The Capacitated Cluster Edit Problem

Faisal N. Abu-Khzam

Department of Computer Science and Mathematics
Lebanese American University
Beirut, Lebanon

APAC 2012

Overview

- The Cluster Edit problem
- Aspects of desired solutions
- **The Capacitated versions**
- Effect of parameterization
- Experiments & concluding remarks

Cluster Edit

- ***Cluster graph***: graph whose connected components are cliques.
  - AKA ***transitive graph***
- ***Edge-edit*** operations
  - delete an edge
  - add an edge
- The Cluster Edit problem:

  Given: a simple undirected graph G; parameter k

  Question: can we perform k edge-edit operations to obtain a cluster graph?

Cluster Edit

- **NP-Complete** (many proofs!)
  [Krivanek-Moravek, 1986]
  [Bansal et al., 2004]
  [Shamir et al., 2004]
  [Komusiewicz-Uhlmann, 2012]

- **Fixed-Parameter Tractable**, due to a sequence of algorithms, including:
  [Gramm, Guo, Huffner & Niedermier, 2005]
  [Bocker, Briesemeister, Bui & Truss, 2009]
  [Böcker, 2011]

- **Linear Kernel**
  [Guo, 2009]
  [Chen & Meng, 2012] (2*k*-kernel)

Sources of difficulty

- **Sizes and number** of clusters unknown
- Existing fixed-parameter algorithms run in $O^*(c^k)$.
  (c = 1.618.. in the most recent algorithm of Böcker.)
- Yet, the number of edge-edit operations could be larger than n
- Exact $O^*(c^n)$ algorithms lag behind …
- Approximation might help. However:
  - Constant-factor approximation is known only when the number of clusters is 2 [Shamir et al., 2004]

Why Cluster Edit?

- Accurate clustering
- Unsupervised clustering
- Examples:
  - vertices of graph may correspond to protein sequences while an edge means “common domain”
  - graph may correspond to a social network where people may be clustered according to “common interests” etc…
- Best used on nearly clustered data

How Frequent are False Positives/Negatives?

- Frequency of errors could be low in general
- Our assumption: errors per vertex even less frequent
- Introduce new parameters: the vertex capacities
  - ***Add capacity***: amount of edges that can be added
  - ***Delete capacity***: amount of edges that can be deleted
- Similar work appeared recently in a paper by Komusiewicz & Uhlmann, who assume a bound on the total number of edge-edit operations per vertex.
- In some applications, it is very important to distinguish between the expected number of false positives and that of false negatives.

How Small can a Cluster be?

- Obviously, trivial clusters are not significant in real applications

- In general, clusters cannot be too small

- Consequently, add another parameter!

s = lower bound on cluster size

  - An application-specific threshold below which cluster size is not acceptable

  - Ranges between a small constant and a fraction of n

Capacitated Cluster Edit

- Given: a graph $G=(V,E)$; parameters a,d,s and k
- Question: can G be transformed into a cluster graph so that:
  - At most ***a*** edge additions allowed per vertex
  - At most ***d*** edge deletions allowed per vertex
  - Each (resulting) cluster has at least ***s*** elements
  - At most ***k*** total edge-edits permitted
- We refer to this problem by (a,d,s,k)-Cluster Edit, or (a,d,s,k)-CE
- In general, (a,d,s,k)-CE is NP-hard
  - the original formulation is a special case: set $s = 1$ and $a = d = k$

(a,d,s,k) Cluster Edit

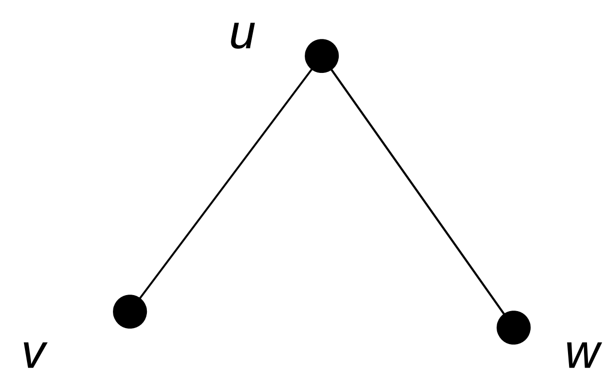
- When a=d=c we refer to problem as (c,s,k)-CE
- Capacities change during search. We assume the input is given with two capacity functions:
  - *a: V* → *{0, ..., c*} gives the number of edges that can be added per vertex
  - *d: V*→ *{0, …, c*} gives the number of edges that can be deleted per vertex
- In this formulation, a,d ≤ c and, for all v in V(G),
  - a(v) ≤ a and d(v) ≤ d

(a,d,s,k) Cluster edit

- Problem can be modified further by introducing **yet another parameter**:
  - the number of outliers
- Outliers may be identified during search:
  - Whenever a vertex has more/less than we can afford deleting/adding

Cluster Edit Terminology

- Permanent edge: not to be deleted
  - An added edge is implicitly permanent
- Forbidden edge: not to be added
  - Deleted edges are forbidden
- To cliquify a subgraph H is to make G[H] a clique by adding the missing edges
- Permanent clique:
  - all edges in clique are permanent
- Conflict triple
  - Induced path of length two
  - Edge-edit operations are needed as long as such paths exist



Special Cases

- (a,0,n,k)-Cluster Edit: Solvable in polynomial time
- (0,k,n,k)-Cluster Edit: NP-Hard, aka. Cluster Deletion
  - Solvable in $O^*(1.415^k)$ time [Böcker-Damaschke, 2011]
- (0,d,n,k)- Cluster Edit
  - Solvable in $O^*(1.415^k)$ time (run-time is better when s is small)
- (a,d,s,k)-Cluster Edit where a+d ≥ 4: NP-hard [Komusiewicz-Uhlmann, 2012]
- (1,1,s,k)-CE and (0,2,s,k)-CE: solvable in polynomial-time (to appear!)

(0,2,s,k) Cluster Edit

- If two adjacent vertices u and v have three common neighbors, then
  - uv becomes permanent!
- Any clique of size 5 or more becomes permanent
- If degree(v) > 5 then every “permanent” neighbor u of v must have at least 3 common neighbors with v. Otherwise, delete edge uv
- If degree(v) > 5 then
  - N[v] becomes a cluster (thus deleted)
- What if s = 6?
  - Then no vertex can be of degree < 5
  - And a vertex of degree-5 cannot loose any more edges, so its closed neighborhood becomes a cluster
- In general, we can deal with vertices of degrees 4 or 5. And (0,k,n,k)-CE is solvable in polynomial-time on graphs of degree < 4 [Komusiewicz-Uhlmann, 2012]

(1,1,s,k) Cluster edit

- Every clique of size 3 or more becomes permanent
- If two adjacent vertices have a common neighbor, then their edge becomes permanent
- If two non-adjacent vertices share exactly one common neighbor and one of them is of degree > 2, then their edge is forbidden
- If vertex u has only one neighbor v in a clique of size > 2, then delete the edge uv

(1,1,s,k) Cluster edit

- Connected components of the resulting instance are either cliques or triangle-free
  - every K3 is permanent and every vertex connected to two elements of a clique becomes part of the same (permanent clique),
- Any reduced instance (G,1,1,s,k) has maximum degree three.
- It follows that (1,1,5,k)-Cluster Edit is automatically solved by reduction rules:
  - If a component of the reduced instance is not a clique, then No solution!
- Moreover, setting s = 4 yields a simple poly-time algorithm:
  - Only C4 can lead to a clique of size 4 in only one way.
  - Note: cannot cliquify the neighborhood of a degree-three vertex.

General Problem Reduction

Reduction Procedures

- Preprocessing strategies that can be applied at any point during the search algorithm
- Reductions are based on rules, each of the form: <condition, action>
- If a reduction is not possible, then we have a no instance

Reduction Rule 1

- If there is a vertex *v* such that d(v) = 0, then
  - every edge incident on v becomes permanent
  - consequently: ***Cliquify*** *N(v)*

Reduction Rule 2

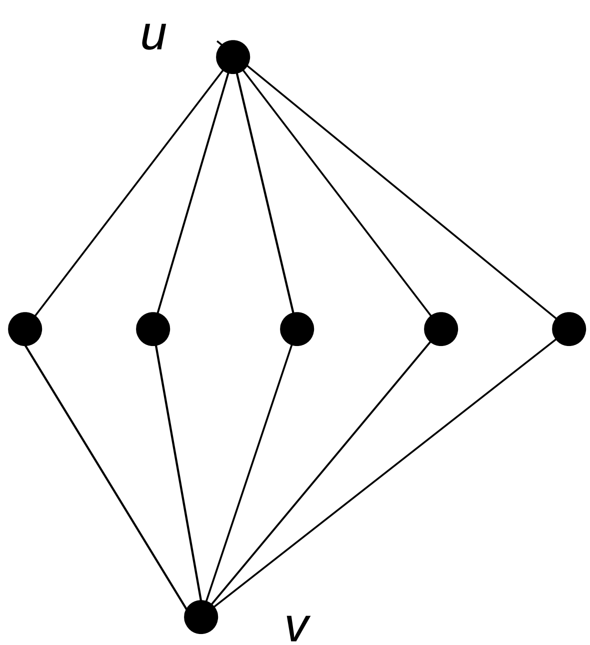
- If a vertex *v* satisfies a(v) = 0, then
  - every non-edge incident on v becomes forbidden.
- Moreover if a(v) = d(v) = 0, then:
  - all edges between N(v) and V\N[V] are deleted
  - Consequently: cluster containing v is completely identified, thus ***deleted***

Reduction Rule 3

- If edge uv if permanent while edge uw is forbidden, then vw is forbidden
- If edges uv and uw are permanent, then edge vw is permanent

Reduction Rule 4

- If two non-adjacent vertices u & v have d(u) + d(v) + 1 common neighbors (or just 2d+1), then
  - Add edge uv



If d = 2, add uv

Same holds if two adjacent vertices have 2d common neighbors (uv becomes a permanent edge)

Reduction Rule 6

If there is a vertex v satisfying

$$\deg(v) < s - a(v) - 1, \text{ then}$$

No instance (or v is an outlier!)

In particular, if $\deg(v) < s - a - 1$, then No

Reduction Rule 7

- If two non-adjacent vertices u and v have less than s – 2a common neighbors, then
  – set edge uv as forbidden

- Soundness: for u and v to belong to the same cluster, they must have at least s-2 common neighbors. After adding uv, the maximum number of common neighbors we can add is 2a – 2 (a-1 edges between u and N(v) and vice versa). The total number of common neighbors after adding all possible edges remains less than s-2 (= s-2a+2a-2).

- If u and v are adjacent and have < s – 2a - 2 common neighbors, then
  – delete edge uv

Reduced Instances

- By Rules 4 and 7, any two vertices u,v that are at distance two from each other satisfy:
  - $s-2a \le |N(u) \cap N(v)| \le 2d$
- What if $s - 2a > 2d$? (or $s > 2(a+d)$)
- Two cases:
  - No two non-adjacent vertices have common neighbors
  - No instance

Complexity of (a,d,s,k)-Cluster Edit

- Theorem: If $s > 2(a+d)$, then (a,d,s,k) is solvable in polynomial-time

- Proof:
  - When $s > 2(a + d)$, every reduced instance is a cluster graph.

- Thus, problem is hard only when the percentage of error is a (relatively) large fraction of the minimum cluster size!
  - When $a=d$, each of a and d is larger than $s/4$
  - In general $a+d \geq s/2$

Further Reduction

Large Cliques and Large Neighborhoods

Large Cliques

- If a clique C is of size ≥ 2d + 2 then C is permanent
  - Any two elements of C have at least 2d common neighbors, so (by Rule 4) they belong to the same cluster.
- Reduction Rule 8: If a vertex v has more than d neighbors in a permanent clique C, then ***join*** v to C
- Reduction Rule 9: If v has less than |C| - a neighbors in a permanent clique C, then ***detach*** v from C

Reduction Rule 10

- In a reduced instance, if a vertex v has **more** than 2(a+d) neighbors, then:

  N[v] is a cluster → cliquify and delete N[v]

- Soundness:
  - Any vertex u is either a neighbor of v or has $< 2d + 1$ common neighbors with v. In this latter case, v has more than 2a vertices that are not common with u. Edge uv would then be forbidden by Rule 7.

(a,d,s,k)-Cluster Edit Kernel

- Using the recent kernelization of Chen & Meng, we assume given graph has at most 2k vertices

- How about number of edges?

(a,d,s,k)-Cluster Edit Kernel

- **Theorem**: There is a reduction algorithm that takes an arbitrary instance of (a,d,s,k)-Cluster Edit and either determines that no solution exists or produces an equivalent instance whose size is bounded by 2(a+d)k

- Simply:
  - By Rule 10, each remaining vertex has degree ≤ 2(a+d)
  - Total number of edges ≤ 2k(2(a+d)/2).

Preliminary Experiments for (a,d,**n**,k)-CE

- We used a simple cluster-graph generator followed by random selection of edges to delete/add.
- On graphs of size ~1000 or more, with k > 140 and a = d = 3, an early version of (a,d,**n**,k) was much faster than most recent CE algorithm (> 400 hours to ~1 hour on some graphs of size 1000) [A. Ghrayeb, 2011].
- Using the reduction procedure described in this talk, and a few more reduction rules, many instances of the capacitated version can be solved during the preprocessing phase.
- More work is currently underway to implement and test reduction procedures when the parameter s is set.

Summary

- Adding the parameters **a**, **d** and **s** allowed us to solve Cluster Edit in polynomial time when $s > 2(a+d)$, which is a common case.
- When a+d is not small (with respect to s), we obtained a linear-size kernel.
- When $s \leq 2(a+d)$, setting s to be a (reasonably) large constant should yield improved fixed-parameter algorithms.

Lessons Learned

- General intractable problems are often
  - **Easy** to define
  - **Hard** to solve
  - **Hard** to apply
- Application-driven parameterizations could achieve practicality in two ways:
  - **Better problem models**
  - **Faster algorithmic solutions**