Injective Coloring of Circular Convex Bipartite Graph

Vivek Muskan 2017MT10755

TDMSS Pavan Srinivas 2017MT10752

Harsh Kumar 2017MT10729

Department of Mathematics, IIT Delhi

Advance Algorithm (MTL760) Term Paper

Professor In charge : Prof. B.S Panda

Department of Mathematics, IIT Delhi

Supervisors : Juhi Chaudhary, Priyamvada, Pooja Goyal

Department of Mathematics, IIT Delhi

Primary Citations : B.S. Panda , Priyamvada (2020), ”Injective coloring of some subclasses of bipartite graphs and chordal graphs”, IIT Delhi

Index

1. Abstract

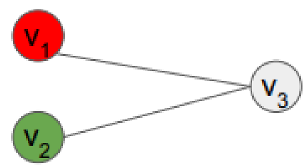
A vertex coloring of a graph G = (V, E) that uses k colors is called an injective k-coloring of G if no two vertices having a common neighbor have the same color. The minimum k for which G has an injective k-coloring is called the injective chromatic number of G. Algorithms to find a valid and optimal injective coloring of convex bipartite graphs, a subclass of bipartite graphs, have been discussed by Prof. B.S Panda and Priyamvada in their paper on Injective Colouring of some subclasses of bipartite graphs and chordal graphs published in 2020. Based on the results from the previously mentioned paper we have taken a step further and tried to find an algorithm for optimal injective coloring in any general circular convex bipartite graph. We were able to give the proof of correctness. The implementation of the algorithm in Python and results obtained for certain Circular convex bipartite graph examples are also included in this paper.

2. Objective

2.1 Basic Definition

- **Injective Coloring:** An **Injective Colouring** of a graph G is the vertex-colouring in which vertices having common neighbour get different colours. The minimum number of colours required for G to be injectively coloured is its **injective chromatic number** denoted by $\mathbf{X_i(G)}$**.**

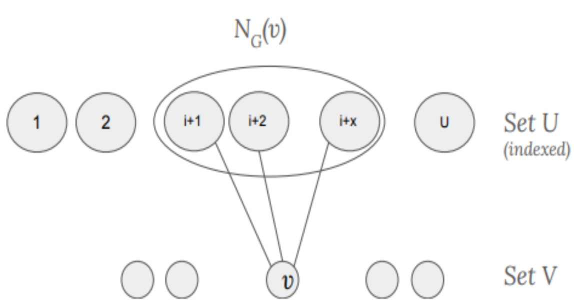
Here in the diagram given below, v1 and v2 share a common neighbor v3 and hence should be colored differently.



Note: (v_1, v_3) or (v_2, v_3) can have the same colors.

- **Convex Bipartite Graph:** Let G = (U ∪ V, E) be a bipartite graph with vertex partitions U and V and let $\mathbf{N_G(v)}$ denote the neighbourhood of any vertex $v \in$.G

  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 $\mathbf{N_G}(v)$ are adjacent to each other with respect to the indexing i.e., if d(v) = x then all v ∈ $\mathbf{N_G(v)}$ have continuous indices of form i+1, i+2, …, i+x. Figure on the left shows how the adjacent vertices of any vertex v ∈ V are ordered, while on the right is an example of a Convex bipartite graph.



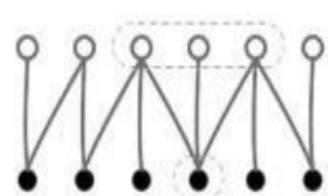
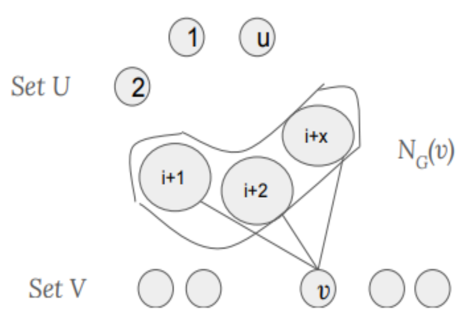


Fig 1: *Example of a Convex bipartite graph, convex over the set of white vertices*

- **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.

  Ex: Below is an example on the left showing how the vertices of partition U can be arranged in a circular manner such that for every vertex v in V, the corresponding neighbors in U form a continuous arch in the Circular arrangement. The figure on the right is an example of a Circular Convex Bipartite graph, convex over white vertex set.

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



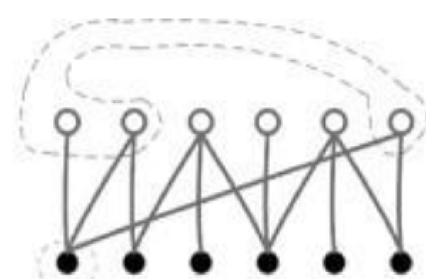


Fig 2: *Example of a Circular Convex bipartite graph, convex over the set of white vertices*

2.2 Problem Statement:

Find a way to *Injectively* colour a given *Circular Convex Bipartite graph* optimally.

3. Literature Survey

3.1. Injective Coloring of Convex Bipartite Graph

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 the highest index in $N_G(x_i) \leq N_G(x_j)$.

Note that lex-convex ordering of a graph G = (X, Y, E) that is convex over Y can be determined in **O(|X|log|X|)** time (B.S Panda and Priyamvada Injective Coloring, 2020).

Algorithm for a Convex Bipartite Graph:

Given a lex-convex ordering of a Convex Bipartite Graph G, we can find a polynomial time algorithm to Injectivelt colour the graph G with minimum number of colours i.e., injective chromatic number of colours. The algorithm is pretty simple but effective. It first colours the vertices of the partition over which the graph is convex, let's say Y. These are coloured with i^{th} indexed vertex getting ($i \bmod \Delta$) or Δ. It also maintains an array **L[y_i]** of size Δ for each y_i, $1 \leq i \leq n_y$, that keeps track of colors used up by its neighbors. Then for each x in X, the algo checks the colours already used in the arrays of neighbours of x, and assigns the least colour missing in those arrays combination.

Algorithm : Injective Coloring of Convex Bipartite Graph

Input : A convex bipartite graph G = (X, Y, E) with maximum degree Δ and lex-convex ordering of G, $\sigma = (x_1, x_2, \ldots, x_{nx}, y_1, y_2, \ldots, y_{ny})$.
Output : An injective coloring, f of G.

```
begin
    Initialize S = ∅ and L[y_i] = ∅, for each 1 ≤ i ≤ n_y;
    for i = 1 to n_y do
        if (i mod Δ ≠ 0) then
            f (y_i) = i mod Δ;
        else
            f (y_i) = Δ;
    i = 1;
    while i ≤ n_y do
        if (y_i has a neighbor x, taken in the order (x_1, x_2, . . . , x_nx), which is not yet
        colored) then
            Let {y′_1 , . . . , y′_dG(x)} be the neighbors of x;
            S = L[y′_1] ∪ . . . ∪ L[y′_dG(x)];
            Assign f(x) = min {c ∈ {1, . . . , n_x} \ S};
            for r = 1 to d_G(x) do
                L[y′_r ] = f (x) ∪ L[y′_r ];
        else
            i = i + 1;
    return f ;
```

Above algorithm gives an injective coloring of a convex bipartite graph, G = (X, Y, E) using Δ colors in **O(nm)** time. We can actually prove that this algorithm uses optimal colours. The proof involves use of *Lemma* which states that, *let* $\sigma = (x_1, x_2, \ldots, x_{nx}, y_1, y_2, \ldots, y_{ny})$ *be a lex-convex ordering of G then* $x_a y_d$ *and* $x_b y_c \in E$ *with* $x_a, x_b \in X$, $y_c, y_d \in Y$ *and* $a < b$, $c < d$, *implies that* $x_a y_c \in E$. The basic approach to proof of optimality is to assume that the colouring is not injective then do some work and get a contradiction.

4. Research Output

4.1. Approach 1 : Convexity Based

4.1.1 Idea and Algorithm

Idea is to modify previous algorithm so that we can apply it on circular convex bipartite graphs too. Below are the steps to proceed. Let G = (X, Y, E) be any circular convex bipartite graph (convexity over Y), then

Colouring Y:

- Find x_i from X which has maximum degree, let $N_G(x_i)$ be the set of its neighbors
- Consider any end of $N_G(x_i)$ as starting point to color set Y as in previous algo

After coloring of set Y, redefine the ordering of Y as :

- $\{y_i, y_{i+1}, \ldots, y_{ny}, y_1, \ldots, y_{d(x)-ny-i-1}\}$, i.e., overshooting y_{ny} in circular pattern

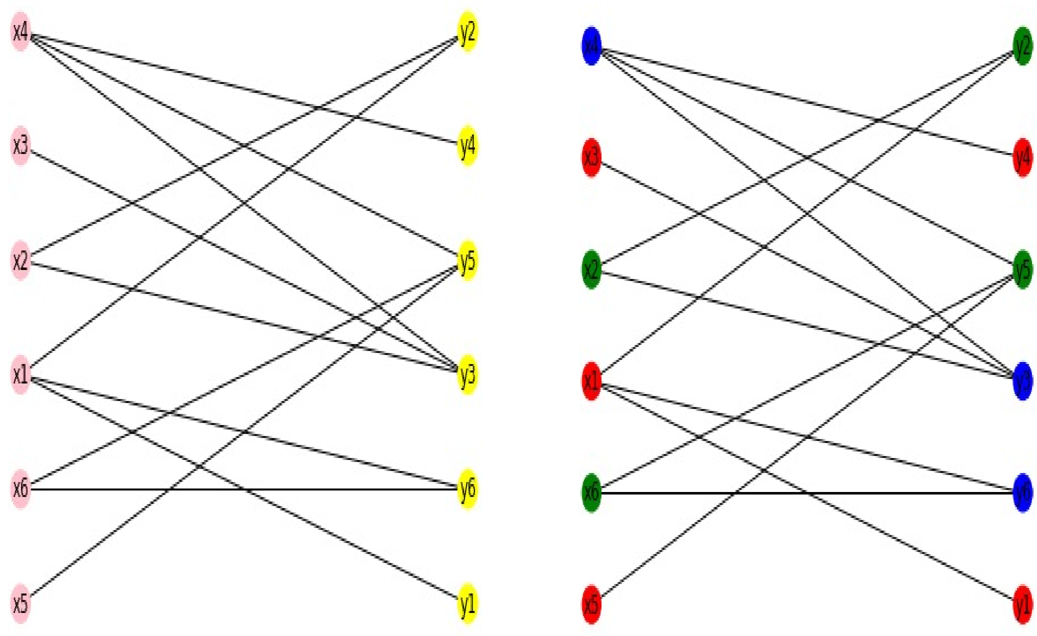
And then color X using the previous algorithm.

4.1.2 Discussion and Results

Discussion:

The motivation to this approach has come from the property that convex bipartite graphs are subsets of Circular convex bipartite graphs. Even though this works on some circular convex bipartite graphs (graph in result 1), it fails in general Circular convex bipartite. The main issue here is that this algorithm is constructed on the assumption that Δ colours are sufficient to colour the vertices in partition Y(convex partition).

Result1 (Approach 1): A circular convex bipartite graph with $\Delta = 3$



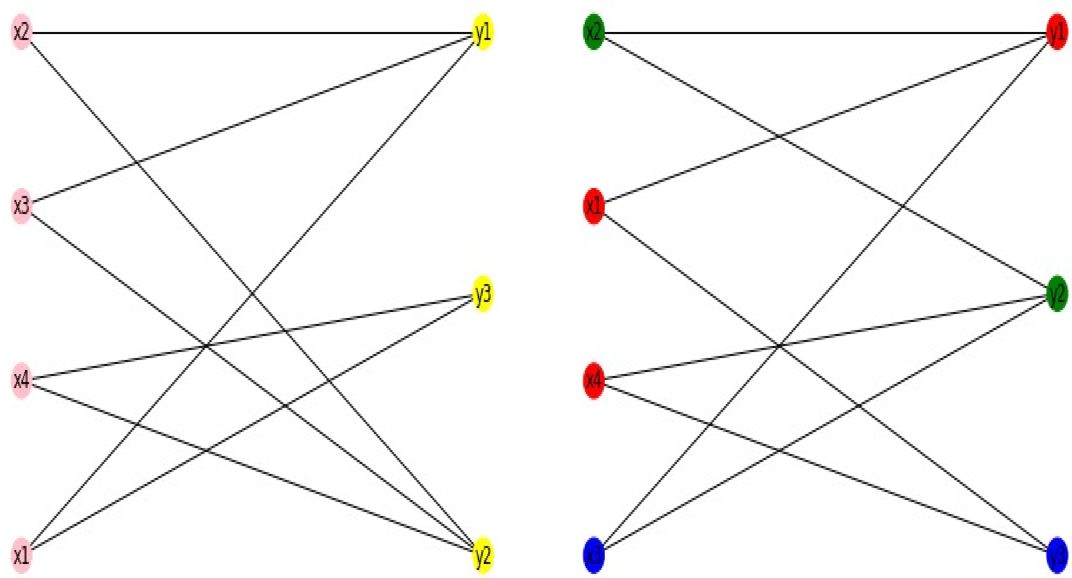
Input **Output**

Injective Colouring of a circular convex bipartite graph given in circular ordering.

Although the algorithm seems to do fine it will be seen in the result 2 that this approach fails. As you can see the vertices x1 and x4 though they share common neighbour y3, they end up getting the same colours. This approach fails because there exists Circular convex bipartite graphs whose Injective Chromatic number can be greater than Δ. On the side note the working time complexity of this approach is $O(mn)$, while its space complexity is $O(n\Delta)$ for maintaining that extra array.

Result 2: Failed!

A circular convex bipartite graph with $\Delta = 3$, but $X_i(G) = 4$**.**



Input **Output**

Injective Colouring of using the proposed algorithm(1)

A circular convex bipartite graph cannot always be coloured using Δ colors injectively. Even though $\Delta = 3$, every vertex in Y are adjacent to each other requiring $|Y| = 4$ colours.

4.2 Approach 2 : Injective Property Perseverance Based

4.2.1 Idea and Algorithm

As we concluded from the discussion of approach 1 that it restricts the no. of colors to be Δ in case of coloring Y. We remove that restriction from Y by colouring them

Similar to that of vertices in partition X i.e., even the vertices of X are equipped with extra arrays to store the colours used by its neighbours and so the vertex y gets the colour

$$\mathbf{f(y) := \min \{c \in \{1, \ldots, n_y\} \setminus S\}}, \text{ where } \mathbf{S = \cup\, L(x_i)\; x_i \in N_G(y)}$$

The algorithm for the same is as shown below:

Final Algorithm : Injective Coloring of Circular Convex Bipartite Graph

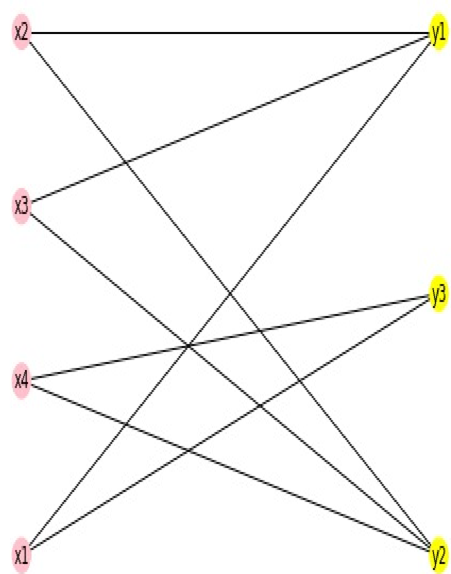
---

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

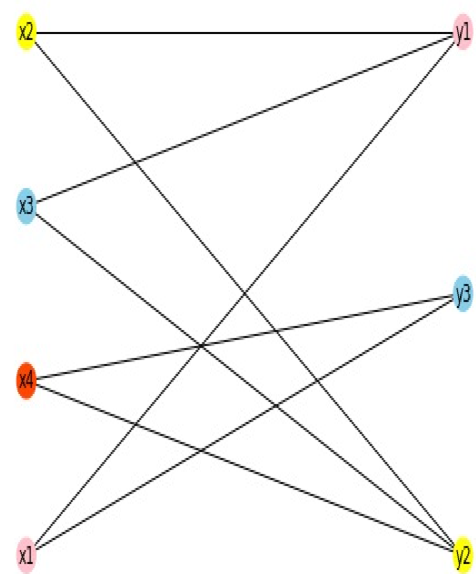
4.2.2 Results and Discussion

Result 1: Previous Counterexample

Chromatic No: 4



Input

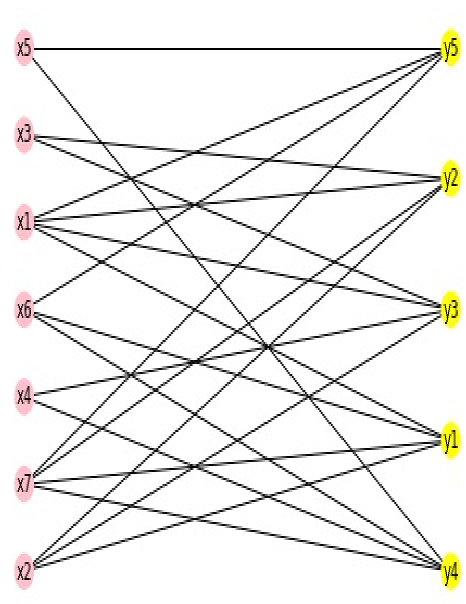


Output

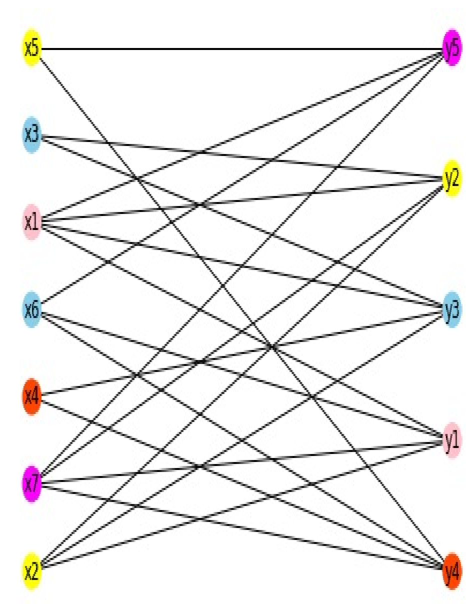
As seen above the algorithm is able to optimally colour(uses 4 colours) the graph which is a counterexample to our approach 1. The Complexity of this algorithm is the same as that of approach 1 which is O(mn) . Also the space complexity is the same, i.e., O(nΔ). Note that *this algorithm works in general for any Bipartite Graph.*

Result 2 :A more complex Circular convex bipartite Graph

Chromatic No: 5



Input



Output

4.2.3 Proof of Correctness

Let $G = (X, Y, E)$ be a bipartite graph.

Note : Since we have colored X and Y in the same manner, proof of correctness is the same for both sets and hence we are proving for X because Y will follow the same.

Consider $x_1, x_2 \in X$ and $y \in Y$ such that $y \in N_G(x_1) \cap N_G(x_2)$ i.e. y is common neighbor of x_1 and x_2.

Claim : $f(x_1) \neq f(x_2)$ where $f(x)$:= denotes the color of x.

Proof:

Define $X_1^o := \{ x^o \in X \mid x^o \in N_G(y^o)\ \forall y^o \in N_G(x_1) \}$ i.e. set of all neighbors of neighbors of x_1. Recall that we have defined

$$\boldsymbol{f(x_1) := min\ \{c \in \{1, . . . , n_x\} \setminus S\}},\ \text{where}\ \boldsymbol{S = U\ L(y_i),\ \ y_i \in N_G(x_1)}$$

It is clear by the definition of X_1^o and S that $S := \{ f(x^o) \mid x^o \in X_1^o \}$ i.e. set of all colors of neighbors of neighbors of x_1.

Claim : $x_2 \in X_1^o$

Since $y \in N_G(x_2) \Rightarrow x_2 \in N_G(y)$, also $y \in N_G(x_1)$ (common neighbor).

Using definition of X_1^o , $x_2 \in X_1^o$ as $x_2 \in X$ such that $x_2 \in N_G(y)$ where $y \in N_G(x_1)$.

Using above claim we proceed as:

$x_2 \in X_1^o$

$\Rightarrow \quad f(x_2) \in S$

$\Rightarrow \quad f(x_2) \notin \{1, . . . , n_x\} \setminus S$

$\Rightarrow \quad f(x_2) \notin min\ \{c \in \{1, . . . , n_x\} \setminus S\}$

$\Rightarrow \quad f(x_2) \neq f(x_1)$

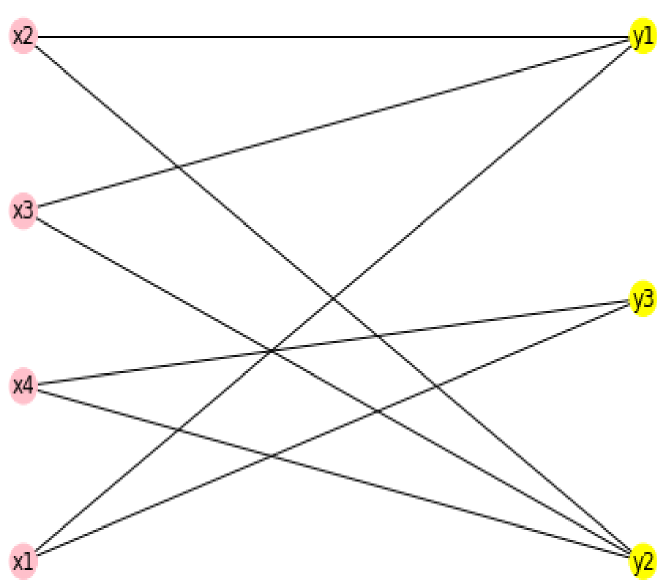
So, we proved that given any two arbitrary x's, if they share a common neighbor then their color is different. Similarly we can prove for set Y and hence we conclude that our algorithm preserves the injective coloring property.

4.3. Final Algorithm Implemented in Python

Input Graph

```
# Define you graph here in adjacency list faishon
# Assume convexity over Y
# Graph 1
X = [
    [1,3],
    [1,2],
    [1,2],
    [2,3]
]
Y = [
    [1,2,3],
    [2,3,4],
    [1,4]
]
```

Visualization of Input Graph



The following is the code based on which this visualization is possible.

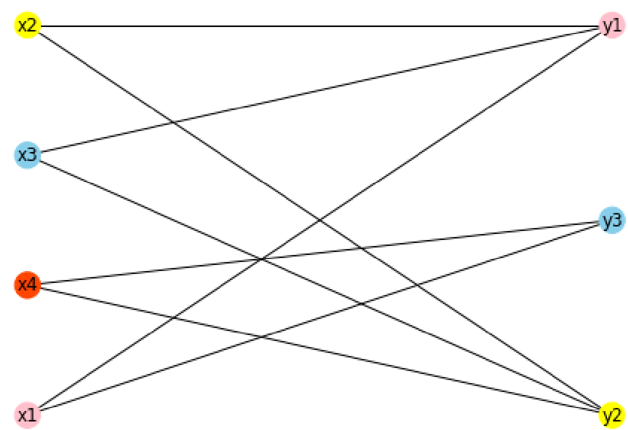
```
# Verify your graph here
_X = ['x'+str(i) for i in range(1,len(X)+1)]
_Y = ['y'+str(i) for i in range(1,len(Y)+1)]
_E = []
for i in range(1,len(X)+1):
    for y in X[i-1]:
        _E.append(('x'+str(i), 'y'+str(y)))

# Potting graph
import networkx as nx
import matplotlib
from networkx.algorithms import bipartite
colors = ['pink','yellow','skyblue','orangered','magenta','cyan','white','blue','green','red','grey','olive']

G = nx.Graph()

G.add_nodes_from(_X, bipartite=0)
G.add_nodes_from(_Y, bipartite=1)
G.add_edges_from(_E)

clr = [colors[0] for i in range(len(X))]
for i in range(len(Y)): clr.append(colors[1])

pos = nx.bipartite_layout(G,_X,align='vertical')
nx.draw(G,pos,with_labels=1, node_color = clr)
```

Below given snippet is the main implemented algorithm for injective coloring.

```
# Finding Injective coloring in this section

colorX = [-1 for i in range(len(X))]
colorY = [-1 for i in range(len(Y))]

# 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

# Getting Injective Chromatic number here
C = set()
for x in colorX: C.add(x)
for y in colorY: C.add(y)
```

Displaying Final injectively colored status of Input Graph and Chromatic no.

```
# Displaying Result here

for x in colorX: clr_map.append(colors[x-1])
for y in colorY: clr_map.append(colors[y-1])

print("Chromatic No : ",len(C))
nx.draw(G,pos,with_labels=1, node_color = clr_map)
```



Final Output

5. Conclusion

Set of all Circular Convex Bipartite graphs contains the set of all Convex bipartite graphs. For any convex bipartite graph the injective chromatic number is Δ. But this is not the case for Circular convex bipartite graphs (i.e. chromatic no of circular convex bipartite graphs can be >= Δ). This is the main reason why approach 1 failed. Approach 2 is correct and works for in general any bipartite graphs as we have given proof of correctness. Our conjecture is that this algorithm (Approach 2) is optimal and hence uses exactly $X_i(G)$ colors for a given graph G. Our conjecture is based on the fact that its optimality in case of convex bipartite graphs can be proven using the same idea as discussed in paper by Prof. B.S Panda and Priyamvada and some examples of circular convex bipartite graphs (any bipartite in general) we worked on during our project. We would also like to point out that the number of colours used by algorithm 2 did not change while starting with different initialization(ordering like lex-convex does not affect the number of colours). Both space and time complexity are the same in both approaches algorithm 1 & 2. Codes are shared in the link given below (open with Google colab or Jupyter Notebook).

https://drive.google.com/file/d/11JlIJT6LDyQv02FTtwLzGaXpT1boSKGY/view?usp=sharing

6. References and Bibliography

1. Fig 1 & 2: https://www.sciencedirect.com/science/article/pii/S0304397514003508
2. Injective coloring of some subclasses of bipartite graphs and chordal graphs B.S. Panda * , Priyamvada1 Department of Mathematics, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India.
3. Domination in some subclasses of bipartite graphs Arti Pandey a , B.S. Panda b, * a Department of Mathematics, Indian Institute of Technology Ropar, Nangal Road, Rupnagar, Punjab 140001, India b Department of Mathematics, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India