Real FPT Algorithms For The Vertex Cover Problem

Exposition by William Gasarch

February 24, 2026

The Vertex Cover Problem

Def If G is a graph then a **vertex cover** is a set of vertices such that every edge has at least one endpoint in that set.

The Vertex Cover Problem

Def If G is a graph then a **vertex cover** is a set of vertices such that every edge has at least one endpoint in that set.

Def $\mathrm{VC} = \{(G, k)\colon G$ has a vertex cover of size k $\}$.

The Vertex Cover Problem

Def If G is a graph then a **vertex cover** is a set of vertices such that every edge has at least one endpoint in that set.

Def $\mathrm{VC} = \{(G, k)\colon G$ has a vertex cover of size k $\}$.
VC is known to be NPC.

The Vertex Cover Problem

Def If G is a graph then a **vertex cover** is a set of vertices such that every edge has at least one endpoint in that set.

Def $\mathrm{VC} = \{(G, k)\colon G$ has a vertex cover of size k $\}$.
VC is known to be NPC.

Def $\mathrm{VC}_k = \{G\colon G$ has a vertex cover of size k $\}$.

The Vertex Cover Problem

Def If G is a graph then a **vertex cover** is a set of vertices such that every edge has at least one endpoint in that set.

Def $\mathrm{VC} = \{(G,k) \colon G$ has a vertex cover of size k $\}$.
VC is known to be NPC.

Def $\mathrm{VC}_k = \{G \colon G$ has a vertex cover of size k $\}$.

VC_k is clearly in $O(n^k)$ time.

The Vertex Cover Problem

Def If G is a graph then a **vertex cover** is a set of vertices such that every edge has at least one endpoint in that set.

Def $\mathrm{VC} = \{(G, k) \colon G$ has a vertex cover of size k $\}$.
VC is known to be NPC.

Def $\mathrm{VC}_k = \{G \colon G$ has a vertex cover of size k $\}$.

VC_k is clearly in $O(n^k)$ time.

We showed how to do it in $O(f(k)n^3)$ time using GMT but the algorithm was nonconstructive.

The Vertex Cover Problem

Def If G is a graph then a **vertex cover** is a set of vertices such that every edge has at least one endpoint in that set.

Def $\mathrm{VC} = \{(G,k)\colon G$ has a vertex cover of size k $\}$.
VC is known to be NPC.

Def $\mathrm{VC}_k = \{G\colon G$ has a vertex cover of size k $\}$.

VC_k is clearly in $O(n^k)$ time.

We showed how to do it in $O(f(k)n^3)$ time using GMT but the algorithm was nonconstructive.

We showed how to do it in $O(f(k)n^3)$ time using GMT but with a constructive algorithm.

The Vertex Cover Problem

Def If G is a graph then a **vertex cover** is a set of vertices such that every edge has at least one endpoint in that set.

Def $\mathrm{VC} = \{(G, k) \colon G$ has a vertex cover of size k $\}$.
VC is known to be NPC.

Def $\mathrm{VC}_k = \{G \colon G$ has a vertex cover of size k $\}$.

VC_k is clearly in $O(n^k)$ time.

We showed how to do it in $O(f(k)n^3)$ time using GMT but the algorithm was nonconstructive.

We showed how to do it in $O(f(k)n^3)$ time using GMT but with a constructive algorithm.

Both of the above algorithms are **insane.**

The Vertex Cover Problem

Def If G is a graph then a **vertex cover** is a set of vertices such that every edge has at least one endpoint in that set.

Def $\mathrm{VC} = \{(G, k) \colon G$ has a vertex cover of size k $\}$.
VC is known to be NPC.

Def $\mathrm{VC}_k = \{G \colon G$ has a vertex cover of size k $\}$.

VC_k is clearly in $O(n^k)$ time.

We showed how to do it in $O(f(k)n^3)$ time using GMT but the algorithm was nonconstructive.

We showed how to do it in $O(f(k)n^3)$ time using GMT but with a constructive algorithm.

Both of the above algorithms are **insane.**

We present some **sane** algorithms.

An Old Quote (not mine) Refuted

An Old Quote (not mine) Refuted

There is a thin line that separates **sane** *from* **insane**.

An Old Quote (not mine) Refuted

There is a thin line that separates **sane** *from* **insane***.*

Counterexample

An Old Quote (not mine) Refuted

There is a thin line that separates **sane** *from* **insane***.*

Counterexample

1) The OBS algs I showed you **totally insane**

An Old Quote (not mine) Refuted

There is a thin line that separates **sane** *from* **insane**.

Counterexample

1) The OBS algs I showed you **totally insane**
2) Today's algs are **extremely sane**

An Old Quote (not mine) Refuted

There is a thin line that separates **sane** *from* **insane***.*

Counterexample
1) The OBS algs I showed you **totally insane**
2) Today's algs are **extremely sane**

So for us

An Old Quote (not mine) Refuted

There is a thin line that separates **sane** *from* **insane**.

Counterexample

1) The OBS algs I showed you **totally insane**
2) Today's algs are **extremely sane**

So for us
There is a fat line that separates **sane** *from* **insane**.

An Old Quote (not mine) Refuted

There is a thin line that separates **sane** *from* **insane***.*

Counterexample

1) The OBS algs I showed you **totally insane**
2) Today's algs are **extremely sane**

So for us
There is a fat line that separates **sane** *from* **insane***.*
That quote I will claim for myself.

An FPT Alg for VC_k In Time $O(2^k n)$

$\mathrm{ALG1}(G, k)$ (We later use $\mathrm{ALG1}$ as a building block.)

An FPT Alg for VC_k In Time $O(2^k n)$

$\mathrm{ALG1}(G, k)$ (We later use $\mathrm{ALG1}$ as a building block.)
1) Input G, k

An FPT Alg for VC_k In Time $O(2^k n)$

$\mathrm{ALG1}(G, k)$ (We later use $\mathrm{ALG1}$ as a building block.)
1) Input G, k
2) If $k = 0$ and G has at least one edge, output NO.

An FPT Alg for VC_k In Time $O(2^k n)$

$\mathrm{ALG1}(G, k)$ (We later use $\mathrm{ALG1}$ as a building block.)
1) Input G, k
2) If $k = 0$ and G has at least one edge, output NO.
3) Pick edge (u, v).

An FPT Alg for VC_k In Time $O(2^k n)$

$\mathrm{ALG1}(G, k)$ (We later use $\mathrm{ALG1}$ as a building block.)
1) Input G, k
2) If $k = 0$ and G has at least one edge, output NO.
3) Pick edge (u, v).
4) If $\mathrm{ALG1}(G - u, k - 1) = \mathrm{YES}$ then output YES.

An FPT Alg for VC_k In Time $O(2^k n)$

$\mathrm{ALG1}(G, k)$ (We later use $\mathrm{ALG1}$ as a building block.)
1) Input G, k
2) If $k = 0$ and G has at least one edge, output NO.
3) Pick edge (u, v).
4) If $\mathrm{ALG1}(G - u, k - 1) = \mathrm{YES}$ then output YES.
5) If $\mathrm{ALG1}(G - v, k - 1) = \mathrm{YES}$ then output YES.

An FPT Alg for VC_k In Time $O(2^k n)$

$\mathrm{ALG1}(G, k)$ (We later use $\mathrm{ALG1}$ as a building block.)
1) Input G, k
2) If $k = 0$ and G has at least one edge, output NO.
3) Pick edge (u, v).
4) If $\mathrm{ALG1}(G - u, k - 1) = \mathrm{YES}$ then output YES.
5) If $\mathrm{ALG1}(G - v, k - 1) = \mathrm{YES}$ then output YES.
6) Output(NO)

An FPT Alg for VC_k In Time $O(2^k n)$

$\mathrm{ALG1}(G, k)$ (We later use $\mathrm{ALG1}$ as a building block.)
1) Input G, k
2) If $k = 0$ and G has at least one edge, output NO.
3) Pick edge (u, v).
4) If $\mathrm{ALG1}(G - u, k - 1) = \mathrm{YES}$ then output YES.
5) If $\mathrm{ALG1}(G - v, k - 1) = \mathrm{YES}$ then output YES.
6) Output(NO)

Correctness By Induction on k.

An FPT Alg for VC_k In Time $O(2^k n)$

$\mathrm{ALG1}(G, k)$ (We later use ALG1 as a building block.)
1) Input G, k
2) If $k = 0$ and G has at least one edge, output NO.
3) Pick edge (u, v).
4) If $\mathrm{ALG1}(G - u, k - 1) = \mathrm{YES}$ then output YES.
5) If $\mathrm{ALG1}(G - v, k - 1) = \mathrm{YES}$ then output YES.
6) Output(NO)

Correctness By Induction on k.

Speed With the proper data structures for graphs each iteration takes $O(n)$ time. Hence the algorithm is $O(2^k n)$.

An FPT Alg for VC_k In Time $O(2^k n)$

$\mathrm{ALG1}(G, k)$ (We later use $\mathrm{ALG1}$ as a building block.)
1) Input G, k
2) If $k = 0$ and G has at least one edge, output NO.
3) Pick edge (u, v).
4) If $\mathrm{ALG1}(G - u, k - 1) = \mathrm{YES}$ then output YES.
5) If $\mathrm{ALG1}(G - v, k - 1) = \mathrm{YES}$ then output YES.
6) Output(NO)

Correctness By Induction on k.

Speed With the proper data structures for graphs each iteration takes $O(n)$ time. Hence the algorithm is $O(2^k n)$.

Easy to code and the constant for the big-O is small.

An FPT Alg for VC_k In Time $O(2^k n)$

$\mathrm{ALG1}(G, k)$ (We later use $\mathrm{ALG1}$ as a building block.)
1) Input G, k
2) If $k = 0$ and G has at least one edge, output NO.
3) Pick edge (u, v).
4) If $\mathrm{ALG1}(G - u, k - 1) = \mathrm{YES}$ then output YES.
5) If $\mathrm{ALG1}(G - v, k - 1) = \mathrm{YES}$ then output YES.
6) Output(NO)

Correctness By Induction on k.

Speed With the proper data structures for graphs each iteration takes $O(n)$ time. Hence the algorithm is $O(2^k n)$.

Easy to code and the constant for the big-O is small.

I think this algorithm is folklore.

Lemma Needed For Better FPT Alg

Lemma 1 Let $k, k' \in \mathbb{N}$ such that $k' \leq k$.

Lemma Needed For Better FPT Alg

Lemma 1 Let $k, k' \in \mathbb{N}$ such that $k' \leq k$.
Let $G' = (V', E')$ be a graph.

Lemma Needed For Better FPT Alg

Lemma 1 Let $k, k' \in \mathbb{N}$ such that $k' \leq k$.
Let $G' = (V', E')$ be a graph. Assume every vertex has degree $\leq k$.

Lemma Needed For Better FPT Alg

Lemma 1 Let $k, k' \in \mathbb{N}$ such that $k' \leq k$.
Let $G' = (V', E')$ be a graph. Assume every vertex has degree $\leq k$.
If there is a VC of size $\leq k'$ then

Lemma Needed For Better FPT Alg

Lemma 1 Let $k, k' \in \mathbb{N}$ such that $k' \leq k$.
Let $G' = (V', E')$ be a graph. Assume every vertex has degree $\leq k$.
If there is a VC of size $\leq k'$ then
$|E'| \leq kk' \leq k^2$.

Lemma Needed For Better FPT Alg

Lemma 1 Let $k, k' \in \mathbb{N}$ such that $k' \leq k$.
Let $G' = (V', E')$ be a graph. Assume every vertex has degree $\leq k$.
If there is a VC of size $\leq k'$ then
$|E'| \leq kk' \leq k^2$.
$|V'| \leq 2k^2$.

Lemma Needed For Better FPT Alg

Lemma 1 Let $k, k' \in \mathbb{N}$ such that $k' \leq k$.
Let $G' = (V', E')$ be a graph. Assume every vertex has degree $\leq k$.
If there is a VC of size $\leq k'$ then
$|E'| \leq kk' \leq k^2$.
$|V'| \leq 2k^2$.
Proof Every vertex in U covers $\leq k$ edges. Hence

Lemma Needed For Better FPT Alg

Lemma 1 Let $k, k' \in \mathbb{N}$ such that $k' \leq k$.
Let $G' = (V', E')$ be a graph. Assume every vertex has degree $\leq k$.
If there is a VC of size $\leq k'$ then
$|E'| \leq kk' \leq k^2$.
$|V'| \leq 2k^2$.
Proof Every vertex in U covers $\leq k$ edges. Hence
$|E'| \leq |U| \times k = k'k \leq k^2$.

Lemma Needed For Better FPT Alg

Lemma 1 Let $k, k' \in \mathbb{N}$ such that $k' \le k$.
Let $G' = (V', E')$ be a graph. Assume every vertex has degree $\le k$.
If there is a VC of size $\le k'$ then
$|E'| \le kk' \le k^2$.
$|V'| \le 2k^2$.
Proof Every vertex in U covers $\le k$ edges. Hence
$|E'| \le |U| \times k = k'k \le k^2$.
$|V'| \le \sum_{v \in V'} \deg(v) \le 2|E'| \le 2k^2$.

Alg Used for Within Better FPT Alg

Lemma Let $k, k' \in \mathbb{N}$ such that $k' \leq k$.

Alg Used for Within Better FPT Alg

Lemma Let $k, k' \in \mathbb{N}$ such that $k' \leq k$.
There is an algorithm that, on input graphs G' such that every vertex has degree $\leq k$, finds if there is a VC of size $\leq k'$ in time $O(k^2 2^k)$.

Alg Used for Within Better FPT Alg

Lemma Let $k, k' \in \mathbb{N}$ such that $k' \leq k$.
There is an algorithm that, on input graphs G' such that every vertex has degree $\leq k$, finds if there is a VC of size $\leq k'$ in time $O(k^2 2^k)$.

$\mathrm{ALG2}$

Alg Used for Within Better FPT Alg

Lemma Let $k, k' \in \mathbb{N}$ such that $k' \leq k$.
There is an algorithm that, on input graphs G' such that every vertex has degree $\leq k$, finds if there is a VC of size $\leq k'$ in time $O(k^2 2^k)$.

ALG2

1) Input $G' = (V', E')$ and k. Time: $O(1)$.

Alg Used for Within Better FPT Alg

Lemma Let $k, k' \in \mathbb{N}$ such that $k' \le k$.
There is an algorithm that, on input graphs G' such that every vertex has degree $\le k$, finds if there is a VC of size $\le k'$ in time $O(k^2 2^k)$.

ALG2

1) Input $G' = (V', E')$ and k. Time:$O(1)$.
2) Compute $|V'|$. If $|V'| > 2k^2$ then output NO and stop. Time $O(k^2)$.

Alg Used for Within Better FPT Alg

Lemma Let $k, k' \in \mathbb{N}$ such that $k' \le k$.
There is an algorithm that, on input graphs G' such that every vertex has degree $\le k$, finds if there is a VC of size $\le k'$ in time $O(k^2 2^k)$.

ALG2

1) Input $G' = (V', E')$ and k. Time: $O(1)$.
2) Compute $|V'|$. If $|V'| > 2k^2$ then output NO and stop. Time $O(k^2)$.
(By Lemma 1, if there is such a VC then G has $\le 2k^2$ vertices.)

Alg Used for Within Better FPT Alg

Lemma Let $k, k' \in \mathbb{N}$ such that $k' \leq k$.
There is an algorithm that, on input graphs G' such that every vertex has degree $\leq k$, finds if there is a VC of size $\leq k'$ in time $O(k^2 2^k)$.

$\mathrm{ALG2}$

1) Input $G' = (V', E')$ and k. Time:$O(1)$.
2) Compute $|V'|$. If $|V'| > 2k^2$ then output NO and stop. Time $O(k^2)$.
(By Lemma 1, if there is such a VC then G has $\leq 2k^2$ vertices.)

2) (G' has $\leq 2k^2$ vertices.) Run $\mathrm{ALG1}(G', k')$ to find the answer.
Time: $2^{k'} \times 2k^2 = O(k^2 2^k)$.

Alg Used for Within Better FPT Alg

Lemma Let $k, k' \in \mathbb{N}$ such that $k' \leq k$.
There is an algorithm that, on input graphs G' such that every vertex has degree $\leq k$, finds if there is a VC of size $\leq k'$ in time $O(k^2 2^k)$.

$\mathrm{ALG2}$

1) Input $G' = (V', E')$ and k. Time:$O(1)$.
2) Compute $|V'|$. If $|V'| > 2k^2$ then output NO and stop. Time $O(k^2)$.
(By Lemma 1, if there is such a VC then G has $\leq 2k^2$ vertices.)

2) (G' has $\leq 2k^2$ vertices.) Run $\mathrm{ALG1}(G', k')$ to find the answer.
Time: $2^{k'} \times 2k^2 = O(k^2 2^k)$.

Total time: $O(k^2) + O(k^2 2^k) = O * 2^2 2^k)$.

An FPT Alg for VC_k In Time $O(kn + k^2 2^k)$

An FPT Alg for VC_k In Time $O(kn + k^2 2^k)$

1) Input (G, k)

An FPT Alg for VC_k In Time $O(kn + k^2 2^k)$

1) Input (G, k)
2) Find C_0, the set of vertices of degree $\geq k$.

An FPT Alg for VC_k In Time $O(kn + k^2 2^k)$

1) Input (G, k)
2) Find C_0, the set of vertices of degree $\geq k$. $O(kn)$ time.

An FPT Alg for VC_k In Time $O(kn + k^2 2^k)$

1) Input (G, k)
2) Find C_0, the set of vertices of degree $\geq k$. $O(kn)$ time.
3) Set $G' = G - C_0$ and $k' = k - |C_0|$.

An FPT Alg for VC_k In Time $O(kn + k^2 2^k)$

1) Input (G, k)
2) Find C_0, the set of vertices of degree $\geq k$. $O(kn)$ time.
3) Set $G' = G - C_0$ and $k' = k - |C_0|$.
4) Now need to solve (G', k'). (G', k') satisfy the premise of $\mathrm{ALG2}$. So run $\mathrm{ALG2}(G', k')$ for the answer. Time: $2k^2 2^k$.

An FPT Alg for VC_k In Time $O(kn + k^2 2^k)$

1) Input (G, k)
2) Find C_0, the set of vertices of degree $\geq k$. $O(kn)$ time.
3) Set $G' = G - C_0$ and $k' = k - |C_0|$.
4) Now need to solve (G', k'). (G', k') satisfy the premise of $\mathrm{ALG2}$. So run $\mathrm{ALG2}(G', k')$ for the answer. Time: $2k^2 2^k$.

Total time: $O(kn + k^2 2^k)$.

An FPT Alg for VC_k In Time $O(kn + k^2 2^k)$

1) Input (G, k)
2) Find C_0, the set of vertices of degree $\geq k$. $O(kn)$ time.
3) Set $G' = G - C_0$ and $k' = k - |C_0|$.
4) Now need to solve (G', k'). (G', k') satisfy the premise of $\mathrm{ALG2}$. So run $\mathrm{ALG2}(G', k')$ for the answer. Time: $2k^2 2^k$.

Total time: $O(kn + k^2 2^k)$.

Note: The constant on the big-O is small. Maybe 3.

An FPT Alg for VC_k In Time $O(kn + k^2 2^k)$

1) Input (G, k)
2) Find C_0, the set of vertices of degree $\geq k$. $O(kn)$ time.
3) Set $G' = G - C_0$ and $k' = k - |C_0|$.
4) Now need to solve (G', k'). (G', k') satisfy the premise of $\mathrm{ALG2}$. So run $\mathrm{ALG2}(G', k')$ for the answer. Time: $2k^2 2^k$.

Total time: $O(kn + k^2 2^k)$.

Note: The constant on the big-O is small. Maybe 3.

Algorithm due to Sam Buss.