A LINEAR-TIME ALGORITHM FOR THE IDENTIFYING CODE PROBLEM ON BLOCK GRAPHS

G. Argiroffo[1] S. Bianchi[1] Y. Lucarini[1,2] A. Wagler[3]

[1] Universidad Nacional de Rosario, Dept. de Matemática
Rosario, Argentina

[2] CONICET
Argentina

[3] Université Clermont Auvergne (LIMOS, UMR 6158 CNRS)
Clermont-Ferrand, France

IX Latin-American Algorithms, Graphs and Optimization Symposium
Marseille, France, September 11 - 15, 2017

This work was partially supported by PICT 2013-0586 ANPCyT Argentina.

Outline

Outline

1. Identifying codes in graphs
2. The identifying code problem on block graphs
3. Concluding Remarks

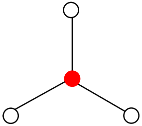
Let $G = (V, E)$ be a graph and $N[i]$ be the closed neighborhood of $i \in V$.

Let $G = (V, E)$ be a graph and $N[i]$ be the closed neighborhood of $i \in V$.

DEFINITION

A subset $C \subseteq V$ is

- dominating if $N[i] \cap C$ are non-empty sets for all $i \in V$,



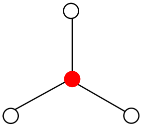
IDENTIFYING CODES IN GRAPHS

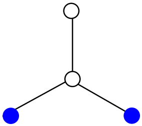
Let $G = (V, E)$ be a graph and $N[i]$ be the closed neighborhood of $i \in V$.

DEFINITION

A subset $C \subseteq V$ is

- dominating if $N[i] \cap C$ are non-empty sets for all $i \in V$,
- identifying if $N[i] \cap C$ are distinct sets for all $i \in V$,



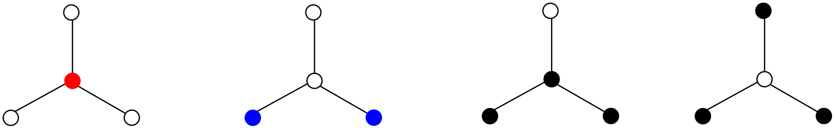


Let $G = (V, E)$ be a graph and $N[i]$ be the closed neighborhood of $i \in V$.

DEFINITION

A subset $C \subseteq V$ is

- dominating if $N[i] \cap C$ are non-empty sets for all $i \in V$,
- identifying if $N[i] \cap C$ are distinct sets for all $i \in V$,
- an **identifying code** if it is dominating and identifying.

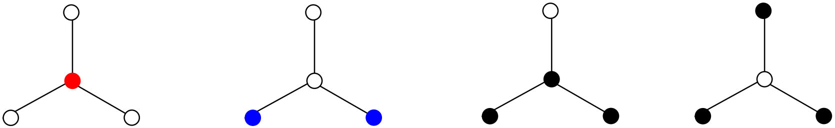


Let $G = (V, E)$ be a graph and $N[i]$ be the closed neighborhood of $i \in V$.

DEFINITION

A subset $C \subseteq V$ is

- dominating if $N[i] \cap C$ are non-empty sets for all $i \in V$,
- identifying if $N[i] \cap C$ are distinct sets for all $i \in V$,
- an **identifying code** if it is dominating and identifying.



REMARK

G is **identifiable** if and only if it has no true twins, i.e., two nodes $i \neq j$ with $N[i] = N[j]$ [Karpovsky et al. 1998].

THE IDENTIFYING CODE PROBLEM

The identifying code problem is hard in general and even remains hard for:

- bipartite graphs [Charon et al. 2003],
- interval graphs [Foucaud 2013],
- split graphs [Foucaud 2013].

The identifying code problem

The identifying code problem is hard in general and even remains hard for:

- bipartite graphs [Charon et al. 2003],
- interval graphs [Foucaud 2013],
- split graphs [Foucaud 2013].

Our aim:

Study the identifying code problem of **block graphs**.

A *block* graph is a graph in which every maximal 2-connected subgraph (block) is a clique.

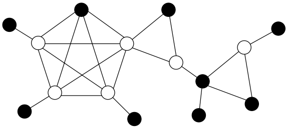
The identifying code problem is hard in general and even remains hard for:

- bipartite graphs [Charon et al. 2003],
- interval graphs [Foucaud 2013],
- split graphs [Foucaud 2013].

OUR AIM:

Study the identifying code problem of **block graphs**.

A *block* graph is a graph in which every maximal 2-connected subgraph (block) is a clique.



A block graph B (the black vertices form an identifying code of B).

OUTLINE

Previous work on the identifying code problem for trees

Known results

- A tree is a particular case of a block graph.
- Trees have been addressed in the context of the identifying code problem [Auger 2014, Bertrand et al. 2005, Blidia et al. 2007, Karpovsky et al. 1998].
- There is a linear-time algorithm that solves the identifying code problem on trees [Auger 2014].

Let $G = (V, E)$ be a graph and $v \in V$.

$C \subseteq V$ is a $\{v\}$-almost ID of G if the sets $C \cap N[u]$ are nonempty and pairwise distinct for all $u \in V - \{v\}$. [Auger 2014]

Let $G = (V, E)$ be a graph and $v \in V$.

$C \subseteq V$ is a $\{v\}$-almost ID of G if the sets $C \cap N[u]$ are nonempty and pairwise distinct for all $u \in V - \{v\}$. [Auger 2014]

C satisfies the property

- ID (for identifying) if C is an identifying code in G,

Let $G = (V, E)$ be a graph and $v \in V$.

$C \subseteq V$ is a $\{v\}$-almost ID of G if the sets $C \cap N[u]$ are nonempty and pairwise distinct for all $u \in V - \{v\}$. [Auger 2014]

C satisfies the property

- ID (for identifying) if C is an identifying code in G,
- CO (for code) if $v \in C$,

Let $G = (V, E)$ be a graph and $v \in V$.

$C \subseteq V$ is a $\{v\}$-almost ID of G if the sets $C \cap N[u]$ are nonempty and pairwise distinct for all $u \in V - \{v\}$. [Auger 2014]

C satisfies the property

- ID (for identifying) if C is an identifying code in G,
- CO (for code) if $v \in C$,
- ADJ (for adjacent) if v has a neighbour in C,

Let $G = (V, E)$ be a graph and $v \in V$.

$C \subseteq V$ is a $\{v\}$-almost ID of G if the sets $C \cap N[u]$ are nonempty and pairwise distinct for all $u \in V - \{v\}$. [Auger 2014]

C satisfies the property

- ID (for identifying) if C is an identifying code in G,
- CO (for code) if $v \in C$,
- ADJ (for adjacent) if v has a neighbour in C,
- FN (for favoured neighbour) if v has a neighbour w with $N[w] \cap C = \{v\}$.

Let $G = (V, E)$ be a graph and $v \in V$.

$C \subseteq V$ is a $\{v\}$-almost ID of G if the sets $C \cap N[u]$ are nonempty and pairwise distinct for all $u \in V - \{v\}$. [Auger 2014]

C satisfies the property

- ID (for identifying) if C is an identifying code in G,
- CO (for code) if $v \in C$,
- ADJ (for adjacent) if v has a neighbour in C,
- FN (for favoured neighbour) if v has a neighbour w with $N[w] \cap C = \{v\}$.

Let call P either any of the properties above or $\overline{\text{ID}}$, $\overline{\text{CO}}$, $\overline{\text{ADJ}}$ and $\overline{\text{FN}}$.

There exist dependence relationships between the properties on a code C:

There exist dependence relationships between the properties on a code C:

- If C satisfies FN then C satisfies CO.

There exist dependence relationships between the properties on a code C:

- If C satisfies FN then C satisfies CO.
- If C satisfies ID then C satisfies CO or ADJ.

There exist dependence relationships between the properties on a code C:

- If C satisfies FN then C satisfies CO.
- If C satisfies ID then C satisfies CO or ADJ.
- If C satisfies ID, CO and FN then C satisfies ADJ.

There exist dependence relationships between the properties on a code C:

- If C satisfies FN then C satisfies CO.
- If C satisfies ID then C satisfies CO or ADJ.
- If C satisfies ID, CO and FN then C satisfies ADJ.

Let $\Gamma_{\mathsf{P}_1,\ldots,\mathsf{P}_k}(v,G)$ the function that returns the minimum size of a $\{v\}$-almost ID code in G satisfying P_i with $i = 1,\ldots,k$ or ∞ if no such code exists.

There exist dependence relationships between the properties on a code C:

- If C satisfies FN then C satisfies CO.
- If C satisfies ID then C satisfies CO or ADJ.
- If C satisfies ID, CO and FN then C satisfies ADJ.

Let $\Gamma_{\mathsf{P}_1,\ldots,\mathsf{P}_k}(v,G)$ the function that returns the minimum size of a $\{v\}$-almost ID code in G satisfying P_i with $i=1,\ldots,k$ or ∞ if no such code exists.
It can be proved that for any graph G and $v \in V(G)$ it holds:

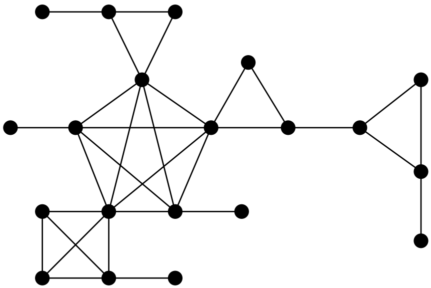
$$\Gamma_{\mathsf{ID}}(G) = min \begin{cases} \Gamma_{\mathsf{ID,CO,ADJ,FN}}(v,G) \\ \Gamma_{\mathsf{ID,CO,ADJ,\overline{FN}}}(v,G) \\ \Gamma_{\mathsf{ID,CO,\overline{ADJ}}}(v,G) \\ \Gamma_{\mathsf{ID,\overline{CO},ADJ}}(v,G) \end{cases}$$

REMARK

- A block graph B is identifiable if and only if each maximal clique K of B satisfies that all vertices in K, except at most one, have a neighbour that is not in $V(K)$.

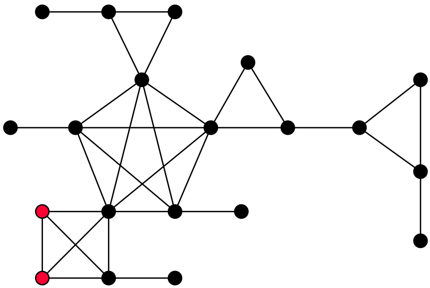
REMARK

- A block graph B is identifiable if and only if each maximal clique K of B satisfies that all vertices in K, except at most one, have a neighbour that is not in $V(K)$.



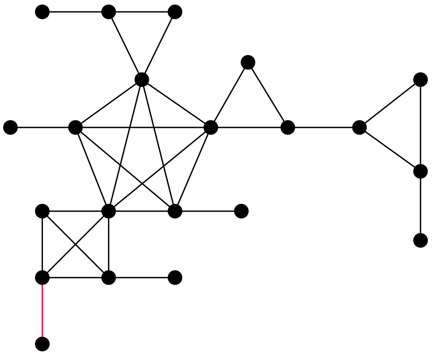
REMARK

- A block graph B is identifiable if and only if each maximal clique K of B satisfies that all vertices in K, except at most one, have a neighbour that is not in $V(K)$.



REMARK

- A block graph B is identifiable if and only if each maximal clique K of B satisfies that all vertices in K, except at most one, have a neighbour that is not in $V(K)$.

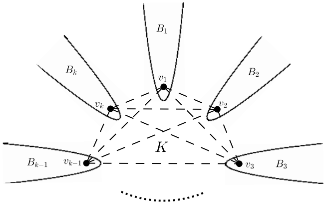


REMARK

- A block graph B is identifiable if and only if each maximal clique K of B satisfies that all vertices in K, except at most one, have a neighbour that is not in $V(K)$.
- If B is a block graph, $v_1 \in V(B)$ and K is a maximal clique with $V(K) = \{v_1, v_2, \ldots, v_k\}$ then if we delete all the edges in K we obtain k block subgraphs, say $B_1, B_2, \ldots, B_k$ containing $v_1, v_2, \ldots, v_k$ respectively.

REMARK

- A block graph B is identifiable if and only if each maximal clique K of B satisfies that all vertices in K, except at most one, have a neighbour that is not in $V(K)$.
- If B is a block graph, $v_1 \in V(B)$ and K is a maximal clique with $V(K) = \{v_1, v_2, \ldots, v_k\}$ then if we delete all the edges in K we obtain k block subgraphs, say $B_1, B_2, \ldots, B_k$ containing $v_1, v_2, \ldots, v_k$ respectively.



THE IDENTIFYING CODE PROBLEM ON BLOCK GRAPHS

THEOREM

Let B be a block graph and $v_1, v_2, \ldots, v_k$ the vertices of the maximal clique K. Let $B_1, B_2, \ldots, B_k$ be the block graphs, containing $v_1, v_2, \ldots, v_k$ respectively, obtained from B by deletion of the edges in K. Let C be a code in B and $C_i = C \cap V(B_i)$ for all $i \in \{1, 2, \ldots, k\}$.

THEOREM

Let B be a block graph and $v_1, v_2, \ldots, v_k$ the vertices of the maximal clique K. Let $B_1, B_2, \ldots, B_k$ be the block graphs, containing $v_1, v_2, \ldots, v_k$ respectively, obtained from B by deletion of the edges in K. Let C be a code in B and $C_i = C \cap V(B_i)$ for all $i \in \{1, 2, \ldots, k\}$.

- *If C is a v_1-almost ID code in B then C_i is a v_i-almost ID code in B_i for all $i \in \{1, 2, \ldots, k\}$.*

THEOREM

Let B be a block graph and $v_1, v_2, \ldots, v_k$ the vertices of the maximal clique K. Let $B_1, B_2, \ldots, B_k$ be the block graphs, containing $v_1, v_2, \ldots, v_k$ respectively, obtained from B by deletion of the edges in K. Let C be a code in B and $C_i = C \cap V(B_i)$ for all $i \in \{1, 2, \ldots, k\}$.

- *If C is a v_1-almost ID code in B then C_i is a v_i-almost ID code in B_i for all $i \in \{1, 2, \ldots, k\}$.*
- *If C satisfies ID then there exists at most one $i \in \{1, 2, \ldots, k\}$ such that C_i satisfies $\overline{\text{ADJ}}$.*

THEOREM

Let B be a block graph and $v_1, v_2, \ldots, v_k$ the vertices of the maximal clique K. Let $B_1, B_2, \ldots, B_k$ be the block graphs, containing $v_1, v_2, \ldots, v_k$ respectively, obtained from B by deletion of the edges in K. Let C be a code in B and $C_i = C \cap V(B_i)$ for all $i \in \{1, 2, \ldots, k\}$.

- *If C is a v_1-almost ID code in B then C_i is a v_i-almost ID code in B_i for all $i \in \{1, 2, \ldots, k\}$.*
- *If C satisfies ID then there exists at most one $i \in \{1, 2, \ldots, k\}$ such that C_i satisfies $\overline{\text{ADJ}}$.*
- *If C is a v_1-almost ID code in B then there exists at most one $i \in \{2, \ldots, k\}$ such that C_i satisfies $\overline{\text{ADJ}}$.*

THEOREM

Let B be a block graph, K a maximal clique $V(K) = \{v_1, v_2, \ldots, v_k\}$. Let C_i be a v_i-almost ID code in B_i, $\forall\, i \in \{1, 2, \ldots, k\}$ and $C = \bigcup_{i=1}^{k} C_i$, then

THEOREM

Let B be a block graph, K a maximal clique $V(K) = \{v_1, v_2, \ldots, v_k\}$. Let C_i be a v_i-almost ID code in B_i, $\forall\, i \in \{1, 2, \ldots, k\}$ and $C = \bigcup_{i=1}^{k} C_i$, then

- *If $u, v \in V' = V(B) - V(K)$, they are dominated and separated by C.*

THEOREM

Let B be a block graph, K a maximal clique $V(K) = \{v_1, v_2, \ldots, v_k\}$. Let C_i be a v_i-almost ID code in B_i, $\forall i \in \{1, 2, \ldots, k\}$ and $C = \bigcup_{i=1}^{k} C_i$, then

- *If $u, v \in V' = V(B) - V(K)$, they are dominated and separated by C.*
- *Let $v \in V'$ and $v_j \in V(K)$ such that $d(v, v_j) = 1$. Then v and v_j are dominated and separated by C if there is $i \in \{1, 2, \ldots, k\}$ $i \neq j$ such that C_i is CO.*

THEOREM

Let B be a block graph, K a maximal clique $V(K) = \{v_1, v_2, \ldots, v_k\}$. Let C_i be a v_i-almost ID code in B_i, $\forall i \in \{1, 2, \ldots, k\}$ and $C = \bigcup_{i=1}^{k} C_i$, then

- *If $u, v \in V' = V(B) - V(K)$, they are dominated and separated by C.*
- *Let $v \in V'$ and $v_j \in V(K)$ such that $d(v, v_j) = 1$. Then v and v_j are dominated and separated by C if there is $i \in \{1, 2, \ldots, k\}$ $i \neq j$ such that C_i is CO.*
- *Let $v \in V'$ and $v_j \in V(K)$ such that $d(v, v_j) = 2$. Then v and v_j are dominated and separated by C if C_j satisfies CO or there is $i \in \{1, 2, \ldots, k\}$ $i \neq j$ such that C_i is CO and $v \notin V(B_i)$.*

THEOREM

Let B be a block graph, K a maximal clique $V(K) = \{v_1, v_2, \ldots, v_k\}$. Let C_i be a v_i-almost ID code in B_i, $\forall i \in \{1,2,\ldots,k\}$ and $C = \bigcup_{i=1}^{k} C_i$, then

- *If $u, v \in V' = V(B) - V(K)$, they are dominated and separated by C.*
- *Let $v \in V'$ and $v_j \in V(K)$ such that $d(v, v_j) = 1$. Then v and v_j are dominated and separated by C if there is $i \in \{1,2,\ldots,k\}$ $i \neq j$ such that C_i is CO.*
- *Let $v \in V'$ and $v_j \in V(K)$ such that $d(v, v_j) = 2$. Then v and v_j are dominated and separated by C if C_j satisfies CO or there is $i \in \{1,2,\ldots,k\}$ $i \neq j$ such that C_i is CO and $v \notin V(B_i)$.*
- *If $v_i, v_j \in V(K)$ with $i \neq j$ then v_i and v_j are dominated and separated by C if either C_i is ADJ or C_j is ADJ.*

Remind that given $v \in V(B)$,

$$\Gamma_{\mathsf{ID}}(B) = min \begin{cases} \Gamma_{\mathsf{ID,CO,ADJ,FN}}(v,B) \\ \Gamma_{\mathsf{ID,CO,ADJ,\overline{FN}}}(v,B) \\ \Gamma_{\mathsf{ID,CO,\overline{ADJ}}}(v,B) \\ \Gamma_{\mathsf{ID,\overline{CO},ADJ}}(v,B) \end{cases}$$

Remind that given $v \in V(B)$,

$$\Gamma_{\mathsf{ID}}(B) = min \begin{cases} \Gamma_{\mathsf{ID,CO,ADJ,FN}}(v,B) \\ \Gamma_{\mathsf{ID,CO,ADJ,\overline{FN}}}(v,B) \\ \Gamma_{\mathsf{ID,CO,\overline{ADJ}}}(v,B) \\ \Gamma_{\mathsf{ID,\overline{CO},ADJ}}(v,B) \end{cases}$$

How to compute $\Gamma_{\mathsf{ID,CO,ADJ,FN}}(v,B)$?

Let $C \subset V(B)$ be a $v-$almost ID code satisfying ID, CO, ADJ and FN.
Let $v = v_1 \in V(K)$ with K maximal clique in B.

Let $C \subset V(B)$ be a $v-$almost ID code satisfying ID, CO, ADJ and FN.
Let $v = v_1 \in V(K)$ with K maximal clique in B.
It holds that:

- C_i is a v_i-almost ID code in B_i $\forall\, i \in \{1,2,\ldots,k\}$.

Let $C \subset V(B)$ be a $v-$almost ID code satisfying ID, CO, ADJ and FN.
Let $v = v_1 \in V(K)$ with K maximal clique in B.
It holds that:

- C_i is a v_i-almost ID code in $B_i \; \forall \, i \in \{1, 2, \ldots, k\}$.
- C_1 is CO (since C is CO).

Let $C \subset V(B)$ be a $v-$almost ID code satisfying ID, CO, ADJ and FN.
Let $v = v_1 \in V(K)$ with K maximal clique in B.
It holds that:

- C_i is a v_i-almost ID code in B_i $\forall\, i \in \{1,2,\ldots,k\}$.
- C_1 is CO (since C is CO).
- There is w such that $N[w] \cap C = \{v_1\}$ (since C is FN).

Let $C \subset V(B)$ be a $v-$almost ID code satisfying ID, CO, ADJ and FN.
Let $v = v_1 \in V(K)$ with K maximal clique in B.
It holds that:

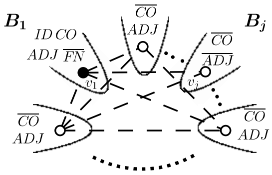
- C_i is a v_i-almost ID code in B_i $\forall\, i \in \{1, 2, \ldots, k\}$.
- C_1 is CO (since C is CO).
- There is w such that $N[w] \cap C = \{v_1\}$ (since C is FN).

  If $w \notin V(B_1)$ then C_1 is $\overline{\text{FN}}$ and $w = v_j$ for some $j \neq 1$.

Let $C \subset V(B)$ be a $v-$almost ID code satisfying ID, CO, ADJ and FN.
Let $v = v_1 \in V(K)$ with K maximal clique in B.
It holds that:

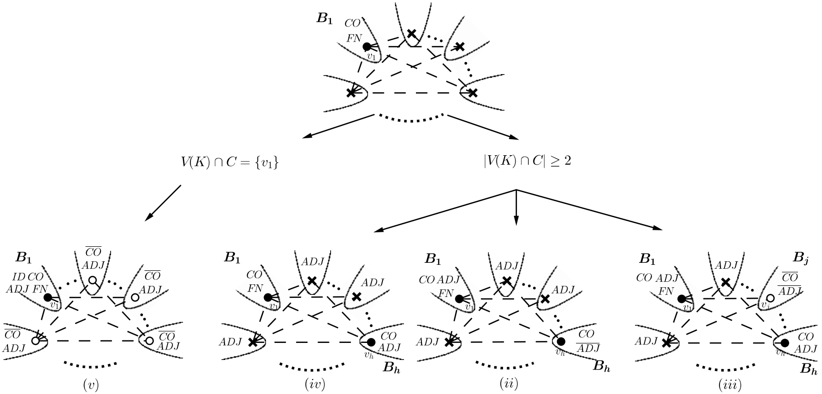
- C_i is a v_i-almost ID code in B_i $\forall\, i \in \{1, 2, \dots, k\}$.
- C_1 is CO (since C is CO).
- There is w such that $N[w] \cap C = \{v_1\}$ (since C is FN).

  If $w \notin V(B_1)$ then C_1 is $\overline{\text{FN}}$ and $w = v_j$ for some $j \neq 1$.
  Moreover

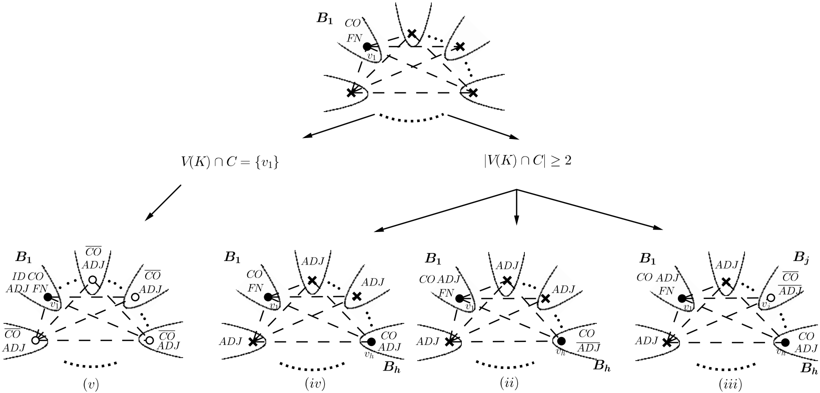


If $w \in V(B_1)$ then C_1 is FN.

If $w \in V(B_1)$ then C_1 is FN.



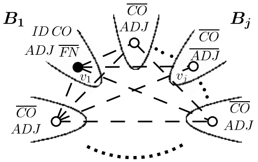
If $w \in V(B_1)$ then C_1 is FN.



Conversely, C_i is a v_i-almost ID of B_i $\forall i \in \{1, \ldots, k\}$ satisfying any of $(i), (ii), (iii), (iv), (v)$ then C satisfies the properties ID, CO, ADJ and FN.

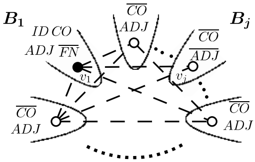
Hence, if C_i is a v_i-almost ID of B_i $\forall i \in \{1,\ldots,k\}$ satisfying (i), then

- C_1 satisfies ID, CO, ADJ, $\overline{\text{FN}}$,
- $\exists j \in \{2,3,\ldots,k\}$ such that C_j satisfies $\overline{\text{CO}}$, $\overline{\text{ADJ}}$,
- $\forall i \in \{2,3,\ldots,k\}$, $i \neq j$, C_i satisfies $\overline{\text{CO}}$, ADJ.



Hence, if C_i is a v_i-almost ID of B_i $\forall i \in \{1,\ldots,k\}$ satisfying (i), then

- C_1 satisfies ID, CO, ADJ, $\overline{\text{FN}}$,
- $\exists j \in \{2,3,\ldots,k\}$ such that C_j satisfies $\overline{\text{CO}}$, $\overline{\text{ADJ}}$,
- $\forall i \in \{2,3,\ldots,k\}$, $i \neq j$, C_i satisfies $\overline{\text{CO}}$, ADJ.

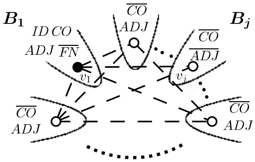


In order to obtain $\Gamma_{\text{ID,CO,ADJ,FN}}(v,B)$ we need to compute:

- $\Gamma_{\text{ID,CO,ADJ,}\overline{\text{FN}}}(v_1,B_1)$
- $\Gamma_{\overline{\text{CO}},\overline{\text{ADJ}}}(v_j,B_j)$ for only one $j \in \{1,\ldots,k\}$
- $\Gamma_{\overline{\text{CO}},\text{ADJ}}(v_i,B_i)$ for all $i \neq j$

Hence, if C_i is a v_i-almost ID of B_i $\forall i \in \{1,\ldots,k\}$ satisfying (i), then

- C_1 satisfies ID, CO, ADJ, $\overline{\text{FN}}$,
- $\exists j \in \{2,3,\ldots,k\}$ such that C_j satisfies $\overline{\text{CO}}$, $\overline{\text{ADJ}}$,
- $\forall i \in \{2,3,\ldots,k\}$, $i \neq j$, C_i satisfies $\overline{\text{CO}}$, ADJ.



In order to obtain $\Gamma_{\text{ID,CO,ADJ,FN}}(v,B)$ we need to compute:

- $\Gamma_{\text{ID,CO,ADJ,}\overline{\text{FN}}}(v_1,B_1)$
- $\Gamma_{\overline{\text{CO}},\overline{\text{ADJ}}}(v_j,B_j)$ for only one $j \in \{1,\ldots,k\}$
- $\Gamma_{\overline{\text{CO}},\text{ADJ}}(v_i,B_i)$ for all $i \neq j$

And all functions corresponding to the cases $(ii),(iii),(iv),(v)$

Name	Function	Name	Function
f_1	$\Gamma_{\text{ID,CO,ADJ,FN}}$	f_6	$\Gamma_{\text{CO,ADJ,}\overline{\text{FN}}}$
f_2	$\Gamma_{\text{ID,CO,ADJ,}\overline{\text{FN}}}$	f_7	$\Gamma_{\text{CO,}\overline{\text{ADJ}}\text{,FN}}$
f_3	$\Gamma_{\text{ID,CO,}\overline{\text{ADJ}}}$	f_8	$\Gamma_{\text{CO,}\overline{\text{ADJ}}\text{,}\overline{\text{FN}}}$
f_4	$\Gamma_{\text{ID,}\overline{\text{CO}}\text{,ADJ}}$	f_9	$\Gamma_{\overline{\text{CO}}\text{,ADJ}}$
f_5	$\Gamma_{\text{CO,ADJ,FN}}$	f_{10}	$\Gamma_{\overline{\text{CO}}\text{,}\overline{\text{ADJ}}}$

List of main functions

Name	Function	Name	Function
f_1	$\Gamma_{\text{ID,CO,ADJ,FN}}$	f_6	$\Gamma_{\text{CO,ADJ,}\overline{\text{FN}}}$
f_2	$\Gamma_{\text{ID,CO,ADJ,}\overline{\text{FN}}}$	f_7	$\Gamma_{\text{CO,}\overline{\text{ADJ}}\text{,FN}}$
f_3	$\Gamma_{\text{ID,CO,}\overline{\text{ADJ}}}$	f_8	$\Gamma_{\text{CO,}\overline{\text{ADJ}}\text{,}\overline{\text{FN}}}$
f_4	$\Gamma_{\text{ID,}\overline{\text{CO}}\text{,ADJ}}$	f_9	$\Gamma_{\overline{\text{CO}}\text{,ADJ}}$
f_5	$\Gamma_{\text{CO,ADJ,FN}}$	f_{10}	$\Gamma_{\overline{\text{CO}}\text{,}\overline{\text{ADJ}}}$

List of main functions

Name	Function	Name	Function
f_{11}	$\Gamma_{\text{CO,}\overline{\text{ADJ}}} = min\{f_7, f_8\}$	f_{15}	$\Gamma_{\overline{\text{CO}}} = min\{f_9, f_{10}\}$
f_{12}	$\Gamma_{\text{CO,ADJ}} = min\{f_5, f_6\}$	f_{16}	$\Gamma_{\text{ADJ}} = min\{f_5, f_6, f_9\}$
f_{13}	$\Gamma_{\text{CO,FN}} = min\{f_5, f_7\}$	f_{17}	$\Gamma_{\text{ID,CO,ADJ}} = min\{f_1, f_2\}$
f_{14}	$\Gamma_{\text{CO,}\overline{\text{FN}}} = min\{f_6, f_8\}$		

List of auxiliary functions

$$f_1(v_1,B) = \Gamma_{\text{ID,CO,ADJ,FN}}(v_1,B) =$$

$$= min \begin{cases} f_2(v_1,B_1) + \min\limits_{j=2,\dots,k} \left\{ f_{10}(v_j,B_j) + \sum\limits_{\substack{i=2 \\ i\neq j}}^{k} f_9(v_i,B_i) \right\} \\ f_5(v_1,B_1) + \min\limits_{h=2,\dots,k} \left\{ f_{11}(v_h,B_h) + \sum\limits_{\substack{i=2 \\ i\neq h}}^{k} f_{16}(v_i,B_i) \right\} \\ f_5(v_1,B_1) + \min\limits_{\substack{j,h=2,\dots,k \\ j\neq h}} \left\{ f_{10}(v_j,B_j) + f_{12}(v_h,B_h) + \sum\limits_{\substack{i=2 \\ i\neq j,h}}^{k} f_{16}(v_i,B_i) \right\} \\ f_{13}(v_1,B_1) + \min\limits_{h=2,\dots,k} \left\{ f_{12}(v_h,B_h) + \sum\limits_{\substack{i=2 \\ i\neq h}}^{k} f_{16}(v_i,B_i) \right\} \\ f_1(v_1,B_1) + \sum\limits_{i=2}^{k} f_9(v_i,B_i) \end{cases}$$

Given $v \in V(B)$ and according to the given functions,

Given $v \in V(B)$ and according to the given functions,

$$\Gamma_{\mathsf{ID}}(B) = min\{f_1(v,B), f_2(v,B), f_3(v,B), f_4(v,B)\}.$$

Given $v \in V(B)$ and according to the given functions,

$$\Gamma_{\mathsf{ID}}(B) = min\{f_1(v,B), f_2(v,B), f_3(v,B), f_4(v,B)\}.$$

Algorithm ICB

Input: a connected block graph B and its list of maximal cliques.
Output: $\Gamma_{\mathsf{ID}}(B)$.

1: randomly select a vertex v_1 and call RICB(v_1, B);
2: return $\Gamma_{\mathsf{ID}}(v_1, B) = min\{f_1(v_1,B), f_2(v_1,B), f_3(v_1,B), f_4(v_1,B)\}$.

Given $v \in V(B)$ and according to the given functions,

$$\Gamma_{\mathsf{ID}}(B) = min\{f_1(v,B), f_2(v,B), f_3(v,B), f_4(v,B)\}.$$

Algorithm ICB
Input: a connected block graph B and its list of maximal cliques. **Output:** $\Gamma_{\mathsf{ID}}(B)$. 1: randomly select a vertex v_1 and call RICB(v_1, B); 2: return $\Gamma_{\mathsf{ID}}(v_1, B) = min\{f_1(v_1,B), f_2(v_1,B), f_3(v_1,B), f_4(v_1,B)\}$.

Algorithm ICB randomly selects a vertex v_1 in block graph B and calls Algorithm RICB(v_1, B) that computes the values of all 10 functions in a recursive manner in smaller and smaller block graphs.

Initialization: List L when $V(B) = \{v\}$

Name	Function	$f_j(v,\{v\})$	Name	Function	$f_j(v,\{v\})$
f_1	$\Gamma_{\text{ID,CO,ADJ,FN}}$	∞	f_6	$\Gamma_{\text{CO,ADJ},\overline{\text{FN}}}$	∞
f_2	$\Gamma_{\text{ID,CO,ADJ},\overline{\text{FN}}}$	∞	f_7	$\Gamma_{\text{CO},\overline{\text{ADJ}},\text{FN}}$	∞
f_3	$\Gamma_{\text{ID,CO},\overline{\text{ADJ}}}$	1	f_8	$\Gamma_{\text{CO},\overline{\text{ADJ}},\overline{\text{FN}}}$	1
f_4	$\Gamma_{\text{ID},\overline{\text{CO}},\text{ADJ}}$	∞	f_9	$\Gamma_{\overline{\text{CO}},\text{ADJ}}$	∞
f_5	$\Gamma_{\text{CO,ADJ,FN}}$	∞	f_{10}	$\Gamma_{\overline{\text{CO}},\overline{\text{ADJ}}}$	0

THE LINEAR-TIME ALGORITHM FOR THE IDENTIFYING CODE PROBLEM ON BLOCK GRAPHS

Initialization: List L when $V(B) = \{v\}$

Name	Function	$f_j(v, \{v\})$	Name	Function	$f_j(v, \{v\})$
f_1	$\Gamma_{\text{ID,CO,ADJ,FN}}$	∞	f_6	$\Gamma_{\text{CO,ADJ,}\overline{\text{FN}}}$	∞
f_2	$\Gamma_{\text{ID,CO,ADJ,}\overline{\text{FN}}}$	∞	f_7	$\Gamma_{\text{CO,}\overline{\text{ADJ}}\text{,FN}}$	∞
f_3	$\Gamma_{\text{ID,CO,}\overline{\text{ADJ}}}$	1	f_8	$\Gamma_{\text{CO,}\overline{\text{ADJ}}\text{,}\overline{\text{FN}}}$	1
f_4	$\Gamma_{\text{ID,}\overline{\text{CO}}\text{,ADJ}}$	∞	f_9	$\Gamma_{\overline{\text{CO}}\text{,ADJ}}$	∞
f_5	$\Gamma_{\text{CO,ADJ,FN}}$	∞	f_{10}	$\Gamma_{\overline{\text{CO}}\text{,}\overline{\text{ADJ}}}$	0

Algorithm RICB

Input: a block graph B, its list of maximal cliques and $v_1 \in V(B)$.
Output: the list L of the values of the ten functions f_j on (v_1, B).

if v_1 has degree 0 in B **then**
  initialize L;
else
  let K be a maximal clique with $V(K) = \{v_1, \ldots, v_k\}$ and delete its edges;
  let $B_1, \ldots, B_k$ be the remaining block graphs, resp., containing $v_1, \ldots, v_k$;
  let $L_i = RICB(v_i, B_i)$ for all $i \in \{1, \ldots, k\}$;
  compute the ten functions on (v_1, B) from L_i for all $i \in \{1, \ldots, k\}$.
end if
return the list L of the values of the ten functions f_j on (v_1, B).

The linear-time algorithm for the identifying code problem on block graphs

Theorem

For each of the ten functions f_j, we can compute $f_j(v_1,B)$ from $L_i(v_i,B_i)$ for all $i \in \{1,\ldots,k\}$ in time $O(k)$.

The linear-time algorithm for the identifying code problem on block graphs

Theorem

For each of the ten functions f_j, we can compute $f_j(v_1, B)$ from $L_i(v_i, B_i)$ for all $i \in \{1, \ldots, k\}$ in time $O(k)$.

Theorem

Algorithm ICB computes in linear time $\Gamma_{ID}(B)$ of an identifiable block graph B (or returns ∞ if no identifying code exists in B).

THE LINEAR-TIME ALGORITHM FOR THE IDENTIFYING CODE PROBLEM ON BLOCK GRAPHS

COROLLARY

If B is a vertex-weighted block graph, the ICB can be easily modified in order to return the minimum weighted identifying code number.

The linear-time algorithm for the identifying code problem on block graphs

Corollary

If B is a vertex-weighted block graph, the ICB can be easily modified in order to return the minimum weighted identifying code number.

Corollary

Algorithm ICB could be modified in order to obtain an identifying code of minimum size.

Outline

1. Identifying codes in graphs
2. The identifying code problem on block graphs
3. Concluding Remarks

The Identifying code Problem (ICP):

- Find $C \subseteq V$ of minimum size such that $C \cap N[i] \neq \emptyset$ and $C \cap N[i] \neq C \cap N[j]$, $i,j \in V$.

Future work

The Identifying code Problem (ICP):

- Find $C \subseteq V$ of minimum size such that $C \cap N[i] \neq \emptyset$ and $C \cap N[i] \neq C \cap N[j]$, $i,j \in V$.

Problems related to the Identifying Code Problem in graphs:

- Locating dominating Problem (LDP)
  Find $L \subseteq V$ of minimum size such that $L \cap N[i] \neq \emptyset$ and $L \cap N(i) \neq L \cap N(j)$, $i,j \in V - L$.

The Identifying code Problem (ICP):

- Find $C \subseteq V$ of minimum size such that $C \cap N[i] \neq \emptyset$ and $C \cap N[i] \neq C \cap N[j]$, $i,j \in V$.

Problems related to the Identifying Code Problem in graphs:

- Locating dominating Problem (LDP)
  Find $L \subseteq V$ of minimum size such that $L \cap N[i] \neq \emptyset$ and $L \cap N(i) \neq L \cap N(j)$, $i,j \in V - L$.

- Open locating dominating Problem (OLDP)
  Find $O \subseteq V$ of minimum size such that $O \cap N(i) \neq \emptyset$ and $O \cap N(i) \neq O \cap N(j)$, $i,j \in V$.

Future work

The Identifying code Problem (ICP):

- Find $C \subseteq V$ of minimum size such that $C \cap N[i] \neq \emptyset$ and $C \cap N[i] \neq C \cap N[j]$, $i,j \in V$.

Problems related to the Identifying Code Problem in graphs:

- Locating dominating Problem (LDP)
  Find $L \subseteq V$ of minimum size such that $L \cap N[i] \neq \emptyset$ and $L \cap N(i) \neq L \cap N(j)$, $i,j \in V - L$.
- Open locating dominating Problem (OLDP)
  Find $O \subseteq V$ of minimum size such that $O \cap N(i) \neq \emptyset$ and $O \cap N(i) \neq O \cap N(j)$, $i,j \in V$.

Our future goals:

- Adapt the linear-time algorithm for ICP to LDP and OLDP.
- Extend the results to graphs whose 2-connected components are cycles or complete bipartite graphs.

Thanks