Shape, square

Description automatically generated

University of Essex

Department of Mathematical Sciences

1905876

|  |
| --- |
| Opinion Forming - A network theory problem |

|  |
| --- |
| Supervisor – Dr Joseph Bailey  April 15, 2022 |

Table of Contents

[Introduction 2](#_Toc100572553)

[Origin of Opinion forming 2](#_Toc100572554)

[Implementation of Opinion Forming 3](#_Toc100572555)

[Different aspects of Opinion forming 3](#_Toc100572556)

[Agent based Models 3](#_Toc100572557)

[Introduction to graph theory 4](#_Toc100572558)

[Different kinds of graphs 4](#_Toc100572559)

[Some formulas 5](#_Toc100572560)

[Centrality 6](#_Toc100572561)

[Adjacency Matrix 7](#_Toc100572562)

[Isomorphism 8](#_Toc100572563)

[Euler’s Analysis of Seven Bridges of Königsberg 8](#_Toc100572564)

[Graph Colouring 9](#_Toc100572565)

[Some popular lemmas in graph theory 10](#_Toc100572566)

[Introduction to network analysis 11](#_Toc100572567)

[Some puzzles which use the application of graph theory 11](#_Toc100572568)

[Bootstrap Percolation 11](#_Toc100572569)

[Majority Bootstrap Percolation 11](#_Toc100572570)

[Proof for max number of edges theorem. 12](#_Toc100572571)

[Conclusion 15](#_Toc100572572)

[Additional things that could be done 16](#_Toc100572573)

[Code Analysis 17](#_Toc100572574)

[Interesting insights 18](#_Toc100572575)

[Appendix 19](#_Toc100572576)

[Code Explanation 22](#_Toc100572577)

[Glossary 23](#_Toc100572578)

[Bibliography 25](#_Toc100572579)

# Introduction

If we consider a group of individuals as a network, the ways in which opinions are spread and changed can be thought of as a graph colouring problem.

A network where each vertex represents a person, the colour of the vertex determines the opinion of that individual and the edge joining two vertices is a relationship between individuals. It plays an important role in influencing the decisions people make and acts as a medium for the spread of information and ideas.

Opinion spread can therefore be modelled by considering the influence of an individual’s neighbours have upon the individual.

In the simplest case, we consider two opinions (colours for practical simulations) and opinions are formed/changed based on a simple rule.

Rule – at each timestep every vertex update, the colour of that vertex updates itself based on the majority of the neighbouring vertices, if at all the number of vertices of the opposing colour is the same, the vertex retains its colour.

It is simple to show how some setups are quite straightforward and settle into stable configurations whilst some settle into cycles (all v swap vertices). We discover that there are longer step times, a cycle of 1-2 once it starts looping and common ending colour vector which it ends up on.

In this project we investigate discovering, classifying, and determining dynamics of graphs based on their initial configurations through an approach based on exhaustive simulations backed by theoretical results.

# Origin of Opinion forming

The beginning of opinion forming was with people’s exposure to relevant information and

experiences, they then process the information and come to a judgement with a reasoning behind it. The different reasonings are then aggregated through either informal interactions or opinion polls.

# Implementation of Opinion Forming

## The formation of opinions within a group of people has been a subject of interest in many areas such as discussed below.

## Different aspects of Opinion forming

**Social**

Distance plays a great deal in forming connections and is a fundamental element of establishing social links. Geo-social platforms are also highly correlated, and this could be further used to network.

One of the great aspects of opinion forming when looking at it from the social point of view, we consider the identification of influential spreaders of the information and the impact of homophily. Homophily is the principle where the contact between similar people occurs faster than among dissimilar people. Spreading of the rumours is another big problems/aspect of the social side of opinion forming.

**Psychology**

**Economics**

**Finance**

**Political**

One of the big impacts of opinion forming in politics is during the elections and protests. One can easily model the dynamics of an election using data available on social media and then study the characteristics of distinct group of people who are like minded.

‘In this light, the formation of public opinion is understood to be a process that revolves around individuals. It begins with their exposure to politically relevant experiences and information. Each individual processes this information, thereby coming to a judgment that yields an attitude. The attitudes of different individuals are then aggregated, either through informal interactions or more formal mechanisms, such as elections or opinion polls.’ [5]

**Public opinion - Media (POV)**

Media provide people with cues as to what could ideally lead into formed opinions, but these are usually short lived. One example could be people evaluating the performance of a politician and/or their party based on the issues which are glorified by the media themselves. If the issue stays longer on media’s agenda, people start to take sides, and form biases ultimately leading to opinions.

‘For example, in the evaluation of President George Bush in 1991, his overall job approval rating was high, corresponding with the victory in the Gulf War. However, in 1993 his approval rating was far lower than it was in 1991 because the Persian Gulf crisis was overshadowed by intense media coverage of economic recession. During this time, Bush's approval rating was more strongly linked to his performance concerning the economy than to his performance on foreign policy matters.’[6]

**Agent based Models**

Agent-based models may be employed to describe a variety of characteristics of the agents involved and the way they interact, allowing us to understand the evolution of the opinions of the individuals, and if and how they reach a final consensus or whether the agents polarize around a small number of different opinions.

# Introduction to graph theory

A graph is a mathematical structure containing two finite sets V and E and can be denoted by G = (V, E). V represents the vertices and E stands for the edges. Graphs have order and size, the number of vertices in a graph is called the order of the graph and the number of edges is the size.

Chart

Description automatically generated

Figure 1: Order of the graph - 6, size of the graph - 6

The figure below can be used to show how to name the vertices and the edges of a graph.

The vertex-set can be called as VA = (a, b, e, d, c)

The edge-set can be called as EA= (ab, be, ed, dc, ca, ad, bd)

Chart, radar chart

Description automatically generated

Figure 2: vertex and edge set

## Different kinds of graphs

1. Fully connected Graphs – A graph which has a path(edge) between each pair of vertices is called a fully connected graph.
2. Disconnected Graphs – A graph where the vertices are split into 2 or more disjoint groups, such that one cannot link a vertex in one group to the vertex in another group by traversing along through the edges is called a disconnected graph.
3. Planar graphs – A graph which can be drawn on a plane such that no edges cross each other and only intersect at the vertices of that graph.
4. Circular Graphs – It is an undirected graph whose vertices can be associated with a finite system of chords of a circle such that two vertices are adjacent if and only if the corresponding chords cross each other.

A close-up of a dart board

Description automatically generated with low confidenceDiagram, radar chart

Description automatically generated

1. Bipartite Graphs – A graph G whose vertices can be divided into two disjoint sets M and N such that each edge of the original graph G connects a vertex of set M and set N, such a graph is called a bipartite graph.

Diagram

Description automatically generated

1. Loop less/Simple Graphs – When there is at most one edge joining two vertices, no edge may join a vertex to itself, and the edges are not directed, the graph formed is called a loop less graph or a simple graph.
2. Digraphs or Directed Graphs are graphs where the edges are directed between any given two vertices. The direction of the edge is denoted by the arrow on that very edge.

A picture containing diagram

Description automatically generated

## Some formulas

The maximum number of edges possible in a single graph with ‘n’ vertices is nC2 where nC2 is equal to n(n-1)/2.

The number of single graphs possible with ‘n’ vertices = 2nC2. = 2n(n-1)/2.

Chart, line chart

Description automatically generated

For example, consider the triangle above.

n = 3.

nC2 = 3

2nC2 = 23 = 8

The 8 graphs are as follows:

Chart, line chart

Description automatically generated

## Centrality

Centrality is the measure of closeness of a node from the other nodes in a graph. It can be calculated using various measures, couple of the most important ones are node prominence and structural importance. Centrality indices answers the question “What characterizes an important vertex?”. The answer is given in terms of the real-valued function on the vertices of the graph, where the values produced are expected to provide a ranking of the most important nodes.

The word ‘important’ here has a wide number of meanings, leading to many different definitions of centrality. It can be categorized mainly by either the network flow or by walk structure which further disassociates into various categories.

The **Betweenness centrality** is the measure of a vertex within a graph. It is basically the number of times a node acts as a bridge along the shortest path between two other nodes.

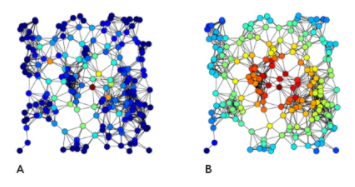
The **Closeness centrality** of a node is the average length of the shortest path between the node and all the other nodes in the graph.

The **Eigenvector centrality** is the measure of the influence a node has on the network.

The **Degree centrality** can be defined as the number of links incident upon a node, that is the number of ties the node has.

The **harmonic centrality** reverses the sum and reciprocal operations in the definition of closeness centrality.

The **Katz centrality** is a generalization of degree centrality. Degree centrality measures the number of direct neighbours, and Katz centrality measures the number of all nodes that can be connected through a path, while the contributions of distant nodes are penalized

A picture containing vector graphics

Description automatically generated

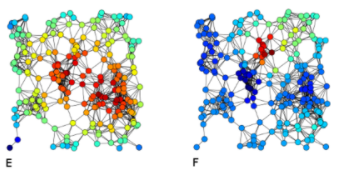


Figure 3: Same network but different applications of centrality indices.

There are a couple limitations to the centrality indices, one of them being obvious wherein one application of centrality is often sub-optimal for a different application. The other one

## Adjacency Matrix

An adjacency matrix is a square n X n matrix that is used to represent a finite graph by storing the nodes labelled as 1 if they are adjacent and 0 if they are not.

In case of a simple graph, as in the figure below. The diagonal will always be made up of zeroes since edges from a vertex to itself (loops) are not allowed in simple graphs.

The adjacency matrix of an undirected simple graph is symmetric, and therefore has a complete set of real eigenvalues and an orthogonal eigenvector basis The set of eigenvalues of a graph is the spectrum of the graph.

Diagram

Description automatically generated

Figure 5: Adjacency matrix

The representation of the matrix is different for undirected and undirected graphs.

A screenshot of a computer

Description automatically generated with low confidence

## Isomorphism

Two graphs are said to be isomorphic if there exists a one-to-one correspondence between their vertices. For graphs G1 and G2, if the number of edges joining any two vertices of G1 is equal to the number of edges joining the corresponding vertices of G2, they are said to be isomorphic graphs.

Chart, scatter chart

Description automatically generated

Figure 4: Isomorphic graphs

In the diagram above, a1 has two edges and its corresponding vertex d2 has two edges as well. Similarly, d1<->a2, b1<->b2 and c1<->c2 all have the same number of edges to their corresponding vertex, making this and isomorphic pair of graphs.

## Euler’s Analysis of Seven Bridges of Königsberg

The foundation of graph theory started when Leonhard Euler laid negative resolution to ‘*Seven Bridges of Königsberg*’. This is an infamous mathematical problem where two mainland in the city of *Königsberg* in Prussia (now Russia) were set on both sides of Pregel river. These two mainland were connected by seven bridges.

Diagram

Description automatically generated

Figure 6: Seven bridges problem

The problem was to devise a way wherein one would cross each bridge one and only once. Euler proved that this problem has no solution. How he did this is by first pointing out that choice of route inside either landmass are irrelevant as the only important feature is the sequence in which the bridges are crossed. This enabled him to rephrase the problem in a more abstract way where the landmasses were nodes, and the bridges were the edges. It does not really matter whether the edges are curved or straight. The resulting structure is a graph.

Diagram

Euler's abstract representation of the seven bridges problem.

Figure 7: Euler's abstract representation of seven bridges problem

## Graph Colouring

Graph colouring was first introduced when in 1852, when Francis Guthrie postulated the four-colour conjecture, observed that 4 colours were sufficient to colour the map of any region such that no region sharing the same border (adjacent borders) have the same colour. Francis’ brother approached his teacher (Augustus De Morgan) at the university college, who later wrote to William Hamilton in 1852.

This ultimately transformed into the problem of deciding whether it is possible to colour the vertices of every planar graph with four colours such that no two adjacent vertices are assigned the same colour.

**Vertex colouring** arises more commonly then edge colouring or map colouring. It can be defined as the assignment where f:VG->C from its vertex-set onto a k element set C whose elements are called colours (C = 1,2,3…k). For any k, such an assignment is called vertex-colouring.

Diagram

Description automatically generated with medium confidence

Figure 8: Vertex-colouring

A **colour class** in a vertex-colouring of a graph G is a subset of VG containing all the vertices of a given colour.

A **proper vertex-colouring** is vertex-colouring of the graph is such that the endpoints of each edge are appointed different colours.

The **Chromatic number** of a graph is G is denoted by x(G), is the minimum number of different colours required for a proper vertex-colouring of G. A graph G is k-chromatic if x(G) = k.

# Some popular lemmas in graph theory

**Regularity lemma** states that every dense graph can be partitioned into a small number of regular pairs and a few leftover edges. Since regular pairs behave as random bipartite graphs in many ways, the Regularity Lemma provides us with an approximation of an arbitrary dense graph with the union of a constant number of random-looking bipartite graphs.

**Handshaking lemma** states that in every finite undirected graph, the number of vertices that touch an odd number of edges is even.

**Diagram

Description automatically generated**

From the figure above, the vertices 4, 5, 2 and 6 are even number of vertices who are connected to an odd number of edges. The sum of the degrees of their edges is 2 + 3 + 2 + 3 + 3 + 1 = 14. This is known as the **degree sum formula**. It states that the sum of the degrees of the vertices in a graph is twice of the number of edges present in the graph.

**Graph removal lemma** states that when a graph contains a few copies of a given subgraph, then all the copies can be eliminated by removing a small number of edges.

# Introduction to network analysis

The study of graphs can be divided into mainly two sections based on their symmetry namely as Symmetric and Asymmetric relations.

*Symmetric relations* are the kind of binary relations where if a=b true then b=a is also true, where a, b belong to the set X. Set X here can be called symmetric.

Asymmetric relations can be better explained with the help of digraphs, Digraphs are graphs made of a set of vertices connected by directed graphs.

Network analysis has many applications in statistics, particle physics, electrical engineering, economics, etc.

## Some puzzles which use the application of graph theory

1. 4 cube problem
2. Ramsey theory
3. Six people at a party problem
4. The eight circles problem

## Bootstrap Percolation

In statistical mechanics, bootstrap percolation is a percolation process in which a random initial configuration of active cells is selected from a lattice or other space, and then cells with few active neighbours are successively removed from the active set until the system stabilizes. The order in which this removal occurs makes no difference to the final stable state. Bootstrap percolation can be interpreted as a cellular automaton, resembling Conway's Game of Life, in which live cells die when they have too few live neighbours. However, unlike Conway's Life, cells that have become dead never become alive again. [2]

## Majority Bootstrap Percolation

In majority bootstrap percolation on a graph G, an infection spreads according to the following deterministic rule: if at least half of the neighbours of a vertex v are already infected, then v is also infected, and infected vertices remain infected forever. Percolation occurs if eventually every vertex is infected. [3]

## Proof for max number of edges theorem.

Text, letter

Description automatically generated

We know that the degree of each vertex in a simple graph is 1 less than the number of vertices(n).

= n-1

For n=2,

Degree = n-1 = 2-1 = 1

Since the sum of the degrees is even for a simple undirected graph, we can denote it by ∑d(v)=2m

For n=2, ∑d(v)=n-1

For n vertices the total degree is n(n-1)

This implies, 2m = n(n-1)

m = n(n-1)/2

m being the number of edges

**Degree for the bipartite graph**

For a vertex, the number of adjacent vertices is called the degree of the vertex and is denoted deg(v). The degree sum formula for a bipartite graph state that

Summation of degree (v) = Summation of degree (u) = |E|

**Cases for graphs**

All cases assume that the point of contact retains their opinion when at a 50-50 situation,

Case 1

For first iteration, al the vertices change to opposite colour since they are each connected to 2 vertices of the opposite colour. For the second iteration, the state goes back to the starting state and so on.

Case 2

The graph remain unchanged as the vertex is connected to 1 same and one opposite colour, at this point, the vertex can either change to the opposite colour which will result in a case like case 1 and ultimately leading into cycles. Or it chooses not to change the colour leading into the same graph repeatedly.

Case 3

For this graph, again there are 2 possibilities, one wherein the graph remains the same because the vertex decided not to change due to a 50-50 probability. On the other hand, middle blue vertex changes into red and then the first vertex changes into red as well in the following iteration ultimately resulting in an entire red graph.

**Outcomes we already know**

**Attempt at explaining the cyclic graphs:**

**Case 1**

We start with a simple graph G made up of 4 vertices A, B, C, and D and the edge set AB, BC, CD, and DA. Each vertex represents an individual and is assigned a colour to show either a similarity or difference in opinion. We use the colours red and blue to differentiate here. Assuming if a vertex(person) comes across a case where they are in contact with 1 similar and 1 dissimilar opinion, they go with their original choice and retain their opinion.

We start off with A and C being red and B and D being blue. This is a special case wherein the graph would change but remain the same in terms of isomorphism. Let us take vertex A to understand this. Vertex A is of the opinion ‘X’ on some issue M, and so is vertex C. But vertex B and D are of the opinion ‘Y’ on the same issue. A is connected to both B and D and hence has A changes its opinion from ‘X’ to ‘Y’ since for A both B and D go with opinion Y. Similarly, B, C, and D all change their opinion only to land up in an identical situation. This cycle would go on and on.

**Case 2**

Consider the same graph in case 1 except this time, the vertex pair with same opinions are A, B as opposed to C, D. In this case, no vertices would never change their opinion assuming the same clause wherein if a vertex comes across a situation where it faces 1 similar and 1 opposed opinion it retains its original choice. Hence there is no cycle, and the graph remains exactly the same.

**Modelling opinion forming**

Chart, radar chart

Description automatically generated

An observer interconnected with a set of subjects by mutual relations of trust (Solid green line) or distrust (dotted red line). Starting from a small set of opinions (in this case just one) marked with a + in black and a connection of unknown subjects (grey question marks), the observer gradually forms opinions on all subjects.

The formed opinion is determined as a product of the opinion and the sign of the relation between the source subject and the target subject. A positive opinion is formed when the source opinion and the relation are either both positive or both negative; a negative opinion is formed otherwise.

# Conclusion

After considering the code driven approach to model and simulate results for the given problem statement, we can conclude that there all configurations either settle into a stable state, start looping after a certain number of cycle or end up in a loop.

We also see that the increase in the number of configurations is exponential, which is an expected outcome as the configurations depend on the graph list and colour vector list which in turn are increasing exponentially. This means that when the number of vertices increases, the cycles are bigger and so are the loops, implying much more complex patterns in opinion distribution.

# Additional things that could be done

* Adding more than 2 colours in the colour vectors
* Adding weights to the individual
* Probabilistic approach
* Removing all the isomorphic graphs from the code

# Code Analysis

Imports and libraries

Colour vector list shortening which reduces time complexity

Graph vector list shortening which reduces the time complexity by calculating only the connected graphs

The output is saved as a csv file

The time for which the code runs for is calculated and the difference can be seen when n increases

Another jupyter notebook code has been imported from a GitHub repo to visualise all the graphs that can be made for an arbitrary n and out of all the possible ones, how many are connected and isomorphic respectively.

# Interesting insights

# Appendix

All possible graphs with 3 vertices

A picture containing text, skiing, snow, slope

Description automatically generated

Unique graphs with 3 vertices

A picture containing text, sky, line, envelope

Description automatically generated

All possible graphs with 4 verticesA picture containing shape

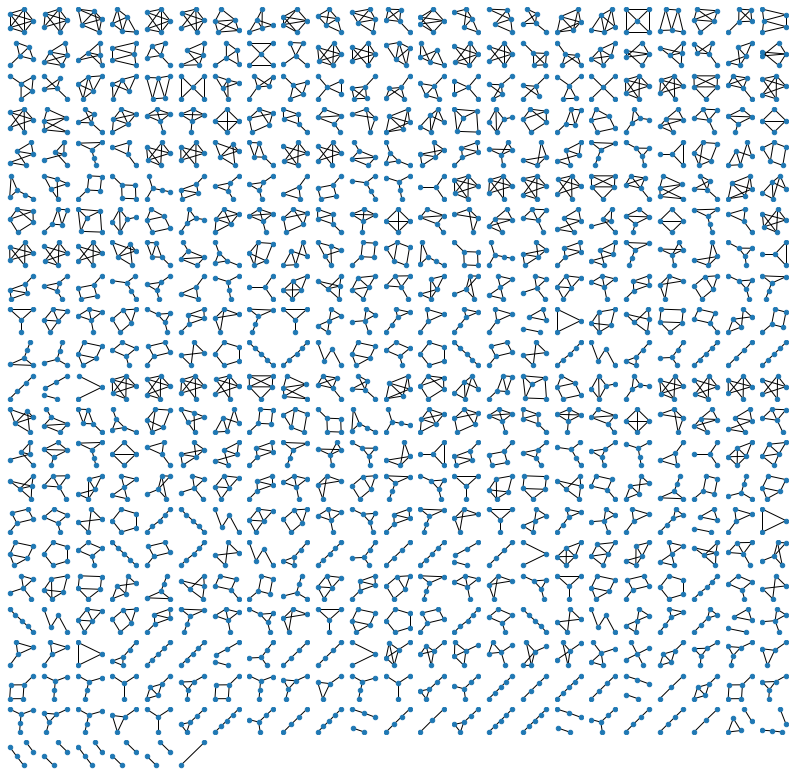
Description automatically generated

Unique graphs with 4 vertices

Shape, polygon

Description automatically generated

All possible graphs with 5 vertices



Unique graphs with 5 vertices

Diagram

Description automatically generated with medium confidence

# Code Explanation

Using the code in [graph.py](https://github.com/Sanskar-16/capstone/blob/master/code/graph.py), we run simulation for various number of graphs and colour vertices.

Initially we import all the necessary libraries for the project including some of the infamous ones like NumPy, Pandas and Time. We also import the network library to deal with matrices and graph related components.

Next, we declare all the variables and data structures, followed by a custom list containing all the features of a graph in the format of a dictionary, we do this, so it is easier to assign and manipulate the values when they are assigned pairwise to their respective keys [].

We have also implemented the code in the conventional format where it follows object-oriented programming []. This makes it easier to maintain and modify the code.

We also use 2 clever functions to generate and manipulate the list of all the possible graphs and all the possible colour vectors for that number of vertices, the interesting part about this function is that while generating the colour vectors for the user input, it halves the list to save memory. Let us take an example to understand why this is feasible:

For a graph denoted by the adjacency matrix

[[0,1,1],

[1,0,1],

[1,1,0]]

It will produce a similar result when multiplied to either the colour vector [1, -1, 1] or the colour vector [-1, 1, -1] i.e., the next colour vector for both the iterations would be a vector consisting of all the vertices of the same colour denoted by either [1, 1, 1] or [-1, -1, -1]. This means the graphs gets infinitely stuck in a loop for the same colour. In order to avoid redundancy, we eliminate half the colour vector list by inferring results about all the colour vectors.

When we run the program, it asks the user for an input. This variable decided the number of vertices the simulation runs for.

This produces a csv file, which consists of various columns which are displayed below:

Graphical user interface, application

Description automatically generated

Using the code in [analysis.ipynb](https://github.com/Sanskar-16/capstone/blob/master/code/analysis.ipynb), we analyse the csv file obtained containing the results.

# Glossary

Lemma - A subsidiary or intermediate theorem in an argument or proof.

Arc – An arc is a directed line

Edge – A line joining a pair of vertices

Loop – Edge/Arc that joins a vertex to itself

Adjacent – Next to or adjoining something else.

Walk – Series of vertices and edges.

Path – A walk where no repeated vertices.

Clique – In an undirected graph G = (V, E), it is a subset of vertices C ⊆ V, such that every two distinct vertices are adjacent.

Width – In an undirected graph in which every subgraph has a vertex of degree at most k, that is some vertex in the subgraph touches k or fewer of the subgraph’s edges. The width of the graph is the smallest value of k for which it is k-degenerate. It measures how sparse the graph is.

Size – The size of the graph is the number of edges in the graph.

Order – The number of vertices in a graph is the order of the graph.

Dump

Relate this to how this project could be extended by adding extra features - For the most part, this research effort has focused on the second step in the opinion formation process, the social psychology of how individuals process information (either from direct or indirect experience) when forming an attitude. The critical outcome or dependent variable is an attitude. Typically, this is defined as an individual's overall evaluation of an object with some degree of favor or disfavor. In this view, an attitude consists of three types of elements: cognitive (beliefs); affective (emotions or feelings); and behavioral (intent to act). These elements can affect one another. While they are usually consistent, they may not be. Attitudes are also related to one another.

In a way that has important consequences for attitude strength and stability, this interattitudinal structure may vary with regard to its elaboration, its coherence, and the degree to which it has a hierarchical structure. A central aim of the research is to examine how different conditions (both situational and dispositional) affect the processing of information and thus the resulting attitude. The research is designed on the presumption that all individuals operating under similar conditions will process information in the same way. Consequently subjects' responses are aggregated, and inferences are made regarding how the ‘average’ person will form, maintain, or change their opinions under the conditions examined.

# Bibliography

[1] - <https://www.sciencedirect.com/science/article/pii/S0166218X19304512>

[2] - <https://en.wikipedia.org/wiki/Bootstrap_percolation>

[3] - <https://arxiv.org/abs/math/0702373>

[4] - <https://www.iro.umontreal.ca/~hahn/IFT3545/GTWA.pdf>

[5] - <https://www.sciencedirect.com/topics/social-sciences/opinion-formation>

[1] Mastroeni, L., Vellucci, P., & Naldi, M. (2019). Agent-based models for opinion formation: A bibliographic survey. *IEEE Access*, *7*, 58836-58848.

/ TODO

* Add to the introduction

This project would investigate discovering, classifying, and determining dynamics of graphs based on their initial configurations, either through theoretical or more exhaustive simulated results.

* Add a report sort of thing for the code
* Explain the code in appendix
* Add proper references
* Add a section explaining the clever part of the code, name it the method section or something
* Include a graph section showing all the interesting graphs
* Fix the way output file gets saved