Connectivity of Flight Networks

Aaron Kilboy Alan Brett Bram Siebert Brian Regan
Jack Collins Maria Bridges

2nd Annual Stokes Modelling Workshop

June 18, 2015

Defining Connectivity

- Algebraic Connectivity
- Network Diameter
- Happiness Factor

$$\psi = \frac{Kc}{e^d}$$

- K = Constant
- c = Algebraic Connectivity
- d = Network Diameter

Base Case

- Connected graph of N nodes requires minimum N-1 edges
- What is our "best" worst case network?

Figure: Star Network, Diameter = 2 Algebraic Connectivity = 0.0112

Capacity

- It is not feasible to send every single flight through one airport
- Limit the amount of edges we allow a node to take
- This increased network diameter

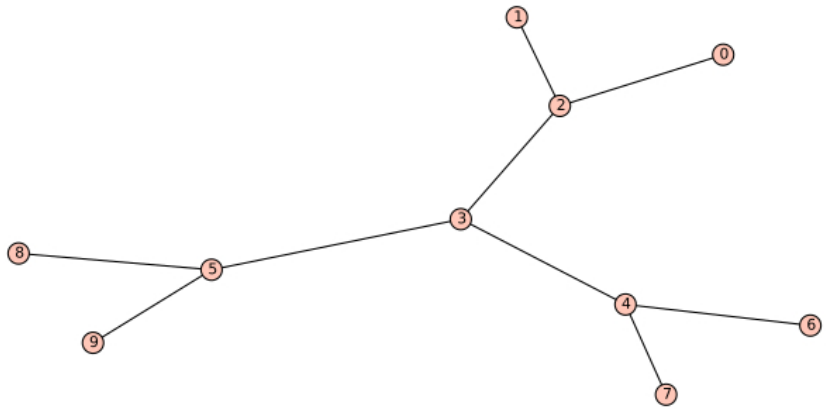
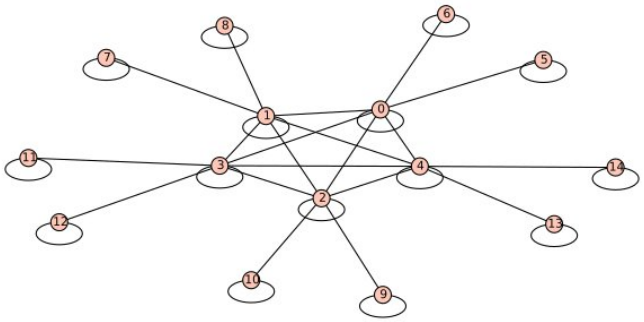


Figure: Capacity = 3

Hub Network

- Adding edges to the skeleton
- Creating "Hubs"
- Each Hub Airport is connected to every other Hub
- Clusters of Regional airports connected to one hub
- Limits diameter to just 3



Testing

- Using C++ we measure the connectivity of our algorithm on a small scale
- We tested the Hub network against two alternatives of the skeleton

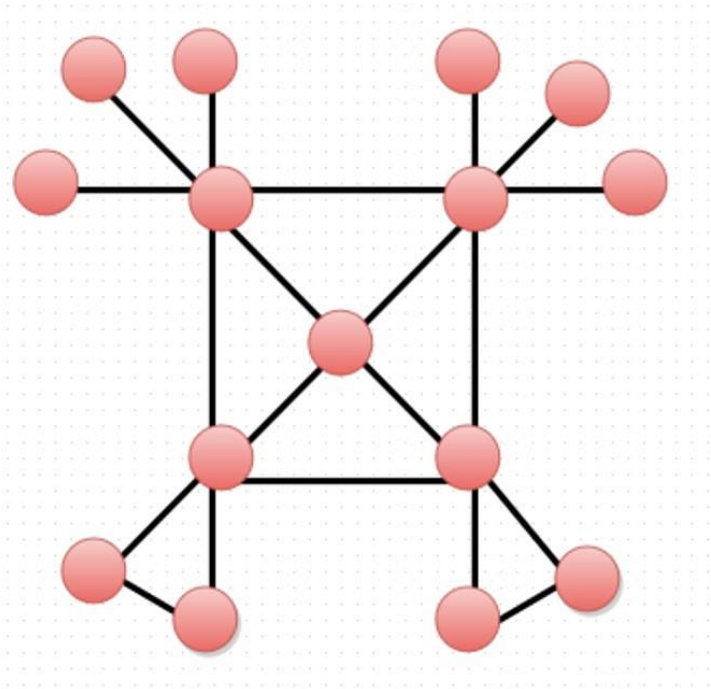
Testing

Stokes › Stokes › main.cpp › main(void)

```cpp
#include <iostream>
#include <iomanip>
#include <stdlib.h>
#include <fstream> // For files
#include <math.h>
#include <time.h>
#include "Vector.h"
#include "Matrix.h"

using namespace std;

int main(void )
{
    int num_of_nodes = 176;
    int num_of_hubs = 16;

    matrix A(num_of_nodes);
    A.zero();

    int count = 16;
    for (int i=0; i<num_of_nodes; i++) {
        for (int j=0; j<num_of_hubs; j++) {
            if (i != j)
                A.setij(i, j, 1);
        }
        for (int k=0; k<10; k++) {
            A.setij(count+k, i, 1);
            A.setij(i, count+k, 1);
        }
        count = count + 10;
    }
    cout << endl << endl;

    matrix D(num_of_nodes);
    D.zero();
    for (int i =0; i<16; i++) {
        D.setij(i,i,25);
    }

    matrix L(num_of_nodes);
    L.zero();
    for (int i=0; i<num_of_nodes; i++) {
        for (int j=0; j<num_of_nodes; j++) {
            L.setij(i,j, D.getij(i,j) - A.getij(i,j));
        }
    }

    L.print();
}
```

Source Code

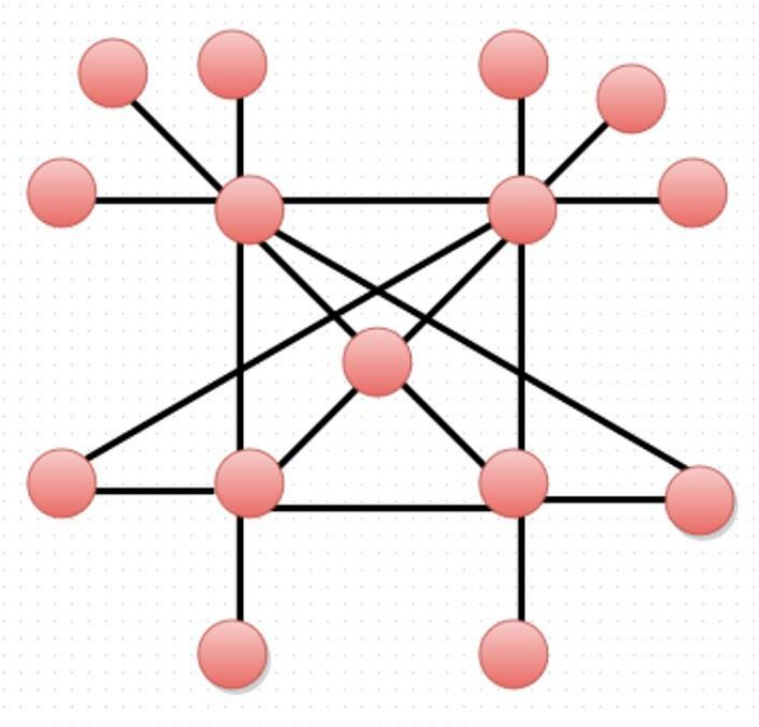
Testing



Connectivity = 0.44684

Diameter = 4

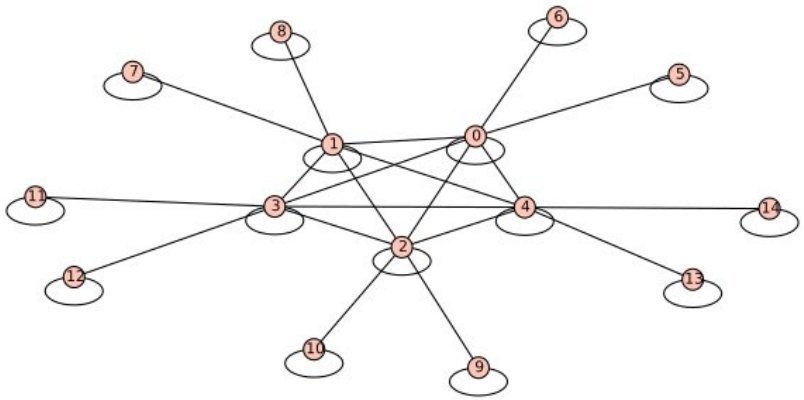
Testing



Connectivity = 0.55405

Diameter = 4

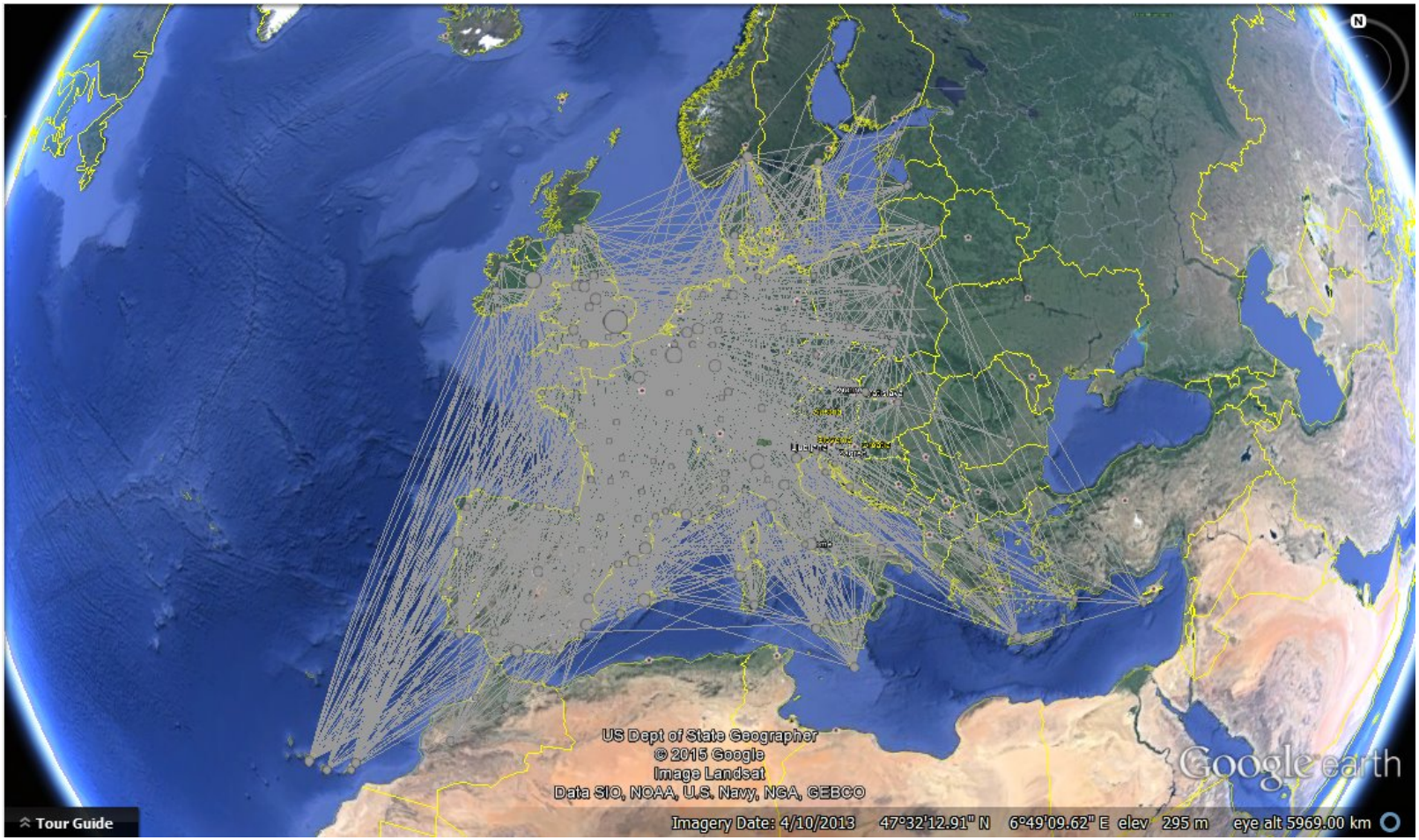
Testing



Connectivity = 0.68338

Diameter = 3

Implementing Algorithm



Ryanair Network, Algebraic Connectivity $= 0.89$ Diameter $= 4$

How many Hubs?

- Airports $= 176$
- Aeroplanes $= 312$
- H $=$ Hubs
- R $=$ Regional

$$\begin{aligned} \binom{H}{2} + R &= 312 \\ H + R &= 176 \end{aligned}$$

Hub to Hub

- Sensibly choose hubs
- Connect all hubs to other hubs
- Connect regional airports to closest Hubs

Implementing Algorithm

Ryanair Hubs

Implementing Algorithm

- Create Hubs and link together
- Calculate distance between any two airports
- Pair Regional airport to Hub
- Remove Hubs from list of Airports
- Pair every Regional with nearest Hub

Implementing Algorithm

Our Network, Algebraic Connectivity $= 0.48$ Diameter $= 3$

Open questions

- Optimize Hub locations based on more than just geographical convenience
- Assumes all planes are in the air at the same time
- Human factors