Minimal forbidden induced subgraphs characterization of Block $B_0$-VPG graphs

L. Alcón  F. Bonomo  M. P. Mazzoleni

Dto. de Matemática, FCE-UNLP, La Plata, Argentina
Dto. de Computación FCEN-UBA, Buenos Aires, Argentina

Latin American Workshop on Cliques in Graphs 2016
A **VPG representation** of a graph $G$ is a collection of paths on a grid where the paths represent the vertices of $G$ such that:

two vertices of $G$ are adjacent in $G$ if and only if the corresponding paths share at least one vertex of the grid.

A graph which has a VPG representation is called a **VPG graph**.
Introduction:

A VPG representation of a graph $G$ is a collection of paths on a grid where the paths represent the vertices of $G$ such that:

- two vertices of $G$ are adjacent in $G$ if and only if the corresponding paths share at least one vertex of the grid.

A graph which has a VPG representation is called a VPG graph.
The subclasses in which the number of bends (turns on a grid point) of each path is at most $k$, known as $B_k$-VPG, have been widely studied.

Figure: A graph $G$ and a $B_2$-VPG representation of $G$. 

In this work, we consider the subclass $B_0$-VPG.

A $B_0$-representation of $G$ is a VPG representation in which each path in the representation is either a horizontal path or a vertical path on the grid.

A graph is $B_0$-VPG if it has a $B_0$-VPG representation.

The recognition problem is NP-complete for $B_0$-VPG. Although, there exists a polynomial time algorithm for deciding whether a given Chordal graph is $B_0$-VPG [5].
In this work, we consider the subclass \( B_0 \)-VPG.

A \( B_0 \)-representation of \( G \) is a VPG representation in which each path in the representation is either a horizontal path or a vertical path on the grid.

A graph is \( B_0 \)-VPG if it has a \( B_0 \)-VPG representation.

The recognition problem is \textbf{NP}-complete for \( B_0 \)-VPG. Although, there exists a \textbf{polynomial time} algorithm for deciding whether a given Chordal graph is \( B_0 \)-VPG \cite{5}.
Golumbic and Ries gave a characterization by forbidden induced subgraphs for the following three subclasses of Chordal $B_0$-VPG:

✓ Split $B_0$-VPG,
✓ Chordal $B_0$-VPG bull free,
✓ Chordal $B_0$-VPG claw free.

Moreover, for the class Split $B_0$-VPG this yields a linear time recognition algorithm.
Golumbic and Ries gave a characterization by forbidden induced subgraphs for the following three subclasses of Chordal $B_0$-VPG:

✓ Split $B_0$-VPG,
✓ Chordal $B_0$-VPG bull free,
✓ Chordal $B_0$-VPG claw free.

Moreover, for the class Split $B_0$-VPG this yields a linear time recognition algorithm.
In this work we study another subclass of Chordal $B_0$-VPG: Block $B_0$-VPG.

We present a minimal forbidden induced subgraphs characterization of this class.

Moreover, the proof of the main theorem provides an alternative recognition algorithm for $B_0$-VPG in the class of block graphs.
In this work we study another subclass of Chordal $B_0$-VPG: Block $B_0$-VPG.

We present a minimal forbidden induced subgraphs characterization of this class.

Moreover, the proof of the main theorem provides an alternative recognition algorithm for $B_0$-VPG in the class of block graphs.
In this work we study another subclass of Chordal $B_0$-VPG: Block $B_0$-VPG.

We present a minimal forbidden induced subgraphs characterization of this class.

Moreover, the proof of the main theorem provides an alternative recognition algorithm for $B_0$-VPG in the class of block graphs.
The following lemma is very important to obtain our results:

**Lemma (Golumbic et al.)**

*In a $B_0$-VPG representation of a clique, all the corresponding paths share a common grid point.*

---

**Figure:** A line clique and a cross clique.
Block graphs:

A block graph is a connected graph in which every two-connected component (block) is a clique.

A graph is Chordal if it does not contain any chordless cycle of length at least four.
Block graphs:

A block graph is a connected graph in which every two-connected component (block) is a clique.

A graph is Chordal if it does not contain any chordless cycle of length at least four.
A **diamond** is a graph obtained from $K_4$ by deleting exactly one edge.

![Diagram of a diamond](image)

**Figure:** Diamond.

Block graphs are connected chordal diamond-free graphs.
A **diamond** is a graph obtained from $K_4$ by deleting exactly one edge.

![Diamond](image)

**Figure:** Diamond.

* Block graphs are connected chordal diamond-free graphs.
A **cutpoint** is a vertex whose removal from the graph increases the number of connected components.

An **endblock** is a block having exactly one cutpoint.

An **almost endblock** is a block $B$ having at least two cutpoints and such that exactly one of these cutpoints belongs to blocks (different from $B$) that are not endblocks.

An **internal block** is a block that is neither an endblock nor an almost endblock.
A **cutpoint** is a vertex whose removal from the graph increases the number of connected components.

An **endblock** is a block having exactly one cutpoint.

An **almost endblock** is a block $B$ having at least two cutpoints and such that exactly one of these cutpoints belongs to blocks (different from $B$) that are not endblocks.

An **internal block** is a block that is neither an endblock nor an almost endblock.
A **cutpoint** is a vertex whose removal from the graph increases the number of connected components.

An **endblock** is a block having exactly one cutpoint.

An **almost endblock** is a block $B$ having at least two cutpoints and such that exactly one of these cutpoints belongs to blocks (different from $B$) that are not endblocks.

An **internal block** is a block that is neither an endblock nor an almost endblock.
A **cutpoint** is a vertex whose removal from the graph increases the number of connected components.

An **endblock** is a block having exactly one cutpoint.

An **almost endblock** is a block $B$ having at least two cutpoints and such that exactly one of these cutpoints belongs to blocks (different from $B$) that are not endblocks.

An **internal block** is a block that is neither an endblock nor an almost endblock.
**Figure**: Endblock=red, almost endbloque=blue, internal block=green.
A 3-cutpoint is a cutpoint that belongs to exactly three blocks.

A 2-cutpoint is a cutpoint that belongs to exactly two blocks, one of which is an endblock.
A **3-cutpoint** is a cutpoint that belongs to exactly three blocks.

A **2-cutpoint** is a cutpoint that belongs to exactly two blocks, one of which is an endblock.

**Figure:** 3-cutpoint=viiolet, 2-cutpoint=red.
The block-cutpoint-tree of a graph $G$, denoted $bc(G)$, is a graph such that:

- its vertices are in one-to-one correspondence with the blocks and cutpoints of $G$;
- two vertices of $bc(G)$ are adjacent if and only if one corresponds to a block $H$ of $G$ and the other to a cutpoint $c$ of $G$, and $c \in H$. 
The block-cutpoint-tree of a graph $G$, denoted $bc(G)$, is a graph such that:

- its vertices are in one-to-one correspondence with the blocks and cutpoints of $G$;

- two vertices of $bc(G)$ are adjacent if and only if one corresponds to a block $H$ of $G$ and the other to a cutpoint $c$ of $G$, and $c \in H$. 

The block-cutpoint-tree of a graph $G$, denoted $bc(G)$, is a graph such that:

- its vertices are in one-to-one correspondence with the blocks and cutpoints of $G$;

- two vertices of $bc(G)$ are adjacent if and only if one corresponds to a block $H$ of $G$ and the other to a cutpoint $c$ of $G$, and $c \in H$. 

Figure: A graph $G$ and its block-cutpoint-tree.
A thin spider $N_n$ is the graph whose $2n$ vertices can be partitioned into a clique $K = \{c_1, .., c_n\}$ and a stable set $S = \{s_1, .., s_n\}$ such that $s_i \sim c_j$ if and only if $i = j$.

We say that $N_n$ is a thin spider of size $n$.

**Figure:** The thin spider $N_5$.

Golumbic and Ries proved that $N_5 \notin B_0$-VPG.
A thin spider $N_n$ is the graph whose $2n$ vertices can be partitioned into a clique $K = \{c_1, \ldots, c_n\}$ and a stable set $S = \{s_1, \ldots, s_n\}$ such that $s_i \sim c_j$ if and only if $i = j$.

We say that $N_n$ is a thin spider of size $n$.

![Figure: The thin spider $N_5$.](image)

Golumbic and Ries proved that $N_5 \notin B_0$-VPG.
Let $\mathcal{F}$ denote the family of block graphs obtained from $N_5$ by applying the following procedure:

- consider a complete subgraph of size 4 having at least two 2-cutpoints, say $v_1$ and $v_2$, with endblocks $B_1$ and $B_2$, respectively;

- contract $v_1$ and $v_2$ into a single vertex $x$;

- replace $B_1 - \{x\}$ and $B_2 - \{x\}$ by two thin spiders of size 3, making $x$ adjacent to the vertices of the cliques of both the spiders.
Let \( F \) denote the family of block graphs obtained from \( N_5 \) by applying the following procedure:

★ consider a complete subgraph of size 4 having at least two 2-cutpoints, say \( v_1 \) and \( v_2 \), with endblocks \( B_1 \) and \( B_2 \), respectively;

★ contract \( v_1 \) and \( v_2 \) into a single vertex \( x \);

★ replace \( B_1 - \{ x \} \) and \( B_2 - \{ x \} \) by two thin spiders of size 3, making \( x \) adjacent to the vertices of the cliques of both the spiders.
Let $\mathcal{F}$ denote the family of block graphs obtained from $N_5$ by applying the following procedure:

- consider a complete subgraph of size 4 having at least two 2-cutpoints, say $v_1$ and $v_2$, with endblocks $B_1$ and $B_2$, respectively;

- contract $v_1$ and $v_2$ into a single vertex $x$;

- replace $B_1 - \{x\}$ and $B_2 - \{x\}$ by two thin spiders of size 3, making $x$ adjacent to the vertices of the cliques of both the spiders.
Let $\mathcal{F}$ denote the family of block graphs obtained from $N_5$ by applying the following procedure:

★ consider a complete subgraph of size 4 having at least two 2-cutpoints, say $v_1$ y $v_2$, with endblocks $B_1$ and $B_2$, respectively;

★ contract $v_1$ and $v_2$ into a single vertex $x$;

★ replace $B_1 - \{x\}$ y $B_2 - \{x\}$ by two thin spiders of size 3, making $x$ adjacent to the vertices of the cliques of both the spiders.
In the following figure we offer two examples of graphs in $\mathcal{F}$. 

\begin{tikzpicture}
  \node at (0,0) [circle,fill,inner sep=1.5pt] (v1) {}; \node at (1,0) [circle,fill,inner sep=1.5pt] (v2) {}; \node at (0,1) [circle,fill,inner sep=1.5pt] (v3) {\(B_1\)}; \node at (1,1) [circle,fill,inner sep=1.5pt] (v4) {}; \node at (0,2) [circle,fill,inner sep=1.5pt] (v5) {}; \node at (1,2) [circle,fill,inner sep=1.5pt] (v6) {\(B_2\)};
  \draw (v1) -- (v2) -- (v3) -- (v4) -- (v1) -- (v5) -- (v6) -- (v4) -- (v6) -- (v5);
  \draw [red] (v1) -- (v6);
\end{tikzpicture}

\begin{tikzpicture}
  \node at (0,0) [circle,fill,inner sep=1.5pt] (x) {}; \node at (1,0) [circle,fill,inner sep=1.5pt] (n1) {}; \node at (2,0) [circle,fill,inner sep=1.5pt] (n2) {\(N_3\)}; \node at (0,1) [circle,fill,inner sep=1.5pt] (n3) {\(N_3\)}; \node at (1,1) [circle,fill,inner sep=1.5pt] (n4) {\(x\)};
  \draw (n1) -- (x) -- (n2) -- (n3) -- (x) -- (n4) -- (n1) -- (n4) -- (n2) -- (n3) -- (n4);
\end{tikzpicture}
More examples of graphs in $\mathcal{F}$. 

(a) (b) (c) (d)
**Proposition**

*The family $\mathcal{F}$ is infinite.*

**Corollary**

*Each graph in $\mathcal{F}$ is minimal, i.e., it does not contain another graph in $\mathcal{F}$ as induced subgraph.*

**Lemma**

*The graphs of $\mathcal{F}$ are not $B_0$-VPG graphs.*
Proposition

The family $\mathcal{F}$ is infinite.

Corollary

Each graph in $\mathcal{F}$ is minimal, i.e., it does not contain another graph in $\mathcal{F}$ as induced subgraph.

Lemma

The graphs of $\mathcal{F}$ are not $B_0$-VPG graphs.
Proposition

The family $\mathcal{F}$ is infinite.

Corollary

Each graph in $\mathcal{F}$ is minimal, i.e., it does not contain another graph in $\mathcal{F}$ as induced subgraph.

Lemma

The graphs of $\mathcal{F}$ are not $B_0$-VPG graphs.
The following theorem allows us to determine whether a Block VPG graph is $B_0$-VPG in terms of minimal forbidden induced subgraphs.

**Theorem**

*Let $G$ be a Block VPG graph. Then $G$ is $B_0$-VPG if and only if $G$ is $\mathcal{F}$-free.*

*Moreover, the graphs of $\mathcal{F}$ are minimal not $B_0$-VPG.*
Bibliography:


Thank you !!!
Gracias a todos por venir!!
¡Esperamos verlos pronto!