Fixed-Parameter Algorithms, IA166

Sebastian Ordyniak

Faculty of Informatics
Masaryk University Brno

Spring Semester 2013

Outline

FPT versus W[1]-hard

Comparison of a running time of 2^k to a running time of n^k by considering the quotient $\frac{n^k}{2^k}$:

	$n = 50$	$n = 150$
$k = 2$	625	5625
$k = 5$	390625	31640625
$k = 20$	$1.8 \cdot 10^{26}$	$2.1 \cdot 10^{35}$

fast FPT versus faster FPT

Comparison of a running time of 1.29^k to a running time of 2^k by considering the quotient $\frac{2^k}{1.29^k} = 1.54^k$:

k	2	10	20	40	80
1.54^k	9	80	6240	$3.9 \cdot 10^7$	$1.5 \cdot 10^{15}$

Outline

Bounded Search Tree

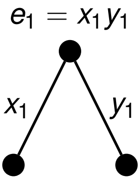
Recall How We solved MINIMUM VERTEX COVER

Input: Graph G and integer k.

$$e_1 = x_1 y_1$$

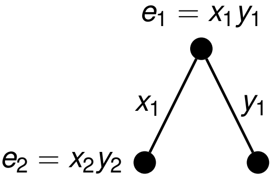
Recall How We solved MINIMUM VERTEX COVER

Input: Graph G and integer k.



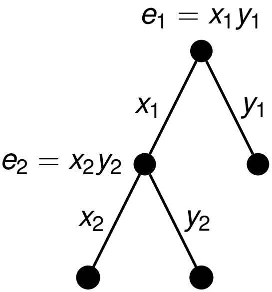
Recall How We solved MINIMUM VERTEX COVER

Input: Graph G and integer k.



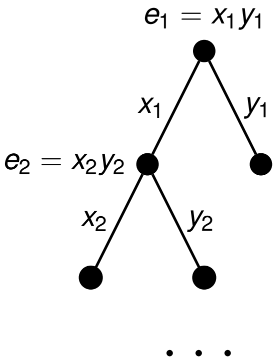
Recall How We solved MINIMUM VERTEX COVER

Input: Graph G and integer k.



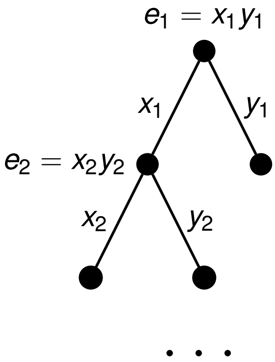
Recall How We solved MINIMUM VERTEX COVER

Input: Graph G and integer k.



Recall How We solved MINIMUM VERTEX COVER

Input: Graph G and integer k.



Running time

- at every node there are 2 choices;
- every choice decreases k;
- height of the search tree is at most k;
- complete search possible in $O(2^k n^c)$

Detailed Analysis

Algorithm for MINIMUM VERTEX COVER

VC(G, k) *{*
If $|E(G)| = 0$ *return* YES*;*
If $k = 0$ *return* NO*;*

Let $\{u, v\} \in E(G)$*:*
return VC$(G \setminus \{u\}, k-1)$ *Or* VC$(G \setminus \{v\}, k-1)$*;*
}

Theorem

The algorithm VC(G, k) solves MINIMUM VERTEX COVER in time $(2^k - 1)p(n) = O(2^k p(n)) = O^*(2^k)$.

Detailed Analysis

Theorem

The algorithm VC(G, k) solves MINIMUM VERTEX COVER in time $(2^k - 1)p(n) = O(2^k p(n)) = O^*(2^k)$.

Proof:

Proof using induction over k:

IB: VC$(G, 0)$ solves MINIMUM VERTEX COVER in time $O(1)$;

IS: VC(G, k) needs time
$p(n)+2(2^{k-1}-1)p(n-1) \leq p(n)(1+2^k-2) = (2^k-1)p(n)$.

$\square$

Bounded Search Tree Method

We build a search tree T such that at every node of T the following holds:

- the branching rules are exhausitive, i.e., every solution has to agree with at least one of the branches;
- every branch decreases k by at least 1
- the time spend is at most some polynomial p in n

Bounded Search Tree Method

Remarks

- The running time of such an algorithm is at most $O(|V(T)|p(n))$.
- The size of T is at most $O(B^k)$ where B is the maximum number of branches of any node of T.
- If B can be bounded by a function b of k we obtain an FPT-algorithm with running time $O((b(k))^k p(n))$

Outline

Minimum Vertex Cover — revisited

We have seen that Minimum Vertex Cover can be solved in time $O^*(2^k)$. Can we do better?

Observation

Minimum Vertex Cover can be solved in polynomial-time on graphs with maximum degree 2.

Observation

Let G be graph and $v \in V(G)$ then every vertex cover S of G either:

- contains v, or
- contains all neighbors of v.

Minimum Vertex Cover — revisited

As long as G contains a vertex v of degree at least 3 we can branch as follows:

- take v into the vertex cover and decrease k by 1, or
- take the neighbors of v into the vertex cover and decrease k by at least 3.

If G has maximum degree at most 2 we can solve vertex cover in polynomial-time.

MINIMUM VERTEX COVER — revisited

As long as G contains a vertex v of degree at least 3 we can branch as follows:

- take v into the vertex cover and decrease k by 1, or
- take the neighbors of v into the vertex cover and decrease k by at least 3.

If G has maximum degree at most 2 we can solve vertex cover in polynomial-time.

What is the running time of this algorithm?

MINIMUM VERTEX COVER — revisited

Recall

The running time of a search tree algorithm is proportional to the number of nodes of the search tree (omitting polynomial factors).

What is the number of nodes of this search tree?

The number of search tree nodes is given by the following recurrence function $T(k) = T(k-1) + T(k-3)$.

MINIMUM VERTEX COVER — revisited

Question

How can we find $T(k)$ given that $T(k) = T(k-1) + T(k-3)$?

Observation

$T(k) = c^k$ for some constant c.

Hence, we have to find c such that:

$$c^k = c^{k-1} + c^{k-3}$$

, or equivalently

$$c^3 - c^2 + 1 = 0$$

MINIMUM VERTEX COVER — revisited

We need to find the positive roots of the polynomial:

$$c^3 - c^2 + 1 = 0$$

Remark

Every such polynomial has a unique positive root which can unfortunately only be found using numerical methods. In practice one can use an algebra package such as R, mathematica, or wolfram alpha.

In this case we obtain $c = 1.47$ and hence our algorithm runs in time $O^*(1.47^k)$.

Outline

Branching Vectors

In the previous example we had 2 branches:

- 1 branch decreasing the parameter by 1 and
- 1 branch decreasing the parameter by 3.

This gives rise to the **branching vector** $(1, 3)$.

In general, the branching vector contains the number by which the parameter is decreased for each of the available branches. The branching vector is often given in ascending order.

The size of the search tree can be directly computed from its branching vector (using a numerical tool).

Branching Vector

Example

Consider the branching vector $(2, 5, 6, 6, 7, 7)$.

The value $c > 0$ has to satisfy:

$$c^k = 2c^{k-7} + 2c^{k-6} + c^{k-5} + c^{k-2}$$

or equivalently:

$$c^7 - 2 - 2c - c^2 - c^5 = 0$$

The unique positive root is 1.4483 and hence the size of the search tree is at most 1.4483^k.

It is hard to compare branching vectors intuitively!

Outline

MINIMUM VERTEX COVER — revisited II

We have seen that MINIMUM VERTEX COVER can be solved in time $O^*(1.47^k)$. Can we do better?

Yes, by employing different branching rules depending on the minimum degree of the remaining graph G (let $\delta(G)$ denoted the minimum degree of G):

G has a vertex v with $N(v) = \{u\}$ ($\delta(G) = 1$)

We show that G has a k-VC iff $G \setminus N[u]$ has a $(k-1)$-VC.

Proof: It suffices to show that G has a minimum VC that contains u. Let S be a minimum VC that does not contain u. Hence, $v \in S$ and the set $S' = S \setminus \{v\} \cup \{u\}$ is a minimum VC that contains u. $\square$

Minimum Vertex Cover — revisited II

We have seen that Minimum Vertex Cover can be solved in time $O^*(1.47^k)$. Can we do better?

Yes, by employing different branching rules depending on the minimum degree of the remaining graph G (let $\delta(G)$ denoted the minimum degree of G):

G has a vertex v with $N(v) = \{u\}$ ($\delta(G) = 1$)

We show that G has a k-VC iff $G \setminus N[u]$ has a $(k-1)$-VC.

Proof: It suffices to show that G has a minimum VC that contains u. Let S be a minimum VC that does not contain u. Hence, $v \in S$ and the set $S' = S \setminus \{v\} \cup \{u\}$ is a minimum VC that contains u. □

No branching necessary!

Minimum Vertex Cover — revisited II

G has a vertex v with $N(v) = \{a, b\}$ ($\delta(G) = 2$)

We distinguish the following cases:

Case 1) $\{a, b\} \in E(G)$: We show that G has a k-VC iff $G \setminus (\{v, a, b\})$ has a $(k-2)$-VC.

Proof: It suffices to show that G has a minimum VC S that contains a and b. Clearly S contains either a or b. W.l.o.g. we can assume it does not contain a. But then $v \in S$ and the set $S' = S \setminus \{v\} \cup \{b\}$ is also a minimum VC that contains a and b. $\square$

Minimum Vertex Cover — revisited II

G has a vertex v with $N(v) = \{a, b\}$ ($\delta(G) = 2$)

We distinguish the following cases:

Case 1) $\{a, b\} \in E(G)$: We show that G has a k-VC iff $G \setminus (\{v, a, b\})$ has a $(k-2)$-VC.

Proof: It suffices to show that G has a minimum VC S that contains a and b. Clearly S contains either a or b. W.l.o.g. we can assume it does not contain a. But then $v \in S$ and the set $S' = S \setminus \{v\} \cup \{b\}$ is also a minimum VC that contains a and b. $\square$

No branching necessary!

Minimum Vertex Cover — revisited II

G has a vertex v with $N(v) = \{a, b\}$ ($\delta(G) = 2$)

Case 2) $|N(a) \cup N(b)| < 3$: Hence, $N(a) = N(b) = \{v, u\}$ we show that G has a k-VC iff $G \setminus (N(v) \cup \{u\})$ has a $(k-2)$-VC.

Proof: It suffices to show that G has a minimum VC S that contains v and u. Suppose not then S contain a and b and consequently $S' = S \setminus \{a, b\} \cup \{v, u\}$ is a minimum VC that contains v and u. $\square$

Minimum Vertex Cover — revisited II

G has a vertex v with $N(v) = \{a, b\}$ ($\delta(G) = 2$)

Case 2) $|N(a) \cup N(b)| < 3$: Hence, $N(a) = N(b) = \{v, u\}$ we show that G has a k-VC iff $G \setminus (N(v) \cup \{u\})$ has a $(k-2)$-VC.

Proof: It suffices to show that G has a minimum VC S that contains v and u. Suppose not then S contain a and b and consequently $S' = S \setminus \{a, b\} \cup \{v, u\}$ is a minimum VC that contains v and u. $\square$

No branching necessary!

Minimum Vertex Cover — revisited II

G has a vertex v with $N(v) = \{a, b\}$ ($\delta(G) = 2$)

Case 3) $|N(a) \cup N(b)| = d \geq 3$: Then G has a k-VC iff $G \setminus N[v]$ has a $(k-2)$-VC or $G \setminus (N[a] \cup N[b])$ has $(k-d)$-VC with $d = |N(a) \cup N(b)| \geq 3$.

Proof: It suffices to show that G has a minimum VC S that contains either a and b or all neighbors of a and b. Clearly, if S contains neither a nor b then S has to contain all neighbors of a and b. Hence, we can assume that w.l.o.g. S contains a but not b. It follows that S contains all neighbors of b. In particular $v \in S$. But then $S' = (S \setminus \{v\}) \cup \{b\}$ is a minimum VC that contains a and b.
Branching vector is $(2, 3)$**. Branching number less than** 1.33**.**

Minimum Vertex Cover — revisited II

G has a vertex v with $N(v) = \{a, b, c\}$ ($\delta(G) = 3$)

We distinguish the following cases:

Case 1) $\{a, b\} \in E(G)$ (Triangle): Then G has a k-VC iff $G \setminus N[v]$ has a $(k-3)$-VC or $G \setminus N[c]$ has a $(k-d)$-VC where $d = |N(c)|$.

Proof: The reverse direction is trivial.

MINIMUM VERTEX COVER — revisited II

G has a vertex v with $N(v) = \{a, b, c\}$ ($\delta(G) = 3$)

Case 1) $\{a, b\} \in E(G)$ (Triangle): Then G has a k-VC iff $G \setminus N[v]$ has a $(k-3)$-VC or $G \setminus N[c]$ has a $(k-d)$-VC where $d = |N(c)|$.

Proof, continued: For the forward direction it suffices to show that G has a minimum VC that contains $N(v)$ or $N(c)$.

Let S be a minimum VC of G. If $v \notin S$ then $N(v) \subseteq S$. So we may assume that $v \in S$.
Because $\{a, b\} \in E(G)$ either $a \in S$ or $b \in S$. W.l.o.g. we can assume that $a \in S$. If in addition $c \in S$ then $S' = S \setminus \{v\} \cup \{b\}$ is also a minimum VC with $N(v) \subseteq S'$. If $c \notin S'$ then $N(c) \subseteq S'$. This shows that G has a minimum VC that contains either $N(v)$ or $N(c)$. $\square$

MINIMUM VERTEX COVER — revisited II

G has a vertex v with $N(v) = \{a, b, c\}$ ($\delta(G) = 3$)

Case 1) $\{a, b\} \in E(G)$ (Triangle):

Branching vector is $(3, 3)$. Branching Number less than 1.27.

Minimum Vertex Cover — revisited II

G has a vertex v with $N(v) = \{a, b, c\}$ ($\delta(G) = 3$)

Case 2) $\{v, d\} \subseteq N(a) \cap N(b)$ (4-cycle): Then G has a k-VC iff $G \setminus N[v]$ has a $(k-3)$-VC or $G \setminus \{v, d\}$ has a $(k-2)$-VC.

Proof (sketch): A minimum VC that contains v but not d must contain a and b. But then $S' = S \setminus \{v\} \cup \{x\}$ is also a minimum VC. Therefore G has a minimum VC that contains $N(v)$ or $\{v, d\}$. $\square$

MINIMUM VERTEX COVER — revisited II

G has a vertex v with $N(v) = \{a, b, c\}$ ($\delta(G) = 3$)

Case 2) $\{v, d\} \subseteq N(a) \cap N(b)$ (4-cycle): Then G has a k-VC iff $G \setminus N[v]$ has a $(k-3)$-VC or $G \setminus \{v, d\}$ has a $(k-2)$-VC.

Proof (sketch): A minimum VC that contains v but not d must contain a and b. But then $S' = S \setminus \{v\} \cup \{x\}$ is also a minimum VC. Therefore G has a minimum VC that contains $N(v)$ or $\{v, d\}$. $\square$

Branching vector is $(2, 3)$. Branching number less than 1.33.

Minimum Vertex Cover — revisited II

G has a vertex v with $N(v) = \{a, b, c\}$ ($\delta(G) = 3$)

Case 3) otherwise, i.e., there are no edges between a, b, and c and $\{a, b, c\}$ pairwise have only v as a common neighbor (Tree-neighborhood):

Then G has a k-VC iff either:

(1) $G \setminus N[v]$ has a $(k-3)$-VC, or

(2) $G \setminus N[c]$ has a $(k-3)$-VC, or

(3) $G \setminus N[a] \setminus N[b] \setminus \{c\}$ has a $(k-y)$-VC with $y = |N(a) \cup N(b)| + 1 \geq 6$.

Branching vector is $(3, 3, 6)$**. Branching number less than** 1.35**.**

Minimum Vertex Cover — revisited II

G has a vertex v with $N(v) = \{a, b, c\}$ ($\delta(G) = 3$)

Case 3) Tree-neighborhood: Then G has a k-VC iff either: (1) $G \setminus N[v]$ has a $(k-3)$-VC, or (2) $G \setminus N[c]$ has a $(k-3)$-VC, or (3) $G \setminus N[a] \setminus N[b] \setminus \{c\}$ has a $(k-y)$-VC with $y = |N(a) \cup N(b)| + 1 \geq 6$.

Proof: Let S be a minimum VC of G. If $v \notin S$ or $c \notin S$ case (1) or (2) applies, so suppose $v \in S$ and $c \in S$.
If $a \in S$ then $S - \{v\} \cup \{b\}$ is also a minimum VC so case (1) again applies. The case $b \in S$ is similar. So suppose $a, b \notin S$. This shows that $N(a) \cup N(b) \cup \{c\} \subseteq S$ so case (3) applies. □

Minimum Vertex Cover — revisited II

G has a vertex v with $N(v) = \{a, b, c, d\}$ ($\delta(G) = 4$)

In this case G has a k-VC iff either $G \setminus \{v\}$ has a $(k-1)$-VC or $G \setminus N[v]$ has a $(k-4)$-VC.

Branching vector is $(1, 4)$. Branching number less than 1.39.

Now we have a branching rule for every case (If different choices are possible rules with lower branching number should be preferred).

Minimum Vertex Cover — revisited II

Summary of the cases

- degree 1, no branching only reduction;
- degree 2, 1.33;
- degree 3, 1.27, 1.33, 1.35;
- degree ≥ 4, 1.39.

Theorem

The algorithm solves Minimum Vertex Cover in time $O^*(1.39^k)$.

Further Improvements

As one might expect the above search tree algorithm can be improved with more detailed and longer case studies. The current fastest algorithm has running time $O^*(1.28^k)$.

Improvements are obtained by:

- More branching rules: consider larger vertex neighborhoods and distiguish more cases.
- Making smarter choices for the branching vertex (neighborhood), e.g., choose a degree 3 vertex with a high degree neighbor instead of just any degree 3 vertex.
- Analyze subsequent recursive calls: note that the degree 4 rule is a bottleneck, which therefore should only be applied when every vertex has degree 4. But this means that subsequently a degree 1, 2, or 3 rule may be applied.

Tutorial – Topics

- The following is just a small selection of possible topics (papers) for the tutorial.
- Please ask for papers on topics you are interested in (you mind find out which topics these are during the curse of this lecture).
- If you know you are more interested in applications do not hesitate to ask as well!

Tutorial – Topics

- Jianer Chen, Iyad A. Kanj, Ge Xia: **Improved upper bounds for vertex cover.** Theor. Comput. Sci. 411(40-42): 3736-3756 (2010);
- Yixin Cao, Jianer Chen, Yang Liu: **On Feedback Vertex Set New Measure and New Structures.** SWAT 2010: 93-104;
- Jianer Chen, Yang Liu, Songjian Lu, Barry O'Sullivan, Igor Razgon: **A fixed-parameter algorithm for the directed feedback vertex set problem.** J. ACM 55(5): (2008);
- Dániel Marx, Igor Razgon: **Fixed-parameter tractability of multicut parameterized by the size of the cutset.** STOC 2011: 469-478;

Tutorial – Topics

- Jiong Guo, Jens Gramm, Falk Hüffner, Rolf Niedermeier, Sebastian Wernicke: **Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization**. J. Comput. Syst. Sci. 72(8): 1386-1396 (2006);
- Stefan Kratsch, Magnus Wahlström: **Compression via Matroids: A Randomized Polynomial Kernel for Odd Cycle Transversal**. SODA 2012: 94-103;
- Hans L. Bodlaender, Bart M. P. Jansen, Stefan Kratsch: **Kernel Bounds for Path and Cycle Problems**. IPEC 2011: 145-158;
- Hans L. Bodlaender, Bart M. P. Jansen, Stefan Kratsch:**Cross-Composition: A New Technique for Kernelization Lower Bounds**. STACS 2011: 165-176;

Tutorial – Topics

- Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, Dimitrios M. Thilikos: **Bidimensionality and Kernels.** SODA 2010: 503-510;
- Torben Hagerup: **Simpler Linear-Time Kernelization for Planar Dominating Set.** IPEC 2011: 181-193.

I will put a txt-file containing the titles of the papers on IS. If you want I can also upload the papers there!