Advanced Algorithmics

Strategies for Tackling Hard Problems

Sebastian Wild

Markus Nebel

Lecture 5

2017-05-04

3.1 Problem Kernels

Preprocessing

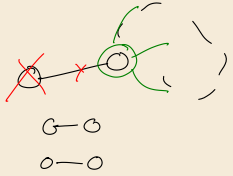
Idea: Reduce size of the instance (in poly-time) without changing the outcome

heuristics CNF-SAT : clause one literal

→ reduces the size, but in we not

Examples: Vertex Cover (VC)

- isolated vertices
- vertices of degree 1



Assume p-Vertex-Cover

- $H = \{v \in V : \deg(v) > k\}$

Remove H from G

Adjust parameter $k - |H|$

Buss' reduction for VC

G' = resulting graph — all vertices have degree $\leq k$ $\quad \geq 2$

$\Rightarrow$ k nodes can only $\leq k^2$ edges ; if $m > k^2$ $\not\in$ No instance

$\rightsquigarrow$ can assume $m \leq k^2$ $\quad n \leq m \leq k^2$

$\rightsquigarrow$ size of remaining instance is $O(k^2)$

Definition 3.27 (Kernalization)

Let (L, κ) be a parameterized problem. A function $K : \Sigma^\star \to \Sigma^\star$ is *kernelization* of L w.r.t. κ if it maps any $x \in L$ to an instance $x' = K(x)$ with $k' = \kappa(x')$ so that

1. (self-reduction) $x \in L \iff x' \in L$
2. (poly-time) K is computable in poly-time.
3. (kernel-size) $|x'| \leq g(k)$ for some computable function g

We call x' the *(problem) kernel* of x and g the *size of the problem kernel*. ◀

Theorem 3.28 (Buss's Reduction is Kernelization)
Buss' reduction yields a kernelization for p-Vertex-Cover with kernel size $\mathcal{O}(k^2)$. ◀

see above

□

Theorem 3.29 (FPT ↔ kernel)
A computable, parameterized problem (L, κ) is fixed-parameter tractable if and only if there is a kernelization for L w.r.t. κ. ◀

Proofs "<=" we have kernelization K, A decider for L in time T

FPT-algo

w.l.o.g. T weakly inc

(1) $x \overset{?}{\in} L \rightsquigarrow x' = K(x)$

$|x'| \leq g(\underbrace{\kappa(x)}_{k})$

poly-time

(2) $A(x')$ runs in time $T(|x'|) \underset{\text{inc.}}{\leq} T(g(k)) =: f(k)$

$\Rightarrow O(f(k) + p(|x|))$ FPT-algo.

"=>" Let A be fpt-algo for (L, κ) with time $f(k)\, n^c$

Kernelization

(1) Simulate A on x for $\leq n^{c+1}$ steps
(poly-time)

Case 1: A terminated $\rightsquigarrow$ Know answer
Return some small trivial
Yes / No - instance

Case 2: A not finished
$\rightarrow$ $n < f(k)$ $\qquad n^{c+1} < n^c f(k)$
$\Rightarrow$ Original instance is a kernel $\square$

Theorem 3.30 (Kernel for Max-SAT)

p-MAX-SAT has a problem kernel of size $\mathcal{O}(k^2)$ which can be constructed in linear time. ◀

Proof: $(x \vee y \vee \bar{z}) \wedge (x \vee y \vee \bar{z}) \wedge (\bar{x} \vee z) \wedge (\bar{y} \vee z)$

$\rightsquigarrow \{\{x, y, \bar{z}\}, \{x, y, \bar{z}\}, \{\bar{x}, z\}, \{\bar{y}, z\}\}$

(*) Assumption: Each variable shows up at most once per clause

Observation: m = #clauses

assume $k \leq \lceil \frac{m}{2} \rceil$

always possible: choose any assignment

check if fulfills $\geq k$ clauses

if not negate it $\rightsquigarrow \lceil \frac{m}{2} \rceil$ satisfied.

$\rightsquigarrow$ only interesting case is $k > \lceil \frac{m}{2} \rceil$ $\quad m < 2k$ (***)

We partition the clauses

$F_L = \{$ clauses : $\geq k$ literals $\}$

$\rightsquigarrow$ (***) $\geq k$ variables

$F_S = \{ \; \dots \; < k \; \dots \; \}$

If $|F_L| =: L \geq k$ Yes-instance

since we can pick k variables from k long clauses

to make k clauses true

$\rightsquigarrow$ remains $L < k$

consider $(F_S, k-L)$

$(F_S, k-L)$ Yes $\Leftrightarrow$ (F, k) Yes

$\Rightarrow$ If $(F_S, k-L)$ is Yes $\rightsquigarrow$ assignment $k-L$ short clauses

this cannot fix more than $k-\ell$ variables

for ℓ long clauses, find ℓ "free" variables

'$\subseteq$' trivial.

F_S has only clauses with $\leq k$ literals

$\Rightarrow$ $< k \cdot m \leq 2k^2$

computable in poly-time $\square$

Corollary : p-Max-Sat $\in$ FPT

Vertex Cover as (Integer) Linear Program

Consider optimization version of Vertex-Cover:
Given: Graph $G = (V, E)$
Goal: Vertex cover of G with minimal cardinality.

$\rightsquigarrow$ equivalent to the following linear program

$$\begin{aligned} &\min \textstyle\sum_{v \in V} x_v \\ &\text{s. t. } x_u + x_v \geq 1 \quad \text{for all } \{u, v\} \in E \\ &\quad \sout{x_v \in \{0, 1\} \quad \text{for all } v \in V} \end{aligned}$$

Consider *relaxation* to $x_v \in \mathbb{R}$, $x_v \geq 0$.
$\rightsquigarrow$ LP that can by solved in poly-time.

black box

For an *optimal* solution $\vec{x}$ of the *relaxation*, we define

$$\sum x_v \leq \sum x_v^I$$

$$\begin{aligned} I_0 &= \{v \in V : x_v < \tfrac{1}{2}\} \\ V_0 &= \{v \in V : x_v = \tfrac{1}{2}\} \\ C_0 &= \{v \in V : x_v > \tfrac{1}{2}\} \end{aligned}$$

Theorem 3.31 (Kernel for Vertex Cover)
Let $(G = (V, E), k)$ an instance of p-Vertex-Cover.

1. There exists a minimal vertex cover S with $C_0 \subseteq S$ and $S \cap I_0 = \emptyset$.
2. V_0 implies a problem kernel $\big(G[V_0], k - |C_0|\big)$ with $|V_0| \leq 2k$.

Here $G[V_0]$ is the induced subgraph of V_0 in G.

Proof: ad 1 Assume S is optimal VC for G

$S' = (S \setminus I_0) \cup C_0$ is also optimal VC

$= (S \setminus S_I) \cup \overline{S}_C$

$S_I = S \cap I_0 \qquad \overline{S}_C = C_0 \setminus S$

"S' VC" only edges with I_0 endpoints could remain uncovered

$e = \{v, w\}$, $v \in I_0 \Rightarrow x_v < \frac{1}{2}$

$\Rightarrow x_w > \frac{1}{2}$ since $x_v + x_w \geq 1$

$\Rightarrow w \in C_0 \qquad S' \supseteq C_0 \qquad e$ covered.

"$|S'|$ opt" $\quad |\overline{S_c}| \overset{!}{\leq} |S_I|$

Define $\varepsilon := \min\{x_v - \frac{1}{2} : v \in C_0\} > 0$

Define $\vec{x}' = \vec{x}$ except

$u \in S_I \quad x'_u := x_u + \varepsilon$

$v \in \overline{S_c} \quad x'_v := x_v - \varepsilon \geq \frac{1}{2}$

$\overline{S_c} \subseteq C_0$

$\vec{x}'$ also fulfills constraints of LP

$\forall e = \{u,v\} \quad x'_u + x'_v \geq 1 \quad C_0 \setminus S$

only interesting edges have $v \in \overline{S_c}$ (for other $x'_u \geq x_u$)

(1) $u \in I_0 \setminus S$

$\rightsquigarrow u \notin S, v \notin S \quad \not\subseteq S \quad VC$

(2) $u \in S_I \quad \rightsquigarrow \quad x'_v + x'_u = x_v + x_u \geq 1$

(3) $u \notin I_0 \Rightarrow x'_u \geq \frac{1}{2}, \; x'_v \geq \frac{1}{2} \quad x'_v + x'_u \geq 1$

Since $\vec{x}$ optimal, $\sum_V x'_v \geq \sum_V x_v \Rightarrow |S_I| - |\overline{S_c}| \geq 0$

$\sum_V x_v + \varepsilon(|S_I| - |\overline{S_c}|) \Rightarrow |\overline{S_c}| \leq |S_I|$ ☐(1)

ad 2 LP objective $\leq$ ILP objective for reduced instance $(G[V_0], k - |C_0|)$

$\Rightarrow |S| \geq \sum_{v \in V_0} x_v = \frac{1}{2}|V_0|$

opt. vertex cover for $G[V_0]$

We apply the above reduction iteratively, until the optimal $x = (0.5, \ldots, 0.5)$

If now $|V_0| > 2k$, we know for sure that there is no VC (with k nodes)

$\rightarrow$ return no-instance

otherwise $|V_0| \leq 2k \rightsquigarrow$ kernel. $\square$

3.2 Depth-Bounded Exhaustive Search

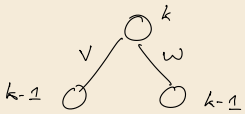
After a kernelization, we still have solve the kernel

Example: VC

for every edge $\{v, w\}$: either take v or w

- solution gets v
  remove v from graph
  solve resulting problem recursively.
  $\rightsquigarrow S_v$
- solution with w
  remove w
  recurse on
  $\rightsquigarrow S_w$

return smaller S_v, S_w



2^k