Parameterized Intractability

Lawqueen Kanesh
lawqueen@iitj.ac.in

IIT Jodhpur

December 7, 2022

Overview

- Introduction
- Parameterized Reduction
- Independent Set and Clique
- Clique to Clique on Regular Graphs
- Independent Set to Partial Vertex Cover
- Clique to Multicolored Clique
- Multicolored Independent Set to Dominating Set

Parameterized Intractability

- Polynomial Time Algorithms
- NP-hard/complete Problems
- Exact Exponential Algorithms
- Basic techniques to design fixed-parameter tractable (FPT) algorithms

Parameterized Intractability

- Polynomial Time Algorithms
- NP-hard/complete Problems
- Exact Exponential Algorithms
- Basic techniques to design fixed-parameter tractable (FPT) algorithms
- Is it always possible to design an FPT ($f(k)n^{\mathcal{O}(1)}$) algorithm for the give parameterized problem?

Parameterized Intractability

- Polynomial-time reduction from problem P to problem Q:
  - Given an instance I of $P \rightarrow$ construct an instance I' of Q
  - Reduction should take **polynomial** time.
  - I is a yes instance of P iff I' is a yes instance of Q.

Parameterized Intractability

- Polynomial-time reduction from problem P to problem Q:
  - Given an instance I of $P \rightarrow$ construct an instance I' of Q
  - Reduction should take **polynomial** time.
  - I is a yes instance of P iff I' is a yes instance of Q.
- If an NP-hard problem can be solved in polynomial time, then every problem in NP can be solved in polynomial time (i.e., P = NP).

Parameterized Intractability

- Polynomial-time reduction from problem P to problem Q:
  - Given an instance I of $P \rightarrow$ construct an instance I' of Q
  - Reduction should take **polynomial** time.
  - I is a yes instance of P iff I' is a yes instance of Q.
- If an NP-hard problem can be solved in polynomial time, then every problem in NP can be solved in polynomial time (i.e., P = NP).
- Can we use this polynomial time reductions to show that a problem is not FPT?

Parameterized Intractability

- Independent Set and Vertex Cover

Parameterized Intractability

- Independent Set and Vertex Cover
- Graph G has a vertex cover k if and only if it has an independent set of size $n - k$.

Parameterized Intractability

- Independent Set and Vertex Cover
- Graph G has a vertex cover k if and only if it has an independent set of size $n - k$.
- Vertex Cover is FPT, but Independent Set is not known to be FPT.

Parameterized Intractability

To build a complexity theory for parameterized problems, we need two concepts:

- An appropriate notion of reduction.
- An appropriate hypothesis.

Parameterized Reductions

Parameterized Reduction: Let $P, Q \subseteq \Sigma^* \times \mathbb{N}$ be two parameterized problems. A parameterized reduction from P to Q is an algorithm that, given an instance (I, k) of P, outputs an instance(I', k') of Q such that

Parameterized Reductions

Parameterized Reduction: Let $P, Q \subseteq \Sigma^* \times \mathbb{N}$ be two parameterized problems. A parameterized reduction from P to Q is an algorithm that, given an instance (I, k) of P, outputs an instance(I', k') of Q such that

- (I, k) is a yes-instance of P iff (I', k') is a yes-instance of Q

Parameterized Reductions

Parameterized Reduction: Let $P, Q \subseteq \Sigma^* \times \mathbb{N}$ be two parameterized problems. A parameterized reduction from P to Q is an algorithm that, given an instance (I, k) of P, outputs an instance(I', k') of Q such that

- (I, k) is a yes-instance of P iff (I', k') is a yes-instance of Q
- $k' \leq g(k)$ for some computable function g

Parameterized Reductions

Parameterized Reduction: Let $P, Q \subseteq \Sigma^* \times \mathbb{N}$ be two parameterized problems. A parameterized reduction from P to Q is an algorithm that, given an instance (I, k) of P, outputs an instance(I', k') of Q such that

- (I, k) is a yes-instance of P iff (I', k') is a yes-instance of Q
- $k' \leq g(k)$ for some computable function g
- the running time is $f(k)|x|^{\mathcal{O}(1)}$ for some computable function f.

It seems that parameterized complexity theory cannot be built on assuming P $\neq$ NP we have to assume something stronger. Let us choose a basic hypothesis:

Engineers' Hypothesis

Independent Set cannot be solved in time $f(k)n^{\mathcal{O}(1)}$.

It seems that parameterized complexity theory cannot be built on assuming P $\neq$ NP we have to assume something stronger. Let us choose a basic hypothesis:

Engineers' Hypothesis

Independent Set cannot be solved in time $f(k)n^{\mathcal{O}(1)}$.

Clique

Clique: Given a graph G, integer k, find a set S of k vertices such that $G[S]$ is a clique.

Clique

Clique: Given a graph G, integer k, find a set S of k vertices such that $G[S]$ is a clique.

Can we show Clique is not FPT?

Parameterized Reduction from Independent Set to Clique

Theorem

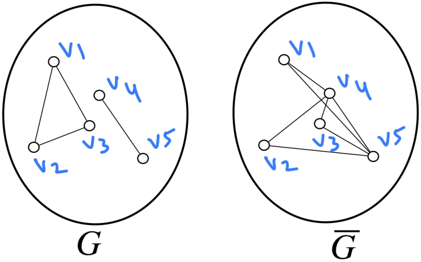
There is a parameterized reduction from Independent Set parameterized by k to Clique parameterized by k.

Parameterized Reduction from Independent Set to Clique

Theorem

There is a parameterized reduction from Independent Set parameterized by k to Clique parameterized by k.

$$(G, k) \iff (\overline{G}, k)$$

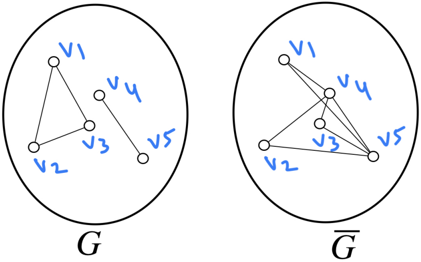


Parameterized Reduction from Clique to Independent Set

Theorem

There is a parameterized reduction from Clique parameterized by k to Independent Set parameterized by k.

$$(G, k) \iff (\overline{G}, k)$$



Clique on regular graphs

Regular graphs: Degree of all the vertices is same.

Clique on regular graphs

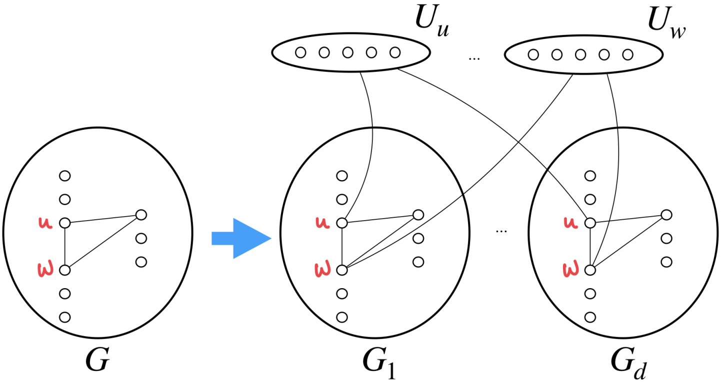
Regular graphs: Degree of all the vertices is same.

Theorem

There is a parameterized reduction from Clique to Clique on regular graphs.

Clique on regular graphs

If d is maximum degree of G, take d copies of G, $G_1, \cdots, G_d$.
For each vertex v in G, add a set U_v of $d - deg_G(v)$ vertices.
Make U_v adjacent to all the copies of v.



Partial Vertex Cover

Partial Vertex Cover: Given a graph G, integers k and t, find k vertices that cover at least t edges.

Partial Vertex Cover

Partial Vertex Cover: Given a graph G, integers k and t, find k vertices that cover at least t edges.

When do a set of vertices cover more edges?

Partial Vertex Cover is not FPT parameterized by k

Independent Set on regular graphs is not FPT.

Partial Vertex Cover is not FPT parameterized by k

Independent Set on regular graphs is not FPT.

Theorem

There is a parameterized reduction from Independent Set on regular graphs parameterized by k to Partial Vertex Cover parameterized by k.

$$(G, k) \iff (G, k, rk)$$

Multicolored Clique

Multicolored Clique: The vertices of the input graph G are colored with k colors and we have to find a clique containing one vertex from each color.

Multicolored Clique

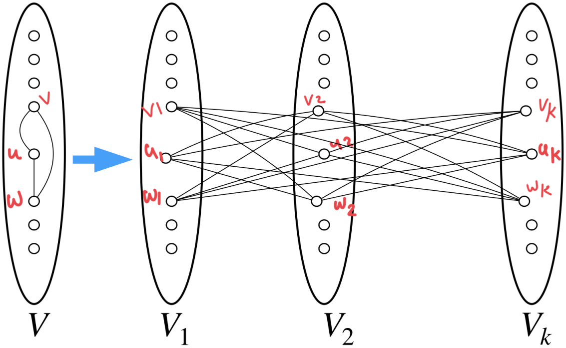
Multicolored Clique: The vertices of the input graph G are colored with k colors and we have to find a clique containing one vertex from each color.

Theorem

There is a parameterized reduction from Clique parameterized by k to Multicolored Clique parameterized by k.

Multicolored Clique

Create G' by replacing each vertex v with k vertices, one in each color class. If u and v are adjacent in the original graph, connect all copies of u with all copies of v.



$$(G, k) \iff (G', k)$$

Dominating Set

Dominating Set: Given a graph G, find a set S of k vertices that dominate every vertex, that is $N[v] \cap S \neq \emptyset$ for every vertex v in G.

Dominating Set

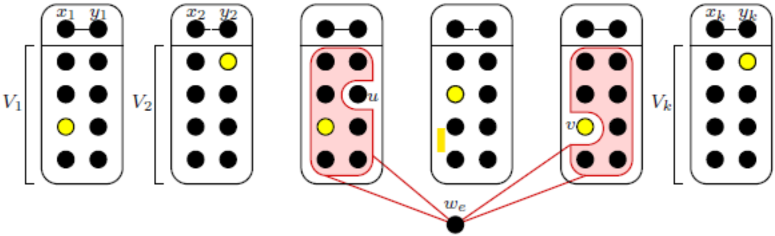
Multicolored Independent Set is not FPT.

Dominating Set

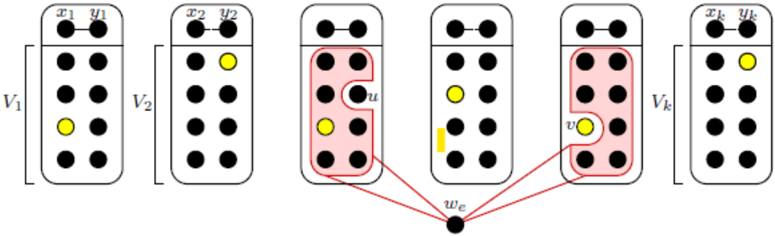
Multicolored Independent Set is not FPT.

Theorem

There is a parameterized reduction from Multicolored Clique parameterized by k to Dominating Set parameterized by k.

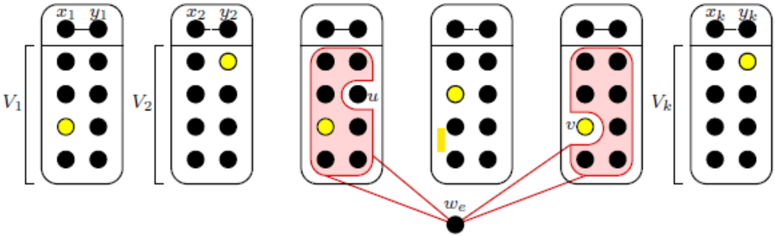


For every edge $e = uv$, $u \in V_i, v \in V_j$, add an additional vertex w_e, make w_e adjacent to all vertices in $V_i \cup V_j \setminus \{u, v\}$



For every edge $e = uv$, $u \in V_i, v \in V_j$, add an additional vertex w_e, make w_e adjacent to all vertices in $V_i \cup V_j \setminus \{u, v\}$

Add x_i, y_i for each V_i, make adjacent to every vertex in V_i.



For every edge $e = uv$, $u \in V_i, v \in V_j$, add an additional vertex w_e, make w_e adjacent to all vertices in $V_i \cup V_j \setminus \{u, v\}$

Add x_i, y_i for each V_i, make adjacent to every vertex in V_i.

Make each V_i clique.

Thank You!