Research Plan Force-directed Graph Drawing

Michiel van Heusden, 4173309
Maurits van der Veen, 4167287
Kevin Oosterlaak, 4012372 Jesse Ceelen, 4061837

January 18, 2016

Contents

1 intro

Data requires often visualization before people understand what it means. This visualization is done with graphs and charts. This research focusses on the drawing methods of graphs. These graphs consist of objects and the relations between eachother. In graph theory the objects are called vertices and their relations edges. [1] Generating readable graphs becomes difficult according to the amount of vertices and edges. To generate these correctly many algorithms were developed. One family of functions to do this is called *Force Directed Graph Drawing*. The idea behind force directed graph drawing is to use physics based algorithms to calculate the positions of each correctly. In this research a few algorithms wil be examined and how correctly they can generate a graph. Not only will we compare different algorithms, we will also inspect how some algorithm-specific constants influence the results.

The algorithms that will be subject to our research are as follows:

- Hooke-Coulomb's Algorithm
- The Fruchterman Reingold Algorithm
- Eades' algorithm

More on these and the constants are in subsection 1.1.

1.1 Algorithms

As this paper focusses on the algorithms behind force directed graph drawing, the most important part of it are the researched algorithms. Most of these algorithms are (partially) based on physics.

The attractive and repulsive forces are between two vertices each time. At the end of each iteration the total forces per vector are summed.

1.1.1 Hooke-Coulomb

The Hooke-Coulomb algorithm is not named after its inventors, but after the laws it follows. The attractive forces are calculated using Hooke's Spring Law. The repulsive ones are based on particle physics. To be more precise, they are calculated using Coulomb's law of charged particles.

1.1.2 Fruchterman Reingold

The Fruchterman Reingold algorithm on the otherhand is named after its inventors. [2] For the method Fruchterman and Reingold proposed there are only two principles for graph drawing. Namely that connected verticed should be drawn near eachother and vertices should not be drawn *too* close to eachother. This algorithm can be seen as based off of the idea of ideally distributed vertices. It does not focus so much on each individual node as well as how far each node should be apart from eachother. Like the Hooke Coulomb algorithm it is based off of particle physics: Particles that are close to eachother are repulsive, whilst at a distance they are attracted to eachother. This algorithm uses a few input variables of which three will be changed:

- The allowed surface area W
- The optimal distance between vertices k
- Attractive weight α
- Repulsive weight r

During our testing all but the allowed surface area will be changed. Changing the surface area will only change the scale of the vertices. The influence the other values have to how the graph will look becomes clear from their usage. Both α and r are weights to the forces. Thus directly changing how much influence the attractive and repulsive forces have on the vertices. k on the other hand will not be directly changed, but by changing constant c in its formula.

$$k = c\sqrt{\frac{area}{numberofvertices}} \tag{1}$$

The constant k plays an important part in calculation of both the attractive and repulsive forces. These can be calculated using formulas 2 and 3 respectively. In both of these equations d is the distance between the two vertices on which the forces apply.

$$f_a(d) = \frac{d^2}{k} \tag{2}$$

$$f_r(d) = -\frac{k^2}{d} \tag{3}$$

However when blindly applying these forces a good looking graph will never converge. Therefore a cooling function is required. This cooling function indicates how much the maximum change is per vertex. As the name suggests, this function returns a lower value every iteration. Thanks to this function eventually the change in the graph per iteration will eventually become non-existent.

1.1.3 Eades

The algorithm proposed by Eades

2 Research Question (Staat niet bij de opdracht)

Which force directed graph drawing algorithm yields the highest quality graphs?

3 Problem Discription

3.1 SAMENVATTING (SOORT VAN) VAN HET VORIGE VERSLAG

4 The Experiment

5 Result Data

5.1 Tabellen

5.2 Grafieken

5.3 Toelichting

5.4 Statistische Hypothese

5.5 De uitwerking hiervan

6 Conclusion & Discussion

7 Reflection

References

[1] John Adrian Bondy and Uppaluri Siva Ramachandra Murty. *Graph theory with applications*, volume 290. Macmillan London, 1976.

[2] Thomas MJ Fruchterman and Edward M Reingold. Graph drawing by force-directed placement. *Softw., Pract. Exper.*, 21(11):1129–1164, 1991.