseen data. The portion of the dataset used for testing is called the *testing set*.

**Classification** is is the process of taking some sort of input and assigning a label to it. Classification systems are often used when predictions are of a discrete nature, e.g. "yes or no", "spam or regular email", "winning or losing position" etc.

**Regression** is used when the predicted value falls on a continuous spectrum. It is a "continuous counterpart" of classification, used to answer questions of "How much? How many?" nature.

**Gradient descent** (GD) is an iterative optimization algorithm to find a (generally local) minimum of an error function. It moves from the current point to a new one, by taking steps proportional to the negative of the gradient (or the approximate gradient).

**Stochastic gradient descent** (SGD) is a variant of GD, which updates the parameters in each iteration by using only one sample of the training set. Hence, instead of using the overall error of the training set as in the classic GD, the SGD works with an error of a single sample.

**Overfitting** is a situation where the model learns the training data by heart, instead of learning the big picture. This happens when the model is too complex for the size of the training data, e.g. when using a 100-degree polynomial as a model to fit 100 data points.

Overfitting prevents the model from generalizing to new data.

### Reinforcement learning

Reinforcement learning allows machines and software agents to automatically determine the ideal behavior based on the feedback from the environment, in order to maximize the performance. This automated learning scheme implies that there is little need for a human expert in the domain of application.

**Self-play** means learning from game matches played by the agent against itself (or in the case of AlphaGo, against its previous versions). Self-play is especially useful when the evaluation function of the game is unknown in advance.

### Game-tree search

**Monte Carlo tree search** (MCTS) is a Monte Carlo heuristic of the classical tree search. However, instead of traversing the entire game tree, the MCTS selects the most promising moves, expanding the search tree based on random sampling.

In each iteration, the game is played-out to the very end by choosing moves at random. The final outcome of each playout is then used to weight the nodes in the game tree accordingly. Thus, better nodes are more likely to be chosen in future playouts.

### Neural networks

Inspired by biological neural networks, an artificial neural network (ANN) is a network of interconnected nodes that make up a model. ANNs can be defined as statistical learning models that are used to approximate functions which depend on a large number of inputs. Neural networks are typically used when the volume of inputs is far too large for standard machine learning approaches.

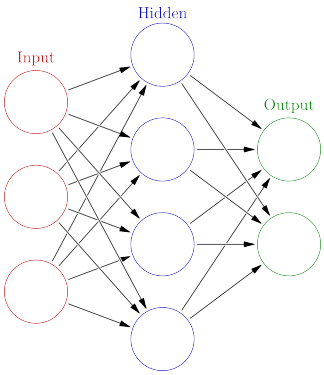



Figure 3: A shallow neural network with 3 layers (input, hidden and output)

**Convolutional neural network** (CNN) is a neural network suitable for high-dimensional inputs (e.g. a large number of pixels in an image). CNNs are frequently used in computer vision (for identifying objects in an image, for face detection in photos etc.).

They are invariant to expectable transformations of input, such as translations of objects in a picture or changes in illumination.

**Deep neural network** (DNN) is a neural network with many hidden layers. It can model complex non-linear relationships, e.g. in speech, in images, in videos or in board positions of Go.

### Rules of Go

*Black* and *White* place pieces (*stones*) on the unoccupied intersections (*points*) of a *board* with a $19 \times 19$ grid of lines. Players take turns, Black moves first. There are only 2 basic rules of Go:

**The rule of liberty** Every stone remaining on the board must have at least one open point (an intersection, called a *liberty*) directly next to it (up, down, left, or right), or must be part of a connected group that has at least one such liberty next to it.

Stones or groups of stones which lose their last liberty are removed from the board.

**The "ko" rule** The stones on the board must never repeat a previous position of stones. This is to prevent unending cycles.

There are several scoring rules to determine the winner of a game. In the match against Lee Sedol, the *area scoring* was used: a player's score is the number of player's stones on the board, plus the number of empty intersections surrounded by that player's stones.

**AlphaGo**

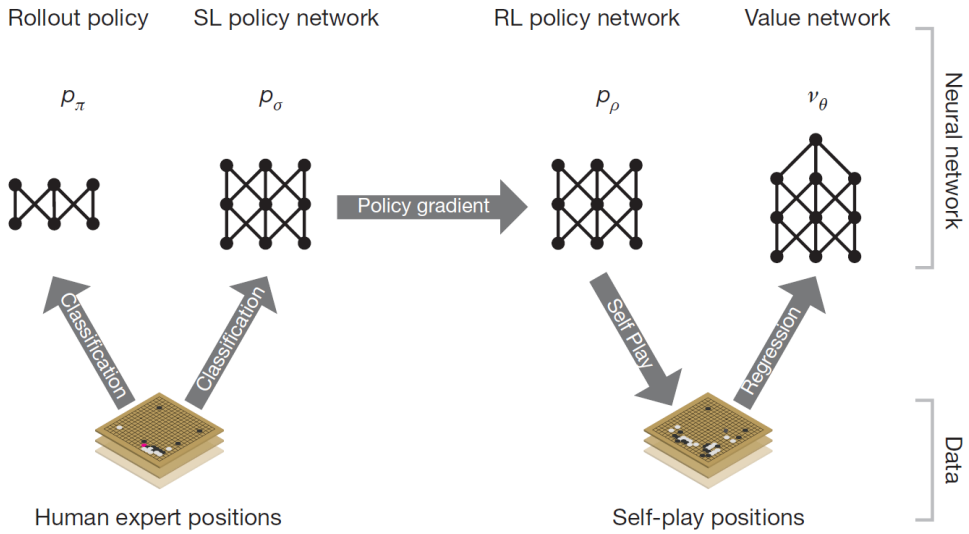



Figure 4: Training the neural networks: the pipeline and the architecture

**Rollout policy** $p_\pi$ is a CNN rapidly sampling actions during a *rollout* (a fast-forward simulation from a position to the end of the game). It predicts expert human moves much faster but less accurately than $p_\sigma$ (see below).

**Policy network** is a CNN selecting moves. It addresses the problem of the game-tree breadth.

**SL policy network** $p_\sigma$ is trained by supervised learning to predict expert human moves.

**RL policy network** $p_\rho$ is trained by reinforcement learning to win in the games of self-play.

**Value network** $v_\theta$ is a CNN evaluating board positions, so as to address the problem of the game-tree depth. It is trained by regression to predict the outcome in positions of the self-played games.

AlphaGo combines the policy and value networks with the MCTS, thus achieving the main result of the article:

**Theorem 1.** *The (distributed version of) AlphaGo plays Go at the super-human level.*

**Proof** The proof is left as an exercise to the reader. This exercise consists of making an effort to follow the slides. :-) □

Jaroslav Hančl

jarda.hancl@gmail.com

Presented paper by Samuel Zbarsky

# The maximum number of subset divisors of a given size

(*Discrete Mathematics 339 (2016) 1727-1733*)

## Introduction

Let $X$ be any finite set of positive integers. We write $\sum X$ for the sum $\sum_{x\in X} x$ of all elements of $X$. Let $B$ be the subset of finite set $A$ of positive integers. We say that $B$ *is a divisor of* $A$ if $\sum B \mid \sum A$. We define $d_k(A)$ to be the number of $k$-subset divisors of $A$ and $d(k,n) = \max d_k(A)$ be the maximum value of $d_k(A)$ over all $n$-sets $A$ of positive integers.

Similarly, for positive integer $s \geq 1$, we say that $B$ *is an* $s$*-divisor of* $A$ if $\sum B \mid s\sum A$. We define $d_k^s(A)$ to be the number of $k$-subset $s$-divisors of $A$ and $d^s(k,n) = \max d_k^s(A)$ be the maximum over all sets $A$ of $n$ positive integers.

We are interested in finding the values $d(k,n)$ and $d^s(k,n)$ for any couple $(k,n)$ of positive integers satisfying $k \leq n$. Let $a_1 < a_2 < \cdots < a_n$ be the elements of $A$. Huynh notes that for any choice of $a_1, a_2, \ldots, a_{n-1}$ one can find $a_n$ such that every $k$-subset of $\{a_1, a_2, \ldots, a_{n-1}\}$ is a divisor of $A$. This motivates following conjecture

**Conjecture 1.** *For all but finitely many values of* $k$ *and* $n$ *we have* $d(k,n) = \binom{n-1}{k}$.

Unfortunately, this conjecture is not true since $d(n,n) = 1$ and $d(1,n) = n$. However, in all other cases we can declare victory since

**Theorem 2.** *For all but finitely many pairs* $(k,n)$ *satisfying* $1 < k < n$ *we have* $d(k,n) = \binom{n-1}{k}$.

In case of $s$-divisors the situation slightly changes, because the case $k = n-1$ has to be removed and one can only prove

**Theorem 3.** *For all but finitely many pairs* $(k,n)$ *satisfying* $1 < k < n-1$ *we have* $d^s(k,n) = \binom{n-1}{k}$.

## The structure of the proof

For convenience, we rescale our situation by dividing every element of $A$ by a factor $\sum A$. Hence, our elements of $A$ are positive rationals and $\sum A = 1$. Therefore, any set $B \subset A$ is a divisor of $A$ whenever $\sum B = 1/m$ for some positive integer $m$, and is an $s$-divisor of $A$ whenever $\sum B = s/m$. Clearly, $d(k,n)$ and $d^s(k,n)$ does not change.

The observation of Huynh motivates following definition. We say that any set $A$ *is an* $k$*-antipencil* if the set of $k$-subset divisors of $A$ is $\binom{A\setminus\{a_n\}}{k}$. Similarly, any set $A$ *is an* $(k,s)$*-antipencil* if the set of $k$-subset $s$-divisors of $A$ is $\binom{A\setminus\{a_n\}}{k}$.

Now we can reveal the structure of the proof. We prove scaled Theorem 3 in four steps thanks to following four propositions. Theorem 2 follows from Theorem 3 in case $1 < k < n-1$, moreover, the case $k = n-1$ is an easy exerise.

**Proposition 4.** *For all* $k \geq 2$ *there exists* $n_0 = n_0(k,s)$ *such that for all* $n \geq n_0$*, if a set* $A$ *of positive rational numbers with* $|A| = n$ *and* $\sum A = 1$ *satisfies* $d^s(A) \geq \binom{n-1}{k}$ *then* $A$ *is an* $(k,s)$*-antipencil.*

**Proposition 5.** *There exists* $k_1 = k_1(s)$ *such that for all pairs* $(k,n)$ *with* $k \geq k_1$ *and* $k \leq 2n/3$*, if a set* $A$ *of positive rational numbers with* $|A| = n$ *and* $\sum A = 1$ *satisfies* $d^s(A) \geq \binom{n-1}{k}$ *then* $A$ *is an* $(k,s)$*-antipencil.*

**Proposition 6.** *There exists* $k_2 = k_2(s)$ *such that for all pairs* $(k,n)$ *with* $k \geq k_2$ *and* $2n/3 < k < n - (6s^2+3s)^2$*, if a set* $A$ *of positive rational numbers with* $|A| = n$ *and* $\sum A = 1$ *satisfies* $d^s(A) \geq \binom{n-1}{k}$ *then* $A$ *is an* $(k,s)$*-antipencil.*

**Proposition 7.** *There exists* $k_3 = k_3(s)$ *such that for all pairs* $(k,n)$ *with* $k \geq k_3$ *and* $n - (6s^2+3s)^2 \leq k < n-1$*, if a set* $A$ *of positive rational numbers with* $|A| = n$ *and* $\sum A = 1$ *satisfies* $d^s(A) \geq \binom{n-1}{k}$ *then* $A$ *is an* $(k,s)$*-antipencil.*

Define $K = \max(k_1, k_2, k_3)$ and $N = \max_{k\leq K} n_0(k,s)$. For any pair $(k,n)$ with $1 < k < n-1$, if $k \geq K$ then Theorem

3 follows from Proposition 5, Proposition 6 and Proposition 7. If $k < K$ and $n \geq N$ then Theorem 3 is a consequence of Proposition 4. Therefore, we are left with finitely many cases of pairs $(k,n)$, particularly the pairs with $k < K$ and $n < N$, which proves Theorem 3.

**Auxiliary lemma**

Let $A,B$ be two $k$-subsets of finite set $X$ of positive integers. Denote by $a_1 \leq a_2 \leq \cdots \leq a_k$ the elements of $A$ and by $b_1 \leq b_2 \leq \cdots \leq b_k$ the elements of $B$. We say that $A \leq_I B$ if $a_i \leq b_i$ for any $i \in [k]$.

**Lemma 8.** *Fix $d > 1$. Then the size of the largest antichain of the partial order $\leq_I$ is less than $\frac{2}{n\sqrt{d}} \left| \binom{X}{d} \right|$ for any sufficiently large $n$.*

**Lemma 9.** *Fix positive integers $k,m,a,b$. Then for positive integers $n$, the number of pairs of positive integers $(x,y)$ such that*

$$\frac{m}{n} = \frac{a}{x} + \frac{b}{y}$$

*and all three fractions are in lowest terms is at most $O(n^{1/k})$.*

Tereza Hulcová

tereza.hulcova@gmail.com

Presented paper by Vadim Lozin, Raffaele Mosca, Viktor Zamaraev

# Independent Domination versus Weighted Independent Domination

(http://arxiv.org/abs/1602.09124)

## Introduction

Independent domination is one of the problems for which the complexity of weighted and unweighted version is known to be different in some classes of graphs. For instance, it has been proven that weighted independent domination is NP-hard on chordal graph, but the unweighted case is easy. We will prove that WID is NP-hard on a subclass of chordal graphs, and as a secondary result we will show that it is easy on a different class of graphs.

## More formally

**Observation 1.** *Independent domination problem is NP-hard.*

**Definition 2.** Graph $G$ is called *sat-graph* if there exists a partition $A \cup B = V(G)$ such that

1. $A$ is a clique.
2. $B$ is an induced matching.
3. There are no triangles $b, b', a$, where $a \in A$ and $b, b' \in B$.

**Theorem 3.** *The WID problem is NP-hard in the class of $(C_4, Sun_3)$-free sat-graphs.*

**Fact 4.** *ID is NP-hard in the class of $(C_3, C_4, C_5, C_6)$-free graphs.*

**Proof** Reduction ID in $(C_3, C_4, C_5, C_6)$-free graphs to WID in $(C_4, Sun_3)$-free sat graphs. □

**Theorem 5.** *The WID problem is polynomially solvable in $(P_5, \overline{P_5})$-free graphs.*

**Definition 6.** Let $G = (V, E)$ be a graph. A vertex set $M \subseteq V$ is called *module* of $G$ if every vertex outside $M$ is adjacent either to each vertex in $M$ or to no vertex.

**Definition 7.** A vertex $v$ is *good* if WID is solvable in graph $G - N(v)$ in polynomial time.

**Lemma 8.** *The decision tree $T(G)$ has $O(n^2)$ vertices.*

**Lemma 9.** *Module in the graph G can be found in polynomial time.*

**Lemma 10.** *In $(P_5, \overline{P_5})$-free graphs the good vertex can be found in polynomial time.*

Adam Kabela

kabela@kma.zcu.cz

Presented paper by Xiang-Ke Chang, Xing-Biao Hu, Hongchuan Lei and Yeong-Nan Yeh

# Combinatorial proofs of addition formulas

## Introduction

The paper presents an addition formula for weighted partial Motzkin paths. As the main result, combinatorial proof of this formula is given. It is shown that one can apply this formula to evaluate Hankel matrices defined by sequences related to these paths.

More applications of the formula for evaluating determinants of various Hankel matrices are presented, also a similar addition formula for weighted large Schröder paths is given.

## More formally

Let $\mathscr{P}_n^k$ denote the set of lattice paths from $(0,0)$ to $(n,k)$ and never going below the x-axis, consisting of up steps $(1,1)$, horizontal steps $(1,0)$ and down steps $(1,-1)$, and let $\mathscr{P}_n^{\geq r}$ denote such paths ending in $(n,k)$ for any $k \geq r$.

The paths $\mathscr{P}_n^0$ are *Motzkin paths*. The number of distinct Motzkin paths ending in $(n,0)$ is the *nth Motzkin number* (see A001006 in [1]). The paths $\mathscr{P}_n^k$ with a weight assigned to each step are *weighted partial Motzkin paths*. The weights considered in the paper are $w(1,1) = xy$, $w(1,0) = ux$, $w(1,-1) = vx/y$ and $u^s v^t x^n y^k$ is the weight of the path, where $s$ is the number of horizontal steps and $t$ is the number of down steps.

The weight of the set of lattice paths is the sum of weights of all its paths. Let

$$a_{n,k} := \frac{1}{x^n y^k} w(\mathscr{P}_n^k).$$

For the case $u = v = 1$, $a_{n,0}$ is the nth Motzkin number, and the matrix $(a_{i,j})_{0 \leq i,j \leq n-1}$ is the Motzkin triangle (see A026300 in [1]). Let

$$a_{n,r}(y) := \frac{1}{x^n y^r} w(\mathscr{P}_n^{\geq r}),$$

So $a_{n,r}(y) = \sum_{k \geq r} a_{n,k} y^{k-r}$, here $y$ marks the distance from the end point of each path to the line $y = r$.

**Theorem 1.** *For integers $m \geq 0$ and $n \geq 0$, we have*

$$a_{m+n,0}(y) = \sum_{r=0}^{min(m,n)} v^r a_{m,r}(y) a_{n,r}(y).$$

**Proof** For a lattice path $P = (P_0, P_1, ..., P_{m+n})$ of $\mathscr{P}_{m+n}^k$, we consider a decomposition of $P$ to $L_m(P)$, $M_m(P)$ and $R_m(P)$, where $L_m(P) = (P_0, P_1, ..., P_m)$, $M_m(P) = (P_m, P_{m+1}, ..., Pj)$, $R_m(P) = (P_j, P_{j+1}, ..., P_n)$ such that $P_j$ is the rightmost point of $(P_m, P_{m+1}, ..., P_n)$ with the smallest $y$ coordinate. We let $\mathscr{Q}_r$ denote the set of all paths in $\mathscr{P}_{m+n}^{\geq 0}$ such that the number of down steps minus the number of up steps in $M_m(P)$ equals $r$.

We consider bijection $\varphi$ that assigns to every $P$ a path obtained by reversing the subpath $M_m(P)$ in $P$, and we note that

$w(P) = \frac{v^r}{y^{2r}} w(\varphi(P))$. In other words, $w(\mathscr{Q}_r) = \frac{v^r}{y^{2r}} w(\mathscr{P}_m^{\geq 0}) w(\mathscr{P}_n^{\geq 0})$. Thus,

$$\begin{aligned} a_{m+n,0}(y) &= \frac{1}{x^{m+n}} w(\mathscr{P}_{m+n}^{\geq 0}) \\ &= \frac{1}{x^{m+n}} \sum_{r=0}^{min(m,n)} w(\mathscr{Q}_r) \\ &= \sum_{r=0}^{min(m,n)} v^r \frac{1}{x^m y^r} w(\mathscr{P}_m^{\geq r}) \frac{1}{x^n y^r} w(\mathscr{P}_n^{\geq r}) \\ &= \sum_{r=0}^{min(m,n)} v^r a_{m,r}(y) a_{n,r}(y). \end{aligned}$$

□

For a sequence $\{a_n\}_{n\geq 0}$, its *nth Hankel matrix* is the matrix $(a_{i,j})_{0\leq i,j\leq n-1}$, where $a_{i,j} = a_{i+j}$.

**Corollary 2.** *The determinant of the nth Hankel matrix of* $\{a_{n,0}(y)\}_{n\geq 0}$ *equals* $v^{\frac{n(n-1)}{2}}$.

**Proof** Let $A_n(y)$ denote the nth Hankel matrix. By Theorem 1, $A_n(y) = A \cdot D \cdot A^T$, where $D = diag(v^0, v^1, ..., v^{n-1})$ and

$$A = \begin{pmatrix} a_{0,0}(y) & 0 & 0 & ... & 0 \\ a_{1,0}(y) & a_{1,1}(y) & 0 & ... & 0 \\ ... & & & & \\ a_{n-1,0}(y) & a_{n-1,1}(y) & a_{n-1,2}(y) & ... & a_{n-1,n-1}(y) \end{pmatrix}.$$

We note that $a_{i,i}(y) = a_{i,i} = 1$ for $0 \leq i \leq n-1$, and we obtain the following:

$$\det(A_n(y)) = \det D \cdot (\det A)^2 = v^{\frac{n(n-1)}{2}} (a_{0,0}(y) \cdot a_{1,1}(y) \cdot ... \cdot a_{n-1,n-1}(y))^2 = v^{\frac{n(n-1)}{2}}.$$

□

## References

[1] N. J. A. Sloane: On-line Encyclopedia of Integer Sequences (OEIS). Published electronically at http://oeis.org/, 2014.

Mark Karpilovskij

m.karpilovsky@email.cz

Presented paper by N. Bousquet, A. Lagoutte, S. Thomassé

# The Erdős-Hajnal conjecture for paths and antipaths

(*Journal of Combinatorial Theory, Series B 113 (2015) 261-264*)

## Introduction

The famous Erdős-Hajnal conjecture states that for every strict class of graphs closed under induced subgraphs there is a constant $c > 0$ such that every $n$-graph in the class contains a clique or an independent set of size at least $n^c$.

The presented paper solves this conjecture for the class $\mathscr{C}_k$ of graphs which do not induce a path of length $k$ or its complement.

## More formally

An *n-graph* is a simple undirected graph on $n$ vertices. For a graph $G$ we denote its complement by $\overline{G}$. A complete subgraph of $G$ is a *clique* in $G$ and a complete subgraph of $\overline{G}$ is an *independent set* in $G$. For a vertex $v$ we denote by $N(v)$ its neighborhood and $deg(v) = |N(v)|$ denotes its degree. We consider only classes of graphs closed under induced subgraphs and we call such a class *strict* if it does not contain all graphs.

**Definition 1.** A strict class of graphs $\mathscr{C}$ has the *Erdős-Hajnal property* if there exists a constant $c > 0$ such that every $n$-graph in $\mathscr{C}$ contains either a clique of size at least $n^c$ or an independent set of size at least $n^c$.

**Definition 2.** For an integer $k \geq 2$ let $\mathscr{C}_k$ be the class of all graphs not inducing the path $P_k$ or its complement $\overline{P}_k$.

The main result of the article is the following theorem.

**Theorem 3.** *The class $\mathscr{C}_k$ has the Erdős-Hajnal property for every integer $k \geq 2$.*

We say that an $n$-graph is an $\varepsilon$-*independent set* if it contains at most $\varepsilon\binom{n}{2}$ edges and it is an $\varepsilon$-*clique* if it is a complement of an $\varepsilon$-independent set. To prove Theorem 3, the authors used the following result of Fox and Sudakov.

**Theorem 4.** *For every positive integer $k$ and every $\varepsilon \in (0, 1/2)$ there is a $\delta > 0$ such that every $n$-graph $G$ satisfies one of the following:*

1. *$G$ induces all graphs on $k$ vertices.*
2. *$G$ contains an $\varepsilon$-independent set of size at least $\delta n$.*
3. *$G$ contains an $\varepsilon$-clique of size at least $\delta n$.*

A subgraph $H$ of a graph $G$ is a biclique of size $t$ if it is a (not necessarily induced) complete bipartite graph with each part consisting of at least $t$ vertices. The class $\mathscr{C}$ of graphs then has the *strong Erdős-Hajnal property* if there is a constant $c$ such that every $n$-graph in $\mathscr{C}$ contains a biclique of size at least $cn$. As the name suggests, the strong Erdős-Hajnal property implies the (weak) Erdős-Hajnal property.

**Theorem 5.** *Let $\mathscr{C}$ be a class of graphs with the strong Erdős-Hajnal property. Then $\mathscr{C}$ has the Erdős-Hajnal property.*

It turns out that it is easier to prove the strong Erdős-Hajnal property for the class $\mathscr{C}_k$ than to prove the basic one directly. The authors do so using the following key lemma.

**Lemma 6.** *For every integer $k \geq 2$ there exist numbers $\varepsilon_k > 0$ and $0 < c_k \leq 1/2$ such that every connected $n$-graph $G$ with $n \geq 2$ vertices satisfies at least one of the following:*

1. *$G$ has a vertex of degree more than $\varepsilon_k n$.*
2. *For every vertex $v$ of $G$ there is an induced $P_k$ starting at that vertex.*
3. *The complement $\overline{G}$ contains a biclique of size at least $c_k n$.*

Finally, using Theorems 4 and 5 and Lemma 6, the authors give a short proof of their main result.

**Theorem 7.** *The class $\mathscr{C}_k$ has the strong Erdős-Hajnal property.*

Pavel Klavík

klavik@kam.mff.cuni.cz

# Visual Complex Analysis

Visual Complex Analysis, based on a book of the same name by Tristam Needham, and this link: `http://pavel.klavik.cz/orgpad/complex_analysis.html`.

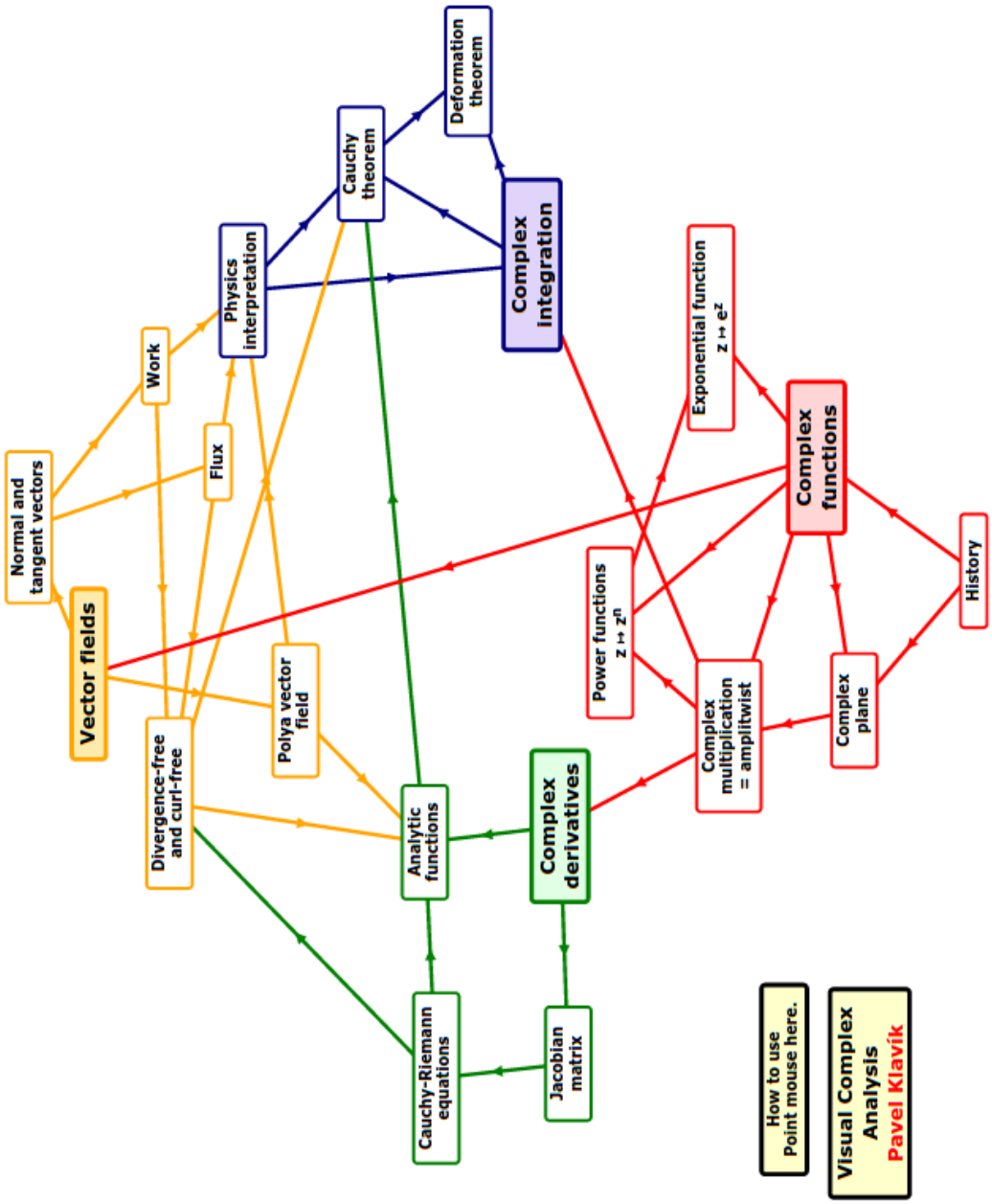



Figure 5: Orgpad: Visual Complex Analysis

Dušan Knop

knop@kam.mff.cuni.cz

Presented paper by András Frank and Éva Tardos

# An application of simultaneous diophantine approximation

(http://link.springer.com/article/10.1007%2FBF02579200)

## Introduction

The aim of the paper is to show how to use simultaneous diophantine approximation to

- design a preprocessing routine, that will
- turn weakly polynomial algorithms into strongly polynomial ones.

We will present the routine and show some application – namely showing a result that rounding routine for a problem

$$\max w^T x \colon Ax \leq b$$

**Theorem 1.**

1. *$x \in P$ is w-maximal if and only if it is $\overline{w}$-maximal.*
2. *A set of rows of A is an optimal dual basis for* $\max(w^T x \colon Ax \leq b)$ *if and only if it is an optimal basis for* $\max(\overline{w}^T x \colon Ax \leq b)$.

## Definitions and tools

**Theorem 2.***[Simultaneous diophantine approximation] Given a positive integer N and n real numbers $\alpha_1, \alpha_2, \ldots, \alpha_n$ there are integers $p_1, p_2, \ldots, p_n$ and q such that $1 \leq q \leq N^n$ and*

$$|q\alpha_i - p_i| < \frac{1}{N}, \qquad \text{for every } i = 1,2,\ldots,n.$$

*Moreover, if $0 \leq \alpha_1, \alpha_2 \ldots, \alpha_n \leq 1$ then there is a strongly polynomial-time algorithm achieving worser bound $1 \leq q \leq 2^{n^2+n}N^n$.*

## Preprocessing routine

**Theorem 3.** *For every vector $w \in \mathbb{R}^n$ and positive integer N there exists integer vectors $v_1, v_2, \ldots, v_k$ ($k \leq n$) and positive reals $\lambda_1, \lambda_2, \ldots, \lambda_k$ such that*

1. $w = \sum_{i=1}^{k} \lambda_i v_i$,
2. $\|v_i\|_\infty \leq N^n \qquad i = 1,2,\ldots,k$ *and*
3. $\frac{\lambda_i}{\lambda_{i-1}} \leq \frac{1}{N\|v_i\|_\infty} \qquad i = 2,3,\ldots,k.$

## Algorithm

**Input:** A rational vector $w$ and an integer $N$.

**Output:** An integral vector $\overline{w}$ such that $\|\overline{w}\|_\infty \leq 2^{4n^3} N^{n(n+2)}$ and $sign\, w \cdot b = sign\, \overline{w} \cdot b$ whenever $b$ is an integer vector with $\|b\|_1 \leq N-1$.

Peter Korcsok

korcsok@iuuk.mff.cuni.cz

Presented paper by Alexander Igamberdiev, Wouter Meulemans, André Schulz

# Drawing Planar Cubic 3-Connected Graphs with Few Segments

(http://link.springer.com/chapter/10.1007/978-3-319-27261-0_10)

## Introduction

Given a graph, there are many possibilities for the actual drawing of the graph into the plane. We are also trying to find as simple visualization as possible – e.g. we want to avoid crossings and bends of the edges. In this talk, we are trying to draw the graph using not too many geometric objects – e.g. when we want to draw a path in the graph, the vertices can be placed along one line, that can represent all the edges, and we do not need an unique line segment for each edge.

The aim of this talk is to introduce three algorithms for finding a plane drawing of a planar cubic 3-connected graph using $3n+3$ line segments ($n$ is the number of vertices):

1. the Deconstruction algorithm (DEC)
2. the Windmill algorithm (WIN)
3. a revision and modification of Mondal's algorithm [1] (MON)

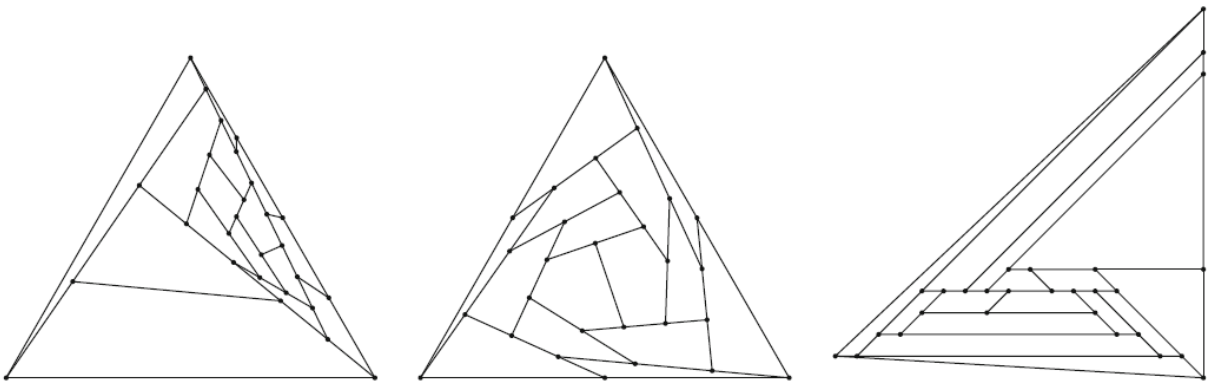


Figure 6: Result of the algorithms DEC, WIN, and MON, respectively, for the same graph and outer face.

In the end of this talk, a comparison of the algorithms is given.

## References

[1] Mondal, D., Nishat, R., Biswas, S., Rahman, S.: Minimum-segment convex drawings of 3-connected cubic plane graphs. J. Comb. Opt. **25**, 460–480 (2013)

Stanislav Kučera

kucerast@iuuk.mff.cuni.cz

Presented paper by Takayasu Kuwata, Hiroshi Maehara

# Another exploration problem

(http://www.sciencedirect.com/science/article/pii/S0012365X16000030)

## Introduction

It's a hot summer day and we follow a group of divers as they try to dive as deep into an ocean as possible, while holding breath. For small groups we will determine the exact value of the deepest point at least one of them can reach. For a bit bigger groups as well as for a group of general size we show some bounds. The exercises 10-15 are (possibly still open) problems listed at the end of the paper.

## More formally

The capacity of lungs of each member is exactly one liter of air that allows the diver to hold the breath for exactly two minutes. To swim one unit distance she needs one half of a liter of air.

**Definition 1.** Let $d(n)$ denote the maximum depth at least one of the members of a party having $n$ members can reach.

**Definition 2.** A *helper* is a member who meets the last group on their way back and has a larger amount of food than each member of the group when they meet.

**Definition 3.** A *supporter* is a member of the group which first dived, who separates from the others midway to the destination leaving them some extra air.

**Lemma 4.** *If an expedition party can reach the destination at depth d, then there is a feasible plan for them to reach the same depth d that satisfies the following (a)–(d).*

*(a) Just one member reaches the destination at depth d. Let A stand for this member.*

*(b) The diver A dives with the first group, and returns last. It takes* $2d$ *days to finish A's journey.*

*(c) The last group of members on their way back never meet a helper while their air remains.*

*(d) When a supporter separates from other ongoing members, the supporter leaves air to the others so that each of them has one liter of air.*

**Theorem 5.** $d(1) = 1$; $d(2) = 5/3$; $d(3) = 2$; $d(4) \geq 2 + \frac{41}{90}$.

**Conjecture 6.** $d(4) = 2 + \frac{41}{90}$

**Theorem 7.** $d(5) \geq 2 + 3/5$, $d(6) \geq 2 + 4/5$

**Theorem 8.** *If* $n \geq 2^k - 1$ *then* $d(n) \geq 2\left(\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{k+1}\right)$. *Therefore* $\lim_{n\to\infty} d(n) = \infty$.

**Theorem 9.** $d(n) < 2\left(\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{n}\right) + 1$

**Exercise 10.** *Find an algorithm to make an efficient plan for n divers or to achieve the depth d.*

**Exercise 11.** *Prove (or disprove)* $d(4) = 2 + \frac{41}{90}$.

**Exercise 12.** *Determine* $d(5)$ *and* $d(6)$.

**Exercise 13.** *Improve the bounds in Theorem 11 and Theorem 12.*

**Exercise 14.** *Is* $d(n)$ *strictly increasing (that is,* $d(n) < d(n+1)$ *for every* $n > 0$*)?*

**Exercise 15.** *Can* $d(n)$ *always be achieved a symmetric plan for every* $n > 0$*?*

Robert Lukotka

lukotka@dcs.fmph.uniba.sk

Presented paper by Gunnar Brinkmann, Jan Goedgebeur, Jonas Hägglund, Klas Markström

# Generating snarks

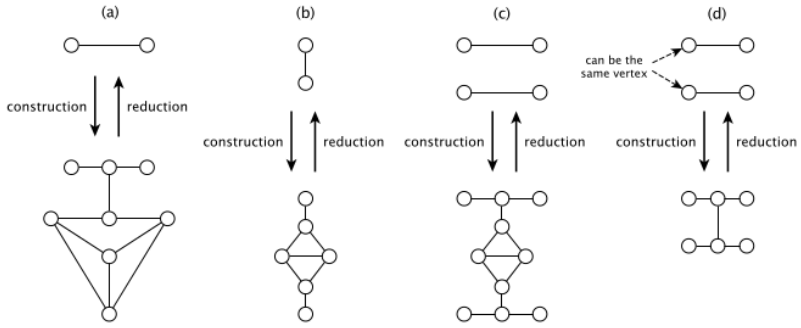
(http://arxiv.org/abs/1206.6690 and http://arxiv.org/abs/1512.05944)

## Introduction

We show how to generate non-isomorphic cubic graphs of given order (feasible up to 30 vertices). The generated graphs are used to produce complete lists of various interesting subclasses of cubic graphs. The talk is based on [3] and [2].

## More formally

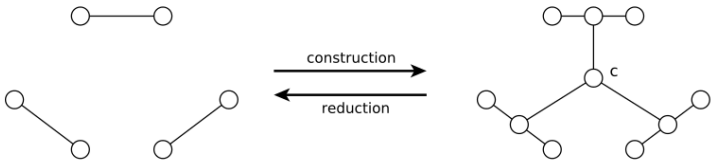
**Lemma 1.** *All cubic graphs can be generated using four operations (the figure is from [1]).*




*After certain point, opperation (d) is sufficient*

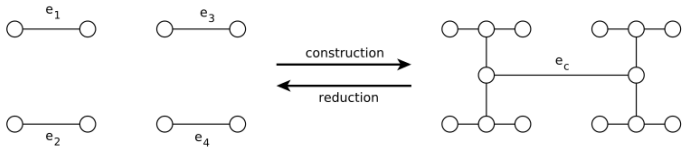
We describe how to avoid creating isomorphic copies of the same graph. We will describe strategies that allow us to cut branches that cannot produce snarks. The following strategies allow us to generate larger graphs of high girth.

**Lemma 2.** *Every connected cubic graph $G$ with girth $k \geq 5$ can be reduced using the following reduction: (the figure is from [2]).*




*The obtained graph has girth at least $k-1$ and at most 3 disjoint cycles of length $k-1$.*

**Lemma 3.** *Every connected cubic graph $G$ with girth at least $k \geq 5$ can be reduced using the following reduction: (the figure is from [2]).*



*The girth of the reduced graph is at least $k-2$ if $k \in \{5,6\}$ and at least $k-1$ otherwise.*

## References


[1] G. Brinkmann, J. Goedgebeur, J. Hägglund, K. Markström: Generation and Properties of Snarks, arXiv:1206.6690 [math.CO], 2013.

[2] G. Brinkmann and J. Goedgebeur: Generation of cubic graphs and snarks with large girth, arXiv:1512.05944 [math.CO], 2015.

[3] G. Brinkmann, J. Goedgebeur, J. Hägglund, K. Markström: Generation and Properties of Snarks, Journal of Combinatorial Theory, Series B 103, 468–488, 2013

Tomáš Masařík

masarik@kam.mff.cuni.cz

# Backbone coloring

The aim of this talk is to study backbone coloring and to show an elegant proof.

## Introduction

**Definition 1.** Graphs $G(V,E)$ and its spanning subgraph $H(V,E')$ are *backbone k-colorable* ($\mathrm{BCC}(G,H) \leq k$) if there is a mapping $f : V(G) \to \{1,\ldots,k\}$ such that:

- $\forall\{u,v\} \in E$ holds $|f(u) - f(v)| \geq 1$,
- $\forall\{u,v\} \in E'$ holds $|f(u) - f(v)| \geq 2$.

Backbone coloring generalize $L_{2,1}$-labeling which in this context is equivalently defined as $\mathrm{BCC}(G^2,G)$. Both are motivated by frequency assignment problem.

The backbone coloring is also studied for graph $G$ and its spanning tree $T$. The special version aims to find special $T$ such that $\mathrm{BCC}(G,T) = \chi(G)$ where $\chi(G)$ is usual proper coloring of graph $G$.

Filip Mišún

filip.misun@gmail.com

Presented paper by Tri Lai

# Enumeration of antisymmetric monotone triangles and domino tilings of quartered Aztec rectangles

(http://arxiv.org/pdf/1410.8112.pdf)

## Introduction

It has been shown, that both for 2-enumeration of antisymmetric monotone triangles and enumeration of domino tilings of the quartered Aztec rectangles holds the same formula. The goal of this talk is to explain this phenomenon directly by building a correspondence between antisymmetric monotone triangles and domino tilings of the quartered Aztec rectangles.

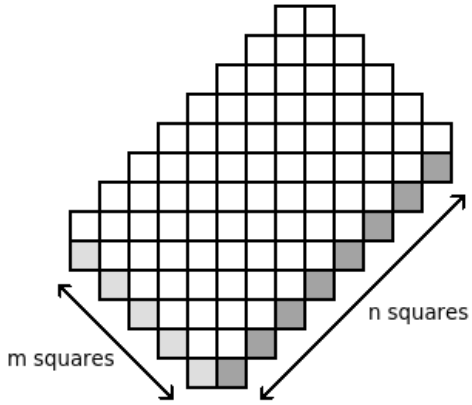
## More formally

**Definition 1.** A *monotone triangle* of order $n$ is a triangular array of integers

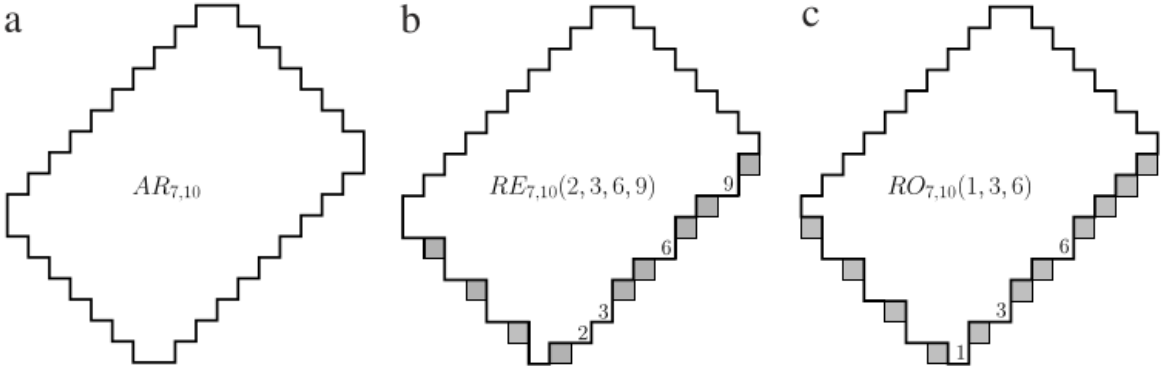
$$\begin{array}{ccccccccccc}
 & & & & a_{1,1} & & & & & & \\
 & & & a_{2,1} & & & a_{2,2} & & & & \\
 & & a_{3,1} & & & a_{3,2} & & & a_{3,3} & & \\
 & \iddots & & & \iddots & \vdots & & \ddots & & & \ddots \\
a_{n,1} & & & a_{n,2} & & \cdots & & & a_{n,n-1} & & a_{n,n}
\end{array}$$

whose entries are strictly increasing along the rows and weakly increasing along both rising and descending diagonals from left to right. An *antisymmetric monotone triangle* (AMT) is a monotone triangle, which has $a_{i,k} = -a_{i,i+1-k}$ for any $1 \leq i \leq n$ and $1 \leq k \leq i$.
For any AMT $\tau$, we denote by $S(\tau)$ the set consisting of all positive entries $a_{i,j}$, which do not appear on the row above. For any given number $q$, we define the *q-weight* of an AMT to be $q^{|S(\tau)|}$.
Assume that $n \geq 2$ and $0 < a_1 < a_2 < ... < a_{\lfloor n/2 \rfloor}$, the *q-enumeration* $A_n^q(a_1, a_2, ..., a_{\lfloor n/2 \rfloor})$ of AMTs is defined as the sum of $q$-weights of all AMTs of order $n$ whose positive entries on the bottomost row are $a_1, a_2, ..., a_{\lfloor n/2 \rfloor}$.

**Definition 2.** An *Aztec rectangle* is a region on the square lattice of such a shape, that is illustrated in the following figure. Aztec rectangle of size $m \times n$ (denoted by $AR_{m,n}$) has $m$ squares along the southwest side and $n$ squares along the southeast side.




Lets take the $AR_{m,n}$ and some $\lfloor (m+1)/2 \rfloor$ integers $1 \leq a_1 < a_2 < ... < a_{\lfloor (m+1)/2 \rfloor} \leq n$. Remove all squares at even positions (from the bottom to top) on the southwest side of $AR_{m,n}$ and remove all the squares on the southeast side except for the squares at positions $a_1, a_2, ..., a_{\lfloor (m+1)/2 \rfloor}$. The region obtained by this process, will be denoted by $RE_{m,n}(a_1, a_2, ..., a_{\lfloor (m+1)/2 \rfloor})$ (see (b) in the following figure).
We also have an odd-analog $RO_{m,n}(a_1, a_2, ..., a_{\lfloor m/2 \rfloor})$ of the above region when removing odd squares (instead of the even ones) from the southwest side, and removing all squares from the southeast side, except for the squares at the positions $a_1, a_2, ..., a_{\lfloor m/2 \rfloor}$ (see (c) in the following figure).




**Definition 3.** Let $R$ be a finite region on the square lattice. A *domino tiling* of $R$ is a covering of $R$ by dominoes such that there are no gaps or overlaps. By $\mathbb{T}(R)$ we denote the number of domino tilings of the region $R$.

**Theorem 4.** *Assume that $k, a_1, a_2, ..., a_k$ are positive integers, such that $a_1 < a_2 < ... < a_k$. The AMTs with positive entries $a_1, a_2, ..., a_k$ on the bottom are 2-enumerated by*

$$A^2_{2k}(a_1, a_2, ..., a_k) = \frac{2^{k^2}}{0!2!4!...(2k-2)!} \prod_{1 \leq i < j \leq k} (a_j - a_i) \prod_{1 \leq i < j \leq k} (a_i + a_j - 1)$$

$$A^2_{2k+1}(a_1, a_2, ..., a_k) = \frac{2^{k^2}}{1!3!5!...(2k-1)!} \prod_{1 \leq i < j \leq k} (a_j - a_i) \prod_{1 \leq i \leq j \leq k} (a_i + a_j - 1)$$

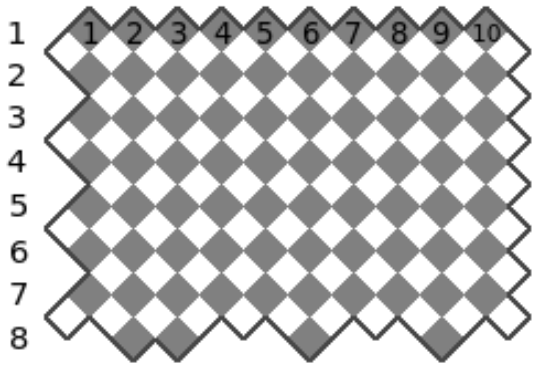
**Theorem 5.** *For any $1 \leq k < n$ and $1 \leq a_1 < a_2 < ... < a_k \leq n$ holds:*

$$\mathbb{T}(RE_{2k-1,n}(a_1, a_2, ..., a_k)) = \mathbb{T}(RE_{2k,n}(a_1, a_2, ..., a_k)) = A^2_{2k}(a_1, a_2, ..., a_k)$$

$$\mathbb{T}(RO_{2k,n}(a_1, a_2, ..., a_k)) = \mathbb{T}(RO_{2k+1,n}(a_1, a_2, ..., a_k)) = A^2_{2k+1}(a_1, a_2, ..., a_k)$$

**Proof** Only the first equation will be proved. We start by some preparations and definitions, which will be used in the proof:

- We always rotate the rectangles by 45° clockwise to help the visualisation of our arguments.
- We also color the squares of the rectangles black and white, so that any two squares sharing an edge have different colors, and that the bottommost squares are black.
- We number all the black rows in $AR_{m,n}$ by $1,2,...,m+1$ from top to bottom. We also label all squares on each black row by $1,2,...,n$ from left to right (we also label all black squares removed on the bottommost row) (the following figure illustrates $RE_{7,10}(2,3,6,9)$ as an example; in the figure, the black squares are labeled only in the first (black) row, however).




- Let $T$ be a tiling of an Aztec rectangle. We say a black square is *matched upward* or *matched downward*, depending on whether the white square covered by the same domino is above or below it.

The proof will proceed in the following steps:

1. We prove $\mathbb{T}(RE_{2k-1,n}(a_1,a_2,...,a_k)) = \mathbb{T}(RE_{2k,n}(a_1,a_2,...,a_k))$. From now on, we only need to show that $\mathbb{T}(RE_{2k-1,n}(a_1,a_2,...,a_k)) = A^2_{2k}(a_1,a_2,...,a_k)$.
2. Denote by $\mathscr{T}(R)$ the set of all tilings of a region $R$, and $\mathscr{A}_n(a_1,a_2,...,a_{\lfloor n/2 \rfloor})$ the set of all AMTs of order $n$ having positive entries $a_1 < a_2 < ... < a_{\lfloor n/2 \rfloor}$ on the bottom. Next, we define a map $\Phi : \mathscr{T}(RE_{2k-1,n}(a_1,a_2,...,a_k)) \to \mathscr{A}_{2k}(a_1,a_2,...,a_k)$ as follows. Pick $T$ from $\mathscr{T}(RE_{2k-1,n}(a_1,a_2,...,a_k))$, we describe $\tau := \Phi(T)$. The positive entries in the $i$th row of $\tau$ are the labels of the matched-upward squares on the $i$th black row (by the antisymmetry, $\tau$ is completely determined by its positive entries).
3. We show that $\Phi$ is well defined, i.e. we need to verify that $\Phi(T) \in \mathscr{A}_{2k}(a_1,a_2,...,a_k)$.
4. We show that $|\Phi^{-1}(\tau)| = 2^{S(\tau)}$ for any $\tau \in \mathscr{A}_{2k}(a_1,...,a_k)$.

□

Jan Musílek

stinovlas@kam.mff.cuni.cz

Presented paper by Thomas D. Hansen, Haim Kaplan, Robert E. Tarjan, Uri Zwick

# Hollow Heaps

(http://arxiv.org/abs/1510.06535)

## Introduction

Hollow heaps are very simple data structure with the same amortized efficiency as the classical Fibonacci heap. All heap operations except delete and delete-min take $O(1)$ worst-case time while delete and delete-nmin take $O(\log n)$ amortized time on a heap of $n$ items. The simplicity of implementation is achieved by introducing two ortogonal concepts – use of lazy deletion and the use of directed acyclic graph instead of tree or set of trees to represent a heap.

## Definitions

**Definition 1.** A *heap* is a data structure consisting of a set of *items*, each with a *key* selected from a totally ordered universe. Heaps support the following operations:

- *make-heap()*: Return a new, empty heap.
- *find-min(h)*: Return an item of minimum key in heap $h$, or null if $h$ is empty.
- *insert(e,k,h)*: Return a heap formed from heap $h$ by inserting item $e$, with key $k$. Item $e$ must be in no heap.
- *delete-min(h)*: Return a heap formed from non-empty heap $h$ by deleting the item returned by *find-min(h)*.
- *meld($h_1$,$h_2$)*: Return a heap containing all items in item-disjoint heaps $h_1$ and $h_2$.
- *decrease-key(e,k,h)*: Given that $e$ is an item in heap $h$ with key greater than $k$, return a heap formed from $h$ by changing the key of $e$ to $k$.
- *delete(e,h)*: Return a heap formed by deleting $e$, assumed to be in $h$, from $h$.

**Definition 2.** A tree is *heap-ordered* if and only if for every arc $(v,w)$ it holds that $v.key \leq w.key$. Heap-order implies that the root has a minimum key.

**Definition 3.** In hollow heaps, we have the items represented as nodes of set of trees and we have a pointer to the minimum root (which we call the minimum node of the heap). We implement heap operation as follows:

- *make-heap()*: Return empty forest.
- *find-min(h)*: Return the minimum node of the heap $h$.
- *meld($h_1$,$h_2$)*: If one is empty, return the other. Otherwise unite their sets of trees and update the minimum node.
- *insert(e,k,h)*: Create a new node, store the item into it and meld the resulting one-node heap with $h$.
- *decrease-key(e,k,h)*: Let $u := e.node$. We create the new node $v$, move $e$ from $u$ to $v$ (making $u$ hollow), set $v.key = k$. Then we move some of the children of $u$, and their subtrees, to $v$. Then we meld this one-root heap with $h$.
- *delete-min(h)*: Delete the item in the minimum node.

- *delete(e,h)*: Let $u := e.node$. We delete $e$ from $u$ making $u$ hollow. If $u$ is the minimum node, we destroy it, turn its children into new roots and make some links to reduce the number of trees. Then meld all the resulting trees into one heap.

### Theorems

**Lemma 4.** ***(Rank-invariant)*** *A node $u$ of rank $r$ has exactly $r$ children, of ranks $0,1,\ldots,r-1$, unless $r > 2$ and $u$ was made hollow by a* decrease-key*, in which case $u$ has exactly two children, of ranks $r-2$ and $r-1$.*

**Theorem 5.** *A node of rank $r$ has at least $F_{r+3} - 1$ descendants, both full and hollow.*

**Corollary 6.** *The rank of a node in a multi-root hollow heap of $N$ nodes is at most $\log_\varphi N$.*

**Theorem 7.** *The amortized time per multi-root hollow heap operation is $O(1)$ for each operation other than a delete or delete-min, and $O(\log N)$ per delete or delete-min on a heap of $N$ nodes.*

Michal Opler

opler@iuuk.mff.cuni.cz

Presented paper by Jessica De Silva, Theodore Molla, Florian Pfender, Troy Retter, Michael Tait

# Increasing paths in edge-ordered graphs: the hypercube and random graphs

(http://arxiv.org/abs/1502.03146v1)

## Introduction

An *edge-ordering* of graph $G = (V,E)$ is a bijection $\varphi : E \to \{1,2,\ldots,|E|\}$. For a given edge-ordering $\varphi$, a sequence of edges $P = e_1, e_2, \ldots, e_k$ is an *increasing path* if it is a path in $G$ which satisfies $\varphi(e_i) < \varphi(e_j)$ for all $i < j$.

What is the maximum length of increasing path that can be guaranteed in edge-orderings of $G$? In this talk we will provide lower bounds for the hypercube $Q_n$ and the random graph $G(n,p)$.

## Groundwork

**Definition 1.** For a graph $G$ let $f(G)$ be the largest integer $l$ such that every edge-ordering of $G$ contains an increasing path of length $l$.

**Pedestrian algorithm($G$, $\varphi$).**

1. Place a distinct marker (pedestrian) on each vertex of $G$.
2. Consider the edges in the order given by $\varphi$. When an edge $e$ is considered, the pedestrians currently at the vertices incident to $e$ switch places if and only if the switch does not cause either pedestrian to move to a vertex it has already traversed.

**Lemma 2.** *Let $G$ be a graph and $k \in \mathbb{Z}^+$. If $f(G) < k$, there exist sets $V_1, V_2, \ldots V_n \subseteq V(G)$ such that $|V_i| \leq k$ and $E(G) \subseteq \cup_{i=1}^{n} E(G[V_i])$.*

**Definition 3.** For a graph $G$ and a positive integer $k$, let $\zeta_k(G)$ be the maximum number of edges induced by any $k$ vertices of $G$, i.e.

$$\zeta_k(G) = \max_{U \in \binom{V(G)}{k}} |E(G[U])|.$$

**Lemma 4.** *Let $G$ be any connected graph with average degree $d$. If $G$ and $k \in \mathbb{N}$ satisfy $2\zeta_k(G) - k + 1 < d$, then $f(G) \geq k$.*

**Corollary 5.** *If $G$ is any graph with average degree $d$ then $f(G) \geq \sqrt{d}$.*

## Main results

**Theorem 6.** *Let $Q_d$ denote the $d$-dimensional hypercube. For all $d \geq 2$*

$$\frac{d}{\log_2 d} \leq f(Q_d) \leq d.$$

**Theorem 7.** *For any function $\omega(n) \to \infty$ and any $p \leq \frac{\log n}{\sqrt{n}} \omega(n)$, with high probability*

$$f(G(n,p)) \geq \frac{(1-o(1))np}{\omega(n)\log n}.$$

Veronika Slívová

veronika.slivova@gmail.com

# Chess: Retrograde analysis

## Introduction

Chess retrograde analysis is a type of problem, where solver must determine, which moves were played to reach the given position. Following diagrams are examples of retrograde analysis problems. Stars correspond to dificulty of the problem (more stars ↔ more difficult).

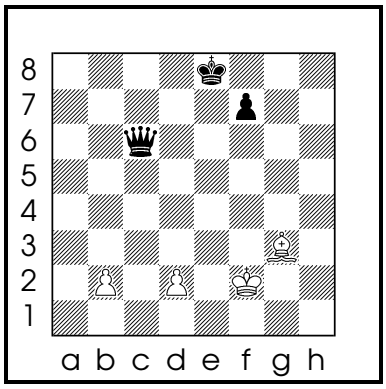
## Chess terminology

**Chess figures:** king ♔ ♚, queen ♕ ♛, rook ♖ ♜, bishop ♗ ♝, knight ♘ ♞, pawn ♙ ♟

**Special moves:** castling 0-0 or 0-0-0, en passant c5×b5

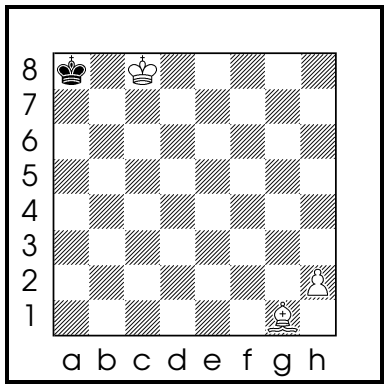
## Retrograde analysis

**1**



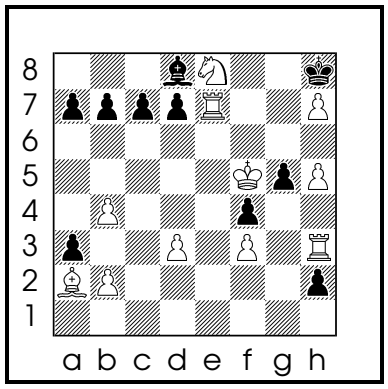
(*) Is it possible that no pawn changed?

**2**



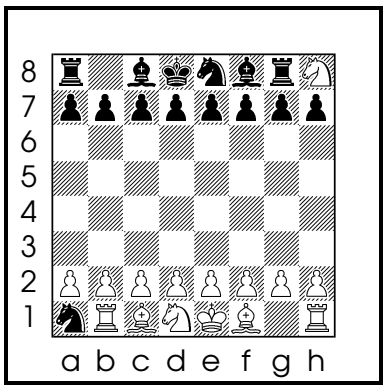
(*) White on turn. What was the last black turn?

**3**



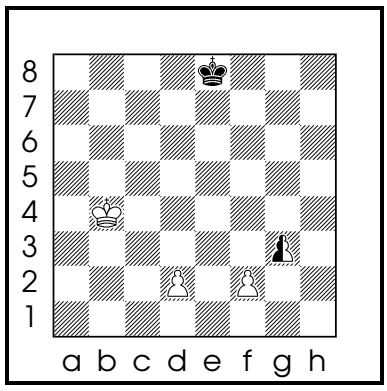
(**) White is on turn and mate the black in 2 moves.

**4**



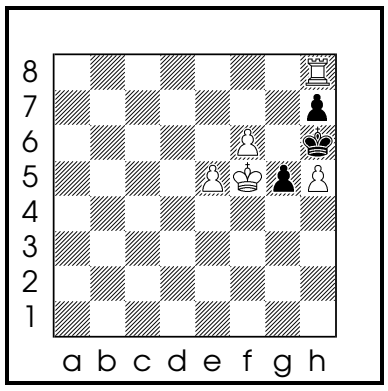
(**) Mate in one.

**5**



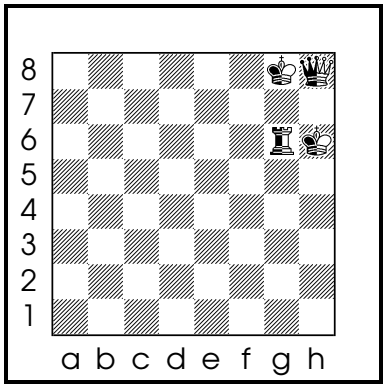
(**) No piece has moved from white to black square and vice versa. What color is the pawn on g3?

**6**



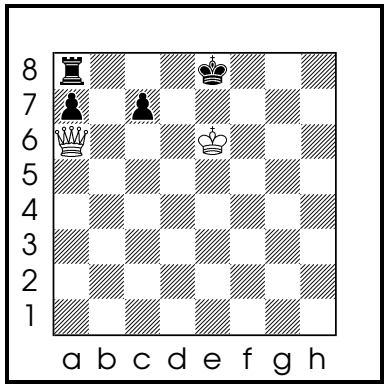
(**) White is on turn and mate in two.

**7**



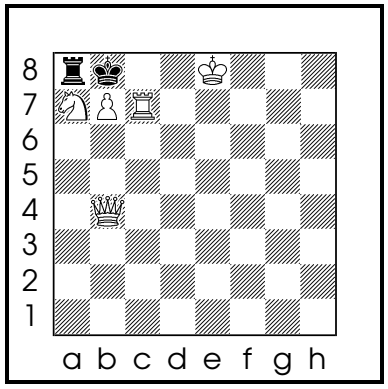
(**) Colour the pieces

**8**



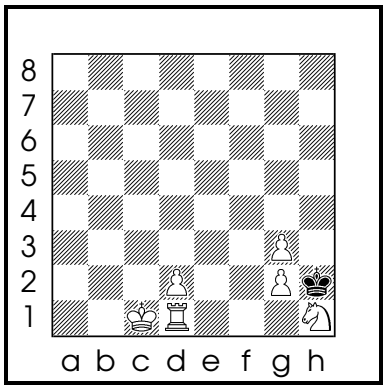
(***) White is on turn and mate the black in 2 moves.

**9**



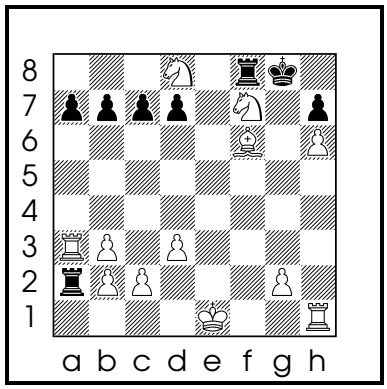
(***)Mate in one.

**10**



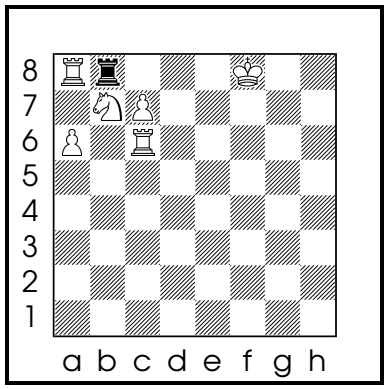
(***) What was the last black turn?

**11**



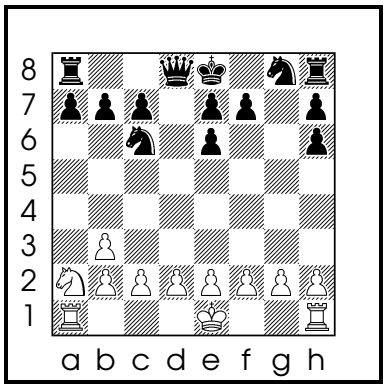
(***) White on turn. Can he castle?

**12**



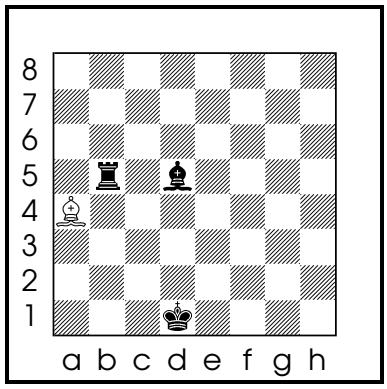
(****) The black king is invisible. Who wins?

**13**



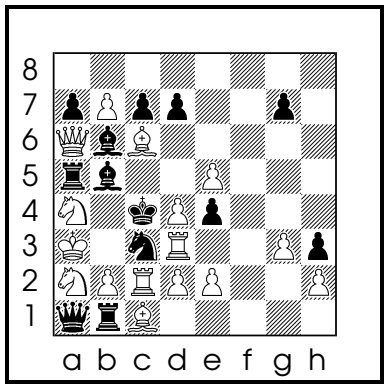
(*****) Where was the white queen captured?

**14**



(*****) Where is the white king?

**15**



(******) Who win the game?

## References


[1] The chess mysteries of Sherlock Holmes written by Raymond Smullyan (ISBN: 0394737571).

[2] The chess mysteries of the arabian knights written by Raymond Smullyan.

[3] http://www.janko.at/Retros/PedagogicalCollection.htm.

[4] http://alangullette.com/lit/dunsany/chess/problems.htm.

Jakub Sosnovec

j.sosnovec@email.cz

Presented paper by Tsai-Lien Wong & Xuding Zhu

# Every graph is $(2,3)$-choosable

(link.springer.com/article/10.1007%2Fs00493-014-3057-8)

## Introduction

A *total weighting* of a graph $G$ is a mapping $\varphi$ that assigns to each vertex and edge a real number. It is *proper* if for any adjacent vertices $u$ and $v$, we have

$$\varphi(u)+\sum_{e\in E(u)}\varphi(e)\neq\varphi(v)+\sum_{e\in E(v)}\varphi(e).$$

**1-2-3 Conjecture.** [1] *Every graph with no isolated edges has a proper edge weighting (i.e., total weighting with zero weights on vertices) using colours* $\{1,2,3\}$.

**1-2 Conjecture.** [2] *Every graph has a proper total weighting using colours* $\{1,2\}$.

A graph $G$ is $(k,\ell)$*-choosable* if the following holds. If every vertex $v$ is assigned a list $L(v)$ of $k$ permissible weights and every edge $e$ is assigned a list $L(e)$ of $\ell$ permissible weights, then $G$ has a proper total weighting $\varphi$ satisfying that $\varphi(z)\in L(z)$ for every $z\in V(G)\cup E(G)$.

**Theorem 1.** *Every graph is* $(2,3)$*-choosable.*

## Sketch of proof

Let $x_z$ be a variable associated to $z\in V(G)\cup E(G)$. Fix an arbitrary orientation $D$ of $G$. We write $uv\in E(D)$ to mean the edge oriented from $u$ to $v$. Consider the polynomial

$$P_G(\{x_z : z\in V(G)\cup E(G)\})=\prod_{uv\in E(D)}\left(\left(x_v+\sum_{e\in E(v)}x_e\right)-\left(x_u+\sum_{e\in E(u)}x_e\right)\right)$$

Let $\varphi$ be a total weighting, then $P_G(\varphi)$ denotes the evaluation of $P_G$, where each variable $x_z$ is assigned the value $\varphi(z)$.

**Observation 2.** *The mapping* $\varphi$ *is a proper total weighting of* $G$ *if and only if* $P_G(\varphi)\neq 0$.

An *index function* of $G$ is a mapping $\eta : E(G)\cup V(G)\to\mathbb{N}_0$. An index function $\eta$ of $G$ is *valid* if $\sum_z \eta(z)=|E|$. For a valid index function $\eta$, let $c_\eta$ be the coefficient of the monomial $\prod_z x_z^{\eta(z)}$ in the expansion of $P_G$. An index function $\eta$ of $G$ is *non-singular* if there exists a valid index function $\eta'$ such that $\eta'\leq\eta$ and $c_{\eta'}\neq 0$.

**Main Theorem.** *Every graph* $G$ *has a non-singular index function* $\eta$ *such that* $\eta(v)\leq 1$ *for* $v\in V(G)$ *and* $\eta(e)\leq 2$ *for* $e\in E(G)$.

**Combinatorial Nullstellensatz.** [3] *Let* $F$ *be an arbitrary field and let* $f=f(x_1,\ldots,x_n)$ *be a polynomial from* $F[x_1,\ldots,x_n]$. *Suppose the degree of* $f$ *is* $\sum_{i=1}^n t_i$, *where* $t_1,\ldots,t_n\in\mathbb{N}_0$, *and suppose that the coefficient of* $\prod_{i=1}^n x_i^{t_i}$ *in* $f$ *is nonzero. Then, if* $S_1,\ldots,S_n$ *are subsets of* $F$ *with* $|S_i|>t_i$, *then there exist* $s_1\in S_1,\ldots,s_n\in S_n$ *such that* $f(s_1,\ldots,s_n)\neq 0$.

Let $A_G$ be the matrix whose rows are indexed by edges in $D$ and columns are indexed by edges and vertices of $G$ and for $e=uv\in E(D)$ and $z\in V(G)\cup E(G)$,

$$A_G[e,z]=\begin{cases}1 & \text{if } z=v, \text{ or } z\neq e \text{ is an edge incident to } v\\ -1 & \text{if } z=u, \text{ or } z\neq e \text{ is an edge incident to } u\\ 0 & \text{otherwise}\end{cases}$$

**Observation 3.** *We can express the polynomial $P_G$ in the following way.*

$$P_G(\{x_z : z \in V(G) \cup E(G)\}) = \prod_{e \in E(D)} \sum_{z \in V(G) \cup E(G)} A_G[e,z] x_z$$

If $z$ is a vertex or edge of $G$, then $A_G(z)$ denotes the column of $A_G$ indexed by $z$. For a valid index function $\eta$, let $A_G(\eta)$ be the matrix with the property that each of its columns is a column of $A_G$ and each column $A_G(z)$ of $A_G$ occurs $\eta(z)$ times as a column of $A_G(\eta)$.

**Observation 4.** *For an edge $e = \{u,v\} \in E(G)$, we have $A_G(e) = A_G(u) + A_G(v)$.*

The *permanent* of an $m \times m$ matrix $A$ is defined as $\sum_{\sigma \in S_m} \prod_{i=1}^{m} A[i, \sigma(i)]$ and satisfies the following. If a column $C$ of $A$ is a linear combination of two column vectors $C = \alpha C' + \beta C''$ and $A'$ (respectively, $A''$) is obtained from $A$ by replacing the column $C$ with $C'$ (respectively, with $C''$), then $\operatorname{per}(A) = \alpha \operatorname{per}(A') + \beta \operatorname{per}(A'')$.

**Observation 5.** $c_\eta \neq 0$ *if and only of* $\operatorname{per}(A_G(\eta)) \neq 0$.

Assume $A$ is a square matrix whose columns are expressed as linear combinations of columns of $A_G$. The index function $\eta_A$ is defined as follows. If $z$ is a vertex or edge of $G$, then $\eta_A(z)$ is the number of columns of $A$ in which $A_G(z)$ appears with nonzero coefficient. Note that $\eta_A$ may not be uniquely defined, as the columns of $A_G$ are not linearly independent.

**Lemma 6.** *Let $\sigma$ be an index function of $G$. Then $G$ has a non-singular index function $\eta$ with $\eta \leq \sigma$ if and only if there is a square matrix $A$ whose columns are expressed as linear combinations of columns of $A_G$ such that* $\operatorname{per}(A) \neq 0$ *and $\eta_A \leq \sigma$.*

**Theorem 7.** *Assume $G$ is a connected graph and $F$ is a spanning tree of $G$. Then there is a matrix $A$ whose columns are linear combinations of columns of $A_G$ such that* $\operatorname{per}(A) \neq 0$ *and $\eta_A(v) \leq 1$ for $v \in V(F)$, $\eta_A(e) = 0$ for $e \in E(F)$ and $\eta_A(e) \leq 2$ for $e \in E(G-F)$.*

## References


[1] M. Karoński, T. Łuczak, A. Thomason: Edge weights and vertex colours, *J. Combinatorial Theory Ser. B* 9, 2004.

[2] J. Przybyło, M. Woźniak: On a 1-2 conjecture, *Discrete Mathematics and Theoretical Computer Science* 12, 2010.

[3] Noga Alon: Combinatorial Nullstellensatz, *Combinatorics, Probability and Computing* 8, 1999.

Jakub Svoboda

3.14159.jakub@gmail.com

Presented paper by Jessica Enright, Lorna Stewart

# Games on interval and permutation graph representations

## Introduction

In this paper we look at combinatorial games on graphs in which two players antagonistically build a representation of a subgraph of a given graph. We show that for a large class of these games, determining wheter a given instatnce is a winning position for the next player is PSPACE-hard. Also, for some special cases better complexity is shown.

## Definitions

**Definition 1. PSPACE** is the set of all decision problems that can be solved by a Turing machine using polynomial amount of space.

**Definition 2. Kayles game** (or only Kayles). For a given graph $G = (V,E)$, two players takes turns choosing a vertex $v \in V$ that has not been chosen before and that is not adjacent to any previously chosen vertex. The last player to choose a vertex wins.

**Definition 3. Representation games** are games played on graphs in which two players construct a representation a of a subgraph of a given graph by taking turns choosing a vertex and adding a crresponding element to the representation. At each stage, the representation that has been constructed correctly represents the subgraph induced by the vertices that have been chosen so far. The game ends when all of the vertices have been played, or when the representation cannot be extended to include any of the unchosen vertices. The last player to make a move wins.

**Definition 4.** Let $S$ be a set and $\Phi$ be a symmetric binary relation on $S$. For any $Q \subseteq R \subseteq S$ and $s \in S$, $s$ is *consistent* with $(Q,R,S)$ if for all $q \in Q$, $s\Phi q$ and for all $r \in R \backslash Q, \neg(s\Phi r)$.

**Definition 5.** Let $S$ be a set, $\Phi$ be a symmetric binary relation on $S$, $n$ be a positive integer, $R_1, R_2, \ldots, R_n$ be nonempty finite subsets of $S$ and $Q_1, Q_2, \ldots, Q_n$ be subsets of $S$ such that $Q_i \subseteq R_i$ fro all $1 \leq i \leq n$. Then $(Q_1,R_1)$, $(Q_2,R_2)$,..., $(Q_n,R_n)$ are *separating set pairs for* $(S,\Phi)$ if for all $1 \leq i \leq n$, all of the following hold:

1. there exists $s_i \in S$ that is consistent with $(Q_i,R_i)$.
2. for all $s_i$, $s_i' \in S$ consistent with $(Q_i,R_i)$, for all $r \in R_1 \cup R_2 \cup \cdots \cup R_n$: $s_i \Phi r$ if and only if $s_i' \Phi r$.
3. for all $1 \leq j \leq n$, $j \neq i$, for all $s_i$ consistent with $(Q_i,R_i)$ and all $s_j$ consistent with $(Q_j,R_j)$: $\neg(s_i \Phi s_j)$.

**Definition 6.** Let $S$ be a set and $\Phi$ be a symmetric binary relation on $S$. Then $(S,\Phi)$ is *separable* if for every positive integer $n$, there exist $n$ separating set pairs for $(S,\Phi)$.

## Results

**Theorem 7.** *Given graph $G = (V,E)$ and $U \subseteq V$. Decision if this is winning position in Kayles is in PSPACE [1].*

**Theorem 8.** *Let $S$ be a set and let $\Phi$ be a symmetric binary relation on $S$. If $(S,\Phi)$ is polynomial time separable then the $(S,\Phi)$ Game is PSPACE-hard.*

## References

[1] Thomas J. Schaefer, On the complexity of some two-person perfect-information games, J. Comput. System Sci. 16 (1978) 185–225

Jana Syrovátková

syrovatkova@kam.mff.cuni.cz

Presented paper by Charles H. Pence, Lara Buchak

# Oyun: A New, Free Program for Iterated Prisoner‘s Dilemma Tournaments in the Classroom

**Prisoner‘s dilemma**

| | B: Cooperate | B: Defect |
|---|---|---|
| A: Cooperate | 3,3 (mutual cooperation) | 0, 5 („sucker‘s payoff“) |
| A: Defect | 5, 0 (defector‘s payoff) | 1, 1 (mutual defection) |

Defection is **strongly dominant** – regardless of what my opponent does, defecting gives me a higher payoff than cooperating, so it seems that I should choose to defect even if I don’t know what he’ll do.

For both of us is better if we both were **to cooperate** rather than defect.

**Iterated prisoner’s dilemma**

One game – defection will be evolutionarily favored. Real situations – individuals each interact repeatedly.

**First experiment (Axelrod, 1984)**

14 different strategies, head-to-head against, 5 games of 200 moves each. Score = sum of its scores number of rounds = probability that 2 strategies would meet again = 0,99654

Both won **Tit-for-Tat** – cooperates in first round, on each subsequent round repeats the previous action of the oponent

**Oyun**

Finite state machine, environment where iterated prisoner’s dilemma strategies can be tested, refined, and explored

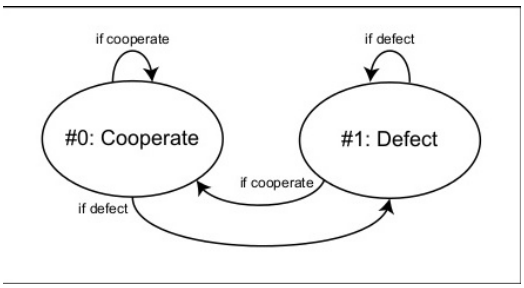
**Numbered set of states and instructions:**

what act to perform in that state (C or D)

how to respond (which state to move into if the other player C or D)

**Syntax:**

| | |
|---|---|
| John Doe | Author |
| Tit-For-Tat | Name of the strategy |
| 2 | Number of states |
| C, 0, 1 | State #0, if C $\rightarrow$ leave, D $\rightarrow$ 1 |
| D, 0, 1 | State #1, C $\rightarrow$ next round be in 0 |




**Tournaments**

Round-robin tournament – 5 games with 168, 359, 306, 622 and 319 rounds.

Evolutionary – uniformly distributed population across all. Next generation – relative fitness.

**Other strategies**

groups, define one as host, rest as parasite (how to set host to stay in game?)

Tit-for-Two-Tats, AW, Consolation Prizefighter

## Appendix: Selected Student Entries

**Jane Doe**
**Tit-for-Two-Tats**
```
3
C, 0, 1
C, 1, 2
D, 0, 1
```

**Urocerus gigas**
**Host**
```
12
C, 1, 11
C, 2, 11
C, 3, 11
D, 7, 4
C, 5, 6
C, 5, 5
C, 6, 6
C, 8, 8
C, 9, 9
C, 10, 10
C, 10, 11
D, 10, 11
```

**Rhyssa persuasoria**
**Parasite**
```
12
C, 1, 11
C, 2, 11
C, 3, 11
D, 7, 4
D, 5, 6
D, 5, 5
C, 6, 6
C, 8, 8
C, 9, 9
C, 10, 10
C, 10, 11
D, 10, 11
```

**Angelo Wong**
**AW**
```
19
C, 0, 1
D, 1, 2
C, 2, 3
C, 3, 4
D, 4, 5
C, 5, 6
C, 6, 7
C, 7, 8
C, 8, 9
D, 9, 10
C, 10, 11
C, 11, 12
C, 12, 13
C, 13, 14
C, 14, 15
C, 15, 16
C, 16, 17
C, 17, 18
D, 18, 18
```

**Robert Justin Sutton**
**Consolation Prizefighter**
```
20
C, 0, 1
D, 2, 2
C, 3, 3
C, 3, 4
D, 5, 5
D, 6, 6
C, 7, 7
C, 7, 8
D, 9, 9
D, 10, 10
D, 11, 11
C, 12, 12
C, 12, 13
D, 14, 14
D, 15, 15
D, 16, 16
D, 17, 16
D, 18, 19
C, 18, 19
D, 19, 19
```

Matas Šileikis

matas.sileikis@gmail.com

# Half-sandwich of Two Random Graphs

**Summary**

We consider relations between two models of random graphs on vertex set $[n] = \{1, \dots, n\}$, as $n$ grows. We show that the uniform Erdős-Rényi random graph $\mathbb{G}(n,m)$ can be treated (with high probability) as a subgraph of the random $d$-regular graph $\mathbb{R}(n,d)$ in such a way that the former graph consists of "almost all" edges of the latter. The assumption that we make is that $d = d(n)$ grows faster than $\log n$ but slower than $n$.

**Definitions and Results**

**Definition 1.** A sequence of probability events $A_n$, $n = 1, 2, \dots$ holds *with high probability (w.h.p.)* if $\mathbb{P}(A_n) \to 1$ as $n \to \infty$. For two sequences $a_n$ and $b_n$ we write $a_n \ll b_n$ and $b_n \gg a_n$ if $a_n/b_n \to 0$, as $n \to \infty$.

**Definition 2.** The random graph $\mathbb{G}(n,m)$ is picked uniformly among graphs on $n$ vertices with $m$ edges and the random graph $\mathbb{R}(n,d)$ is picked uniformly among graphs with $n$ vertices that are $d$-regular (that is, with degree of every vertex equal to $d$).

It is an easy fact that whenever average degree $2m/n$ of $\mathbb{G}(n,m)$ grows faster than $\log n$, then with high probability all vertex degrees are $2m(1 \pm \varepsilon_n)/n$, where $\varepsilon_n = o(1)$ are some non-random numbers.

Kim and Vu [1] stated the following.

**Conjecture 3.** *If $d = d(n) \gg \log n$, then one can pick parameters $m_- = m_-(n), m_+ = m_+(n) \sim nd/2$ and define three random graphs $\mathbb{G}(n,m_-), \mathbb{R}(n,d)$, and $\mathbb{G}(n,m_+)$ on the same vertex set in such a way that*

$$\mathbb{G}(n,m_-) \subseteq \mathbb{R}(n,d) \subseteq \mathbb{G}(n,m_+) \qquad \text{w.h.p.}$$

The purpose of such a statement is to have an easy way to prove *monotone properties*.

**Definition 4.** A *monotone increasing (decreasing) property* $\mathscr{P}$ is a family of graphs closed under addition (removal) of edges.

An example of an increasing property is connectivity: if you add an edge to a connected graph, it stays connected. Note that if you add a vertex, but no edges connecting it to old vertices, it might lose the property. So essentially we are considering a sequence of monotone properties $\mathscr{P}_n, n = 1, 2, \dots$, where $\mathscr{P}_n$ are the graphs on $n$ vertices with property $\mathscr{P}$.

In [2] the following theorem was proved (in [1] Theorem 5 was proved with a stronger upper bound on $d$).

**Theorem 5.** *If $\log n \ll d \ll n$, then one can pick a parameter $m = m(n) \sim nd/2$ and define random graphs $\mathbb{G}(n,m)$ and $\mathbb{R}(n,d)$ on the same vertex set in such a way that*

$$\mathbb{G}(n,m) \subseteq \mathbb{R}(n,d) \qquad \text{w.h.p.}$$

**Corollary 6.** *Suppose that $\mathscr{P}$ is an increasing property and $\mathbb{G}(n,m) \in \mathscr{P}$ for some $m \gg n \log n$. Then there exists $d = d(n) \sim 2m/n$ such that $\mathbb{R}(n,d) \in \mathscr{P}$ w.h.p.*

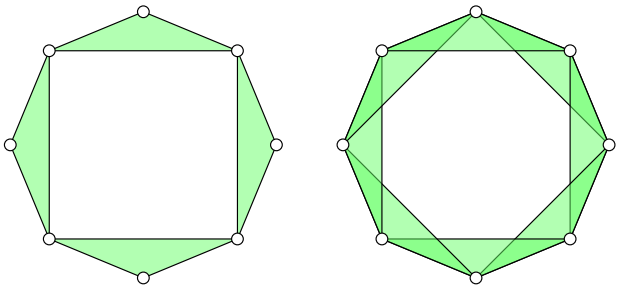



Figure 7: Hamilton cycles for $n = 8, k = 3$ and $\ell = 1, 2$.

## Extension to Hypergraphs

In [2] the results of the previous section were also extended to $k$-uniform hypergraphs (also known as $k$-graphs).

**Definition 7.** A *$k$-graph* on *vertex set* $[n]$ is a family of $k$-element subsets (*edges*) of $[n]$.

**Definition 8.** A $k$-graph is *$d$-regular* if each vertex of it belongs to exactly $d$ edges.

Random $k$-graphs $\mathbb{G}^{(k)}(n,m)$ and $\mathbb{R}^{(k)}(n,d)$ are defined similarly as in the graph case. Note that a $d$-regular $k$-graph has $nd/k$ edges, and the largest possible value of parameter $d$ is $\binom{n-1}{k-1}$, which is of order $n^{k-1}$.

**Theorem 9.** *If* $\log \ll d \ll n^{k-1}$*, then one can pick a parameter* $m = m(n) \sim nd/k$ *and define random $k$-graphs* $\mathbb{G}^{(k)}(n,m)$ *and* $\mathbb{R}^{(k)}(n,d)$ *on the same vertex set in such a way that*

$$\mathbb{G}^{(k)}(n,m) \subseteq \mathbb{R}^{(k)}(n,d) \qquad \text{w.h.p.}$$

**Definition 10.** Given $\ell = 1, \ldots, k-1$, a Hamilton $\ell$-cycle is a $k$-graph such that for some cyclic order of vertices every vertex belongs to an edge, every edge consists of consecutive vertices, and consecutive edges overlap in $\ell$ vertices (see Figure).

For hypergraphs, Corollary 6 still holds, but we instead give a particular example. In view of a results of Dudek and Frieze (see [2] for references), Theorem 9 implies the following.

**Corollary 11.** *Let* $\ell = 1, \ldots, k-1$*. If* $d = d(n)$ *grows faster than* $\log n$ *(if* $\ell = 1$*) or faster than* $n^{\ell-1}$ *(if* $\ell > 1$*), then* $\mathbb{R}^{(k)}(n,d)$ *contains a Hamilton $\ell$-cycle w.h.p.*

In [2] it is conjectured that the assumption $d$ is of optimal order for $\ell > 1$, but for $\ell = 1$ the assumption on $d$ can be relaxed.

**Conjecture 12.** *There is an integer constant $C$ such that* $\mathbb{R}^{(k)}(n,d)$ *has a Hamilton* 1*-cycle w.h.p., if* $d \geq C$*.*

## References

[1] J. H. Kim and V. H. Vu: Sandwiching random graphs: universality between random graph models. *Adv. Math.*, 188(2):444–469, 2004.

[2] A. Dudek, A. Frieze, A. Ruciński, and M. Šileikis: arxiv:1508.06677, 2015.

Martin Töpfer

mtopfer@gmail.com

Presented paper by Ararat Harutyunyana, Reza Naserasr b, Mirko Petruševski c, Riste Škrekovski d,e,f, Qiang Sunb

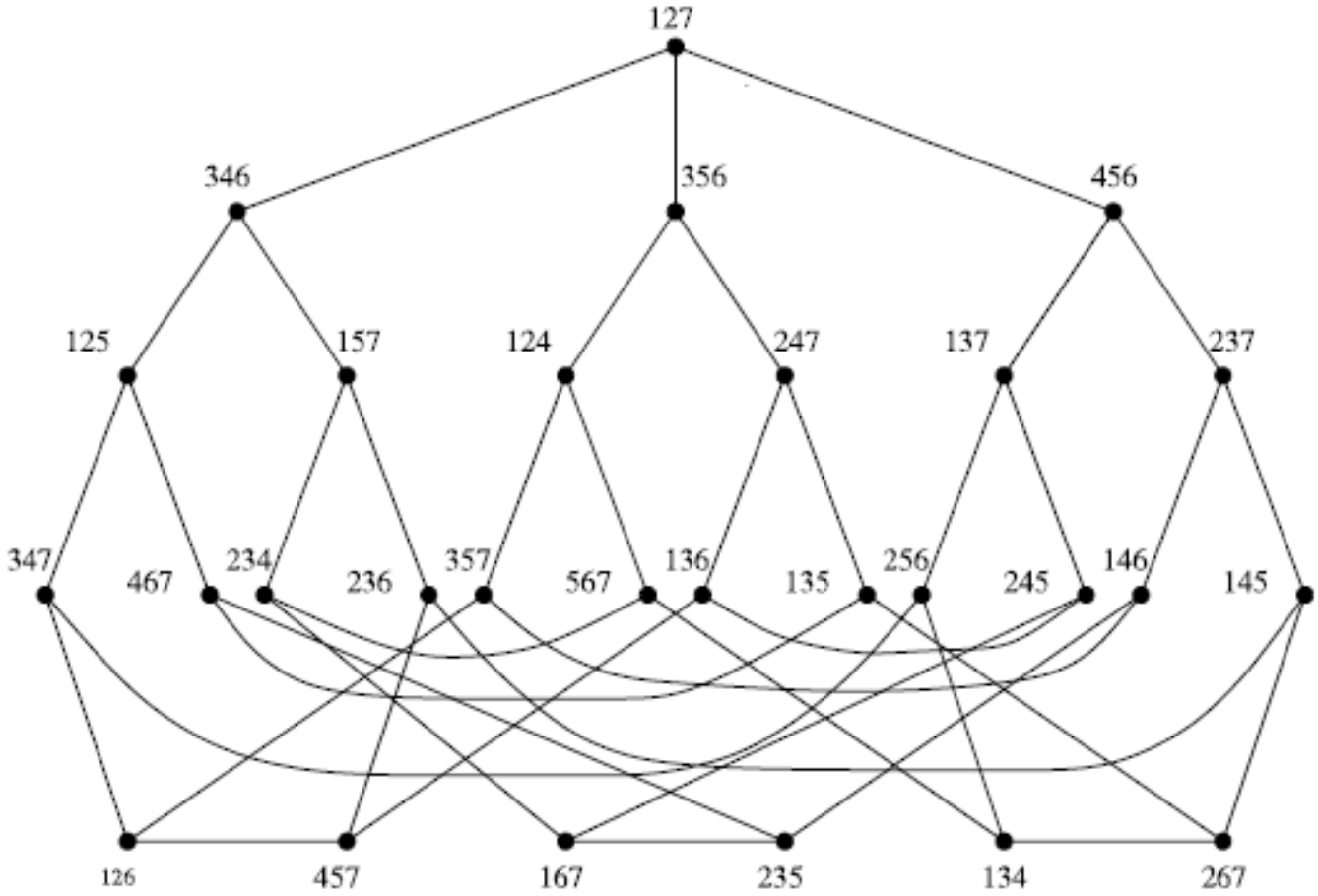
# Mapping planar graphs into the Coxeter graph

(http://www.math.univ-toulouse.fr/~aharutyu/CoxeterFinal.pdf)

## Definitions

**Definition 1.** *Odd girth* is the length of the shortest odd cycle in graph.

**Definition 2.** The *Coxeter graph* is a subgraph of Kneser graph $K(7,3)$ obtained by deleting the vertices corresponding to the lines of the Fano plane.




**Definition 3.** A *homomorphism* of a graph $G$ to another graph $H$ is a mapping $\varphi : V(G) \to V(H)$ which preserves adjacency, i.e. $uv \in E(G) \to \varphi(u)\varphi(v) \in E(H)$.

**Theorem 4.** *Every planar graph of odd girth at least 17 admits a homomorphism to the Coxeter Graph.*

**Conjecture 5.** *Every planar graph of odd girth at least 11 admits a homomorphism to the Coxeter Graph.*

**Definition 6.** Distinct vertices $x$ and $y$ are *weakly adjacent* if there exists a path in $X$ containing both of them and with all the internal vertices of degree 2. $d_{weak}(v)$ denotes the number of weakly adjacent 2-vertices of $v$.

## Usefull lemmas

**Lemma 7.** *The Coxeter graph satisfies the following:*

1. *It is distance-transitive.*
2. *It is of diameter 4.*
3. *Its girth is seven.*
4. *For a vertex A, we have* $|N(A)| = 3$, $|N_2(A)| = 6$, $|N_3(A)| = 12$, $|N_4(A)| = 6$.
5. *Let A and B be a pair of vertices in Cox. If* $d(A,B) \leq 3$, *then a 7-cycle passes through A and B. If* $d(A,B) = 4$, *then a 9-cycle passes through A and B.*

**Lemma 8.** *Let P be a* $u\breve{}v$ *path of length* $l, l \leq 5$. *Consider a partial Cox-coloring* $\varphi$ *given by* $\varphi(u) = A$ *and* $\varphi(v) = B$. *Then,* $\varphi$ *is extendable to P if and only if:*

1. $l = 2$ *and* $d(A,B) \in 0,2$, *or*
2. $l = 3$ *and* $d(A,B) \in 3,1$, *or*
3. $l = 4$ *and* $d(A,B) \neq 1$, *or*
4. $l = 5$ *and* $A \neq B$.

**Lemma 9.** $P_7, T_{123}, T_{034}$ *are reducible configurations. If v is a 3-vertex in X,* $d_{weak} \leq 6$.

**Lemma 10.** $T_{1334}, T_{2234}, T_{2333}$ *are reducible configurations. If v is a 4-vertex in X,* $d_{weak} \leq 12$ *with equality only for* $T_{0444}$, *otherwise* $d_{weak} \leq 10$.

**Lemma 11.** $T_{24}T_{24}, T_{24}T_{33}, T_{33}T_{33}, T_{14}1T_{14}$ *are reducible configurations.*

## Discharging

Let us define $w_0(v) = 45d(v) - 120$. Then from Euler formula follows that sum of all charges in vertices is -204. Rules for discharging:
(R1)For each pair $x, y$ of weakly adjacent vertices in $X$ with $d(x) = 2$ and $d(y) \geq 3$, $y$ sends charge of 6 to $x$.
Given a $3^+$-vertex $x \in V(X)$, we say $x$ supports $v$ if: (i) $w_1(x) > 0$, and (ii) $x$ is a leaf vertex of $T(v)$ on a shortest thread of $T(v)$.
(R2)Whenever $y$ supports a vertex $x$ with $w_1(x) < 0$, then $y$ gives charge of 3 to $x$ if $d(x,y) = 1$, and charge of 1.5 to $x$ if $d(x,y) \neq 1$.

Jan Voborník

vobornik.jan@gmail.com

Presented paper by Noga Alon, Michal Feldman, Omer Lev and Moshe Tennenholtz

# How Robust is the Wisdom of the Crowds?

(http://www.tau.ac.il/~nogaa/PDFS/ijcai15.pdf)

## Introduction

We introduce the study of adversarial effects on wisdom of the crowd phenomena. In particular, we examine the ability of an adversary to influence a social network so that the majority of nodes are convinced by a falsehood, using its power to influence a certain fraction, $\mu < 0.5$ of $N$ experts.

We are interested in providing an agent, who does not necessarily know the graph structure nor who the experts are, to determine the true value of a binary property using a simple majority.

We prove bounds on the social graph's *maximal degree*, which ensure that with a high probability the adversary will fail (and the majority vote will coincide with the true value) when he can choose who the experts are, while each expert communicates the true value with probability $p > 0.5$. When we examine *expander* graphs as well as random graphs we prove such bounds even for stronger adversaries, who are able to pick and choose not only who the experts are, but also which ones of them would communicate the wrong values, as long as their proportion is $1-p$. Furthermore, we study different propagation models and their effects on the feasibility of obtaining the true value for different adversary types.

## More formally

We consider a social network given by undirected graph $G = (V,E)$ with $|V| = n$ nodes, corresponding to agents. The agents are interested in a binary *ground truth* which can be either *red* $(R)$ or *blue* $(B)$. Without a loss of generality assume that the ground truth in the world the agents live in is *red*.

**Opinion**: Denote by $c(v) \in R,B$ the opinion of node $v$.

**Experts**: A set $V' \subseteq V$ of size $\mu|V|$, constitutes the *expert* set.

**Forming an Opinion**: Every agent forms his opinion based on the majority opinion of his expert neighbors. That is, if $|\{u \in N(v) \cap V' | c(u) = R\}| > |\{u \in N(v) \cap V' | c(u) = B\}|$, then $c(v) = R$. If the inequality is reversed, then $c(v) = B$ and if there is a tie, $v$ decides randomly s.t. $\Pr[c(v) = R] = \Pr[c(v) = B] = 1/2$.

We consider three models of expert formation:

- **Strong adversary**: an adversary chooses an expert set $V' \subseteq V$ (such that $|V'| = \mu|V|$), and assigns opinions to agents in $V'$ satisfying the following equations: $|\{v \in V' | c(v) = R\}| \geq (\frac{1}{2} + \delta)|V'|$ and $\{v \in V' | c(v) = B\} = V' \setminus \{v \in V' | c(v) = R\}$, for some fixed $\delta$.
- **Weak adversary**: an adversary chooses an expert set $V' \in V$ (such that $|V'| = \mu|V|$). Experts receive signals about the state of the world, and are more likely to be correct than incorrect. Specifically, for every agent $v \in V'$ independently, it holds that $c(v) = R$ with probability $1/2 + \delta$ and $c(v) = B$ with probability $1/2 - \delta$, for some fixed $\delta$.
- **Passive adversary (random process)**: a set of $\mu|V|$ nodes are chosen uniformly at random forming expert set $V'$ . Opinion formation of agents in $V'$ is as in the weak adversary model.

**Robust network**: A network is said to be *robust* against a particular adversary if, with high probability, the majority of agents hold the true opinion, despite adversary's attempt to deceive.

**Fact 1.** *[Chernoff's inequality] Let $X_i \in [0,1], i = 1,\ldots,n$ be independent random variables and $X = \sum_{i=1}^{n} X_i$, $\mu = \mathbb{E}[X]$ and $\delta \in (0,1)$. Then $Pr[X > (1+\delta)\mu] \leq e^{-\delta^2\mu/3}$.*

### Weak Adversaries

**Theorem 2.** *For $0 < \varepsilon < \mu$, $\delta < \frac{1}{2}$, if $n$ is sufficiently large, there is an absolute positive constant $c_1$ so that if the largest degree $\Delta$ satisfies*

$$\Delta \leq c_1 \frac{\varepsilon\delta^4\mu n}{\log(1/\varepsilon)},$$

*then majority over all vertices gives the truth with probability at least $1-\varepsilon$.*

**Proof** We split the vertices that are not experts to three groups.

$$V_H = \{v \in V - V' : |N_v \cap V'| \geq M\}$$
$$V_L = \{v \in V - V' : 1 \leq |N_v \cap V'| < M\}$$
$$V_N = \{v \in V - V' : N_v \cap V' = \emptyset\}$$

And split them to sets of agents with true and false opinions $V_{HT}, V_{HF}, \ldots$.

We prove four inequalities which hold for large enough $n$ and together imply the theorem. To make sure they all hold simultaneously, we use the union bound. First follows from Chernoff's inequality.

$$\Pr[|V'_T| - |V'_F| < \delta\mu n] < \frac{\varepsilon}{4} \tag{5}$$

By Chernoff again, we get

$$\Pr[|V_{NF}| - |V_{NT}| > \frac{\delta\mu n}{4}] < \frac{\varepsilon}{4}. \tag{6}$$

Using Chernoff once more and Markov's inequality,

$$\Pr\left[|V_{HF}| - |V_{HT}| > \frac{\delta\mu n}{4}\right] < \Pr\left[|V_{HF}| > \frac{\delta\mu|V_H|}{4}\right] < \frac{\varepsilon}{4}. \tag{7}$$

And finally by the second moment method.

$$\Pr[|V_{LF}| - |V_{LT}| \geq \frac{\delta\mu n}{2}] \leq \frac{\varepsilon}{4} \tag{8}$$

□

### Strong Adversaries

**Theorem 3.** *Leg $G = (V,E)$ be an $(n,d,\lambda)$-graph, let $A$ and $B$ be subsets of $V$ and assume that $|A| > |B|$. Let $X$ be the set of all vertices $v$ of $G$ satisfying $|N(v) \cap B| \geq |N(v) \cap A|$, where $N(v)$ is the set of neighbors of $v$ in $G$. Then*

$$|X| \leq \frac{2\lambda^2}{d^2} \frac{|A|(1-|A|/n) + |B|(1-|B|/n)}{(|A|-|B|)^2} n^2.$$

**Theorem 4.** *Let $G = (V,E)$ be an $(n,d,\lambda)$-graph and suppose that*

$$\frac{d^2}{\lambda^2} > \frac{1}{\delta^2\mu(1-\mu+2\delta\mu)}.$$

*Then for any strong adversary as above the majority gives the truth.*

Pavel Veselý

vesely@iuuk.mff.cuni.cz

Presented paper by Timothy M. Chan

# Improved deterministic algorithms for linear programming in low dimensions

(https://cs.uwaterloo.ca/~tmchan/detlp10_15.pdf)

Given LP $\min c^T x$ s.t. $Ax \leq b$ with $d$ variables and $n$ constrains where *$d$ is small* and *$n$ large*, how quickly can we solve it?

Constraints $Ax \leq b$ form a set $H$ of $n$ halfspaces in $\Re^d$, and we seek a point $p$ that lies in $\bigcap H$, while minimizing a linear function $c^T x$.

## Strongly polynomial-time algorithms

We are interested in running time bounded by a polynomial in the number of integers on input.

**Definition 1.** The algorithm runs in *strongly polynomial time* if:

- the number of operations in the arithmetic model of computation is bounded by a polynomial in the number of integers (or rationals) in the input instance; and
- the space used by the algorithm is bounded by a polynomial in the size of the input.

**Open problem:**[1] *Is there a strongly polynomial algorithm for linear programming?*

There is (at least) an algorithm with running time linear in the number of constrains:

**Theorem 2.** *There exists a deterministic algorithm for linear programming using $\mathcal{O}(d)^{d/2}(\log d)^{3d}n$ arithmetic operations.*

We show:

- A simple algorithm using $\mathcal{O}(d)^{3d}(\log d)^d n$ arithmetic operations.
- A bit better, but still simple algorithm using $\mathcal{O}(d)^{2d+\mathcal{O}(\delta)}n$ arithmetic operations for any $\delta > 0$.
- Ideas leading to $d/2$ in the exponent.
- Why this is the best possible for this approach.

## $\varepsilon$-nets

**Definition 3.** For a point $p \in \Re^d$ and a set of halfspaces $H$, we define a violation set $\text{Violate}_H(p) = \{h \in H : p \notin H\}$.

A subset $R \subseteq H$ is an $\varepsilon$-net of $H$ if for every $p \in \Re^d$,

$$\text{Violate}_H(p) > \varepsilon|H| \Rightarrow \text{Violate}_R(p) \geq 1\,.$$

**Lemma 4.** *Given a set $H$ of $m \geq d$ halfspaces in $\Re^d$, we can construct an $\varepsilon$-net of $H$ of size $\mathcal{O}(d/\varepsilon \cdot \log m)$ in $\mathcal{O}(m/d)^{d+\mathcal{O}(1)}$.*

[1] That is, something for you to solve after the lecture.

### $\varepsilon$-net construction via sensitive approximations

**Definition 5.** Let $\rho_H(p) = |\text{Violate}_H(p)|/|H|$ be the fraction of violated halfspaces.

A subset $R \subseteq H$ is a sensitive $\varepsilon$-approximation of $H$ w.r.t. set of points $P$ if for every $p \in P$,

$$|\rho_R(p) - \rho_H(p)| \leq (\varepsilon/2)\sqrt{\rho_H(p)} + \varepsilon^2/2\,.$$

We also use the observation that for a construction of $\varepsilon$-net we can enumerate only vertices with $|\text{Violate}_H(p)| = \lfloor \varepsilon n \rfloor + 1$. This motivates the definition a *cap*:

**Definition 6.** $\text{Cap}(H, \alpha) = \{p \in \Re^d : \rho_H(p) \leq \alpha\}$.

**Theorem 7.** *Given a set $H$ of $m$ halfspaces in $\Re^d$ with $m$ a power of 2, and parameter $\varepsilon > 0$, we can compute an $\varepsilon$-net of size $\mathcal{O}(\frac{d}{\varepsilon}\log^2 m \log\frac{d}{\varepsilon})$ in deterministic time*

$$\mathcal{O}(\frac{1}{\varepsilon})^{d/2}(\log^2 m \log\frac{d}{\varepsilon})^d m^{1.59}\,.$$

Peter Zeman

zeman@kam.mff.cunicz

Presented paper by Eugene M. Luks

# Isomorphism of Graphs of Bounded Valence Can Be Tested in Polynomial Time

(http://www.sciencedirect.com/science/article/pii/0022000082900095)

The graph isomorphism problem asks whether two input graphs $X$ and $Y$ are isomorphic (the same after relabeling), i.e., whether there exists a bijective mapping $f\colon V(X) \to V(Y)$ such that $xx' \in E(X) \iff f(x)f(x') \in E(Y)$. It is one of the most famous problems in theoretical computer since. It is clearly in NP, however, it is not known to be polynomially-solvable or NP-complete. Aside integer factorization, this is a prime candidate for an intermediate problem with complexity between P and NP-complete. It is known that the graph isomorphism problem belongs to the low hierarchy in NP [1], which means that it is unlikely NP-complete (unless the polynomial-time hierarchy collapses to some finite level).

Recently, Babai [2] showed that the graph isomorphism problem can be solved in quasi-polynomial time. The worst case running time of a quasi-polynomial time algorithm is $2^{\mathcal{O}(\log n)^c)}$, for some fixed $c$. Note that this is slower that polynomial, but faster than exponential time.

In this talk, we present the paper of Eugene Luks which a seminal paper in the field of graph isomorphism. The main result is that the graph isomorphism problem for graphs of bounded degree can be solved in polynomial time. The ideas presented in this paper are essential to understand Babai's result [2].

**References**

[1] Schöning, Uwe. "Graph isomorphism is in the low hierarchy." Journal of Computer and System Sciences 37.3 (1988): 312-323.

[2] Babai, László. "Graph Isomorphism in Quasipolynomial Time." http://arxiv.org/abs/1512.03547.

# List of participants

Adam Kabela
Andreas Emil Feldmann
Dušan Knop
Elif Garajová
Filip Mišún
Jakub Sosnovec
Jakub Svoboda
Jana Novotná
Jana Syrovátková
Jan Musílek
Jan Voborník
Jaroslav Hančl
Jaroslav Horáček
Karel Ha
Karel Král
Mark Karpilovskij
Martin Tancer
Martin Töpfer
Matas Šileikis
Michael Pokorný
Michal Opler
Michal Černý
Milan Hladík
Mirek Rada
Ondřej Pangrác
Pavel Dvořák
Pavel Klavík
Pavel Veselý
Peter Korcsok
Peter Zeman
Petra Pelikánová
Petr Glivický
Petr Hliněný
Radovan Červený
Robert Lukotka
Robert Šámal
Stanislav Kučera
Tereza Hulcová
Tomáš Masařík
Tomáš Toufar
Tomáš Valla
Václav Blažej
Veronika Slívová


JAGUARS