



Cairo university  
Faculty of computers and information  
Decision support department

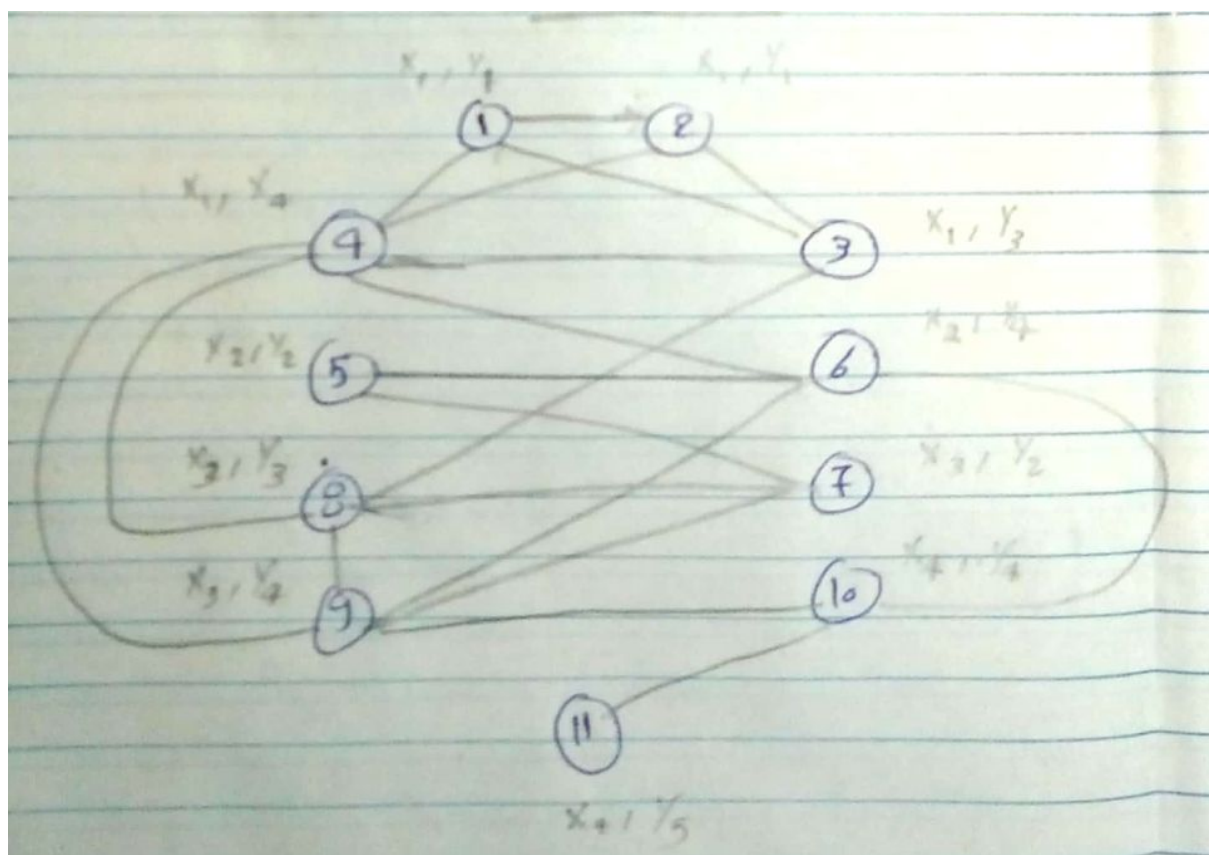
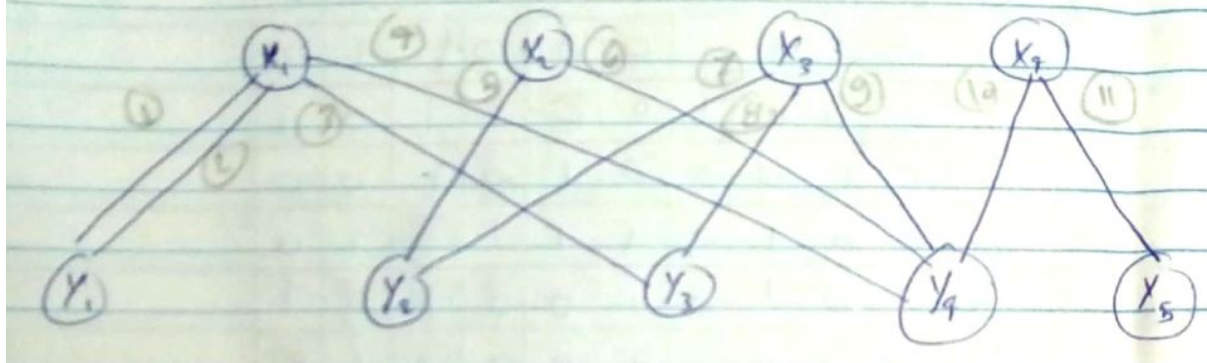
**Name:** Ahmed Ehab Hussein

**Id:** 20160007

**Group:** CS\_DS\_1

Construct the line graph  $L(G)$  and the adjacency matrix of  $L(G)$ .

P	$x_1$	$x_2$	$y_2$	$x_4$	$y_3$
$x_1$	2	0	1	1	0
$x_2$	0	1	0	1	0
$x_3$	0	1	1	1	0
$x_4$	0	0	0	1	1



[illegible]

Model the problem as a classic Minimum Vertex Coloring problem, and use JuMP to find a minimum proper 4-coloring of the vertices of  $L(G)$ .

```
using JuMP
using GLPK
```

```
mat = [
    0 1 1 1 0 0 0 0 0 0 0;
    1 0 1 1 0 0 0 0 0 0 0;
    1 1 0 1 0 0 0 0 0 0 0;
    1 1 1 0 0 1 0 0 1 1 0;
    0 0 0 0 0 1 1 0 0 0 0;
    0 0 0 1 1 0 0 0 1 1 0;
    0 0 0 0 1 0 0 1 1 0 0;
    0 0 1 0 0 0 1 0 1 0 0;
    0 0 0 1 0 1 1 1 0 1 0;
    0 0 0 1 0 1 0 0 1 0 1;
    0 0 0 0 0 0 0 0 0 1 0;
]
```

```
rows = size(mat, 1)
cols = size(mat, 2)
```

```
model = Model(optimizer_with_attributes(GLPK.Optimizer,
    "tm_lim" => 60000, "msg_lev" => GLPK.OFF))
```

```
@variable(model, color[1:cols], Bin)
@variable(model, v[1:rows, 1:cols], Bin)
```

```
@objective(model, Min, sum(color))
```

```
# make sure all vertices must be colored with at least one
color
```

```
@constraint(model, [i=1:rows], sum(mat[i, h] * v[i, h] for
h=1:cols) == 1)
```

```
# make sure any two adj nodes can't have the same color
for i=1:rows
```

```
    if sum(mat[i,:]) > 0
```

```
        @constraint(model, sum((mat[i, j] * v[i, j]) -
color[j] for j=1:cols if mat[i, j] > 0) <=0)
```

```
        end
```

```
    end
```

```
optimize!(model)
```

```
println("Model:")
```

```

println(model)

for i=1:rows
    for j=1:cols
        if value(v[i, j]) > 0
            println("v[$i, $j] = ", value(v[i, j]))
        end
    end
end

println()
for i in 1:cols
    println("color[$i] = ", value(color[i]))
    if value(color[i]) > 0
        println("This color will use")
    end
end

println("-" ^ 50)
println("Objective value = ", objective_value(model))
println("-" ^ 50)

```

## Output

Model:

Min color[1] + color[2] + color[3] + color[4] + color[5] +  
color[6] + color[7] + color[8] + color[9] + color[10] +  
color[11]

Subject to

$v[1,2] + v[1,3] + v[1,4] = 1.0$   
 $v[2,1] + v[2,3] + v[2,4] = 1.0$   
 $v[3,1] + v[3,2] + v[3,4] = 1.0$   
 $v[4,1] + v[4,2] + v[4,3] + v[4,6] + v[4,9] + v[4,10] = 1.0$   
 $v[5,6] + v[5,7] = 1.0$   
 $v[6,4] + v[6,5] + v[6,9] + v[6,10] = 1.0$   
 $v[7,5] + v[7,8] + v[7,9] = 1.0$   
 $v[8,3] + v[8,7] + v[8,9] = 1.0$   
 $v[9,4] + v[9,6] + v[9,7] + v[9,8] + v[9,10] = 1.0$   
 $v[10,4] + v[10,6] + v[10,9] + v[10,11] = 1.0$   
 $v[11,10] = 1.0$   
 $v[1,2] - \text{color}[2] + v[1,3] - \text{color}[3] + v[1,4] - \text{color}[4] \leq 0.0$   
 $v[2,1] - \text{color}[1] + v[2,3] - \text{color}[3] + v[2,4] - \text{color}[4] \leq 0.0$   
 $v[3,1] - \text{color}[1] + v[3,2] - \text{color}[2] + v[3,4] - \text{color}[4] \leq 0.0$   
 $v[4,1] - \text{color}[1] + v[4,2] - \text{color}[2] + v[4,3] - \text{color}[3] +$   
 $v[4,6] - \text{color}[6] + v[4,9] - \text{color}[9] + v[4,10] - \text{color}[10] \leq 0.0$   
 $v[5,6] - \text{color}[6] + v[5,7] - \text{color}[7] \leq 0.0$   
 $v[6,4] - \text{color}[4] + v[6,5] - \text{color}[5] + v[6,9] - \text{color}[9] +$   
 $v[6,10] - \text{color}[10] \leq 0.0$   
 $v[7,5] - \text{color}[5] + v[7,8] - \text{color}[8] + v[7,9] - \text{color}[9] \leq 0.0$   
 $v[8,3] - \text{color}[3] + v[8,7] - \text{color}[7] + v[8,9] - \text{color}[9] \leq 0.0$   
 $v[9,4] - \text{color}[4] + v[9,6] - \text{color}[6] + v[9,7] - \text{color}[7] +$   
 $v[9,8] - \text{color}[8] + v[9,10] - \text{color}[10] \leq 0.0$   
 $v[10,4] - \text{color}[4] + v[10,6] - \text{color}[6] + v[10,9] - \text{color}[9]$   
 $+ v[10,11] - \text{color}[11] \leq 0.0$   
 $v[11,10] - \text{color}[10] \leq 0.0$   
color[1] binary  
color[2] binary  
color[3] binary

color[4] binary  
color[5] binary  
color[6] binary  
color[7] binary  
color[8] binary  
color[9] binary  
color[10] binary  
color[11] binary  
v[1,1] binary  
v[2,1] binary  
v[3,1] binary  
v[4,1] binary  
v[5,1] binary  
v[6,1] binary  
v[7,1] binary  
v[8,1] binary  
v[9,1] binary  
v[10,1] binary  
v[11,1] binary  
v[1,2] binary  
v[2,2] binary  
v[3,2] binary  
v[4,2] binary  
v[5,2] binary  
v[6,2] binary  
v[7,2] binary  
v[8,2] binary  
v[9,2] binary  
v[10,2] binary  
v[11,2] binary  
v[1,3] binary  
v[2,3] binary  
v[3,3] binary  
v[4,3] binary  
v[5,3] binary  
v[6,3] binary  
v[7,3] binary  
v[8,3] binary  
v[9,3] binary  
v[10,3] binary  
v[11,3] binary  
v[1,4] binary

v[2,4] binary  
v[3,4] binary  
v[4,4] binary  
v[5,4] binary  
v[6,4] binary  
v[7,4] binary  
v[8,4] binary  
v[9,4] binary  
v[10,4] binary  
v[11,4] binary  
v[1,5] binary  
v[2,5] binary  
v[3,5] binary  
v[4,5] binary  
v[5,5] binary  
v[6,5] binary  
v[7,5] binary  
v[8,5] binary  
v[9,5] binary  
v[10,5] binary  
v[11,5] binary  
v[1,6] binary  
v[2,6] binary  
v[3,6] binary  
v[4,6] binary  
v[5,6] binary  
v[6,6] binary  
v[7,6] binary  
v[8,6] binary  
v[9,6] binary  
v[10,6] binary  
v[11,6] binary  
v[1,7] binary  
v[2,7] binary  
v[3,7] binary  
v[4,7] binary  
v[5,7] binary  
v[6,7] binary  
v[7,7] binary  
v[8,7] binary  
v[9,7] binary  
v[10,7] binary



v[11,7] binary  
v[1,8] binary  
v[2,8] binary  
v[3,8] binary  
v[4,8] binary  
v[5,8] binary  
v[6,8] binary  
v[7,8] binary  
v[8,8] binary  
v[9,8] binary  
v[10,8] binary  
v[11,8] binary  
v[1,9] binary  
v[2,9] binary  
v[3,9] binary  
v[4,9] binary  
v[5,9] binary  
v[6,9] binary  
v[7,9] binary  
v[8,9] binary  
v[9,9] binary  
v[10,9] binary  
v[11,9] binary  
v[1,10] binary  
v[2,10] binary  
v[3,10] binary  
v[4,10] binary  
v[5,10] binary  
v[6,10] binary  
v[7,10] binary  
v[8,10] binary  
v[9,10] binary  
v[10,10] binary  
v[11,10] binary  
v[1,11] binary  
v[2,11] binary  
v[3,11] binary  
v[4,11] binary  
v[5,11] binary  
v[6,11] binary  
v[7,11] binary  
v[8,11] binary

v[9,11] binary  
v[10,11] binary  
v[11,11] binary

v[1, 2] = 1.0  
v[2, 1] = 1.0  
v[3, 1] = 1.0  
v[4, 1] = 1.0  
v[5, 6] = 1.0  
v[6, 4] = 1.0  
v[7, 5] = 1.0  
v[8, 3] = 1.0  
v[9, 4] = 1.0  
v[10, 4] = 1.0  
v[11, 10] = 1.0

color[1] = 0.0  
color[2] = 0.0  
color[3] = 0.0  
color[4] = 1.0  
This color will use  
color[5] = 0.0  
color[6] = 1.0  
This color will use  
color[7] = 0.0  
color[8] = 0.0  
color[9] = 1.0  
This color will use  
color[10] = 1.0  
This color will use  
color[11] = 0.0

-----  
Objective value = 4.0  
-----