Eindhoven University of Technology

BACHELOR

The effect of clustering in random graphs

Adriaans, E.L.A.

Award date:
2015

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
The effect of clustering in random graphs

Supervisors:
J.S.H. van Leeuwaarden
C. Stegehuis

Author:
Erwin Adriaans
0814098
e.l.a.adriaans@student.tue.nl

July 9, 2015
Abstract

The goal of this thesis is to extend the configuration model to arbitrarily sized clusters and furthermore to a graph after bond percolation. To achieve this we first explain the configuration model and the model for graphs with single edges and triangles. Where in the latter model we already see some results of the effect of clustering on the size of the giant component. This is then extended to a generic model for arbitrary size clusters. We see that graphs that show more clustering have a smaller giant component which grows slower when we increase the amount of neighbours. If we use different sorts of graphs we see that their growth speeds differ a lot. We also find that the giant component differs in graphs with the same clustering coefficient. Thereafter we study graphs after percolation. We do this first for the configuration model and work towards a random graph with arbitrarily sized clusters. If we use percolation on graphs with single edges and triangles we already see that a graph with more clustering is more resilient to percolation. When we look at this effect for random graphs with bigger clusters we see that this reasoning does not hold for increasingly larger clusters. The most resilient graph, out of random graphs containing complete graphs only, turns out to be the graph with only complete graphs on 4 vertices. The explanation for this is that the size of the giant component also plays a role in when the giant component start to emerge. Since graphs with more clustering have a smaller giant component than graphs with less clustering their giant components emerge later. There can be done a lot more research on the behaviour on percolation of certain types of graphs after percolation. Our model with only 2 variables (two types of complete graphs in a random graph) could be extend to a model with an arbitrary amount of different complete graphs. Our model of percolation seems plausible but the simulation and theoretical results have not been verified.
Contents

1 Introduction 3

2 Random graphs 4
   2.1 Configuration model 4
   2.2 Generating functions 4
   2.3 Component sizes 5
   2.4 Giant component 5
   2.5 Power-law with exponential cutoff 6
   2.6 Simulation 7

3 Random graphs with clustering 9
   3.1 Configuration model with clustering 9
   3.2 Generating functions 9
   3.3 Component sizes 10
   3.4 Giant component 11
   3.5 Simulation 12

4 Arbitrary size clusters 14
   4.1 Generating functions 14
   4.2 Component sizes 14
   4.3 Giant component 15
   4.4 Effects of clustering 15

5 Percolation 20
   5.1 Percolation on a random graph 20
   5.2 Single edges and triangles 21
   5.3 Arbitrary sized clusters 22
   5.4 Simulation on bigger clusters 25

6 Conclusion 30
   6.1 recommendations 31
1 Introduction

Understanding the fundamental properties of the many complex networks we are all part of is an extremely hot and important topic. If we do not understand complex network behaviour things can go wrong on a large scale. On 10 August 1996 in Western North America a blackout was triggered by a fault between a power line and a tree. This failure triggered a series of failures leading to the loss of power for 7.5 million people [1]. But meanwhile scientists have done some research on networks. One of them is Mark Newman, a physicist who became known for his research on complex networks where in particular for work on random graph theory and its applications [2]. Sections 2 and 3 describe some of his work. [3 [4].

A random graph is a set of vertices with edges connecting pairs of vertices at random. Random graphs are good models for real-world networks of various types. The most classical random graph model is the Erdős-Rényi model, in this type of graphs an edge between two vertices is present with probability \( p \). These random graphs, however, result in a Poisson degree distribution. Not all graphs have a Poisson degree distribution, and in fact most graphs do not. Newman developed a technique to solve random graphs for any arbitrary degree distribution, where by solve we mean a method to determine the statistical properties of a graph exactly. This is translated into my own words in Section 2. His method makes use of generating functions. Making use of that we can find generating functions for a the distribution of the degree of a vertex at the end of a randomly chosen edge, and more importantly the component sizes in the graph. With these we can even determine the position where a giant component first appears.

This method is however based on graphs with single edges, but many networks show clustering. It is likely that two friends of an individual are also friends of each other. In another paper Newman [4] describes a way to give exact solutions for a graph with clustering; see Section 3. Newman gives a method for solving graphs with single edges and triangles, were a triangle is a complete graph on 3 vertices (the connection between two friends).

Section 2 and 3 are written to gain a good understanding of the methods of Newman. In Section 4 we show an extension of Newman’s model. We did not look at a graph with single edges and triangles, but any point could have two complete graphs on an arbitrary amount of vertices. We can still derive all the properties. We then consider the effects of clustering in a graph. What happens with the giant component when we construct a graph with complete graphs on more vertices?

Finally we look at simple random graphs after bond percolation. We determine generating functions for the graph after percolation. This turns out to be still possible in a similar way. There are however some differences that need to be taken into account. In [4] Newman gives a way to determine the size of the giant component in a graph with edges and triangles after percolation. We found a way to do this for a graph with complete graphs of arbitrary size. With this method we cannot exactly determine the point where a giant component appears. So our last contribution is to describe a graph with arbitrarily sized clusters in terms of generating functions. Finally we show some simulation results the size of a giant component in graphs with clusters for some percolation probabilities.
2 Random graphs

We start with unipartite random graphs and we expand this model to get to our final model. This part is based on the paper of Newman et al. [3]. This means that we do not look at the Erdős-Rényi model. But to the configuration model instead.

2.1 Configuration model

The configuration model creates a random graphs on $N$ vertices. Where in the Erdős-Rényi model any vertex has a Binomial degree distribution, in this graph any degree distribution is possible. For any vertex in the graph, a random integer is drawn from a distribution to represent its degree. All these random integers are independent identically distributed, under the constraint that the sum of the degrees must be even to make a matching possible. Next all vertices are connected to each other at random, i.e. one graph is chosen at random from the set of all possible graphs with this degree sequence for the vertices. In this way any possible graph over $N$ vertices can be generated and according to the needs of the network any specific probability distribution can be chosen.

2.2 Generating functions

All calculations done are based on generating functions. The most basic and most essential generating function is the one for the probability distribution of vertex degrees. We have

$$G_0(x) = \sum_{k=0}^{\infty} p_k x^k, \quad (1)$$

with $p_k$ the probability for a random vertex to have degree $k$. As a result we have that $G_0(1) = 1$, since the sum over all possible outcomes evaluates to 1.

With this generating function we can for example easily calculate the expectation of the probability distribution. This means that the average degree $z$ is given by

$$z = \sum_{k=0}^{\infty} kp_k = G'_0(1). \quad (2)$$

The next quantity that is important for us is the distribution of the degree of a vertex we arrive at by choosing a random edge. This distribution is in some ways different from the first generating function $C$. It is twice as likely that you reach a vertex with degree 2 than a vertex with degree 1, since it has two edges instead of one. This does not mean you have twice the probability of reaching a vertex with degree 2, since when there are twice as many vertices with degree 1 than degree 2 this evens out. This leads to a probability distribution proportional to $kp_k$. However this does not sum to 1, so this has to be normalized again. We also have to compensate for the fact that we arrive at this vertex over an edge so we have to subtract one power of $x$, which gives

$$G_1(x) = \frac{\sum_{k=0}^{\infty} kp_k x^k}{x \sum_{k=0}^{\infty} kp_k} = \frac{x G'_0(x)}{x G'_0(1)} = \frac{1}{z} G'_0(x) =: \sum_{k=0}^{\infty} q_k x^k \quad (3)$$

with $q_k$ the probability for a vertex at the end of a randomly chosen edge to have degree $k$. Having the functions $G_0(x)$ and $G_1(x)$ we can look at the number of second neighbours of a random chosen vertex, where second neighbours simply means the neighbours of the neighbours. Looking at a randomly chosen vertex with degree $k$ you can see its $k$ edges as randomly chosen edges in the graph. The vertices you reach by following these edges have
generating function $G_1(x)$. This means that the pgf of the number second neighbours looks as
\[ \sum_{k=0}^{\infty} p_k (G_1(x))^k = G_0(G_1(x)). \]
Now we can find the mean number of second neighbours $z_2$ for a single vertex as
\[ z_2 = \left[ \frac{d}{dx} G_0(G_1(x)) \right]_{x=1} = G_0'(1) G_1'(1) = G_0''(1), \]
since $G_1(1) = 1$ and furthermore $G_1'(1)$ can be obtained from equation (1) and (3) as $G_0''(1)/z$ making the equation complete.
This above reasoning works only if the graphs contains no loops. This means that no neighbours should be connected to each other. Likewise also second neighbours should not be connected to neighbours of other second neighbours. The probability that this happens goes to zero as $N^{-1}$ since a neighbour is chosen out of $N$ vertices at random.

### 2.3 Component sizes

Having these core concepts we can consider some properties of the graph. First we consider the size of components in a graph. We start with the generating function $H_1(x)$ that gives the distribution of component sizes reached by traversing a randomly chosen vertex. This function explicitly excludes the giant component. When we are before the point that a giant component appears, the sizes of components are finite. Then we also know that the probability of any loops in a component goes to zero as $N^{-1}$. When traversing a randomly chosen edge the vertex we reach has degree $k$ with probability $p_k$, where the probabilities $p_k$ are equal to the coefficients of the generating function $G_0(x)$. Any edge going out of this vertex can be seen as a randomly chosen edge, which means that at any end the vertex belongs to a component satisfying the generating function of $H_1(x)$. Hence we have the generating function satisfies
\[ H_1(x) = x q_0 + x q_1 H_1(x) + x q_2 (H_1(x))^2 + \ldots \]
We can rewrite this equation as
\[ H_1(x) = x G_1(H_1(x)). \]
The next step is to determine $H_0(x)$, the generating function for the distribution of the sizes of components reached by choosing a random vertex. Since we start at a vertex, the component we find is at least one. This vertex has $k$ edges going out with probability $p_k$, where the probabilities $p_k$ are equal to the coefficients of the generating function $G_0(x)$. Any edge going out of this vertex can be seen as a randomly chosen edge, which means that at any end the vertex belongs to a component satisfying the generating function of $H_1(x)$.

### 2.4 Giant component

We shall consider the giant component of graphs in the upcoming sections. The giant component is the largest connected component of a random graph and of the order $O(N)$. We know that not every graph contains a giant component. The mean size of a component (denoted by $\langle s \rangle$) in a graph, is given by
\begin{align*}
\langle s \rangle &= H_0'(x)|_{x=1} = G_0(H_1(x)) + x G_0'(H_1(x)) \cdot H_1'(x)|_{x=1} \\
&= 1 + G_0'(1) \cdot H_1'(1).
\end{align*}
We can rewrite this equation further, making use of equations (3) and (7), which gives

\[ H'_1(x) = G_1(H(x)) + xG'_1(H(x)) \cdot H'_1(x). \] (10)

With this result we have

\[ \langle s \rangle = 1 + \frac{G_0(1)}{1 - G'_1(1)}. \] (11)

This expression diverges when \( G'_1(1) = 1 \) i.e. the moment when a giant component appears. Note that \( G'_1(1) = 1 \) is equivalent with

\[ \sum_{k=0}^{\infty} pk(k - 2) = 0. \] (12)

The more positive this sum becomes, the larger the giant component is. We can also write this sum as

\[ \sum_{k=0}^{\infty} pk(k - 1) - \sum_{k=0}^{\infty} pk = 0, \] (13)

which implies \( z_2 = z_1 \). It makes sense that the giant component appears if the number of seconds neighbours gets bigger than the number of direct neighbours.

If there is a giant component in the graph the generating function \( H_0(x) \) no longer holds for the entire graph. The function \( H_0(x) \) then generates the size of the components for all vertices in the graph excluding the giant component. As a result \( H_0(1) \neq 1 \), but it takes the value \( 1 - S \), where \( S \) is the part of the graph occupied by the giant component. We can put this into equation (7) to find that

\[ S = 1 - H_0(1) = 1 - G_0(H_1(1)). \] (14)

From now on we use \( H_1(1) \equiv u \). We can find the solution of \( H_1(x) \) in equation (7), by solving

\[ u = G_1(u), \] (15)

where \( u \) is the smallest non negative solution of this equation in the interval \( (0, 1] \). This equation still holds when there is no giant component, which would imply that \( u = 1 \) and hence \( S = 0 \).

### 2.5 Power-law with exponential cutoff

A distribution that is widely used as degree distribution for random graphs is the power law with exponential cutoff. There appear to be many real-world networks that have a distribution that behaves according to this distribution [5]. This is the reason we use this distribution to give an example of the above machinery. These distributions usually have probabilities according to

\[ p_k = Ck^{-\tau}e^{-k/\kappa} \quad \text{for} \quad k \geq 1. \] (16)

Here \( C \) is a constant that normalizes the distribution so

\[ \sum_{k=0}^{\infty} Ck^{-\tau}e^{-k/\kappa} = 1 \quad \iff \quad C = \frac{1}{\sum_{k=0}^{\infty} k^{-\tau}e^{-k/\kappa}}. \] (17)
But the part in the denominator is the \( \tau \)-th polylogarithm of \( e^{-k/\kappa} \) so \( C = (\text{Li}_\tau(e^{-1/\kappa}))^{-1} \). The parameters \( \tau \) and \( \kappa \) are both non-negative parameters. If we substitute equation (16) into equation (1) we find

\[
G_0(x) = \sum_{k=0}^{\infty} k^{-\tau} e^{-k/\kappa} e^k = \frac{\text{Li}_\tau(xe^{-1/\kappa})}{\text{Li}_\tau(e^{-1/\kappa})}.
\]  

Next we can find the function of \( G_1(x) \), which is given by Equation (3). First we need the derivative of \( G_0(x) \), which is

\[
G_0'(x) = \frac{\text{Li}_{\tau-1}(xe^{-1/\kappa})}{xe\text{Li}_\tau(e^{-1/\kappa})}.
\]

With this result and the fact that \( z = G_0'(1) \) we can get the formula

\[
G_1(x) = \frac{\text{Li}_{\tau-1}(xe^{-1/\kappa})}{\text{Li}_{\tau-1}(e^{-1/\kappa})}.
\]

### 2.6 Simulation

We assumed that there are no short loops in the graph. By simulation we check whether this assumption is reasonable. We perform simulations to look at the differences between the theoretical values and the results. For this simulation we wrote a program in Java.

In this simulation we generate undirected unipartite random graphs. First we generate \( N \) vertices with a degree \( k_i \). These vertices have half-edges which can be seen as stubs of the vertex. Then we check if the sum of degrees is even, or else there is no proper match between the stubs. Next we have to connect pairs of stubs with each other. So we shuffle the entire list of stubs and then choose consecutive pairs. We create an edge between the pairs and make them neighbours.

We use the same distribution as in Section 2.5, namely the power-law with exponential cutoff. We can generate the degrees \( k_i \) in the following way. We take a random real number \( 0 \leq r < 1 \) and then generate random integers

\[
k = \lceil -\kappa \ln(1 - r) \rceil,
\]

which generates integers \( k \geq 1 \). This transformation is proportional to the distribution of \( e^{-k/\kappa} \). The next step is that we accept this integer with probability \( k^\kappa \), which can be seen as a Bernoulli distribution. If a number is rejected we generate a new one until one is accepted.

We use a Breadth First Search to find the giant component in the graph. We start at a random vertex in the graph and then look at its neighbours. Next we look at the neighbours of the neighbours and so on. In this way we find the size of the component of a randomly chosen vertex is in. We repeat this until we have found all components and consecutively the size of the giant component.

For some values of \( \tau \) we plot the value of \( S \) against the value of \( \kappa \). The results are shown in Figure 1.
Figure 1: The size of the giant component against the value of kappa, with vertex degree distribution according to Eq. (16).

This result was achieved by choosing $N = 1000000$ for the simulation on a random graph. The points represent the result obtained from the simulation. The solid lines are the theoretical values of the size of giant component. In this simulation and all of the following simulations, we did 10 runs on 1000000 vertices. This gives a 95%-confidence interval of $O(10^{-4})$. The precision of our simulation is further disregarded because this is small enough for our results.
3 Random graphs with clustering

Next we look at clustering. This is a common phenomenon in real life: if someone has 2 friends it is very likely that they know each other too. This is not accounted for in the previous model. Most models of clustered networks are hard to compute. The ideal situation would be to generalize the standard random graphs for which one could calculate properties exactly. The problem is that the ability to calculate the properties of a random graphs uses the assumption that the graph contain no short loops. When you deliberately create triangles your assumption does not hold any more. In the paper of Newman on random graphs with clustering [4], it is shown that it is still possible to use generating functions for these kind of graphs. In this section we use this to derive the generating functions for a graph with clustering. For this we make use of the definition of the clustering coefficient given as [3]

\[ C = \frac{3 \cdot \text{(number of triangles)}}{\text{(number of triples)}} = \frac{3N_3}{N_3}. \] (22)

This can be seen as a coefficient that measures the degree of clustering, where a triple is a vertex that is connected to two other vertices.

3.1 Configuration model with clustering

For this model we use the configuration model again. There are just some adjustments that change the model to a graph with clustering. First we have to generate 2 random integers that represent the ‘single’ vertex degree and the number of triangles, called \( s_i \) and \( t_i \) respectively. Both can have any arbitrary underlying degree distribution. So when a vertex is part of single edge, then two vertices are connected and when a vertex is part of a triangle then three vertices are connected. So now we have to choose pairs of our stubs and trios of incomplete triangles, connecting those all together. Here we have the constraint that the sum of stubs must be even and the sum of triangles a triple.

3.2 Generating functions

Given any degree distribution (for both the single edges and triangles) we have the joint degree distribution \( p_{st} \) for \( s \) edges and \( t \) triangles. This means that the probability of having degree \( k \) for a vertex changes compared to the previous model, since when a vertex is part of triangle this already accounts for 2 edges. This also means that there are multiple possibilities to get to a certain degree, e.g. a vertex with degree 2 can have 1 triangle or 2 single edges. This leads us to

\[ p_k = \sum_{s,t=0}^{\infty} p_{st} 1_{(s+2t=k)} \quad \forall \ k \geq 0. \] (23)

With this we can define the generating function for an arbitrary vertex in the graph. This function has two variables this time, one for the single edges and one for the triangles. The generating function is

\[ G_p(x, y) = \sum_{s,t=0}^{\infty} p_{st} x^s y^t. \] (24)

Now we can also derive the generating function for the degree distribution \( p_k \), the total degree of a vertex. This generating function is given by

\[ f(z) = \sum_{k=0}^{\infty} p_k z^k = \sum_{k=0}^{\infty} \sum_{s,t=0}^{\infty} p_{st} 1_{(s+2t=k)} z^k = \sum_{s,t=0}^{\infty} p_{st} z^{s+2t} = G_p(z, z^2). \] (25)
With these generating functions we can calculate the clustering coefficient. It is intuitively clear that the number of triangles in the graph depends on the expected number of triangles per vertex. We can write this, for a graph with \( N \) vertices, as

\[
3N\Delta = N \sum_{s,t=0}^{\infty} tp_{st} = N \left( \frac{\partial G_p}{\partial y} \right)_{x=y=1}.
\]  

(26)

This is actually just the amount of vertices times the expected amount of triangles per vertex. This makes sense since any triangle consists of 3 vertices. Then we have the number of triples in the graph. A vertex with degree 2 is a connected triple, but a vertex with degree 3 consist of \( \binom{3}{2} = 3 \) triples; you can choose any pair of its outgoing edges. So for the number of triples we find

\[
N_3 = N \sum_{k=0}^{\infty} \binom{k}{2} p_k = N \sum_{k=0}^{\infty} \frac{1}{2} k(k-1)p_k = \frac{1}{2} N \left( \frac{\partial^2 f(z)}{\partial z^2} \right)_{z=1}.
\]  

(27)

So given any arbitrary degree distribution for the single edges and the triangles you can calculate your clustering coefficient in advance.

Next we look at a generating function like equation (3), but here we have to make a distinction between choosing an arbitrary edge in a single edge and an edge in a triangle. Just like explained before equation (3), the probability of choosing an arbitrary vertex with \( s \) single edges is proportional to the amount of single edges times its probability. The same holds for triangles. Both have to be normalized again, by respectively the expected number of single edges \( \langle s \rangle \) and triangles \( \langle t \rangle \). This results in the size-biased distributions

\[
q_{st} = \frac{s \cdot p_{st}}{\langle s \rangle}, \quad r_{st} = \frac{t \cdot p_{st}}{\langle t \rangle}.
\]  

(28)

With these we can derive the corresponding generating functions. We have to take into account that both \( q_{st} \) and \( r_{st} \) are distributions for \( s \) single edges and \( t \) triangles but both already include the edge or triangle traversed. As a result we have the following generating functions:

\[
G_q(x, y) = \sum_{s,t} q_{st} x^{s-1} y^t = \frac{1}{\langle s \rangle} \sum_{s,t} s p_{st} x^{s-1} y^t = \frac{1}{\langle s \rangle} \frac{\partial G_p}{\partial x},
\]  

\[
G_r(x, y) = \sum_{s,t} r_{st} x^s y^{t-1} = \frac{1}{\langle t \rangle} \sum_{s,t} t p_{st} x^s y^{t-1} = \frac{1}{\langle t \rangle} \frac{\partial G_p}{\partial y}.
\]  

(29)

(30)

### 3.3 Component sizes

We can also find a generating function for the size of small components, i.e. components in a graph without a giant component. For the generating functions starting from an arbitrary edge we have to make a distinction between the size of a component of a vertex accessed via a single edge \( (H_q) \) and via a triangle \( (H_r) \). Analogously of the argumentation of equation (3), we find that

\[
H_q(z) = zG_q(H_q(z), H_r(z)^2),
\]  

\[
H_r(z) = zG_r(H_q(z), H_r(z)^2).
\]  

(31)

(32)

Similarly we can find the generating function for the size of a component to which a random vertex in the graph belongs

\[
H_p(z) = zG_p(H_q(z), H_r(z)^2).
\]  

(33)
First we look at the mean size of a component to which a vertex belongs in a graph without a giant component

\[ H'_p(1) = 1 + G^{(1,0)}_p(1,1)H'_q(1) + 2G^{(0,1)}_p(1,1)H'_r(1). \]  

(34)

The derivatives of \( H_q \) and \( H_r \) can be calculated from equation (31) and (32). These derivatives depend on partial derivatives of \( g_r \) and \( g_r \) which can be found using equations (29) and (30). These can then be written as second derivatives of \( G_0 \).

\begin{align*}
  H'_q(1) &= 1 + \frac{1}{\langle s \rangle} G^{(2,0)}_p(1,1)H'_q(1) + \frac{2}{\langle s \rangle} G^{(1,1)}_p(1,1)H'_r(1) \\
  H'_r(1) &= 1 + \frac{1}{\langle t \rangle} G^{(1,1)}_p(1,1)H'_q(1) + \frac{2}{\langle t \rangle} G^{(0,2)}_p(1,1)H'_r(1)
\end{align*}

(35)

(36)

If we define \( H \) as the Hessian matrix of \( G_p(x,y) \) with \( x = y = 1 \), \( h \) as vector \((H'_q(1), H'_r(1))\) and \( 1 = (1,1) \) we can write equation (35) and (36) in matrix form as

\[ h = 1 + \alpha H \beta h \]

(37)

with

\[ \alpha = \begin{pmatrix}
\frac{1}{\langle s \rangle} & 0 \\
0 & \frac{1}{\langle t \rangle}
\end{pmatrix}, \quad \beta = \begin{pmatrix}
1 & 0 \\
0 & 2
\end{pmatrix}. \]

We can rearrange this equation as \((I - \alpha H \beta)h = 1\). If we invert this equation to determine \( h \) and use the result to solve equation (34) we can find the mean component size.

### 3.4 Giant component

Next we have a look at the giant component. This is an even more important property in graphs with clustering. The questions are when a giant component emerges and how it looks like in a graph with clustering. We look at the mean size of a component and see when this diverges. By computing we find

\[ H = \begin{pmatrix}
\langle s^2 \rangle - \langle s \rangle & \langle st \rangle \\
\langle st \rangle & \langle t^2 \rangle - \langle t \rangle
\end{pmatrix}. \]

Next we notice that the equation for the mean size of components diverges at the point where \( \text{det}(I - \alpha H \beta) = 0 \). Thus we find the following equation for the point where a giant component forms

\[ \left( \frac{\langle s^2 \rangle - \langle s \rangle}{\langle s \rangle} - 2 \right) \left( \frac{\langle t^2 \rangle - 3}{\langle t \rangle} \right) = 2 \frac{(st)^2}{\langle s \rangle \langle t \rangle}. \]

(38)

We define \( u \) to be the mean probability that a vertex reached by traversing a single edge is not part of the giant component and \( v \) the corresponding probability for a vertex traversed by a triangle. (The probability of a triangle not belonging to the giant component is \( v^2 \) since both vertices at the end of the corner should not lead to the giant component.) For a vertex not to belong to the giant component all of its connected vertices also should not belong to the giant component. For its \( k \) edges leading to \( s \) edges and \( t \) triangles this happens with probability \( u^s v^{2t} \). The degree distribution equals \( q_{st} \) so we find

\[ u = \sum_{s,t=0}^{\infty} q_{st} u^s v^{2t} = G_q(u, v^2). \]

(39)

Similarly we find

\[ v = G_r(u, v^2). \]

(40)
Then the probability for a randomly chosen vertex not to belong to the giant component is

\[ 1 - S = \sum_{s,t=0}^{\infty} p_{st} u^s v^t = G_p(u, v^2). \quad (41) \]

Here $S$ represent the fraction of the graph occupied by the giant component.

### 3.5 Simulation

As in [4] we consider a doubly Poisson degree distribution with $\mu$ and $\nu$ being mean for respectively single edges and triangles. For this distribution we have the generating function

\[ G_p(x, y) = \sum_{s,t=0}^{\infty} \frac{\mu^s}{s!} e^{-\mu} \frac{\nu^t}{t!} e^{-\nu} x^s y^t = e^{\mu(x-1)} e^{\nu(y-1)}. \quad (42) \]

By simple computations we find that $G_p(x, y) = G_q(x, y) = G_r(x, y)$, so $u = v = 1 - S$, which results in

\[ S = 1 - e^{-\mu} e^{\nu(1-S^2-1)}. \quad (43) \]

For this simulation we use $\mu + 2\nu = 2$ so that a vertex has 2 neighbours on average. From Section 3.4 we know that a giant component forms. We can also derive the clustering coefficient in terms of $\nu$ using equations (27) and (26)

\[ C = \frac{\nu}{2(\mu + 2\nu)^2 + 2\nu} = \frac{\nu}{2 + \nu}. \quad (44) \]

Figure 2 shows the theoretical results against the simulation results.

**Figure 2:** The size of the giant component against the clustering coefficient.

Figure 2 shows the results of the giant component for different values of $C$. These were derived from an undirected unipartite graphs with single edge and triangle degree according to the doubly Poisson distribution with respectively mean $\mu$ and $\nu$. We see that the
We see that the giant component decreases as the clustering coefficient increases. This is a logical result: the mean vertex degree is held constant. When the clustering coefficient increases, there are more triangles in the graph and less single edges. In a triangle the third edge is abundant in creating a larger giant component, i.e. when you leave out one of the three edges, the three vertices are still connected. Because of this a graph with more triangles has more abundant edges in creating a larger giant component as a result the giant component is smaller.

We use $\mu + 2\nu = k$ and then let $k$ run from 0 to 10; the result can be found in Figure 3.

Figure 3: The size of the giant component against the mean amount of neighbours

Independent of the distribution of the amount of edges in triangles or single edges, the giant component starts to appear at $k = 1$. With equation (35) it can be shown that independent of the ratio between $\mu$ and $\nu$ the giant component appears at $k = 1$. That is the point where the expected amount of neighbours is greater than one, so a branching process would have an infinite expectation. We see that the giant component grows slower for a higher value of $\mu$. A random graph then contains more triangles, which always have a third edge that does not contribute to a larger giant component.
4 Arbitrary size clusters

It has been shown that in a random graph with clustering with single edges and triangles the calculations on properties of the graph are still possible. Furthermore these calculations can be extended to higher order clusters \[4\]. We look at a random graph where a vertex is involved in 2 complete graphs of arbitrary size.

4.1 Generating functions

When you replace a triangle by a complete graph on \(k\) vertices, the most important change is that any point involved in this graph has \(k - 1\) neighbours instead of 2 in case of a triangle. We use two complete graphs on \(k_1\) and \(k_2\) vertices. We have a generating function for the joint degree distribution for 2 variables

\[ G_p(x, y) = \sum_{s,t=0}^{\infty} p_{st} x^s y^t, \]

where \(s\) and \(t\) account for the number of complete graphs on \(k_1\) and \(k_2\) vertices. We then derive the generating for the total degree distribution as

\[ f(z) = \sum_{k=0}^{\infty} p_k z^k = \sum_{s,t=0}^{\infty} p_{st} z^{s(k_1 - 1) + t(k_2 - 1)} = g_p(z^{k_1 - 1}, z^{k_2 - 1}). \]

With this result we can determine the clustering coefficient. The number of triangles in a complete graph on \(k\) vertices is given by \((\frac{k^3}{6})\). Thus we can derive

\[ 3N\Delta = 3N \left(\frac{k_1(k_1 - 1)(k_1 - 2)}{6} \sum_{s,t=0}^{\infty} s p_{st} + \frac{k_2(k_2 - 1)(k_2 - 2)}{6} \sum_{s,t=0}^{\infty} t p_{st}\right) \]

\[ = N \left(\frac{1}{2}(k_1 - 1)(k_1 - 2) \left(\frac{\partial G_0}{\partial x}\right)_{x=y=1} + \frac{1}{2}(k_2 - 1)(k_2 - 2) \left(\frac{\partial G_0}{\partial y}\right)_{x=y=1}\right). \]

The equation for the number of connected triples remains the same as in equation \[(27)\]. The generating functions for the distribution of the number of vertices reached by traversing an edge excluding that edge is the same as given in \[(28)\]. The probability of reaching a certain edge is proportional to its degree. The generating functions for these distributions then also satisfy equations \[(29)\] and \[(30)\].

4.2 Component sizes

First we start with the sizes of small components (i.e. graphs excluding a giant component). By an argumentation analogous to the previous generating functions \[(31)\] and \[(32)\] we can show that

\[ H_q(z) = zG_q(H_q(z)^{k_1 - 1}, H_r(z)^{k_2 - 1}), \]

\[ H_r(z) = zG_q(H_q(z)^{k_1 - 1}, H_r(z)^{k_2 - 1}). \]

Similarly we can find the generating function of the size of a component

\[ H_p(z) = zG_0(H_q(z)^{k_1 - 1}, H_r(z)^{k_2 - 1}). \]

Then the mean size of component changes too

\[ H_p'(1) = 1 + (k_1 - 1)G_0^{(1,0)}(1, 1)H_q'(1) + (k_2 - 1)G_0^{(0,1)}(1, 1)H_r'(1). \]
The derivatives of $H_q$ and $H_r$ can be calculated from equation (48) and (49).

\[
H_q'(1) = 1 + \frac{(k_1 - 1)}{\langle s \rangle} G_0^{(2,0)}(1,1) H_q'(1) + \frac{(k_2 - 1)}{\langle s \rangle} G_0^{(1,1)}(1,1) H_q'(1), \quad (52)
\]

\[
H_r'(1) = 1 + \frac{(k_1 - 1)}{\langle t \rangle} G_0^{(1,1)}(1,1) H_r'(1) + \frac{(k_2 - 1)}{\langle t \rangle} G_0^{(0,2)}(1,1) H_r'(1). \quad (53)
\]

This result can be written into the same matrix equation as before (37), the only change is that $\beta$ is different which is now

\[
\beta = \begin{pmatrix} k_1 - 1 & 0 \\ 0 & k_2 - 1 \end{pmatrix}. \quad (54)
\]

Now we can invert the equation $(I - \alpha H \beta) h = 1$ and use this to find the average component size by filling it in equation (51).

### 4.3 Giant component

We look at the point where the giant component appears in a graph. This happens when $\text{det}(I - \alpha H \beta) = 0$. This gives us the equation

\[
(k_1 - 1) \langle s^2 \rangle - k_1 \left( (k_1 - 1) \langle t^2 \rangle - k_2 \right) = (k_1 - 1)(k_1 - 1) \frac{(st)^2}{\langle s \rangle \langle t \rangle}. \quad (55)
\]

Note that using $k_1 = 2$ and $k_2 = 3$ gives the same equation as for a graph with single edges and triangles (Eq. (38)).

We can now define $u$ as the probability that a vertex, reached by traversing a complete graph on $k_1$ vertices, belongs to the giant component and $v$ a similar probability but then for a vertex reached by traversing a complete graph on $k_2$ vertices. The probability that a complete graph of $k_1$ or $k_2$ vertices does not lead to the giant component is $u^{k_1 - 1}$ and $v^{k_2 - 1}$. For any vertex not to belong to the giant component, all of its neighbours cannot belong to the giant component. So similar to the reasoning of $u$ before we find that

\[
u = \sum_{s,t=0}^{\infty} q_{st} u^{(k_1 - 1)s} v^{(k_2 - 1)t} = G_q(u^{k_1 - 1}, v^{k_2 - 1}) \quad (56)
\]

and analogous for $v$ we have

\[
v = G_r(u^{k_1 - 1}, v^{k_2 - 1}). \quad (57)
\]

The probability for a randomly chosen vertex not to belong the giant component is $G_p(u^{k_1 - 1}, v^{k_2 - 1})$. The expected size of the giant component $S$ is then

\[
S = 1 - G_p(u^{k_1 - 1}, v^{k_2 - 1}). \quad (58)
\]

### 4.4 Effects of clustering

We have a look at what the effect of clustering is on the giant component. At first we see what happens when we only use complete graphs in the random graph. We used a complete graph on 2, 3, ..., 10 vertices all from a Poisson distribution with a $k$ as the average amount of neighbours for a vertex. The theoretical results are in Figure 4.
Figure 4: The size of the giant component against the average amount of neighbours for different complete graphs.

In this figure the most left curve (blue) is for a graph in which every vertex only is in a complete graphs on 2 vertices i.e. single edges between two vertices. The most right curve (purple) is the graph in which every vertex is in a complete graph on 10 vertices. At first we see that independently of the size of the complete graph, a giant component starts to appear at $k = 1$. If we look at the point where a giant component appears for a graph with only single edges then we can prove that this is true for graphs with complete graphs according to a Poisson distribution. The point where the giant component emerges is at

$$\frac{\langle s^2 \rangle}{\langle s \rangle} - 2 = 0. \quad (59)$$

which gives us

$$\frac{k^2 + k}{k} - 2 = 0 \Leftrightarrow k = 1. \quad (60)$$

Further we notice that the size of the giant component decreases as the size of a complete graph increases. This is because a complete graph on more vertices contains more edges that are useless to make a larger giant component. Next we look at another sort of graphs. This time we do not only use complete graphs, but half of the edges are single edges. The amount of single edges per vertex is $\mu = \frac{1}{2}k$ and amount of complete graphs on $n$ vertices per vertex is $\nu = \frac{1}{2(n-1)}k$. The result is given in Figure 5.
Figure 5: The size of the giant component against the average amount of neighbours for different complete graphs combined with single edges

The most left curve is the one for only single edges, downwards to the most right curve where we have single edges and complete graphs on 10 vertices. Again all of the giant components start to appear at $k = 1$. We can show this by using equation (38). We find that

$$
\left( \frac{1}{2} k + 1 - 2 \right) \left( (n - 1) \left( \frac{1}{2(n-1)} + 1 \right) \right) = (n - 1) \left( \frac{1}{2} k \frac{1}{2(n-1)} \right)^2
$$

$$
\Leftrightarrow \frac{1}{2} k - 1 + n - 1 - n = \frac{1}{4} k
$$

$$
\Leftrightarrow k = 1.
$$

In comparison with Figure 4 we see that the curves are a lot closer to each other. All of the curves tend to a giant component that has the size of the graph a lot faster. This happens because the single edges connect the complete graphs with each other. In comparison to the example before, where we only used complete graphs, a complete graph has a higher probability to have outgoing edges. This also means that for a randomly chosen vertex, there is a higher probability that it belongs to the giant component. This results in a greater size of a giant component than in Figure 4.

Except for this there is another difference between the two figures which cannot be seen so easily.
In Figure 6 we plotted a curve of a graph with half of the edges being single edges and the other half consisting of complete graphs on 10 vertices and three graphs consisting of only complete graphs on respectively 3, 4 and 5 vertices. Despite the fact that the graph with half of the edges being complete graphs on 10 vertices starts off slower than the other three it has the greatest giant component for approximately $k > 4.5$.

So one can ask if the clustering coefficient actually does say something about the giant component in a graph or that it is dependent on the complete graphs present in the graph. We look at three different graphs, all consisting half of single edges and the other half consisting of respectively complete graphs on 4, 5 and 6 vertices. By using equation (27) and (47) we find that the clustering coefficient of a graph consisting of single edges and complete graphs of $k$ vertices is given by

$$C = \frac{\frac{1}{2}((\mu + (k-1)\nu)^2 + (k-1)(k-2)\nu)}{4 + (k-1)(k-2)\nu} = \frac{(k-1)(k-2)\nu}{4 + (k-1)(k-2)\nu}. \quad (61)$$

We used an average amount of neighbours of 2 for all graphs (i.e. $\mu + (k-1)\nu = 2$). We can express $\mu$ and $\nu$ in terms of $C$. Next we let $C$ run from 0 to 1/2, and plot this against the giant component.
Figure 7: The size of the giant component against the clustering coefficient

Here we can see that there are differences in the giant components of different graphs for equal clustering coefficients. The differences between all curves get larger up to a certain point where the difference gets smaller again. We have seen in Figure 4 that a graph with only complete graphs of more vertices has a smaller giant component. When the clustering coefficient is increased, $\mu$ goes to zero meaning that the graphs change to a graph containing only complete graphs. So if we would extend the curves of $K_{25}$ and $K_{26}$ they would get lower than $K_{24}$ and $K_{26}$ would end up under $K_{25}$. We can also conclude that the size of the giant component of a graph does not only depend on its clustering coefficient but also on which complete graphs are present in the graph. So the clustering coefficient is helpful for your understanding of a graph if you know which complete graphs are present. If we would expand this model to more than two variables (3 or more different complete graphs) this would not even hold any more, since there are infinite many linear combinations possible to get an average of $k$ neighbours. If you have complete graphs on $\{k_1, k_2, \ldots, k_n\}$ vertices, than you have an average of $k$ neighbours if $\mu_1/(k_1 - 1) + \ldots + \mu_n/(k_n - 1) = k$. Even if we know the clustering coefficient than you can still freely choose $n - 2$ of the $\mu_k$. values.
5 Percolation

Apart from the fact that the giant component is smaller when there is more clustering there are also advantages to more clustering in a graph. If we look at for example an energy network, it can happen that a certain power cable (an edge in the graph) breaks. If this happens in a triangle, there is no effect on the giant component. When this happens to a single edge, the effect can be bigger, since there is no guarantee that the two vertices are still connected. The process of some edges breaking and others not is called percolation.

5.1 Percolation on a random graph

At first we see what happens to a graph after percolation. For a graph we have a probability $\phi$. Any edge in the graph is open with probability $\phi$ and closed with probability $1-\phi$, which actually means that the edge is taken out of the graph. We are interested in the giant component of a graph after percolation. But to get there we first need to adjust our generating function. A vertex that had $n$ edges before percolation has a probability $r_k$ of $\binom{n}{k}(1-\phi)^{n-k}\phi^k$ to have $k$ edges left. Here $r_k$ follows a binomial distribution with parameters $n$ and $\phi$.

So we find that

$$G_0(\phi)(x) = \sum_{k=0}^{\infty} \sum_{n=0}^{\infty} r_n p_k x^n.$$ (62)

Similarly as the reasoning for equation (3) we can find a generating function of the degree of a vertex by traversing a random edge.

$$G_1(\phi)(x) = \sum_{k=0}^{\infty} \sum_{n=0}^{\infty} n r_n p_k x^k = \frac{1}{\phi x} G_0'(\phi)(x) =: \sum_{k=0}^{\infty} t_k x^k.$$ (63)

With these generating functions we can determine the generating function of component sizes of the graph after percolation. If we look at a random edge in the graph we know that with probability $1-\phi$ the edge is closed, so the component is 0. With probability $\phi$ the edge is open and the vertex at the end again satisfies the same relation.

$$H_1(\phi)(x) = 1 - \phi + \phi x G_1(H_1(\phi)(x)).$$ (64)

Furthermore for starting at a random vertex, which is always present, we find the same formula as in equation (8) with $H_1$ replaced by $H_{1(\phi)}$. Here there is no extra term because a vertex is still in the graph even after percolation.

$$H_0(\phi)(x) = x G_0(H_{1(\phi)}(x)).$$ (65)

We know that a giant component appears when the mean component size diverges. For the mean size of a component we find here that

$$\langle s \rangle = H_0'(\phi)(1) = G_0(H_1(\phi)(x)) + x G_0'(H_1(\phi)(x)) \frac{H_1(\phi)(x)}{1 - G_1'(1)}$$

$$= 1 + G_0'(1) H_1'(1),$$

where

$$H_1'(1) = \phi G_1(H_1(\phi)(1)) + \phi G_1'(H_1(\phi)(1)) H_1'(1)$$

$$\Rightarrow H_1'(1) = \frac{\phi}{1 - \phi G_1'(1)}.$$ (67)

So we find for the mean component size that

$$\langle s \rangle = 1 + \frac{\phi G_0'(1)}{1 - \phi G_1'(1)}.$$ (68)
This diverges when \( \phi G'_1(1) = 1 \). We can rewrite this equation likewise as in equation (12) to find that it equals

\[
\sum_{k=0}^{\infty} \sum_{n=0}^{k} p_k r_n (n-2) = 0.
\]

This actually means that a giant component exists if and only if \( z_2 > z_\phi \), which is the same as we found for a graph before percolation. We now derive a formula to calculate \( S_\phi \), the giant component in a graph after percolation. We know that \( H_0(\phi)(x) \) is the generating function for a graph without a giant component. If a giant component exists, this function still generates the sizes of the components, but only for the part of the graph excluding the giant component. Thus we find that

\[
S = 1 - H_{0(\phi)}(1) \iff S = 1 - G_{0(\phi)}(w),
\]

with \( w \equiv H_{1(\phi)}(1) \) the smallest non-negative solution of

\[
w = G_{0(\phi)}(w).
\]

**5.2 Single edges and triangles**

Now we study percolation in a random graph with triangles. We use \( u \) and \( v \) for the probability that respectively a single vertex or a vertex in a triangle is not connected to the giant component. At first we look at a single vertex reached by traversing a random edge, which is not connected to the giant component in two cases. Either the edge is taken out by the percolation, which happens with probability \( 1 - \phi \) or the edge is open, but the vertex at the end of the edge does not belong to the giant component. If this vertex has \( s \) edges and \( t \) triangles this happens with probability \( \phi u^s v^{2t} \). The distribution of \( s \) and \( t \) for this point is the same as equation (29), hence

\[
u = 1 - \phi + \phi \sum_{s,t=0}^{\infty} q_s t u^s v^{2t} = 1 - \phi (1 - G_u(u, v^2)).
\]

We can do the same thing for triangles: we again look at a vertex reached by following an edge in a triangle, but now it gets somewhat more complicated. We have three options that can happen:

- First there can be no edges open, which happens with probability \( (1 - \phi)^2 \).
- Also we can have one connected neighbour. Now one of the two edges should be open, but the third vertex cannot be open, else there would be 2 connected neighbours. This happens with a probability of \( 2\phi(1 - \phi)^2 \).
- We can also have two connected neighbours. We can either have that both of the edges are open which happens with probability \( \phi^2 \). Or one of both edges is open and the third edge is open which happens with probability \( 2\phi^2 (1 - \phi) \).

This leads to

\[
v^2 = (1 - \phi)^2 + 2\phi(1 - \phi)^2 G_r(u, v^2) + (\phi^2 + 2\phi^2(1 - \phi)) G_r(u, v^2)^2.
\]

Now we can find the size of the giant component, which is given by \( S = 1 - G_0(u, v^2) \). This can be seen as traversing a random edge in the graph, where the vertex at the end does not lead to the giant component through any of its neighbours. For different values of \( \phi \) between 0.3 and 1.0 we have looked at the size of the giant component. We did this
4 times for different cluster coefficients for a graph with single edges and triangles. The result is in Figure 8.

![Graph showing size of giant component against percolation probability]

Figure 8: The size of the giant component against the percolation probability, theoretical and simulation results.

As you can see, for a higher clustering coefficient a giant component appears earlier than for lower clustering coefficient. This is because a graph with a higher clustering coefficient has got more vertices that do not influence the giant component. In a triangle you can take out 1 edge while all vertices are still connected. At the same time, the size of the giant component is smaller as \( \phi \) approaches 1 in more clustered graphs. So a graph with more clustering seems to be better against percolation. What is further noticeable is that all curves seem to intersect at the same point.

### 5.3 Arbitrary sized clusters

In this section we find a theoretical approach for graphs after percolation. If we want to continue the reasoning as we used in Subsection 5.2 we need to find the probabilities that a certain point has \( n \) neighbours. It is however very hard to find a direct formula for this probability. It may be still possible for smaller values of \( n \) but if you take a high value the number of possibilities increases fast. Here by possibilities we mean the possible ways to construct for example 3 neighbours for a certain vertex. The vertex can have 3 direct neighbours, which can have mutual edges but no further outgoing edges. The vertex can also have 2 direct neighbours, and one of the 2 edges has 1 neighbour. A vertex with 1 direct neighbour can have 1 neighbour with also 1 neighbour or the first neighbour has 2 neighbours. So when we are looking at combination for more neighbours this increases rapidly.

We can also look at this as different connected subgraphs of a complete graph on 4 vertices, we use the same reasoning as Gilbert [6]. We define \( P_N \) as the probability that a complete graph after percolation is connected. We can compute \( P_N \) by a recurrence relation, since a vertex is always connected to 0, 1, \ldots, or \( N - 1 \) other vertices with probability 1, we
find that

\[ P_N = 1 - \sum_{k=1}^{N-1} P_k \binom{N-1}{k-1} (1 - \phi)^k (N-k) \]  \hspace{1cm} \text{(74)}

where the \( k \)-th term is the probability that a vertex is connected to \( k - 1 \) vertices. Each term in the sum is composed of three terms. The first one \( P_k \) is the probability that \( k \) vertices are connected after percolation. Then we have \( \binom{N-1}{k-1} \), which represent the possible ways to choose \( k - 1 \) connected neighbours out of a total \( N-1 \) possible neighbours. Finally there is \( (1 - \phi)^k (N-k) \) which accounts for the fact that none of the connected neighbours are allowed to have any outgoing edges to the other \( N - k \) vertices. Further we define \( P_1 = 1 \), since a vertex is always connected to itself. With this we can find that:

\[ P_1 = 1, \]
\[ P_2 = \phi \]
\[ P_3 = 3\phi^2 - 2\phi^3 \]
\[ P_4 = 16\phi^3 - 33\phi^4 + 24\phi^5 - 6\phi^6 \]

etc.

With these probabilities we can determine the probabilities \( u \) and \( v \) of respectively a vertex in a complete graph on \( k_1 \) and \( k_2 \) being connected to the giant component. Which is given by

\[ u^{k_1-1} = \sum_{n=1}^{k_1-1} \binom{k_1-1}{n-1} P_n(\phi) (1 - \phi)^n (k_1-n) g_r(u^{k_1-1}, v^{k_2-1})^{n-1} \]  \hspace{1cm} \text{(75)}

with \( P_n(\phi) = P_n \) given by equation [74]. We can derive the same formula for \( v^{k_2-1} \):

\[ v^{k_2-1} = \sum_{n=1}^{k_2-1} \binom{k_2-1}{n-1} P_n(\phi) (1 - \phi)^n (k_2-n) g_r(u^{k_1-1}, v^{k_2-1})^{n-1} \]  \hspace{1cm} \text{(76)}

Next we can find the giant component by \( S = 1 - G_0(u^{k_1-1}, v^{k_2-1}) \). The down side is that we cannot determine the point where the giant component appears for the graph after percolation. To do this we have to find generating functions that describe the graph after percolation. If we look at a vertex with \( n \) edges, then the degree of this vertex after percolation follows a binomial distribution with parameters \( n \) and \( \phi \). With this we can find the probability generating function for the vertex degrees, given by

\[ G_{\phi}(x, y) = \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \sum_{s=0}^{\infty} \sum_{t=0}^{\infty} \sum_{k=0}^{s(k_1-1) t(k_2-1)} c_k d_l x^k y^l \]

\[ = \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \sum_{s=0}^{\infty} \sum_{t=0}^{\infty} u_k v_l p_{st} x^k y^l \]

\[ = \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} c_k d_l x^k y^l \]  \hspace{1cm} \text{(77)}

where \( u_k \) is a binomial distribution with parameters \( s(k_1-1) \) and \( \phi \), \( v_l \) also with parameters \( t(k_2-1) \) and \( \phi \), \( c_k \) the probability that from any amount of complete graphs \( K_{k_1} \), \( k \) edges remain connected to a vertex and \( d_l \) the same probability for complete graphs \( K_{k_2} \). We can now find the generating function of the degree distribution for the degree distribution of a vertex found by traversing an edge in a complete graph. The probability
of picking a certain vertex is still proportional to the amount of complete graphs it is connected to. So we find (for traversing an edge in a complete graph on $k_1$ edges)

$$G_q(\phi)(x, y) = \frac{1}{x} \sum_{k=0}^{\infty} \sum_{i=0}^{\infty} k c_k d_i x^i y^j = \frac{1}{\phi(k_1 - 1) \langle s \rangle} \frac{\partial G_p(\phi)}{\partial x},$$

(78)

and similarly the generating function for traversing an edge in a graph on $k_2$ vertices

$$G_r(\phi)(x, y) = \frac{1}{\phi(k_2 - 1) \langle t \rangle} \frac{\partial G_p(\phi)}{\partial y}. $$

(79)

Here $\langle s \rangle$ and $\langle t \rangle$ represent the average amount of complete graphs on respectively $k_1$ and $k_2$ vertices per vertex. We see that these formulas look a lot like the formulas for a graph before percolation. The difference is in the $1/\phi$ term which can be easily explained. The $1/\phi$ term appears because of the percolation and $1/(k_1 - 1)$ compensates for the fact that we now look at the edges of a vertex and not at the amount of complete graphs.

Now we can determine the generating functions for the distribution of the number of vertices accessible via a vertex at the end of a randomly chosen vertex in a complete graph on $k_1$ vertices. If we pick a random edge this vertex is still in the graph with probability $\phi$, so with a probability $1 - \phi$ this edge does not lead to a component at all. The vertex we reach at the end of the edge follows the degree distribution generated by $G_q$. We now have to take into account that there are not simply $i$ connected vertices with probability $\phi^i (1 - \phi)^{k_1 - 1 - i}$ since these connected vertices can also have neighbours in this complete graph, which the generating function of $G_q$ does not take into account. Thus we find

$$H_q(\phi)(z) = 1 - \phi + \phi z G_q \left( \sum_{i=0}^{k_1-1} \binom{k_1-1}{i-1} P_i (1 - \phi)^i \langle s(\phi) \rangle^i \right)$$

(80)

and

$$H_r(\phi)(z) = 1 - \phi + \phi z G_r \left( \sum_{i=0}^{k_2-1} \binom{k_2-1}{i-1} P_j (1 - \phi)^j \langle t(\phi) \rangle^j \right).$$

(81)

Where the arguments of $G_q$ and $G_r$ can be explained as:

- $P_i$ is the probability that a complete graph of $i$ vertices is still connected after percolation.
- $(1 - \phi)^{i(k_1 - 1 - i)}$ is the probability that there are no outgoing edges from any of these $i$ vertices to other vertices in the complete graph, such that there are no other vertices connected to the original vertex.
- $\langle s(\phi) \rangle^i$ is the generating function that these $i$ vertices follow. Since we condition on the fact that they are connected we already know that they are in the component.

So they follow a different generating function.

From now on we write $a = \sum_{i=0}^{k_1-1} \binom{k_1-1}{i-1} P_i (1 - \phi)^i \langle s(\phi) \rangle^i$ and $b = \sum_{j=0}^{k_2-1} \binom{k_2-1}{j-1} P_j (1 - \phi)^j \langle t(\phi) \rangle^j$. Now we look at the generating functions of a vertex at the end of an edge of which we now is still present after percolation and in
a complete graph on respectively \( k_1 \) or \( k_2 \) vertices. Hence we have

\[
H_{s(t)}(z) = zG_p(a, b),
\]

\[
H_{t(t)}(z) = zG_r(a, b).
\]

The probability that a randomly chosen vertex in the graph belongs to a component of a given size is then generated by

\[
H_{p(t)}(z) = zG_p(a, b).
\]

Like before we can determine the mean size of a component in a graph. Here we find

\[
H'_{p(t)}(1) = G_p(1, 1) + G^{(1,0)}_p(1, 1) a'_{z=1} + G^{(0,1)}_p(1, 1) b'_{z=1},
\]

where \( a'_{z=1} = \sum_{i=0}^{k_1-1} i P_i (k_1 - 1) (1 - \phi) (k_1 - 1 - i) H'_{s(t)}(1) =: \hat{a} H'_{s(t)}(1), \)

\[
b'_{z=1} = \sum_{j=0}^{k_2-1} j P_j (k_2 - 1) (1 - \phi) (k_2 - 1 - j) H'_{s(t)}(1) =: \hat{b} H'_{s(t)}(1)
\]

Where we can derive \( H'_{s(t)}(1) \) and \( H'_{t(t)}(1) \) from equations (80) and (82).

\[
H'_{s(t)}(1) = 1 + G^{(1,0)}_s(1, 1) \hat{a} H'_{s(t)}(1) + G^{(0,1)}_s(1, 1) \hat{b} H'_{t(t)}(1),
\]

\[
H'_{t(t)}(1) = 1 + G^{(1,0)}_t(1, 1) \hat{a} H'_{s(t)}(1) + G^{(0,1)}_t(1, 1) \hat{b} H'_{t(t)}(1),
\]

We can write the result into a matrix equation. We define \( \mathbf{H} \) as the Hessian matrix of \( G_p(x, y)_{x=y=1} \), \( \mathbf{h} = (H'_{s(t)}(1), H'_{t(t)}(1)), \) \( \mathbf{1} = (1, 1), \)

\[
\alpha = \begin{pmatrix} 1/\langle s \rangle & 0 \\ 0 & 1/\langle t \rangle \end{pmatrix} \quad \text{and} \quad \beta = \begin{pmatrix} \hat{a} & 0 \\ 0 & \hat{b} \end{pmatrix}.
\]

\[
\mathbf{h} = \mathbf{1} + \alpha \mathbf{H} \beta \mathbf{h}
\]

\[
\Leftrightarrow \mathbf{1} = (\mathbf{I} - \alpha \mathbf{H} \beta) \mathbf{h}
\]

This equation diverges if \( \det(\mathbf{I} - \alpha \mathbf{H} \beta) = 0 \). So the moment that a giant component appears can be given as the equation

\[
\left(1 - \hat{a} \langle s^2 \rangle / \langle s \rangle + \hat{a} \right) \left(1 - \hat{b} \langle t^2 \rangle / \langle t \rangle + \hat{b} \right) = \hat{a} \hat{b} \langle st \rangle ^2 / \langle s \rangle \langle t \rangle.
\]

5.4 Simulation on bigger clusters

So like we said, a graph with more clustering seems to be more resilient against percolation. The question remains if this effect holds for even more clustered graphs. For this we look at graphs consisting of only complete graphs \( (K_n) \) on a varying amount of vertices. We use that all graphs have an average of 2 neighbours per vertex to compare them. So the parameter \( \nu = \frac{2}{\nu} \). Where the number of complete graphs a vertex is in contained in follows a Poisson distribution with parameter \( \nu \). We can derive the clustering coefficient of any of these graphs. Three times the amount of triangles and the amount of triples are given by by using equations (27) and (47). We find that three times the amount of triangles is given by

\[
3N_\Delta = \frac{1}{2}(n-1)(n-2) \left( \frac{\partial G_\theta(x)}{\partial x} \right)_{x=y=1} = \frac{1}{2}(n-1)(n-2) \sum_{s,t=0}^\infty sp_{st} = n - 2
\]

25
and the amount of triples is
\[ f(z) = e^{\nu(z^{n-1})}, \]
\[ f''(1) = (\nu(n-1)\nu)^2 + \nu(n-1)(n-2) = 4 + 2n - 4 = 2n. \]
So \[ C = \frac{n-2}{n}. \]

Notice first that we have plotted curves through the simulation results. So these are no actual curves, which is also why some of the curves have a kink right after the giant component appears. In all the following results (in this subsection) we have plotted curves for the simulation results too; this gives us a clearer image of what happens. What is interesting is that the giant component appears earlier when the clustering coefficient is lowered up to a certain moment. For the values we have used here, it seems that it changes when the clustering coefficient is larger than \( \frac{1}{2} \). A possible reason for this is that the giant component of the graphs is just too small to really see this effect. So now we do the same simulation for graphs with vertices with 4 neighbours on average. The result of it is presented in Figure 10.

Figure 9: Simulations of the size of the giant component against the percolation probability

Figure 10: Simulations of the size of the giant component against the percolation probability
We see that in all of the graphs the giant component starts to appear earlier than in Figure 9. This is expected since all of the graphs now have more edges so also more edges that it can miss. The order in which the giant components appear is almost the same as in Figure 10, there is a slight change since $K_2$ appears before $K_6$ now. Further we see that the differences between the points where the giant component appears have become smaller. So it seems that this is not dependent on the average amount of neighbours. So the moment the giant component starts to appear seems to be more depending on the graph than the clustering coefficient or the amount of neighbours. There are however as many combinations possible as one would want to. This makes it practically impossible to find an optimal graph for a certain demand. We look at the same graphs but with an even higher average amount of neighbours.

![Giant component graph](image)

Figure 11: Simulations of the size of the giant component against the percolation probability

What is most interesting is that the order in which the giant component appear is still the same as in Figures 9 and 10. Even though we know that a graph with bigger complete graphs can miss more edges, a giant component starts to appear earlier for smaller complete graphs. But we have to take into account that this result holds for graphs with degree distributions following a Poisson distribution. It is not said that the order is the same for graphs with other distributions.

There are more things we can vary in a graph. Next we do the same as before but now we do not take the same amount of neighbours for every graph. We look at different graphs where the common factor is that the giant component covers the same fraction of the graph, here we choose 0.6. The result can be found in Figure 12.
Here we see that the giant components appear in order of the sizes of the complete graphs. In the graph consisting of the largest complete graphs the giant component appears as first in terms of $\phi$. This result is very logical, since these graphs can miss more edges such that all edges in a complete graph are still connected. What is noticeable is that graphs of $K_3$ and $K_2$ appear much later than the other three graphs. So it seems that using bigger complete graphs is only effective up to a certain moment.

We can also look at different graphs but with the same clustering coefficient, here we take $C = \frac{1}{2}$. We know from Figure 9 that a graph with $K_4$ and 2 neighbours on average has a clustering coefficient of $\frac{1}{2}$, but here we also took a graph with $K_3$ & $K_5$ and $K_2$ & $K_6$. By using equations (47) and (27) we can derive the parameters to get $C = \frac{1}{2}$ for any of these graphs.

We see that there is a different point where the giant component appears for the different graphs. Although the graphs of $K_{3\times5}$ and $K_{2\times6}$ seem to got the same starting point they got an entirely different behaviour afterwards. The curve of $K_{2\times6}$ even seems to go linear. It is however hard to explain this. The best explanation might be that this is a
by coincidence perfect combination between the two random graphs. Having seen all of
the above we can conclude what is important for a graph to be resilient to percolation.

• The size of the clusters used in the graph, since in the graphs containing $K_2$ and $K_3$
  the giant component appears much later than in graphs with higher clusters. These
  bigger complete graphs can miss more edges such that they are still connected.

• The size of the giant component, when a graph has a bigger giant component than
  another graph it turns out to earlier form a giant component again.

With these two arguments we can also explain all of the above Figure 9-13. In the first
Figure 9-11 the graphs containing $K_4$ only the giant component appears first. We saw in
Figure 12 that the graphs containing $K_2$ and $K_3$ are a lot less resilient, even though they
have a bigger giant component they still appear later. The graphs on $K_5$ and $K_6$ appear
before $K_4$ but just a little, now the giant component of the graph on $K_4$ is a lot larger,
which explains why it appears earlier now.
6 Conclusion

In this report we have at first studied the theory of random graphs with arbitrary degree distributions as written by Newman et al. [3]. Where we start with the configuration model. We use generating functions do describe properties of the graph, like the degree distribution, second-neighbours degree distribution, component sizes and the point where a giant component appears. We have done a simulation and computed exact results for some example graphs and we saw that they match almost exactly.

Then we looked at random graphs with clustering, also based on the approach by Newman [4]. Many real-world problems involving graphs show clustering so we extended the configuration model. The extension is to two variables, which represent the single edges and the triangles of a vertex. We did again determine some core properties of this graph. We found that with very similar reasoning we could again find generating functions for the same properties as in the first model. We also defined the clustering coefficient as a measure of clustering in a graph. At the end we saw that the difference between the theoretical- and simulation results is negligible and the size of the giant component decreases as the clustering coefficient increases.

This model was further extended to two complete graphs of arbitrary size rather than single edges and triangles. The model needed some changes to make it generic and the formulas got more complicated. Yet we found all generating functions and the point where the giant component diverges. We did some simulations on some graphs. We started with graphs containing only one size of complete graphs, where the giant component is a lot smaller in graphs with bigger complete graphs. If we took graphs consisting of halve single edges and halve complete graphs, the giant components rose a whole lot faster than in a graph with only complete graphs. We also saw that graphs consisting of different complete graphs but with equal clustering coefficient have a different giant component size.

At last we looked at random graphs after percolation. We started off with the simplest form of random graphs we treated. We were able to determine every property we also determined before. Then we went on with a graph with clustering. We started with a graph with single edges and triangles for simplicity reasons, for which we determined the size of the giant component. This approach did however not give us the point where a giant component arises. So we wanted to use generating functions for a graph with clustering, for which we directly started with arbitrary sized clusters. We managed to determine the generating functions so we could also find the moment a giant component appears. Then we did some simulations on graphs with arbitrarily sized clusters with percolation. We saw that a graph consisting of only complete graphs of 4 vertices was the most resilient graph of the other graphs consisting of only complete graphs for the same average amount of neighbours. The reason that this is a complete graph on 4 vertices but on another amount can be explained by two argument.

- If we construct a random graph on complete graphs of 2 or 3 vertices, this graph is less resistant to percolation. A complete graph on 4 or more vertices can miss more edges such that the complete graph is still a connected component.
- A random graph with a larger giant component is more resilient to percolation, it has more connected vertices so a giant component appears earlier when the percolation probability goes from 0 to 1.

In a graph with only complete graphs of 1 size, complete graphs of 4 vertices seem to be the best of the two. They are form a larger connected component than complete graphs on 5 or 6 vertices and on the other hand are they more resilient than complete graphs on 2 or 3 vertices. It is hard to get a feeling for this, and if you use combinations or different distribution the effect can be evened or in the favour of a different size of complete graph.
6.1 recommendations

In Section 4 we made an extension of the random graphs with clustering model to two clusters of arbitrary size. This model could be extended to one in which you could add an arbitrary amount of different sized clusters. The model then would consist of \( n \) variables. All of the calculations would get some more complicated but in a similar way it is possible to derive the generating functions.

When we looked at the effect of clustering with triangles on the giant component we saw that all the curves intersect at exactly the same point. You could look at combinations of different graphs and see what happens when you do the same then. Is there a logical explanation why this happens? Does this also happen if you look at random graphs consisting of more complete graphs of different sizes.

In the Section 5 we opposed a model that describes the properties of random graphs with two arbitrary sized complete graphs after percolation. We did not yet test our model with an example to compare the theoretical and simulation results. Next we looked at the resilience of graphs consisting of only complete graphs. You could also look at the resilience of random graphs consisting of combinations of two complete graphs. There could also be some more research on what the effect is of the distribution you use. We only used a Poisson degree distribution in our simulations, but there might be other results with different distributions.

This entire model could also be extended to one in which you can generate an arbitrary amount of different sized clusters per vertex. There would be much more practical applications. It makes it possible to model real-world networks such that you can simulate what happens when edges break randomly across the network. If the giant component decreases by a significant size this could have enormous effect, think of an epidemic, infrastructures or power outages.
References


