On Clique (Cluster) Problems and their Applications

Zhuoli Xiao

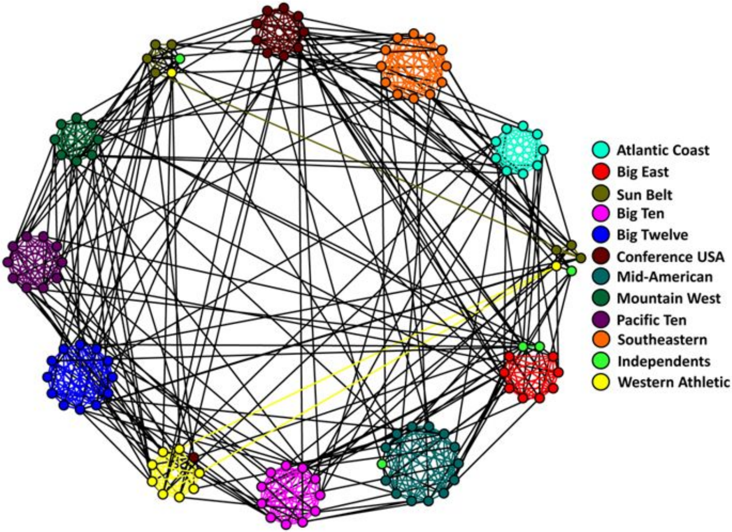
University of Victoria

zhuolixiao0818@gmail.com

November 6, 2018

Overview

Motivation: Cliques and Community Detection



Cliques

- *Complete* graph $G = (V, E)$: for every $u, v \in V$ there is $uv \in E$
- *Clique*: vertex set of complete graph or subgraph
- Note: singleton is clique

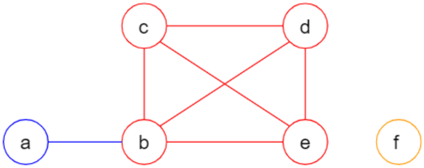


Figure: An Example of Cliques.

Problem Definition: k-Edge Clique Partition (ECP)

Input: Graph $G = (V, E)$, positive integer k
Parameter: k
Question: Does there exist *edge clique partition* CP where

- $CP = \{C_1, C_2, \ldots, C_l\}$
- $l \leq k$
- each $C_i \in CP$ is clique in G
- for each edge $uv \in E$, there is exactly one $C_i \in CP$ with $u, v \in CP$

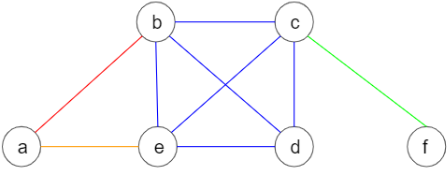


Figure: Edge clique partition $CP = \{\{a, b\}, \{a, e\}, \{b, c, d, e\}, \{c, f\}\}$ for G

Fixed Parameter Tractability (FPT)[5]

Class FPT (Fixed-parameter tractability): Parameterized decision problem k-P is *fixed-parameter tractable*, or a member of FPT, if there exists an algorithm A that solves every instance $< X, k >$ of k-P in running time $O(|X|^{O(1)} + f(k))$ or $O(|X|^{O(1)} \cdot f(k))$.

A called *fixed-parameter algorithm* for k-P

Instance of k-Edge Clique Partition: $I =< G, k >$

FPT Algorithm for ECP: Reduction Rule 1

Rule 1 [5]

If there is vertex $v \in V$ with $deg(v) = 0$, then remove v from G.

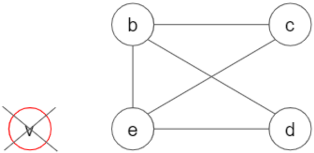


Figure: Reducing $I =< G, k >$ to $I' =< G - v, k >$

Rule 2 [5]

If there is a vertex $v \in V$ with $deg(v) = 1$, then
(1) remove v from G
(2) decrease k by one.

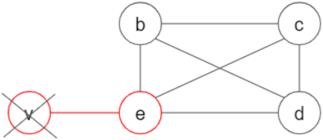


Figure: Reducing $I =< G, k >$ to $I' =< G - v, k - 1 >$

FPT Algorithm for ECP: Reduction Rule 3

Rule 3 [5]

If there is edge $uv \in E$ with $N(u) \cap N(v) = \emptyset$, then
(1) remove edge uv from G
(2) decrease k by one

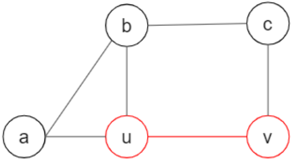


Figure: Reducing $I =< G, k >$ to $I' =< G - uv, k - 1 >$

FPT Algorithm for ECP: Reduction Rule 4

Rule 4 [5]

If there is vertex $v \in V$ with $|N[v]| > k$, $N[v]$ is clique, then
(1) remove $E(N[v])$ from G
(2) reduce k by one

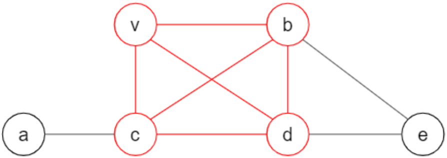


Figure: Reducing $I =< G, 3 >$ to $I' =< G - \{vb, vc, vd, bc, bd, cd\}, 2 >$

FPT Algorithm for ECP: Kernelization

Reduction Rules

- Run in polynomial time
- Are applied as long as applicable
- When not applicable, we attain an instance of size at most k^2

An instance whose size depends only on the parameter is called **kernelized**

FPT Algorithm for ECP: Bounded Search Tree Algorithm

Algorithm 1 Bounded Search-Tree Algorithm for a kernelized instance of k-EDGE CLIQUE PARTITION

Input: A graph $G = (V, E)$, an initially empty set CP to store cliques, an integer k.
Question: Is there an edge clique partition of size at most k for G?

function $C_{Partition}(G, CP, k)$
    if $E = \emptyset$ **then return** TRUE
    else if $k = 0$ **then return** FALSE
    else
        choose $uv \in E$ such that $|N(u) \cap N(v)|$ is minimum
        find a set S of all cliques in $N(u) \cap N(v)$
        for each $K \in S$ **do**
            $K' = K \cup \{u, v\}$
            $CP' = CP \cup \{K'\}$
            $k' = k - 1$
            $G' := G - E(K')$
            if $C_{Partition}(G', CP', k')$ **then return** TRUE
    return FALSE

Figure: Bounded Search Tree Algorithm for k-EDGE CLIQUE PARTITION [5]

FPT Algorithm for ECP: Bounded Search Tree Algorithm

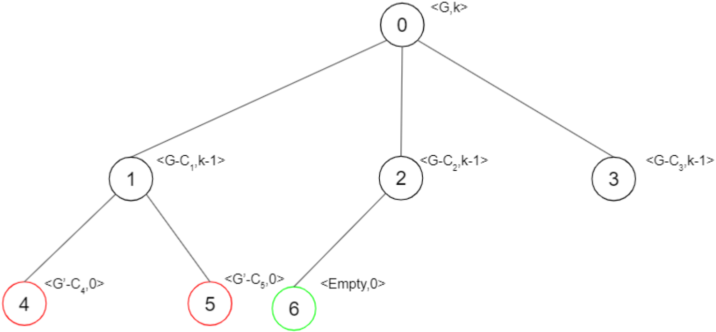


Figure: The Corresponding Bounded Search Tree

FPT Algorithm for ECP: Running Time Analysis [5]

- Reduction Rules
  - Rule 1: $O(n)$
  - Rule 2: $O(n)$
  - Rule 3: $O(n^3)$
  - Rule 4: $O(n^3)$
- Bounded Search Tree
  - $O((2^{k^2-2})^k 2^{k^2-2} k^4)$

Total running time: $O(n^3 k + (2^{k^2-2})^k 2^{k^2-2} k^4)$

New Reduction Rules for ECP

Rule 5

If there are $u, v, w \in V$ s.t. $uv, uw, vw \in E$, $deg(u) = 2$, $deg(v) = 2$, $deg(w) \geq 2$ then

(1) remove vertices u and v
(2) reduce k by one

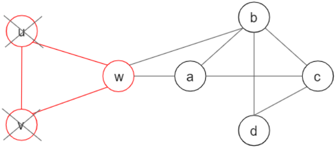
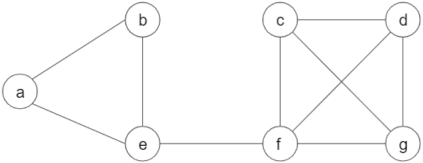
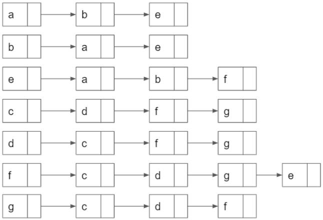


Figure: Reducing $I =< G, k >$ to $I' =< G - \{uv, uw, vw\}, k - 1 >$

Application of ECP: Graph Data Compression



Before Compression



After Compression

Cliques: $\{a, b, e\}$, $\{e, f\}$,$\{c, d, f, g\}$

Problem Definition: k-Edge Clique Cover (ECC) [4]

Input: Graph $G = (V, E)$, positive integer k
Parameter: k
Question: Does there exist *edge clique cover* CC where

- $CC = \{C_1, C_2, \dots, C_l\}$
- $l \leq k$
- each $C_i \in CC$ is clique in G
- for each $uv \in E$, there is $C_i \in CC$ with $u, v \in CC$

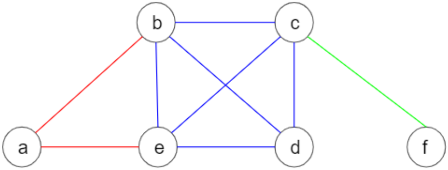


Figure: Edge clique cover $CC = \{\{a, b, e\}, \{b, c, d, e\}, \{c, f\}\}$ for G

Application of ECC: Compact Letter Display [3]

Treatments: conditions under which experiment is performed

Given list of pairs of significant different treatments: how to represent the significant difference between treatments with a binary matrix M with minimum number of columns?

		Irrigation regime				
		None	Light	Medium	Heavy	Xheavy
Polymer	No P4	No water No P4 (Treatment 1)	Light water No P4 (Treatment 2)	Medium water No P4 (Treatment 3)	Heavy water No P4 (Treatment 4)	Xheavy water No P4 (Treatment 5)
	With P4	No water With P4 (Treatment 6)	Light water With P4 (Treatment 7)	Medium water With P4 (Treatment 8)	Heavy water With P4 (Treatment 9)	Xheavy water With P4 (Treatment 10)

Figure: An Experimental Design Table with 10 treatments [6]

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Figure: Binary Matrix M for $H = \{(T_1, T_4), (T_1, T_5)\}$

Application of ECC: Algorithm

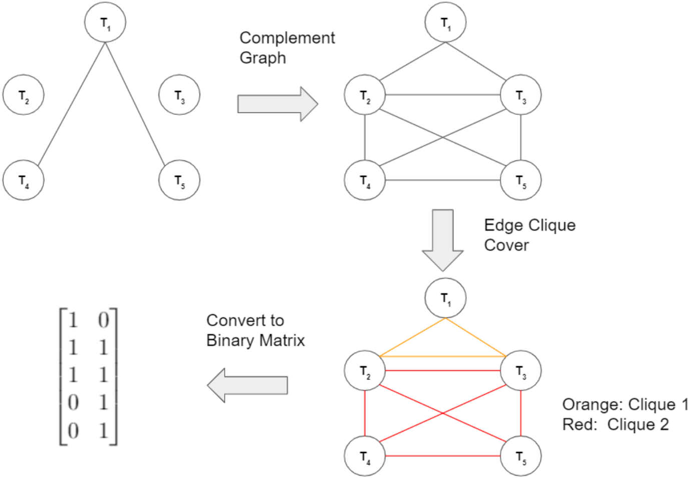


Figure: Create a Binary Matrix for a Given List of Significant Difference Pairs

Contributions

- Exploiting Clique-based Problems
- Investigated FPT algorithms (reduction rules and Bounded Search Tree algorithms) for $k-$Edge Clique Partition (ECP) and k-Edge Clique Cover (ECC) from literature
- Replenished proofs and analyzed running time for existing FPT Algorithms for ECP and ECC
- Suggested new reduction rules for ECP
- Surveyed existing applications of ECP and ECC
- Introduced new applications for ECP and ECC

Future Work

- Develop new lemmas and reduction rules for both k-EDGE CLIQUE PARTITION and k-EDGE CLIQUE COVER
- Find more applications of k-EDGE CLIQUE PARTITION
- More research and experiments on application of k-EDGE CLIQUE PARTITION on compact representation of graphs

References

[1] Downey G.R, Fellows R.M. and Stege U. "Parameterized complexity: A framework for systematically confronting computational intractability." Contemporary trends in discrete mathematics: From DIMACS and DIMATIA to the future. Vol. 49. 1999.

[2] Downey G.R and Fellows R.M. Fundamentals of Parameterized Complexity. Texts in Computer Science. Springer, 2013.

[3] Gramm J. and Guo J."Algorithms for compact letter displays: Comparison and evaluation.", Computational Statistics & Data Analysis 52.2 (2007): 725-736.

[4] Gramm J. and Guo J., "Data reduction and exact algorithms for clique cover.", Journal of Experimental Algorithmics (JEA) 13 (2009).

[5] Mujuni E. and Rosamond F., "Parameterized Complexity of the Clique Partition Problem.", Proceedings of the fourteenth symposium on Computing: the Australasian theory, Volume 77. Australian Computer Society, Inc., 2008.

[6] Weiss N., "Introductory Statistics", Addison Wesley, 2007.

Problem Definition: k-Edge Clique Cover (ECC)

Input: A graph $G = (V, E)$, a positive integer k.
Parameter: k
Question: Does there exist an *annotated edge clique cover* $ACC = \{C_1, C_2, \ldots, C_l\}$ of size $l \leq k$ where each $C_i \in ACC$ is a clique in G, such that for each edge $uv \in E - A$, $u, v \in C_i$ for some $C_i \in ACC$??

Reduction Rules for ECC

Rule 1 [4]

Let $I =< G, A, k >$ be an instance of k-Annotated Edge Clique Cover, and let $v \in V$ be a vertex such that $deg(v) = 0$. Then remove v from G.

Rule 2 [4]

If $I =< G, A, k >$ is an instance of k-Annotated Edge Clique Cover with $A \neq \emptyset$, and where $v \in V$ is a vertex in G with all incident edges of v are elements in A, then I' is obtained by creating G' from G via removing v and updating the set of covered edges to $A' = A - \{e \in E | e \text{ is incident to } vin\text{G}\}$.

Reduction Rules for ECC

Rule 3 [4]

Let $I =< G, A, k >$ be an instance of k-ANNOTATED EDGE CLIQUE COVER. If there exists an edge uv in G such that $N_{\{u,v\}}$ induces a clique, then include all edges induced by $N_{\{u,v\}} \cup \{u, v\}$ into A and reduce k by one.

Rule 4 [4]

Let $I =< G, A, k >$ be an instance of k-ANNOTATED EDGE CLIQUE COVER, and let $uv \in E$ with $N[u] = N[v]$. Then include all of u's incident edges into A.

Proof for Kernel

- When Rule 4 is not not applicable, largest clique is of size k.
- k cliques cover at most k^2 vertices.
- If there are more than k^2 vertices, some vertex does not appear in the solution, so its adjacent edge is not covered.
- It's a no instance.

Application of ECC: Algorithm

1. Build graph $G = (V, E)$ where each vertex $v_i \in V$ is treatment T_i that appears in H and there is an edge $e_{ij} \in E$ if and only if $(T_i, T_j) \notin H$.
2. Find a minimum edge clique cover for graph G.
3. Enumerate each clique in the obtained minimum edge clique cover.
4. Let the size of binary matrix M be defined by the size of a minimum edge clique cover for graph G.
5. We next build M as follows: if v_i is in clique j then $M[i][j] = 1$. Otherwise, $M[i][j] = 0$. Repeat until every cell in M is filled.

Application of ECP: Compression Algorithm

When given graph $G = (V, E)$, generate compact representation CP with S for G:

1. Include all singletons from G into set S
2. Compute edge clique partition CP for G
3. $CP \cup \bigcup S$ is compact representation for G