THE FUTURE OF COMPUTER SCIENCE

Kristina Vuškovic
University of Leeds
Scheduling 1957 →

Increase in speed and memory
Improvements in programming languages, computer-human interface, ...

Algorithms

We can now solve problems we could not dream of solving a few decades ago
AND YET

DESPITE MORE THAN A CENTURY OF TECHNOLOGICAL ADVANCES IT SEEMS THAT SOME TRAIN JOURNEYS ARE SLOWER NOW THAN THEY WERE IN VICTORIAN TIMES

MANCHESTER PICCADILLY → OLDHAM

27 MINUTES 2017
16 MINUTES 1895
Annually the internet produces the amount of CO₂ as all 31 million cars in the UK going on a roadtrip around the world.

In 2016 data centers consumed 3% of global electricity and contribute 2% to the world's total CO₂ emissions.
FACEBOOK

A SOCIAL NETWORK THAT MAKES THE WORLD MORE CONNECTED

OR

ADVERTIZING COMPANY AND BIGGEST SURVEILLANCE-BASED ENTERPRISE IN THE HISTORY OF MANKIND
Flaubert (paraphrased by Julian Barnes)

"The railway would merely permit more people to move about, meet and be stupid"

Railway → Internet
FURTHER ACHIEVEMENTS OF

MASSIVE COMPUTING POWER

ILLUSION OF PROGRESS

IGNORANCE

UNREGULATION

WORKING TOGETHER
BRINGING DOWN THE GLOBAL ECONOMY

SUPER COMPUTERS + COMPLEX EQUATIONS
→ MODELS THAT DEPEND ON HOUSE PRICES RISING FOREVER AREN'T ALL THAT RISKY

IT'S ROLE?

Perhaps did not cause the housing bubble (that led to the current economic downturn) but it certainly helped
• by providing false reassurance
• by magnifying the scale
ALGORITHMS THAT ENABLE GIG ECONOMIES
UBER, DELIVEROO ...

PARLIAMENTARY INQUIRY INTO GIG ECONOMIES FOUND THAT

"UBER DRIVERS WERE TREATED AS VICTORIAN-STYLE SWEATED LABOUR"
ONE OTHER REASON THAT THEORETICIANS ARE MORE CAUTIOUS ABOUT PREDICTIONS FOR THE FUTURE:

WE ARE ACTUALLY AWARE OF HOW LITTLE WE KNOW

AND HOW MUCH TIME IT TAKES TO DEVELOP TOOLS AND TECHNIQUES THAT CAN ENABLE EFFECTIVE ANALYSIS OF BIG DATA
WHAT INSPIRES FUNDAMENTAL RESEARCH THAT YIELDS IMPORTANT ALGORITHMIC BREAKTHROUGHS?
Often it is not the actual applications but deep mathematical questions that were formulated even before computers.
Robertson and Seymour's Graph Minors Project

23 papers, 1980's → 2004

Structural study of minor-closed classes of graphs

Closed under: deleting vertices, deleting edges, contracting edges

E.g. forests, planar graphs...
Kuratowski's Theorem 1930
(reformulated by Wagner 1937)
A graph is planar iff it has no
K$_5$ or K$_{3,3}$ minor

Wagner's Conjecture (now R-S theorem)
Every minor closed class of graphs
can be characterized by a finite family
of excluded minors
ANOTHER FORMULATION:

IN EVERY INFINITE SET \( \{G_1, G_2, \ldots \} \) OF FINITE GRAPHS
THERE ARE 2 GRAPHS S.T. ONE IS A MINOR OF THE OTHER.

I.E. THE SET OF (ISOMORPHISM CLASSES OF) FINITE GRAPHS IS WELL-QUASI-ORDERED W.R.T. "MINOR" RELATION.
ROBERTSON AND SEYMOUR

IF A MINOR-CLOSED CLASS DOES NOT CONTAIN
ALL GRAPHS THEN EVERY GRAPH IN IT IS GLUED
TOGETHER IN A TREE-LIKE FASHION FROM GRAPHS
THAT CAN ALMOST BE EMBEDDED IN A FIXED SURFACE

TREE DECOMPOSITION, TREE WIDTH

HALIN (WAGNER'S STUDENT) 1964

INFINITE GRAPHS

REDECOVERED IN 1981 SIMULTANEOUSLY & INDEPENDENTLY
BY ARNBORG AND PROSKUROWSKI

WITH THE GOAL OF OBTAINING EFFICIENT ALGORITHMS
FOR HARD PROBLEMS

AND ROBERTSON AND SEYMOUR
TREE DECOMPOSITION

\((T, X)\)

\(\text{TREE } \forall v \in V(T), X_v\)

\(W = \max_{v \in V(T)} |X_v| - 1\)

\(\text{tw}(G) = \text{minimum width over all tree decompositions of } G\)

**Tree Decomposition of Width** \(r\) **Corresponds to** Decomposing a Graph into **Pieces of Size** \(\leq r+1\) **By** A Sequence of **Non-Crossing Cutsets of Size** \(\leq r\).
A graph \( H \in O(n^3) \) algorithm that given a graph \( G \) tests whether \( H \) is a minor of \( G \).

\[ \rightarrow \] **Every minor-closed property of graphs can be tested in polynomial time**.
FINDING TREE DECOMPOSITIONS

(ARNBORG, CORNEIL, PROSKUROWSKI 1987)

GIVEN GRAPH G AND INTEGER k
IT IS NPC TO DETERMINE WHETHER tw(G) ≤ k

(BODLANDER 1993)

O(k^3n) ALGORITHM TO CHECK IF tw(G) ≤ k
(SO FOR FIXED k, LINEAR TIME)
FURTHER ALGORITHMIC CONSEQUENCES FOR
GRAPH CLASSES WITH BOUNDED TREE WIDTH

MANY NP-HARD PROBLEMS CAN BE SOLVED
EFFICIENTLY BY DYNAMIC PROGRAMMING

IN FACT

(COURCELLE 1982)

ALL PROBLEMS THAT CAN BE FORMULATED IN
MONADIC SECOND ORDER LOGIC CAN BE SOLVED
IN LINEAR TIME ON GRAPHS OF BOUNDED TREE WIDTH
APPLICATIONS OF TREE DECOMPOSITION

• COMPILERS

Allocate registers by computing proper colorings on control flow graphs which turn out to have small tree width in practice (Thorup 1998)

More recently Krause (Isolde Adler's former PhD student)

First polynomial time graph coloring register allocator that optimally allocates registers for structured programs based on tree decomposition

It is now default register allocator in the major C compiler SDCC
• BIOINFORMATICS

USEFUL MODELS THAT HAVE SMALL TREEWIDTH
(OR AT LEAST IN LARGE PERCENT OF CASES)

→ CAN SOLVE EXACTLY
  GRAPH ISOMORPHISM
  MAX WEIGHT INDEPENDENT SET
  MAX WEIGHT CLIQUE

BY DYNAMIC PROGRAMMING IN LINEAR TIME
PARAMETRIZED ALGORITHMS AND COMPUTATIONAL EXPERIMENTS (PACE) CHALLENGE

SET UP IN 2016 TO PROMOTE THE USE OF TREE DECOMPOSITION IN PRACTICE.

ISOLDE WAS A CO-CHAIR OF THE 1ST PACE CONTEST

LUKAS LARISCH AND FELIX SALFELDER (LEEDS)
WINNERS OF PACE 2017 TRACK 1:

FOR BEST IMPLEMENTATION OF AN EXACT TREE WIDTH ALGORITHM
HEREDITARY GRAPH CLASSES
CLOSSED UNDER DELETING VERTICES

FOR A FAMILY OF GRAPHS $F$
$G$ is $F$-FREE if $\forall F \in F$ $G$ does not contain an induced subgraph isomorphic to $F$

HOLE = CHORDLESS CYCLE OF LENGTH $\geq 4$

$G$ is CHORDAL if it is HOLE-FREE

CLIQUES

SPLIT GRAPHS

TREE WIDTH $(K_n) = n-1$

CLIQUE WIDTH $(K_n) = 2$

CLIQUE WIDTH UNBOUNDED
$$G \text{ CHORDAL} \Rightarrow G \text{ IS A CLIQUE OR}
G \text{ HAS A CLIQUE CUTSET}$$

Blocks of decomposition
$$G_i = G[S U C_i]$$

$\rightarrow$ Efficient (Linear time) algorithms
Recognition, $X, W, L$...
PERFECT GRAPHS

INTRODUCED BY BERGE IN LATE 50'S AND EARLY 60'S

MOTIVATED BY STUDY OF COMMUNICATION THEORY

LINKS VARIOUS MATHEMATICAL DISCIPLINES

IN UNEXPECTED WAYS:

GRAPH THEORY

COMBINATORIAL OPTIMIZATION

SEMIDEFINITE PROGRAMMING

POLYHEDRAL AND CONVEXITY THEORY
\( \chi(G) \geq \omega(G) \quad \forall G \)

**Bound can be tight**

**But also arbitrarily bad**

\( G \) is **perfect** if \( \chi(H) = \omega(H) \quad \forall H \subseteq G \)

Strong Perfect Graph Conjecture (Berge 1961)

\( G \) is perfect if and only if \( G \) is (odd hole, odd antihole)-free.
(Lovász '72) Perfect Graph Theorem

G perfect $\iff \overline{G}$ perfect

(Grötschel, Lovász, Schrijver '81)

\( \chi, \omega \) poly-time for perfect graphs using ellipsoid method

Open: Can we solve these optimization problems for perfect graphs in poly-time by purely combinatorial algs

(Chudnovsky, Robertson, Seymour, Thomas '02)

Strong Perfect Graph Theorem

(Chudnovsky, Cornuéjols, Liu, Seymour, Vušković '03)

Perfect graphs can be recognized in poly-time
COMMONLY APPEARING CUTSETS IN DECOMPOSITIONS OF COMPLEX HEREDITARY CLASSES

STAR CUTSET

2-JOIN
TOTALLY UNIMODULAR AND BALANCED MATRICES

DECOMPOSITION BASED RECOGNITION ALGORITHMS

(SEYMOUR '80)

REGULAR MATROIDS
1-, 2-, 3-SEPARATIONS

(CONFORTI, CORNUEJOLS, RAO '91)

BALANCED 0,1 MATRICES
2-JOIN, DOUBLE STAR CUTSET

(CONFORTI, CORNUEJOLS, KAPOOR, VUŠKOVIC '93)

BALANCEABLE MATRICES (BIPARTITE GRAPHS)
2-JOIN, 6-JOIN, DOUBLE STAR CUTSET

\[ 3\text{-SEPARATIONS} \]
(Conforti, Cornuéjols, Vuškovic '01)

G 4-hole-free Berge $\Rightarrow$ G is bipartite
Line graph of bipartite
Or G has Star cutset or 2-join

(Chudnovsky, Robertson, Seymour, Thomas '02)

G Berge $\Rightarrow$ G or $\overline{G}$ is bipartite
Line graph of bipartite
Double split
Or G has

- Skew cutset
- 2-join
2-joins easy
star cutsets big problem!

(Conforti, Rao '89)

Recognizing linear balanced matrices

Cleaning

1. \( G \rightarrow \text{clean } G' \quad G \in \mathcal{B} \iff G' \in \mathcal{B} \)

2. Apply decomposition based recognition algorithm to \( G' \)

(Can design blocks of decomposition that are class-preserving for clean graphs and keep the size of decomposition tree polynomial)

Cleaning used in recognition algorithms for balanceable matrices/bipartite graphs/even-hole-free graphs

(Conforti, Cornuéjols, Kapoor, Vuškovic '97)

Perfect graphs
(Trotignon, Vuškovic '09)
2-JOINS AND W, X

(Chudnovsky, Trotignon, Trunck, Vuškovic '13)
PERFECT GRAPHS WITH NO SKEN CUTSET \( \lambda \rightarrow W, X \)

WORKED ON TRIGRAPHS

(Chudnovsky, Lagoutte, Seymour, Spirkl '15)
PERFECT GRAPHS WITH BOUNDED CLIQUE NUMBER \( X \)

(Chudnovsky, Lo, Haffray, Trotignon, Vuškovic '15)
4-HOLE-FREE PERFECT \( X \)