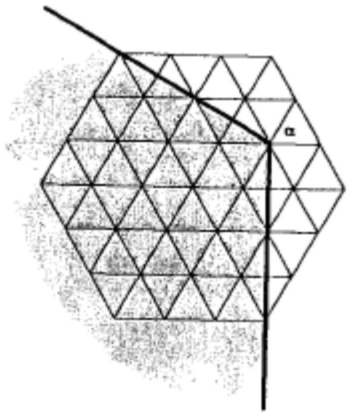
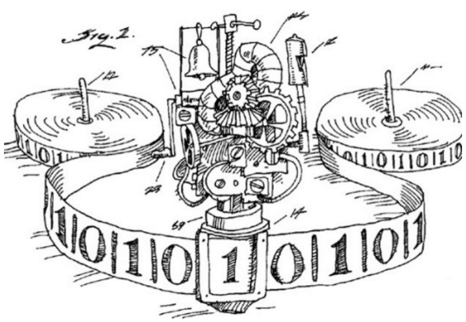
RUB

Combinatorics meets computation
(and quantum information)



Michael Walter

Ruhr University Bochum

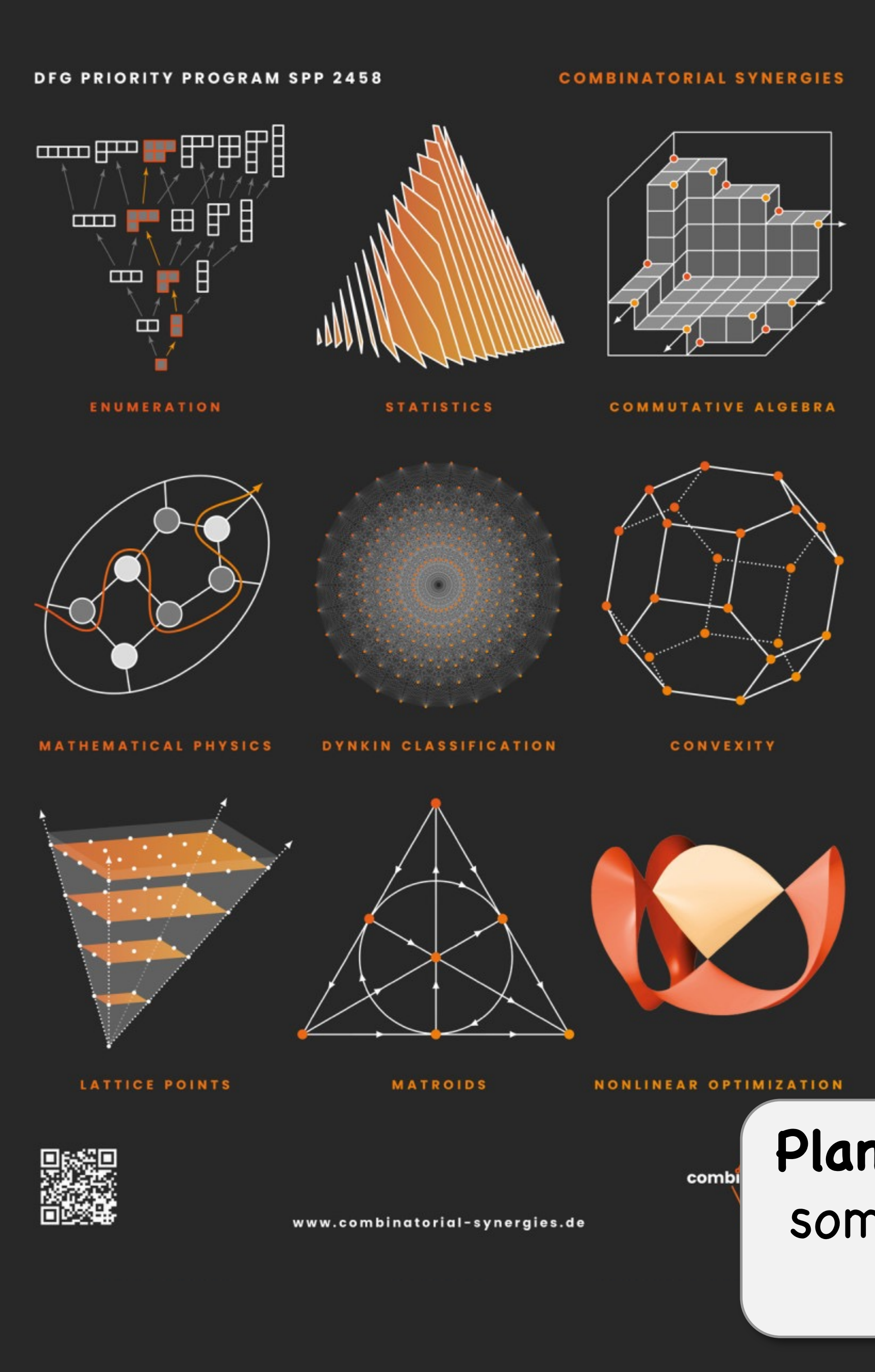


24 Hours of Combinatorial Synergies, Magdeburg, June 2023

SPONSORED BY THE
Federal Ministry of Education and Research

European Research Council
Established by the European Commission

CASA
Cyber Security in the Age of Large-Scale Adversaries



Quo vadis, combinatorics?

"first field of math where ideas and concepts of **computer science**, in particular **complexity theory**, had a profound impact."

Lovász–Shmoys–Tardos

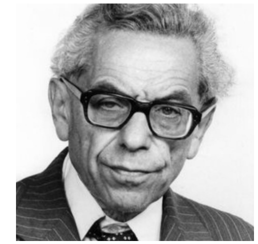
"due to the **complexity** of math observations are at beginning of a revolution [...] in the **interplay of data and structure**"

SPP 2458 homepage

Plan: Vistas of & connections between some of these themes, as offered by the lens of computation.

Is Randomness Useful?

Is Randomness Useful?



Erdös

Probabilistic method: show existence with probability > 0, instead of by *explicit* construction. Widely used.

Can randomness also help compute faster?
Or can we always "derandomize"?

Example: Given n x n matrices A, B, C, is AB = C?

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

Deterministic: Multiply matrices AB and compare with C.

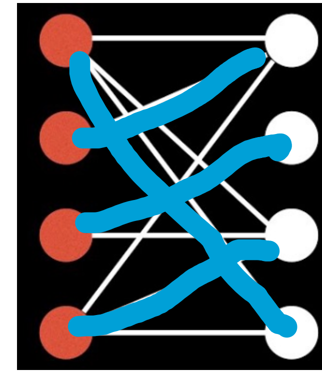
easy in time $O(n^3)$

tricky in time $O(n^{2.37\ldots})$

Randomized: Pick random vector x and compare A(Bx) with Cx.

easy in time $O(n^2)$

Bipartite Perfect Matchings



When does a bipartite graph admit perfect matching – edge set that covers all vertices once?

Permanent:

$$\mathrm{perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)}$$

A adjacency matrix, a_{ij} = 1 iff i^{th} left – j^{th} right vertex, else = 0

- ☺ counts number of perfect matchings
- ☺ nonzero iff perfect matching exists
- ☹ hard to compute Valiant

Determinants are much easier, but how about the signs…?!

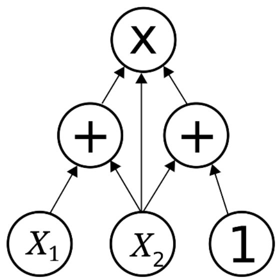
Idea: Consider **symbolic** Tutte matrix T with t_{ij} = $\mathbf{x}_{ij}$ iff edge, else = 0.

Graph admits perfect matching iff **det(T)** is **nonzero polynomial**!

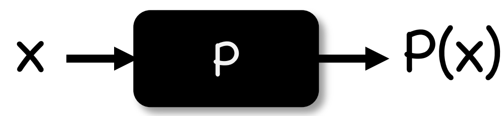
Polynomial Identity Testing (PIT)

Given a polynomial P, is it zero or not?

How is P given? For example arithmetic circuit, or even just as black box.



NB: cannot simply check all coeffs, since exp. many!



How to solve it?
Just plug in a random point ☺

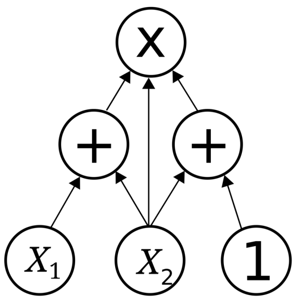
Schwartz-Zippel Lemma:

If P nonzero and pick random x_1, x_2, … in S, then $\Pr(P(x)=0) \leq \deg(P) / |S|$.

How to derandomizing? Find generic points, which as we know is tricky…

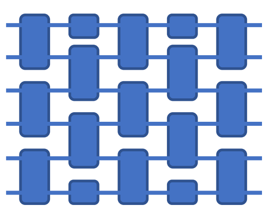
What do we know?

Is randomness is really required to compute faster? At heart of polynomial identity testing (PIT) problem. Wide open, despite intense efforts.



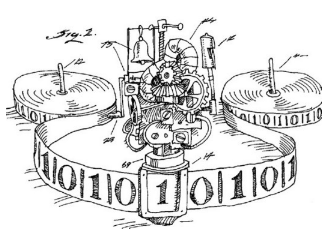
Symbolic determinants are as hard as the general problem. Only for special circuit classes, "hitting sets" have been constructed, exploiting *combinatorial* structure (e.g. sparsity).

Fundamental question with surprising connections:



Positive solution would imply major circuit lower bounds.

hardness problems <-> pseudorandomness Kabanets-Impagliazzo

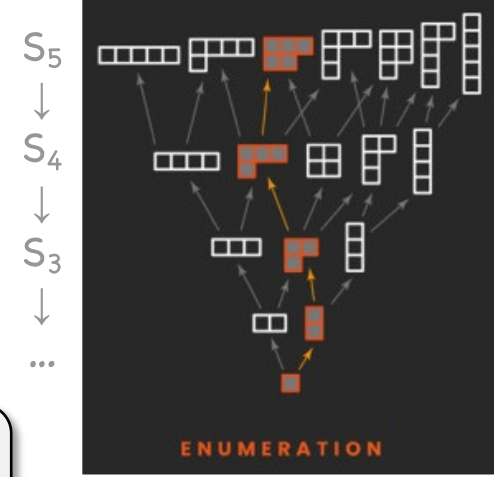


Noncommutative problem has recently been derandomized!

Garg et al, Ivanyos et al, Hirai, ...

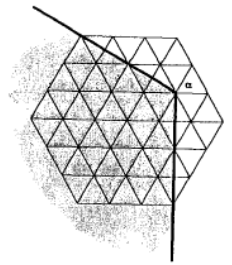
What is Counting?

Multiplicities



A natural and rich source of counting problems:

Given a group representation, how does it decompose into irreducibles ("irrep")? What are the **multiplicities**?



Example: Kostka numbers obtained by decomposing irrep of GL(n) into weight spaces.

= # of semistandard Young tableaux of given shape and content

1	2	2
2	3	

A combinatorial interpretation! Accident?

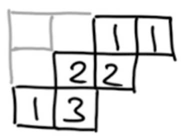
Littlewood-Richardson Coefficients

Given tensor product of GL(m) irreps, how does it decompose?

$$V_\lambda^m \otimes V_\mu^m = \bigoplus_\nu c_\nu^{\lambda\mu} V_\nu^m$$

Littlewood-Richardson coefficients

Famously, these *too* have combinatorial formulas:

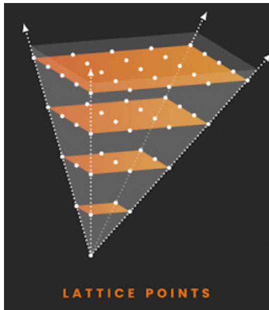


\# LR tableaux of given shape and content

\# honeycombs (or hives) with boundary conditions

Structural consequences, e.g. **saturation**: $c_{s\nu}^{s\lambda,s\mu} > 0 \Rightarrow c_\nu^{\lambda\mu} > 0$ Knutson-Tao

Does *every* multiplicity have a combinatorial formula?



The Kronecker Challenge

Let's look at tensor product multiplicities for the symmetric group S_k:

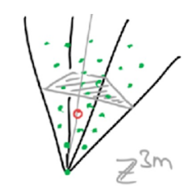
$$[\lambda] \otimes [\mu] = \bigoplus_{\nu} g_{\lambda\mu\nu} [\nu]$$

Kronecker coefficients

Many interesting connections – from combinatorics to geometry, to quantum information, and even the complexity of *matrix multiplication!*

Despite 75+ years of research, many properties remain mysterious!

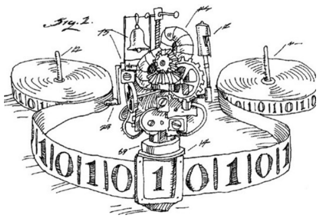
- ☹ no combinatorial interpretation
- ☹ not saturated, but we don't really understand "why"
- ☹ no effective way to decide when zero or not



What is a combinatorial formula, anyways?

"Know one if you see one"?

How Could Computer Science Possibly Help?



"Computational Problem": math problem, but answer should be given by an algorithm.

problem instance → **Algorithm** → answer

problem instance: encoded in bits

Algorithm: in some formal model, e.g. Turing machine

answer: encoded in bits

We often distinguish *decision*, *counting*, and *search* problems.

- *decision*: is there a solution?
- *counting*: how many?
- *search*: find one!

> **Complexity Theory** seeks to compare and classify computational problems according to their difficulty.

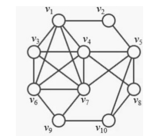
Why is det easy, but per hard?

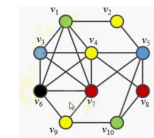
Structural and algorithmic solutions often inform each other, but not the same...

Complexity Classes

P = { problems that can be solved by efficient (poly-time) algorithm }

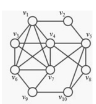
It is often easier to **verify** a proof than to **find** one...



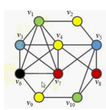


e.g. verifying a 3-coloring of a graph *vs* finding one

We can model this as follows:



problem instance →



certificate →

Efficient Algorithm → accept/reject

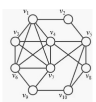
If answer YES, there should be (not too large) certificate that is accepted.
If answer NO, no certificate should be accepted.

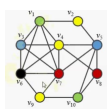
NP = { decision problems with efficiently checkable "YES certificates" }

Complexity vs Counting

P = { problems that can be solved by efficient algorithm }

NP = { decision problems with efficiently checkable "YES certificates" }


problem instance →


certificate →

Efficient Algorithm → accept/reject

#P = { problems that *count* # of "YES certificates" of such algorithm }

Proposal: How we should define "combinatorial formula"!

Mulmuley
cf. Pak–Panova

"# of not too large gadgets that satisfy easy to test criterion"

e.g. # tableaux with desired shape & weight, or # solutions to system of equations

Complexity of Multiplicities

	Kostka →	Littlewood-Richardson →	Kronecker
Positivity: > 0	P	P Saturation + LP Knutson-Tao, Mulmuley	NP-hard comb. interp. of <u>special</u> Kroneckers Mulmuley-Ikenmeyer-W
↓			
Counting: = ?	#P-complete contingency tables Narayanan	#P-complete LR rule cf. Vergne-W for quivers	#P-hard

#P-hard = any #P problem can be reduced to it

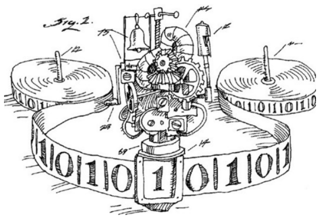
#P-complete = …and it's in #P

← *can we understand this phase transition?* →

Absurd/amusing: Kostka numbers compute <u>your</u> favorite combinatorial quantities… ☺

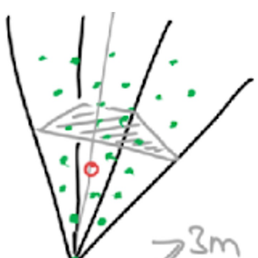
What do we know?

Computational complexity allows organizing mathematical problems by difficulty. Multiplicities give rise to most difficult counting problems.



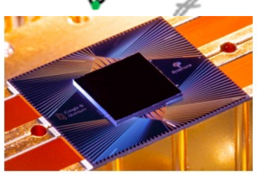
Perspective has already offered surprising new connections between combinatorics & computation, but still much to do.

Combinatorial synergies:



Explicit examples + structure from irregularity.

Mulmuley-Ikenmeyer-W



Kronecker coefficients count quantum certificates!

difficulty of finding combinatorial formula
⇔ separating classical vs quantum computing

Christandl-Harrow-W, Bravyi et al, ...

Polytopes and Complexity

Horn Problem

Given vectors α, β, $\gamma \in \mathbf{R}^n$, are there Hermitian matrices $A + B = C$ with these as eigenvalues?

This is a nonlinear and nonconvex problem… yet, magically:

Horn Cones: Possible (α,β,γ) form convex polyhedral cone Horn(n).

Mumford
Kirwan

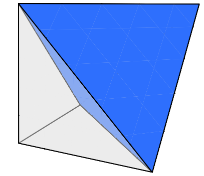
Horn conjectured recursive system of **inequalities**, established in later works. Essential ones known. Mathematically *extremely* well-understood… ☺

Klyachko
Knutson-Tao
Belkale
Ressyare
…

Yet, exponential # of facets and rays. Arguably not efficient! ☹

Can we hope to describe Horn(n) more effectively? By an algorithm…?

Computational Horn Problem



Given vectors α, β, $\gamma \in \mathbf{Q}^n$, are there Hermitian matrices A + B = C with these as eigenvalues?

Consider as **computational problem**. What is its computational complexity**?**

NP: To certify that answer YES, can simply show you A, B, C.

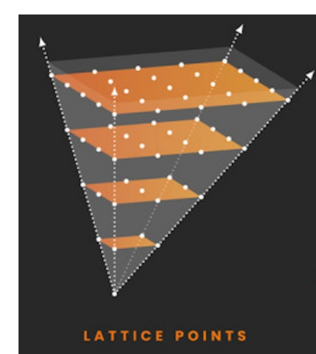
CoNP: To certify that answer NO, can hand you a separating hyperplane.

easy to verify one if you get one (not obvious)

Computer science tells us: Problems in NP ∩ CoNP unlikely to be hard!

P: In fact there is an efficient algorithm. ☺

Mulmuley, Bürgisser-Ikenm.



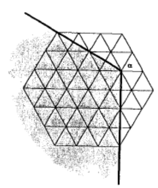
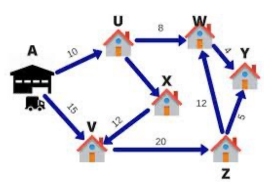
Why? Problem ⇔ stretched LR coefficient $c^{s\alpha,s\beta}_{s\gamma} > 0$.

...and by **saturation**, independent of s → use *linear* programming!

Moment Cones

Any nice group **G** and representation **V** defines convex polyhedral **moment cones** or **polytopes**. These can be described either via symplectic geometry or asymptotic invariant theory.

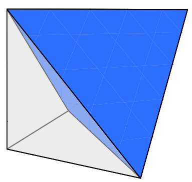
We will not define them explicitly, but mention some famous applications:

G commutative	**G noncommutative**
Schur-Horn 	 Horn & asymptotic Kronecker
matrix scaling & balancing: *statistics, numerics, ML, …*	*quantum marginals* 
linear programming: *widely used paradigm* 	Brascamp-Lieb
	noncommutative PIT *tensor ranks*
interesting and know how to solve it ☺	interesting, but no general solution (yet)

Typically exp. # facets & vertices, but succinctly "encoded" by group action!

What's the deal with Kronecker?

Given vectors, α, β, $\gamma \in \mathbf{Q}^n$, is some stretched Kronecker coefficient $g_{s\alpha,s\beta,s\gamma} > 0$?



Bürgisser-…-W, Christandl-…-W, cf. Vergne-W, Ressayre

It's a moment polytope!

☺ again NP ∩ CoNP

☺ poly time for *fixed n*

☹ no poly time algorithm known

Why no contradiction to NP-hardness of deciding $g_{\alpha,\beta,\gamma} > 0$?
Kronecker coefficients are **not** saturated! ☺

"it's a feature, not a bug"!

New perspective gives rise to the fastest practical algorithms for stretched Kronecker problem… useful for experimental mathematics?

Polynomial Identity Testing, revisited

Given matrix of linear forms: $L(x) = \sum_i x_i L_i$ Is $P(x) = \det L(x)$ nonzero?

No deterministic algorithm known, as difficult as general PIT!

Noncommutative PIT: For x_i in free skew field, is $L(x)$ invertible?

Equivalently, are there are matrices A_i s.th. $\det \sum_i A_i \otimes L_i \neq 0$?

semi-invariants of generalized Kronecker quiver

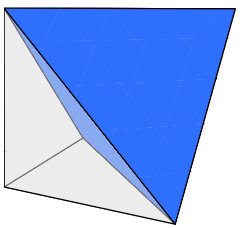
A moment polytope problem in disguise ➔ **numerical "optimization"** algo for this **algebraic** problem. Efficient because quivers are nice…!

Noncommutative PIT is in **P** ☺

Garg et al, cf.
Ivanyos et al, Hirai

What do we know?

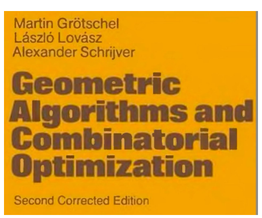
Asymptotic "combinatorial" problems give rise to interesting polyhedral cones or polytopes. Structural & algorithmic insights go hand in hand.



Moment polytopes capture some answers, and connect combinatorics with many other areas - from invariants and analysis to computer science, quantum info, and statistics…

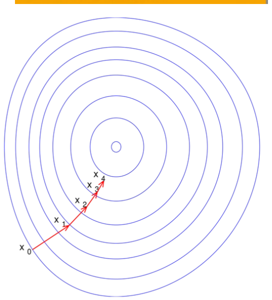
…, Bürgisser-…-W-Wigderson

Intriguing synergies:



Can we turn this around and design group actions to capture known (interesting but difficult) combinatorial polytopes?

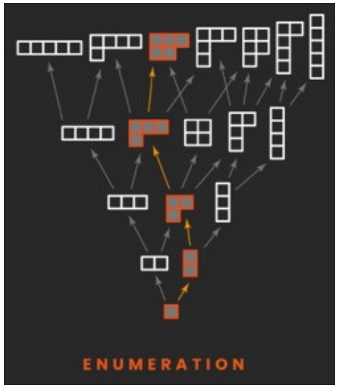
"non-commutative combinatorial polytopes"?



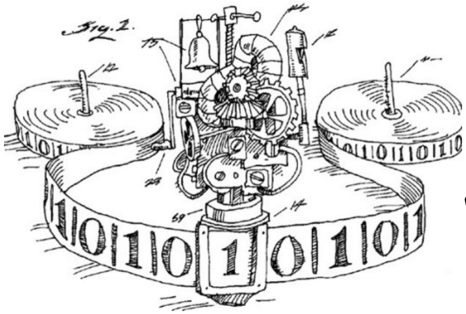
Fastest known algorithms rely on optimization.
Is there a theory of *nonlinear* linear programming…?

algos not just return YES/NO, but also find solution!

Summary



Combinatorics offers challenges, puzzles, and surprising connections...



...pushing the boundary of computer science, which in turn offers new tools and perspectives.

This SPP will offer fantastic opportunities to exploit synergies both ways, and to many other areas!

Motivation ranges from the desire to get new insights into complex combinatorial structures, to the development of faster algorithms, to the very foundations of the theory of computation.

Thank you for your attention!