Recognizing (p,q)-cluster graphs

Pinar Heggernes, Daniel Lokshtanov, Christophe Paul and Jesper Nederlof

---

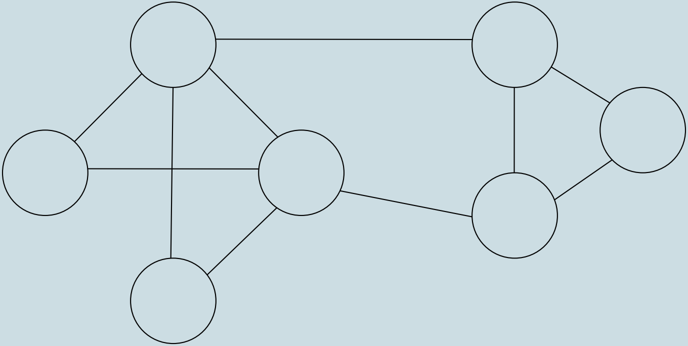
29 June 2010

36th International Workshop on Graph Theoretic

Outline

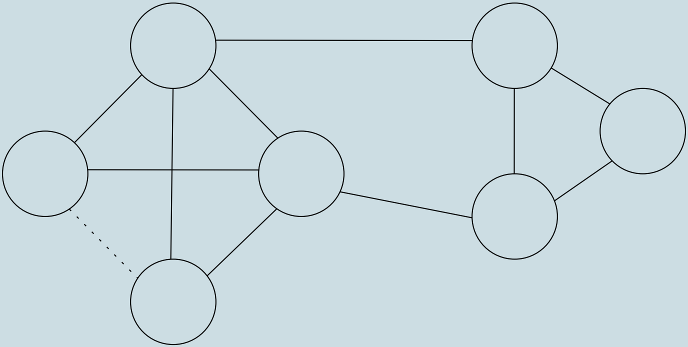
Cluster Editing

- We are given a graph which should be a disjoint union of cliques, but it is not due to faulty data.



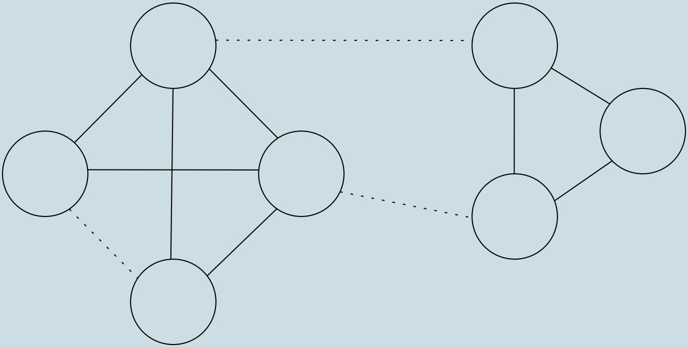
Cluster Editing

- We are given a graph which should be a disjoint union of cliques, but it is not due to faulty data.



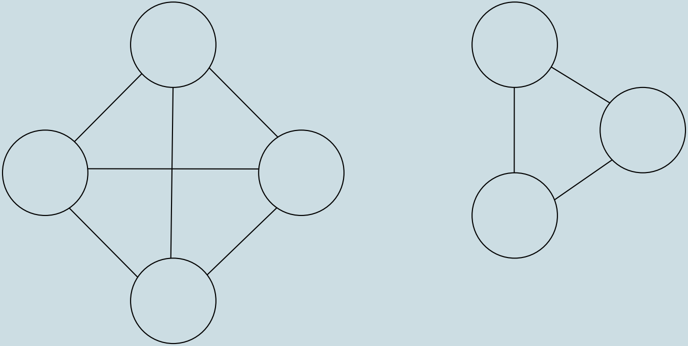
Cluster Editing

- We are given a graph which should be a disjoint union of cliques, but it is not due to faulty data.



Cluster Editing

- We are given a graph which should be a disjoint union of cliques, but it is not due to faulty data.



Cluster Editing

- The classical Cluster Editing problem is:
  - Given a graph G and integer k, is it possible to add and remove at most k edges to obtain a disjoint union of cliques?
- This problem, and many variants, are well studied and known to be NP-complete and FPT parameterized by k.

The high school problem

- Suppose you have to partition a huge number of 1st years children in classes on a school.

The high school problem

- Suppose you have to partition a huge number of 1st years children in classes on a school.
- All the children indicate their friendships.

The high school problem

- Suppose you have to partition a huge number of 1st years children in classes on a school.
- All the children indicate their friendships.
- In this application, friendships are symmetric.

The high school problem

- Suppose you have to partition a huge number of 1st years children in classes on a school.
- All the children indicate their friendships.
- In this application, friendships are symmetric.
- Minimize the number of friendships broken plus the number of non-friends in the same class.

The high school problem

- Suppose you have to partition a huge number of 1st years children in classes on a school.
- All the children indicate their friendships.
- In this application, friendships are symmetric.
- Minimize the number of friendships broken plus the number of non-friends in the same class.
    - Then this is Cluster Editing on the "friendship graph".

The high school problem

- Suppose you have to partition a huge number of 1st years children in classes on a school.
- All the children indicate their friendships.
- In this application, friendships are symmetric.
- Minimize the number of friendships broken plus the number of non-friends in the same class.
  - Then this is Cluster Editing on the "friendship graph".
  - Probably lots of editing has to be done → FPT algorithms are slow.

The high school problem

- Suppose you have to partition a huge number of 1st years children in classes on a school.
- All the children indicate their friendships.
- In this application, friendships are symmetric.
- Minimize the number of friendships broken plus the number of non-friends in the same class.
  - Then this is Cluster Editing on the "friendship graph".
  - Probably lots of editing has to be done → FPT algorithms are slow.
  - There might be classes with very few friends → Unbalanced.

The high school problem

- Suppose you have to partition a huge number of 1st years children in classes on a school.
- All the children indicate their friendships.
- In this application, friendships are symmetric.
- Minimize the number of friendships broken plus the number of non-friends in the same class.
  - Then this is Cluster Editing on the "friendship graph".
  - Probably lots of editing has to be done → FPT algorithms are slow.
  - There might be classes with very few friends → Unbalanced.
- Today: Minimize the maximum of the number of friendships broken and the number of non-friends over all classes.

(p,q)-cluster graphs

Definition ((p,q)-clustering)

Given a graph $G = (V, E)$, integers p, q

Asked a partition $C_1, \ldots, C_l$ of V s.t. for every C_i:

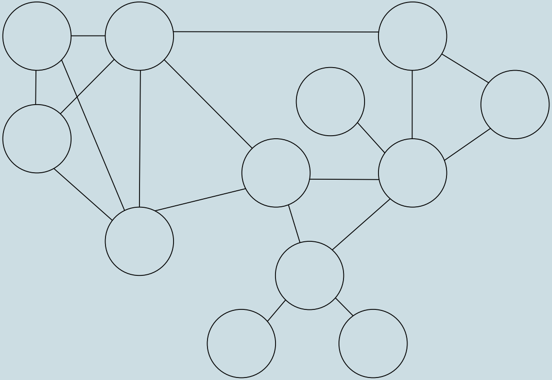
- the number of edges with exactly one endpoint in C_i is at most p, and
- the number of non-adjacent pairs $u, v \in C_i$ (with $u \neq v$) is at most q.

Definition ((p,q)-cluster graphs)

If (G, p, q) is a YES-instance, then G is called a (p,q)-cluster graph.

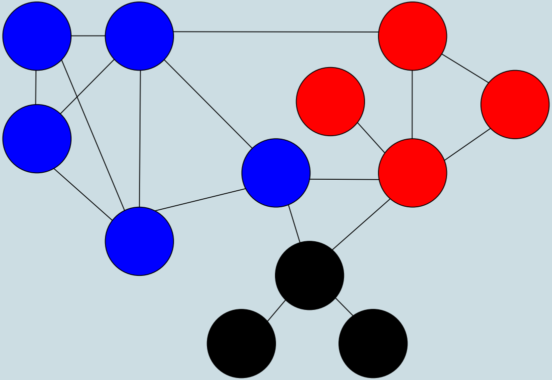
(p,q)-cluster graphs

Is this graph a (2,3)-cluster graph (2 missing, 3 redundant)?



(p,q)-cluster graphs

Is this graph a (2,3)-cluster graph (2 missing, 3 redundant)? Yes!



Special cases

- We can assume G is connected (otherwise, consider the connected components independently)

Special cases

- We can assume G is connected (otherwise, consider the connected components independently)
- (0,0)-clustering: YES iff G is a clique.

Special cases

- We can assume G is connected (otherwise, consider the connected components independently)
- (0,0)-clustering: YES iff G is a clique.
- (p,0)-clustering: YES iff there are at most p non-adjacent pairs.

UNIVERSITETET I BERGEN
Institutt for informatikk

Special cases

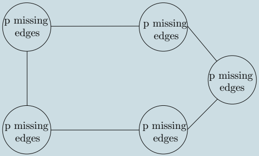
- We can assume G is connected (otherwise, consider the connected components independently)
- (0,0)-clustering: YES iff G is a clique.
- (p,0)-clustering: YES iff there are at most p non-adjacent pairs.
- (p,1)-clustering: YES iff there exists an edge whose removal gives a (p,0)-cluster graph.

Special cases

- We can assume G is connected (otherwise, consider the connected components independently)
- (0,0)-clustering: YES iff G is a clique.
- (p,0)-clustering: YES iff there are at most p non-adjacent pairs.
- (p,1)-clustering: YES iff there exists an edge whose removal gives a (p,0)-cluster graph.
- (p,2)-clustering: solvable with dynamic programming in polynomial time.

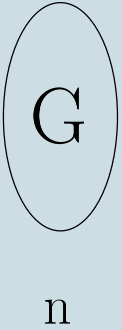
Special cases

- We can assume G is connected (otherwise, consider the connected components independently)
- (0,0)-clustering: YES iff G is a clique.
- (p,0)-clustering: YES iff there are at most p non-adjacent pairs.
- (p,1)-clustering: YES iff there exists an edge whose removal gives a (p,0)-cluster graph.
- (p,2)-clustering: solvable with dynamic programming in polynomial time.



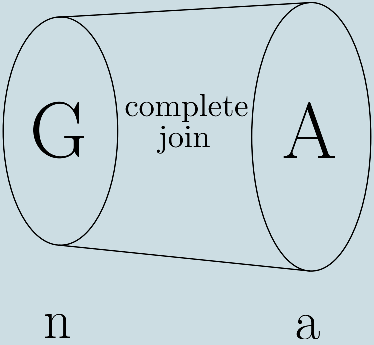
NP-completeness of (p,q)-clustering

Reduction from Clique: Let (G, k) be an instance of Clique. Create



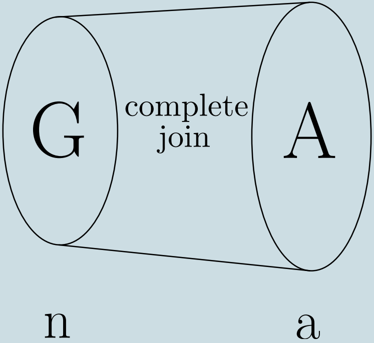
NP-completeness of (p,q)-clustering

Reduction from Clique: Let (G, k) be an instance of Clique. Create



NP-completeness of (p,q)-clustering

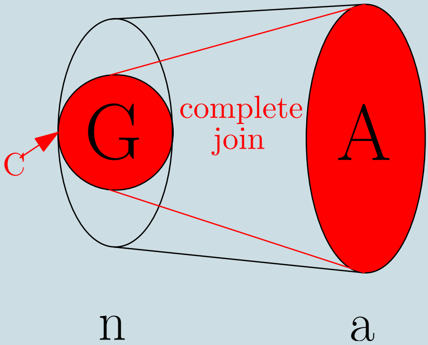
Reduction from Clique: Let (G, k) be an instance of Clique. Create



$a = (n-k)k+1$ $\quad$ $q = (n-k+1)a-1$

NP-completeness of (p,q)-clustering

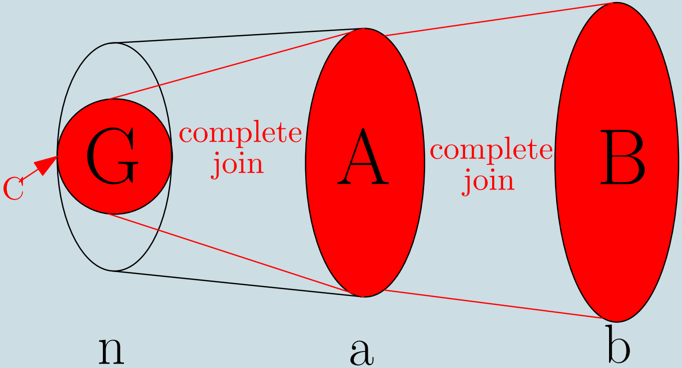
Reduction from Clique: Let (G, k) be an instance of Clique. Create



$a \geq (n-k)k+1 \qquad q = (n-k+1)a-1$

NP-completeness of (p,q)-clustering

Reduction from Clique: Let (G, k) be an instance of Clique. Create



$a = (n-k)k+1$ $\quad q = (n-k+1)a-1$ $\quad b = q-a+2$ $\quad p = bk$

(0,q)-clustering

Can we partition the vertices of G into cliques such that each clique has at most q edges to other cliques?

- Solvable in polynomial time!

Definition

A high degree vertex is a vertex of degree at least $q+1$. A good clique is a clique having at most q edges to other cliques.

Lemma

G is a $(0,q)$-cluster graph iff all its high degree vertices are in good cliques.

Lemma

Every good clique containing a high degree vertex is the closed neighborhood of a vertex.

Proof.

- Consider a good clique C of size c with high degree vertex v

□

Lemma

Every good clique containing a high degree vertex is the closed neighborhood of a vertex.

Proof.

- Consider a good clique C of size c with high degree vertex v
- There are at least $p+1-(c-1)$ edges leaving C from v

$\square$

Lemma

Every good clique containing a high degree vertex is the closed neighborhood of a vertex.

Proof.

- Consider a good clique C of size c with high degree vertex v
- There are at least $p+1-(c-1)$ edges leaving C from v
- If all other vertices of C have at least one leaving edge, there are at least $p+1-(c-1)+c-1=p+1$ edges.

□

UNIVERSITETET I BERGEN
Institutt for informatikk

Lemma

Every good clique containing a high degree vertex is the closed neighborhood of a vertex.

Proof.

- Consider a good clique C of size c with high degree vertex v
- There are at least $p+1-(c-1)$ edges leaving C from v
- If all other vertices of C have at least one leaving edge, there are at least $p+1-(c-1)+c-1=p+1$ edges.
- This doesn't happen since C is a good clique, so there is a vertex w without leaving edges.

$\square$

Lemma

Every good clique containing a high degree vertex is the closed neighborhood of a vertex.

Proof.

- Consider a good clique C of size c with high degree vertex v
- There are at least $p+1-(c-1)$ edges leaving C from v
- If all other vertices of C have at least one leaving edge, there are at least $p+1-(c-1)+c-1=p+1$ edges.
- This doesn't happen since C is a good clique, so there is a vertex w without leaving edges.
- Then $N[w]=C$ since C is a clique.

□

(0,q)-clustering

Lemma

G is a $(0, q)$-cluster graph iff all its high degree vertices are in good cliques.

Lemma

Every good clique containing a high degree vertex is the closed neighborhood of a vertex.

These two facts together give us a polynomial time algorithm!

Further remarks

- We introduced (p,q)-clustering.
- NP-complete when p, q are part of the input.
- Special cases $(p, 0)$, $(p, 1)$, $(p, 2)$ and $(0, q)$ are polynomial time solvable.
- We also proved $(1, 1)$, $(1, 2)$ and $(1, 3)$ to be polynomial time solvable.
- Very recently, Lokshtanov and Marx found $f(p)n^c$ and $f(q)n^c$ algorithms.