Injective Colouring in Circular-Convex Bipartite Graphs

Course Project Presentation

by

Harsh Kumar
2017MT10729

Vivek Muskan
2017MT10755

T D M S S Pavan Srinivas
2017MT10752

Under the supervision of
Professor B.S.Panda
Juhi Chaudhary, Priyamvada, Pooja Goyal

Department of Mathematics
INDIAN INSTITUTE OF TECHNOLOGY DELHI

CONTENTS

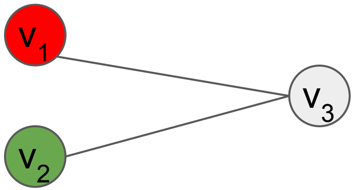
Problem Statement

Find injective coloring algorithm for circular convex bipartite graph

Injective Coloring

An ***Injective Colouring*** of a graph G is the vertex-colouring in which vertices having common neighbour gets different colours.

The minimum number of colours required for G to be injectively coloured is its ***injective chromatic number*** denoted by $\mathbf{X}_{i}\mathbf{(G)}$



v1 and v2 share a common neighbor v3 and hence colored different

<u>Note</u>: *(v1 ,v3) or (v2, v3) can have same colors.*

Decide Injective Colouring Problem

Given a graph G and an integer K,
Is it possible to Injectively color the Graphs G using K-colours.

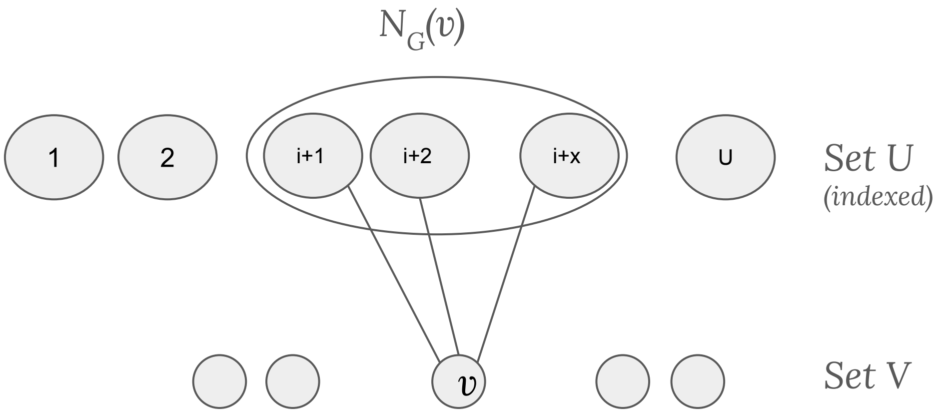
This problem is Np-Complete in general graphs and also in bipartite graphs.

For any general graph $\Delta \leq X_i(G) \leq \Delta^2 - \Delta + 1$

Convex Bipartite Graph

Let $G = (U \cup V, E)$ be a bipartite graph with vertex partitions U and V.
Let $N_G(v)$ denote the neighbourhood of the vertex $v \in V$.

The graph ***G is said to be convex over the vertex set U*** if the vertices in U can be indexed from 1 to $|U|$ such that, for all $v \in V$ the vertices in the $N_G(v)$ are adjacent with respect to the indexing i.e.,
if $d(v) = x$ then all $v \in N_G(v)$ have continuous indices i.e. $i+1, i+2, \ldots, i+x$.



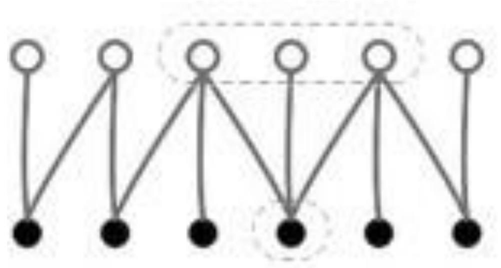


Fig 1: *Example of a Convex bipartite graph, convex over the set of white vertices*

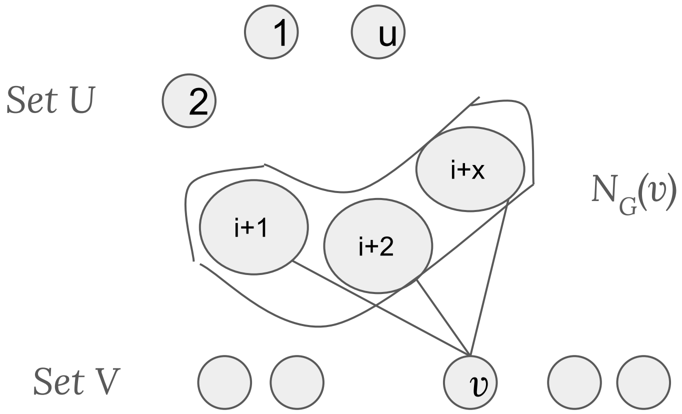
Problem Statement

Find injective coloring algorithm for circular convex bipartite graph

Circular Convex Bipartite Graph

A Bipartite Graph G = (U ∪ V, *E*), is said to be circular convex over U if there exists a circular ordering in U such that for every $v \in$ V the neighbourhood of v forms a (continuous)circular arc

A convex bipartite graph is a circular convex bipartite, but converse is not true.



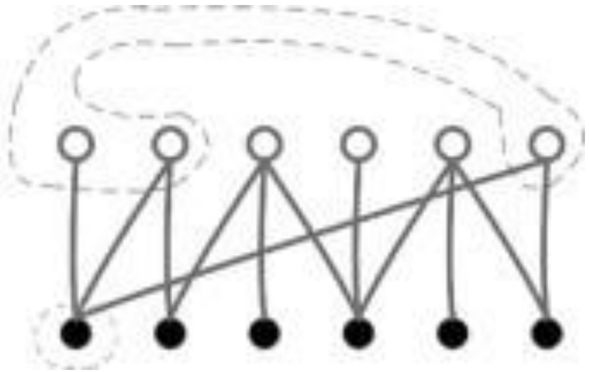


Fig 2: *Example of a Circular Convex bipartite graph, convex over the set of white vertices*

Literature: Algorithm for Convex Bipartite Graph

Lex-Convex Ordering: Let G = (X U Y, E) be a convex bipartite graph, then

An ordering $\sigma = (x_1, x_2, \ldots, x_{nx}, y_1, y_2, \ldots, y_{ny})$ is said to be a lexicographic convex(lex-convex) ordering if:

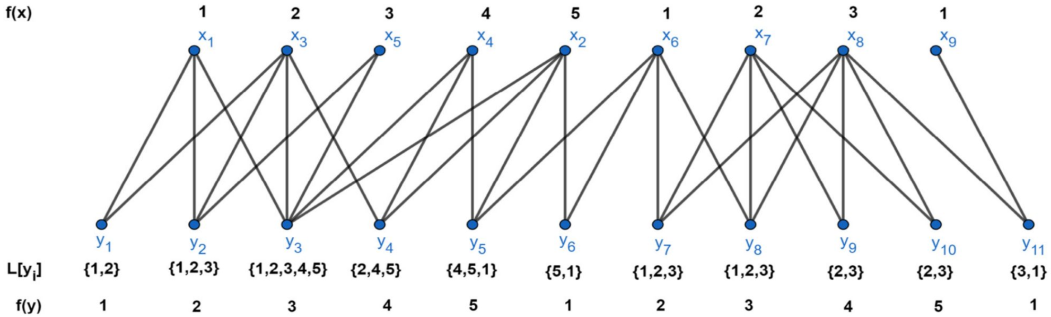
a. *σ is a convex ordering* &

b. *for x_i and x_j with $i < j$, the lowest index in $N_G(x_i) \leq N_G(x_j)$. Also in the case for the highest index.*

- Lex-convex ordering of a graph G that is convex over Y can be determined in O(|X|log|X|) time
- Given a lex-convex ordering of a convex bipartite graph G, this below polynomial time algorithm computes an optimal injective coloring of G
- The algorithm maintains an array $L[y_i]$ of size Δ for each vertex y_i in Y, $1 \leq i \leq n_y$, that stores the set of colors used by its neighbors

Algorithm 2 Convex Bipartite Graph

Input: A convex bipartite graph $G = (X, Y, E)$ with maximum degree Δ and lex-convex ordering of G, $\sigma = (x_1, x_2, \ldots, x_{n_x}, y_1, y_2, \ldots, y_{n_y})$.
Output: An injective coloring, f of G.
begin
  Initialize $S = \emptyset$ and $L[y_i] = \emptyset$, for each $1 \leq i \leq n_y$;
  for $i = 1$ *to* n_y **do**
    if $(i \bmod \Delta \neq 0)$ **then**
      $f(y_i) = i \bmod \Delta$;
    else
      $f(y_i) = \Delta$;
  $i = 1$;
  while $i \leq n_y$ **do**
    if (y_i *has a neighbor x, taken in the order* $(x_1, x_2, \ldots, x_{n_x})$*, which is not yet colored*) **then**
      Let $\{y'_1, \ldots, y'_{d_G(x)}\}$ be the neighbors of x;
      $S = L[y'_1] \cup \ldots \cup L[y'_{d_G(x)}]$;
      Assign $f(x) = \min \{c \in \{1, \ldots, n_x\} \setminus S\}$
      for $r = 1$ *to* $d_G(x)$ **do**
        $L[y'_r] = f(x) \cup L[y'_r]$;
    else
      $i = i + 1$;
  return f;



Approach 1

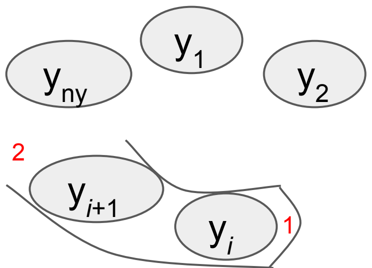
- Similar to previous algorithm

Let $y_1 \rightarrow y_2 \rightarrow \dots \rightarrow y_{ny} \rightarrow y_1$ be the circular convex ordering of the partition Y of G.

1. Initially find x_i that has maximum degree, its neighbours forms an arc A in Y.
2. Starting with one end point of A, colour towards other in the fashion $1,2,\dots,\Delta,1,2,\dots$.(Y done)
3. Later on x_i's are coloured following the while loop of the Algorithm 2.

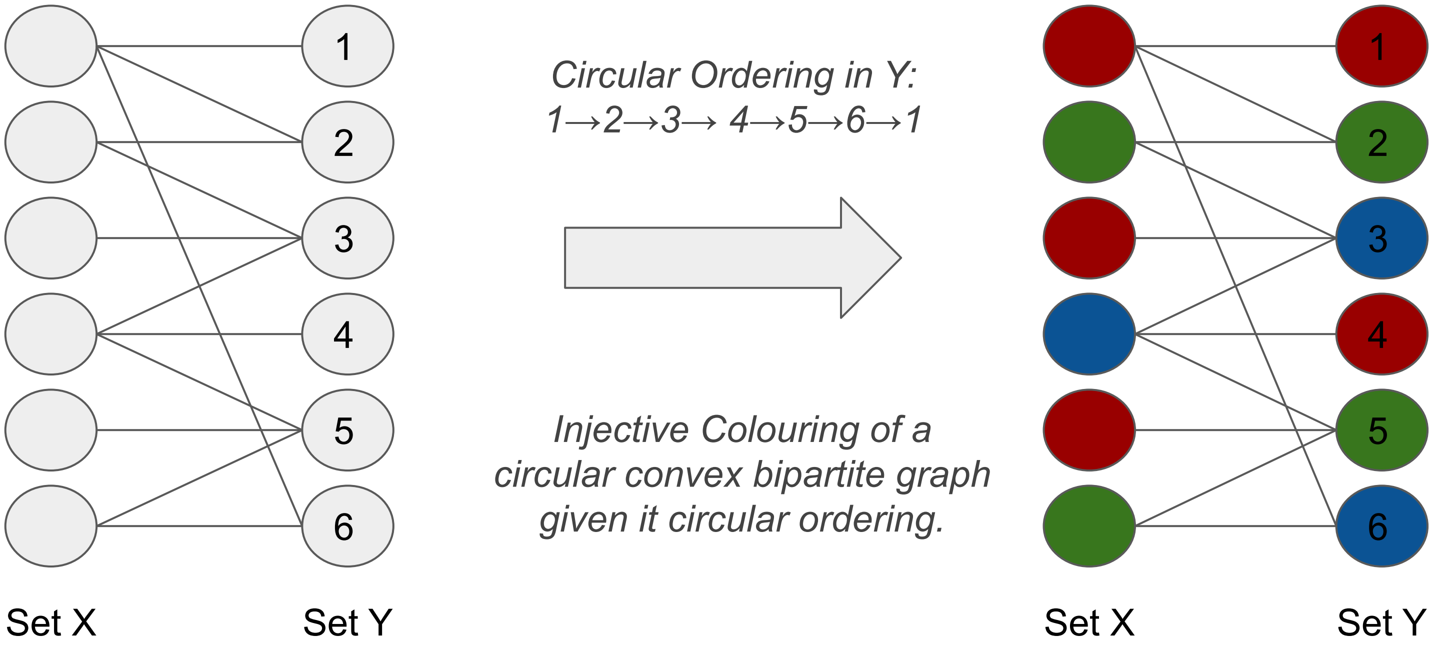
- The only difference is that the neighbours of any particular 'x' may be in the order something like, $\{y_i, y_{i+1}, \dots, y_{ny}, y_1, \dots, y_{d(x)-ny-i-1}\}$, i.e., overshooting y_{ny}.

Complexity of the algorithm is O(mn), where m = |vertex set| and n = |edge set| of G.



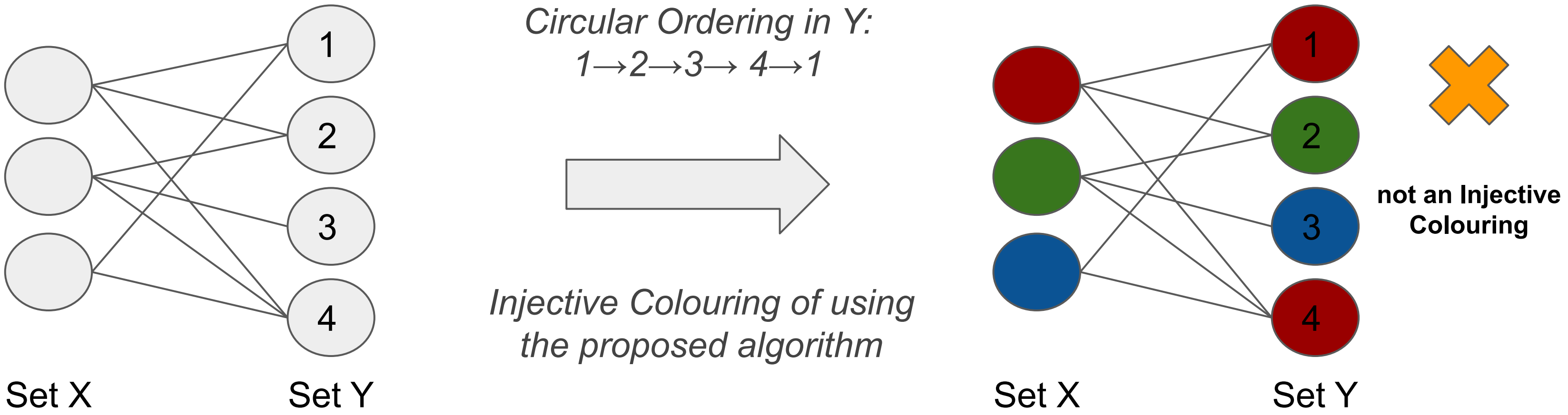
Result 1 (Approach 1)

Result 1: A circular convex bipartite graph with Δ = 3



Result 2: *Failed!*

Result 2: A circular convex bipartite graph with Δ = 3, but cannot be coloured using above algorithm



A circular convex bipartite graph cannot always be coloured using Δ colors injectively.

Even though with Δ = 3, every vertex in Y are adjacent to each other requiring |Y| = 4 colours.

Approach 2- Modification in Approach 1

Let G = (X ∪ Y, *E*) be a circular convex bipartite graph with convexity over Y.

Let $y_1 \rightarrow y_2 \rightarrow \ldots \rightarrow y_{ny} \rightarrow y_1$ be its circular convex ordering of vertex set Y.

The algorithm maintains two set of arrays $L[x_i]$, 1<= i <= nx and $L[y_j]$, 1<= j <= ny.

These arrays stores the colors used by their respective neighbours.

1. Now assign each ***x*** a color **min{ {1,2,...nx}\ ∪ L[y_i'] }**, where "y_i'" $1 <= i <= d_G(x)$ are the neighbours of x.
2. Similarly colour each y in Y.

Time Complexity = O(mn)

m = number of vertices in G

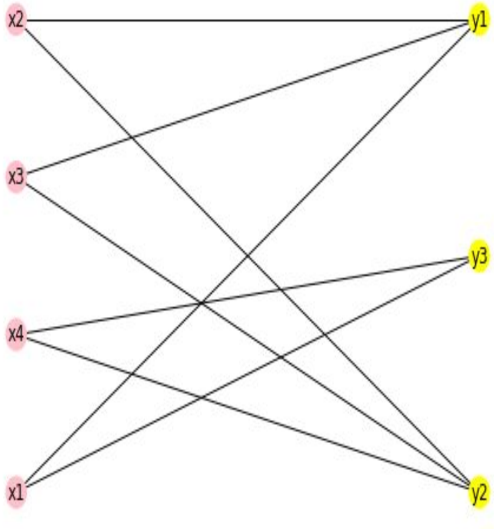
n = number of edges in G

Algorithm implemented in python

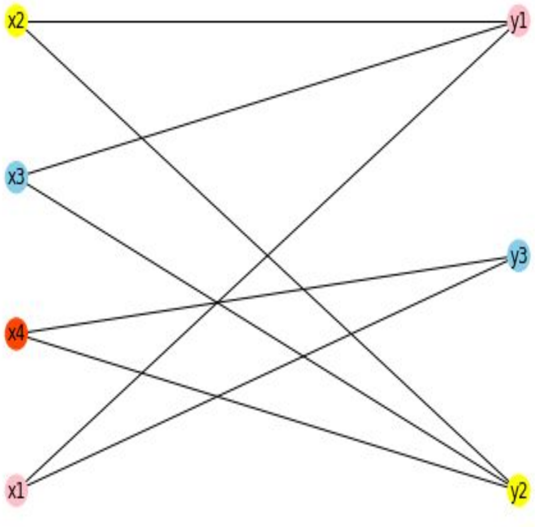
```
# Coloring X
for x in range(1,len(X)+1):
    if(colorX[x-1] == -1):
        S = set()
        for _y in X[x-1]:
            for _x in Y[_y-1]:
                if (colorX[_x-1] != -1): S.add(colorX[_x-1])

        for i in range(1,len(X)+1):
            if i not in S:
                colorX[x-1] = i
                break

# Coloring Y
for y in range(1,len(Y)+1):
    if(colorY[y-1] == -1):
        S = set()
        for _x in Y[y-1]:
            for _y in X[_x-1]:
                if (colorY[_y-1] != -1): S.add(colorY[_y-1])

        for i in range(1,len(Y)+1):
            if i not in S:
                colorY[y-1] = i
                break
```

Result 1 (*previous Counterexample*)

Chromatic No : 4



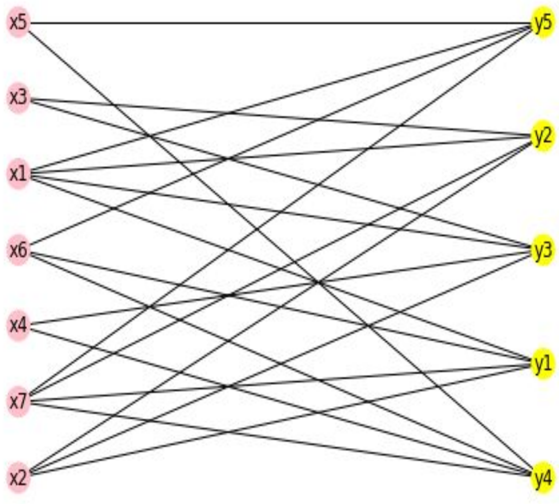
Input



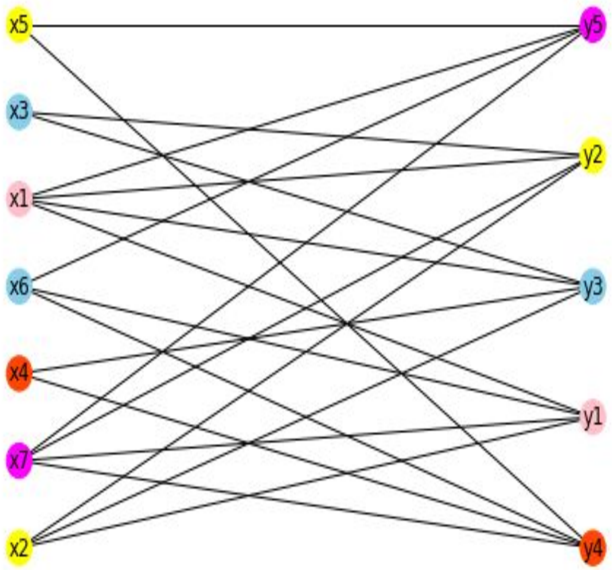
Output

Results 2

Chromatic No : 5



Input



Output

Conclusion and Further Works

- A circular convex bipartite graph may have injective chromatic no >= Δ
- Approach 1 failed as it uses exactly Δ colors
- Approach 2 works for any bipartite graph
- Complexity of Approach 2 is $O(mn)$

Optimality of the Algorithm in the approach 2 needs to be proved.

REFERENCES

1. Fig 1 & 2: https://www.sciencedirect.com/science/article/pii/S0304397514003508
2. Algorithm 1: https://www.sciencedirect.com/science/article/pii/S0166218X20305217
3. Algorithm 2: https://www.sciencedirect.com/science/article/pii/S0166218X20305217
4. Fig 3 & 4: https://www.sciencedirect.com/science/article/pii/S0166218X20305217

THANK YOU