Eternal Connected Vertex Cover Problem in Graphs: Complexity and Algorithms

Kaustav Paul, **Arti Pandey**

Department of Mathematics
Indian Institute of Technology Ropar

CALDAM 2024
(IIT Bhilai)

Outline of The Talk

VERTEX COVER

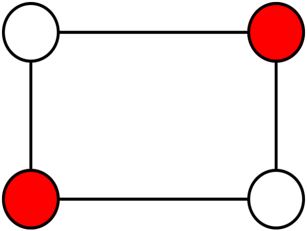
- A *vertex cover* of a graph $G = (V, E)$ is a set $S \subseteq V$ such that for every edge in E, at least one of its endpoints belongs to S.

VERTEX COVER

- A *vertex cover* of a graph $G = (V, E)$ is a set $S \subseteq V$ such that for every edge in E, at least one of its endpoints belongs to S.
- A minimum vertex cover of G is a vertex cover of G of minimum cardinality and its cardinality is the vertex cover number of G, denoted by *mvc(G)*.

VERTEX COVER

- A *vertex cover* of a graph $G = (V, E)$ is a set $S \subseteq V$ such that for every edge in E, at least one of its endpoints belongs to S.
- A minimum vertex cover of G is a vertex cover of G of minimum cardinality and its cardinality is the vertex cover number of G, denoted by *mvc(G)*.
- Example :



CONNECTED VERTEX COVER

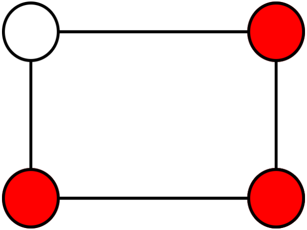
- A *connected vertex cover* of a graph $G = (V, E)$ is a set $S \subseteq V$ such that S is a vertex cover and $G[S]$ is connected.

CONNECTED VERTEX COVER

- A *connected vertex cover* of a graph $G = (V, E)$ is a set $S \subseteq V$ such that S is a vertex cover and $G[S]$ is connected.
- A minimum connected vertex cover of G is a connected vertex cover of G of minimum cardinality and its cardinality is the connected vertex cover number of G, denoted by *cvc(G)*.

CONNECTED VERTEX COVER

- A *connected vertex cover* of a graph $G = (V, E)$ is a set $S \subseteq V$ such that S is a vertex cover and $G[S]$ is connected.
- A minimum connected vertex cover of G is a connected vertex cover of G of minimum cardinality and its cardinality is the connected vertex cover number of G, denoted by *cvc(G)*.
- Example :



Minimum Vertex Cover Problem (MIN-VC)

Instance: A graph $G = (V, E)$.

Solution: A vertex cover D of minimum cardinality.

Minimum Vertex Cover Decision Problem (DECIDE-VC)

Instance: A graph $G = (V, E)$ and a positive integer $k \leq |V|$.

Question: Does there exist a vertex cover of cardinality $\leq k$?

Minimum Connected Vertex Cover Problem (MIN-CVC)

Instance: A graph $G = (V, E)$.

Solution: A connected vertex cover D of minimum cardinality.

Minimum Connected Vertex Cover Decision Problem (DECIDE-CVC)

Instance: A graph $G = (V, E)$ and a positive integer $k \leq |V|$.

Question: Does there exist a connected vertex cover of cardinality $\leq k$?

Eternal Vertex Cover

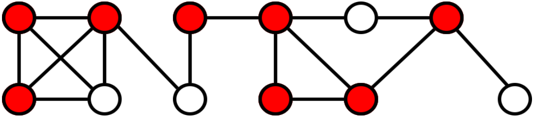
- The *Eternal Vertex Cover* problem is a dynamic variant of the vertex cover problem.
- The setting is as follows

Eternal Vertex Cover

- The *Eternal Vertex Cover* problem is a dynamic variant of the vertex cover problem.
- The setting is as follows
- It is a 2 player game, (*attacker* & *defender*) on a simple undirected graph G, which is played in rounds.

Eternal Vertex Cover

- The *Eternal Vertex Cover* problem is a dynamic variant of the vertex cover problem.
- The setting is as follows
- It is a 2 player game, (*attacker* & *defender*) on a simple undirected graph G, which is played in rounds.
- At Round 0, the defender can choose some vertices to place the guards.

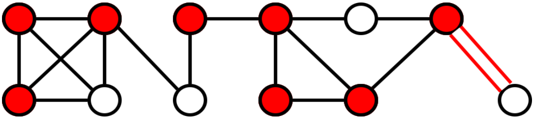


Eternal Vertex Cover

- At any round $i > 0$, the attacker gets to choose exactly one edge to *attack*.

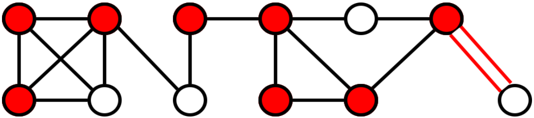
Eternal Vertex Cover

- At any round $i > 0$, the attacker gets to choose exactly one edge to *attack*.



Eternal Vertex Cover

- At any round $i > 0$, the attacker gets to choose exactly one edge to *attack*.



- Then, at the same round, the defender *defends* this attack if she can move the guards along the edges of the graph such that at least one guard moves along the attacked edge.

Eternal Vertex Cover

- Hence, at any round i, for each guard, the defender needs to decide from one of the following

Eternal Vertex Cover

- Hence, at any round i, for each guard, the defender needs to decide from one of the following
  1. To move from the current vertex to an adjacent vertex.

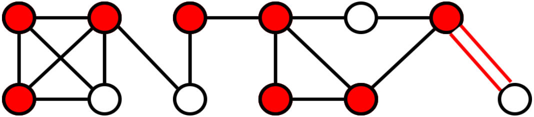
Eternal Vertex Cover

- Hence, at any round i, for each guard, the defender needs to decide from one of the following
  1. To move from the current vertex to an adjacent vertex.
  2. Not to move at all.

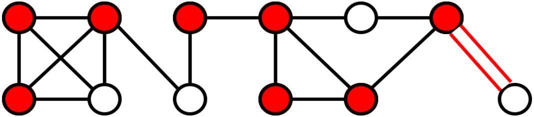
Eternal Vertex Cover

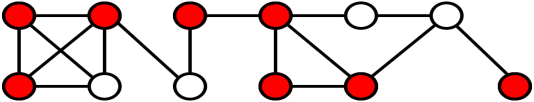
- Hence, at any round i, for each guard, the defender needs to decide from one of the following
  1. To move from the current vertex to an adjacent vertex.
  2. Not to move at all.
- After the movement of guards, if some guards move through the attacked edge and settle at the other end point of the attacked edge, then the defender successfully defends the attack at round i.

Eternal Vertex Cover

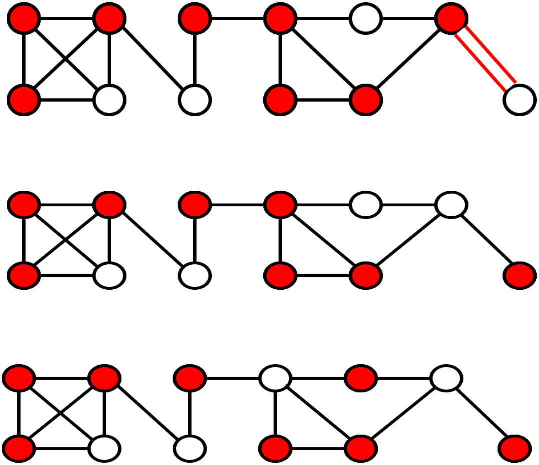


Eternal Vertex Cover

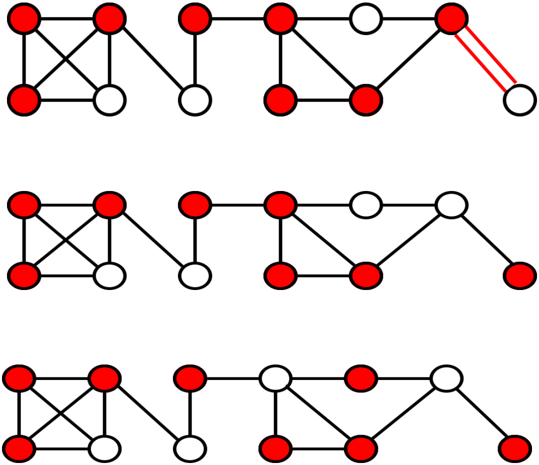




Eternal Vertex Cover



Eternal Vertex Cover



- Note that at each round $i \geq 0$, the set of underlying vertices of the defensive configuration of the guards should form a vertex cover.

Eternal Vertex Cover

- If no such move exists, *attacker* wins.

Eternal Vertex Cover

- If no such move exists, *attacker* wins.
- If the *defender* can defend the graph against every infinite sequence of attacks, then the defender wins.

Eternal Vertex Cover

- If such a guard allocation exists for which the defender has a winning strategy, the set of underlying vertices of the guards is called an *eternal vertex cover*.

Eternal Vertex Cover

- If such a guard allocation exists for which the defender has a winning strategy, the set of underlying vertices of the guards is called an *eternal vertex cover*.
- The minimum number of guards needed is called the *eternal vertex cover number* of the graph G and is denoted by *evc(G)*.

Eternal Vertex Cover

- If such a guard allocation exists for which the defender has a winning strategy, the set of underlying vertices of the guards is called an *eternal vertex cover*.
- The minimum number of guards needed is called the *eternal vertex cover number* of the graph G and is denoted by *evc(G)*.
- This problem was introduced in 2009 *(Klostermeyer et al. (2009))*.

Eternal Connected Vertex Cover

- If the additional condition is added to the Eternal Vertex Cover problem:
  Let at round i, if S_i be the set of vertices in which the guards are assigned, then $G[S_i]$ needs to be connected graph.

Eternal Connected Vertex Cover

- If the additional condition is added to the Eternal Vertex Cover problem:
  Let at round i, if S_i be the set of vertices in which the guards are assigned, then $G[S_i]$ needs to be connected graph.
- If such a guard allocation exists for which the defender has a winning strategy, the set of underlying vertices of the guards is called an *eternal connected vertex cover*.

Eternal Connected Vertex Cover

- If the additional condition is added to the Eternal Vertex Cover problem:
  Let at round i, if S_i be the set of vertices in which the guards are assigned, then $G[S_i]$ needs to be connected graph.
- If such a guard allocation exists for which the defender has a winning strategy, the set of underlying vertices of the guards is called an *eternal connected vertex cover*.
- The minimum number of guards with which the defender can come up with a winning strategy is called the *eternal connected vertex cover number* of the graph G and is denoted by *ecvc(G)*.

Eternal Vertex Cover

Minimum Eternal Vertex Cover Problem (MIN-EVC)

Instance: A graph $G = (V, E)$.

Solution: An eternal vertex cover D of minimum cardinality.

Minimum Eternal Vertex Cover Decision Problem (DECIDE-EVC)

Instance: A graph $G = (V, E)$ and a positive integer $k \leq |V|$.

Question: Does there exist an eternal vertex cover of cardinality $\leq k$?

Eternal Connected Vertex Cover

MINIMUM ETERNAL CONNECTED VERTEX COVER PROBLEM (MIN-ECVC)

Instance: A graph $G = (V, E)$.

Solution: An eternal connected vertex cover D of minimum cardinality.

MINIMUM ETERNAL CONNECTED VERTEX COVER DECISION PROBLEM (DECIDE-ECVC)

Instance: A graph $G = (V, E)$ and a positive integer $k \leq |V|$.

Question: Does there exist an eternal connected vertex cover of cardinality $\leq k$?

- **NP-Hardness results**
  - EVC is NP-Hard for general graphs. *(Fomin et al. (2010))*
  - EVC is NP-Hard even for locally connected graphs. *(Babu et al.(2021))*
  - EVC problem is NP-Hard even for bipartite graphs. *(Babu et al. (2022))*

- **NP-Hardness results**
  - EVC is NP-Hard for general graphs. *(Fomin et al. (2010))*
  - EVC is NP-Hard even for locally connected graphs. *(Babu et al.(2021))*
  - EVC problem is NP-Hard even for bipartite graphs. *(Babu et al. (2022))*
- **Approximation results**
  - A 2-approximation algorithm has been given for EVC on general graphs. *(Fomin et al. (2010))*

- **Algorithmic results**
  - EVC has been solved for paths, cycles and trees. *(Klostermeyer et al. (2009))*
  - The eternal vertex cover number has been given for generalized trees, where each edge of a tree is replaced by an elementary bipartite graphs. *(Hisashi et al. (2015))*
  - Polynomial-time algorithms have been proposed for cactus graphs and chordal graphs. *(Babu et al. (2021))*
  - A polynomial-time algorithm has been given to solve EVC for co-bipartite graphs. *(Babu et al. (2022))*
  - Linear time solvable for chain graphs and split graphs and efficiently solvable in P_4-free graphs *(Paul et al. (2023))*.

Literature Overview | Eternal Connected Vertex Cover

- **Combinatorial results**
  - For any undirected simple graph G, $mvc(G) \leq evc(G) \leq 2mvc(G)$. *(Klostermeyer et al. (2009))*
  - A characterization has been given for graphs for which $evc(G) = 2mvc(G)$. *(Klostermeyer et al. (2009))*
  - An upper bound has been given for $evc(G)$, i.e. $evc(G) \leq cvc(G) + 1$. *(Klostermeyer et al. (2009))*
  - A characterization has also been given for graphs satisfying some certain property, for which $evc(G) = mvc(G)$. *(Babu et al. (2021))*
  - Recently, all the bipartite graphs have been characterized for which $evc(G) = mvc(G)$. *(Neeldhara et al. (2023))*

Literature Overview | Eternal Connected Vertex Cover

All the following results were given in *(Fujito et al. (2020))*

- Efficiently solvable for chordal graphs.
- Efficiently solvable for cactus graphs, block graphs, and any graphs in which every block is either a simple cycle or a clique.
- NP-hard for locally connected graphs.
- A 2-approximation algorithm for general graphs

Literature Overview | Eternal Connected Vertex Cover

- *(Fujito et al. (2020))* also proved the following two theorems

Theorem 1

For any connected vertex cover C of any connected graph G, $C \cup \{x\}$ (where $x \in V \setminus C$) forms an initial configuration of eternal connected vertex cover. Hence $cvc(G) \leq cevc(G) \leq cvc(G) + 1$.

Theorem 2

Let $G = (V, E)$ be a connected graph. If $ecvc(G) = cvc(G)$, then for every vertex $v \in V$, there exists a minimum connected vertex cover that contains v.

Our contribution

In this paper,

1. A linear time algorithm is proposed to compute $cvc(G)$ and $ecvc(G)$ for chain graph G.
2. We show
   1. NP-hardness of the DECIDE-ECVC problem for Hamiltonian graphs.
   2. $cvc(\mu(G))$ can be computed efficiently, if at least one of the Hamiltonian cycle of G is given ($G = (V, E)$ is Hamiltonian).
   3. $mvc(\mu(G)) = |V| + 1$ and $ecvc(\mu(G)) = |V| + 2$.
3. We propose a polynomial time algorithm to solve the MIN-ECVC problem for cographs.

Algorithm for Chain graphs

Algorithm for Chain graphs

Chain graph

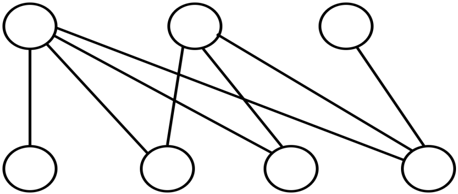
A bipartite graph $G = (X \cup Y, E)$ is said to be a chain graph if vertices in X can be ordered $(x_1, x_2, ..., x_{|X|})$, such that $N(x_1) \subseteq N(x_2) \subseteq ... \subseteq N(x_{|X|})$.

Algorithm for Chain graphs

Chain graph

A bipartite graph $G = (X \cup Y, E)$ is said to be a chain graph if vertices in X can be ordered $(x_1, x_2, ..., x_{|X|})$, such that $N(x_1) \subseteq N(x_2) \subseteq ... \subseteq N(x_{|X|})$.

- Example :



Algorithm for Chain graphs

At first, we compute the minimum connected vertex cover number for chain graphs.

- Calculate $mvc(G)$ for the chain graph $G = (X \cup Y, E)$.

Algorithm for Chain graphs

At first, we compute the minimum connected vertex cover number for chain graphs.

- Calculate $mvc(G)$ for the chain graph $G = (X \cup Y, E)$.
- Then two cases may appear
  1. $mvc(G) < |X|$.
  2. $mvc(G) = |X|$.

Algorithm for Chain graphs

At first, we compute the minimum connected vertex cover number for chain graphs.

- Calculate $mvc(G)$ for the chain graph $G = (X \cup Y, E)$.
- Then two cases may appear
  1. $mvc(G) < |X|$.
  2. $mvc(G) = |X|$.
- Here we assume $|X| \leq |Y|$.

Algorithm for Chain graphs

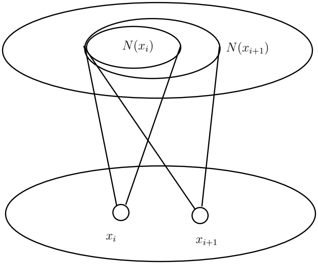
- **Case -1** **(**$mvc(G) < |X|$**)**

Algorithm for Chain graphs

- **Case -1** $(mvc(G) < |X|)$
- At first, we prove the following theorem

Theorem

For any minimum vertex cover S of G, if $x_i \in S$, then $x_{i+1} \in S$, for each $i \in [p-1]$. Similarly if $y_i \in S$, then $y_{i-1} \in S$, for each $i \in \{2, 3, \ldots, q\}$.



Algorithm for Chain graphs

- Since $mvc(G) < |X|$,

Algorithm for Chain graphs

- Since $mvc(G) < |X|$,
  - Any min vertex cover S has non empty intersection with X and Y.

Algorithm for Chain graphs

- Since $mvc(G) < |X|$,
  - Any min vertex cover S has non empty intersection with X and Y.
  - So, by the above theorem $y_1, x_p \in S$, hence $G[S]$ is connected.

Algorithm for Chain graphs

- Since $mvc(G) < |X|$,
  - Any min vertex cover S has non empty intersection with X and Y.
  - So, by the above theorem $y_1, x_p \in S$, hence $G[S]$ is connected.
- So, $cvc(G) = mvc(G)$.

Algorithm for Chain graphs

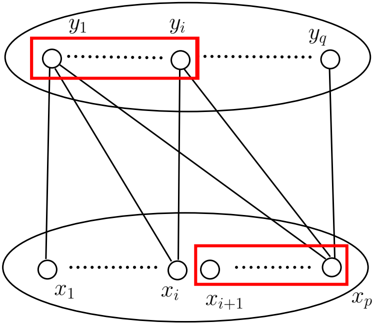
- **Case -2** **($mvc(G) = |X|$)**

Algorithm for Chain graphs

- **Case -2** ($mvc(G) = |X|$)

Observation 1

If there exists an index $i \in [p-1]$, such that $deg(x_i) = i$, then $N(x_i) \cup \{x_{i+1}, ..., x_p\}$ forms a connected vertex cover, hence $cvc(G) = mvc(G)$.



Algorithm for Chain graphs

Observation 2

If such an index i does not exist, then $cvc(G) = mvc(G) + 1$. Also, $X \cup \{y_1\}$ forms a minimum connected vertex cover.

Algorithm for Chain graphs

Observation 2

If such an index i does not exist, then $cvc(G) = mvc(G) + 1$. Also, $X \cup \{y_1\}$ forms a minimum connected vertex cover.

Hence, the following theorem can be concluded.

Theorem

Given a chain graph $G = (X \cup Y, E)$, $cvc(G)$ can be computed in linear time.

Algorithm for Chain graphs

- Now for computing $ECVC(G)$.

Algorithm for Chain graphs

- Now for computing $ECVC(G)$.
- Calculate $cvc(G)$.

Algorithm for Chain graphs

- Now for computing $ECVC(G)$.
- Calculate $cvc(G)$.
- Now, two cases can arise
  1. $cvc(G) \leq |X|$.
  2. $cvc(G) > |X|$.

- **Case 1** ($cvc(G) \leq |X|$)

- **<u>Case 1</u> ($cvc(G) \leq |X|$)**
- This implies $mvc(G) \leq |X|$.

- **Case 1** **($cvc(G) \leq |X|$)**
- This implies $mvc(G) \leq |X|$.
- Hence, there does not exist any connected vertex cover that contains x_1.

- **Case 1** **($cvc(G) \leq |X|$)**
- This implies $mvc(G) \leq |X|$.
- Hence, there does not exist any connected vertex cover that contains x_1.
- So, $ecvc(G) = cvc(G) + 1$.

- **Case 2** **(**$cvc(G) > |X|$**)**

Frame Title

- **<u>Case 2</u> ($cvc(G) > |X|$)**
- In this case, we have proved the following
  1. If $|X| < |Y|$, then $ecvc(G) = cvc(G) + 1$.
  2. If $|X| = |Y|$, then $ecvc(G) = cvc(G)$.

Frame Title

- **Case 2** ($cvc(G) > |X|$)
- In this case, we have proved the following
  1. If $|X| < |Y|$, then $ecvc(G) = cvc(G) + 1$.
  2. If $|X| = |Y|$, then $ecvc(G) = cvc(G)$.
- Hence, by the above cases, we conclude the following theorem.

Theorem

Given a chain graph $G = (X \cup Y, E)$, $ecvc(G)$ can be computed in linear time.

NP-hardness of the DECIDE-ECVC problem for Hamiltonian graphs

Definition

Hamiltonian graphs

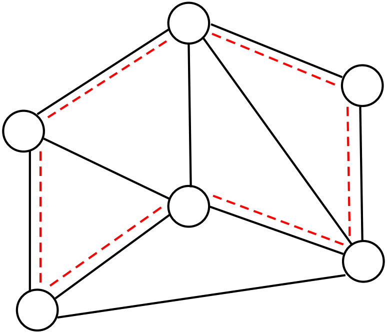
Given a graph G, a *Hamiltonian cycle* is a cycle C, such that C contains all the vertices of G. A graph that contains a Hamiltonian cycle is called a *Hamiltonian graph*.

Definition

Hamiltonian graphs

Given a graph G, a *Hamiltonian cycle* is a cycle C, such that C contains all the vertices of G. A graph that contains a Hamiltonian cycle is called a *Hamiltonian graph*.

- Example :



The Reduction

- The DECIDE-VC problem for Hamiltonian graphs is NP-hard.

The Reduction

- The DECIDE-VC problem for Hamiltonian graphs is NP-hard.
- We reduce the DECIDE-VC problem for Hamiltonian graphs to the DECIDE-ECVC problem for Hamiltonian graphs.

The Reduction

- The DECIDE-VC problem for Hamiltonian graphs is NP-hard.
- We reduce the DECIDE-VC problem for Hamiltonian graphs to the DECIDE-ECVC problem for Hamiltonian graphs.
- Let (G, k) be an instance of the DECIDE-VC problem.

The Reduction

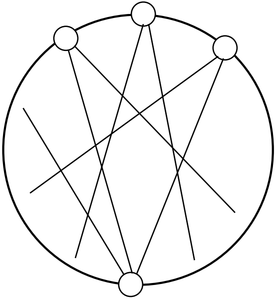
- The DECIDE-VC problem for Hamiltonian graphs is NP-hard.
- We reduce the DECIDE-VC problem for Hamiltonian graphs to the DECIDE-ECVC problem for Hamiltonian graphs.
- Let (G, k) be an instance of the DECIDE-VC problem.
- We construct H as follows:
  $H = (V(G) \cup \{u, v, w\}, E(G) \cup \{lk | l \in \{u, v\}, k \in V(G)\} \cup \{uw, vw\})$.

The Reduction

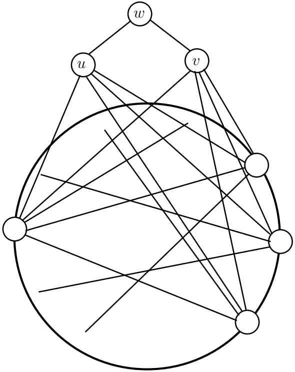
- The DECIDE-VC problem for Hamiltonian graphs is NP-hard.
- We reduce the DECIDE-VC problem for Hamiltonian graphs to the DECIDE-ECVC problem for Hamiltonian graphs.
- Let (G, k) be an instance of the DECIDE-VC problem.
- We construct H as follows:
  $H = (V(G) \cup \{u, v, w\}, E(G) \cup \{lk | l \in \{u, v\}, k \in V(G)\} \cup \{uw, vw\})$.
- $(H, k+3)$ is an instance of the DECIDE-ECVC problem.

The Reduction

- Here is an example of the reduction



(a) A Hamiltonian graph G



(b) The graph H

Figure: Visual depiction of the above reduction

Idea of the proof

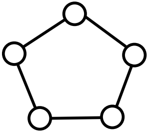
- If S be a vertex cover of G, with $|S| \leq k$, then $S \cup \{u, v, w\}$ is a eternal connected vertex cover of cardinality at most $k + 3$.
- For the converse, we showed that if S' be any eternal connected vertex cover of size $k + 3$, then S' contains u, v, w.
- Hence $S' \setminus \{u, v, w\}$ is a vertex cover of G of cardinality at most k.
- This way, we show that the DECIDE-ECVC problem is NP-hard for Hamiltonian graphs.

More Results on Hamiltonian graphs

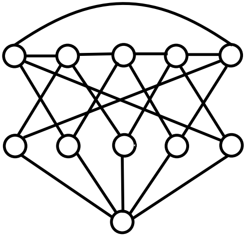
Mycielskian

Let $G = (V, E)$ be a graph. The Mycielskian of G, denoted as $\mu(G)$ contains G itself as a subgraph, together with $n + 1$ additional vertices: a vertex u_i corresponding to each vertex v_i of G, and an extra vertex w. Each vertex u_i is adjacent to w. In addition, for each edge $v_i v_j$ of G, the Mycielskian includes two edges, $u_i v_j$ and $v_i u_j$.

- Example:



G



$\mu(G)$

More Results on Hamiltonian graphs

- Moreover, we prove the following results

Theorem

Given a Hamiltonian graph $G = (V, E)$, (where $|V| = n$)

- $mvc(\mu(G)) = n + 1$.
- $ecvc(\mu(G)) = n + 2$.
- $cvc(\mu(G))$ can be computed in linear time if at least one Hamiltonian cycle representation of G is given.

Algorithm to solve the MIN-ECVC problem for cographs

Algorithm for Cographs

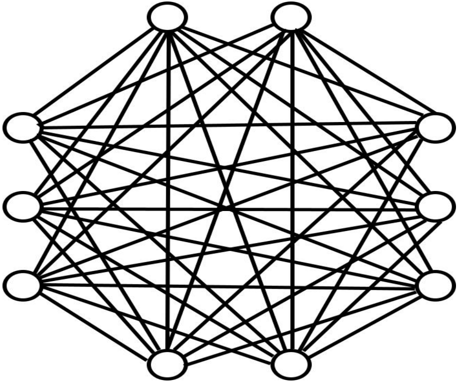
Cograph

A graph $G = (V, E)$ is called a cograph if it can be generated from K_1 by complementation and disjoint union. Recursively, the class of cographs can be defined as follows

1. K_1 is a cograph.
2. Complement of a cograph is a cograph.
3. G_1 and G_2 are cograph, then $G_1 \cup G_2$ is a cograph.

Algorithm for Cographs

- Cographs are exactlty P_4-free graphs.
- Example :



Algorithm for cographs

Theorem (Stewart, 1978)

Cographs can be represented as a join of k graphs, $G_1, G_2, \ldots, G_k$, where each G_i is either K_1 or a disconnected graph. This representation can be found in time $O(n^2)$.

- We introduce an algorithm *ECVC_CHECK* to compute $ecvc(G)$, when G has the property: Every minimum vertex cover of G is connected.
- This algorithm runs in $O(n^2)$ time.
- If $k \geq 3$, then every minimum vertex cover of G is also connected. Hence *ECVC_CHECK* works.

Algorithm for cographs

- If $k = 2$, then all the minimum vertex covers may not be connected.
- Let $G = G_1 \oplus G_2$, where G_1, G_2 are also cographs.
- Then we did an exhaustive case analysis on the following cases
  1. $mis(G) < |G_1| = |G_2|$.
  2. $mis(G) = |G_1| = |G_2|$.
  3. $mis(G) < |G_1| < |G_2|$.
  4. $|G_1| < mis(G) < |G_2|$.
  5. $|G_1| < mis(G) = |G_2|$.
  6. $|G_1| = mis(G) < |G_2|$.
- Hence we conclude the following theorem.

Theorem

Given a cograph G, $ecvc(G)$ can be computed in time $O(n^2)$.

Future Aspects

- One can try to solve the MIN-ECVC problem for the graphs where the MIN-CVC problem is solvable.
- Try to prove or disprove the following statement

Given a graph $G(V, E)$; $ecvc(G) = cvc(G)$, if for every vertex $v \in V$, there exists a minimum connected vertex cover that contains v.

References

- Babu J., Chandran L., Francis M., Prabhakaran V., Rajendraprasad D., Warrier N., On graphs whose eternal vertex cover number and vertex cover number coincide, Discrete Applied Mathematics, Volume 319, 2022, Pages 171-182, ISSN 0166-218X.
- Babu J., Misra N., Nanoti S.G, Eternal Vertex Cover on Bipartite Graphs, Computer Science – Theory and Applications, 2022, Pages 64-76.
- Babu J., Prabhakaran V., A new lower bound for the eternal vertex cover number of graphs, Journal of Combinatorial Optimization, 2021, Pages 1-17, ISSN 13826905.
- Babu J., Prabhakaran V., Sharma A., A Substructure based Lower Bound for Eternal Vertex Cover Number, Theoretical Computer Science, Volume 890, 2021, Pages 87-104, ISSN 0304-3975.

References

Bruno E., Jérôme M., Laurent G., Complexity and approximation results for the connected vertex cover problem in graphs and hypergraphs, Journal of Discrete Algorithms, Volume 8, Issue 1, 2010, Pages 36-49, ISSN 1570-8667.

Fomin F., Gaspers S., Golovach P., Kratsch D., Saurabh S., Parameterized Algorithm for Eternal Vertex Cover, Information Processing Letters, Volume 110, 2010, Pages 702-706, ISSN 0020-0190.

Fujito T., Hisashi A., Inoue S., On the Eternal Vertex Cover Numbers of Generalized Trees, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2015, E98.A(6), Pages 1153-1160.

References

- Fujito T., Nakamura T., Eternal Connected Vertex Cover Problem, Theory and Applications of Models of Computation, 2020, Pages 181-192.
- Klostermeyer W., Mynhardt C., Graphs with equal eternal vertex cover and eternal domination numbers, Discrete Mathematics, Volume 311, 2011, Pages 1371-1379, ISSN 0012-365X.
- Klostermeyer W., Mynhardt C.M., Edge protection in graphs, The Australasian Journal of Combinatorics, Volume 45, 2009.
- Paul K, Pandey A., Some Algorithmic Results for Eternal Vertex Cover Problem in Graphs. InInternational Conference and Workshops on Algorithms and Computation 2023 Mar 13 (pp. 242-253). Cham: Springer Nature Switzerland.

Thank You!

Questions?