A simple perturbative model for approximating discrete-time weighted network evolution

Linus Lee

October 2, 2016

Abstract

In this article we observe a simple, scalable model for simulating evolution of social networks. In the model, the future connection strengths between two nodes depends on the strengths of all possible paths between them, with the order of those paths determining their relative impact. We find that the computationally simple model may be a usable approximation for real-world scenarios through examining behaviors of small-scale simulations of the model. Then larger, more coupled networks are simulated to investigate the nature of larger, heavily coupled social networks such as those we observe in the realistic, more densely connected social networks (including online social networks), professional networks and in digital computer networking clusters and datacenters. Finally, we use the preliminary simulation results to draw some conclusion regarding properties of different kinds of networks that behave similarly to social networks.

Contents

1 Introduction

Complex networks evolving and interacting over time is increasingly a common and essential occurrence in day-to-day life, as social spheres of individuals grow and expand with the increased connectivity, so do the connections and complexity in networks between the machines that make it possible. These smaller, local networks are then tied together again by trade, by economics, by politics, and by foreign interests into a community of billions of nodes and interacting parties.

While the behavioral properties of networks with simple on/off connections, such as simple computer networks, are well understood and studied within the classical field of graph theory, networks with variable connection 'strengths', which comprise and better describe the majority of the networks in reality, are not as well studied.

Based on the core assumption that the complexity of a large network is the result of several emergent behaviors of a simpler unitary principle, as is often the case, we may attempt to build out a science of graphs and networks evolving through weighted connections, such as social networks, trade networks, and variable-connection computer networks.

In particular, the simple principle would have to display two emergent behaviors: long-run equilibria and stability with scale.

Long-run equilibria in this case refer to states of equilibria that are reached when no external input is applied to the system. This is a necessity, because in vacuum, social networks do not spontaneously display new behavior; a declining relationship between two groups with no stimuli will continue to decline, for example.

Stability with scale refers to the fact that, seemingly, larger networks can sustain their stability much longer and keep more connections alive than smaller networks. This can be seen from human networks and machine networks alike. Larger-population social structures will often last longer in human societies, and networks with redundancy are also often more interlinked in computer networks.

2 The mathematical model

An approximate model of social networks is achieved by a simple algorithm modeling an aspect of the behavior of true social networks.

Specifically, the algorithm (the Interactive Sum Model, or the ISM) mimics the *propagative* property of strengths of connections between nodes on a social network, that the future connection strength between nodes A and B will depend perturbatively on the strengths of ties connecting A and B to its common, shared nodes.

ISM models this behavior by the following equation,

$$\rho_{i+1}(n_1, n_2) \propto \sum_{k=0}^{\infty} \left[\prod_{j=0}^{k} \rho_i(n_0, n_k) \right] \tag{1}$$

where $\rho_i(n_a, n_b)$ denotes the strength of direct connection between nodes n_a and n_b in the ith iteration,

In approximating evolution in this way, we make several assumptions. Most prominently, we assume that, indeed, the strength of a relationship measured in some unit evolves linearly to the sum of perturbative relationships with each other through common friends. Alongside the implicit assumption of the mechanics, the model also assumes that the only "decay" of the strengths of connections is caused by the relative strengthening of other connections. In other words, the decay of connections in a network is, effectively, a consequence of a global normalization of the network's connection strengths in each iteration of evolution. Or, at the very least, such a behavior is simulated with said mathematical process.

2.1 Perturbed influence hypothesis

The ISM model is based on a single operation dictating much of social development of an organic social network and creating emergent behavior, which we call the perturbed influence hypothesis. The hypothesis states that the future strengths of the direct connection between two nodes is influenced by the current strengths of connections between nodes that the two concerned nodes "share" in common. In other words, if node A and node B have "common friends" within a single "step", this relationship will work to strengthen their future ties.

This social analogy is further quantified by formulating that for nodes A and B whose common friend is C, the future connection between the two nodes A, B will vary proportionally to the products of the connections between A, C, and B, C. Thus, stronger connections lead to stronger common connections between nodes sharing other nodes, but weaker connections are not disregarded completely, forming a smooth spectrum of possible connections.

In practice, ISM was computationally simulated by mapping a single network of a population n to an $n \times n$ square matrix representing the entire network, where the value of the matrix's element at (i, j) represents the connection strength between ith and jth nodes from the perspective of the ith node, where most of the time, $(i, j) = (j, i)$ was assumed true in simulations for sake of simplicity.

Using this modeling paradigm the iterative evolutions of the social network can be simply modeled as iterative powers of the network matrix. However, because in exponentiation of matrices of graphs nodes with more connections are counted redundantly, a normalization process is necessary, as mentioned previously.

3 Computed networks and results

In computation for this particular investigation a Ruby script[1] was used to generate matrices representing a given network of various number of nodes as a graph, where M_{ij} is the strength of the connection of node i to node j. Then the script computed their evolution using matrix operations, and the output was piped into a Node application to be postprocessed into vector graphics forms of the initial and final networks to be described more visually.

Iteration of large networks as matrix representations are computed through *normalized powers of matrices.* In other words, the $(i+1)$th generation of a network M is computed by the equation

$$M_{i+1} = M_i^n \left(N^{-1}\right)^n \tag{2}$$

where N is the *normalized network* of a given network and n is the degree of perturbation of the particular computation, usually taken as the number of nodes in a given network in small networks and a reasonably large number in many larger simulations.

The term M_i^n accounts for the evolution of n-th order connections in the next iteration through the power of adjacency matrix method derived from graph theory. However, for each iteration and for each order, this value

[1] `github.com/thesephist/webb`

must be normalized against a set standard to prevent divergent behavior of (super)exponentially increasing strengths in subsequent iterations.

The normalization network N is simply a matrix in which all nodes have "unit" connection, or, equivalently, when all connection values are at their undisturbed, normal state.

From these two standard network matrices, the next iterative matrix was calculated by normalizing each evolution of the network's matrix with its potential maximum. This, it turns out, leads many networks to the necessary convergent behavior, as we will observe in subsequent sections.

3.1 Trivial networks

In the initial investigation, two forms of *trivial networks* were simulated.

The first is the trivial network of a uniformly connected network, or UCN, with no variation in connection strength from node to node. Common sense dictates that over time, without perturbation, this kind of a network will not change in its structure. And indeed, just that is observed (Fig. 1) [2].

Informally, because no peer-to-peer connection is stronger than any other, the relative strengths of the connections remain static in a UCN, and computational results are, of course, demonstrative of that behavior.

The second kind of a trivial network is a partitioned UCN. A partitioned UCN is a network composed of smaller, perfectly uniform UCNs. In this case, for sake of variety, we observe both a 2-partition and a 3-partition UCN evolving over iterations (Fig. 2).

In both cases, the smaller, isolated sub-networks behave in identical ways to a single UCN as expected. However, by adding small perturbations and random connection variations into these networks, major changes in future evolutionary paths of these networks can be created, as will be demonstrated in proceeding sections.

[2] These graphs are not direct, numerically accurate representations of the raw computational output, but rather visual analogies derived from them. Raw output can be found in Appendix A, while a more complete set of data can be found at github.com/thesephist/webb. Rougly, the size of nodes and opacity of lines in these diagrams are scaled to the particular maxima and minima values in that particular data output

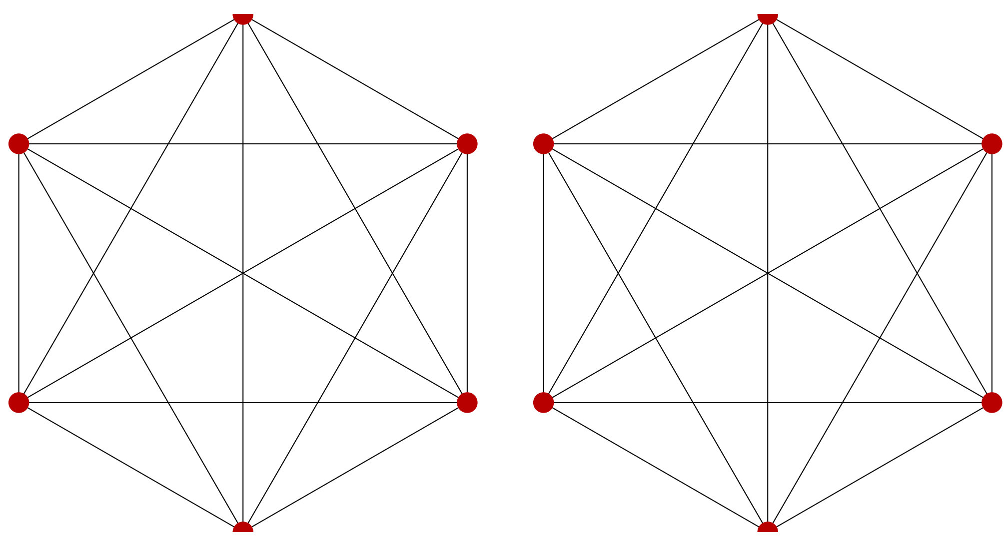


Figure 1: (Left) Initial graph of a uniformly connected network, (Right) Uniformly connected network after several iterations

3.2 Small nonuniform networks

Small networks in this investigation ranged from 8-50 nodes, testing small-scale computational generalizations to study the behavior of ISM under ideal circumstances with no random variation in connection strengths. In this investigation, six small networks were simulated.

These smaller networks are classified as either *sectionalized*, *ring*, or *cross-connected*. Alternatively, certain networks may fit into more than one of the three categories.

Sectionalized networks are networks consisting primarily of *groupings* of nodes that are bound together by little to no connection between groups. They are a generalized form of partitioned UCNs (Fig. 2), where the network is essentially a set of multiple, smaller UCNs. As a general observation, sectionalized networks' behavior are described approximately as smaller, separate networks, bound together by single cross-group connections.

This kind of a network manifests in reality as segregated social groups or geographically separated groups, such as cohorts of political affiliation, generations, or nationality. The vast majority of the population in these groups do not show much social association, except for the few members who are the social "bridges" in the inter-cohort connections.

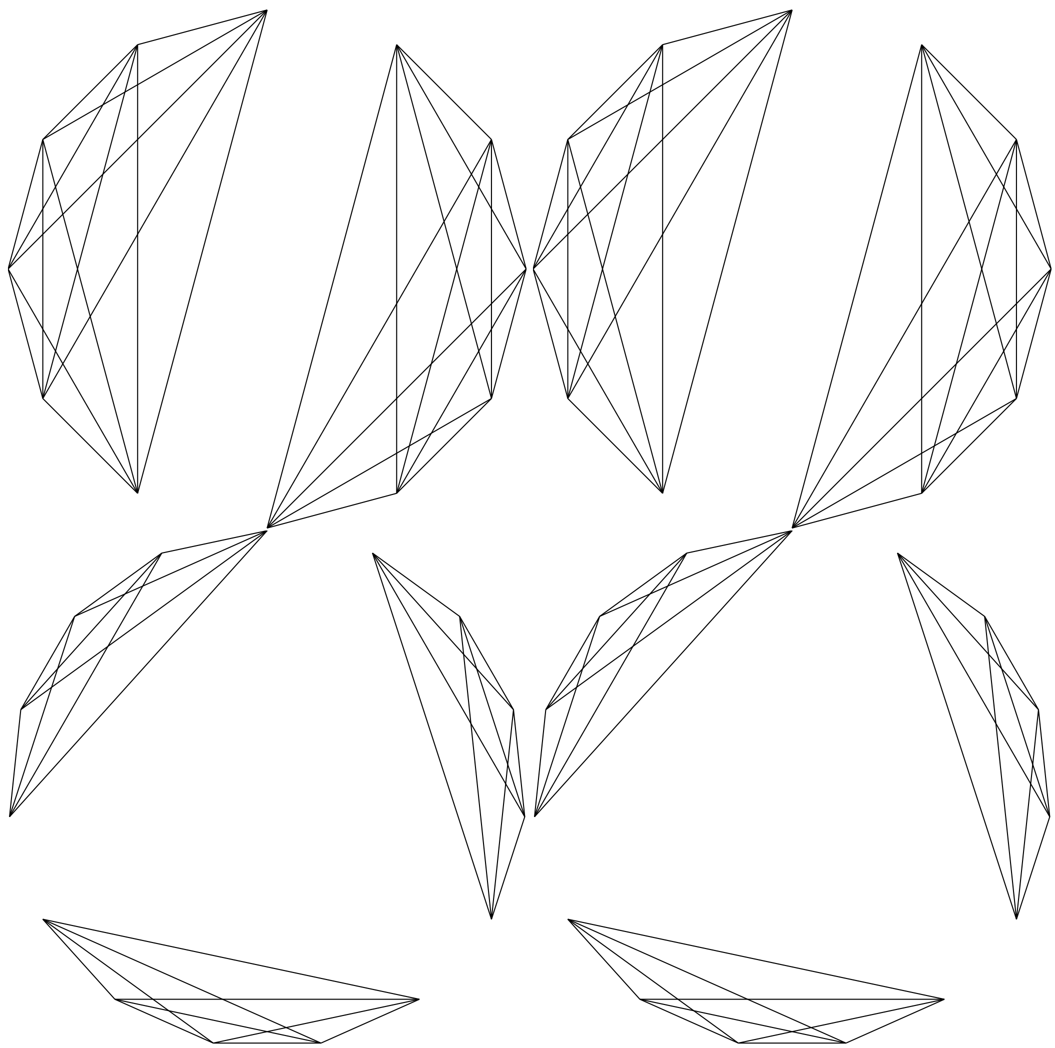


Figure 2: (Top) Initial and final graphs of a 2-partition UCN, (Bottom) the equivalents for a 3-partition UCN

Ring or ring-like networks consist overwhelmingly of nodes that are connected to two or fewer "adjacent" nodes (Fig. 3, 4). As a general trend, ring-like networks evolve out of their original structure, often showing strong and balanced development of ties *across* networks (Fig. 3).

More often than not, networks in the real world are not nearly perfect ring-like networks. Most realistic networks are *mostly* ring-like, with certain connections stretching across the network, as in Fig. 4. Examples range from networks of people to networks of indexable webpages [5]. These types of networks are notable not only because they are most representative of most real networks, but also because they exhibit the most interesting behavior.

In essence, ring-like networks have *hubs*, or nodes whose connections stretch not only adjacently to neighboring nodes but also across networks, such that they become the "connector" that ends up associating closer, other adjacent nodes to farther-away ones. This kind of growth around the hub nodes can be observed in Fig. 4.

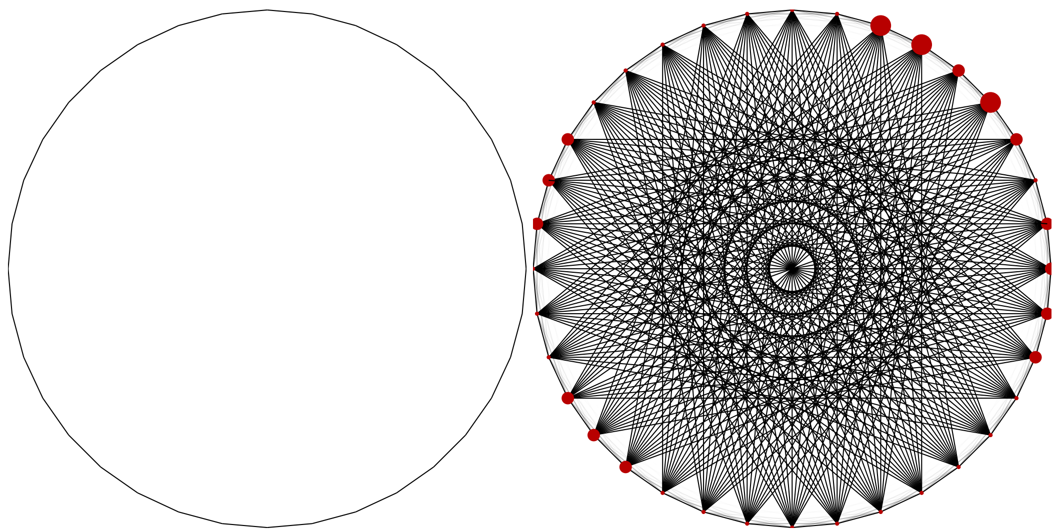


Figure 3: Perfectly ring-like network, with some computational perturbation introduced, after 10 iterations, still exhibits mostly ring-like behavior.

Ring-like networks with interconnects are often the best fit for simulating social structures that exist in the real world, where most individuals are most familiar with those living physically or digitally closest to them. In social networks, these adjacency-based networks are then punctuated by those nodes whose connections also bridge the far cross-network gaps. Take, for example, a small village whose traders are the only ones familiar with people across the village, and all other citizens know their neighbor the best.

Lastly, cross-connected networks primarily feature connections that are agnostic to any distinguishing factor between nodes. This may mean that all connection strengths are equal, as in a UCN, but cross-connected networks also include networks whose connections follow a normal distribution

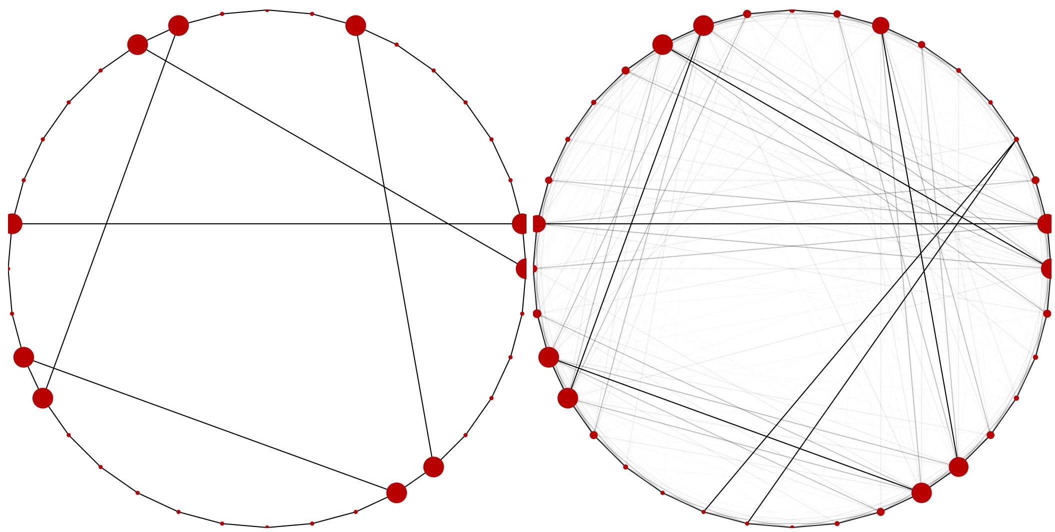


Figure 4: Ring-like network with some cross-connections introduced evolves to become a more cross-connected network.

in strength and occurrence.

Over the next several sections, we will demonstrate that the vast majority of networks evolve to become cross-connected.

3.3 Large and heavily coupled networks

In the second part of the investigation, to study the impacts of scale in perturbing behavior of networks and the effectiveness of the ISM model, the same computational process was used to simulate behavior over longer cycles on networks of node counts higher than 50.

4 Behavioral Properties

Contrary to many large-scale behaviors, social networks modeled by ISM does not lend itself easily to mathematically chaotic behavior. In fact, under even relatively large-scale shifts in the network, over iterations, the networks converge on a single, stable social equilibrium point. To observe this, three starting point networks were investigated.

4.1 The Evolution Coefficient

To better study the tendency of ISM networks in time evolution, let us define a *evolution coefficient* ω of a single iteration of a network as

$$\omega = \frac{\sqrt[n]{\prod_{k=1}^{n} \Delta\rho_k^+}}{\sqrt[m]{\prod_{k=1}^{m} \Delta\rho_k^-}} \tag{3}$$

where n and m are the numbers of connections for which the connection strength was increased and decreased, respectively, and $\Delta\rho_k^+$ and $\Delta\rho_k^-$ as the previous connection strength of the kth connection for which the connection was increased and decreased, respectively.

In other words, the evolution coefficient ω of an ISM network is defined as the ratio of the geometric mean of increasing connection strengths to the geometric mean of decreasing connection strengths.

Naturally, an ω above 1 denotes that the system is net increasing in density and connectedness, while an ω lesser than 1 indicates a system tending towards net weakening.

In observing evolution of different networks, we came across three distinctly identifiable types of behavior classifiable by the evolution of ω.

Firstly, a **converging network** is one for which all individual connection strengths ρ converge to constants and the coefficient ω converges to zero. These kinds of networks stabilize over time to be static in the long run, and the network itself converges to the equilibrium point.

A **diverging network** is one for which all individual connection strengths ρ diverge (do not approach a singular constant for each connection) and ω diverges. The former condition necessitates the latter, and this kind of a network continues to grow stronger or continues to grow weaker, eventually imploding or tearing apart within itself due to imbalances within the network. No network of this kind can converge to any equilibrium point, and therefore all diverging networks are *unstable*.

An **oscillating network** is one for which ρ does not converge, but ω converges to a constant. Connection strengths in an oscillating network evolve over time in a non-converting manner, but the total strength pulling the network together stays approximately constant. While more delicate, oscillating networks are also stable in the short run. In an oscillating network , ω represents the overall tendency of evolution for the network. Hence ω hovers around and converges to unit in the long run.

4.2 Impact of network scaling on behavior

The bulk of this investigation observed behaviors of networks of relatively small size, on the order of individual nodes. While these are useful for studying small-scale local changes as well as observing the implications of the simple algorithm, larger-scale simulations are also more relevant.

On running simulations of larger-scale networks[3], one critical behavior emerges: larger-scale networks are significantly more stable over iteration.

To confirm the behavior, 12 networks with node counts ranging from 5 to 250 were iterated using the given method over ten iterations each, starting at a random point. The small networks, usually with node count less than or around 20, came to a regression point, after which ties between certain cliques and nodes stabilized, where the network as a whole approached a static point of equilibrium, separating parts of the network.

In contrast, larger networks, especially those larger than 50 nodes, seem to stay around a dynamic equilibrium point, where the exchange of connection strengths between members and their chaotic effects are random enough and offset often enough by each other that the network can continually sustain its connectedness as a whole much longer.The average connection strength of the smaller networks fall quickly when compared to larger networks, whose average connection strengths remain almost exactly near 1.

This does not imply, however, that smaller network behaviors are not relevant in reality. These smaller networks are almost always embedded inside larger, more stable networks, where in isolation small networks will regress. Observing smaller networks offer glimpses into the behavior of smaller networks that exist either in isolation, or as a consequence of the nodes being nation-states or organizations, rather than individual people.

In practical terms, this emergent stability of larger networks provides another evidence to the credence of this model, and offers an insight into how unstable, sub-networks may link together to form networks that, quite remarkably, keeps balance between billions of people for thousands of years.

[3]Limitations on compute hardware and time prohibited the "scale" tested in this section to hundreds of nodes, from which we draw our current conclusion. However, the observed behavior is sensible and consistent enough that the conclusion is most likely also applicable to the order of millions and billions of nodes.

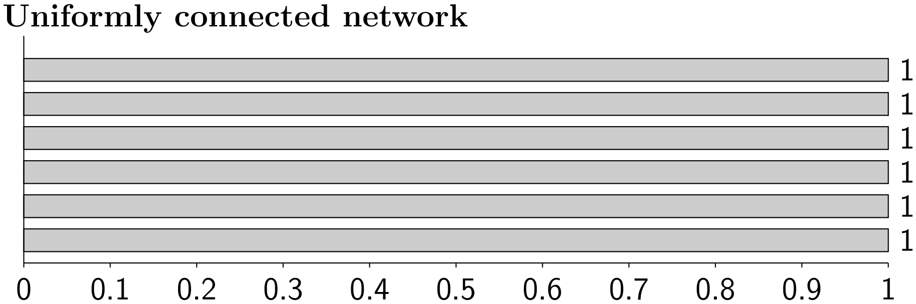
4.3 Evolution of social network topology

Here we follow the convention of defining the topology of a network by a probability distribution. Specifically, we define the topology of a network as the probability distribution $P(\Sigma_\rho)$ that a given node has an *aggregate popularity* of Σ_ρ, defined by the sum of weights of all of its connections [1, 4].

In general, following the behavior of large-scale networks, larger networks tend to be more fully connected, and through evolution stay more connected. However, in certain evolutions of sparsely connected network topologies, the connectedness can increase through evolution, because these networks do not have binary connections – ISM networks tend towards equilibrium, and such equilibria seem to most often lean towards a more fully connected network.

In the following investigations, randomly generated networks[4] of scales 5-50 nodes were simulated for 30 iterations, and their connectedness distributions were compared.

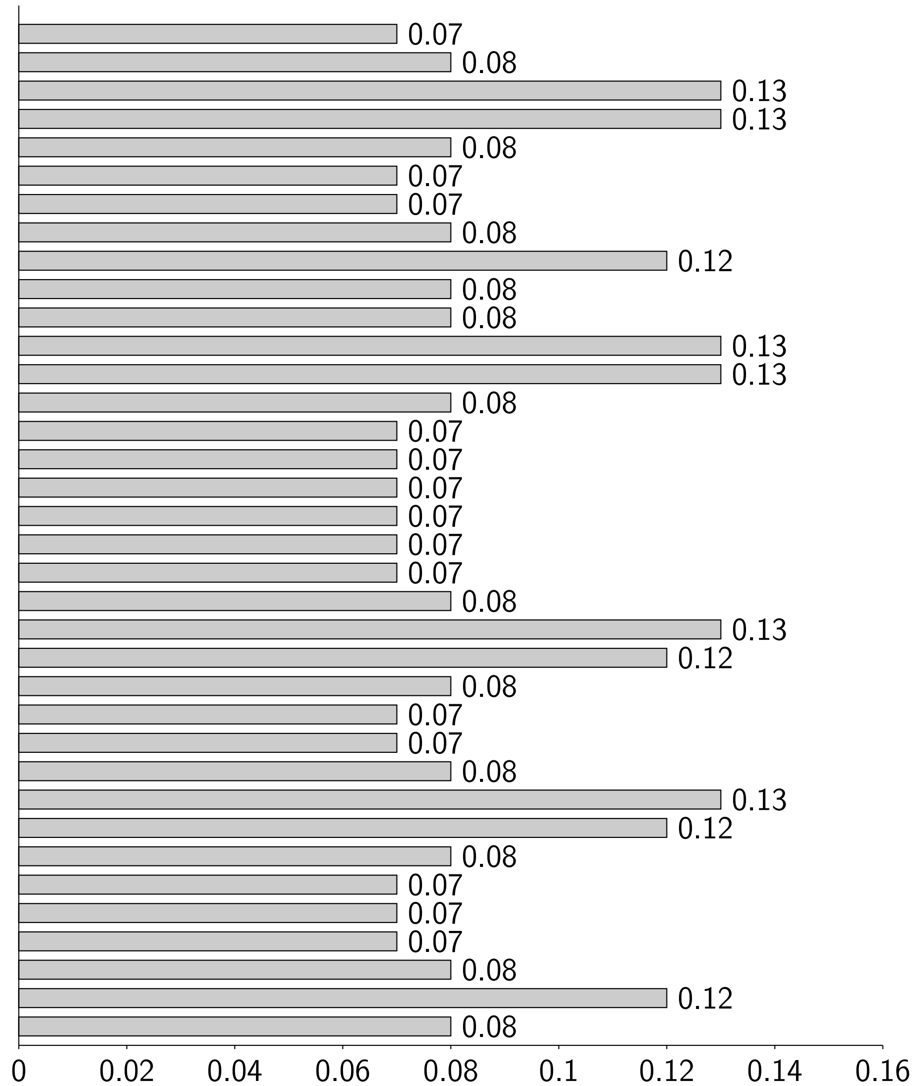
Uniformly connected network



Weakly coupled small-world network (i.e. small-world Kevin Bacon)

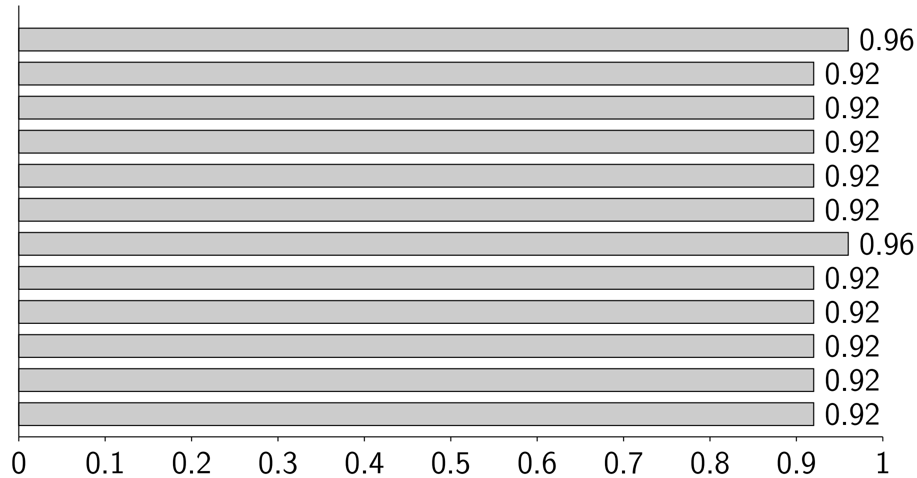
These networks are small-world networks, or networks where each node is only strongly connected to its small number of neighbors, where a couple of nodes are "superconnectors", connecting across the network. In this particular case, there were two superconnectors.

[4]Networks were randomly generated by giving each connection a uniform probability to hold values between 0.95 and 1.05.



Weakly coupled 2-section network

These networks are partitioned networks where two nodes are "superconnectors", connecting across the network. In this particular case, there were two superconnectors, the first and the seventh nodes.



In many similar scenarios, similar outcomes occur, that any initial variations subside within single-digit iterations and the network, if it is to approach an equilibrium, displays that behavior quite quickly. The general trait of the resultant equilibrium conditions seem to be that the "superconnectors" that bridge gaps between social groups remain strongest, and others even out to similar values of cumulative connection strengths.

5 Conclusions and applications

Social networks form the backbone and the roots of the emergent behavior that rules society from the individual level to the level of nations and global organizations interacting with each other, in essence for the goal of each acquiring its self interest by cooperating or disoperating with others. In that process, the understanding of how social networks evolve without such intentions, *in vacuo*, can prove essential.

We observed throughout this investigation that reasonable emergent behavior may be realistically simulated from a set of simple governing rules that apply to individual nodes and their self interests, and we observed some common behavioral traits of these networks.

Specifically, the roles and the significance of connector nodes that bridge sub-networks were clearly observed, and the two different types of behavior, the stabilizing and divergent evolution models, were classified.

Such models of complex networks built on a simplified base not only allows simpler patterns to be observed *in vacuo*, but also allows much more plausible extensions on these models to simulate and investigate evolutions of networks with more realistic and complex rules for connection weight evolution.

A Raw I/O data for computational evolution

A.1 Uniform networks

```
// Single-partition UCN

[
    [1.0, 1.0, 1.0, 1.0, 1.0, 1.0],
    [1.0, 1.0, 1.0, 1.0, 1.0, 1.0],
    [1.0, 1.0, 1.0, 1.0, 1.0, 1.0],
    [1.0, 1.0, 1.0, 1.0, 1.0, 1.0],
    [1.0, 1.0, 1.0, 1.0, 1.0, 1.0],
    [1.0, 1.0, 1.0, 1.0, 1.0, 1.0],
]

// 2- and 3-partition UCN

[
    [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],

    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]
]

// Simple Small World Scenario

[
    [1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0],
    [1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
```

```
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0,
        1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0,
        1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0],
    [1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0]
]

// Kevin Bacon Small World Scenario

[
    [1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0],
    [1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
```

```
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0,
        1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0,
        1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0],
    [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0],
    [1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0]
]
```

References

[1] Campagna, Michele, et. al., December 2014. *Spatial properties of weighted complex networks: a real world case study.* (Conference Text)

[2] Tokuyama, Takeshi. December 2007. *Algorithms and Computation.* 18th International Symposium, ISAAC

[3] Vespignani, Allesandro, et. al., 30 December 2004. *Modeling the evolution of weighted networks.* Physical Review, E 70.

[4] Wang, Ying-Hai, et. al., 2 February 2008. *Properties of weighted complex networks.*

[5] Stromberg, Joseph, 18 February 2013. *Any Two Pages on the Web Are Connected by 19 Clicks or Less.* Smithsonian Magazine.