COMP590: Homework 1

Jason Isaacs
Assigned: February 14, 2016
Due: March 03, 2016

Problem 1: Show that the number of edges in any graph is half the sum of the degrees of its nodes. Conclude that the $Trace$ of $L(\mathcal{G})$ is always an even number and that the number of odd degree nodes in any graph has to be even.

Problem 2: The degree sequence for a graph is a listing of the degrees of its nodes; thus K_3 has the degree sequence $2, 2, 2$. Is there a graph with the degree sequence $3, 3, 3, 3, 5, 6, 6, 6, 6, 6, 6$? How about with the degree sequence $1, 1, 3, 3, 3, 3, 5, 6, 8, 9$?

Problem 3: The complement of graph $\mathcal{G} = (V, E)$, denoted by $\overline{\mathcal{G}}$, is a graph $(V, \overline{E})$, where $uv \in \overline{E}$ if and only if $uv \notin E$. Show that

$$L(\mathcal{G}) + L(\overline{\mathcal{G}}) = nI - \mathbf{1}\mathbf{1}^T. \tag{1}$$

Reminder:
$\mathbf{1}$: $n \times 1$ vector of all ones
I: $n \times n$ identity matrix

Problem 4: Given the adjacency matrix $A(\mathcal{G})$ as an input. Write an algorithm (in your favorite language) that checks whether the graph is connected. This is the exercise that we worked on in class. Your algorithm should not include checking the second eigenvalue of the Laplacian. Rather you should use Theorem 2.8 from Mesbahi and Egerstedt only as a way to check your result.

Problem 5: Simulate the agreement protocol

$$\dot{x}(t) = -L(\mathcal{G})x(t) \tag{2}$$

for a graph on five vertices. Compare the rate of convergence of the protocol as the number of edges increases. Does the convergence of the protocol always improve when the graph contains more edges? Analyze the eigenvalues of Laplacian matrix to support your observation.

Hint:

- Start with the same set of initial conditions for each trial.

- Start with a tree graph.
- Simulate the agreement protocol.
- Measure and record the convergence time (agreement to within say 2%).
- Record the eigenvalues of $L(\mathcal{G})$ using eig() function in MATLAB.
- Add a single edge and repeat until you have a complete graph.

What if you had added edges in a different order? Would that have changed your results? Why or why not?

Bonus Question: Submit your code for implementing Reynolds Flocking that we worked on in week 1. If you want to revisit this problem, the algorithm description can be found at: http://www.red3d.com/cwr/boids/.