Advanced Techniques for Combinatorial Algorithms: Fixed-Parameter Algorithms

Gianluca Della Vedova

Univ. Milano–Bicocca
https://gianluca.dellavedova.org

May 7, 2020

Fixed-Parameter

- An **NP**-hard problem does not go away
- Vertex cover
- Clique
- Independent set
- Dominating set
- Hamiltonian cycle

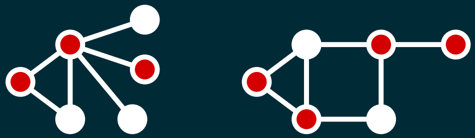
Vertex Cover

Instance

Undirected graph $G = \langle V, E \rangle$, integer k.

Question

Find a set $C \subset V$ such that for each edge $e \in E$ at least one endpoint of e belongs to C, and $|C| \leq k$



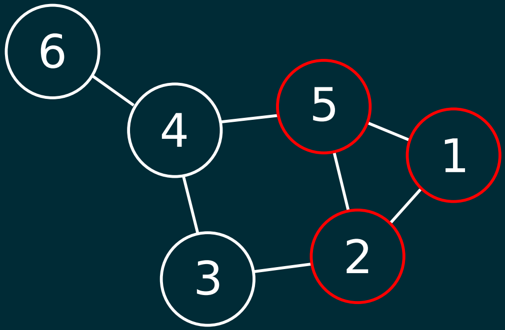
Clique

Instance

Undirected graph $G = \langle V, E \rangle$, integer k.

Question

Find a set $C \subset V$ such that all pairs of vertices in C are connected by an edge, and $|C| \geq k$



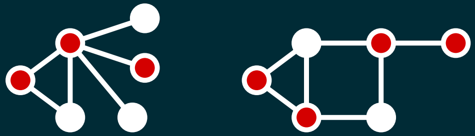
Independent Set

Instance

Undirected graph $G = \langle V, E \rangle$, integer k.

Question

Find a set $I \subset V$ such that no two vertices in K are connected by an edge, and $|K| \geq k$



Dominating Set

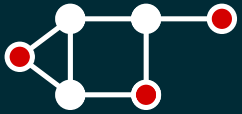
Instance

Undirected graph $G = \langle V, E \rangle$, integer k.

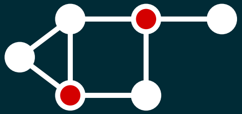
Question

Find a set $D \subset V$ such that for each vertex $v \notin D$, v is adjacent to some $d \in D$, and $|D| \leq k$

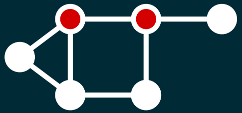
(a)



(b)



(c)



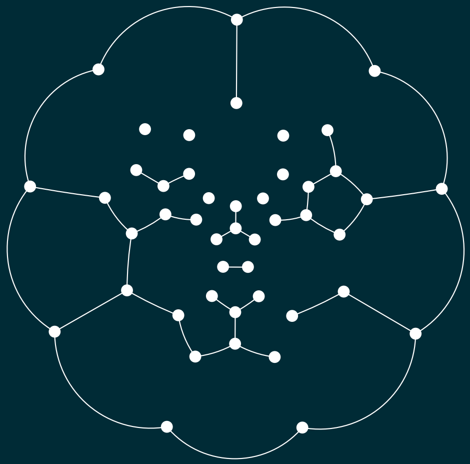
Hamiltonian cycle

Instance

Undirected graph $G = \langle V, E \rangle$.

Question

Find a cycle C that visits each vertex $v \in V$ exactly once.



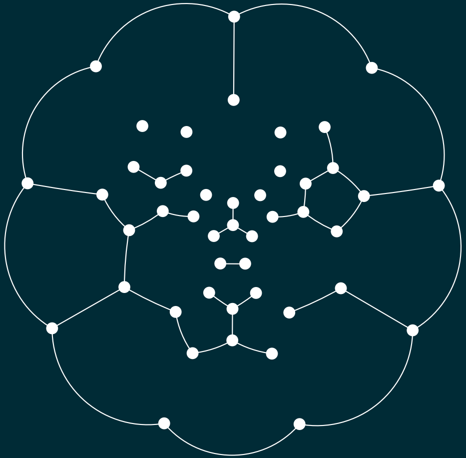
Longest Path

Instance

Undirected graph $G = \langle V, E \rangle$, integer k.

Question

Find a simple (no vertex is visited twice) path P of G, with P consisting of k vertices.



Fixed Parameter Tractable (FPT)

FPT

Parameterized problem: pair (I, k). The problem is in FPT if there is an algorithm with time $f(k)n^\alpha$, with α a constant and f a function

Typical times

- $O(1.1^k)n$
- $O(2^k)n^3$
- $O(2^k + n^3)$
- $O(k^k)n^3$
- $O(2^{2^{2^{\cdots^2}}})n^3$

Bounded Search Tree

The search tree has $O(f(k))n^{\alpha}$ nodes

Vertex Cover

Let $(u, v) \in E$. Then $u \in C$ or $v \in C$

- How is the search tree?
- $2^k n$ time

Smaller Search Tree

Proposition

Let v be a node, let $N(v)$ be its neighbors. Then $v \in C$ or $N(v) \subseteq C$

Proposition

No node has at least three neighbors. Then G consists of vertex-disjoint cycles or paths, and its smallest cover is trivial

Corollary

Consider only vertices v with $|N(v)| \geq 3$

Smaller Search Tree

Two cases

1. $v \in C$, then recurse on $G - v$ and height $k - 1$
2. $v \notin C$, then recurse on $G - N(v)$ and height at most $k - 3$

1. Number of nodes $f(k) = f(k-1) + f(k-3) + 1$
2. $f(k) \leq 5^{\frac{k}{4}} < 1.5^k$

Independent Set on Planar Graphs

Euler formula

Let G be a planar graph. Then G has at most $3n - 6$ edges.

Corollary

Let G be a planar graph. Then there exists a vertex v of G with at most 5 incident edges.

Independent Set on Planar Graphs

Algorithm 1: Finds an independent set of size k, if it exists

$v \leftarrow$ a minimum degree vertex of G **foreach** $x \in \{v\} \cap N(v)$ **do**
- Add x to I;
- $G_1 \leftarrow$ a copy of G where x and the edges incident on it are removed;
- **if** *G_1 is not empty* **then**
  - recurse on $(G_1, k-1, I)$;
- **else**
  - **if** $|I| \geq k$ **then**
    - **return** I;

Independent Set on Planar Graphs

Algorithm 2: Finds an independent set of size k, if it exists

```
v ← a minimum degree vertex of G foreach x ∈ {v} ∩ N(v) do
    Add x to I;
    G_1 ← a copy of G where x and the edges incident on it are removed;
    if G_1 is not empty then
        recurse on (G_1, k − 1, I);
    else
        if |I| ≥ k then
            return I;
```

Time $O(6^k)n$

Reduction to a Kernel

Main idea

1. Fix or exclude portions of the solution, so that the remaining part is small.
2. Then be naive on what remains.

Rule example

1. Let v be a vertex with exactly one neighbor w.
2. Remove v
3. Put w in the cover.

Problem

No such vertex $v \Rightarrow$ nothing is removed

Reduction to a Kernel

1. Find all vertices L with degree $\geq k$. If $|L| > k$, then no cover of size k. Let $k_1 = k - |L|$
2. $G_1 = G - L$. If G_1 has more than $k_1(k+1)$ vertices, then no cover of size k.
3. Find k_1-cover of G_1, time $O(n + k^{2k})$
4. Improve to $O(n + 2^k k^2)$

Independent Set on Planar Graphs

4 color theorem

Let G be a planar graph. Then there is a coloring of the vertices of G that uses at most 4 colors and such that no two adjacent vertices are the same color.

Corollaries

1. The set of vertices that are the same color is an independent set
2. The largest such set has at least $1/4$ of all vertices
3. The set of all vertices is a kernel with $4k$ elements

Question

Is this kernel useful?

Independent Set on Planar Graphs

4 color theorem

Let G be a planar graph. Then there is a coloring of the vertices of G that uses at most 4 colors and such that no two adjacent vertices are the same color.

Corollaries

1. The set of vertices that are the same color is an independent set
2. The largest such set has at least $1/4$ of all vertices
3. The set of all vertices is a kernel with $4k$ elements

Question

Is this kernel useful? No, $4k \geq n \Rightarrow 2^{[4k]} \geq 2^n$

Closest string

Input

$s_1, \ldots, s_m$: strings of length n. Integer k

Problem

Find a string $t = t[1] \cdots t[n]$ such that t has Hamming distance at most k with each s_i

Hint

Let s_i and s_j be two input strings. Find an upper bound on their distance.

Closest string

Input

$s_1, \ldots, s_m$: strings of length n. Integer k

Problem

Find a string $t = t[1] \cdots t[n]$ such that t has Hamming distance at most k with each s_i

Hint

Let s_i and s_j be two input strings. Find an upper bound on their distance. $2k$

Color Coding

Colorful Path

1. Color each vertex with k colors
2. Colorful path: each vertex has a distinct color

Random coloring

Probability that a path is colorful: $\frac{k!}{k^k} \geq e^{-k}$

Dynamic programming

- Given a coloring, find a colorful path (if it exists)
- Keep only the colors
- Time $(2^{O(k)}|E|)$

Color Coding

Dynamic programming

i colors from v to w: $i-1$ colors from v to z and the color from z to w

Time complexity

- At time i, there are $\binom{k}{i}$ sets
- $O(\sum_{i=1}^{k} i\binom{k}{i}) = O(k2^k)$

Derandomize

- k-perfect family H of hash functions $h : [1:n] \mapsto [1:k]$ is such that for each $S \subseteq [1:n]$ with $|S| = k$, there exists h that is 1-to-1 on S
- Compute in linear time a k-perfect family H of $2^{O(k)} \log n$ functions

Super-Sub sequence

Longest Subsequence

$s_1, \ldots, s_m$: strings of length n. Does it exist $t = t[1] \cdots t[k]$ such that for each s_i, $s_i = w_0 t[1] w_1 \cdots t[k] w_k$, for some (possibly empty) strings w_j?

Shortest Supersequence

$s_1, \ldots, s_m$: strings of length n. Does it exist $t = t[1] \cdots t[k]$ such that for each s_i, $t_i = w_0 s_i[1] w_1 \cdots s_i[n] w_k$, for some (possibly empty) strings w_j?

- Which problem is easier?

Attribution

- Vertex Cover figure: By Miym - Own work, CC BY-SA 3.0, `https://commons.wikimedia.org/w/index.php?curid=6017739`
- Clique figure: Public Domain, `https://commons.wikimedia.org/w/index.php?curid=1072101`
- Dominating Set figure: By Miym - Own work, CC BY-SA 3.0, `https://commons.wikimedia.org/w/index.php?curid=6821042`