Computational Complexity Theory

—

The World of P and NP

Jin-Yi Cai
Computer Sciences Dept
University of Wisconsin, Madison

Sept 11, 2012

Supported by NSF CCF-0914969.


Hilbert

Entscheidungsproblem

The rigorous foundation of Computability Theory was established in the 1930s, . . .

Answering a question of Hilbert

Computable yet Not Efficiently Computable

Given N, how fast can one factor it?

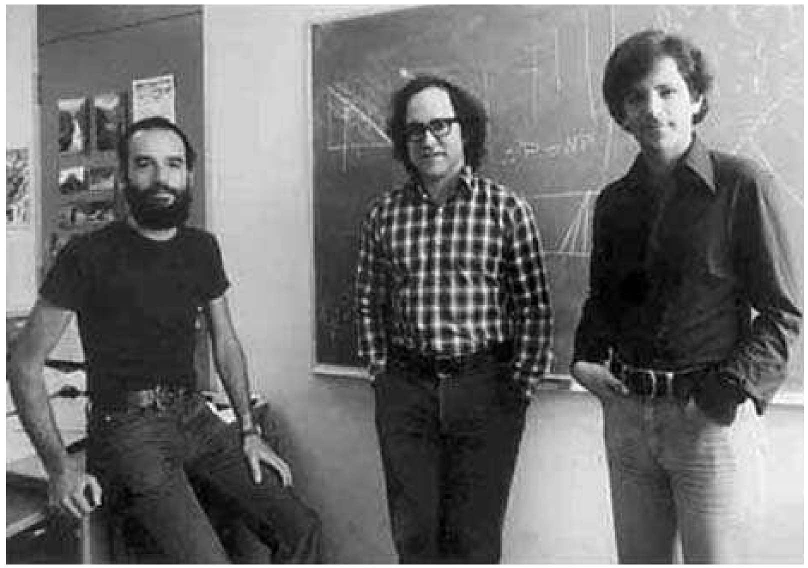
$N = 57720721296971833203785791172827243 1$**?**

$$
\begin{aligned}
N' \quad = \quad & 1375629587706555072328637871393012064224421883558 0062 \\
& 518690227129476541679834062939237944411867525 9?
\end{aligned}
$$

$$N = 9361973132609 \times 616544402332483406165590$$

$$
\begin{aligned}
N' \quad = \quad & 1471865453993855302660887614137521979 \times \\
& 934616397153579777691635581996068965840512375416381885802803 21
\end{aligned}
$$

RSA Crypto System



Based on the presumed computational complexity of factoring, Rivest, Shamir and Adleman proposed a public-key crypto system.

Is factoring intrinsically hard?

The best factoring algorithm runs in time $e^{cn^{1/3}(\log n)^{2/3}}$ (Number Field Sieve).

Shor's factoring algorithm

But by using "quantum" superposition, Shor has found a factoring algorithm which runs in polynomial time.

P and NP

P is deterministic polynomial time.

e.g. Determinant, Graph Matching (monomer-dimer problem), Max-Flow Min-Cut.

NP is non-deterministic polynomial time.

For any given instance x, it is a Yes instance iff there is a short proof which can be checked in P.

e.g. SATisfiability, Graph 3-Coloring, Hamiltonian Circuit, Clique, Vertex Cover, Traveling Salesman, etc.

Also, Factoring, Graph Isomorphism, etc.

The P vs. NP Question

It is generally conjectured that many combinatorial problems in the class NP are not computable in polynomial time.

Conjecture: P $\neq$ NP.

P $=^?$ NP is: Is there a universal and efficient method to discover a mathematical proof when one exists?

Can "clever guesses" be systematically eliminated?

What a topologist has to say

For the pure mathematician the boundary that Gödel delineated between decidable and undecidable, recursive and nonrecursive, has an attractive sharpness that declares itself as a phenomenon of absolutes. In contrast, the complexity classes of computer science for example P and NP require an asymptotic formulation and . . . demand a bit of patience before their fundamental character is appreciated.

What a topologist has to say

Setting aside the constraints of any particular computational model, the creation of a physical device capable of brutally solving NP problems would have the broadest consequences. Among its minor applications it would supersede intelligent, even artificially intelligent, proof finding with an omniscience not possessing or needing understanding. Whether such a device is possible or even in principle consistent with physical law, is a great problem for the next century.

— Michael Freedman

#P

Counting problems:

#SAT: How many satisfying assignments are there in a Boolean formula?

#PerfMatch: How many perfect matchings (Dimer Problem) are there in a graph?

#P is at least as powerful as NP, and in fact subsumes the entire polynomial time hierarchy $\cup_i \Sigma_i^p$ [Toda].

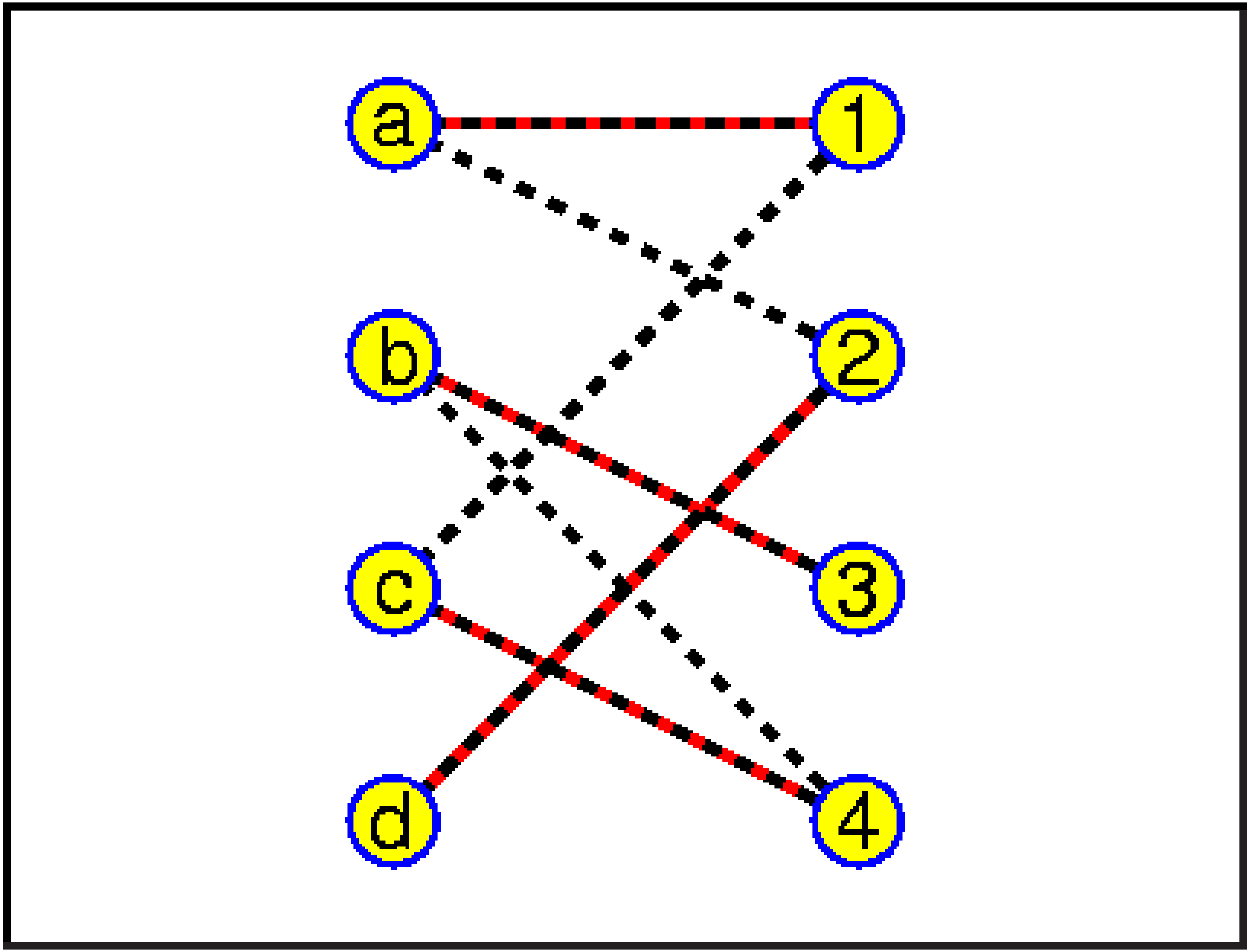
#P-completeness: #SAT, #PerfMatch, Permanent, etc.

Valiant's Holographic Algorithms

Similar to "quantum" superposition, but without using "quantum computers" , Valiant introduced holographic algorithms.

These holographic algorithms also seem to achieve exponential speed-ups for some problems.

Perfect Matchings


a
1
b
2
c
3
d
4

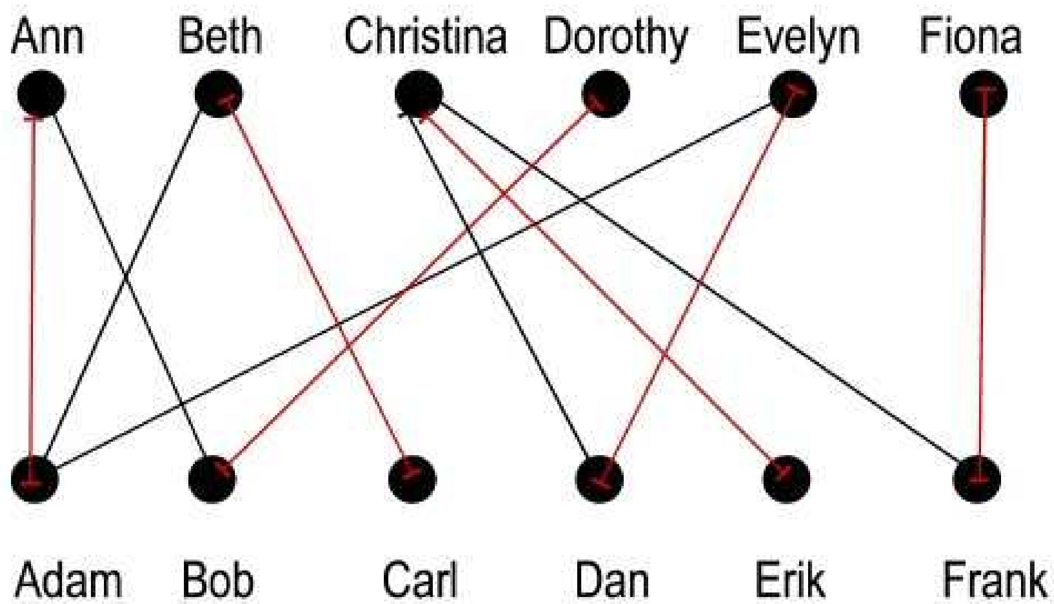
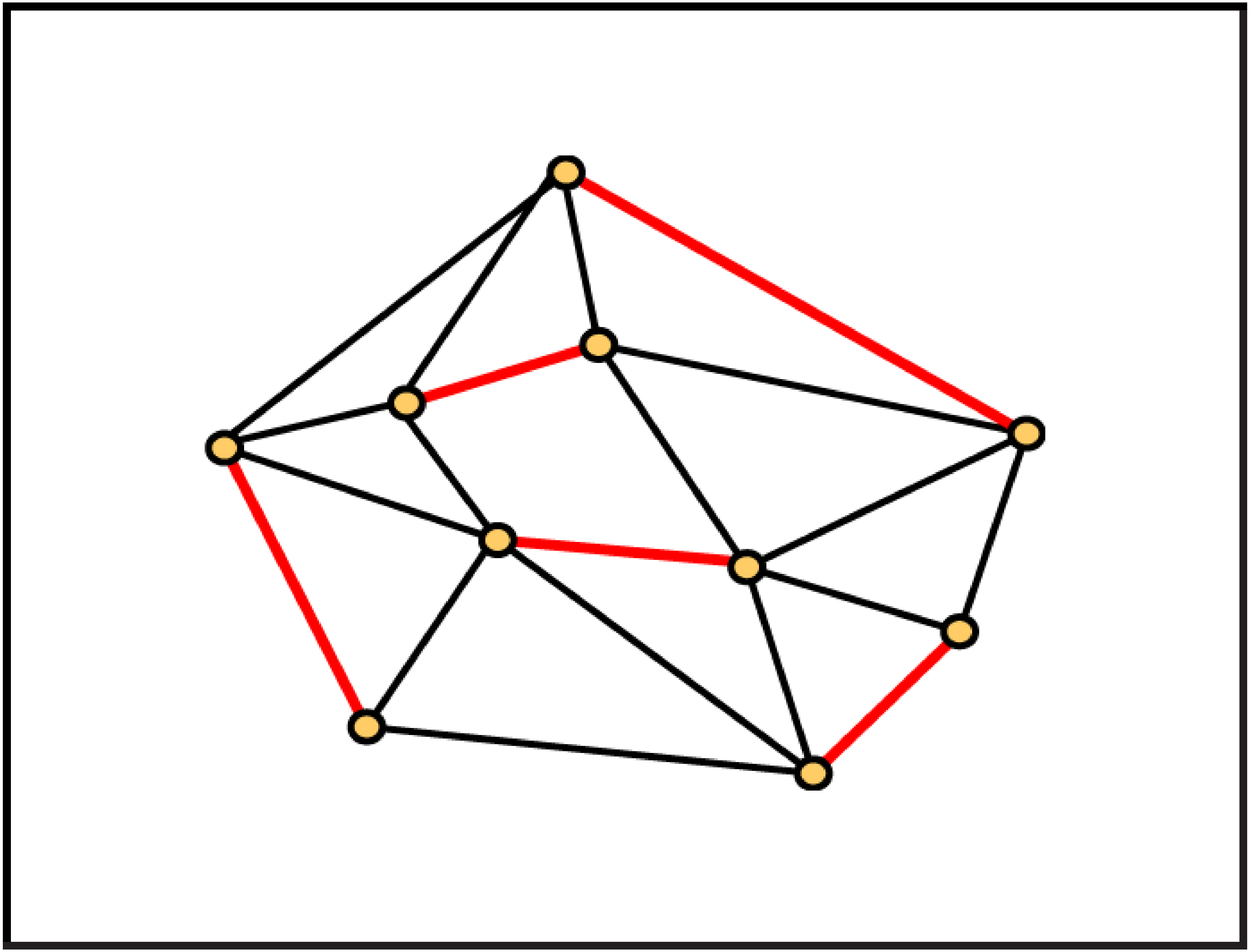
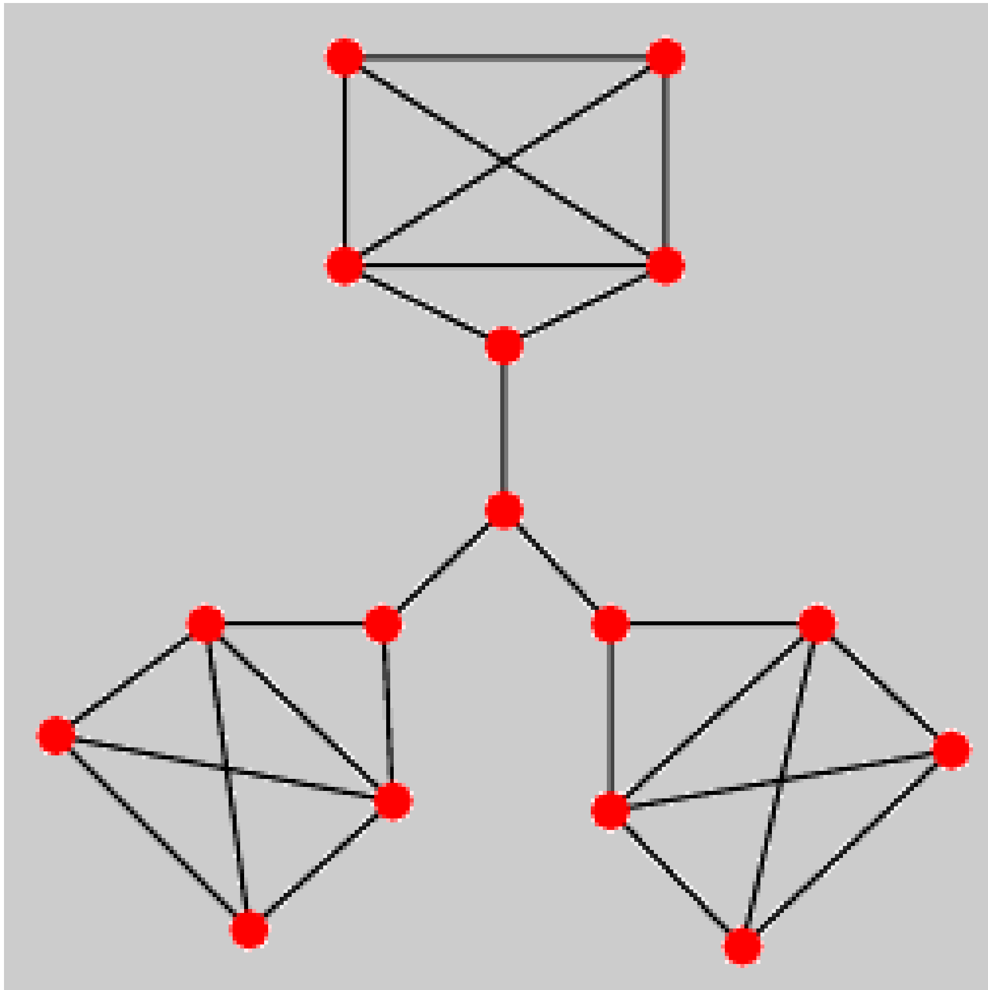


Figure 2 A perfect matching





Some Surprises

Most #P-complete problems are counting versions of NP-complete decision problems.

But the following problems are solvable in P:

- **Whether there exists a Perfect Matching in a general graph. [Edmonds]**
- **Count the number of Perfect Matchings in a planar graph. [Kasteleyn]**

Note that the problem of counting the number of (not necessarily perfect) matchings in a planar graph is still #P-complete [Jerrum].

Holographic Algorithms

Holographic algorithms have two main ingredients:

(1) Use perfect matchings to encode fragments of computations.

(2) Use linear algebra to achieve exponential cancellations.

Some seemingly exponential time computations can be done in polynomial time.

Sample Problems Solved by Holographic Algorithms

#PL-3-NAE-ICE

Input: A planar graph $G = (V, E)$ of maximum degree 3.

Output: The number of orientations such that no node has all edges directed towards it or all edges directed away from it.

Ising problems are motivated by statistical physics.

Remarkable contributions by Ising, Onsager, Fisher, Temperley, Kasteleyn, C.N.Yang, T.D.Lee, Baxter, Lieb, Wilson etc.

A Satisfiability Problem

#PL-3-NAE-SAT

Input: A planar formula Φ consisting of a conjunction of NOT-ALL-EQUAL clauses each of size 3.

Output: The number of satisfying assignments of Φ.

Constraint satisfiability problem.

e.g. PL-3-EXACTLY-ONE-SAT is NP-complete.

and

#PL-3-EXACTLY-ONE-SAT is #P-complete.

Pl-Node -Bipartition

PL-NODE-BIPARTITION
Input: A planar graph $G = (V, E)$ of maximum degree 3.
Output: The cardinality of a smallest subset $V' \subset V$ such that the deletion of V' and its incident edges results in a bipartite graph.

NP-complete for maximum degree 6.

If instead of **NODE** deletion we consider **EDGE** deletion, this is the well known **MAX-CUT** problem.

MAX-CUT is NP-hard (even NP-hard to approximate by the **PCP** Theory.)

A Particular Counting Problem

$\#_7$Pl-Rtw-Mon-3CNF

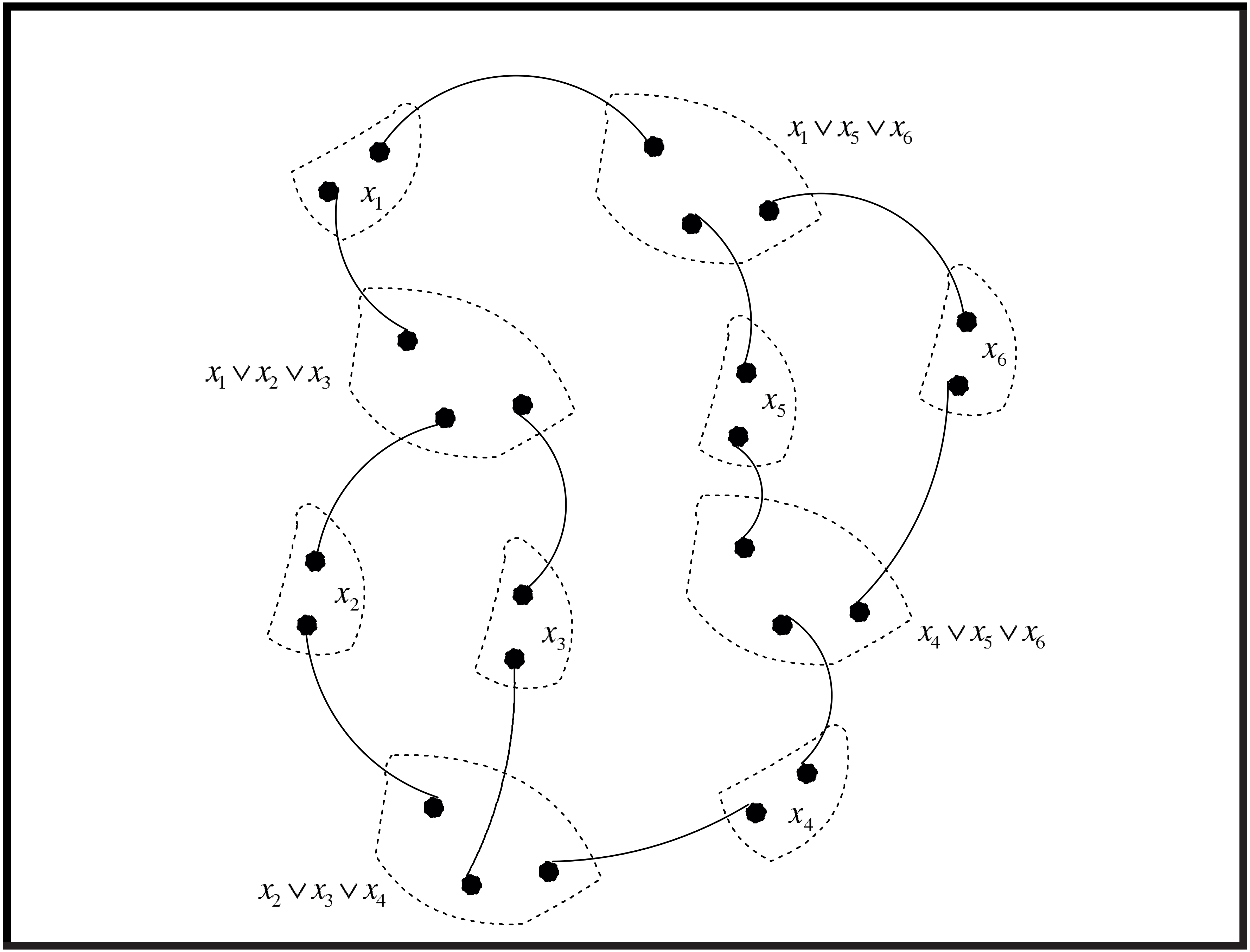
Input: **A planar graph G_Φ representing a Read-twice Monotone 3CNF Boolean formula Φ.**

Output: **The number of satisfying assignments of Φ, modulo 7.**

Here the vertices of G_Φ represent variables x_i and clauses c_j. An edge exists between x_i and c_j iff x_i appears in c_j.

Nodes x_i have degree 2 and nodes c_j have degree 3.

An Instance For Pl-Rtw-Mon-3CNF


$x_1 \vee x_5 \vee x_6$
x_1
$x_1 \vee x_2 \vee x_3$
x_6
x_5
x_2
x_3
$x_4 \vee x_5 \vee x_6$
x_4
$x_2 \vee x_3 \vee x_4$

#P-Hardness

Fact: **#Pl-Rtw-Mon-3CNF is #P-Complete.**

Fact: **$\#_2$Pl-Rtw-Mon-3CNF is NP-hard.**

An Unexpected Algorithm

There is a polynomial time holographic algorithm for $\#_7$Pl-Rtw-Mon-3CNF (Valiant).

Using Matchgate Computations . . . and Holographic Algorithms.

A Matchgate Γ

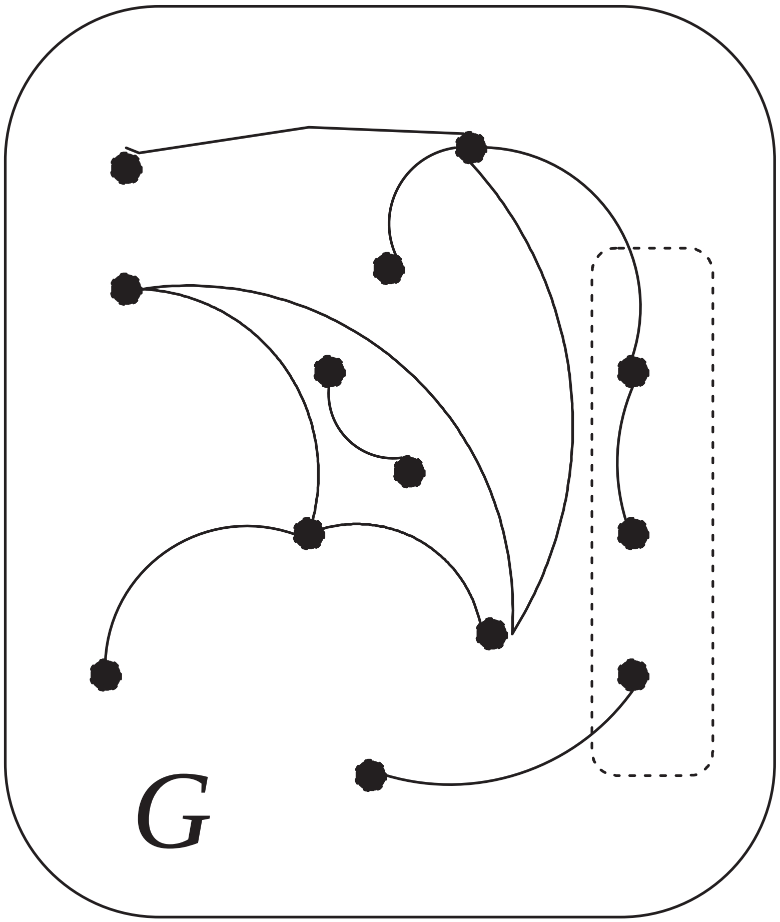


Figure 1: A matchgate Γ

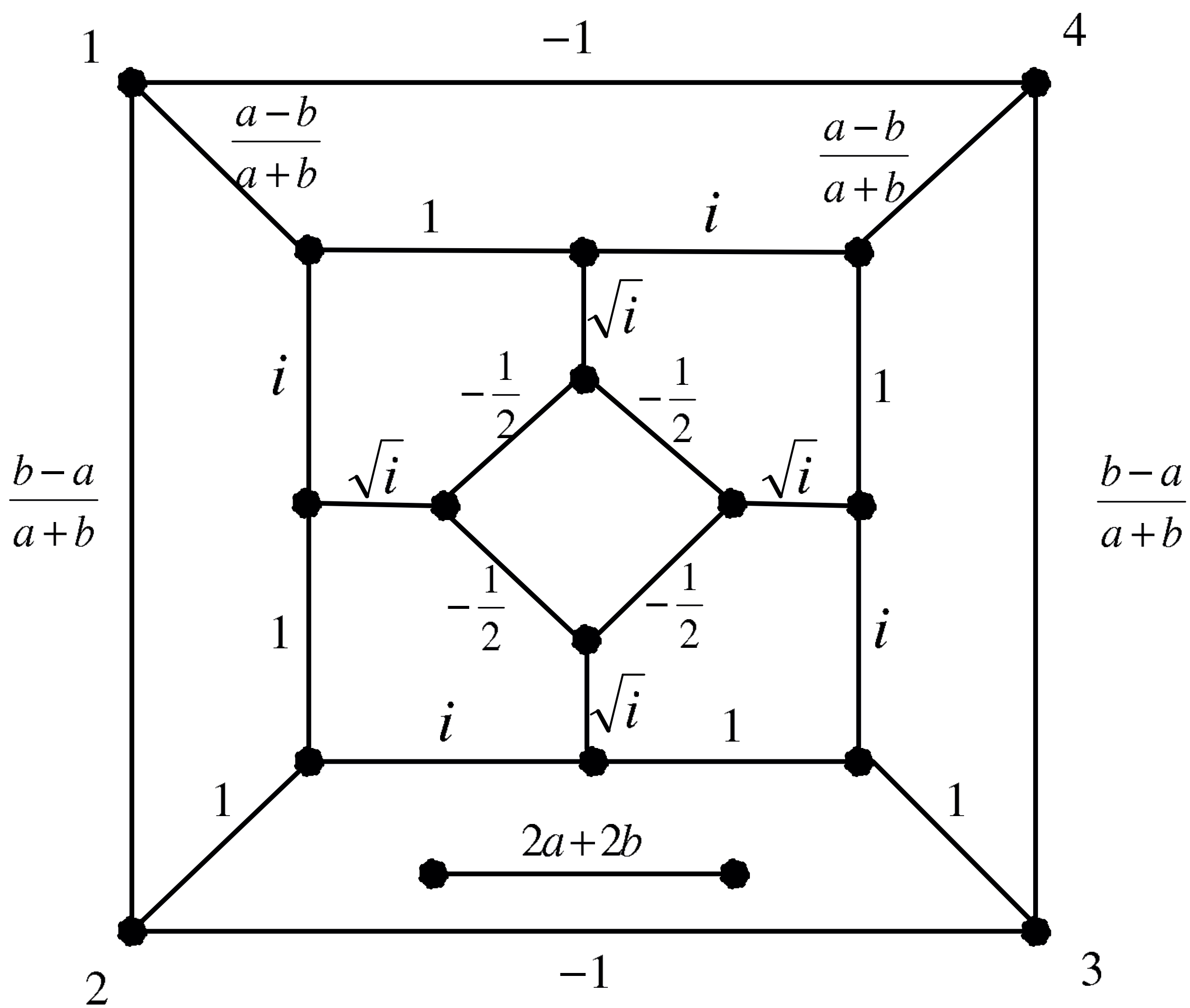
Matchgate

A planar matchgate $\Gamma = (G, X)$ is a weighted graph $G = (V, E, W)$ with a planar embedding, having external nodes, placed on the outer face.

Matchgates with only output nodes are called generators.

Matchgates with only input nodes are called recognizers.

A Matchgate


1
−1
4
$\frac{a-b}{a+b}$
$\frac{a-b}{a+b}$
1
i
$\sqrt{i}$
i
$-\frac{1}{2}$
$-\frac{1}{2}$
1
$\frac{b-a}{a+b}$
$\sqrt{i}$
$\sqrt{i}$
$\frac{b-a}{a+b}$
$-\frac{1}{2}$
$-\frac{1}{2}$
1
i
$\sqrt{i}$
i
1
1
1
$2a+2b$
2
−1
3

Standard Signatures

Define $\mathrm{PerfMatch}(G) = \sum_M \prod_{(i,j)\in M} w_{ij}$**, where the sum is over all perfect matchings** M**.**

A matchgate Γ **is assigned a Standard Signature**

$$G = (G^S) \text{ and } R = (R_S),$$

for generators and recognizers respectively.

$$G^S = \mathrm{PerfMatch}(G - S).$$

$$R_S = \mathrm{PerfMatch}(G' - S).$$

Each entry is indexed by a subset S **of external nodes.**

A Wild Attempt at $\mathrm{P} = \mathrm{P}^{\#\mathrm{P}}$

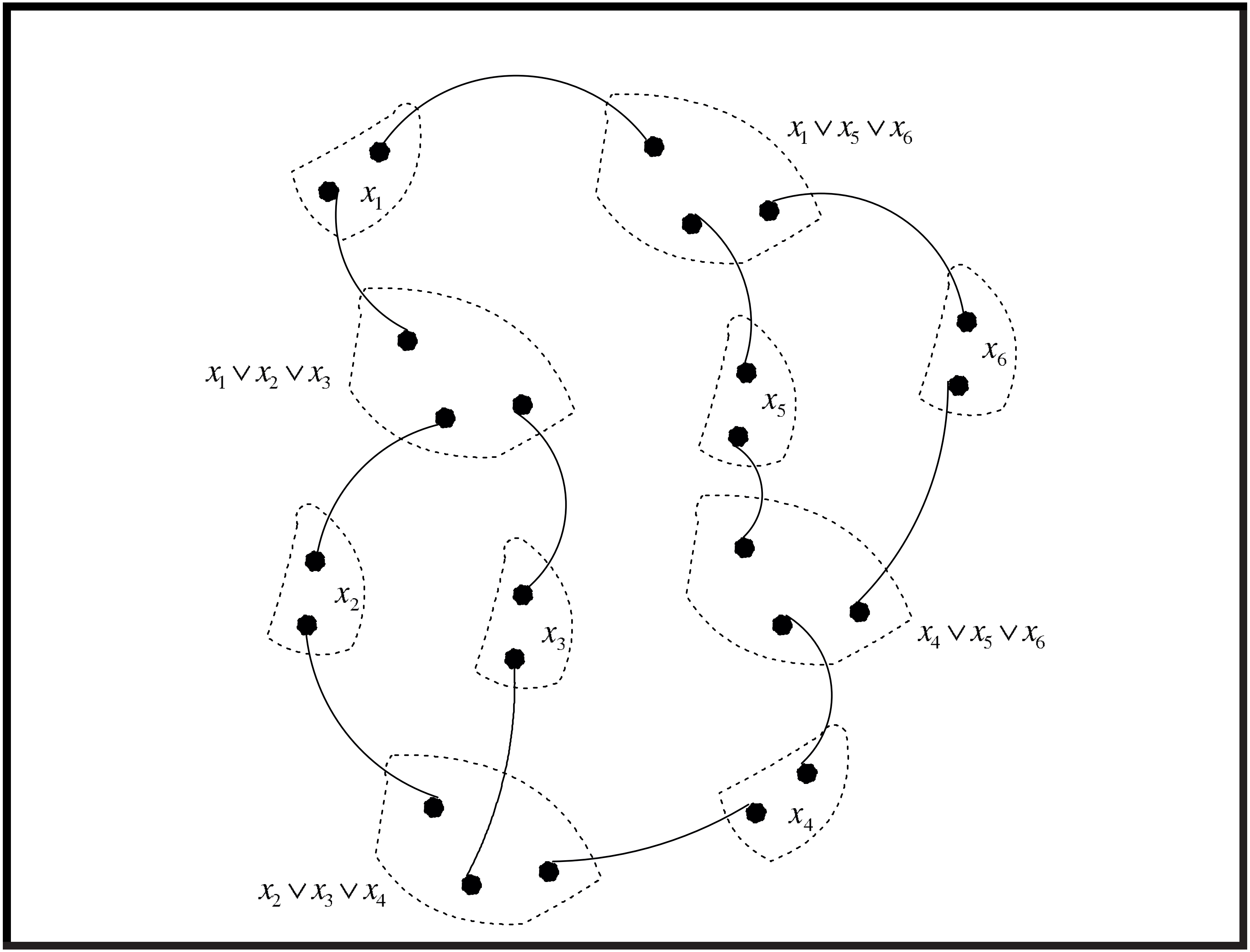
Consider Pl-Rtw-Mon-3CNF again:

#Pl-Rtw-Mon-3CNF

Input: **A planar graph G_Φ representing a Read-twice Monotone 3CNF Boolean formula Φ.**

Output: **The number of satisfying assignments of Φ.**

An Instance For #Pl-Rtw-Mon-3CNF


$x_1 \vee x_5 \vee x_6$
x_1
$x_1 \vee x_2 \vee x_3$
x_6
x_5
x_2
x_3
$x_4 \vee x_5 \vee x_6$
x_4
$x_2 \vee x_3 \vee x_4$

Recognizer Signature

Given Φ as a planar graph G_Φ.

Variables and clauses are nodes.

Edge (x, C): x appears in C.

For each clause C in Φ with 3 variables, we define

$$R_C = (0, 1, 1, 1, 1, 1, 1, 1),$$

where the 8 entries are indexed by $b_1 b_2 b_3 \in \{0, 1\}^3$.

Here $b_1 b_2 b_3$ corresponds to a truth assignment to the 3 variables.

R_C corresponds to an OR gate.

Generator Signature

For each variable x we want a generator G with signature $G^{00} = 1, G^{01} = 0, G^{10} = 0, G^{11} = 1$, or $(1, 0, 0, 1)^{\mathtt{T}}$ for short.

. . . to indicate that the fan-out value from x to C and C' must be consistent.

Exponential Sum

Now we can form the tensor product $\mathbf{R} = \bigotimes_C R_C$ and $\mathbf{G} = \bigotimes_x G_x$.

The sum

$$\langle \mathbf{R}, \mathbf{G} \rangle = \sum_{i_1, i_2, \ldots, i_e \in \{0,1\}} R_{i_1 i_2 \ldots i_e} G^{i_1 i_2 \ldots i_e}$$

counts exactly the number of satisfying assignments to Φ.

The indices of $\mathbf{R} = (R_{i_1 i_2 \ldots i_e})$ and $\mathbf{G} = (G^{i_1 i_2 \ldots i_e})$ match up one-to-one according to which x appears in which C.

Realizability Issue

If these signatures are indeed realizable as signatures of planar matchgates, then by

the Fisher-Kasteleyn-Temperley (FKT) method on planar perfect matchings, we would have shown

$$\#\mathrm{P} = \mathrm{NP} = \mathrm{P} \quad !!!$$

The above G is indeed realizable.

But R is not (realizable as standard signature).

Need more ideas ...

Basis Transformations

The 1st ingredient of the theory of holographic algorithms:

Matchgates

The 2nd ingredient of the theory:

A choice of linear basis

by which the computation is manipulated/interpreted.

Transformation Matrix

So let $\boldsymbol{b}$ denote the standard basis,

$$\boldsymbol{b} = [e_0, e_1] = \left[\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right].$$

Consider another basis

$$\boldsymbol{\beta} = [n, p] = \left[\begin{pmatrix} n_0 \\ n_1 \end{pmatrix}, \begin{pmatrix} p_0 \\ p_1 \end{pmatrix} \right].$$

Let $\boldsymbol{\beta} = \boldsymbol{b}T$. Denote $T = (t^i_j)$ and $T^{-1} = (\tilde{t}^i_j)$.

(Upper index is for row and lower index is for column.)

Contravariant and Covariant Tensors

We assign to each generator Γ a contravariant tensor $\boldsymbol{G} = (G^{\alpha})$.

Under a basis transformation,

$$(G')^{i_1' i_2' \ldots i_n'} = \sum G^{i_1 i_2 \ldots i_n} \tilde{t}_{i_1}^{i_1'} \tilde{t}_{i_2}^{i_2'} \cdots \tilde{t}_{i_n}^{i_n'} \qquad (1)$$

Correspondingly, each recognizer Γ gets a covariant tensor $\boldsymbol{R} = (R_{\alpha})$.

$$(R')_{i_1' i_2' \ldots i_n'} = \sum R_{i_1 i_2 \ldots i_n} t_{i_1'}^{i_1} t_{i_2'}^{i_2} \cdots t_{i_n'}^{i_n} \qquad (2)$$

After this transformation, the signature

$$(0, 1, 1, 1, 1, 1, 1, 1)$$

IS realizable.

Realization for the OR gate

So we want the following

$$(0, 1, 1, 1, 1, 1, 1, 1)$$

as a non-standard signature under some basis.

Let

$$\left[\begin{pmatrix} 1+\omega \\ 1-\omega \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right],$$

where $\omega = e^{2\pi i/3}$ **is a primitive third root of unity.**

The Transformation Matrix from R' to R

$$\left(\begin{pmatrix} 1+\omega & 1 \\ 1-\omega & 1 \end{pmatrix}^{-1}\right)^{\otimes 3}$$ **is** $\frac{1}{8}$ **times**

$$\begin{pmatrix}
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\
-1+\omega & 1+\omega & 1-\omega & -1-\omega & 1-\omega & -1-\omega & -1+\omega & 1+\omega \\
-1+\omega & 1-\omega & 1+\omega & -1-\omega & 1-\omega & -1+\omega & -1-\omega & 1+\omega \\
-3\omega & -2-\omega & -2-\omega & \omega & 3\omega & 2+\omega & 2+\omega & -\omega \\
-1+\omega & 1-\omega & 1-\omega & -1+\omega & 1+\omega & -1-\omega & -1-\omega & 1+\omega \\
-3\omega & -2-\omega & 3\omega & 2+\omega & -2-\omega & \omega & 2+\omega & -\omega \\
-3\omega & 3\omega & -2-\omega & 2+\omega & -2-\omega & 2+\omega & \omega & -\omega \\
3+6\omega & 3 & 3 & -1-2\omega & 3 & -1-2\omega & -1-2\omega & -1
\end{pmatrix}$$

Back to Standard Signature

By covariant transformation, (adding the last 7 rows),

$$(R_{i_1 i_2 i_3}) = \frac{1}{4}(0,1,1,0,1,0,0,1).$$

There indeed exists a matchgate with three external nodes with the standard signature $= \frac{1}{4}(0,1,1,0,1,0,0,1)$**.**

Thus,

$$R'_C = (0,1,1,1,1,1,1,1) = \frac{1}{4}(0,1,1,0,1,0,0,1)\left(\begin{pmatrix} 1+\omega & 1 \\ 1-\omega & 1 \end{pmatrix}\right)^{\otimes 3}.$$

Over Finite Fields

Over the field $\mathbf{Z}_7$ (but not Q) both the generators and recognizers are simultaneously realizable. They are realizable as non-standard signatures.

This gives $\#_7$Pl-Rtw-Mon-3CNF $\in$ P.

Characteristic 7 is Unique

Theorem

Characteristic 7 is the unique characteristic of a field for which there is a common basis of size 1 for generating $(1, 0, 0, 1)^{\mathtt{T}}$ and recognizing $(0, 1, 1, 1, 1, 1, 1, 1)^{\mathtt{T}}$.

Deeper connections with Mersenne numbers $2^p - 1$.

Complexity Dichotomy Theorems

Theorem

Let $\mathcal{F}$ be any finite set of real-valued symmetric constraint functions on Boolean variables. Then there are precisely three classes of #CSP($\mathcal{F}$) problems, depending on $\mathcal{F}$.

(1) #CSP($\mathcal{F}$) is in P.

(2) #CSP($\mathcal{F}$) is #P-hard, but solvable in P for planar inputs.

(3) #CSP($\mathcal{F}$) is #P-hard even for planar inputs.

Furthermore $\mathcal{F}$ is in class (2) iff there is a holographic algorithm based on matchgates and the planar problems are solved by the FKT algorithm.

Outlook

The kinds of algorithms that are obtained by this theory are quite unlike anything before and almost exotic.

The uncertainty of its ultimate prospect makes it exciting.

Is it possible to find an exotic but polynomial time algorithm for an NP-hard problem?

Back to P. vs. NP

We don't have any strong lower bounds.

The belief NP $\neq$ P is based on the experience that the usual algorithmic methods are insufficient for NP-hard problems.

So would it be possible that this new theory leads to a polynomial time algorithm for one of the NP-hard problems?

Valiant: "any proof of P $\neq$ NP may need to explain, and not only to imply, the unsolvability" of NP-hard problems using this approach.

What would Hilbert say?

Is the Computability Theory of Gödel, Church and Turing et. al. a more profound characterization of the mathematical universe of theorem, proof, verification, . . .

Or does P vs. NP capture more the essence of mathematics?

What would Hilbert say?

Some References

Some papers can be found on my web site

`http://www.cs.wisc.edu/~jyc`

THANK YOU!