

Acyclic Matching in Some Subclasses of Graphs

B.S.Panda Juhi Chaudhary*



Department of Mathematics
Indian Institute Of Technology Delhi
IWOCA 2020

June 9, 2020

Plan of Presentation

- Problem Statement
- Known Results
- Motivation
- Our Results
- Future Scope
- References

Definitions

Matching

Definition

A subset $M \subseteq E$ of edges of a graph $G = (V, E)$ is called a **matching** if no two edges in M are adjacent, that is, no two edges of M share a common vertex.

Matching

Definition

A subset $M \subseteq E$ of edges of a graph $G = (V, E)$ is called a **matching** if no two edges in M are adjacent, that is, no two edges of M share a common vertex.

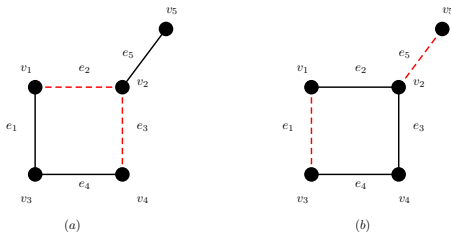


Figure: Graph showing Matching

Maximum matching problem

To find a matching of maximum cardinality.

Maximum matching problem

To find a matching of maximum cardinality.

Maximum matching problem is **Polynomial Time solvable** in general graphs.^a

^aEdmonds, Jack. "Maximum matching and a polyhedron with 0, 1-vertices." Journal of research of the National Bureau of Standards B 69.125-130 (1965): 55-56.

Acyclic Matching

Acyclic Matching

End points of a matching M are called M -saturated vertices.

Acyclic Matching

End points of a matching M are called M -saturated vertices.

Definition

A matching M is said to be an acyclic matching if the subgraph of G induced by the M -saturated vertices of G , i.e. $G[V(M)]$ is acyclic.

Acyclic Matching

End points of a matching M are called M -saturated vertices.

Definition

A matching M is said to be an acyclic matching if the subgraph of G induced by the M -saturated vertices of G , i.e. $G[V(M)]$ is acyclic.



Figure: Graph showing Acyclic Matching

Acyclic Matching

Definition

A matching M is said to be an acyclic matching if the subgraph of G induced by the M -saturated vertices of G , i.e. $G[V(M)]$ is acyclic.



Figure: Graph showing Acyclic Matching

Acyclic Matching

Definition

A matching M is said to be an acyclic matching if the subgraph of G induced by the M -saturated vertices of G , i.e. $G[V(M)]$ is acyclic.



Figure: Graph showing Acyclic Matching

Max-Acy-Matching problem

To find an acyclic matching of maximum size.

Max-Acy-Matching problem

To find an acyclic matching of maximum size.

Max-Acy-Matching-Decide is **NP-complete** in general graphs.^a

^aWayne Goddard, Sandra M Hedetniemi, Stephen T Hedetniemi, and Renu Laskar. Generalized subgraph-restricted matchings in graphs. Discrete mathematics, 293(1):129-138, 2005.

Known Results

- Max-Acy-Matching-Decide is **NP-complete** for
 - Bipartite graphs. ¹
 - Perfect elimination bipartite graphs. ¹
- Max-Acy-Matching problem is **Polynomial time solvable** for
 - Bipartite permutation graphs. ¹ ($O(m+n)$)
 - P_4 -free graphs, $2P_3$ -free graphs. ² ($O(m+n)$)
 - Chordal graphs. ³ $O(n^7)$ (solved under the name of r -degenerate matching)
- Characterization of graphs for which every maximum matching is acyclic. ²

¹B S Panda and Dinabandhu Pradhan. Acyclic matchings in subclasses of bipartite graphs. Discrete Mathematics, Algorithms and Applications, 4(04):1250050, 2012.

²Maximilian Fürst and Dieter Rautenbach. On some hard and some tractable cases of the maximum acyclic matching problem. Annals of Operations Research, 279(1- 2):291-300, 2019.

³Julien Baste and Dieter Rautenbach. Degenerate matchings and edge colorings. Discrete Applied Mathematics, 239:38-44, 2018.

Our Results

Max-Acy-Matching-Decide is shown to be **NP-complete** for

- Comb-convex bipartite graphs
- Dually chordal graphs

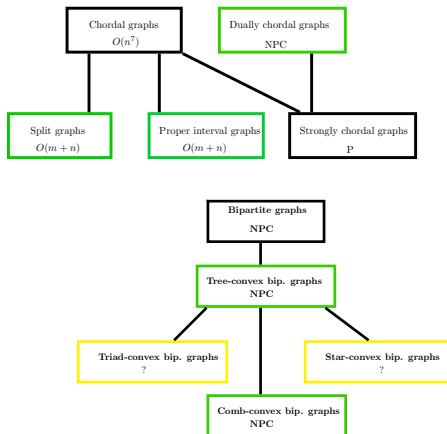
Max-Acy-Matching problem is shown to be **linear time solvable** for

- Split graphs
- Proper interval graphs

- Max-Acy-Matching problem is **hard to approximate** within a ratio of $n^{1-\epsilon}$ for any $\epsilon > 0$, unless $P = NP$

- Max-Acy-Matching problem is **APX-complete** for $2k + 1$ -regular graphs for $k \geq 3$.

Motivation



No approximation results on Max-Acy-Matching problem are known.

Our Results

Max-Acy-Matching-Decide is shown to be **NP-complete** for

- **Comb-convex bipartite graphs**
- Dually chordal graphs

Max-Acy-Matching problem is shown to be **linear time solvable** for

- Split graphs
- Proper interval graphs

- Max-Acy-Matching problem is **hard to approximate** within a ratio of $n^{1-\epsilon}$ for any $\epsilon > 0$, unless $P = NP$.

- Max-Acy-Matching problem is **APX-complete** for $2k + 1$ -regular graphs for $k \geq 3$.

Definition

Comb

A comb is a graph obtained by attaching a pendant vertex (tooth) to every vertex of a path (backbone).

Definition

Comb

A comb is a graph obtained by attaching a pendant vertex (tooth) to every vertex of a path (backbone).

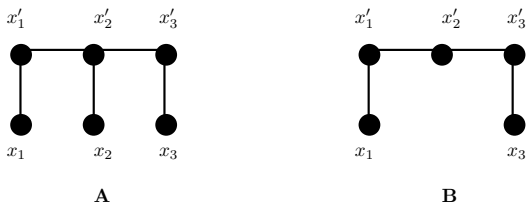


Figure: Comb graph

Comb-convex bipartite graph

Definition

A bipartite graph $G = (X, Y, E)$ is said to be **tree-convex**, if a tree $T = (X, E_X)$ can be defined, such that for every vertex y in Y , the $N(y)$ induces a subtree of T .

Comb-convex bipartite graph

Definition

A bipartite graph $G = (X, Y, E)$ is said to be **tree-convex**, if a tree $T = (X, E_X)$ can be defined, such that for every vertex y in Y , the $N(y)$ induces a subtree of T .

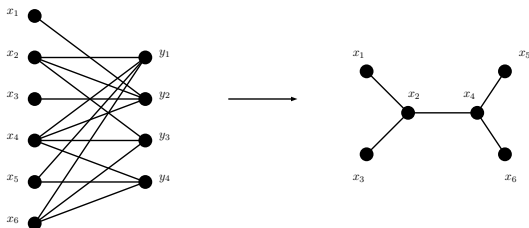


Figure: Tree-convex bipartite graph

Comb-convex bipartite graph

Definition

A bipartite graph $G = (X, Y, E)$ is said to be **tree-convex**, if a tree $T = (X, E_X)$ can be defined, such that for every vertex y in Y , the $N(y)$ induces a subtree of T .

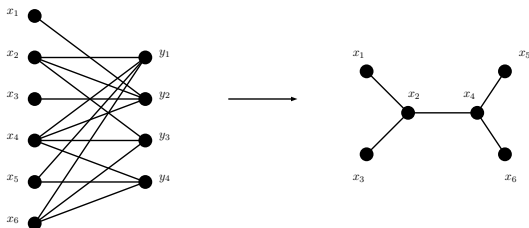


Figure: Tree-convex bipartite graph

If T is a comb then G is called a comb-convex bipartite graph.

Theorem

Max-Acy-Matching-Decide is **NP-complete** for comb-convex bipartite graphs.

Outline of proof:

Theorem

Max-Acy-Matching-Decide is **NP-complete** for comb-convex bipartite graphs.

Outline of proof:

Max-Acy-Matching-Decide \in NP.

Theorem

Max-Acy-Matching-Decide is **NP-complete** for comb-convex bipartite graphs.

Outline of proof:

Max-Acy-Matching-Decide \in NP.

Polynomial time reduction from Max-Acy-Matching-Decide for bipartite graphs^a.

Theorem

Max-Acy-Matching-Decide is **NP-complete** for comb-convex bipartite graphs.

Outline of proof:

Max-Acy-Matching-Decide \in NP.

Polynomial time reduction from Max-Acy-Matching-Decide for bipartite graphs^a.

Construct $H = (X_H, Y_H, E_H)$ from $G = (X, Y, E)$ as follows:

Theorem

Max-Acy-Matching-Decide is **NP-complete** for comb-convex bipartite graphs.

Outline of proof:

Max-Acy-Matching-Decide \in NP.

Polynomial time reduction from Max-Acy-Matching-Decide for bipartite graphs^a.

Construct $H = (X_H, Y_H, E_H)$ from $G = (X, Y, E)$ as follows:

$X_H = X \cup X'$, where $X' = \{x'_i \mid x_i \in X\}$, $Y_H = Y$, and $E_H = E \cup E'$, where $E' = \{x'_i y \mid x'_i \in X' \text{ and } y \in Y\}$.

^aB S Panda and Dinabandhu Pradhan. Acyclic matchings in subclasses of bipartite graphs. Discrete Mathematics, Algorithms and Applications, 4(04):1250050, 2012.

Comb-convex bipartite graphs (continued...)

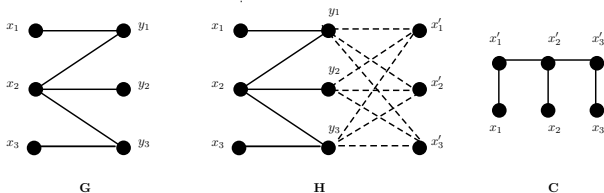


Figure: Construction of H from G

Comb-convex bipartite graphs (continued...)

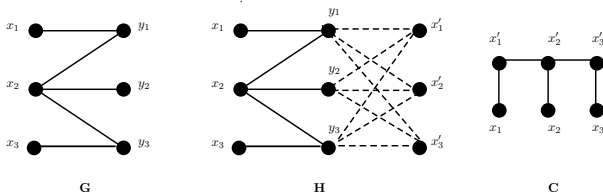


Figure: Construction of H from G

H is a comb-convex bipartite graph if X' is taken as the **backbone** and X is taken as the **teeth** of a comb C .

Comb-convex bipartite graphs (continued...)

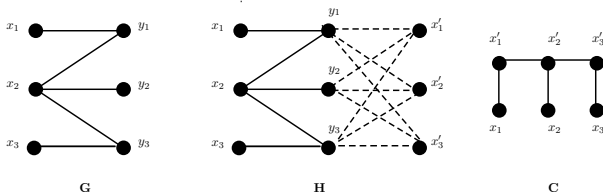


Figure: Construction of H from G

H is a comb-convex bipartite graph if X' is taken as the **backbone** and X is taken as the **teeth** of a comb C .

Claim:

G has an acyclic matching of size $\geq k \Leftrightarrow H$ has an acyclic matching of size $\geq k$.

Our Results

Max-Acy-Matching-Decide is shown to be **NP-complete** for

- Comb-convex bipartite graphs
- Dually chordal graphs

Max-Acy-Matching problem is shown to be **linear time solvable** for

- Split graphs
- Proper interval graphs

• Max-Acy-Matching problem is **hard to approximate** within a ratio of $n^{1-\epsilon}$ for any $\epsilon > 0$, unless $P = NP$.

• Max-Acy-Matching problem is **APX-complete** for $2k + 1$ -regular graphs for $k \geq 3$.

Split graphs

A graph $G = (V, E)$ is called a **split graph** if its vertex set V can be partitioned into two sets I and C such that I is an **independent set** and C is a **clique**.

Split graphs

A graph $G = (V, E)$ is called a **split graph** if its vertex set V can be partitioned into two sets I and C such that I is an **independent set** and C is a **clique**.

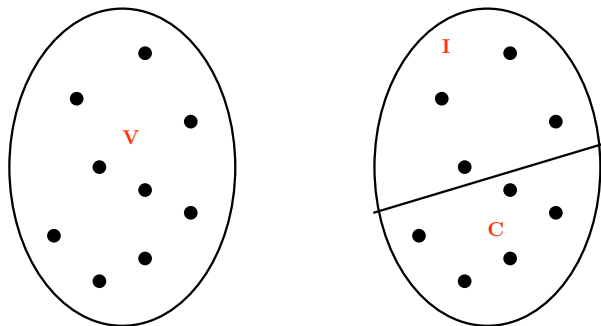


Figure: Split graph

Split graphs

A graph $G = (V, E)$ is called a **split graph** if its vertex set V can be partitioned into two sets I and C such that I is an **independent set** and C is a **clique**.

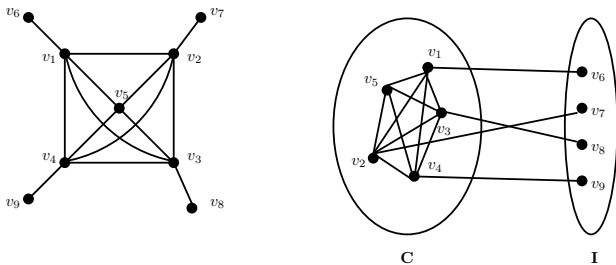


Figure: Split graph

Lemma

Let G be a split graph. If M is an acyclic matching in G , then $1 \leq |M| \leq 2$.

Lemma

Let G be a split graph. If M is an acyclic matching in G , then $1 \leq |M| \leq 2$.

Outline of proof.

- Let $\{a_1b_1, a_2b_2, a_3b_3\} \subseteq M$ for some $a_i, b_i \in V$, $1 \leq i \leq 3$.

Lemma

Let G be a split graph. If M is an acyclic matching in G , then $1 \leq |M| \leq 2$.

Outline of proof.

- Let $\{a_1b_1, a_2b_2, a_3b_3\} \subseteq M$ for some $a_i, b_i \in V$, $1 \leq i \leq 3$.
- I is an independent set $\implies b_1, b_2, b_3 \in C$.

Lemma

Let G be a split graph. If M is an acyclic matching in G , then $1 \leq |M| \leq 2$.

Outline of proof.

- Let $\{a_1b_1, a_2b_2, a_3b_3\} \subseteq M$ for some $a_i, b_i \in V$, $1 \leq i \leq 3$.
- I is an independent set $\implies b_1, b_2, b_3 \in C$.
- $G[\{b_1, b_2, b_3\}]$ forms a cycle.

Lemma

Let G be a split graph. If M is an acyclic matching in G , then $1 \leq |M| \leq 2$.

Outline of proof.

- Let $\{a_1b_1, a_2b_2, a_3b_3\} \subseteq M$ for some $a_i, b_i \in V$, $1 \leq i \leq 3$.
- I is an independent set $\implies b_1, b_2, b_3 \in C$.
- $G[\{b_1, b_2, b_3\}]$ forms a cycle.

CONTRADICTION!!

Lemma

Let G be a split graph. If M is an acyclic matching in G , then $1 \leq |M| \leq 2$.

Outline of proof.

- Let $\{a_1b_1, a_2b_2, a_3b_3\} \subseteq M$ for some $a_i, b_i \in V$, $1 \leq i \leq 3$.
- I is an independent set $\implies b_1, b_2, b_3 \in C$.
- $G[\{b_1, b_2, b_3\}]$ forms a cycle.

CONTRADICTION!!

- Thus, $|M| \leq 2$.

Lemma

Let G be a split graph and let M be a max-acy-matching in G . Then, $|M| = 2 \Leftrightarrow$ there exist a pair of vertices $v_i, v_j \in I$ s.t. $N(v_i) \setminus N(v_j) \neq \emptyset$ and $N(v_j) \setminus N(v_i) \neq \emptyset$.

Lemma

Let G be a split graph and let M be a max-acy-matching in G .
Then, $|M| = 2 \Leftrightarrow$ there exist a pair of vertices $v_i, v_j \in I$ s.t.
 $N(v_i) \setminus N(v_j) \neq \emptyset$ and $N(v_j) \setminus N(v_i) \neq \emptyset$.

Outline of proof.

\Rightarrow Let $M = \{a_i b_i, a_j b_j\}$.

Let $a_i, a_j \in C$ and $b_i, b_j \in I$.

Since $G[\{a_i, a_j, b_i, b_j\}]$ is acyclic and $a_i b_i, a_j b_j, a_i a_j \in E$.

So, $a_i \notin N(b_j)$ and $a_j \notin N(b_i)$.

Hence, $b_i, b_j \in I$ is the required pair of vertices.

Lemma

Let $G = (V, E)$ be a split graph and let M be a maximum acyclic matching in G . Then, $|M| = 2$ if and only if there exist a pair of vertices $v_i, v_j \in I$ such that $N(v_i) \setminus N(v_j) \neq \emptyset$ and $N(v_j) \setminus N(v_i) \neq \emptyset$.

Outline of proof.

\Leftarrow Let $v_1, v_2 \in I$ s.t. $N(v_1) \setminus N(v_2) \neq \emptyset$ and $N(v_2) \setminus N(v_1) \neq \emptyset$.

Let $c_1 \in N(v_1) \setminus N(v_2)$ and $c_2 \in N(v_2) \setminus N(v_1)$.

Define $M = \{v_1c_1, v_2c_2\}$.

Definition

A split graph $G = (V, E)$ is called a **threshold graph** if the vertices in I can be linearly ordered, say $(v_1, v_2, \dots, v_{|I|})$, such that $N(v_1) \subseteq N(v_2) \subseteq \dots \subseteq N(v_{|I|})$.

Definition

A split graph $G = (V, E)$ is called a **threshold graph** if the vertices in I can be linearly ordered, say $(v_1, v_2, \dots, v_{|I|})$, such that $N(v_1) \subseteq N(v_2) \subseteq \dots \subseteq N(v_{|I|})$.

Theorem

Let $G = (V, E)$ be a split graph and let M be a max-acy-matching in G . Then, $|M| = 1 \Leftrightarrow G$ is a threshold graph.

Our Results

Max-Acy-Matching-Decide is shown to be **NP-complete** for

- Comb-convex bipartite graphs
- Dually chordal graphs

Max-Acy-Matching problem is shown to be **linear time solvable** for

- Split graphs
- Proper interval graphs

• Max-Acy-Matching problem is **hard to approximate** within a ratio of $n^{1-\epsilon}$ for any $\epsilon > 0$, unless $P = NP$.

• Max-Acy-Matching problem is **APX-complete** for $2k + 1$ -regular graphs for $k \geq 3$.

Approximation hardness

Approximation hardness

$$\mu_{ac}(G) \leq \mu(G)$$

Approximation hardness

$$\mu_{ac}(G) \leq \mu(G) \leq \frac{n}{2}$$

Approximation hardness

$$\mu_{ac}(G) \leq \mu(G) \leq \frac{n}{2}$$

Observation:

Max-Acy-Matching problem **can be** approximated within a **factor of n** in **polynomial time**.

Approximation hardness

$$\mu_{ac}(G) \leq \mu(G) \leq \frac{n}{2}$$

Observation:

Max-Acy-Matching problem **can be** approximated within a **factor of n** in **polynomial time**.

To show:

Max-Acy-Matching problem **cannot** be approximated within a factor of **$n^{1-\epsilon}$ for any $\epsilon > 0$** , **unless $P = NP$** .

Theorem MAX-IS

The Max-Ind-Set problem for a graph G **cannot** be approximated within a factor of $n^{1-\epsilon}$ for any $\epsilon > 0$, unless $P = NP$.^a

^aDavid Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In Proceedings of the thirty-eighth annual ACM symposium on Theory of computing, pages 681-690, 2006.

Construction A:

Construction A:

Given: $G = (V, E)$, an instance of the Max-Ind-Set problem.

Construction A:

Given: $G = (V, E)$, an instance of the Max-Ind-Set problem.

Construct: $H = (V_H, E_H)$, an instance of the Max-Acy-Matching problem.

Construction A:

Given: $G = (V, E)$, an instance of the Max-Ind-Set problem.

Construct: $H = (V_H, E_H)$, an instance of the Max-Acy-Matching problem.

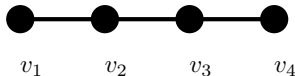
- $V_H = V \cup V'$, where $V' = \{v'_i \mid v_i \in V\}$.
- $E_H = E \cup \{v_i v'_i \mid 1 \leq i \leq n\} \cup \{v_i v'_j \mid v_i v_j \in E\} \cup \{v'_i v'_j \mid v_i v_j \in E\}$.

Construction A:

Given: $G = (V, E)$, an instance of the Max-Ind-Set problem.

Construct: $H = (V_H, E_H)$, an instance of the Max-Acy-Matching problem.

- $V_H = V \cup V'$, where $V' = \{v'_i \mid v_i \in V\}$.
- $E_H = E \cup \{v_i v'_i \mid 1 \leq i \leq n\} \cup \{v_i v'_j \mid v_i v_j \in E\} \cup \{v'_i v'_j \mid v_i v_j \in E\}$.



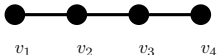
G

Construction A:

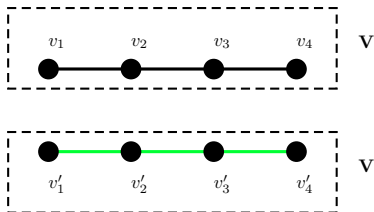
Given: $G = (V, E)$, an instance of the Max-Ind-Set problem.

Construct: $H = (V_H, E_H)$, an instance of the Max-Acy-Matching problem.

- $V_H = V \cup V'$, where $V' = \{v'_i \mid v_i \in V\}$.
- $E_H = E \cup \{v_i v'_i \mid 1 \leq i \leq n\} \cup \{v_i v'_j \mid v_i v_j \in E\} \cup \{v'_i v'_j \mid v_i v_j \in E\}$.



G



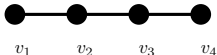
H

Construction A:

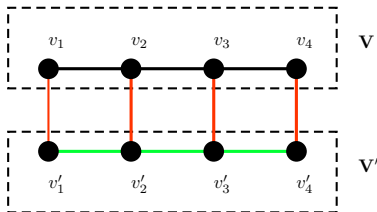
Given: $G = (V, E)$, an instance of the Max-Ind-Set problem.

Construct: $H = (V_H, E_H)$, an instance of the Max-Acy-Matching problem.

- $V_H = V \cup V'$, where $V' = \{v'_i \mid v_i \in V\}$.
- $E_H = E \cup \{v_i v'_i \mid 1 \leq i \leq n\} \cup \{v_i v'_j \mid v_i v_j \in E\} \cup \{v'_i v'_j \mid v_i v_j \in E\}$.



G



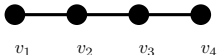
H

Construction A:

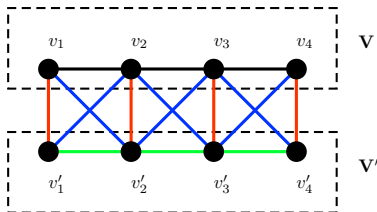
Given: $G = (V, E)$, an instance of the Max-Ind-Set problem.

Construct: $H = (V_H, E_H)$, an instance of the Max-Acy-Matching problem.

- $V_H = V \cup V'$, where $V' = \{v'_i \mid v_i \in V\}$.
- $E_H = E \cup \{v_i v'_i \mid 1 \leq i \leq n\} \cup \{\mathbf{v_i v'_j} \mid \mathbf{v_i v_j} \in \mathbf{E}\} \cup \{v'_i v'_j \mid v_i v_j \in E\}$.



G



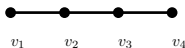
H

Claim:

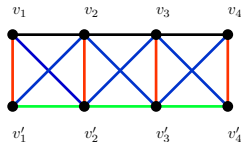
G has an independent set of size $\geq k \Leftrightarrow H$ has an acy-matching of size $\geq k$.

Types of Edges:

- ❶ $Type-I = \{v_i v'_i \mid v_i \in V \text{ and } v'_i \in V'\}.$
- ❷ $Type-II = \{v_i v_j \mid v_i, v_j \in V\}.$
- ❸ $Type-III = \{v'_i v'_j \mid v'_i \in V' \text{ and } v'_j \in V'\}.$
- ❹ $Type-IV = \{v'_i v'_j \mid v'_i, v'_j \in V'\}.$



G



H

Lemma 1

There exists a max-acy-matching in H containing edges of *Type-I* and *Type-II* only.

Lemma 1

There exists a max-acy-matching in H containing edges of *Type-I* and *Type-II* only.

Idea of proof:

Type-IV edge \rightarrow *Type-III* edge \rightarrow *Type-II* edge
 $v'_i v'_j \rightarrow v_i v'_j \text{ or } (v'_i v_j) \rightarrow v_i v_j$

Lemma 1

There exists a max-acy-matching in H containing edges of *Type-I* and *Type-II* only.

Outline of proof:

- Let $v'_i v'_j \in M$. ($v'_i v'_j$ is of *Type-IV*.)
- M is acyclic \implies both v_i and v_j are unsaturated by M .
- Let $M' = (M \setminus \{v'_i v'_j\}) \cup \{v'_i v'_j\}$.
- Assume that $G[V_H(M')]$ contains a cycle C .
- C must contain the vertex v_j .
- Let $v_j u_a, v_j u_b \in E(C)$. Since $N[v_j] = N[v'_j]$, $v'_j u_a, v'_j u_b \in E_H$. Now, $C' = (C \setminus \{v_j u_a, v_j u_b\}) \cup \{v'_j u_a, v'_j u_b\}$ is a cycle in $G[V_H(M)]$.

CONTRADICTION!!

Lemma 2

There exists a max-acy-matching in H containing edges of *Type-I* only.

Outline of proof:

Lemma 2

There exists a max-acy-matching in H containing edges of *Type-I* only.

Outline of proof:

Recall: *Type-I* = $\{v_i v'_i \mid v_i \in V \text{ and } v'_i \in V'\}$.,

Type-II = $\{v_i v_j \mid v_i, v_j \in V\}$.

Lemma 2

There exists a max-acy-matching in H containing edges of *Type-I* only.

Outline of proof:

Recall: *Type-I* = $\{v_i v'_i \mid v_i \in V \text{ and } v'_i \in V'\}$.,

Type-II = $\{v_i v_j \mid v_i, v_j \in V\}$.

In $G[V_H(M)]$, edges of *Type-I* will form a **disjoint union of K'_2 s** and edges of *Type-II* will form a **forest**.

Lemma 2

There exists a max-acy-matching in H containing edges of *Type-I* only.

Outline of proof:

Recall: *Type-I* = $\{v_i v'_i \mid v_i \in V \text{ and } v'_i \in V'\}$.,

Type-II = $\{v_i v_j \mid v_i, v_j \in V\}$.

In $G[V_H(M)]$, edges of *Type-I* will form a **disjoint union of K'_2 s** and edges of *Type-II* will form a **forest**.

\exists pendant edge of *Type-II* (say $v_a v_b$) in $G[V_H(M)]$.

Lemma 2

There exists a max-acy-matching in H containing edges of *Type-I* only.

Outline of proof:

Recall: *Type-I* = $\{v_i v'_i \mid v_i \in V \text{ and } v'_i \in V'\}$.,

Type-II = $\{v_i v_j \mid v_i, v_j \in V\}$.

In $G[V_H(M)]$, edges of *Type-I* will form a **disjoint union of K'_2 s** and edges of *Type-II* will form a **forest**.

\exists pendant edge of *Type-II* (say $v_a v_b$) in $G[V_H(M)]$.

Let $d(v_b) = 1$ in $G[V_H(M)]$.

Lemma 2

There exists a max-acy-matching in H containing edges of *Type-I* only.

Outline of proof:

Recall: *Type-I* = $\{v_i v'_i \mid v_i \in V \text{ and } v'_i \in V'\}$.,

Type-II = $\{v_i v_j \mid v_i, v_j \in V\}$.

In $G[V_H(M)]$, edges of *Type-I* will form a **disjoint union of K'_2 s** and edges of *Type-II* will form a **forest**.

\exists pendant edge of *Type-II* (say $v_a v_b$) in $G[V_H(M)]$.

Let $d(v_b) = 1$ in $G[V_H(M)]$.

Claim: $M' = (M \setminus \{v_a v_b\}) \cup \{v_b v'_b\}$ is an acyclic matching in H .

Lemma 2

There exists a max-acy-matching in H containing edges of *Type-I* only.

Outline of proof:

Recall: *Type-I* = $\{v_i v'_i \mid v_i \in V \text{ and } v'_i \in V'\}$.,

Type-II = $\{v_i v_j \mid v_i, v_j \in V\}$.

In $G[V_H(M)]$, edges of *Type-I* will form a **disjoint union of K'_2 s** and edges of *Type-II* will form a **forest**.

\exists pendant edge of *Type-II* (say $v_a v_b$) in $G[V_H(M)]$.

Let $d(v_b) = 1$ in $G[V_H(M)]$.

Claim: $M' = (M \setminus \{v_a v_b\}) \cup \{v_b v'_b\}$ is an acyclic matching in H .

If M' is not acyclic $\implies G[V_H(M')]$ contains a cycle C .

Lemma 2

There exists a max-acy-matching in H containing edges of *Type-I* only.

Outline of proof:

Recall: *Type-I* = $\{v_i v'_i \mid v_i \in V \text{ and } v'_i \in V'\}$.,

Type-II = $\{v_i v_j \mid v_i, v_j \in V\}$.

In $G[V_H(M)]$, edges of *Type-I* will form a **disjoint union of K'_2 s** and edges of *Type-II* will form a **forest**.

\exists pendant edge of *Type-II* (say $v_a v_b$) in $G[V_H(M)]$.

Let $d(v_b) = 1$ in $G[V_H(M)]$.

Claim: $M' = (M \setminus \{v_a v_b\}) \cup \{v_b v'_b\}$ is an acyclic matching in H .

If M' is not acyclic $\implies G[V_H(M')]$ contains a cycle C .

C must contain the vertex v'_b .

Lemma 2

There exists a max-acy-matching in H containing edges of *Type-I* only.

Outline of proof:

Recall: *Type-I* = $\{v_i v'_i \mid v_i \in V \text{ and } v'_i \in V'\}$.,

Type-II = $\{v_i v_j \mid v_i, v_j \in V\}$.

In $G[V_H(M)]$, edges of *Type-I* will form a **disjoint union of K'_2 s** and edges of *Type-II* will form a **forest**.

\exists pendant edge of *Type-II* (say $v_a v_b$) in $G[V_H(M)]$.

Let $d(v_b) = 1$ in $G[V_H(M)]$.

Claim: $M' = (M \setminus \{v_a v_b\}) \cup \{v_b v'_b\}$ is an acyclic matching in H .

If M' is not acyclic $\implies G[V_H(M')]$ contains a cycle C .

C must contain the vertex v'_b . Let $v'_b v_i, v'_b v_j \in E(C)$ for some $v_i, v_j \in V$.

Lemma 2

There exists a max-acy-matching in H containing edges of *Type-I* only.

Outline of proof:

Recall: *Type-I* = $\{v_i v'_i \mid v_i \in V \text{ and } v'_i \in V'\}$.,

Type-II = $\{v_i v_j \mid v_i, v_j \in V\}$.

In $G[V_H(M)]$, edges of *Type-I* will form a **disjoint union of K'_2 s** and edges of *Type-II* will form a **forest**.

\exists pendant edge of *Type-II* (say $v_a v_b$) in $G[V_H(M)]$.

Let $d(v_b) = 1$ in $G[V_H(M)]$.

Claim: $M' = (M \setminus \{v_a v_b\}) \cup \{v_b v'_b\}$ is an acyclic matching in H .

If M' is not acyclic $\implies G[V_H(M')]$ contains a cycle C .

C must contain the vertex v'_b . Let $v'_b v_i, v'_b v_j \in E(C)$ for some $v_i, v_j \in V$.

Since $N[v_b] = N[v'_b]$, $v_b v_i, v_b v_j \in E_H$ implying that $d(v_b) \geq 2$ in $G[V_H(M')]$.

Lemma 2

There exists a max-acy-matching in H containing edges of *Type-I* only.

Outline of proof:

Recall: *Type-I* = $\{v_i v'_i \mid v_i \in V \text{ and } v'_i \in V'\}$.,

Type-II = $\{v_i v_j \mid v_i, v_j \in V\}$.

In $G[V_H(M)]$, edges of *Type-I* will form a **disjoint union of K'_2 s** and edges of *Type-II* will form a **forest**.

\exists pendant edge of *Type-II* (say $v_a v_b$) in $G[V_H(M)]$.

Let $d(v_b) = 1$ in $G[V_H(M)]$.

Claim: $M' = (M \setminus \{v_a v_b\}) \cup \{v_b v'_b\}$ is an acyclic matching in H .

If M' is not acyclic $\implies G[V_H(M')]$ contains a cycle C .

C must contain the vertex v'_b . Let $v'_b v_i, v'_b v_j \in E(C)$ for some $v_i, v_j \in V$.

Since $N[v_b] = N[v'_b]$, $v_b v_i, v_b v_j \in E_H$ implying that $d(v_b) \geq 2$ in $G[V_H(M')]$.

CONTRADICTION!!!

Lemma 2

There exists a max-acy-matching in H containing edges of *Type-I* only.

Outline of proof:

Recall: $\text{Type-I} = \{v_i v'_i \mid v_i \in V \text{ and } v'_i \in V'\}.$

$\text{Type-II} = \{v_i v_j \mid v_i, v_j \in V\}.$

In $G[V_H(M)]$, edges of *Type-I* will form a **disjoint union of K'_2 s** and edges of *Type-II* will form a **forest**.

\exists pendant edge of *Type-II* (say $v_a v_b$) in $G[V_H(M)]$.

Let $d(v_b) = 1$ in $G[V_H(M)]$.

Claim: $M' = (M \setminus \{v_a v_b\}) \cup \{v_b v'_b\}$ is an acyclic matching in H .

If M' is not acyclic $\implies G[V_H(M')]$ contains a cycle C .

C must contain the vertex v'_b . Let $v'_b v_i, v'_b v_j \in E(C)$ for some $v_i, v_j \in V$.

Since $N[v_b] = N[v'_b]$, $v_b v_i, v_b v_j \in E_H$ implying that $d(v_b) \geq 2$ in $G[V_H(M')]$.

CONTRADICTION!!!

Lemma 3

G has an independent set of size $\geq k \Leftrightarrow H$ has an acy-matching of size $\geq k$.

Lemma 3

G has an independent set of size $\geq k \Leftrightarrow H$ has an acy-matching of size $\geq k$.

Outline of proof:

\Rightarrow Let $I = \{v_1, v_2, \dots, v_l\}$ be an independent set in G of size $\geq k$.

Define $M = \{v_1v'_1, v_2v'_2, \dots, v_lv'_l\}$ in H .

\Leftarrow Let $M = \{e_1, e_2, \dots, e_l\}$ be an acyclic matching in H of size at least k .

By Lemma 2, there exists a matching M' such that $|M'| = |M|$ and M' has edges of Type-I only.

Define a set $I = \{v_i \mid v_iv'_i \in M'\}$.

Lemma 3

G has an independent set of size $\geq k \Leftrightarrow H$ has an acy-matching of size $\geq k$.

Outline of proof:

\Rightarrow Let $I = \{v_1, v_2, \dots, v_l\}$ be an independent set in G of size $\geq k$.

Define $M = \{v_1v'_1, v_2v'_2, \dots, v_lv'_l\}$ in H .

\Leftarrow Let $M = \{e_1, e_2, \dots, e_l\}$ be an acyclic matching in H of size at least k .

By Lemma 2, there exists a matching M' such that $|M'| = |M|$ and M' has edges of Type-I only.

Define a set $I = \{v_i \mid v_iv'_i \in M'\}$.

Corollary 1

G has a maximum independent set of size $k \Leftrightarrow H$ has a maximum acyclic matching of size k .

Approximation hardness (continued...)

I^* = maximum independent set in G .

M^* = maximum acyclic matching in H .

Assumption:

Max-Acy-Matching problem can be approximated within a ratio $\alpha \geq 1$ by using an algorithm ALG , where $\alpha = \bar{n}^{1-\epsilon'}$ for some fixed $\epsilon' > 0$.

Note:

Here, $\bar{n} = |V_H|$.

Outline of the Proof(continued):

$$|M^*(H)| \leq \alpha |M_{ALG}(H)|. \text{ (By Assumption)}$$

Outline of the Proof(continued):

$$|M^*(H)| \leq \alpha |M_{ALG}(H)|. \text{ (By Assumption)}$$

$$|I^*(G)| = |M^*(H)|. \text{ (By Corollary 1)}$$

Outline of the Proof(continued):

$|M^*(H)| \leq \alpha |M_{ALG}(H)|$. (By Assumption)

$|I^*(G)| = |M^*(H)|$. (By Corollary 1)

There exists $I_{ALG}(G)$ such that $|M_{ALG}(H)| = |I_{ALG}(G)|$. (By Lemma 3)

Outline of the Proof(continued):

$$|M^*(H)| \leq \alpha |M_{ALG}(H)|. \text{ (By Assumption)}$$

$$|I^*(G)| = |M^*(H)|. \text{ (By Corollary 1)}$$

There exists $I_{ALG}(G)$ such that $|M_{ALG}(H)| = |I_{ALG}(G)|$. (By Lemma 3)

$$\text{Hence, } |I^*(G)| \leq \alpha |I_{ALG}(G)| = \bar{n}^{1-\epsilon'} |I_{ALG}(G)| = (2n)^{1-\epsilon'} |I_{ALG}(G)| = (2)^{1-\epsilon'} (n)^{1-\epsilon'} |I_{ALG}(G)|.$$

Outline of the Proof(continued):

$$|M^*(H)| \leq \alpha |M_{ALG}(H)|. \text{ (By Assumption)}$$

$$|I^*(G)| = |M^*(H)|. \text{ (By Corollary 1)}$$

$$\text{There exists } I_{ALG}(G) \text{ such that } |M_{ALG}(H)| = |I_{ALG}(G)|. \text{ (By Lemma 3)}$$

$$\text{Hence, } |I^*(G)| \leq \alpha |I_{ALG}(G)| = \bar{n}^{1-\epsilon'} |I_{ALG}(G)| = (2n)^{1-\epsilon'} |I_{ALG}(G)| = (2)^{1-\epsilon'} (n)^{1-\epsilon'} |I_{ALG}(G)|.$$

If we choose ϵ , such that $2^{1-\epsilon'} < n^{\epsilon'-\epsilon}$, then

$$|I^*(G)| < (n)^{\epsilon'-\epsilon} (n)^{1-\epsilon'} |I_{ALG}(G)| = (n)^{1-\epsilon} |I_{ALG}(G)|.$$

Outline of the Proof(continued):

$|M^*(H)| \leq \alpha |M_{ALG}(H)|$. (By Assumption)

$|I^*(G)| = |M^*(H)|$. (By Corollary 1)

There exists $I_{ALG}(G)$ such that $|M_{ALG}(H)| = |I_{ALG}(G)|$. (By Lemma 3)

Hence, $|I^*(G)| \leq \alpha |I_{ALG}(G)| = \bar{n}^{1-\epsilon'} |I_{ALG}(G)| = (2n)^{1-\epsilon'} |I_{ALG}(G)| = (2)^{1-\epsilon'} (n)^{1-\epsilon'} |I_{ALG}(G)|$.

If we choose ϵ , such that $2^{1-\epsilon'} < n^{\epsilon'-\epsilon}$, then
 $|I^*(G)| < (n)^{\epsilon'-\epsilon} (n)^{1-\epsilon'} |I_{ALG}(G)| = (n)^{1-\epsilon} |I_{ALG}(G)|$.

Hence, $|I^*(G)| < (n)^{1-\epsilon} |I_{ALG}(G)|$, which leads to a contradiction to Theorem Max-IS.

Outline of the Proof(continued):

$|M^*(H)| \leq \alpha |M_{ALG}(H)|$. (By Assumption)

$|I^*(G)| = |M^*(H)|$. (By Corollary 1)

There exists $I_{ALG}(G)$ such that $|M_{ALG}(H)| = |I_{ALG}(G)|$. (By Lemma 3)

Hence, $|I^*(G)| \leq \alpha |I_{ALG}(G)| = \bar{n}^{1-\epsilon'} |I_{ALG}(G)| = (2n)^{1-\epsilon'} |I_{ALG}(G)| = (2)^{1-\epsilon'} (n)^{1-\epsilon'} |I_{ALG}(G)|$.

If we choose ϵ , such that $2^{1-\epsilon'} < n^{\epsilon'-\epsilon}$, then
 $|I^*(G)| < (n)^{\epsilon'-\epsilon} (n)^{1-\epsilon'} |I_{ALG}(G)| = (n)^{1-\epsilon} |I_{ALG}(G)|$.

Hence, $|I^*(G)| < (n)^{1-\epsilon} |I_{ALG}(G)|$, which leads to a contradiction to Theorem Max-IS.

Therefore, the Max-Acy-Matching problem cannot be approximated within a factor of $n^{1-\epsilon}$ for any $\epsilon > 0$, unless $P = NP$.

APX-completeness

Lemma 4

The Max-Acy-Matching problem for a k -regular graph G can be approximated with an approximation ratio of $\frac{2k(k-1)+1}{k}$, where k is a constant.

Outline of proof:

Pick an edge in M_{ac} and remove its end points

In each step, we are removing at most k^2 edges.

So, $\frac{kn}{2[2k(k-1)+1]} \leq |M_{ac}|$.

Also, $\mu_{ac} \leq \frac{n}{2}$.

Hence, $\frac{\mu_{ac}}{|M_{ac}|} \leq \frac{2k(k-1)+1}{k}$.

Theorem

^a ^b The Max-Ind-Set problem is APX-complete for k -regular graphs for $k \geq 3$.

^aPaola Alimonti and Viggo Kann. Some apx-completeness results for cubic graphs. Theoretical Computer Science, 237(1-2):123-134, 2000

^bChristos H Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. Journal of computer and system sciences, 43(3):425-440, 1991.

Observation

If G is a k -regular graph in Construction A, then the constructed graph H is a $2k + 1$ -regular graph for $k \geq 3$.

Theorem

The Max-Acy-Matching problem is APX-complete for $2k + 1$ -regular graphs for $k \geq 3$, where k is a constant.

Outline of proof:

Theorem

The Max-Acy-Matching problem is APX-complete for $2k + 1$ -regular graphs for $k \geq 3$, where k is a constant.

Outline of proof:

The Max-Acy-Matching problem for $2k + 1$ -regular graphs for $k \geq 3$ belongs to APX by Lemma 4.

Theorem

The Max-Acy-Matching problem is APX-complete for $2k + 1$ -regular graphs for $k \geq 3$, where k is a constant.

Outline of proof:

The Max-Acy-Matching problem for $2k + 1$ -regular graphs for $k \geq 3$ belongs to APX by Lemma 4.

Given a k -regular graph $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$.

Theorem

The Max-Acy-Matching problem is APX-complete for $2k + 1$ -regular graphs for $k \geq 3$, where k is a constant.

Outline of proof:

The Max-Acy-Matching problem for $2k + 1$ -regular graphs for $k \geq 3$ belongs to APX by Lemma 4.

Given a k -regular graph $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$.

Construct a graph $H = (V_H, E_H)$, an instance of the Max-Acy-Matching problem by Construction A.

Theorem

The Max-Acy-Matching problem is APX-complete for $2k + 1$ -regular graphs for $k \geq 3$, where k is a constant.

Outline of proof:

The Max-Acy-Matching problem for $2k + 1$ -regular graphs for $k \geq 3$ belongs to APX by Lemma 4.

Given a k -regular graph $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$.

Construct a graph $H = (V_H, E_H)$, an instance of the Max-Acy-Matching problem by Construction A.

By Lemma 3 and Corollary 1, the reduction described in Construction A is an L-reduction with $\alpha = 1$ and $\beta = 1$.

Theorem

The Max-Acy-Matching problem is APX-complete for $2k + 1$ -regular graphs for $k \geq 3$, where k is a constant.

Outline of proof:

The Max-Acy-Matching problem for $2k + 1$ -regular graphs for $k \geq 3$ belongs to APX by Lemma 4.

Given a k -regular graph $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$.

Construct a graph $H = (V_H, E_H)$, an instance of the Max-Acy-Matching problem by Construction A.

By Lemma 3 and Corollary 1, the reduction described in Construction A is an L-reduction with $\alpha = 1$ and $\beta = 1$.

Hence, proved.

Future Scope

- Better approximation algorithms for bipartite graphs.
- Parameterized complexity.

References



Andreas Brandstädt, Christian Hundt, and Ragnar Nevries.

Efficient edge domination on hole-free graphs in polynomial time.

In *Latin American Symposium on Theoretical Informatics*, pages 650–661. Springer, 2010.



Andreas Brandstädt, Arne Leitert, and Dieter Rautenbach.

Efficient dominating and edge dominating sets for graphs and hypergraphs.

In *International Symposium on Algorithms and Computation*, pages 267–277. Springer, 2012.



Kathie Cameron.

Induced matchings.

Discrete Applied Mathematics, 24(1-3):97–102, 1989.



Jack Edmonds.

Maximum matching and a polyhedron with 0, 1-vertices.

Journal of Research of the National Bureau of Standards B, 69(125-130):55–56, 1965.



Martin Charles Golumbic and Clinton F Goss.

Perfect elimination and chordal bipartite graphs.

Journal of Graph Theory, 2(2):155–163, 1978.



Dana L Grinstead, Peter J Slater, Naveed A Sherwani, and Nancy D Holmes.

Efficient edge domination problems in graphs.

Information Processing Letters, 48(5):221–228, 1993.

References



Nicholas Korpelainen.

A polynomial-time algorithm for the dominating induced matching problem in the class of convex graphs.

Electronic Notes in Discrete Mathematics, 32:133–140, 2009.



Marilynn Livingston and Quentin F Stout.

Distributing resources in hypercube computers.

In *Proceedings of the third conference on Hypercube concurrent computers and applications: Architecture, software, computer systems, and general issues-Volume 1*, pages 222–231. ACM, 1988.



Chin Lung Lu, Ming-Tat Ko, and Chuan Yi Tang.

Perfect edge domination and efficient edge domination in graphs.

Discrete Applied Mathematics, 119(3):227–250, 2002.



Chin Lung Lu and Chuan Yi Tang.

Solving the weighted efficient edge domination problem on bipartite permutation graphs.

Discrete Applied Mathematics, 87(1-3):203–211, 1998.

Thank You!!

Questions??