Advanced Algorithmics

Strategies for Tackling Hard Problems

Sebastian Wild

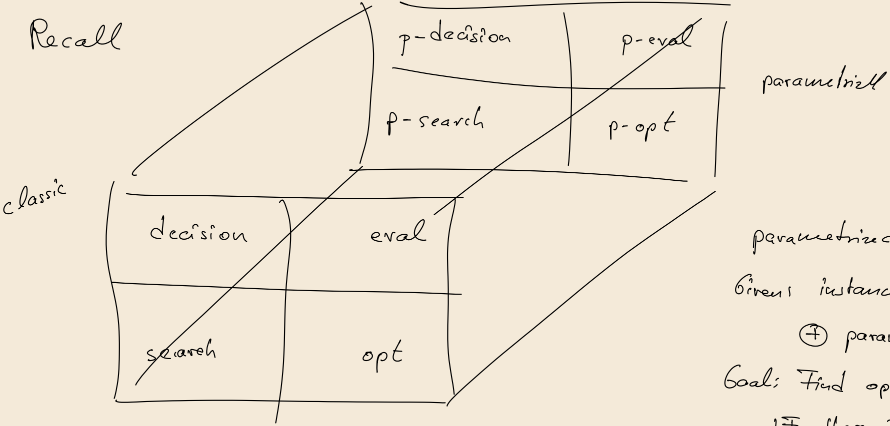
Markus Nebel

Lecture 6

2017-05-08

Glossary of Problem Types 3

Recall



parametrized optimization problem:

Given: instance of opt problem
⊕ parameter threshold

Goal: Find optimal solution
IF there is no better
otherwise abort

as before: decision are used to study
complexity
lower bounds & hardness

easy for specific problem by ~~black~~-box reuse

binary search

keep parameter

Depth-Bounded Search for Vertex Cover

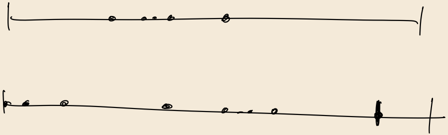
procedure SimpleFptVertexCover($G = (V, E), k$):
 if $E = \emptyset$ **then return** $\emptyset$
 if $k = 0$ **then return** "not possible" *// truncate search*
 Choose $\{v, w\} \in E$ (arbitrarily)
 For u **in** $\{v, w\}$ **do**:
  $G_u := \big(V \setminus \{u\}, E \setminus \{\{u, x\} \in E\}\big)$ *// Remove u from G*
  $C_u := \{u\} \cup$ SimpleFptVertexCover($G_u, k - 1$)
 if $C_v =$ "not possible" **then return** C_w
 if $C_w =$ "not possible" **then return** C_v
 if $|C_v| \le |C_w|$ **then return** C_v else **return** C_w

- Does not need explicit checks of solution candidates!
- runs in time $\mathcal{O}\big(2^k \cdot (n + m)\big)$ $\rightsquigarrow$ fpt-algorithm for p-Vertex-Cover
- Only uses k to *truncate* branches.

$\rightsquigarrow$ With a breadth-first evaluation order, $\mathcal{O}\big(2^k \cdot (n + m)\big)$ for k size of *optimal* VC without knowing it!

replace binary search by galloping search

exponential search



minimization problem $\in FPT$

$f(k) \cdot p(n)$

Independent Set on Planar Graphs

Recall: general problem p-INDEPENDENT-SET is $\mathcal{W}[1]$-hard.

Definition 3.32 (p-PLANAR-INDEPENDENT-SET)

Given: a *planar* graph $G = (V, E)$ and $k \in \mathbb{N}$

Parameter: k

Question: $\exists V' \subset V \quad : \quad |V'| \geq k \ \wedge \ \forall u, v \in V' : \{u, v\} \notin E$?

NP-complete

FPT

Theorem 3.33 (Depth-Bounded Search for Planar Independent Set)

p-PLANAR-INDEPENDENT-SET is in $\mathcal{FPT}$ and can be solved in time $\mathcal{O}(6^k n)$.

Elementary Knowledge on Planar Graphs

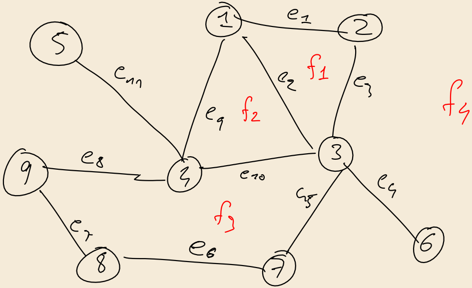
Theorem 3.34 (Euler's formula)

In any finite, connected planar graph G with n nodes, m edges f holds $n - m + f = 2$.

Corollary 3.35

A simple planar graph G on $n \geq 3$ nodes has $\underline{m \leq 3n - 6}$ edges.
The average degree in G is < 6.

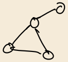
↳ probabilistic method
$\Rightarrow \underline{\exists v \ \deg(v) \leq 5}$



$n - m + f = 2$
$9 - 11 + 4 = 2$

$f = 1$ ✓

$f \geq 2 \rightsquigarrow n \geq 3$

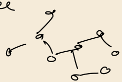


all faces have ≥ 3 edges

$\sum_f \text{edges}(f)$

$3f \leq 2m$
$\parallel$
$6 - 3n + 3m$

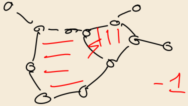
Proof: ∘ tree



$f = 1$

$n = m + 1$

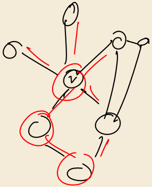
∘ G no a tree



-1 $\quad -1$

Depth-Bounded Search for Planar Independent Set

procedure planarIndependentSet($G = (V, E), k$):
    if $k > |V|$ **then return** "not possible" // *truncate search*
    if $E = \emptyset$ **then return** V
    Choose $v \in V$ with minimal degree; let $w_1, \ldots, w_d$ be v's neighbors
    // *By planarity, we know $d \le 5$.*
    for u **in** $\{v, w_1, \ldots, w_d\}$ **do**
        $D := \{u\} \cup N(u)$
        $G_u := \big(V \setminus D, E \setminus \{\{x,y\} \in E : x \in D\}\big)$ // *Delete u and its neighbors*
        $I_u := \{u\} \cup$ planarIndependentSet($G_u, k-1$)
    return largest I_u or "not possible" **if** none exists



Correctness: Any maximal IS has to contain one node from ✓

Search Space: ≤ 6 recursive calls

always make k one smaller

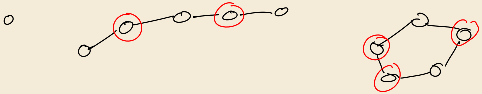
$\Rightarrow \le 6^k$ Line 4 linear time

A Better Algorithm for Vertex Cover

Recall: Branching on endpoints of k edges gives search space of size 2^k for VERTEX-COVER. Can we do better?

Theorem 3.38 (Depth-Bounded Search for Vertex Cover)
p-VERTEX-COVER can be solved in time $\mathcal{O}(1.4656^k n)$.

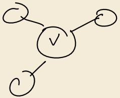
(1) If G has only degrees $0, 1, 2$



trivial linear time

(2) v / $N(v)$ reduction rules

$\deg(v) \geq 3$



same algorithmic structure

Depth-Bounded Search for Vertex Cover

procedure BetterFptVertexCover($G = (V, E), k$):
    if $E = \emptyset$ **then return** $\emptyset$
    if $k = 0$ **then return** "not possible" *// truncate search*
    if all node have degree ≤ 2 **then**
        Find connected components of G
        for each component G_i **do**
            Fill C_i by picking every other node,
            starting with the neighbor of a degree-one node if one exists
        $C := \bigcup C_i$
        if $|C| \overset{\leq}{\cancel{\geq}} k$ **then return** C else **return** "not possible"
    Choose v with maximal degree, let $w_1, \ldots, w_d$ be its neighbors // $d \geq 3$
    For D **in** $\{\{v\}, \{w_1, \ldots, w_d\}\}$ **do**:
        $G_D := \big(V \setminus D, E \setminus \{\{x, y\} \in E : x \in D\}\big)$ *// Remove D from G*
        $C_D := D \cup$ BetterFptVertexCover($G_u, k - |D|$)
    return smallest C_D or "not possible" **if** none exists

Theorem 3.36 (Linear Recurrences)

Let $d_1, \ldots, d_i \in \mathbb{N}$ and $d = \max d_j$. $\vec{d} = (1, 3)$

The solution to the *homogeneous linear recurrence equation*

$$T_n = T_{n-d_1} + T_{n-d_2} + \cdots + T_{n-d_i}, \quad + f_n \qquad (n \geq d)$$

is always given by

$$T_n = \sum_{\ell} \sum_{j=0}^{\mu_\ell - 1} c_{\ell,j} \, z_\ell^n \, n^j$$

where we sum over all roots z_ℓ of multiplicity μ_ℓ of the so-called *characteristic polynomial* $z^d - z^{d-d_1} - z^{d-d_2} \cdots - z^{d-d_i}$. $z^3 - z^2 - 1$ $1.4..$

The d coefficients $c_{\ell,j}$ are determined by the d initial values $T_0, T_1, \ldots, T_{d-1}$. ◀

Corollary 3.37

$T_n = \mathcal{O}(z_0^n n^d)$ for z_0 the root of the characteristic polynomial with *largest absolute value*. ◀

$$T_n = T_{n-d_1} + \cdots + T_{n-d_i} \quad (n \geq d) \qquad \vec{d} = (d_1, \ldots, d_i) \qquad d = \max \vec{d}$$

$$T(z) = \sum_{n \geq 0} T_n z^n$$

$$T(z) = 1 T_0 + \cdots + z^{d-1} T_{d-1} + \sum_{n \geq d} \left(T_{n-d_1} + \cdots + T_{n-d_i} \right) z^n$$

$$= \quad 〃 \quad + \sum_{j=1}^{i} z^{d_j} \sum_{n \geq d} T_{n-d_j} z^{n-d_j}$$

$$= \quad 〃 \quad + \underbrace{\sum_{j=1}^{i} z^{d_j} \left(T(z) - \sum_{n=0}^{d-d_j-1} T_n z^n \right)}_{A}$$

$$T(z) - \sum_{j=1}^{i} z^{d_j} T(z) = 1 T_0 + \cdots + z^{d-1} T_{d-1} - \sum_{j=1}^{i} d_j \sum_{n=0}^{d-d_j-1} T_n z^n$$

$$T(z) = \frac{A}{\underbrace{1 - \sum_{j=1}^{i} z^{d_j}}_{=B}}$$

$$\left(z^d - z^{d-d_1} - \cdots - z^{d-d_i} \right) / z^d$$

first: change of variables $z \to 1/z$

$$T(z) = \underbrace{\frac{A_1}{\left(1 - \frac{z}{z_0}\right)} + \frac{A_2}{\left(1 - \frac{z}{z_0}\right)^2}}_{z_0 \text{ a root of } B} + \cdots + \frac{B_1}{\left(1 - \frac{z}{z_1}\right)} + \frac{B_2}{\left(1 - \frac{z}{z_1}\right)^2} + \cdots$$

z_1 root of B

$$\sum_{n \geq 0} z^n = \frac{1}{1-z} \qquad (|z| < 1)$$

$$T_n = A_1 \cdot z_0^{-n} + A_2 n z_0^{-n} + A_3 \cdot n^2 z_0^{-n} + \cdots \qquad \square.$$

$d = (1, 3)$ $\qquad z^3 - z^2 - 1$

T_k = size of search space for parameter k

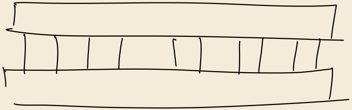
$T_k \leq T_{k-1} + T_{k-3}$ linear recurrence with constant coefficients

Definition 3.39 (p-Closest-String)

Given: S set of m strings $s_1, s_2, \ldots, s_m$ of length L over alphabet Σ and a $k \in \mathbb{N}$.

Parameter: k

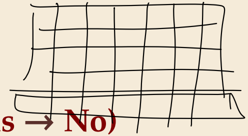
Question: Is there a string s for which $d_H(s, s_i) \leq k$ holds for all $i = 1, \ldots, m$?



d_H = # mismatch positions

Definition 3.40 (Dirty Column)

A column of the $m \times L$ matrix corresponding to m strings of length L is called *dirty* if it contains at least 2 different symbols.



Lemma 3.41 (Many Dirty Columns → No)

Let an instance to CLOSEST-STRING with m strings of length L and parameter k be given. If the corresponding $m \times L$ matrix contains more than $m \cdot k$ dirty columns, then no solution for the given instance exists.

Proof: $> m \cdot k$ columns , m strings

$\exists$ string that participates in $\geq k+1$ mismatch columns