CAAM 574: Combinatorial Optimization
Computer Group Project

Objective: Your objective is to implement a computer code to solve the Maximum Cut Problem (MCP) in C++ or Python (numbers start counting at 0 instead of 1). Only use outside source code or library for planarity testing (if necessary).

Due Date: November 30th! The research report and a 15 minute presentation should be given in class and the source files should be mailed to me at ivhicks@rice.edu.

Minimum Requirement (100 points):

- **Option1:** Create a branch-and-cut solver using Gurobi to compute upper bounds, a simple greedy algorithm for computing lower bounds (extra points for more complicated algorithms), and a combinatorial algorithm for the separation algorithm (finding valid cuts).

- **Option 2:** Create a purely combinatorial solver for finding the max cut of a planar graph as the input graph. This involves planarity testing, finding a planar dual, and developing a purely combinatorial min-weight perfect matching solver; **no integer programming solver or outside code (except planarity testing) allowed.**

- **Option 3:** Derive your own analogue version of Östergard's algorithm for maximum clique for the maximum cut problem.

I/O Requirements: The code must read in a file through standard input. The input file will look like the following:

```
n    m  //# of nodes and # of edges
end1 end2  weight  //for edge(0)
. . .
end1 end2 weight //for edge(m-1)
```

Standard output must be the ends of the edges chosen with their cost and the cost of the best cut found like the following:

end1 end2 weight //for edge(0) of cut
. . .
end1 end2 weight //for edge(n-1) of cut
The cost of the best cut is: (the cost the best cut)

Test Instances: The test instance files will be available on canvas. Below are the names and the optimal costs of the tours:

Nonplanar instances:
att48: 589313 gr21: 49892 hk48: 771712 ulysses22: 117119

Planar instances:
a280.del: 12330 bier127.del: 375761 ch130.del: 22567 ch150.del: 22549
d198.del: 79438 d493.del: 129744 d657.del: 199616 d1291.del: 547939

The Research Report: The group must also turn in a detailed report entailing an intro, a description of the problem, the steps of your algorithm, computational results (i.e. running times) of your code on the five problems and concluding remarks. Cite relevant background. Also email me a critique (5 highest to 1 lowest) of your group members and their participation in the project.

Technical Support: There are primers in the computer lab or online and you are welcome to ask me for help. Good luck!