Clustering to Given Connectivities

Dimitrios M. Thilikos

Joint work with Petr Golovach (Пётр Головач)

AlGCo project team, CNRS, LIRMM

ESIGMA Project Kick-off meeting, May–June 2018

LAMSADE, Paris, June 01, 2018

CLUSTER DELETION problem:

Input: A graph and an integer k

Question: Can remove k edges from G so that the new graph is the disjoint union of cliques?

CLUSTER DELETION problem:

Input: A graph and an integer k

Question: Can remove k edges from G so that the new graph is the disjoint union of cliques?

Variants:

CLUSTER EDITING problem

(known as CORRELATION CLUSTERING)

Cluster Deletion problem:

Input: A graph and an integer k

Question: Can remove k edges from G so that the new graph is the disjoint union of cliques?

Variants:

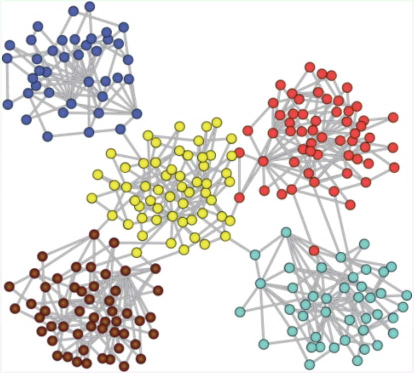
Cluster Editing problem

(known as Correlation Clustering)

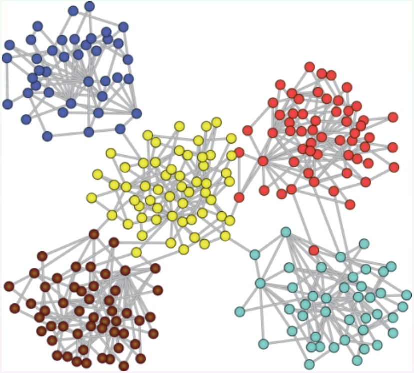
FPT-algorithm: $1.62^{k} + n^{3}$ [S. Böcker, A golden ratio parameterized algorithm for Cluster Editing, Journal of Discrete Algorithms Volume 16, October 2012, Pages 79-89]

What is a good clustering?

What is a good clustering?



What is a good clustering?



A partition of the vertices into vertex sets, called clusters, so that each cluster enjoys some desirable characteristics of "density" or "good interconnectivity", while having few edges between the clusters

FPT-algorithms and relaxations of clique density:

γ-quasi cliques: [P. Heggernes, D. Lokshtanov, J. Nederlof, C. Paul, and J. A. Telle, Generalized graph clustering: Recognizing (p, q)-cluster graphs, WG 2010]

FPT-algorithms and relaxations of clique density:

γ-quasi cliques: [P. Heggernes, D. Lokshtanov, J. Nederlof, C. Paul, and J. A. Telle, Generalized graph clustering: Recognizing (p, q)-cluster graphs, WG 2010]

Given a function μ and a parameter p, each cluster C should satisfy $\mu(C) \leq p$

[D. Lokshtanov and D. Marx, Clustering with local restrictions, Inf. Comput., 222 (2013), pp. 278–292.]

FPT-algorithms and relaxations of clique density:

γ-quasi cliques: [P. Heggernes, D. Lokshtanov, J. Nederlof, C. Paul, and J. A. Telle, Generalized graph clustering: Recognizing (p, q)-cluster graphs, WG 2010]

Given a function μ and a parameter p, each cluster C should satisfy $\mu(C) \leq p$

[D. Lokshtanov and D. Marx, Clustering with local restrictions, Inf. Comput., 222 (2013), pp. 278–292.]

HIGHLY CONNECTED DELETION: each cluster C has edge-connectivity bigger than $|C|/2$

[F. Hüffner, C. Komusiewicz, A. Liebtrau, and R. Niedermeier, Partitioning biological networks into highly connected clusters with maximum edge coverage, IEEE/ACM Trans. Comput. Biology Bioinform., 11 (2014)]

s-DEFECTIVE CLIQUE EDITING: a clique missing s edges

s-Defective Clique Editing: a clique missing s edges
Average-s-Plex Delection: average degree at least $|C| - s$

s-Defective Clique Editing: a clique missing s edges

Average-s-Plex Delection: average degree at least $|C| - s$

μ-Clique Deletion: average degree at least $\mu(|C| - 1)$

[J. Guo, I. A. Kanj, C. Komusiewicz, and J. Uhlmann, Editing graphs into disjoint unions of dense clusters, Algorithmica, 61 (2011), pp. 949–970]

s-Defective Clique Editing: a clique missing s edges

Average-s-Plex Delection: average degree at least $|C| - s$

μ-Clique Deletion: average degree at least $\mu(|C| - 1)$

[J. Guo, I. A. Kanj, C. Komusiewicz, and J. Uhlmann, Editing graphs into disjoint unions of dense clusters, Algorithmica, 61 (2011), pp. 949–970]

Moreover:

Clusters are of diameter at most s (s-clubs)

[S. Shahinpour and S. Butenko, Distance-based clique relaxations in networks: s-clique and s-club, in Models, Algorithms, and Technologies for Network Analysis, 2013]

Every vertex of a cluster should have an edge to all but at most $s-1$ other vertices of it (s-plexes)

[B. Balasundaram, S. Butenko, and I. V. Hicks, Clique relaxations in social network analysis: The maximum k-plex problem, Operations Research, 59 (2011)]

Every vertex of a cluster should have an edge to all but at most $s-1$ other vertices of it (s-plexes)

[B. Balasundaram, S. Butenko, and I. V. Hicks, Clique relaxations in social network analysis: The maximum k-plex problem, Operations Research, 59 (2011)]

# of clusters to be obtained is exactly p

[F. V. Fomin, S. Kratsch, M. Pilipczuk, M. Pilipczuk, and Y. Villanger, Tight bounds for parameterized complexity of cluster editing with a small number of clusters, J. Comput. Syst. Sci., 80 (2014]

t-CUT problem:

Input: A graph G and an integer k

Question: is there a partition of a graph into exactly t nonempty components such that the total number of edges between the components is at most k

t-Cut problem:

Input: A graph G and an integer k

Question: is there a partition of a graph into exactly t nonempty components such that the total number of edges between the components is at most k

The above problem is NP-complete.

[O. Goldschmidt and D. S. Hochbaum, A polynomial algorithm for the k-cut problem for fixed k, Math. Oper. Res., 19 (1994)]

t-CUT problem:

Input: A graph G and an integer k

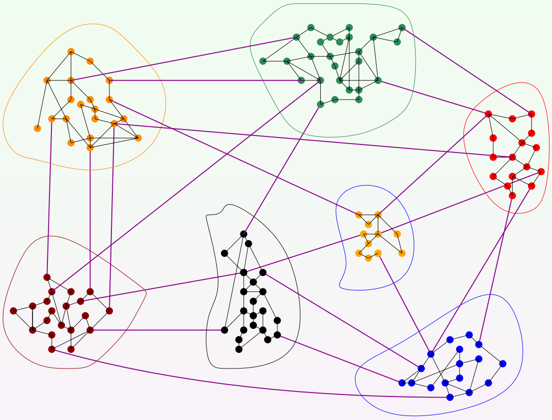
Question: is there a partition of a graph into exactly t nonempty components such that the total number of edges between the components is at most k

The above problem is NP-complete.

[O. Goldschmidt and D. S. Hochbaum, A polynomial algorithm for the k-cut problem for fixed k, Math. Oper. Res., 19 (1994)]

The above problem is NP-complete.

[K. Kawarabayashi and M. Thorup, The minimum k-way cut of bounded size is fixed-parameter tractable, in FOCS 2011, IEEE Computer Society, 2011

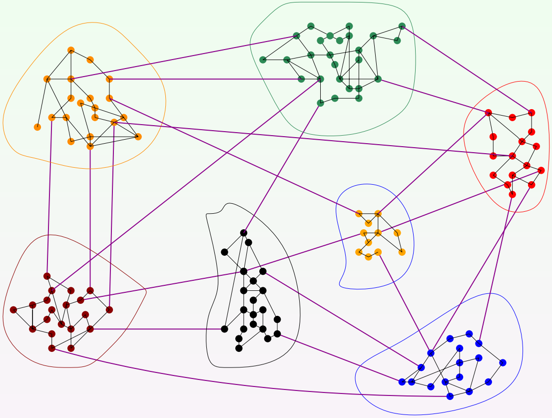


a partition into exactly t nonempty components such that the total number of edges between the components is at most k

Clustering to Given Connectivities

Input: An n-vertex graph G, a t-tuple $\Lambda = \langle \lambda_1, \ldots, \lambda_t \rangle$ of positive integers, $\lambda_1 \leq \ldots \leq \lambda_t$, and a nonnegative integer k.

Task: Decide whether there is a set $F \subseteq E(G)$ with $|F| \leq k$ such that $G - F$ has t connected components $G_1, \ldots, G_t$ where each G_i is edge λ_i-connected for $i \in \{1, \ldots, t\}$.



a partition into exactly t nonempty components such that the total number of edges between the components is at most k and the connectivities of the clusters are $\lambda_1, \ldots, \lambda_t$

The CLUSTERING TO GIVEN CONNECTIVITIES is NP-complete as a special case of t-CUT

The CLUSTERING TO GIVEN CONNECTIVITIES is NP-complete as a special case of t-CUT

Main result:

Theorem

CLUSTERING TO GIVEN CONNECTIVITIES *is* FPT

The Clustering to Given Connectivities is NP-complete as a special case of t-Cut

Main result:

Theorem

Clustering to Given Connectivities *is* FPT

by an algorithm of time $2^{2^{2^{2^{2^{2^{O(k)}}}}}} \cdot n^{O(1)}$.

Theorem

CLUSTERING TO GIVEN CONNECTIVITIES *is* FPT

by an algorithm of time $2^{2^{2^{2^{2^{O(k)}}}}} \cdot n^{O(1)}$.

Theorem

CLUSTERING TO GIVEN CONNECTIVITIES *is* FPT

by an algorithm of time $2^{2^{2^{2^{2^{2^{O(k)}}}}}} \cdot n^{O(1)}$.

Techniques:

1: Use Recursive Understanding to solve the connected case

Theorem

CLUSTERING TO GIVEN CONNECTIVITIES *is* FPT

by an algorithm of time $2^{2^{2^{2^{2^{2^{O(k)}}}}}} \cdot n^{O(1)}$.

Techniques:

1: Use Recursive Understanding to solve the connected case

2: Use DP to reduce the problem to the connected case

Theorem

CLUSTERING TO GIVEN CONNECTIVITIES *is* FPT

by an algorithm of time $2^{2^{2^{2^{2^{O(k)}}}}} \cdot n^{O(1)}$.

Techniques:

1: Use Recursive Understanding to solve the connected case

2: Use DP to reduce the problem to the connected case

Both are too complicated to explain here...

Theorem

Clustering to Given Connectivities is FPT

by an algorithm of time $2^{2^{2^{2^{2^{2^{O(k)}}}}}} \cdot n^{O(1)}$.

Techniques:

1: Use Recursive Understanding to solve the connected case
2: Use DP to reduce the problem to the connected case

Both are too complicated to explain here...

Question: Is there a low parameterized-dependence algorithm?

Merci!