Graph Coloring

Jingyi Cui/1738833

Due: 4/17/20

Contents

Problem 1

In this problem, we will talk about graph coloring. Suppose we have n vertices and let vertices be the set $V = \{1, 2, \cdots, n\}$. Let G be the graph and E be the edge set. The edge set is $E = \{(i, j) : sin(i) + sin(j) < 0, 1 \leq i, j \leq n\}$. Now we will have the graph $G = (V, E)$ with n vertices. We want to find the chromatic numbers[1] of G_n versus n. For coloring the graph, two vertices must be different colors if connecting with the same edge. We want to minimize the number of colors we used, with keeping two vertices' colors being different on the same edge.

Suppose we have n colors to color the graph. We use y_i to denote if we will use the color i or not, and we will have only two values for y_i, 0 or 1. x_{ik} denotes whether the vertex i will have color k, and only two values as well. Thus, y_i and x_{ik} will be binary variables:

$$\text{Define } y_i \in \{0, 1\}, i = \{1, \cdots, n\}, \text{where } y_i = 1 \text{ if and only if we use color } k$$
$$\text{Define } x_{ik} \in \{0, 1\}, i = \{1, \cdots, n\}, k = \{1, \cdots, n\}, \text{where } x_{ik} = 1 \text{ iff vertex } i \text{ will have color k}$$

We want to find the chromatic number of our graph. Then we should minimize the number of colors we used. We will have a few constraints to finish this problem.
The first constraint will be that, all the vertices will be colored with exactly one color. It is saying that for all the vertices, a certain color assigns to only one vertex. Thus, if we sum up our n terms, it will be equal to 1.
The second constraint will be that a vertex cannot be colored with an unused color. If $x_{ik} = 1$, y_k will be 1, meaning that we are using the color k.
Constraint (3) is about the constraint of using colors for vertices that are at the end of same edge. The adjacent vertices, which are connected with same edge, must have different colors. If there exists an edge between v_i and v_j, then x_{ik} and x_{jk} must have different colors (they cannot both be one).
The forth constraint is not that necessary: it will help our program run more quickly. If we do not use color 1, we will not use color 2 and we cannot use color 3 if color 2 is not used, etc.
The last constraint will be about defining x_{ik} and y_i. As introduced above, they are binary variables.

Then we will write down the constraints and the objective function in the mathematical formulations:

$$\text{Minimize } \sum_{k=1}^{n} y_k \text{ subject to}$$

1. $\sum_{k=1}^{n} x_{ik} = 1$, $i = 1, \cdots, n$
2. $x_{ik} \leq y_k$, $i = 1, \cdots, n$ and $k = 1, \cdots, n$
3. $x_{ik} + x_{jk} \leq 1$ for all $(v_i, v_j) \in E$, $k = 1, \cdots, n$
4. $y_k \leq y_{k-1}$, $k = 2, \cdots, n$

[1] chromatic number is the smallest number of colors needed to color the whole graph

5. x_{ik}, $y_k \in \{0, 1\}$

I use Java to generate the input file for LPSolve IDE:

```
import static java.lang.Math.sin;
import java.io.PrintStream;
import java.io.IOException;

public class hw2 {
    public static void main(String args[]) throws IOException {
        // creates a file named hw2.txt for saving our output
        PrintStream output = new PrintStream("hw2.txt");

        int n = 6;

        // gets objective function
        System.setOut(output);
        System.out.print("min: ");
        for (int i = 1; i <= n; i++) {
            // gets value for each object
            System.setOut(output);
            System.out.print("+" + "y_" + i);
        }
        System.setOut(output);
        System.out.println(";");

        // first constraint: sum of x_i_k = 1
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= n; j++) {
                System.out.print("+x_" + i + "_" + j);
            }
            System.out.print("= 1;");
        }
        System.out.println();

        // second constraint: x_i_k <= y_k
        for (int i = 1; i <= n; i++){
            System.setOut(output);
            for (int k = 1; k <= n; k++){
                System.out.print("x_" + i + "_" + k + "-y_"
                        + i + "<=0; ");
            }
            System.out.println();
        }
        System.out.println();
```

```
// third constraint: x_i_k + x_j_k <= 1
for (int i = 1; i <= n - 1; i++){
    for (int j = i + 1; j <= n; j++){
        double number;
        number = sin(i) + sin(j);
        // prints all the sin(i) + sin(j)
        // System.out.println("(" + i + ", " + j + ") and"
        // + number);

        // if sin(i) + sin(j) < 0
        if (number < 0){
            for (int k = 1; k <= n; k++){
                System.out.print("x_" + i + "_" + k
                        + "+x_" + j + "_" + k + "<= 1;");
                System.out.println();
            }
        }
    }
}
System.setOut(output);
System.out.println();

// forth constraint: y_k <= y_k-1 meaning y_k - y_k-1 <= 0
for (int i = 2; i <= n; i++){
    System.setOut(output);
    System.out.print("y_" + i + "-y_" + (i - 1) + "<=0; " );
}
System.setOut(output);
System.out.println();

// fifth constraint: x_i_k and y_i are binary variables
System.setOut(output);
System.out.print("bin ");
for (int i = 1; i <= n; i++) {
    System.setOut(output);
    System.out.print("y_" + i + ", ");
}
for(int i = 1; i <= n; i++){
    System.setOut(output);
    for (int m = 1; m <= n; m++){
        if(i == n && m == n){
            System.out.print("x_" + n + "_" + n + "; ");
        }else {
            System.out.print("x_" + i + "_" + m + ", ");
        }
    }
```

```
            }
        }
}
```

The result is:

min : $+y\_1 + y\_2 + y\_3 + y\_4 + y\_5 + y\_6$;
(1) (6 lines of the following type: ensure that every vertex has exactly one color)
$+x\_1\_1 + x\_1\_2 + x\_1\_3 + x\_1\_4 + x\_1\_5 + x\_1\_6 = 1$;
.
.
(2) (36 lines of the following type: y tracks what are color are assigned to the vertex)
$x\_1\_1 - y\_1 \leq 0$;
.
.
(3) (48 lines of the following type: the edge set corresponded to the requirement $sin(i) + sin(j) < 0$ is $(1,5),(2,5),(3,4),(3,5),(4,5),(3,6),(4,6),(5,6)$. Vertices connected with the same edge are used by different colors)
$x\_1\_1 + x\_5\_1 \leq 1$;
.
.
(4) (smaller numbered colors are used before larger numbered colors, four lines of the following type)
$y\_2 - y\_1 \leq 0$;
.
.
(5) (ensure all the variables are binary)
bin $y\_1, \cdots, y\_6, x\_1\_1, \cdots, x\_6\_6$;

Using the above input, I solved the LP using LPSolve IDE. The entire time I used is 0.047 seconds, which is really short. We get the result:

Value of objective function: 4.00000000

Actual values of the variables:

y_1 1

y_2 1

y_3 1

y_4 1

x_1_1 1

x_2_1	1
x_3_1	1
x_4_2	1
x_5_3	1
x_6_4	1

The result shows that the smallest number of the colors we use is 4. And for vertex $1, 2, 4$, we will use color $1(x\_1\_1, x\_2\_1, x\_3\_1)$. For the vertex 4, we use the color $2(x\_4\_2)$. For the vertex 5, we use the color $3(x\_5\_3)$. For the vertex 6, we use the color $4(x\_6\_4)$.[2]

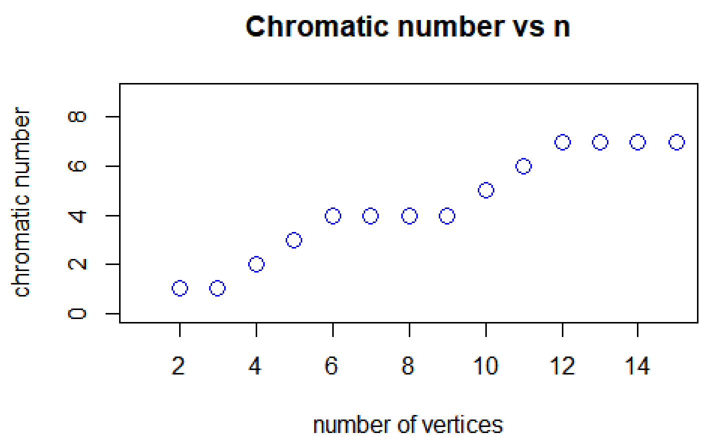
When I try to use $n = 15$, the running time is 123.793 seconds. However, if I try $n = 14$, the total time decreases a lot: 59.268 seconds. I will use $n = 15$ to look at the degrees of the vertices of G_n. The edge set is $E = \{(1, 5), (1, 11), (2, 5), (2, 11), (3, 4), (3, 5), (3, 6), (3, 10), (3, 11), (3, 12), (4, 5), (4, 6), (4, 7), (4, 9), (4, 10), (4, 11), (4, 12), (4, 13), (4, 15), (5, 6), (5, 7), (5, 9), (5, 10), (5, 11), (5, 12), (5, 13), (5, 15), (6, 10), (6, 11), (6, 12), (7, 11), (8, 11), (9, 10), (9, 11), (9, 12), (10, 11), (10, 12), (10, 13), (11, 12), (11, 13), (11, 14), (11, 15), (12, 13)\}$.

Vertex 1 has degree 2. Vertex 2 has degree 2; vertex 3 has degree 6. Vertex 4 has degree 10. Vertex 5 has degree 12. Vertex 6-15 has degree 6, 3, 1, 5, 8, 14, 8, 5, 1, 3, respectively.

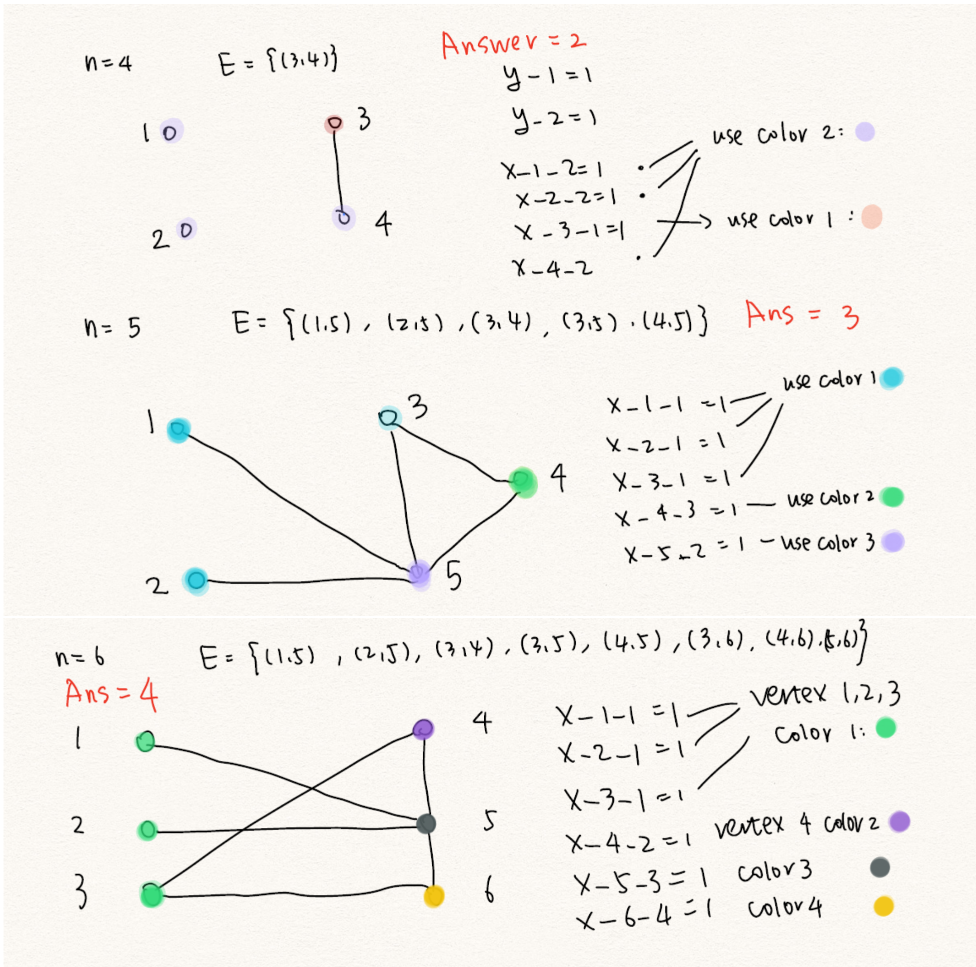
The largest degree is 14 of vertex 11. The smallest degree is 1 of vertex 8 and vertex 14, since $sin(8)$ and $sin(14)$ is two of the largest values in $sin(i), i = 1, 2, \cdots, 15$. $sin(8)$ and $sin(14$ is approaching to 1. And $sin(11)$ is the smallest one: it is approaching to -1. The requirement for edge set is $sin(i) + sin(j) < 0$. For values that approaches to 1, it is hard to find a number to make the sum be negative. For values that approaches to -1, it is easy to find a number to let the sum be negative, since the range of $sin(i), \forall i$ is from -1 to 1.

[2] see page 8 for the figure of G_6

I also draw the chromatic number of G_n versus n. I use n from 2 to 15.



The following figure I draw of G_n for different n: $n = 4, 5, 6$.



We can use a greedy coloring algorithm to show the chromatic number is less than or equal to $\Delta(G) + 1$, where $\Delta(G)$ is the largest degree of vertices in G_n. For $n = 4$, the largest degree is 1 of vertex 3 and vertex 4. The chromatic number is 2, which is less than or equal to 1 + 1 = 2. For $n = 5$, the largest degree is 4 of vertex 5. The chromatic number is 3, which is less than 4 + 1 = 5. For $n = 6$, the largest degree is 5 of vertex 5. The chromatic number is 4, which is less than 5 + 1 = 6.