Weekly Report 9
Yiduo Ke
ORIE 4999 Supervised by Professor Williamson

## What I did

Turns out I did NOT read the paper wrong last week, and that I DID need to do a Cholesky decomposition on the maximized objective. I did need to fix the issue with not being able to do Cholesky decomposition with the maximized objective. Before, I thought I could "fix" the issue by doing the pivoted version of Cholesky decomposition, but that turned out to be wrong, because it would ignore any errors (I turned on error suppression) and do a faulty calculation. I printed my matrices' eigenvalues to see what's going on, and turns out, some of them are verrrrry slightly negative, so the matrices are "almost" positive semidefinite. Prof. Williamson gave the most important tip of adding a very small multiple of the identity matrix to make those eigenvalues zero, and thus allowing the matrices to be Cholesky decomposed. This worked! And the test case results reflect that.

I was also told by Prof. Williamson to start testing on much larger graphs. I did that. Of course, using the brute force (i.e. true optimum) is not feasible on large graphs. I instead used the maximized objective as an upper bound to the optimum (once again Prof. Williamson's advice), which gave very rough estimates on what the Greedy, Coin Flip, SDP, and Trevisan results should look like. The results are favorable! Test_results.txt contains the logs of the tests.

## What I will do Next

Have everything converted to weighted graph version. This is not a trivial task because right now, the graphs are in adjacency list form, which in Julia is a list of Dictionaries where each key is a vertex and each value is the vertex's neighboring vertices:

```
graph_3 = Dict(1=>[2,3], 2=>[1,4,5], 3=>[1,6,7], 4=>[2], 5=>[2], 6=>[3], 7=>[3])
```

I don't think it would make sense to represent weighted graphs in this same form, because then there would need to be nested dictionaries, with each inner Dictionary key being a neighboring vertex of the outer vertex/key, and inner Dictionary value being the value of the edge between the inner vertex and the outer vertex. I am henceforth requiring all input graphs of my MAX-CUT functions of the weighted graph version be in the adjacency matrix form.