3-Coloring, 3-Edge-Coloring, and Constraint Satisfaction

David Eppstein

UC Irvine

These problems are NP-complete

Why do worst-case analysis of exact algorithms?

- With fast computers we can do exponential-time computations of moderate and increasing size
- Algorithmic improvements are even more important than in polynomial-time arena
- Graph coloring is useful e.g. for register allocation and parallel scheduling
- Approximate coloring algorithms have poor approximation ratios
- Interesting gap between theory and practice

Constraint Satisfaction

Given *n* variables, each with a set of possible values

m constraints forbid certain value combinations

find assignment of values to variables
obeying all constraints

values per variable

(a,b)-CSP

variables per constraint

Previous CSP Results

Beigel & Eppstein 1995

Messy case analysis

$(3,2)$-CSP $\mathcal{O}(1.38028^n)$

Randomized restriction

$(k,2)$-CSP $\mathcal{O}\left(\left(\frac{k}{2}\right)^n\right)$

Feder & Motwani 1998

Random permutation of variables

$(k,2)$-CSP $\mathcal{O}(k!^{n/k})$

Schöning 1999

Random walk among assignments

(a,b)-CSP $\mathcal{O}\left(\left(\frac{ab-a}{b}\right)^n\right)$

New CSP Results

$(3,2)$-CSP	$\mathcal{O}(1.36443^n)$
$(4,2)$-CSP	$\mathcal{O}(1.8072^n)$
$(k,2)$-CSP	$\mathcal{O}((0.4518k)^n)$

Ideas:

Continued messy case analysis

Stop backtracking when solvable by matching

Define problem size $= n_3 + (2-\epsilon)n_4$,
choose optimal ϵ for analysis

Combine w/random restriction for $(k,2)$-CSP

3-Vertex-Coloring

Previous results

Lawler 1976
$\mathcal{O}(3^{n/3})$

Schiermeyer 1994
$\mathcal{O}(1.415^n)$

Beigel & Eppstein 1995
$\mathcal{O}(1.3446^n)$

New result

$\mathcal{O}(1.3289^n)$

3-Coloring Main Idea

(Same as Beigel & Eppstein 1995)

Find small set S with many neighbors

Choose colors for vertices in S

Solve remaining vertices as $(3, 2)$-CSP

Neighbors of colored vertices are restricted
to two colors, eliminated from $(3, 2)$-CSP

Time: $\mathcal{O}(3^{|S|} 1.3645^{|V(G) \setminus (S \cup N(S))|})$

How to Find S?

Beigel & Eppstein:

Group all vertices into height-two trees
Local improvement from greedy start
(messy case analysis)

New method:

Eliminate big clumps of degree-3 vertices
(else good reduction to smaller coloring instances)

Find big forest w/degree-4 internal nodes
must cover constant fraction of graph

Remaining vertices $\Rightarrow$ height-two trees
few grandchildren per tree
start with fractional assignment nodes-trees
then use integer flow to make 0-1 assignment

$S =$ big forest internal nodes
$+$ height-two tree roots

3-Edge-Coloring

Previous result

Beigel & Eppstein 1995
$\mathcal{O}(1.5039^n)$

(minor mods to vertex coloring alg)

New result

$\mathcal{O}(2^{n/2})$

3-Edge-Coloring Main Idea

Generalize problem:

Add constraints forcing pairs of edges
to have different colors

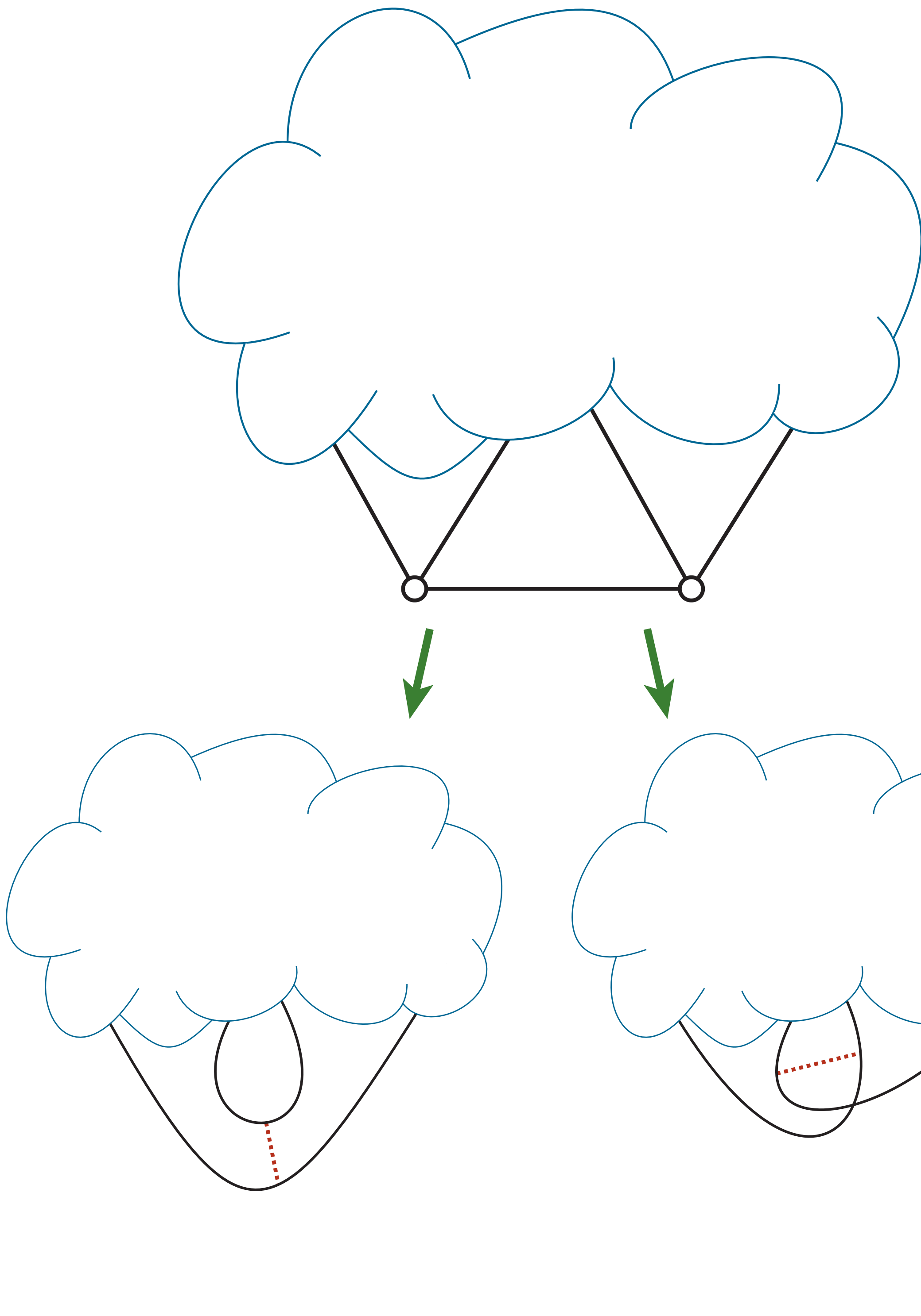
Eliminate edges with four neighbors

Reduce to two subproblems with two fewer vertices
(but one more constraint)

Find many independent reductions by matching

Transform remaining problem to vertex coloring

Edge intersection graph + edge per constraint



3-Edge-Coloring Analysis

Let $m_i = \#$ edges with i neighbors

Can find $m_4/3$ indep. reductions
(else not 3-colorable)

$2^{m_4/3}$ subproblems after reduction
m_3 edges per subproblem

Time $= \mathcal{O}(1.3289^{m_3} 2^{m_4/3})$

Maximized when $m_3 = 0$, $m_4 = 3n/2$:

$$\mathcal{O}(2^{n/2})$$

Conclusions

More efficient algorithms for
several important NP-complete problems

Many other problems for further work

Recent progress:

General graph chromatic number

$\mathcal{O}(2.4422^n)$ [Lawler 1976] $\Rightarrow$ $\mathcal{O}(2.4150^n)$