TRUSTUNION

Study of a Trust Network Model

For the creation of an efficient and reliable network

Nicolas Gauvrit

CONTENTS

CONTEXT

TrustUnion's aim is to develop a relational network that will create and reinforce interpersonal trust between individual members. To achieve this, we envision a system that: (1) creates scarcity in connections, with the lowest limit of possible outgoing and incoming links for each user; and (2) is reliable because of the underlying fundamentals based on the measure of each member's reputation and the reliability of their connections.

This document firstly present a synthesis of the current scientific literature, on the crosspath of human sciences and applied mathematics, regarding the question of real friendship in social networks, the adjacent trustworthiness, the "small world" phenomenon and the optimal path-finding algorithms in relevant graphs.

Secondly, it will follow the presentation of resulting tests – using Monte Carlo methods – of the connectivity and the distance distribution between nodes in such networks depending on their respective density. The goal of this second part is to empirically determine which parameters (maximal count of outgoing and ingoing links per user, dynamic function of connection reliability, etc.) make the connectivity of the network viable while also establishing reasonable distances between two random users.

ILLUSTRATIONS LIST

Part I

CURRENT SCIENTIFIC LITERATURE SYNTHESIS

Here we present the scientific literature synthesis which, on the crosspath of human sciences and algorithmics, focuses on questions regarding the very definition of a trust network or real friendship and the optimal pathfinding algorithms in oriented and non-oriented graphs.

1 NETWORK OF REAL FRIENDSHIP

Mathematical graph theory has allowed us since the middle of the last century to formalize the notion of networks. This theory has been applied in countless domains where elements are linked to each other, like real neural networks [6], soil sciences [5], the potential deleterious interactions between smokers [10], and the network built by researchers of a specific field, where links are defined by them having co-written a scientific paper [36].

In human science studies, the usual mathematical models often rely on social observations or experiences to provide information for graphs, for example in those which examine friendship networks between teenagers [20, 41], between foreign students [16, 22], or in a professional environment [33]. This is the theme of this chapter.

1.1 Friendship networks

Online social networks, like Facebook [31], Twitter [29], Instagram [32] and Youtube [48], are extremely dense and well connected. Facebook, for example, which as of September 2018 was the most used social website, has more than 2 billion daily users[1]. Fifty million companies have a Facebook page. When presented as a graph – meaning a mathematical network model –

1 https://www.wordstream.com/blog/ws/2017/11/07/facebookstatistics

Facebook statistics show a high rate of connections per user. The average user has more than 300 friends – but only 28% are real acquaintances. The average degree of separation between two users – meaning the number of intermediaries needed to connect one person to another following friendship links – is estimated at 3.7 [2]. By comparison, it is 3.4 on Twitter when viewed as an oriented graph [3].

In the dense and well connected world of research, the average degree of separation is around 4 [37]; but if we consider human beings globally, it is often said that any two people generally have a maximum of 6 degrees of separation [11], with a much lower average.

However, this specific figure of 6 is essentially based on an old, isolated study from Travers and Milgram [46] which was criticized for its methodology [27]. Some researchers even consider that this famous six degrees of separation is a scientific myth and that we are, in reality, living in a big world, to use an expression of Kleinfeld [26].

Degrees of separation

The notion of Degrees of separation seems to have been introduced by the Hungarian author Frigues Karinthy in a short story called Chains (1929). In this short story he suggested that two people will always be separated by a maximum of 6 degrees of separation, but the notion was not clearly defined. Later, when Travers and Milgram [45, 46] studied this phenomenon in an experimental way, they interpreted "degrees of separation" as the number of people you need to go through to create a path between two people, meaning a slightly different definition to the Distance notion in graph theory. For example, if Alice knows Bob and Bob knows David, this leads to a distance of 2 between Alice and David, but only 1 degree of separation.

Generally speaking, we have the following formula:

$$degree\ of\ separation = distance - 1$$

The properties of all those networks are well known, but they simply reflect the reality of an acquaintance network. This is not the right model for "real" links that are built among people that personally know each other. For example, although the average Facebook user has 338 "friends" (with a median of 200[2]), the amount of people with whom somebody can be in a real relationship is significantly lower.

Sometimes called Dunbar's number, it describes the number of people with whom someone can maintain a genuine, regular relationship. In its famous series of articles about this specific question, based on real social network observations, Dunbar [14, 23] estimated this value at about 150.

This number doesn't correspond to the size of a person's close friend network, in which they can confidently place their trust; it's more about the acquaintance network in a wider sense. In general we can sustain a maximum of 150 acquaintances in our everyday lives. According to

2 https://www.brandwatch.com/blog/47-facebook-statistics/

The Telegraph, the average Facebook user who has, based on previous estimations, between 150 and 350 “friends”, would trust only 4 of them in the case of unexpected events[3].

A recent, more precise analysis into real connection networks [34] shows us an even more complete set of results. Using a Gaussian Mixture Model (GMM), the authors conclude that we can break down this set of 150 acquaintances into different layers in which the heart – the most trustworthy group of friends – rarely gets above 3 to 5 people.

There have also been studies into some other characteristics of real networks. For example, we can measure the local density of different kinds of network, which indicates when new links seem to be created between two people through an intermediary. In a study from 2007 [25], Jackson compares these coefficients for different kinds of network. Real friendship networks (friendships between inmates in this case) give strongly clustered results, suggesting that we often connect with with friends of friends. Looking at the internet network as a whole, however, we are almost completely randomly connected to one another. Real friendship networks or trust networks are probably more locally connected than others, with a lower proportion of distant links.

1.2 Trust

The literature on social network characteristics allows us to identify certain points that support the TrustUnion concept: Although we might know more than a hundred people, the ones we truly trust represent only a tiny proportion. The average of 3 to 5 people seems like a reasonable estimate if we have a very rigorous definition of trust.

Still, this number of 3 to 5 doesn’t correspond to a strict sense of trust; more to the people with whom we have regular contact. These are what we call “close friends”. We can surely agree that the network of trustworthy people can go beyond the close friends circle. We can trust people with whom we have only a professional relationship, for example. In this section we provide an overview of recent work in psychology regarding interpersonal trust.

Researchers have studied trust as a relational characteristic between two individuals: each unidirectional relationship is stamped by a certain level of trust from one to another. This level of trust is often measured in a global manner, but we can certainly specify our own framework for establishing trust with an individual. For instance, we may trust somebody to be on time but not to repay a loan. In the studies we have been through, the authors don’t make a precise distinction between these elements, rather aggregating them into a generic measurement of trust. The level of trust in a relationship depends on the characteristics of the person to whom it is granted: their notoriety and their proximity (oxytocine, a hormone linked the attachment, raises the propensity to give trust [28]) are the two main factors in this domain. It also depends on the characteristics of the person who is granting their trust to someone.

3 https://www.telegraph.co.uk/news/science/science-news/12108412/Facebook-users-have-155-friends-but-would-trust-justfour-in-a-crisis.html

With regards to proximity (or degree of intimacy in the relationship) and its correlation with trust given, studies in human sciences commonly find that we are more willing to trust people who belong to a group we identify with [1], whether according to nationality, culture, interests, or even some artificial or virtual connection.

Another line of research focuses on the general tendency to give trust, which is a personality trait. The more willing you are to give your trust, the greater the likelihood you will be happier in your life – but also more naïve [40]. The propensity to give trust is also linked to other personality traits; people with anxiety, for example, are less likely to be trusting [35].

Just as with some other traits, the 'trust' element of your personality evolves during childhood through to adulthood, but not that much thereafter. From childhood to adulthood, the tendency is to be able to trust more and more. As soon as we reach adulthood, we don't see much evolution of the ability to trust [42]. In a study that highlights this fact, Sutter and Kocher underline that the majority of players showed trusting behavior. The study was based on the following Trust Game:

Trust Game

The Trust Game was initially used in behavioral economics to measure the level of trust between two strangers [4]. Here are the rules:

The game involves two players that don't know each other. Player A is given $50. He has to decide how much, if anything, he will give to player B. The portion of the money given to player B will be tripled. Therefore, if player A decides to give $10, player B will receive $30 and player A will have $40 left in his hand.

In the second phase of the game, player B has the choice to give back a portion of his money to player A (or to keep it all). The amount given will also be tripled.

Mathematical game theory states that player A should keep everything because it corresponds to a Nash equilibrium. However, giving everything in the first phase can be far more advantageous to player A if player B then gives back part of what he received.

- A suspicious player A will keep everything and end up with $50.
- A trusting player A will give everything. This means player B receives $150. If player B is in a cooperative mood, he can keep $120 and give back $30, for example. The amount being again tripled, player A ultimately receives $90.

If you want to learn more about game theory and the evolution of trust, an excellent minigame dedicated to this purpose is available here: https://ncase.me/trust/.

These two kinds of research lead to the same conclusion: that we give our trust based on instinctive – rather than fully rational – criteria. Bayesian models assume that we adapt the level of trust we give to someone according to the feedback we have from their behavior.

Human beings seem to give their trust quite easily compared to more "rational" models, especially in game theory. Therefore, in behavioral economics, we usually study the real behavior of individuals in games where they can collaborate or compete. In many cases, when these games

are designed to create a social dilemma, game theory leads us to think that betrayal should prevail, but this is rarely the case [19]. In fact, we often observe a behavior adjustment, starting with cooperation. The tit-for-tat strategy becomes the natural starting point [39, 38]. It is as if we quickly learn from our opponent's reaction, and we progressively adjust our degree of trust depending on our opponent's behavior. These studies have inspired the probabilistic model of trust adjustment in artificial networks [43], closely related to what we aim to do at TrustUnion.

Certain psychometric scales of trust have been created, and some of them use up to 3 factors to measure trust [12]: trust in a soulmate, trust in friends and trust in complete strangers. The third is close to the personality traits mentioned above, whereas the second is more dependent on the characteristics of the individual to whom trust is being given.

In a Bierhoff [7] study regarding trust in an online transaction, there are 3 kinds of trust: Interpersonal trust between two specific people leads to a particular relationship and depends on both personalities. General trust – meaning the trust that an individual gives naturally to a stranger – is a personality trait. Finally, we have trust in the system. In the case of the internet, for example, this means the trust that an individual places in the internet globally. In the first two, we can observe a similar pattern as shown in the two pieces of research mentioned above. The last one seems to be an area which is relatively unexplored but quite active. That said, certain models have been developed that integrate some elements of this third path. For example, the trust an online buyer is willing to give to the merchant depends on, based on the model theory from Lee and Turban [30], 4 factors: the liability of the merchant, the subjective liability of the internet itself as a way to do commerce, the contextual situation regarding security (especially the payment method), and some variable parameters like the size of the company, the country where it operates, etc.

In the scope that we are focusing on, we can highlight the fact that people's willingness to trust is based on quite hard-to-follow criteria, but they immediately learn from their mistakes. We can also say that closeness is directly linked to trust, which is very important because it means we can have a relatively dense local trust network that leads to maximum connectivity in our network[4] (see below).

2 RELATIONAL NETWORK MODELING

2.1 Graphs

We will use the terms network and graph interchangeably. A graph is a mathematical model of a real network [9]. A graph is defined by a set of nodes (or vertices) – which, in our case, represent individuals or legal entities – and the edges or links between nodes. Each edge represents a

4 A regular and locally dense (agglomerate) graph makes it easy to link individuals all together but sometimes with longer paths. On the other hand, a completely random graph leads to shorter paths but the probability that two people aren't connected to each other is higher.

trust link between two individuals (nodes) of the network. It can be described in the form of a couple (A, B) where A and B are the related nodes. The links are unidirectional in the case of an oriented graph, or bidirectional in the case of a non-oriented graph. If the network contains the edge (A, B), it means that A gave its trust to B.

The order of a graph G is the number of vertices in G.
The size of a graph G is the number of edges in G.

Let's consider A, a vertex of an oriented graph G. The number of edges of the form (A, B) of G is called the outdegree of A, notated as od_A. The number of edges of the form (B, A) is called the indegree of A, notated as id_A. When calculating the sum of indegrees of all of the vertices, each edge is counted once. The same goes for the sum of outdegrees. We can deduce:

Theorem 1: *In an oriented graph G with vertices set E(G) for size s,*

$$\sum_{A \in E(G)} od_A = \sum_{A \in E(G)} id_A = s.$$

We deduce that the average number of outgoing connections per vertex is necessarily equal to the average number of ingoing connections.

Idea of the Theorem 1 demonstration

Each graph edge (A, B) corresponds to an "exit" accounted for in the outdegree of A. Therefore, the sum of outdegrees,

$$\sum_{A \in E(G)} od_A,$$

is equal of the total number of edges, s.

The same reasoning is valid for the indegrees: each edge will be accounted for once, in the indegree of its destination node. This demonstrates the results of Theorem 1.

The average outdegree is as follows: s/n where n is the number of vertices on the graph. The same is true for the average indegree, thus demonstrating the equality between the two.

In the case of a non-oriented graph, the degree of A simply describes the number of edges that connect to A.

A graph is r-regular if its vertices are all of degree r. An r-regular graph only occurs when r is even. We will always assume ($r \in N$) this for the following section.

Harary graph

Named after the mathematician Frank Harary, this is a particular type of graph which will be used later to build a "small world". We will consider here only r-regular Harary graphs where r is even. By definition, a Harary graph Hr,n of order n, and a parameter r is built as follows:

The graph vertices are natural integers from 0 to n-1, including modulo n, or the set

For each vertex i, we have the edges (i, i + j) and (i, i - j), for j going from 1 to r/2.

We will consider here that the graphs are oriented, but we see in the definition that (A, B) is only an edge of such a graph if (B, A) is also one.

Illustration 1 shows an example of a Harary graph, with n = 10 and r = 4;

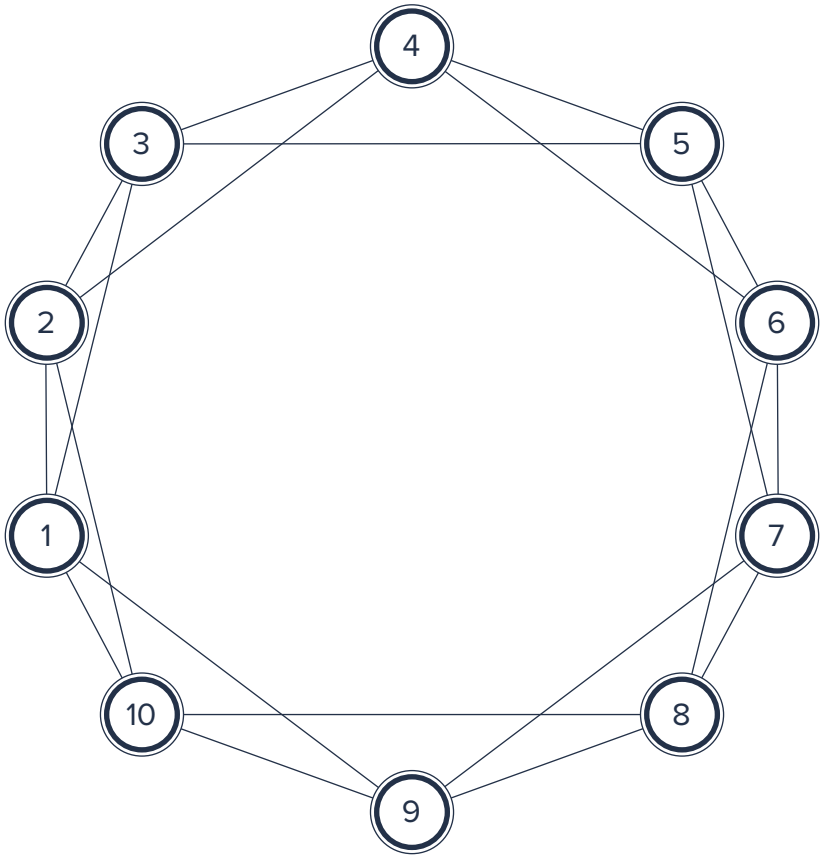

Illustration 1 – Harary graph example: H4,10.

If A and B are two vertices of the graph G, a path from A to B is a series of vertices A0 = A, A1, ..., Ak = B such as:

- For all i∈0, ..., k – 1, (A_i, A_{i+1}) is an arc of G;
- All these arcs are distinct.

The length of this path is k, meaning the number of edges constituting the path. We state that two vertices A and B in G are connected if a path exists from A to B in G. A graph is connected if every node A is connected to every node B in G.

In general, the more connections there are in the graph, the higher the chances of having a connected graph. More precisely we have the following relation:

Theorem 2: *For G a graph with order n. Notation being the degree of the vertex A. If for each vertex A and B*

$$\delta(A) + \delta(B) \geqslant n - 1,$$

Then the graph is connected.

This theorem doesn't apply to the kind of network we are trying to build here, in the sense that degrees are by principal weak, whereas our network will be a vast one, but the reciprocal of the theorem is fortunately false.

To describe a graph, we will use a set of local and global indexes:

DISTANCE: The distance between two nodes A and B, d(A, B) is the minimum number of edges that we need to go through to get from A to B. If the link (A, B) is an edge, this distance is equal to 1. Since the network is oriented (unidirectional), $d(A, B) \neq d(B, A)$ in general. If there is no path between A and B, we have $d(A, B) = \infty$.

DIAMETER: The diameter of a network is the greatest distance between two nodes. A network is fully connected if its diameter is finite.

CHARACTERISTIC PATH LENGTH: The characteristic path length L is the average distance between two nodes in a network. It's usually around 4 in a network of real acquaintances, but this does not preclude the existence of vertices much further from each other.

CLUSTERING COEFFICIENT: This coefficient C (also called the Agglomerate coefficient) measures a network's local property. The higher this coefficient, the higher the tendency of the nodes to regroup locally to form a clique (a set of nodes all linked together). For a node A, we define C_A as follows: from all the links connected to A, we consider all the available possible edges. The proportion of those available edges that are actually part of the network is C_A. The average of C_A calculated based on the set of nodes A becomes C.

Many symbolic graph representations are possible [15]. We often use an incidence matrix for example, meaning matrix M of size n^2 in which M_{ij} is the weight of the link between nodes i and j, and 0 if there is no link. This representation is inefficient and wrongly adapted to sparse graphs[5]. In the case of this kind of graph, an edge list is more efficient. This is how we deal with the programming of graphs in our first analysis in part ii.

In the following part we will mostly focus on ponderated graphs, meaning that each edge has a weight property, which is a positive integer that corresponds, in our case, to the risk associated with a link. It's the inverse representation of link reliability.

For G, a ponderated oriented graph. The non-oriented graph obtained without directional edges is the subjacent graph to G. G will be weakly connected if its subjacent graph is connected. G will be strongly connected if for all nodes A there is a path to all nodes B and likewise from B to A. Our goal is obviously to have a strongly connected oriented graph – or a non-oriented graph.

---

5 A sparse graph is a graph which has few links in relation to the amount of nodes.

As previously mentioned, the distance between two vertices, d(A, B) is the length of the shortest path from A to B. A path which achieves this distance is called a geodesic. However, when the graph is ponderated, the length of the path is not the number of edges taken from one point to another, but the sum of the weight of each edge.

If we limit the number of outgoing connections k per vertex, it will automatically raise the graph's characteristic path length. We can formalize this idea:

Theorem 3: *For G a non-ponderated oriented graph of order n such as the outdegree of each vertex can't be above k. For all vertices A, the number of vertices B such as d(A, B) ≤ s has a maximum*

$$1 + k + k^2 + \ldots + k^s = \frac{k^{s+1} - 1}{k - 1} \sim k^s.$$

This theorem shows that if we limit k too much, some distances will be very high, particularly when the diameter of the graph is high too. We can give an order of the size k^s distant vertices of s maximum steps. This means that the maximal distance t of A to another vertex B is such as k^t exceeds n, and then

$$t \ln(k) \geqslant \ln(n)$$

or

$$t \geqslant \frac{\ln(n)}{\ln(k)}.$$

For example, if we limit k = 6 and we have n = 10^de10, then t > 12, leading to the conclusion that A has a distance of at least 13 steps from B. Therefore, the diameter of the network is at least 13.

2.2 Small world

The term "small world" describes a semi-regular network. In their article Princeps, Watts and Strogatz [47] state that the name was chosen as an analogy to the small world phenomenon which states that we can easily link two people in a maximum of 7 steps (the famous 6 degrees of separation). They present some interesting characteristics of such networks.

To build this kind of network, researchers start form a regular graph of n vertices (Harary graph), where each node is connected to its closest k neighbors. Then, they replace a percentage p of those edges with random edges. The parameter p allows for adjustment of the randomness in the network. If p = 1, the network is perfectly random. In their tests, Watts and Strogatz use n = 1000 and k = 10. The value of k has to be high enough for the network to have a high probability of being connected, a demonstrated property of random networks. As a reminder, the more randomness you add, the higher the risk of having a non-connected network. The researchers rely on the following property [8]: In a random network of n nodes and nk edges, the network has a very high chance of being connected if k >> ln(n).

In the model we are looking for, this condition will be almost exactly matched. For a network of 10^9 individuals, this means that the number of connections needed per individual would equal $\ln(10^9) \approx 21$. Nevertheless, because the network isn't random, we can envision that it will be connected. This specific question will be experimentally elaborated on in part ii.

In the case that k is high enough in relation to ln(n), a perfectly regular network (p = 0) gives us

$$L \sim \frac{n}{2k} \text{ et } C \sim \frac{3}{4}.$$

When the parameter p is raised, L and C decrease (L decreases more quickly than C when p is small) until the random case where

$$L \sim \frac{\ln(n)}{\ln(k)} \text{ et } C \sim \frac{k}{n}.$$

Illustration 2 shows the evolution of C and L plotted against the parameter p. The dataset is from a simulation of 100 random small worlds matching each value of p from 0 to 1 with a 0.01 interval. Each small world has an order of 100 with k = 4. Illustration 3 gives the same information but is based on small worlds of 1000 nodes with k = 5.

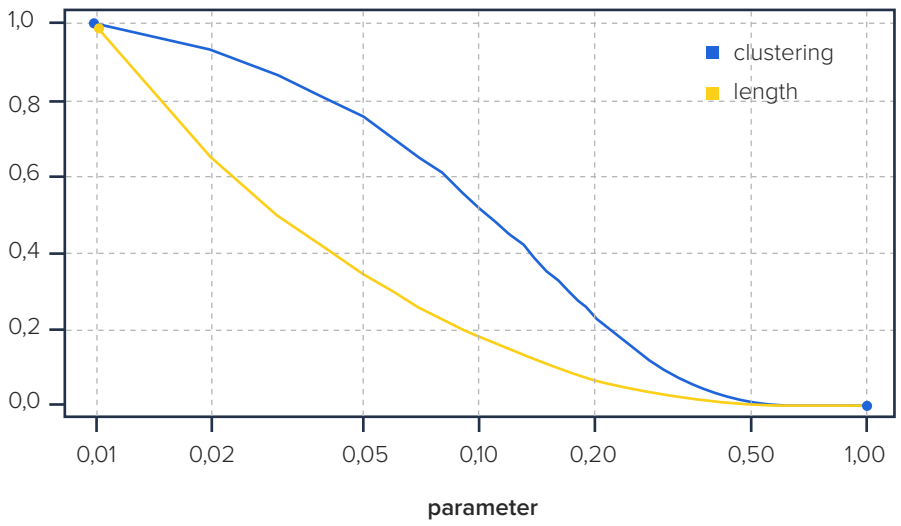

Illustration 2 – Characteristic path length and clustering coefficient of small worlds plotted against p, for small worlds with 100 nodes and k = 4, meaning there are 8 connections per vertex. Each graph plot is based on a random sample of 100 small worlds. The values of L and C are normalized to cover the interval [0,1].

In the case we want to focus on, we would like to limit L as much as possible, and the C value hasn't much importance. As a first approximation, if we want to limit L to a value of λ, we have to constrain

$$\frac{\ln(n)}{\ln(k)} \leqslant \lambda$$

then

$$\ln(k) \geqslant \frac{\ln(n)}{\lambda}$$

$$k \geqslant n^{1/\lambda}.$$

Table 1 gives us some provisional values for a network of 10^9 or 10^7 nodes. Note that the only focus here is to limit the average distance between two nodes. Also that we are relying on approximations which are usually only valid asymptotically. Furthermore, we assume that networks are always connected, which is not always the case in reality. The distance distribution will be experimentally studied in more detail in the second part.

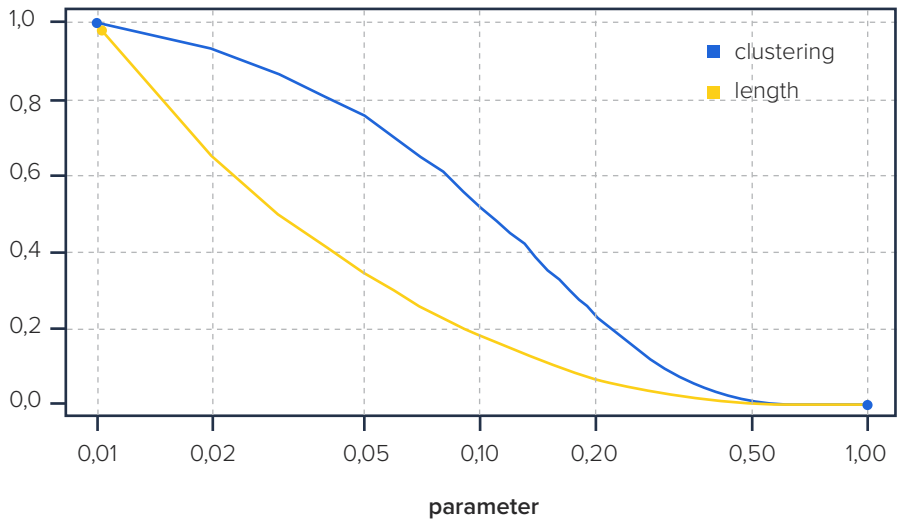

Illustration 3 – Characteristic path length and clustering coefficient of small worlds plotted against p, for small worlds with 1000 nodes and k = 5, meaning there are 10 connections per vertex. Each graph plot is based on a random sample of 100 small worlds. The values of L and C are normalized to cover the interval [0,1].

Table 1 – Number k of outgoing connections per node to ensure a maximal characteristic path length λ for a network of order n, assuming a connected graph. k values are rounded to the closest integer. Those values are based on approximations.

n	λ	k	n	λ	k
10^9	4	5	10^7	4	4
10^9	5	4	10^7	5	3
10^9	7	3	10^7	8	2

3 ALGORITHMS

In this chapter, we'll review algorithms which allow us to find optimal paths (geodesics) as reliably as possible. For this example we will use graph G where each edge is weighted to represent the reliability of the link (or, in fact, its inverse). We are using an oriented graph here, but the results are the same for the non-oriented ones TrustUnion will use. The weight assigned to each edge is a positive number or null. We will consider two kinds of algorithm.

Centralized algorithms allow us to calculate the geodesic path based on the full knowledge of the network. These algorithms are usually inefficient and aren't convenient for scaling (to, say, a billion-node network). They require regular updates for which an incredible number of paths must be calculated.

Routing algorithms are decentralized. Just like internet links on the web, they use routers on a separate layer to users. Discovering a path involves a multi-step calculation: To go from vertex A to vertex B, we link A to router Ra, and B to router Rb, and then we link Ra to Rb. The path calculation is done locally on the router layer; each router has its own table that shows it where to send the information (i.e. which router is next).

3.1 Centralized algorithms

Certain algorithms make it possible to calculate simultaneously all the shortest possible paths, or all the different paths from a starting node. We will not work with these since our goal will always be to calculate the shortest path between two specific nodes. These algorithms build a matrix n * n, where n is the number of vertices, which is iteratively updated to arrive at a shortest path matrix. Such a procedure can't sustain a heavy network, but it is still worth studying. The following detailed algorithms either calculate the shortest path between two specific nodes A and B, or the full set of shortest paths from the starting point A.
The underlying method used for all these algorithms relies on Dijkstra's algorithm [13].

Dijkstra's algorithm

Dijkstra's algorithm operates by iteratively updating a list of markers, distances and origin nodes. To find the path between A and B, the steps are as follows:

INITIALIZATION: At the beginning all the vertices are unmarked. Attribute the value $\delta(A) = 0$ to A and $\delta(C) = \infty$ to all the other nodes.

STEP 1: Identify X, the unmarked vertex having the smallest value $\delta(X)$. If X = B stop the process.

STEP 2: For each outgoing edge of X, (X, Y) with Y being unmarked, update $\delta(Y)$ using $\min(\delta(Y),\delta(X) + d(X,Y))$. In the case where this value equals $\delta(X) + d(X,Y)$, assign X as the predecessor to Y.

STEP 3: Mark X to show all its edges have been calculated. If all the vertices are marked, stop the process, otherwise go back to STEP 1.

Let's see an applied example. We will search for the shortest path between vertices 1 and 2 in the network in illustration 4. Table 2 shows the steps, line by line. The optimal path is shown in the last row. We can see that the shortest path, connecting vertices 1 and 2 via vertex 5, has a length of 5. The optimal path is then 1-5-2.

Where n is the order of G, in the worst case, the Dijkstra algorithm ends up with a table of n columns and n lines (since with each step, at least one new vertex is marked). On average, this algorithm requires half the table, meaning a matrix of $n^2/2$ cells. The computation time is then $O(n^2)$, meaning that if the number of vertices of G is multiplied by 10, the computation time is multiplied by 100. If we switch from a network of 10000 individuals to a network of 10 million (a thousand times more), the computation time increases to a million times more, which makes this algorithm very inefficient if frequently used on a vast network.

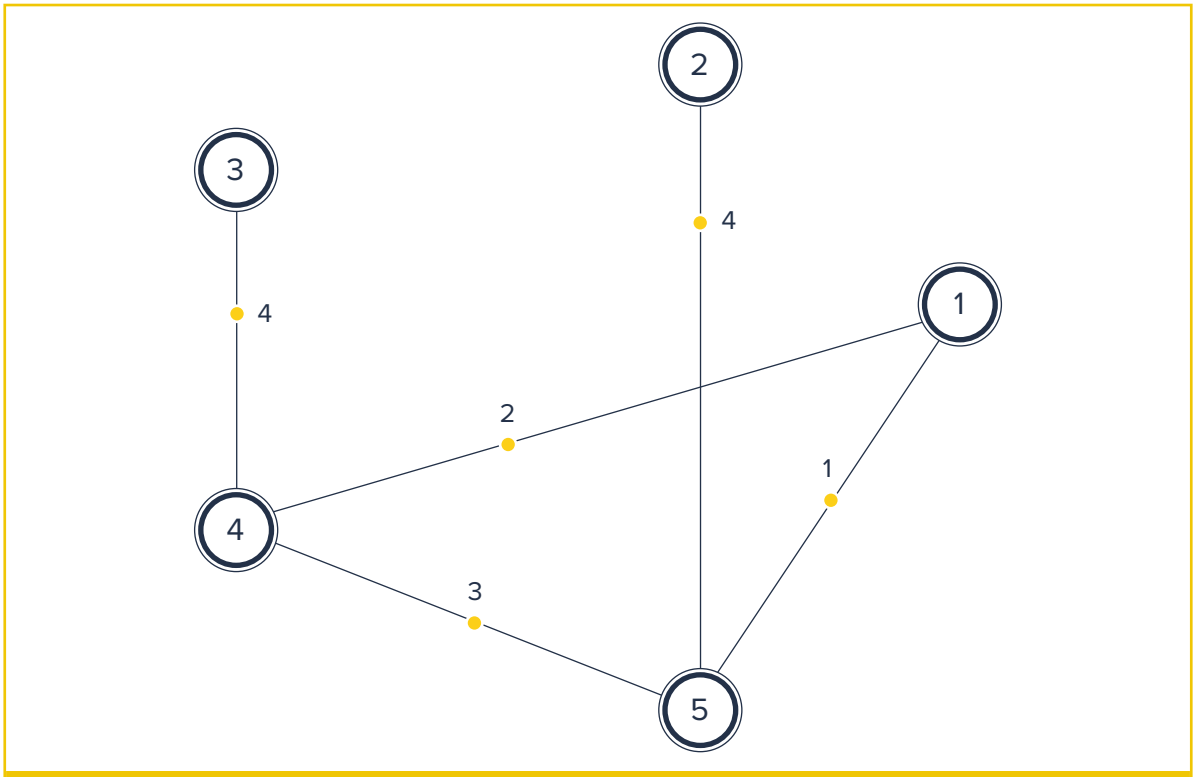

Illustration 4 – A random ponderated graph of order 5

Table 2 – The Dijkstra algorithm steps applied to the network in illustration 4. Each row indicates one round of the algorithm loop. Marked vertices are represented with a dot. The number in parenthesis indicates the preceding node in the path.

Nodes	1	2	3	4	5
Initialisation	0	∞	∞	∞	∞
Step 1	•	∞	∞	2(1)	1 (1)
Step 2	•	5(5)	∞	2(1)	•

Another kind of algorithm based on Dijkstra's method was presented in the eighties [17, 18]. We call this a polynomially bounded algorithm.

Polynomially bounded algorithm

Polynomially bounded algorithms operate as follows (P(X) is the predecessor of X in the geodesic):

INITIALIZATION: Attribute the value $\delta(A) = 0$ to A and $\delta(C) = \infty$ to all the other nodes. We create two distinct vertex sets called NOW and NEXT. To begin with, NOW = {A} and NEXT = Ø. Iteration variable k = 0.

STEP 1: Take X, a random vertex of NOW. If NOW = Ø, go to step 3.

STEP 2: Erase X from NOW. For each outgoing connection (X, Y), if $\delta(X) + d(X,Y) < \delta(Y)$ then fix $\delta(Y) = \delta(X) + d(X,Y)$, P(Y) = X, add Y to NEXT if Y is not in either NEXT or NOW. Go back to step 1.

STEP 3: If NEXT is empty, stop the process. Otherwise, increment k, push all vertices in NEXT into NOW and go back to step 1.

These algorithms are just another way to implement Dijkstra's algorithm, with a computation time also corresponding to $O(n^2)$. Nevertheless, they are more efficient in real use cases than Dijkstra's original version [24].

A last refinement of these two techniques is found in the Two-tree algorithm. In an experimental study in 1993 [21], the computation time of many algorithms was compared. In the case of a quite large but sparse network like the one we are interested in, the best procedure is a Two-tree algorithm in parallel. A detailed description of this algorithm is given on pages 58-59 of Helgason's article [49].

The idea of these algorithms is to create a tree of shortest paths, starting from the origin point and the destination simultaneously until the two paths cross each other. In other words, we use Dijkstra's algorithm from A and B in parallel (in the opposite direction for B, which means inversing the vector arrow in the case of an oriented graph), which allows us to gradually fill the geodesic table from A to B. When a vertex C is validated in both tables, we stop the process and the final path is A – C – B.

During the processing of Dijkstra's algorithm, the destination is often reached long before it is marked. If we stop the process when the C point is established, we generally have a relatively good path, but not necessarily the optimal one. This heuristic method saves time in the computation process, but in the case of a sparse graph the time saving barely exceeds 10% to 20% based on Helgason's empirical estimation.

3.2 Routing algorithm

While centralized algorithms assume that a central processor processes all possible paths with every new calculation (or updates a table of the shortest paths), routing algorithms are used when-

ever we want decentralized processing. In this case, the vertices of the graph – or at least some of them – must have enough memory and processing power to exchange data with each other.

As an example, in a routing system based on distance length vectors, each router (e.g. node of a network) always maintains a table in its memory indicating the distance to each node in the network. When a router r wants to send a message to another router s, it first analyzes its direct neighbors and sends the message to whichever is the closest to s. Furthermore, r keeps receiving information from its neighbors allowing it to update its own table. This kind of technique works very well on a network of order 10^5, and we will study the implementation for our network of order 10^9 vertices.

The email system operates based on this method, but the underlying organization behind it is structured on two layers: each user is connected to a specific router. When we send an email to another user, the first router searches for the destination user's router.

This means that route processing takes place in the router network and not the user one, which considerably reduces the processing power required and increases the speed of execution.

Thorup and Zwick algorithm

The Thorup and Zwick algorithm aims to rationally break down the overall set of nodes and create clusters (not disjoints) each having a center – a specific node in the cluster.

When we search for a path between A and B, we proceed as follows:

- If A and B are in the same cluster, then we use a classical Dijkstra algorithm (or a routing system based on distance length vectors).
- Otherwise, we process the path from A to the center of B's cluster, then the path from center of B's cluster to B.

In 2005, Thorup and Zwick provided a truly optimal routing method [44]. We will not describe the detailed version in this chapter. This method is only applicable to non-oriented graphs. In their article, the authors state that this protocol can't be applied to oriented graphs, which shows the advantages of non-oriented graphs against oriented ones. We have presented the general idea of the procedure above, but for a detailed version please refer to reference [50].

This method doesn't necessarily give the shortest path, but the result will be a maximum of 3 times longer (stretch factor) than the optimal path.

Each node A has to be able to find a path to each B in its own cluster, and to any other cluster's center. If the size of each cluster has an order of $\sqrt{n}$, there are approximately $\sqrt{n}$ clusters (provided we limit the overlapping between clusters). This means that each n vertex holds some information (for example to which vertex to send the information) for about $2\sqrt{n}$ vertices, which is very efficient; without this procedure, each vertex must hold information for n vertices. For a network of 10^{10} nodes, each node must hold and update information on $2 * 10^5$ vertices instead of 10^{10}.

When the network is geometrically organized, for example if the vertices are points on a sphere and the arc weights are the distances based on the surface of this sphere, a very simple algorithm

would consist of simply moving in the right direction of the destination node. For an abstract network, a procedure like this would be feasible if we could put this network into a manageable dimensional space k where the distance would be represented by weight. In all likelihood, in our case this is practically impossible to achieve for the following two reasons:

First, because the dimensions of a manageable space rise rapidly with the number of vertices. In a simple example, if all distances are fixed to 1, a network of 3 vertices can be represented in a plane, but not in a straight line. For 4 vertices, we have to create a 3-dimensional space, and for n points we need a space of n – 1 dimensions.

Second, we are in a dynamic network. Just the slightest update of a weight can change the overall dimensions of the space of the vertices.

Depending on which we plan to use – a centralized system or a decentralized one – there are many feasible algorithms.

4 SYNTHESIS CONCLUSION

In part ii, we will apply some tests to a random small world network in order to figure out the probability of this kind of network being connected and the average distance distribution between nodes. It will allow us to suggest future values for our network – the number of outdegrees and indegrees – to gives enough certainty about the connectivity of the network while ensuring that the average distance between nodes is moderate. We have already mentioned some minimal values for the maximal degrees allowed, but only as a rough approximation to ensure a moderate average distance assuming a connected network, which is normally a prerequisite.

Initial analysis of the literature suggests that if we have a well calibrated small world (with a low rate of random links between 1 and 10%) there is a strong probability of having a connected network, even with a small limit on degrees k (around 15 or even less). Human science literature shows that each individual has only a few really trustworthy links, but it is quite hard to identify what really goes into establishing such a trust link. The bayesian approach, in which the strength of a link is dynamically adjusted according to probability theory, may be a way to adjust the arc weights.

Setting different limits for the indegree and outdegree will result in a distortion of one of the two parameters since the outdegree and indegree should, on average, share the same value. Even if connectivity is probable, it seems to wise to prevent a lack of connectivity with solutions such as abstract hubs (virtual nodes to connect others in an “emergency” situation) or spontaneous link creation (creation of a link between two non connected nodes with poor reliance).

Regarding algorithms for finding optimal paths, two kinds can be envisioned leading to radically different architectures.

Centralized algorithms assume that there is a kind of central authority – which can have no information on the goal of the connections but which has access to all connection request inputs. Such an algorithm assumes that all intermediary nodes know which destination is being reached.

Simpler, decentralized routing algorithms (distance vectors for example) could be too slow on a network above 10^5, but some workarounds are already envisioned. A Thorup and Zwick algorithm could be the solution, but this could only provide an optimal path with limited precision.

In general, what emerges from the scientific literature is that: (1) from a human science point of view it is better to limit connections between individuals; (2) there is a high probability of having a connected network if we assume that we are in a small world situation and the number of allowed connections k is moderate (likely 10 to 15 rather than 3); and (3) that there are specific tools and algorithms available to comply with the current needs, Thorup and Zwick being one of the most interesting.

Part II

MATHEMATICAL SIMULATIONS

In this part, we present the tests performed to study connectivity conditions and the average distance distribution in a sparse trust network. The R scripts which were used for these simulations are provided. Calculations use the package igraph.

1 CONNECTIVITY

To establish the probability of connectivity in a small world, we will start with a small world of an order smaller than the target. We aim to reach a small world size of 10^9 to 10^{10} but during the test, we will work on smaller networks.

However, we believe the probability will not rise provided the ratio between the logarithm of order n of the network and the number of 2k connections per node stays constant. In other words, if we double k when we square n it shouldn't have any negative effect on the connectivity probability. This is what we test with the R script below:

Connectivity script

```
library(igraph)                                  # loading package igraph
sampleSize <- 10                                    # size of the sample
verticesNumber <- 10000                               # number of nodes
k <- 4                        # half of the number of connections per node
connexNumber <- rep(0,6)           # proportion list of non-connected for
                                                          p=0.1,...,0.5

for (j in c(1:5)){
   for (i in 1:sampleSize){
      g <- sample_smallworld(dim = 1,
      size = verticesNumber, nei = k, p = j/10)
      tab <- distance_table(g)
      if (tab$unconnected != 0){
```

```
            connexNumber[j+1] = connexNumber[j+1] + 1
        }
    }
}
points(
    c(0:5)/10,
    connexNumber/sampleSize,
    type="o",
    ylim=c(0,1),
    xlab="p",
    col="red"
)
```

A portion of the results is given in illustration 5. We can see that for a given parameter p, with a constant ln(n)/k, the probability of connectivity seems to rise slightly with n. If we suppose that p is, in the real world, small enough (around 0.3 maximum), then we can conclude that a 10^8 node small world will have a high probability of being connected with k=16, meaning a limit of 32 connections per vertex. Even so, we will be proactive and deploy a solution to avoid a non-connected small world.

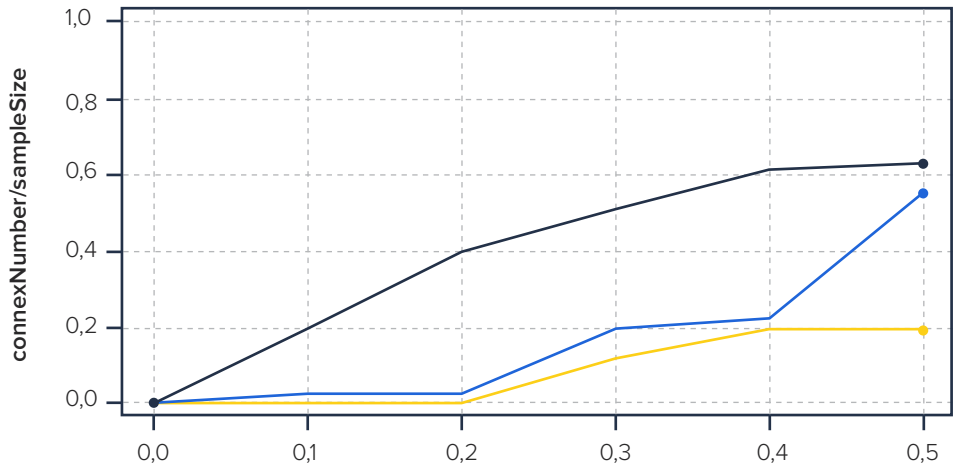

Illustration 5 – Non-connected graph ratio on a sample of 100 random small worlds (10 for the yellow curve). The black curve corresponds to a graph of 10 nodes with k = 2 (4 links per vertex). The blue curve corresponds to a graph of 100 nodes with k = 4 (8 links per vertex). The yellow curve corresponds to a graph of 10000 nodes with k = 16 (32 links per vertex).

To test the risk of non-connectivity more precisely, we've simulated 10 small worlds of order 10000, with a parameter p between 0.2 and 0.4 (which more closely resembles the maximal expected values), with different k parameters. Table 3 shows the corresponding results.

Using the heuristic of doubling k when we square the graph order, we can conclude that we would need between 20 and 30 connections per node to allow a 10^8 network to be mostly

connected. The value of 12 (k = 3 for 10^4) proved insufficient in the test. The value of 16 (k = 4 for 10^4) is the lowest limit.

P	k	Connected ratio
0,2	3	90
0,3	3	30
0,4	3	0
0,2	4	100
0,3	4	90
0,4	4	80

P	k	Connected ratio
0,2	5	100
0,3	5	90
0,4	5	100
0,2	6	100
0,3	6	100
0,4	6	100

Table 3 – Percentages of simulated small worlds found to be connected, for different values of p and k (half of the number of connections per vertex). The graph order is 10000. Each sample consists of 10 small worlds.

This suggests that the likelihood of non-connectivity is very small for small worlds of order 10000 with k = 5, so

$$\frac{\ln(n)}{k} \approx 1.84.$$

Keeping this value, we find a theoretical k for n = 10^{10} of 12.5 (rounded). In other words, even with a network of 10 billion individuals, allowing 25 connections (in and out when the graph is not oriented) is enough to obtain a probably connected network.

2 DISTANCES

In this chapter, we'll focus on path length in a sparse graph. It's not sufficient to consider the graph's characteristic path length since this doesn't reflect the probable occurrence of vertices distributed very far from each other. Here, we study the distribution of distances between two nodes in a sparse small world, according to various parameters. In this part, the notion of distance isn't linked to the reliability of the link. Distance here only refers to the number of edges crossed to join two vertices.

Illustration 6 shows the distance distribution on a sample of small worlds of order 100 with k = 2. Illustration 7 shows the distance distribution on a sample of small worlds of order 100 but with k = 3. Illustration 8 shows the distance distribution on a sample of small worlds of order 10000 with k = 6. In all the illustrations, cases with p = .1, p = .2, p = .3 and p = .4 are represented. These graphs suggest that when we square the order of a graph and we double the k parameter, the distances are not noticeably impacted. We can then suppose that a small world of 10^8 with k = 12 (so 24 connections per vertex) should have a reasonable distance distribution.

To get an even better overview, a simulation was made for small worlds of 10000, but with k = 4. These results are shown in illustration 9. We can see from this that even a k coefficient of 4 gives reasonable distances.

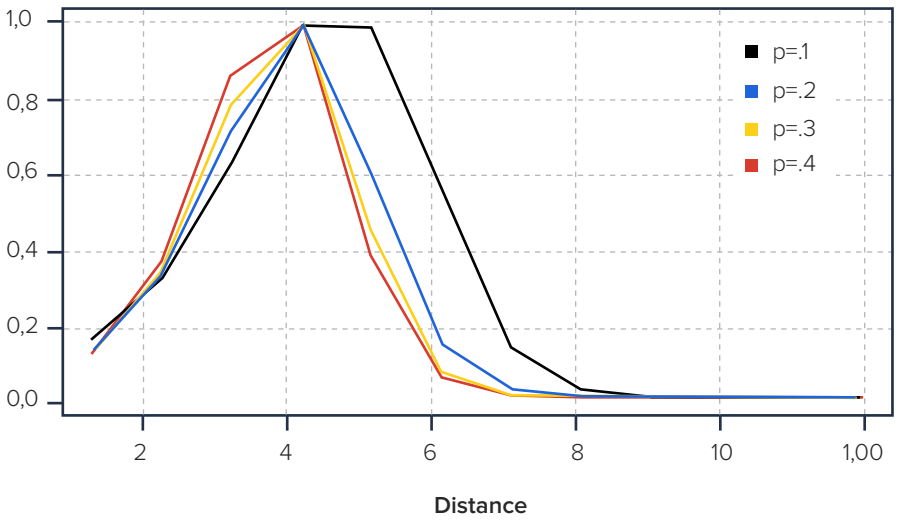

Illustration 6 – Distribution of distances between two vertices on a sample of 10 small worlds of order 100 with k = 2.

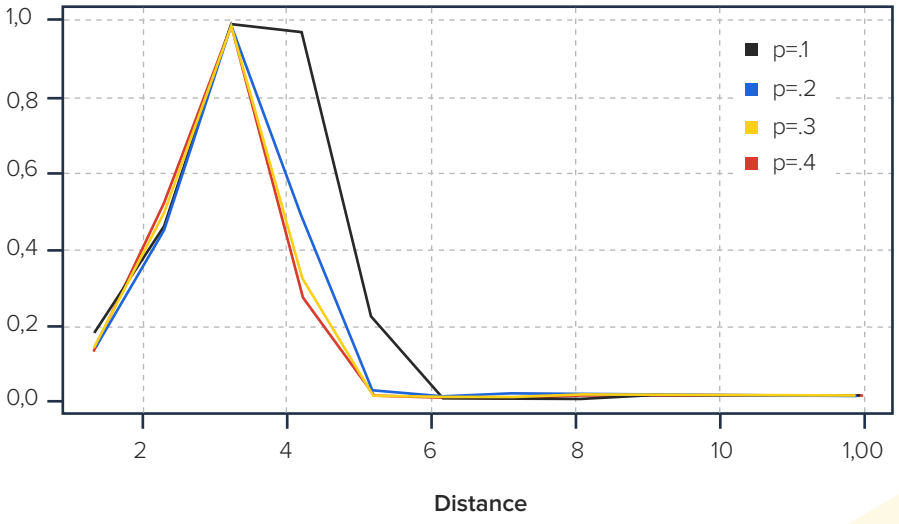

Illustration 7 – Distribution of distances between two vertices on a sample of 10 small worlds of order 100 with k = 3.

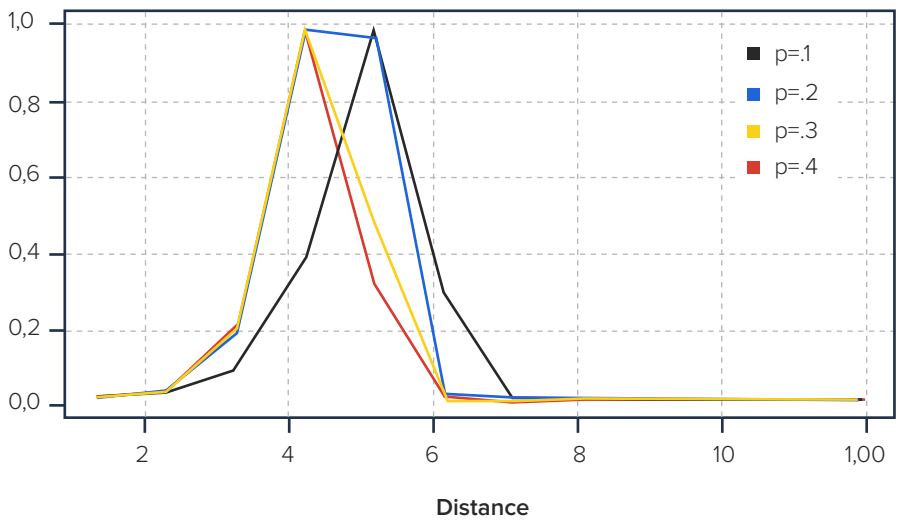

Illustration 8 – Distribution of distances between two vertices on a sample of 10 small worlds of order 10000 with k = 6.

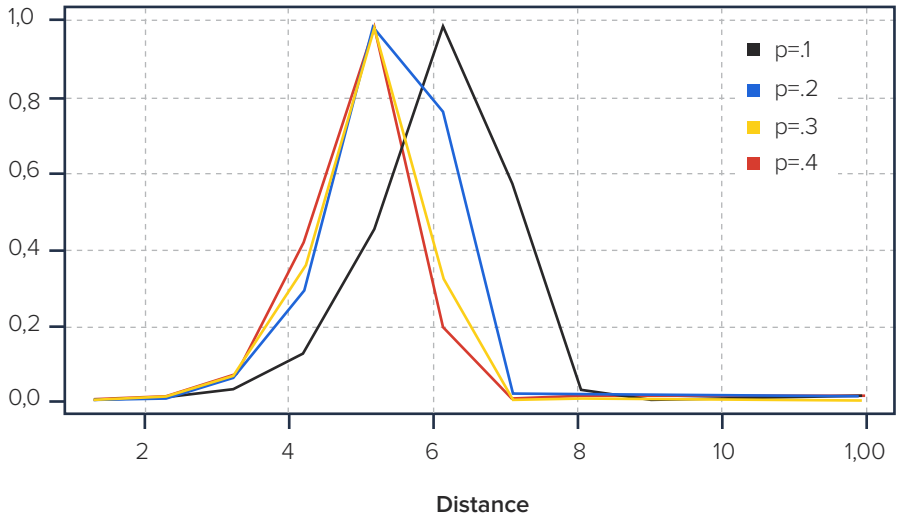

Illustration 9 – Distribution of distances between two vertices on a sample of 10 small worlds of order 10000 with k = 4.

Distances script

```
library(igraph)                                    # loading package igraph
                                                   # parameters declaration
sampleSize <- 5                                       # size of the sample
verticesNumber <- 10000                                  # number of nodes
k <- 6                                                       # k parameter
```

```
maxL <- 20                                    # maximal distance considered
                                    # distance distribution initialization
distance <- rep(0,maxL)

for (i in 1:sampleSize){
    g <- sample_smallworld(dim = 1, size = verticesNumber, nei = k, p = .1)
    tab <- distance_table(g)
    if (length(tab$res) < maxL){
      xx <- c(tab$res, rep(0,maxL-length(tab$res)))
    }
    else{
      xx <- xx[1:maxL]
    }
    distance <- distance + xx
}
distance <- distance/max(distance)
plot(
    c(1:maxL),
    distance,
    type="o",
    pch=16,
    col = "black",
    cex=.5,
    ylim = c(0,1),
    xlab="Distance",
    ylab=""
)
legend(
    "topright",
    legend = c("p=.1","p=.2","p=.3","p=.4"),
    col = c("black", "blue", "darkgreen","red"),
    pch = c(16:19),
    bty="n"
)
```

Since it's hard to know the exact degree of randomness in a real network, we've tested with an even smaller value of p for networks of order 10000 with k = 4 (Illustration 10).

These simulations suggest that the typical distance in a sparse small world of 10000 is acceptable (with a maximum below 10), even with k = 4 and p very small. It also suggests that with a graph of order 10^8, and k = 8 to k = 12, we should obtain distances within the same range, which is totally manageable.

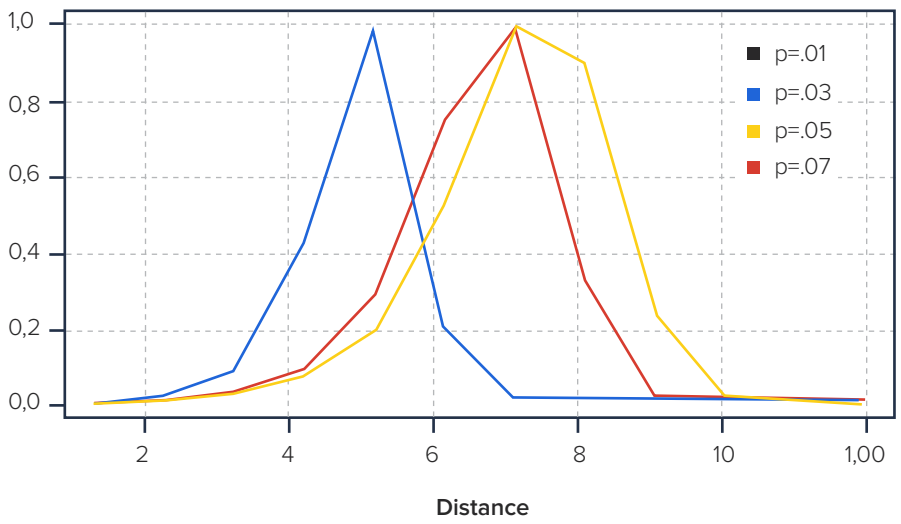

Illustration 10 – Distribution of distances between two vertices on a sample of 10 small worlds of order 10000 with k = 4 for very small values of p. Note: the curves p = .01 and p = .03 overlap.

Let's use the heuristic (which seems to have been validated by the previous tests) whereby the ratio ln(n)/k is an important parameter to estimate the path length distribution. More precisely, if ln(n)/k is constant, it seems that the path lengths stay stable. The examples simulated above suggest that the case presented in the previous chapter (ln(n)/k = 1.84) is enough to ensure short paths. This corresponds to 25 connections per node in the network.

3 CONCLUSION

TrustUnion aims to develop an ambitious network that is more reliable than those currently in use. This concept relies on the foundation of a precise network where the links between individuals are based on real life trustworthy relationships. We assume that the network is a sparse one, meaning that people will be drastically limited by the number of available links they can create.

The ambition to achieve a global network reaching 10^{10} nodes may seem contradictory to the requirement for sparsity; if a network contains few edges, it can be hard to link vertices in a small number of steps.

The analysis conducted in the two parts above shows that:

- Such a network is coherent from a human sciences perspective in the sense that trust links are indeed scarce in real life.
- Such a network is coherent mathematically because even with a small number of links (25 ingoing and outgoing links for each user), we are almost 100% certain to have a connected network with a very reasonable distance between individuals.

In practice, the proposed network should have the following characteristics:

- The property of being non-oriented (links are bidirectional)
- An almost unlimited number of users (10 billion)
- A limited number of authorized links (25 per user)
- A dynamic estimation of the reliability of links between individuals, to strengthen the network's overall reliability.
- An efficient routing algorithm, based on that of Thorup and Zwick.

4 BIBLIOGRAPHIC REFERENCES

[1] Anheier, H. and Kendall, J. (2002). Interpersonal trust and voluntary associations: examining three approaches. The British journal of sociology, 53(3):343–362.

[2] Backstrom, L., Boldi, P., Rosa, M., Ugander, J., and Vigna, S. (2012). Four degrees of separation. In Proceedings of the 4th Annual ACM Web Science Conference, pages 33–42. ACM.

[3] Bakhshandeh, R., Samadi, M., Azimifar, Z., and Schaeffer, J. (2011). Degrees of separation in social networks. In Fourth Annual Symposium on Combinatorial Search.

[4] Berg, J., Dickhaut, J., and McCabe, K. (1995). Trust, reciprocity, and social history. Games and economic behavior, 10(1):122–142.

[5] Berkowitz, B. and Ewing, R. P. (1998). Percolation theory and network modeling applications in soil physics. Surveys in Geophysics, 19(1):23–72.

[6] Bernhardt, B. C., Chen, Z., He, Y., Evans, A. C., and Bernasconi, N. (2011). Graph-theoretical analysis reveals disrupted small-world organization of cortical thickness correlation networks in temporal lobe epilepsy. Cerebral cortex, 21(9):2147–2157.

[7] Bierhoff, H.-W. and Vornefeld, B. (2004). The social psychology of trust with applications in the internet. Analyse & Kritik, 26(1):48–62.

[8] Bollobás, B. and Thomason, A. (1985). Random graphs of small order. In North-Holland Mathematics Studies, volume 118, pages 47–97. Elsevier.

[9] Chartrand, G. (2006). Introduction to graph theory. Tata McGraw-Hill Education.

[10] Christakis, N. A. and Fowler, J. H. (2008). The collective dynamics of smoking in a large social network. New England journal of medicine, 358(21):2249–2258.

[11] Collins, J. J. and Chow, C. C. (1998). It's a small world. Nature, 393(6684):409.

[12] Couch, L. L. and Jones, W. H. (1997). Measuring levels of trust. Journal of research in personality, 31(3):319–336.

[13] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. Numerische mathematik, 1(1):269–271.

[14] Dunbar, R. I. (1992). Neocortex size as a constraint on group size in primates. Journal of human evolution, 22(6):469–493.

[15] Evans, J. (2017). Optimization algorithms for networks and graphs. Routledge.

[16] [16] Furnham, A. and Alibhai, N. (1985). The friendship networks of foreign students: A replication and extension of the functional model. International journal of psychology, 20(3-4):709–722.

[17] Glover, F., Klingman, D., and Phillips, N. (1985a). A new polynomially bounded shortest path algorithm. Operations Research, 33(1):65–73.

[18] Glover, F., Klingman, D. D., Phillips, N. V., and Schneider, R. F. (1985b). New polynomial shortest path algorithms and their computational attributes. Management Science, 31(9):1106–1128.

[19] Güth, W., Ockenfels, P., and Wendel, M. (1997). Cooperation based on trust. an experimental investigation. Journal of Economic Psychology, 18(1):15–43.

[20] Haynie, D. L. (2002). Friendship networks and delinquency: The relative nature of peer delinquency. Journal of Quantitative Criminology, 18(2):99–134.

[21] Helgason, R. V., Kennington, J. L., and Stewart, B. D. (1993). The oneto-one shortest-path problem: an empirical analysis with the two-tree dijkstra algorithm. Computational Optimization and Applications, 2(1):47–75.

[22] Hendrickson, B., Rosen, D., and Aune, R. K. (2011). An analysis of friendship networks, social connectedness, homesickness, and satisfaction levels of international students. International Journal of Intercultural Relations, 35(3):281–295.

[23] Hill, R. A. and Dunbar, R. I. (2003). Social network size in humans. Human nature, 14(1):53–72.

[24] Hung, M. S. and Divoky, J. J. (1988). A computational study of efficient shortest path algorithms. Computers & Operations Research, 15(6):567–576.

[25] Jackson, M. O. and Rogers, B. W. (2007). Meeting strangers and friends of friends: How random are social networks? American Economic Review, 97(3):890–915.

[26] Kleinfeld, J. (2002a). Could it be a big world after all? the six degrees of separation myth. Society, April, 12:5–2.

[27] Kleinfeld, J. S. (2002b). The small world problem. Society, 39(2):61–66.

[28] Kosfeld, M., Heinrichs, M., Zak, P. J., Fischbacher, U., and Fehr, E. (2005). Oxytocin increases trust in humans. Nature, 435(7042):673.

[29] Kwak, H., Lee, C., Park, H., and Moon, S. (2010). What is twitter, a social network or a news media? In Proceedings of the 19th international conference on World wide web, pages 591–600. AcM.

[30] Lee, M. K. and Turban, E. (2001). A trust model for consumer internet shopping. International Journal of electronic commerce, 6(1):75–91.

[31] Lewis, K., Kaufman, J., Gonzalez, M., Wimmer, A., and Christakis, N. (2008). Tastes, ties, and time: A new social network dataset using facebook. com. Social networks, 30(4):330–342.

[32] Lim, B. H., Lu, D., Chen, T., and Kan, M.-Y. (2015). # mytweet via instagram: Exploring user behaviour across multiple social networks. In Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015, pages 113–120. ACM.=

[33] Lin, M., Prabhala, N. R., and Viswanathan, S. (2013). Judging borrowers by the company they keep: Friendship networks and information asymmetry in online peer-to-peer lending. Management Science, 59(1):17– 35.

[34] Mac Carron, P., Kaski, K., and Dunbar, R. (2016). Calling dunbar's numbers. Social Networks, 47:151–155.

[35] Mikulincer, M. (1998). Attachment working models and the sense of trust: An exploration of interaction goals and affect regulation. Journal of personality and social psychology, 74(5):1209.

[36] Newman, M. E. (2001a). Scientific collaboration networks. i. Network construction and fundamental results. Physical review E, 64(1):016131.

[37] Newman, M. E. (2001b). The structure of scientific collaboration networks. Proceedings of the national academy of sciences, 98(2):404–409.

[38] Nowak, M. and Sigmund, K. (1993). A strategy of win-stay, loseshift that outperforms tit-for-tat in the prisoner's dilemma game. Nature, 364(6432):56.

[39] Nowak, M. A. and Sigmund, K. (1992). Tit for tat in heterogeneous populations. Nature, 355(6357):250.

[40] Rotter, J. B. (1980). Interpersonal trust, trustworthiness, and gullibility. American psychologist, 35(1):1.

[41] Selfhout, M., Burk, W., Branje, S., Denissen, J., Van Aken, M., and Meeus, W. (2010). Emerging late adolescent friendship networks and big five personality traits: A social network approach. Journal of personality, 78(2):509–538.

[42] Sutter, M. and Kocher, M. G. (2007). Trust and trustworthiness across different age groups. Games and Economic Behavior, 59(2):364–382.

[43] Teacy, W. L., Luck, M., Rogers, A., and Jennings, N. R. (2012). An efficient and versatile approach to trust and reputation using hierarchical bayesian modelling. Artificial Intelligence, 193:149–185.

[44] Thorup, M. and Zwick, U. (2005). Approximate distance oracles. Journal of the ACM (JACM), 52(1):1–24.

[45] Travers, J. and Milgram, S. (1967). The small world problem. Phychology Today, 1(1):61–67.

[46] Travers, J. and Milgram, S. (1977). An experimental study of the small world problem. In Social Networks, pages 179–197. Elsevier.

[47] Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'smallworld' networks. nature, 393(6684):440.

[48] Zink, M., Suh, K., Gu, Y., and Kurose, J. (2009). Characteristics of youtube network traffic at a campus network–measurements, models, and implications. Computer networks, 53(4):501–514.

[49] Helgason's article : https://link.springer.com/content/pdf/10.1007/BF01299142.pdf

[50] Routing detailed by Thorup and Zwick : http://citeseerx.ist.psu.edu/viewdoc/download?-doi=10.1.1.94.333&rep=rep1&type=pdf

TRUSTUNION