CSE 7350/5350

David W. Matula

Wireless Sensor Network Project

Total Points: 400

Updated: Fall 2015

Due Date: 14 December 2015

Early: 10 point bonus for Monday, Dec 7, 11:59pm CST

Latest: Tuesday, Dec 15 (4pm) CST

Note: After Tuesday, Dec 15 at 4pm, CST late projects will incur a penalty with possible delay of grade. Have your project TIME STAMPED.

1 Introduction and Summary

In this project you are asked to implement an algorithm for determining a coloring, terminal clique, and a selection of bipartite subgraphs that are produced by an algorithm for graph coloring in a random geometric Graph (RGG). These results model a wireless sensor network (WSN) with each bipartite subgraph providing a communication backbone. You are first asked in Part I to prepare several RGG's as a randomized benchmark set. RGG's on the plane (unit square), model WSN's on geographic regions and RGG's on the whole sphere model WSN's that span the globe. In Part I you are further asked to implement the smallest last coloring algorithm for vertex coloring and terminal clique identification in the RGG's. The coloring algorithm is to be applied to the randomized benchmark set and is also to be applied to several original graphs you are asked to create to exhibit the strengths and weaknesses of the coloring and clique determination algorithms particularly for a few considerably larger RGG's.

Regarding algorithm efficiency you are asked to verify a linear time bound O (|V|+|E|) for smallest last graph coloring and suitable time and space bounds for each algorithm you implement for preparing the RGG. You are asked to document your implementations in the form of a report to a technical manager with substantial use of graphic display of results. Section 3 provides more details on the grading issues and the format for the project report. Section 4 provides details on the RGG generation and graph coloring procedure implementation requested as Part I of the project. Specifications are given in Section 5 for determining a partition into bipartite "backbones" involving a majority of the sensors as requested in Part II.

2 Benchmarks

A Randomized Case Benchmark of 10 particular random geometric graphs output from Part I should be part of your graph set input to your Backbone Selection Program in Part II. Your program must find a smallest last vertex ordering, vertex coloring, terminal clique, and several bipartite backbones for each of these ten benchmark graphs as well as the graphs you create as requested in Parts I and II.

3 Project Format

Your project submission should be equivalent to a 10 -15 page word-processor-composed report. This should be initiated by a 2 - 3 page **Executive Summary** containing an **Introduction and Summary**, a **Programming Environment Description**, and **References** as described in the following. The core of your report should contain the **Reduction to Practice** details and **Result Summary** as outlined in the following. In style, the report should be oriented towards a technical manager with a wide perspective, say in *Scientific American* magazine style. Grading partition:

3.1 Executive Summary

Introduction and Summary (50 points)

Give a brief description, in layman's terms, of the project, **including your major result**. Consider this section to be a professionally worded *marketing* effort. That is, would the competent technical manager (your audience) be likely to want to read on and try to understand your claimed good results. **Describe the strongest features of your implementation**. Summarize your use of illustrations, figures, tables, and graphics employed that will reveal your results effortlessly to the reader. The summary should **include a table of results** for the 10 benchmarks described in Section 4. Give clear citations (e.g. [1]) to your sources listed in the **References** section including downloaded programs and previous student's projects that most influenced your work.

Programming Environment Description (40 points)

Give a description of the environment you used to develop the program including both hardware and software components, i.e., the specific type and brand of computer, the computer's processing speed, the amount of memory, the operating system, the language, any graphics tools, and special libraries, etc. The description should include metrics on the program resource utilization allowing judgment of "was the total used respectable or an overkill?" The purpose of this description is to allow comparisons and facilitate reproduction. Consider: would a knowledgeable colleague be able to judge the scope of your work from this description of the environment and resource utilization? The graphics package and languages used to present the report and used to create the graph drawings and displays should also both be described. Discuss the interactions between system components you needed to establish to effect both computational efficiency and effective result display.

References (30 points)

Provide an enumerated list of publications, texts, websites, programs and resource packages in typical reference style as a concluding part of the executive summary. You should include references to several papers related to WSN's, backbones in WSN's, RGG's, graph coloring algorithms, the Smallest Last algorithm and to the major tools used in developing the report. For example:

[1] D.W. Matula, Wireless Sensor Network Project, www.lyle.smu.edu/cse/7350/, 2014

IMPORTANT: If you read previous student projects from the archives in the department office or online regarding the smallest last coloring implementation, backbone determination, and displays, you should include a reference to each of them in your reference list.

3.2 Wireless Sensor Network Backbone Report

Reduction to Practice (120 points)

This is the core of your report and should cover the five following topics appropriately integrated.

• *Data structure Design*: Describe and illustrate the representation and organization of data employed and discuss the relation to the algorithms employed for both Parts I and II.

• *Algorithm Descriptions*: Provide a description of the smallest last vertex ordering and graph coloring algorithm sufficient for your report to be self-contained to someone familiar with other graph search algorithms.

• *Algorithm Engineering*: Describe how the algorithm implementation is engineered to achieve efficiencies in time and space and provide arguments for the efficiencies. In particular, carefully substantiate the linear time and appropriate optimal space bounds for the smallest last vertex ordering, and vertex coloring, in each case where your implementation realizes this bound.

• *Verification Walkthrough*: Present a walkthrough for an RGG in the unit square with N=20 vertices and R=0.40. Draw the graph, and walk through the Smallest Last vertex ordering, the graph coloring and the determination of the backbone for the vertices colored with the first and second colors. Display the backbone bipartite subgraph illustrating the domination provided by the first color set and the planarity of the bipartite subgraph. Plot the degree when deleted values and the original degree values for the vertices ordered by the Smallest Last order.

• *Algorithm Effectiveness*: Include here a summary table of results as an overview of the benchmarks and your additional test graphs. Give your conclusion on the strengths and weaknesses of the RGG generation, coloring, and bipartite backbone selection algorithms on the one hand and your implementation on the other. Indicate applications where your implementation would be strongest and what kind of input would be most difficult to handle. Your analysis and conclusions relevant to your separately created test graphs can be a focal point of this topic.

Benchmark Result Summary and Display (160 points)

You should provide a standard output for each randomized benchmark graph created in Part I as specified in Section 4. You should provide output for each bipartite partition in Part II as specified in Section 5.

These points are awarded on the basis of your overall use of illustrations, figures, tables, bar charts, and computer generated diagrams in your report. Hand drawn and/or computer generated figures may be utilized as appropriate.

4 Part I: (a) Random Geometric Graph Generation and Display

This subproject utilizes pseudo random number generation to create several types of random geometric graphs. A variety of methods possessing different time and space requirements may be employed to prepare the output adjacency list data structure.

Input: N = number of sensors,

A (estimate) = estimated average degree

R = distance bound for adjacency,

T = graph type (square, disk, or sphere).

Output: N = number of sensors (vertices).

M = number of distinct pairwise sensor adjacencies (edges).

R = distance bound for adjacency,

A (realized) = average degree

Degree distribution: Plot the number of vertices having degree *i* as a function of *i* for *i* = 0,1,…, max degree.

RGG Display: Provide a display of the vertices of the RGGs you generate, highlighting the vertices having the minimum degree, the maximum degree. Also for one minimum degree vertex and one maximum degree vertex for Benchmarks 1-3, highlight all the neighbors. This can be done by drawing edges to the neighbors, or by highlighting the neighbors, or by shading the disk of radius R around the specified min-degree (maxdegree) vertex. Draw all the edges only if the average degree is not too large, e.g. $\leq$ 30.

General Procedure: For planar input data generation we shall pick N points in two dimensions according to some specified distribution using pseudo random number generation, and determine the pairs of points within distance R of each other. Various types of summary information should be presented as requested. The graph adjacency structure should be produced and saved for input to the coloring and backbone selection procedures. Develop the random geometric graph's adjacency list data structure for the 10 benchmark data sets created as specified, and also for two-to-four more such cases that illustrate your implementation features in size and "quality".

Uniform Square The points are uniform over the unit square, $0 \leq x \leq 1$ and $0 \leq y \leq 1$.

Uniform Unit Disk The points are uniform over the disk (circle) of radius unity.

Uniform Sphere The points are uniform over the surface of the unit radius sphere. The points may be determined by first picking three uniform random x, y, z values each in the interval [-1, 1], then if the point x, y, z is within unit distance from the origin, normalizing the point to a boundary point on the sphere by dividing each coordinate by the distance of the point from the origin. You may implement the distance check for edges by determining if the normalized x, y, z coordinates for each pair of points on the surfaces of the sphere are within distance R.

(b): Graph Coloring and Bipartite Backbone Selection

This subproject utilizes the smallest-last coloring algorithm to provide a vertex coloring of the graph to implement the backbone selection procedure.

Input: A graph in adjacency list form.

Output: Properties of the coloring of the vertices of the graph, suitably presented to an audience that wants to visualize and understand the results for possible alternative applications of the program as implemented.

(c) Tables and Displays in Report

Your report should include the following for each graph, along with any additional information you feel would "sell" your implementation.

i. Sequential Coloring Plot

For all graphs provide a plot of the degree when deleted function in the smallest last order. You should also show in the same plot the corresponding original degree (upper bound function).

ii. Color Class Size Distribution

For all graphs provide a plot of the size distribution table for the independent sets of vertices colored with color *j* for *j* = 1, 2, …, maxcolor.

Graph Coloring Summary Table

Provide a summary table for coloring results on your benchmark RGGs including for each graph: ID-number, N (number of vertices), R, Distribution type, M (number of edges), min degree, avg. degree, max degree, max min-degree when deleted, number of colors, max color class size (size of the largest color classes), terminal clique size, number of edges in the largest connected bipartite subgraph determined (the backbone), and the percentage of vertices covered by the backbone. For RGG's on the sphere also give the corresponding number of faces for the largest connected bipartite subgraph. These results should be provided also for your own example graphs.

Backbone Selection Procedures and Summary Table

Bipartite backbone selection by independent set pairings: For the first four largest color classes of each of the Benchmark graphs, determine which two of the six possible bipartite subgraphs have the most edges in their maximum connected subgraph (giant component, termed the "backbone").

For each benchmark and each of the two backbones of the benchmark provide the number of vertices, number of edges and percentage of vertices covered (domination percentage) by the vertices of the backbone. Provide these results in a small table in your executive summary and in the report include these results with drawings of the backbones for the square and disk graphs. For the graphs on the sphere also determine the number of faces in the resulting planar connected bipartite subgraphs and draw a hemisphere projection of the "best" bipartite "backbone" showing points and edges.

For those graphs on 1000 and 4000 vertices draw the vertex set of the graph and the full graph with edges.

Benchmark Data Sets

Benchmark	N	Avg. Degree	Distribution
1	1000	30	Square
2	4000	40	Square
3	4000	60	Square
4	16000	60	Square
5	64000	60	Square
6	4000	60	Disk
7	4000	120	Disk
8	4000	60	Sphere
9	16000	120	Sphere
10	64000	120	Sphere