On the Representation Number of Bipartite Graphs

Khyodeno Mozhui K. V. Krishna

Indian Institute of Technology Guwahati,
India

19th International Conference on
Relational and Algebraic Methods in Computer Science
RAMiCS 2021
November 2, 2021

Outline

Word-representable graph

Definition

A simple graph $G = (V, E)$ is said to be a *word-representable graph* if it can be presented by a word over V. Any two vertices a and b are adjacent if and only if $ababab a\ldots$ or $bababab\ldots$ appear in the word w.

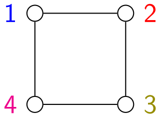


Figure 1 : Cycle C_4.

$w = 413423124$
represents the graph C_4.

- 41414 represents 1 and 4 are adjacent.
- 1331 represents that 1 and 3 are non-adjacent vertices.

k-word-representable graph

- If every letters in w appear exactly k times then w is said to be k-uniform.

Definition

A graph G is said to be a *k-word-representable graph* if there exist a k-uniform word that represents it.

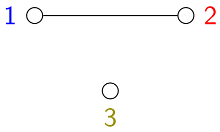


Figure 2 : A Graph G.

- 123312 represents G.
- G is a 2-word-representable graph.

Proposition (Kitaev and Pyatkin (2008))

A k-word-representable graph G is also $(k+1)$-word-representable graph. In particular, each word-representable graph has infinitely many word-representants.

Representation number

Definition

The minimal k such that a graph G is a k-word-representable graph is called the *representation number of G* and is denoted by $\mathcal{R}(G)$.

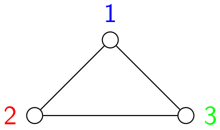


Figure 3 : Graph K_3.

- 123 represents K_3.
- Any complete graph, K_n, on n vertices can be represented by a permutation. So, they have representation number 1.

Permutationally representable graph

Definition

A graph $G = (V, E)$ is said to be a *permtationally representable graph* if it can be represented by a word of the form $w = p_1 p_2 ... p_k$, where p_i is a permutation of V, for all i where $1 \leq i \leq k$. We say G is *permutationally k-representable*.

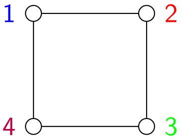


Figure 4 : Cycle C_4.

- $\overbrace{1324}\underbrace{3142}$ represents the graph C_4.
- C_4 is permutationally 2-representable.

Permutation representation number

Definition

The minimal k such that a graph G is a permutationally k-representable graph is called the *permutation representation number of* G, and it is denoted by $\mathcal{R}^p(G)$.

Remark

- *For any permutationally representable graph* G, $\mathcal{R}(G) \leq \mathcal{R}^p(G)$.
- *For all* $n \geq 1$, $\mathcal{R}^p(K_n) = 1$.

Example

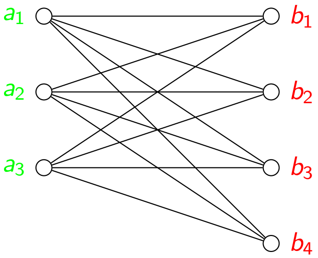


Figure 5 : $K_{3,4}$

- $\overbrace{a_1 a_2 a_3 b_1 b_2 b_3 b_4}\underbrace{a_3 a_2 a_1 b_4 b_3 b_2 b_1}$ represents $K_{3,4}$.
- $\mathcal{R}^p(K_{3,4}) = 2$

Remark

- *Any complete bipartite graph $K_{m,n}$, $\mathcal{R}^p(K_{m,n}) = 2$.*

Semi-transitive Orientation

Definition

A graph $G = (V, E)$ is said to be *semi-transitive* if it admits a semi-transitive orientation, i.e., G admits an acyclic orientation such that for any directed path $v_1 \to v_2 \to \cdots \to v_n$ $\forall v_i, \in V$ where $1 \leq i \leq n$, either

- there is no edge from v_1 to v_n, or
- if there is an edge from v_1 to v_n then there are edges from v_i to v_j, $\forall\ 1 \leq i < j \leq n$.

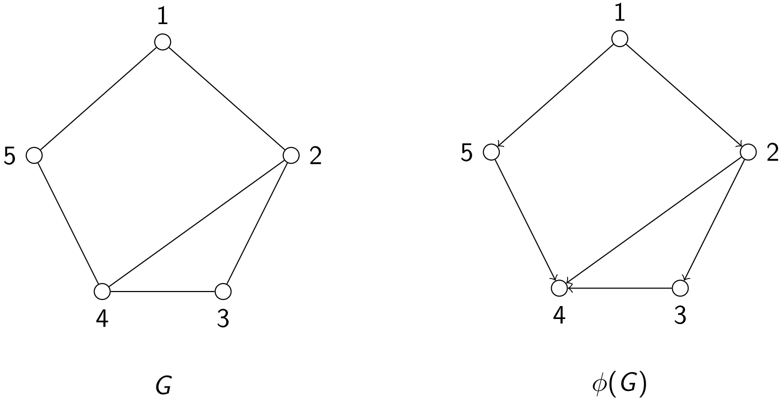


Figure 6 : Semi-transitive orientation of G.

Theorem (Halldórsson et al. (2016))

A graph is word-representable if and only if it admits a semi-transitive orientation.

Theorem (Halldórsson et al. (2016))

Each non-complete word-representable graph G on n vertices is $2(n - \mathcal{K}(G))$-word-representable, where $\mathcal{K}(G)$ is the size of the maximum clique in G.

Corollary

If G is non-complete word-representable on n vertices, then $\mathcal{R}(G) \leq 2(n - \mathcal{K}(G))$.

Computation of word-representable graphs

- The recognition problem for word-representable graph is decidable.
- In Halldórsson et al. (2016), it is established that the recognition problem for word-representable graph is NP-complete.

Classes of graphs which are word-representable

- All graphs on atmost 5-vertices are word-representable.
- Complete graphs.
- Comparability graphs.
- There are classes of graphs that contain both word-representable and non-word-representable graphs: 4-colourable graphs, perfect graphs, etc.

Comparability graphs

Definition

A graph $G = (V, E)$ is a comparability graph if it admits a transitive orientation, i.e, an orientation in which if $\overrightarrow{xy}$ and $\overrightarrow{yz}$ are directed edges, then $\overrightarrow{xz}$ is a directed edge, for all $x, y, z \in V$.

Figure 7 : Transitive orientation of H.

Theorem (Kitaev and Seif (2008))

A graph is permutationally representable if and only if it is a comparability graph.

- Every comparability graph corresponds to a partially ordered set (poset).
- In Yannakakis (1982), the problem of finding the dimension of a poset is NP-hard.

Result (Kitaev and Lozin (2015))

A comparability graph H is permutationally k-representable if and only if the poset induced by this graph has dimension at most k.

Crown graph

Definition

Crown graphs are graphs which are obtained from a complete bipartite graph by removing a perfect matching.

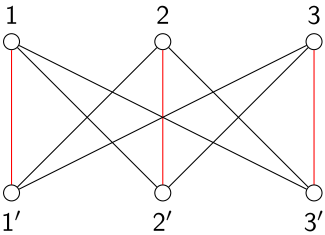


Figure 8 : $K_{3,3}$

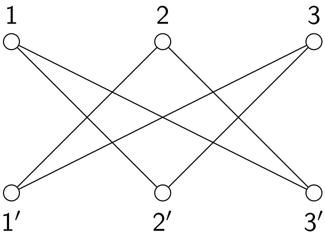


Figure 9 : $H_{3,3}$

Theorem (Halldorsson et al. (2011))

The permutation representation number of a crown graph $H_{n,n}$ is n.

Theorem (Glen et al. (2018))

The representation number of a crown graph $H_{n,n}$ is $\lceil \frac{n}{2} \rceil$.

Conjecture (Glen et al. (2018))

Every bipartite graph on n vertices has representation number at most $\lceil \frac{n}{4} \rceil$.

Main Result

- Every bipartite graph is a comparability graph.
- The class of bipartite graph is precisely the class of comparability graphs that are isomorphic to the Hasse diagram of the corresponding posets.
- We devise an algorithmic procedure that works in polynomial time to construct a word representing permutationally a given bipartite graph.

Relabeling Algorithm

1. Given a bipartite graph $G = (V, E)$ where $V = V_1 \cup V_2$, consider the set, say V_1, with minimum number of vertices.
2. Check whether the graph is a complete bipartite graph. If yes, exit.
3. Else, from the set V_1 choose a vertex with atleast one non adjacent vertex and relabel it first.
4. Relabel the rest of vertices in V_1 and with respect to the relabeling done in V_1, we relabel the vertices in V_2 accordingly.
5. We then create permutations(linear orders) with respect to the vertices in V_1.
6. Concatenate all the permutations and relabel the vertices to their original label.
7. The word produced represents the bipartite graph G, permutationally.

Relabeling Algorithm

1. Given a bipartite graph G, suppose $V_1 = \{a_1, \ldots, a_m\}$ and $V_2 = \{b_1, \ldots, b_n\}$.

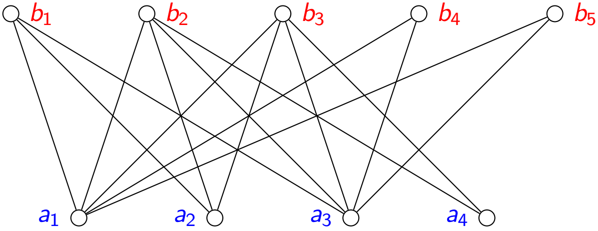
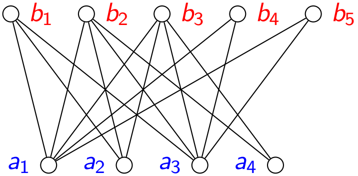


Figure 10 : A bipartite graph G

2. If $N(a) = V_2$ for all $a \in V_1$, then consider the following word and exit:

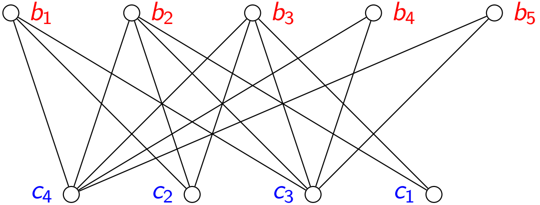
$$w = \overbrace{a_1 a_2 \cdots a_m b_1 b_2 \cdots b_n}\overbrace{a_m a_{m-1} \cdots a_1 b_n b_{n-1} \cdots b_1}.$$

3. Else, choose a in V_1 such that $V_2 \setminus N(a) \neq \varnothing$ and label it as c_1.

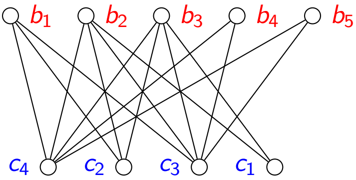


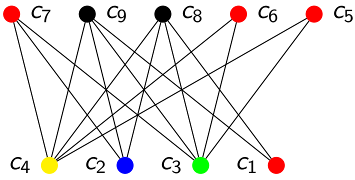
Choosing a to be a_4.

4. Relabel the remaining vertices of V_1 as $c_2, \ldots, c_m$, arbitrarily.



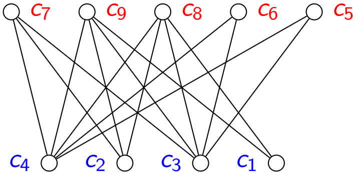
5. Relabel the vertices in V_2 such that the vertices which are not adjacent to c_1 are relabeled first, then the vertices not adjacent to c_2 and so on. Lastly, we relabel the remaining vertices in V_2.





6. Create a list of permutations of the vertices $c_1, \ldots, c_{m+n}$ as per the following:

- $w_1 = c_m c_{m-1} \cdots c_2 c_{m+1} c_{m+2} \cdots c_{m+k_1} c_1 c_{m+k_1+1} c_{m+k_1+2} \cdots c_{m+n}$



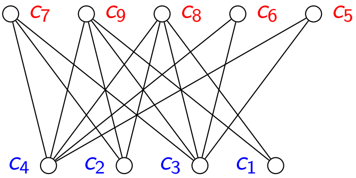
$w_1 = c_4 c_3 c_2 c_5 c_6 c_7 c_1 c_8 c_9$

- For $i = 2$ to m,

  If $V_2 \setminus N(c_i) \neq \varnothing$, then set

  $$w_i = c_1 \cdots c_{i-1} c_{i+1} \cdots c_m \mathrm{Dec}(V_2 \setminus N(c_i)) c_i \mathrm{Dec}(N(c_i));$$

  else, set $w_i = \varepsilon$.

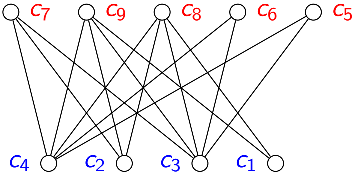


$w_2 = c_1 c_3 c_4 c_6 c_5 c_2 c_9 c_8 c_7$

$w_3 = \varepsilon$

$w_4 = \varepsilon$

7. If $N(c) = V_2$ for some $c \in V_1$ then set
$$w_{m+1} = c_1 c_2 \cdots c_m c_{m+n} \cdots c_{m+1};$$
else, set $w_{m+1} = \varepsilon$.



$$w_5 = c_1 c_2 c_3 c_4 c_9 c_8 c_7 c_6 c_5$$

8. Concatenate the permutations $w_1, w_2, \ldots w_m, w_{m+1}$ to form $w' = w_1 w_2 \cdots w_m w_{m+1}$ which permutationally represents the relabeled graph of G .
$$w' = \overbrace{c_4 c_3 c_2 c_5 c_6 c_7 c_1 c_8 c_9}\,\underbrace{c_1 c_3 c_4 c_6 c_5 c_2 c_9 c_8 c_7}\,\overbrace{c_1 c_2 c_3 c_4 c_9 c_8 c_7 c_6 c_5}$$
9. Replacing the original labels of the vertices of G in the word w'.
$$\overbrace{a_1 a_3 a_2 b_5 b_4 b_1 a_4 b_3 b_2}\,\underbrace{a_4 a_3 a_1 b_4 b_5 a_2 b_2 b_3 b_1}\,\overbrace{a_4 a_2 a_3 a_1 b_2 b_3 b_1 b_4 b_5}$$

Remark

The relabeling algorithm works in $O(mn)$.

Theorem (Correctness of the relabeling algorithm)

The word w generated by the relabeling algorithm represents the bipartite graph G.

Proof: Suppose $G = (V_1 \cup V_2, E)$ such that $m = |V_1| \leq |V_2| = n$. Let G' represent the relabeled graph.
We prove that a and b are adjacent in G' if and only if a and b alternate in the word w'.

- **Case 1:** a and b are adjacent vertices in G'.
  - $w_1 = c_m c_{m-1} \cdots c_2 c_{m+1} c_{m+2} \cdots c_{m+k_1} c_1 c_{m+k_1+1} c_{m+k_1+2} \cdots c_{m+n}$
  - If $w_i \neq \varepsilon$ for $i = 2$ to m,
    $w_i = c_1 \cdots c_{i-1} c_{i+1} \cdots c_m \text{Dec}(V_2 \setminus N(c_i)) c_i \text{Dec}(N(c_i))$
  - If $w_{m+1} \neq \varepsilon$
    $w_{m+1} = c_1 c_2 \cdots c_m c_{m+n} \cdots c_{m+1}$
    else, set $w_{m+1} = \varepsilon$.

In each w_i, the subword ab is produced.

- **Case 2:** a and b are non-adjacent vertices in G'.
  We deal this case in three subcases. In each case, we identify two linear orders (permutations): one with the subword ab and other with the ba.
  - **Subcase 2.1:** $a, b \in V_1$.
    - In w_1, if $a = c_i$ and $b = c_j$ with $i < j$ then ba is a subword.

      $$w_1 = c_m c_{m-1} \cdots c_2 c_{m+1} c_{m+2} \cdots c_{m+k_1} c_1 c_{m+k_1+1} c_{m+k_1+2} \cdots c_{m+n}$$

    - In w_j, if $w_j \neq \varepsilon$ then ab is a subword.

      $$w_j = c_1 \cdots c_{j-1} c_{j+1} \cdots c_m \mathrm{Dec}(V_2 \setminus N(c_j)) c_j \mathrm{Dec}(N(c_j))$$

    - Else, $w_{m+1} \neq \varepsilon$ then ab is a subword.

      $$w_{m+1} = c_1 c_2 \cdots c_m c_{m+n} \cdots c_{m+1}$$

- **Subcase 2.2:** $a \in V_1$ and $b \in V_2$.
  - If $a = c_1$ and $b = c_j$, then ba is a subword of
    $$w_1 = c_m c_{m-1} \cdots c_2 c_{m+1} c_{m+2} \cdots c_{m+k_1} c_1 c_{m+k_1+1} c_{m+k_1+2} \cdots c_{m+n}$$
    The graph has no isolated vertices, $\exists c_k \in V_1$ such that $b \in N(c_k)$,
    $$w_k = c_1 \cdots c_{k-1} c_{k+1} \cdots c_m \text{Dec}(V_2 \setminus N(c_k)) c_k \text{Dec}(N(c_k))$$
    ab is a subword.
  - If $a = c_i$ and $b = c_j$, then ab is a subword of w_1. Since a and b are not adjacent,
    $$w_i = c_1 \cdots c_{i-1} c_{i+1} \cdots c_m \text{Dec}(V_2 \setminus N(c_i)) c_i \text{Dec}(N(c_i))$$
    ba is a subword.

- **Subcase 2.3:** $a, b \in V_2$. If $a = c_i$ and $b = c_j$ with $i < j$. Then ab is a subword of w_1.

$$w_1 = c_m c_{m-1} \cdots c_2 c_{m+1} c_{m+2} \cdots c_{m+k_1} c_1 c_{m+k_1+1} c_{m+k_1+2} \cdots c_{m+n}$$

The graph has no isolated vertices, $\exists c_k \in V_1$ such that $a \in N(c_k)$.

  - If $N(c_k) = V_2$ then $w_{m+1} \neq \varepsilon$, ba is a subword.

$$w_{m+1} = c_1 c_2 \cdots c_m c_{m+n} \cdots c_{m+1}$$

  - Else, $w_k \neq \varepsilon$ then ba is a subword.

$$w_k = c_1 \cdots c_{k-1} c_{k+1} \cdots c_m \mathrm{Dec}(V_2 \setminus N(c_k)) c_k \mathrm{Dec}(N(c_k))$$

Conclusion

Theorem

Let m be the size of the smallest set in the bipartition of a bipartite graph G, the permutation representation number $\mathcal{R}^p(G) \leq m$. Consequently, $\mathcal{R}(G) \leq m$.

Corollary

Let $\{a_{i_1}, \ldots, a_{i_k}\} \subseteq V_1$ be the set of vertices each of which is adjacent to all vertices of V_2 in a bipartite graph G, then $\mathcal{R}^p(G) \leq m - k + 1$.

Remark

The relabeling algorithm proposed for bipartite graphs has a scope to extend it for comparability graphs, in general.

References I

Glen, M., Kitaev, S. and Pyatkin, A.: 2018, On the representation number of a crown graph, *Discrete Applied Mathematics* **244**, 89–93.

Halldorsson, M., Kitaev, S. and Pyatkin, A.: 2011, Alternation graphs, *in* P. Kolman and J. Kratochvil (eds), *Graph-theoretic concepts in computer science*, Lecture Notes in Computer Science, Springer, pp. 191–202. 37th International workshop on graph-theoretic concepts in computer science, 2011 ; Conference date: 21-06-2011 Through 24-06-2011.

Halldórsson, M., Kitaev, S. and Pyatkin, A.: 2016, Semi-transitive orientations and word-representable graphs, *Discrete Applied Mathematics* **201**, 164–171.

Kitaev, S. and Lozin, V.: 2015, *Words and Graphs*, Monographs in Theoretical Computer Science, Springer-Verlag.

Kitaev, S. and Pyatkin, A.: 2008, On representable graphs, *Journal of Automata, Languages and Combinatorics* **13**(1), 45–54.

Kitaev, S. and Seif, S.: 2008, Word problem of the Perkins semigroup via directed acyclic graphs, *Order* **25**(3), 177–194.
URL: *https://doi.org/10.1007/s11083-008-9083-7*

References II

Yannakakis, M.: 1982, The complexity of the partial order dimension problem, *SIAM J. Algebraic Discrete Methods* **3**(3), 351–358.
URL: *https://doi.org/10.1137/0603036*

Thank you.