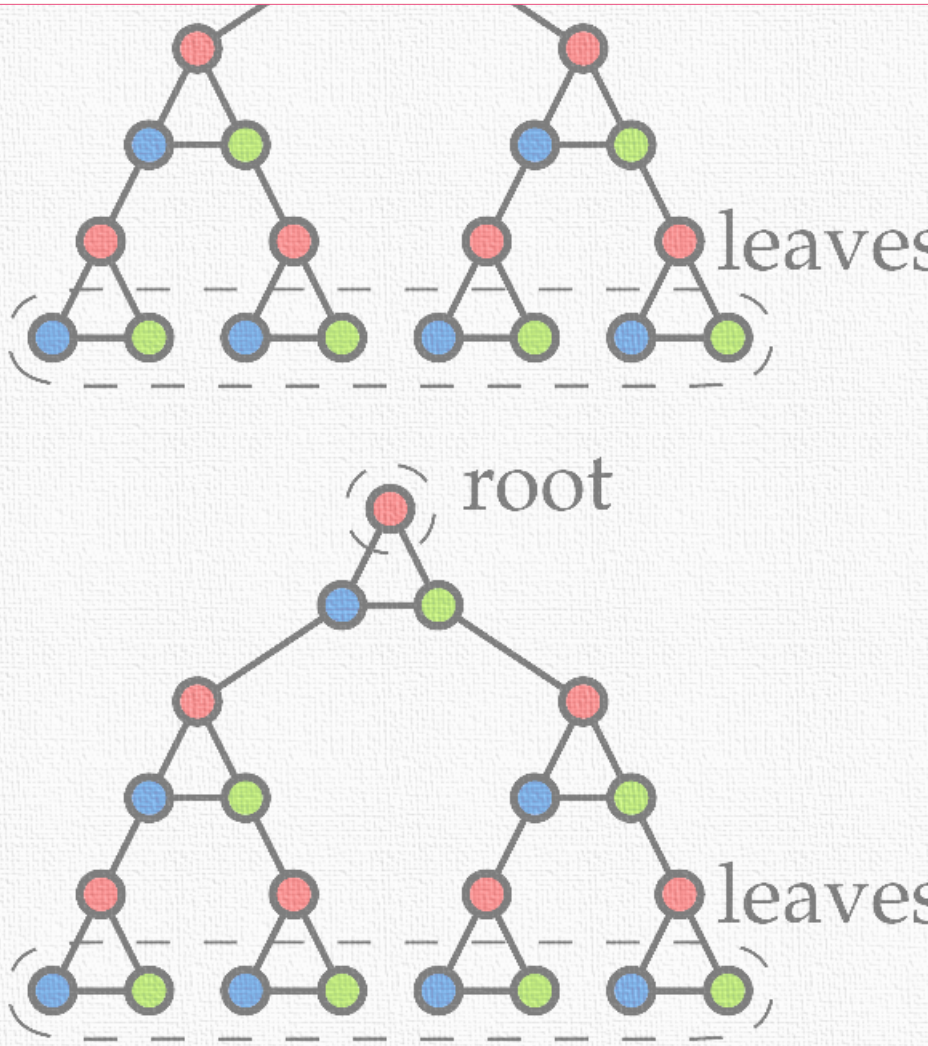
Technische Universiteit
Eindhoven
University of Technology

Kernelization

Astrid Pieterse



About me

- PhD student since September 1st, 2015
- Eindhoven
- Supervisor: Bart Jansen
- Promotor: Mark de Berg

Before that

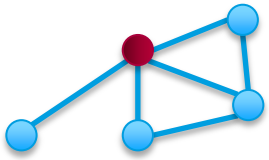
- Mathematics and Computer Science bachelor
- Computer Science and Engineering Master

Algorithms

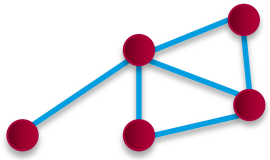
- Find fast algorithms to solve problems
- Some problems are NP-hard
- Solve them anyway?
- Parameterized problem
  - Input (x, k)
  - k is the parameter
  - x is the normal input

What parameter?

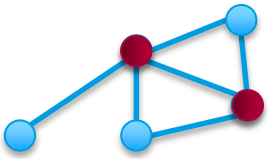
Problem *easy* if k small



(Almost) acyclic?

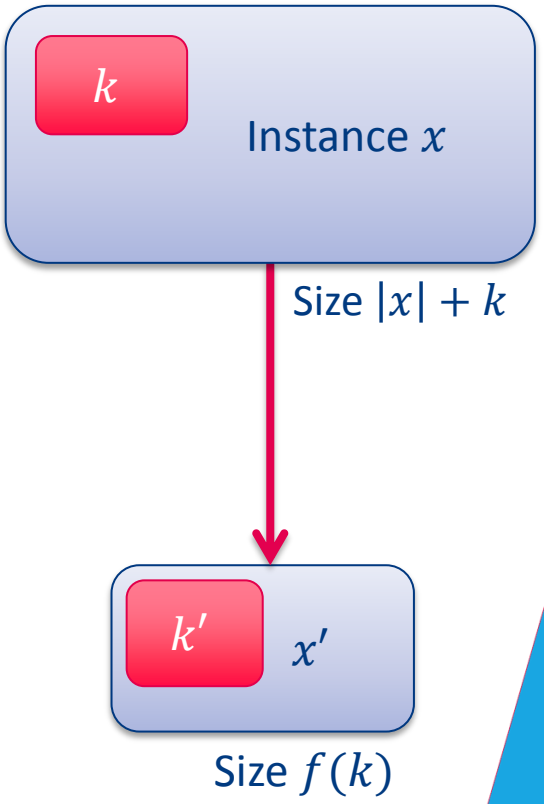


Number of nodes



Solution size

k



Can we do preprocessing?

- Reduce input size
- Polynomial time
- Size of x' depends only on k
  - and is hopefully *small* (polynomial)
- Kernelization

Methods have been found to

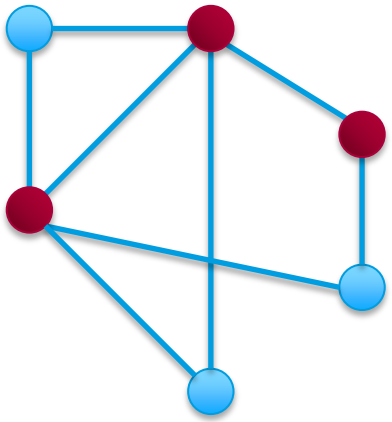
- Find such a preprocessing
- Show it does not exist

Example: Vertex Cover

Subset of vertices such that every edge is covered

Does G have a vertex cover of size k?

Parameter: k



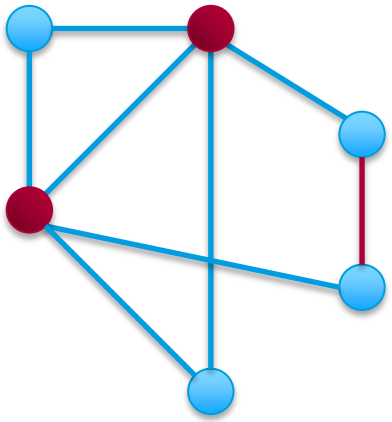
Correct vertex cover

Example: Vertex Cover

Subset of vertices such that every edge is covered

Does G have a vertex cover of size k?

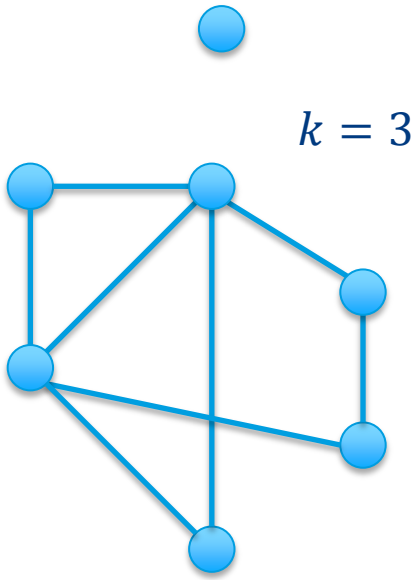
Parameter: k



Incorrect vertex cover

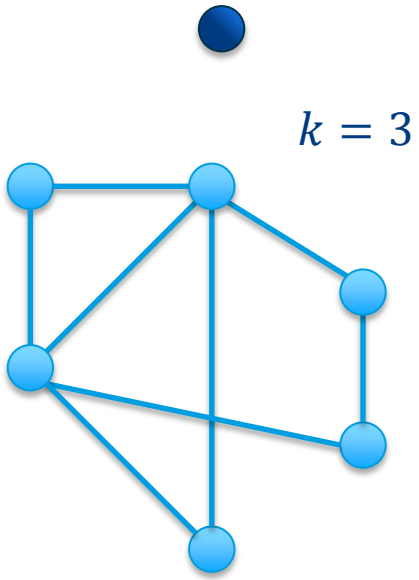
Not covered

Rules to reduce input size



Rules to reduce input size

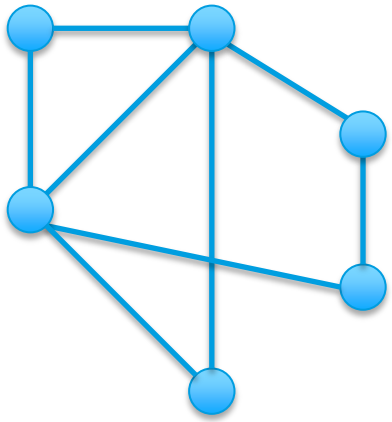
- Remove vertex without edges



Rules to reduce input size

- Remove vertex without edges

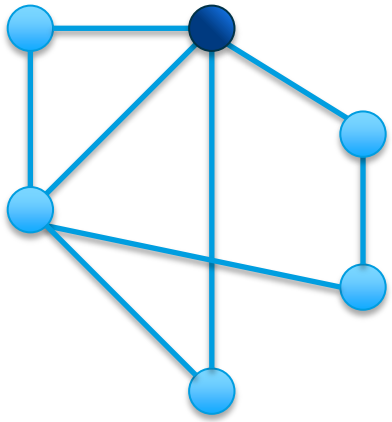
$k = 3$



Rules to reduce input size

- Remove vertex without edges
- Suppose degree > k

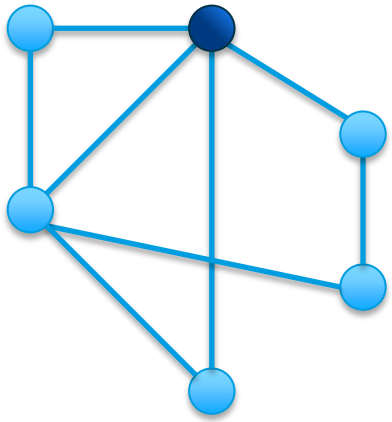
$k = 3$



Rules to reduce input size

- Remove vertex without edges
- Suppose degree > k
  - Add v to cover, decrease k

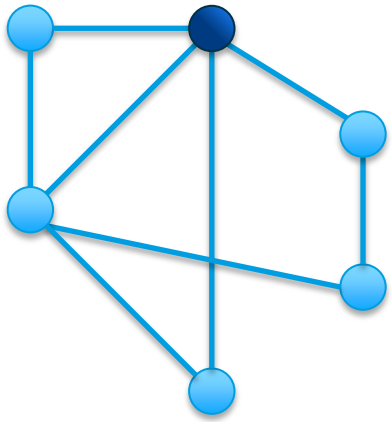
$k = 3$



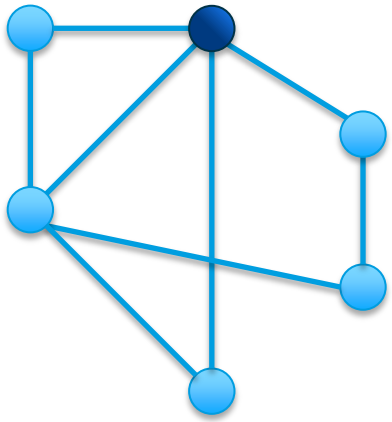
Rules to reduce input size

- Remove vertex without edges
- Suppose degree > k
  - Add v to cover, decrease k

$k = 2$



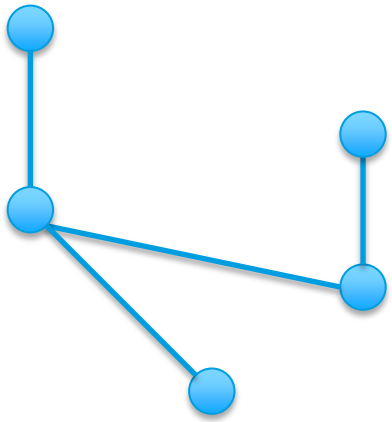
$k = 2$



Rules to reduce input size

- Remove vertex without edges
- Suppose degree > k
  - Add v to cover, decrease k
  - Remove v

$k = 2$



Rules to reduce input size

- Remove vertex without edges
- Suppose degree > k
  - Add v to cover, decrease k
  - Remove v

$k = 1$

Rules to reduce input size

- Remove vertex without edges
- Suppose degree > k
  - Add v to cover, decrease k
  - Remove v
- Apply these rules exhaustively
  - Every vertex has degree $\leq k$
  - By k vertices we cover $\leq k^2$ edges
- 2 Options
  - Many edges: output no
  - Else: instance of size $O(k^2)$!

In this case, a combination of smart rules can provably shrink the instance size!

- But in other cases we can prove this is impossible

Master thesis

- Considered several graph problems
- and one from logic

Current work

- Constraint satisfaction problems
  - Number of *constraints* over variables

    $$(x \vee y \vee z) \wedge (\neg x \vee \neg y) \wedge \cdots$$

  - Can express many other problems
    - Including graph problems
  - But often NP-hard
  - When is *kernelization* possible?
    - Parameter: number of variables