CONSTRUCTION ALGORITHMS FOR EXPANDER GRAPHS

Author: Vlad Stefan Burca
Advisor: Professor Takunari Miyazaki

April 29, 2014

EXPANDER GRAPHS

- Expander graphs are highly connected and sparse graphs
- Applications in:
  - Sorting networks
  - Error correcting codes
  - Efficient computer networks
- But how do we construct them?

http://thinkofablueegg.com/wp-content/uploads/2010/08/map.jpg

EXPANDER GRAPHS

- In order to construct expanders, we need a way of measuring how *good* an expander is:
  - Expansion properties
- The expansion properties can be defined through:
  - Isoperimetric constants
    - A representation of the number of neighbors that a subgraph of the original graph has
  - Spectral property (eigenvalues)
- The best expander graphs are called *Ramanujan Graphs* (they have the best expansion properties)
- Only considered methods that generated k-regular expander graphs

EXPANDER GRAPHS

- So what are the Eigenvalues?
  - Informally, eigenvalues are real numbers that fully characterize a set of linear transformations (i.e. *eigenvectors*)
  - A square matrix of dimensions *n* x *n* will have *n eigenvalues*
  - In this discussion, I will always refer to the *second largest eigenvalue*

PROJECT OBJECTIVES

- Research construction methods for expander graphs
- Implement several such methods
- Compare their results using the eigenvalue definition of the *expansion property* in order to find out which method generates graphs that are more *Ramanujan Graphs*
  - *This property of expanders is also called Spectral Property*

PROJECT DESCRIPTION

- Construct *bipartite expander graphs* of various sizes (specified through input) using 3 types of methods:
  - Margulis' Method (5-regular) [*Explicit Method*]
  - Angluin's Method (3-regular) [*Explicit Method*]
  - Random Methods (3-regular and 5-regular)

- More complex method:
  - Ajtai's method (3-regular)

- Question: Which methods (explicit or random) generate graphs that are more likely to be *Ramanujan Graphs*?

PROJECT DESCRIPTION

- Construct *bipartite expander graphs* of various sizes (specified through input) using 3 types of methods:
  - Margulis' Method (5-regular) [*Explicit Method*] *
  - Angluin's Method (3-regular) [*Explicit Method*] *
  - Random Methods (3-regular and 5-regular) *

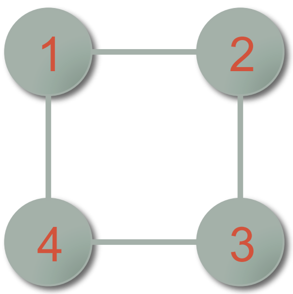
- More complex method:
  - Ajtai's method (3-regular)

- Question: Which methods (explicit or random) generate graphs that are more likely to be *Ramanujan Graphs*?

** Since the construction and analysis methods are similar,*
I will only present the 3-regular (explicit and random) graphs;
I will present results from all methods at the end of the presentation.

PROJECT DESCRIPTION

- How to represent these graphs?



$$\begin{pmatrix} 2 & 4 \\ 1 & 3 \\ 2 & 4 \\ 1 & 3 \end{pmatrix}$$

Second largest eigenvalue (in absolute value) = λ = **2**

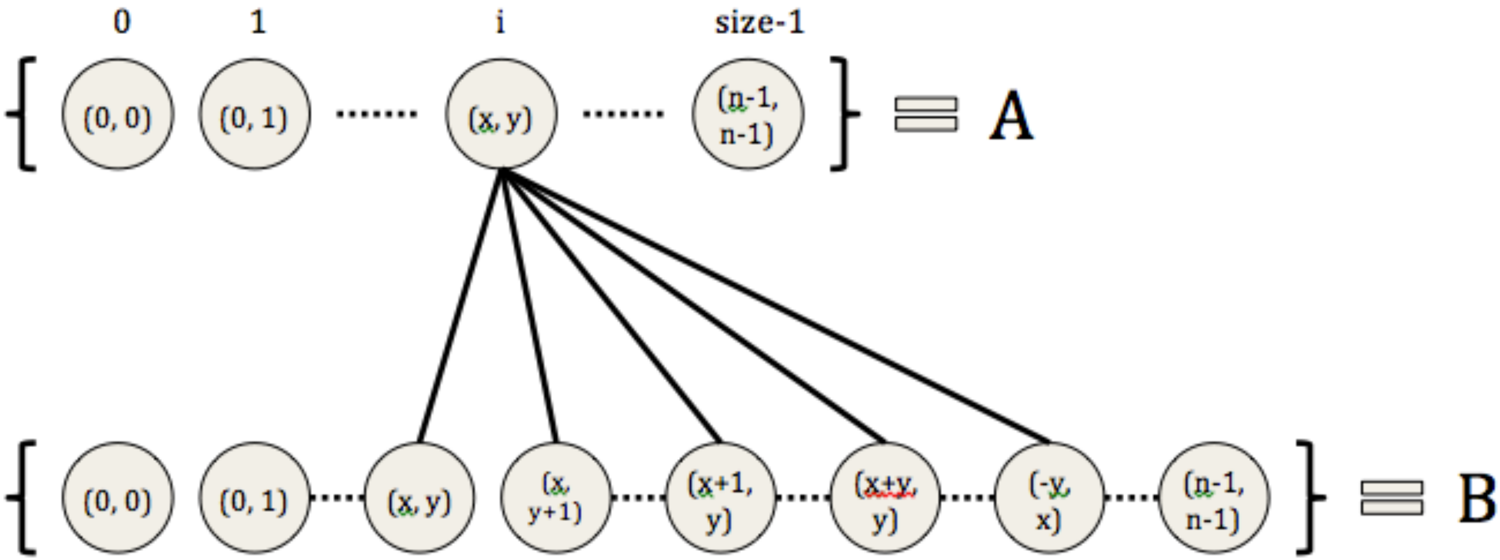
- How to check which ones are more likely to be *Ramanujan Graphs*?
  - Compare the resulting graphs using the Spectral property of the expansion property
    - Compute eigenvalue λ
    - Use inequality $\lambda \leq \sqrt{k-1}$ (k = degree of the nodes)
  - This method tests a graph in terms of how close it is to being a *Ramanujan Graph*

MARGULIS' METHOD (5-regular)

- Nodes of the graph are pairs *(x, y)* where *x* and *y* are integers *modulo n* (specified at input):

$$x, y \in \{0, 1, \ldots, n-1\}$$

- Generates *bipartite expander graphs* because the construction uses 2 sets, *A* and *B*, with nodes that are not connected to each other, within their sets.

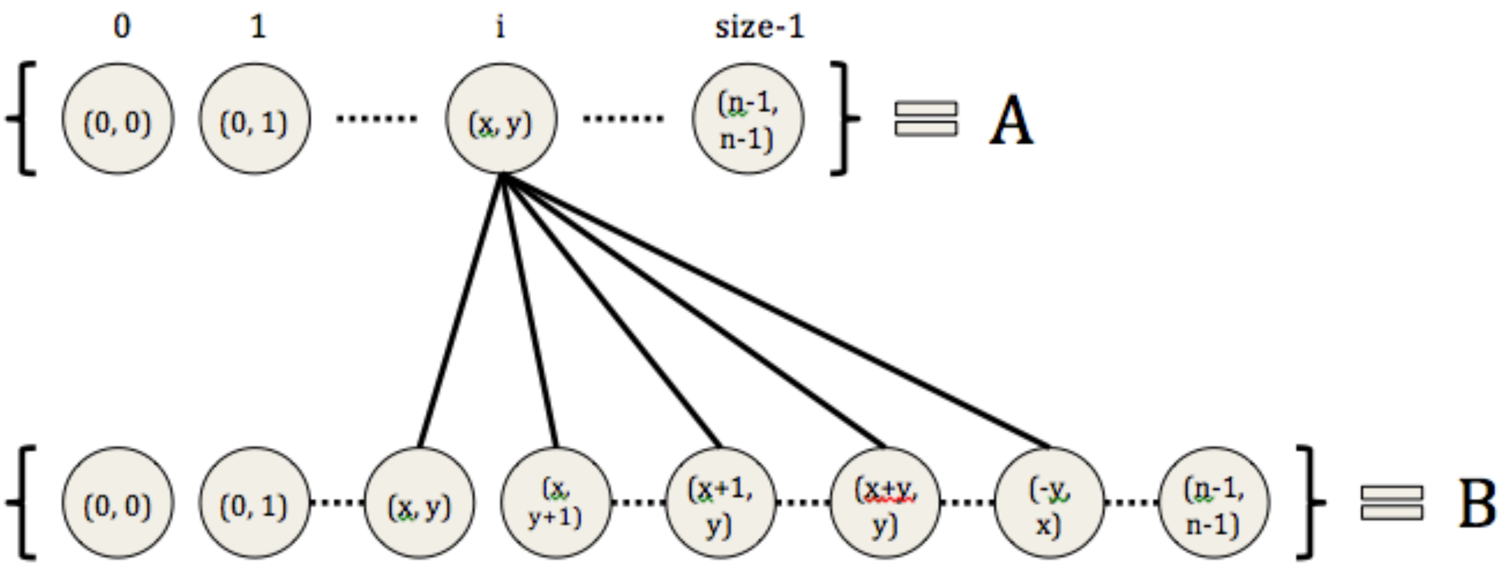


MARGULIS' METHOD (5-regular)

- Connects nodes in *A* to nodes in *B* by the following explicit rules:
  - Connect $(x, y) \in A$ to:

$$(x, y) \in B \quad || \quad (x, y+1) \in B \quad || \quad (x+1, y) \in B \quad || \quad (x+y, y) \in B \quad || \quad (-y, x) \in B$$

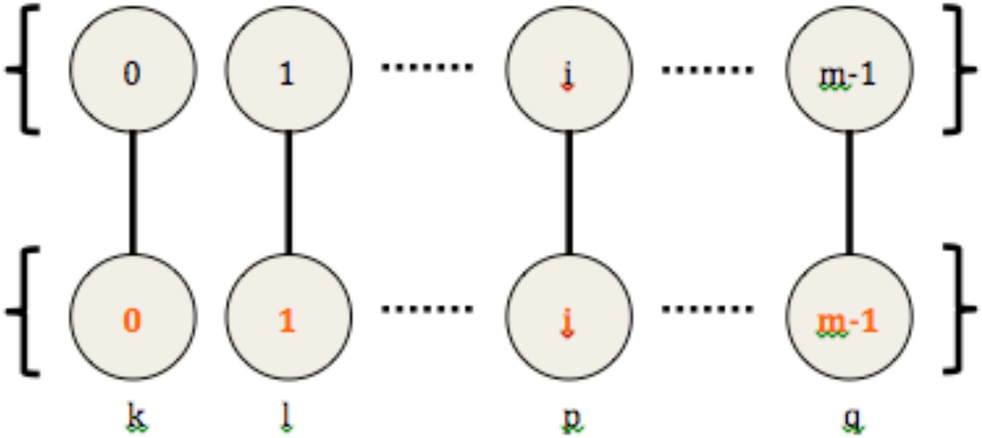
All operations are done modulo n.



RANDOM METHOD (5-regular)

- Nodes of the graph are indexed $\{0,1...,N-1\}$ where *N* is the number of nodes of the generated expander graph (provided through input).

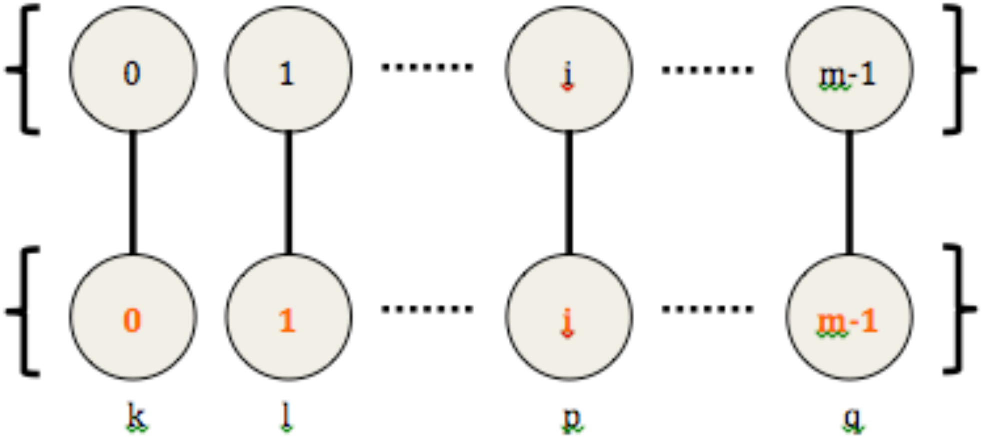
- Generates *bipartite expander graphs* because the construction splits the *N* nodes into two sets each of size $m = N / 2$
  - The nodes are not connected to each other, within their sets.



(Fig. 3) Illustration of the Random method

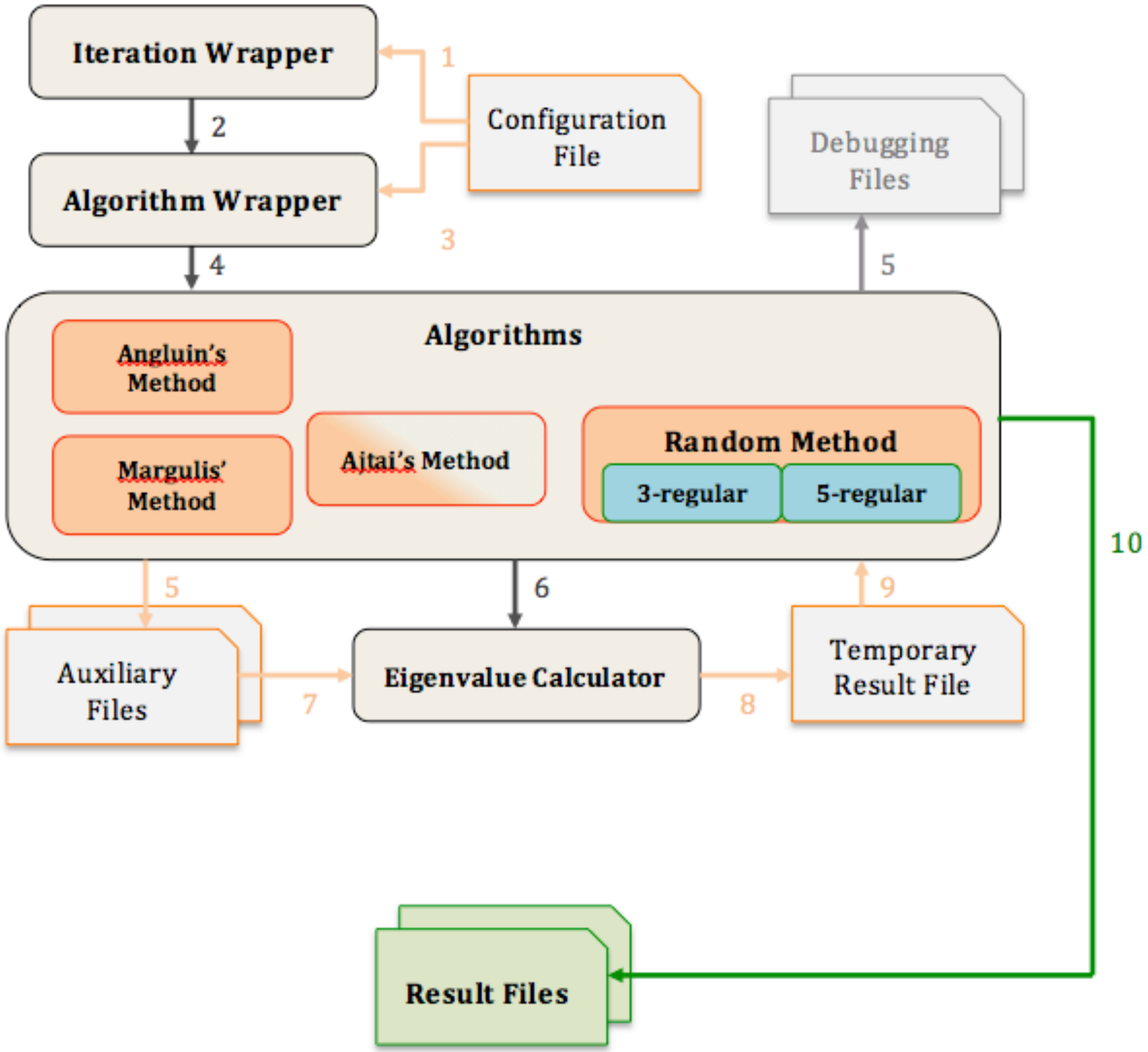
RANDOM METHOD (5-regular)

- After the split, the second half is permuted and the nodes get a new order.
- The first node of the first half is connected to the first node of the permuted, second half (*this constructs one edge*).
- In order to create a *k-regular expander graph* (5-regular in this case), this process is repeated *k* times (5 times).



(Fig. 3) Illustration of the Random method

PACKAGE STRUCTURE



(Fig. 4) Program execution flow of the expander package

PROJECT CHALLENGES

- Generating the adjacency matrices for the construction methods, when *N* is big (~ *1,000,000 – 2,000,000*)

- Generating eigenvalues for huge matrices of *2,000,000 x 5* elements

- Everything due to Python being slow…

- *Solutions:* generated eigenvalues through the *Power Method*, in C

PROJECT ACCOMPLISHMENTS AND RESULTS

- Package ran for values of *N* = { 8 … 1800 } (discreet values due to the construction technique – $N = 2 \times n^2$) for any of the 4 construction methods.

- As the value of *N* increases, the Random Methods still generate graphs with good expansion property, while the Explicit Methods fail to do that.

- The best value of the expander property is achieved by the Explicit Methods at very low value of *N*.

- For both types of constructions, the best expanders occur at small values of *N*

FUTURE WORK

- Ajtai's method
  1. Start with a random 3-regular graph, ***H***, on ***n*** vertices

  2. Given a value ***s*** (as a function of ***n***), swap edges ***(x, y)***, ***(u, v)*** with ***(x, v)***, ***(y, u)*** only if the swap will reduce the number of cycles of length ***s*** in ***H***

  3. Continue this process while the number of cycles is reduced at each iteration by a given factor (function of ***n***)

FUTURE WORK

- Ajtai's method
  1. Create ***H***, random 3-regular graph on ***n*** vertices
  2. Create method that swaps edges ***(x, y)***, ***(u, v)*** with ***(x, v)***, ***(y, u)***
  3. Using the conditions previously stated, perform swaps on edges of graph ***H*** until the stopping condition occurs (the decrease in the number of cycles of length ***s*** is not significant anymore)

FUTURE WORK

- Ajtai's method
  1. Create ***H***, random 3-regular graph on ***n*** vertices
  2. Create method that swaps edges ***(x, y)***, ***(u, v)*** with ***(x, v)***, ***(y, u)***
  3. Using the conditions previously stated, perform swaps on edges of graph ***H*** until the stopping condition occurs (the decrease in the number of cycles of length ***s*** is not significant anymore)
- Research algorithm that would generate Random Expanders that do not have multi-edges

SUMMARY

- *Project's goal:* Experimentally evaluate explicit and random methods for constructing Expander Graphs through the spectral property of the resulted graphs

SUMMARY

- *Project's goal:* Experimentally evaluate explicit and random methods for constructing Expander Graphs through the spectral property of the resulted graphs
- *Project deliverables:* Python/C package that generates Expander Graphs through 2 types of methods (explicit and random) and compares their quality through an eigenvalue based approach

SUMMARY

- *Project's goal:* Experimentally evaluate explicit and random methods for constructing Expander Graphs through the spectral property of the resulted graphs
- *Project deliverables:* Python/C package that generates Expander Graphs through 2 types of methods (explicit and random) and compares their quality through an eigenvalue based approach
- *Project result:* The Random Methods perform better for large number of vertices, but this might be due to the fact that the explicit methods construct too many multi-edges

ACKNOWLEDGEMENTS

- Professor Takunari Miyazaki
- Trinity College Computer Science Department

Trinity College
HARTFORD CONNECTICUT

- Travelers Insurance

TRAVELERS

REFERENCES

- G. A. Margulis. “Explicit constructions of concentrations.” Problemy Peredachi Informatsii 9 (4) (1973) 71-80 (English translation in: Problems of Information Transmission, Plenum. New York, 1975).
- Hoory, Shlomo, Nathan Linial, and Avi Wigderson. "Expander graphs and their applications." Bulletin of the American Mathematical Society 43.4 (2006): 439-561.
- D. Angluin. “A note on construction of Margulis.” Information Processing Letters 8 (1) (1979) 17-19.
- K. Chang. “An experimental approach to studying Ramanujan graphs.” Math Junior Seminar Thesis. (2001).
- C. Hammond. “Efficient algorithms for random expander graphs.”
- M. Ajtai. “Recursive Construction for 3-Regular Expanders.”