A network reduction method inducing scale-free degree distribution

Nicolas Martin, Paolo Frasca, Carlos Canudas de Wit

A network reduction method inducing scale-free degree distribution

Nicolas Martin*, Paolo Frasca*, Carlos Canudas-de-Wit†

***Abstract*— This paper deals with the problem of graph reduction towards a scale-free graph while preserving a consistency with the initial graph. This problem is formulated as a minimization problem and to this end we define a metric to measure the scale-freeness of a graph and another metric to measure the similarity between two graphs with different dimensions, based on spectral centrality. We also want to ensure that if the initial network is a flow network, the reduced network preserves this property. We explore the optimization problem and, based on the gained insights, we derive an algorithm allowing to find an approximate solution. Finally, the effectiveness of the algorithm is shown through a simulation on a Manhattan-like network.**

I. Introduction

Many physical systems can be represented as networks and physical phenomena as processes over networks. Hence the use of these mathematical objects in modelling is becoming more and more common as the computing power and the interest for big data increases. Large networks are common in several fields like transportation, power grid or web [1]. The analysis and the control design of these large networks (with thousands of nodes) may be hard problems. From this complexity follows the necessity of network reduction methods. One can distinguish two approaches to this problem. On the first hand, the approach coming from mathematics and computer science considers networks as static mathematical objects. The reduction is often treated with topological objectives, for instance focusing on the detection of community structures (group of densely connected nodes) [2]–[4]. In the other hand, the approach coming from system analysis and control theory considers the networks as representations of dynamical systems. The work within this approach aims to reduce networks by preserving a consistency in the dynamics or the control properties [5]–[8]. In our approach, several objectives are targeted. We propose a network reduction method with *dynamical* objectives [9]: preservation of the spectral centrality and preservation of the flow network property[1] and a *topological* objective: we want the reduced graph to be *scale-free* (while the initial graph is arbitrary).

Scale-free networks are ubiquitous in a wide range of fields like biological networks [10], social networks [11] and the world-wide web [12] among others. They are characterized by the presence of some nodes (the so-called hubs) with a large degree (number of connections) and a large number of nodes with small degree. Their degree distribution is a power law. A more precise definition is given further. These networks have first been introduced by Price in 1965 [13] but their study came into fashion in 1999 with the work of Barabasi and its collaborators [14]. The scale-freeness of a network implies some interesting properties [15]: robustness to failure, ultra-small world property[2], fitness for immunization strategy [16], network navigability [17], and interesting properties for control design [18]. In the context of our project, we are interested in the controlability gain brought by considering a scale-free network abstracting an arbitrary network. We present here a method allowing to reveal this underlying structure. The question of the advantage in term of controllability will be part of a future work.

We consider directed, weighted, strongly connected[3] graphs and we formulate the problem of graph reduction toward a scale-free graph while keeping a consistent behavior. The essential contribution of this paper is the introduction of a practical reduction method satisfying the following properties:

- The reduced graph has a degree distribution very close to a fixed scale-free degree distribution.
- The spectral centralities of the reduced network and the initial network are equal up to a projection.
- If the initial network is a flow network the output is ensured to also be a flow network.

II. Problem formulation

This section is devoted to introduce the problem of network reduction. First, we introduce some concepts about graph theory and we present the problem as a minimization problem. Then, we give specifications for each element of the problem in view to study a particular case in the following section.

A. Graph-theoretical preliminaries and notation

Consider a directed and weighted graph G, represented by the triple (A,V,E) where A is the adjacency matrix: $A_{i,j} \neq 0$ indicates that there is an edge between node i and node j. V is the set of vertices and E the set of edges. We may note $G = (\cdot, V, E)$ if only the structure of G (and not the weights) are relevant.

Consider the following notations:

- Γ_n is the set of directed, weighted and strongly connected networks with n nodes (we note Γ if the number of nodes is not known or relevant).

*Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, GIPSA-Lab, 38000 Grenoble, France
† CNRS, GIPSA-lab
{nicolas.martin, paolo.frasca, carlos.canudas-de-wit}@gipsa-lab.fr
www.scale-freeback.eu

[1] A network is said to be a flow network if for each node the amount of weights going in the node equals the amount of weights going out the node

[2] The average path length is proportional to the double-logarithm of the number of nodes.

[3] A graph G is said strongly connected, if for all pair (u,v) of nodes in G there is a path from u to v.

- Π_G is the degree distribution of G. It is a vector defined for all k between 1 and $\max\{\deg(v), v \in V\}\}$ by:

$$\Pi_G(k) = \frac{\text{card}\left(\{v \in V, \deg(v) = k\}\right)}{\text{card}(V)}$$

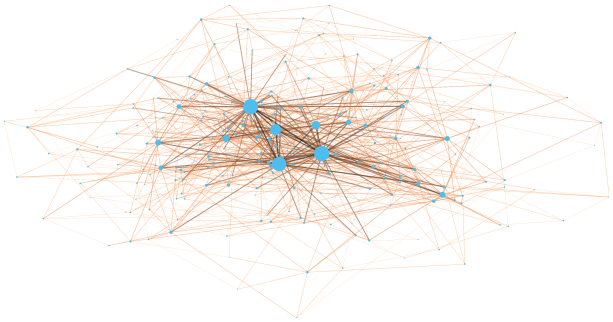
  where $\deg(v)$ is the degree[4] of the node v.
- $G_{|I}$, where $G = (A, V, E)$ and $I \subset V$, is the subgraph of G where only the vertices I and the edges connecting two vertices in I are considered.
- $|G|$ is the number of nodes in G.

We give now a definition of scale-free graphs.

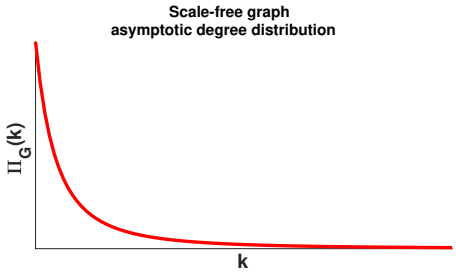
Definition 1 (Scale-free graph): A graph is called scale-free if its degree distribution is proportional to a power law:

$$\Pi_G(k) \propto k^{-\alpha}, \tag{1}$$

where $\alpha > 0$ is called the scale-free coefficient[5]. We denote α-scale-free distribution a scale-free distribution with a coefficient α. In practice we call scale-free graph any graph which is relatively close to a power law.



(a) Example of a scale-free graph. The size of the node is proportional to its degree. The main feature of scale-free graphs is the presence of nodes with high degree, called hubs. For instance, the Barabasi-Albert model allows to generate this type of graphs [14].



(b) Scale-free degree distribution is a power law.

Fig. 1. Scale-free graph

We choose to use partitioning as method of reduction as it preserves the connections of the network: two parts are connected if and only if there is a connection between two nodes in these parts. This property is meaningful when dealing with physical networks.

[4]As we consider directed graph the degree computed can be in-degree or out-degree. In the following, as the computations and the results remain the same for the two cases, the degree used is not precised.

[5]In most applications $\alpha \in [2;3]$

We need, now, to introduce some definitions relative to partitioning.

Definition 2 (Graph partition): A partition $S = \{S_1, ..., S_n\}$ of a connected graph $G = (\cdot, V, E)$, is a partition of the set of vertices V such that, for each part S_i, the subgraph $G_{|S_i}$ is weakly connected: $G_{|S_i}$ is composed by an unique component.

Definition 3 (Graphs coming out of a partition): Let $G_0 = (\cdot, V_0, E_0)$ be a graph, let S be a partition of this graph. We denote $G_1 = (\cdot, V_1, E_1)$ the graph whose nodes correspond to the parts of S and whose edges correspond to the connections between the parts. Explicitly we have:

$$\begin{aligned} &V_1 = \{1, .., |S|\} \\ &(i,j) \in E_1 \Leftrightarrow \exists (v,w) \in S_i \times S_j \,\text{s.t.}\, (v,w) \in E_0 \end{aligned} \tag{2}$$

We will denote this relation by $G_0 \succ G_1$ or by $G_0 \overset{S}{\succ} G_1$ when we want to emphasize the partition.

Note that, since this relation only determines the structure of the reduced graph and not its weights, there is an infinity of weighted graphs coming out of a given partition of a given graph .

Definition 4 (Merging): A merging is a particular partition where only two nodes are combined together.

Let $V = \{1, ..., n\}$ be the set of vertices, the merging of the vertices v and w is denoted by $S_{v,w}$ and:

$$\begin{aligned} S_{v,w} = &\{\{1\}, \{2\}, \ldots, \{v-1\}, \{v+1\}, \ldots \\ &\ldots, \{w-1\}, \{w+1\}, \ldots \{n\}, \{v,w\}\} \end{aligned}$$

Thanks to these definitions we can now introduce the main problem.

B. Graph reduction as an optimization problem

The general problem is to find a partition S of an initial large network $G_0 \in \Gamma$, such that the network G_1 coming out of the partition S of G_0 has a behavior coherent with the initial network and has a degree distribution close to a scale-free distribution while preserving physical properties. This problem can be formally stated as follows:

Given an initial graph $G_0 \in \Gamma \cap \Psi$, find a graph $\tilde{G}$, solution of the following minimization problem.

$$\begin{aligned} \min_{G} \quad & \mathrm{J}_{\mathrm{SF}_\alpha}(G) + \mathrm{J}_{\mathrm{sim}}(G, G_0), \\ \text{subject to} \quad & G_0 \succ G \\ & G \in \Psi \end{aligned} \tag{3}$$

where:

- $\mathrm{J}_{\mathrm{SF}_\alpha}$ is a scale-free cost function, indicating the $\alpha-$scale-freeness of the graph.
- $\mathrm{J}_{\mathrm{sim}}$ is a similarity cost function giving an indication of behavior consistency between the two graphs.
- Ψ is the set of graphs respecting the physical properties imposed.

In what follows we give specifications to the general problem (3) in order to study a particular instance of this problem.

C. Scale-free cost function

We introduce a preliminary definition before giving the definition of the cost function.

Definition 5 (Scale-free target distribution): The α-scale-free target distribution of size n is a vector noted $\Pi^{SF}_{\alpha,n}$ and defined as:

$$\Pi^{SF}_{\alpha,n} = \frac{1}{\sum_{i=1}^{k_{\max}} i^{\alpha}} \begin{pmatrix} 1^{\alpha} \\ 2^{\alpha} \\ \vdots \\ k_{\max}^{\alpha} \end{pmatrix} \tag{4}$$

where $k_{\max}$ is the largest degree for which the number of nodes having this degree is higher than 1 in a α-scale-free graph of size n:

$$k_{\max} = \arg\max_{k} \left\{ n \frac{k^{\alpha}}{\sum_{i=1}^{k_{\max}}} \geq 1 \right\} \tag{5}$$

This definition of $k_{\max}$ gives a reasonable value for the largest degree in a scale-free graph.

Definition 6 (Scale-free cost function): The $\alpha-$scale-free cost function of a graph G is defined as:

$$\mathrm{J}_{\mathrm{SF}_{\alpha}}(G) = \frac{\left\| \Pi_G - \Pi^{SF}_{\alpha,|G|} \right\|_2}{\left\| \Pi^{SF}_{\alpha,|G|} \right\|_2} \tag{6}$$

Note that Π_G and $\Pi^{SF}_{\alpha,|G|}$ are not of the same size in the case where $k_{\max} \neq \max\{\deg(v), v \in V\}\}$. In this case, zeros are added at the end of the smallest vector so that the sizes match.

D. Similarity cost function

The similarity cost function aims to define a notion of consistency that we want to preserve between the initial graph and the reduced graph. In our case we focus on a measure of the spectral centrality.

1) Spectral centrality: With the aim to compare the behavior of two graphs, we associate to every graph $G = (A,V,E)$ the following equation:

$$\begin{cases} x(t+1) = P^{\top} x(t) \\ x(0) = x_0 \end{cases} \tag{7}$$

where $x(t) \in \mathbb{R}^{|G|}$ is the state of the system (the value in each node) at time t and P is the adjacency matrix normalized by row, i.e. each element is divided by the sum of its row:

$$P_{i,j} = \frac{A_{i,j}}{\sum\limits_{k} A_{i,k}}, \tag{8}$$

This matrix P will be called normalized adjacency matrix of G. Equation (7) can be viewed as the evolution of a linear system or as the evolution of a Markovian chain where $P^{\top}$ is the matrix of transition.

As we have specified that G is strongly connected, then the associated Markov chain is irreducible and it always exists a stationary distribution $x^{\star}$ such that [19]:

$$x^{\star} = P^{\top} x^{\star} \tag{9}$$

Vector $x^{\star}$ is unique up to a multiplicative constant. We add condition $\|x^{\star}\|_1 = 1$ to ensure the uniqueness. $x^{\star}$ is the spectral centrality of the graph.

We call Φ the operator associating a graph in Γ to its spectral centrality:

$$\begin{aligned} \Phi : \quad & \Gamma_n \longrightarrow [0,1]^n \\ & G \longmapsto x^{\star}, \quad \text{s.t. } x^{\star} = P^{\top} x^{\star} \text{ and } \|x^{\star}\|_1 = 1 \end{aligned}$$

This vector gives a value at each node which is an indicator of the influence of this node on the graph. For instance in a random walk it would correspond to the percentage of time spent on each node. This definition of the spectral centrality is similar, for instance, to the PageRank definition.

2) Projection operator: To compare two spectral centralities in different dimensions we associate to any partition S an operator of projection σ_S defined as follows:

Definition 7 (Projection operator): Let $x \in \mathbb{R}^n$ and S a partition of the set $\{1,..,n\}$, we define the projection operator σ_S as:

$$\begin{aligned} \sigma_S : \mathbb{R}^n &\longrightarrow \mathbb{R}^{|S|} \\ x &\longmapsto y \ : \ \forall i, \ y_i = \sum_{j \in S_i} x_j \end{aligned} \tag{10}$$

The projection of the vector is then equal to the sum of the components within each cluster.

3) Formulation: With the previous definitions we can give the definition of the similarity cost function.

Definition 8 (Similarity cost function): Let $G_0 \in \Gamma$, G_1 and S such that $G_0 \overset{S}{\succ} G_1$. The similarity cost function is the relative distance between the spectral centrality[6] of G_1 and the projection of the spectral centrality of G_0.

$$\mathrm{J}_{\mathrm{sim}}(G_1, G_0) = \frac{\|\Phi(G_1) - \sigma_S(\Phi(G_0))\|_2}{\|\sigma_S(\Phi(G_0))\|_2} \tag{11}$$

E. Physical property: flow network

In the problem we treat, we assume that the initial network is a flow network [20] (also called transportation network) and we aim to preserve this property through the reduction. The preservation of this property has a strong physical meaning as some networks as electrical networks, water supply networks or generally every networks representing transportation are flow network by their nature. For instance, in electrical network this property is translated by the Kirchoff's circuit law. Thus, by preserving this property we ensure that the reduction method does not violate an intrinsic property of these networks. We give here a definition of the set of graph having this property:

$$\Psi = \left\{ G = (A,V,E), \forall k, \sum_i A_{ik} = \sum_j A_{kj} \right\}$$

Afterwards, we talk without distinction of flow network or flow graph.

[6]Note that if $G_0 \in \Gamma$ and $G_0 \succ G_1$ then $G_1 \in \Gamma$. It is then possible to compute the spectral centrality of G_1.

III. Analysis of the optimization problem

In this section, we will see how the partition can be chosen such that: i) the similarity cost function is null and ii) the output graph remains a flow graph.
These results will allow us to design an algorithm giving an approximation of the solution.

A. Cancelling similarity cost function

The similarity cost function translates the ability of the reduced graph to have a spectral centrality consistent with the spectral centrality of the initial graph. We see here that with a certain condition we can ensure a perfect consistency between the two graphs.

Theorem 1: Let $G_0 = (A_0, V_0, E_0) \in \Gamma_n$. For all edges $(v,w) \in E_0$ there is a graph G_1 coming out of the merging $S_{v,w}$ of G_0, $G_0 \overset{S_{v,w}}{\succ} G_1$, such that the similarity cost function between G_0 and G_1 is null which is $\mathrm{J}_{\mathrm{sim}}(G_1, G_0) = 0$.

In this case P_1, the normalized adjacency matrix of G_1, has the following form:

$$P_1 = FP_0H^\top,$$

where $F, H \in \mathbb{R}^{n-1\times n}$ are defined by:

$$F_{i,j} = \begin{cases} 1 & \text{if } i < n-1 \wedge S_i = \{j\} \\ \beta_v & \text{if } i = n-1 \wedge j = v \\ \beta_w & \text{if } i = n-1 \wedge j = w \\ 0 & \text{else} \end{cases} \tag{12}$$

$$H_{i,j} = \begin{cases} 1 & \text{if } j \in S_i \\ 0 & \text{else} \end{cases} \tag{13}$$

and $\beta_v = \frac{x_0^\star(v)}{x_0^\star(v)+x_0^\star(w)}$, $\beta_w = \frac{x_0^\star(w)}{x_0^\star(v)+x_0^\star(w)}$ where $x_0^\star$ is the spectral centrality of G_0.

The proof is not given here but it will be part of a future work.

B. Preservation of the flow network property

We see here that if the initial graph G_0 is a flow graph, it is possible to preserve this property through the reduction while keeping the result of the previous theorem true.

Theorem 2: Let $G_0 = (A_0, V_0, E_0) \in \Gamma_n \cap \Psi$. For all edges $(v,w) \in E_0$ there is a graph $G_1 = (A_1, E_1, V_1)$ coming out of the merging $S_{v,w}$ of G_0 such that G_1 is a flow graph and the similarity cost function between G_0 and G_1 is null, which is:

$$\begin{aligned} \forall\,(v,w) \in E_0, \quad \exists\, G_1,\ G_0 \overset{S_{v,w}}{\succ} G_1 \quad \text{s.t.} \\ \mathrm{J}_{\mathrm{sim}}(G_1, G_0) = 0 \ \wedge\ G_1 \in \Psi \end{aligned} \tag{14}$$

In this case A_1 has the following form:

$$A_1 = \kappa \mathrm{Diag}(l) FP_0H^\top, \tag{15}$$

where F, H are the merging matrices of $S_{v,w}$ defined in (12)-(13) (1), l is the left eigenvector of FP_0H associated with eigenvalue 1 and $\kappa \in \mathbb{R}^*$ is any arbitrary scalar.

As in the last section, we do not exhibit the proof here. We have seen in this section that for every flow graph G_0 and for every merging $S_{v,w}$, it exists a graph G_1 coming out of the merging $S_{v,w}$ of G_0 such that the spectral centralities of G_0 and G_1 are perfectly consistent and such that G_1 is a flow graph. Let's note that the graph G_1 is not unique as its adjacency matrix is defined up to a multiplicative constant κ. In the following, we choose κ such that the sum of all weights in G_1 is equal to the sum of all weights in G_0 which is:

$$\kappa = \frac{|A_0|_0}{|\mathrm{Diag}(l) FP_0H^\top|_0} \tag{16}$$

where $|\bullet|_0$ is defined as: $|A|_0 = \sum_{i,j} A_{i,j}$ for all matrices A. By this way, the reduced graph G_1 is uniquely defined and we denote it by $G_1 = G_0^{(v,w)}$

IV. Algorithm

In this section we will see how the results of the previous section can be used to design an effective algorithm to provide an approximate solution of the problem (3).

A. Algorithm description and comments

The results of section III suggest that we can merge edges recursively to obtain a reduced graph perfectly consistent with the first one and remaining a flow graph. Thus we propose an algorithm of reduction *step-by-step*, and at each step we look for the edge whose merging minimizes the scale-free cost function $\mathrm{J}_{\mathrm{SF}_\alpha}$. Actually, as the number of linked node can be tremendous, we only look for the best edge within a random subset of edges of size n_{rand} and we discuss then the influence of this random selection.
A partial description of the algorithm is presented in Algorithm 1: Therein E_k is the set of edges of the graph at step k. The inputs are the initial graph $G_{init} \in \Gamma_n \cap \Psi$, a scale-free coefficient α and an integer n_{rand}.
First, we initialize the first graph G_0 (line 1) and a random subset $\bar{E}$ of n_{rand} edges is drawn (line 4). For each edge e, the graph $G_k^{\overset{e}{\succ}}$ is build according to (15) and (16) and its scale-freeness $n_{SF}(e)$ is computed (line 6). Among the n_{rand} edges the best one is the one whose merging minimizes the scale-free cost function (line 8). This edge is merged to build the new graph (line 9).

Algorithm 1 Merge to scale-free

1: $G_0 = G_{init}$;
2: $k = 0$;
3: **while** $\neg stop$ **do**
4: $\quad \bar{E} = rand(E_k, n_{rand})$;
5: $\quad$ **for** $e \in \bar{E}$ **do**
6: $\quad\quad n_{SF}(e) = \mathrm{J}_{\mathrm{SF}_\alpha}(G_k^{\overset{e}{\succ}})$
7: $\quad$ **end for**
8: $\quad e_{best} = \mathrm{argmin}_e\, n_{SF}(e)$
9: $\quad G_{k+1} = G_k^{\overset{e_{best}}{\succ}}$
10: $\quad k$++;
11: **end while**

This algorithm does not provide the global minimum of (3) but an approximation of it. The stopping criterion *stop* is not

discussed here, it may naturally be defined as the step where it is no more possible to find a merging increasing the scale-freeness of the graph or as a fixed number of iterations. The influence of the random process and the size of the subset of edges is discussed in the last section.

B. Algorithm complexity

Proposition 1: Consider the execution of Algorithm 1 on an initial graph G_{init} with N_v nodes, and with n_{rand} the size of the random subset of edges $\bar{E}$. Thus, the complexity in term of numbers of operations is $O(n_{rand}N_v^3)$.

This complexity is polynomial with respect to the size of the initial graph (its number of nodes). Since the naive way to find a partition of a graph by testing all possibilities has an exponential complexity, a polynomial algorithm is significantly faster.
The growth of the computation time is linear in function of n_{rand}. We will investigate in the next section, the influence of this parameter on the performance of the algorithm.

V. Simulations

In this section, an applications is presented: the algorithm is applied on an academic case : a Manhattan-like network which is introduced later. Moreover a study of the influence of the random process of the algorithm, and in particular of the size of the random subset, is presented.

A. Manhattan-like network

This network mimicking the urban network of certain city (in particular Manhattan), consists in a basic grid. In our case, we add some irregularities by removing nodes, adding diagonal shortcuts and adding one-way road. An example of such a network is in Fig.2(a). We randomly add weights on the initial graph while ensuring it to be a flow graph. It is interesting to run our algorithm over this type of network because: i) it is a good representation of some physical networks as urban traffic networks [21], ii) it is far from a scale-free network so it can show the ability of the algorithm to get close to a scale-free distribution and iii) it is easy to build this type of network even with arbitrarily large size making the results presented easily reproducible. The tuning of the different parameters of the algorithm is presented in Table V-A.

Grid size	$\|G\|$	α_{SF}	n_{rand}	Degree
65×65	3824	-2	10	in

The output of the simulation is presented in Figure 2. It appears clearly that the algorithm drives the graph very close to the scale-free distribution targeted. The run time for is simulation is about 180 seconds.

B. Influence of the size of the set of edges

In the previous simulation, n_{rand}, the size of the random subset of edges was equal to 10 and it appears that it is sufficient to obtain a scale-free graph. However we can wonder, how fast the algorithm drives the graph towards a scale-free distribution in function of this parameter. The evolution of the value of the scale-free cost function at each step of the algorithm for different values of n_{rand} is represented in Fig.3.

Let first remark that in every cases the error decreases before going up, even in the case where $n_{rand} = 1$ which correspond to the case where the edge to merge is selected randomly. It shows that a graph (at least this type of graph) naturally tends towards a scale-free structure when it is recursively merged. The figure shows also that there is no significant advantage to have a large value of n_{rand}. Thus, at each step of the algorithm we may select only a small random subset of edges (≈ 15) without reducing the performance of the algorithm in view to reduce the computation time.

VI. Conclusion

In this paper, we have formulated a problem of graph reduction toward a scale-free distribution as an optimization problem where we seek to optimize the scale-freeness of the graph and the similarity (in term of spectral centrality) while ensuring it to remain a flow graph. It appears that in the case of a particular partition, a merging, it is possible to have a perfect consistency with the initial graph and to preserve the physical property. Thus, we design an algorithm which takes advantage of these results and allows to find a graph with an arbitrary scale-free distribution, and with these two properties in a reasonable computation time. Finally a simulation on a large graph is presented, and shows the efficiency of the algorithm.
It is useful to remark that the reduction algorithm is applied to drive an arbitrary initial graph towards a scale-free structure, but it could be used to drive it towards any desired structure while preserving the similarity and the flow graph property.

Acknowledgements

The research leading to these results has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement N 694209)

References

[1] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.

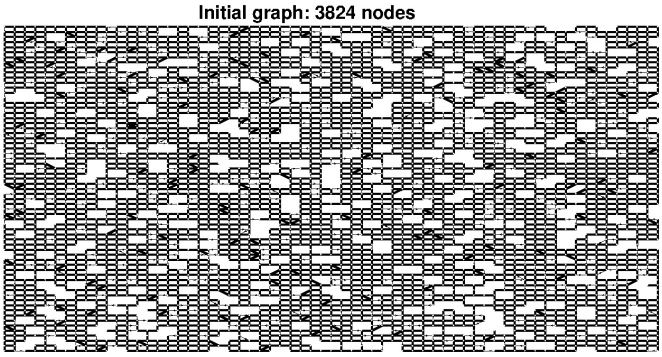
[2] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006. [Online]. Available: http://www.pnas.org/content/103/23/8577.abstract

[3] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Phys. Rev. E*, vol. 72, p. 027104, Aug 2005. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.72.027104

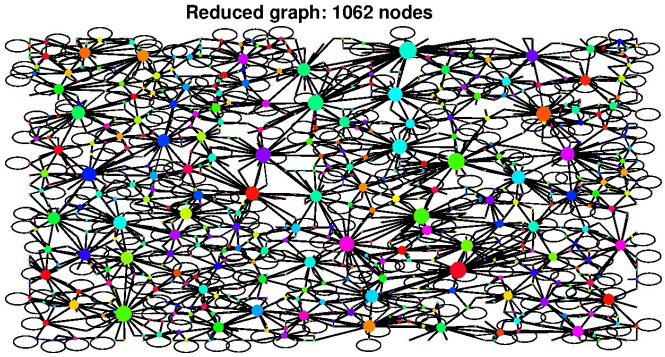
[4] L. Danon, A. Daz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 09, p. P09008, 2005. [Online]. Available: http://stacks.iop.org/1742-5468/2005/i=09/a=P09008

[5] T. Ishizaki, K. Kashima, A. Girard, J.-i. Imura, L. Chen, and K. Aihara, "Clustered model reduction of positive directed networks," *Automatica*, vol. 59, pp. 238–247, 2015.

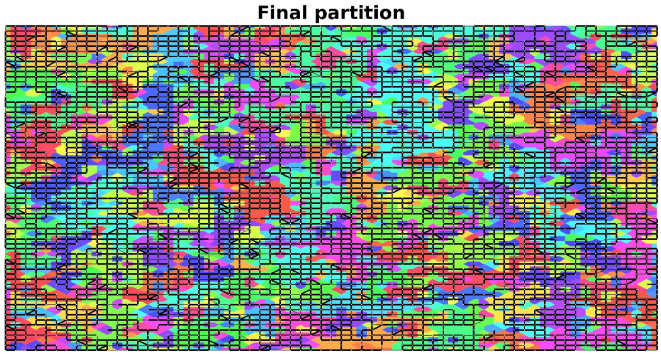
[6] N. Monshizadeh and A. van der Schaft, "Structure-preserving model reduction of physical network systems by clustering," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 4434–4440.



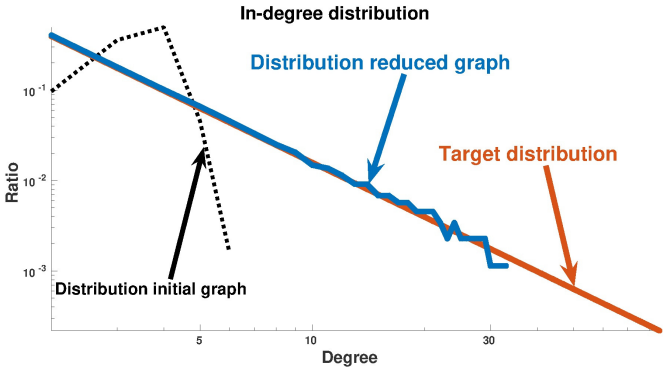
(a) The initial graph. Double-way links are drawn more thick than one-way ones.



(b) The reduced graph issued from the partition. The size of the nodes represents the degree of each nodes, and the color matches with the color of the partition.



(c) Final partition of the initial graph. The color matches with the colors of the nodes in the reduced graph.



(d) In-degree distribution of the initial graph, of the final graph and the target distribution in log-log representation.

Fig. 2. Output of the Manhattan-like grid simulation.

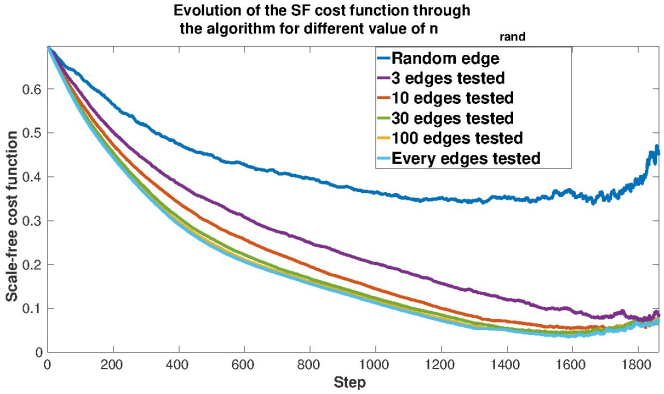


Fig. 3. Comparison of the evolution of the scale-free cost function through the algorithm for different values of n_{rand}.

[7] F. Dorfler and F. Bullo, "Kron reduction of graphs with applications to electrical networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 1, pp. 150–163, 2013.

[8] X. Cheng, Y. Kawano, and J. M. Scherpen, "Graph structure-preserving model reduction of linear network systems," in *Control Conference (ECC), 2016 European*. IEEE, 2016, pp. 1970–1975.

[9] N. Perra and S. Fortunato, "Spectral centrality measures in complex networks," *Physical Review E*, vol. 78, no. 3, p. 036107, 2008.

[10] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási, "The large-scale organization of metabolic networks," *Nature*, vol. 407, no. 6804, pp. 651–654, 2000.

[11] F. Liljeros, C. R. Edling, L. A. N. Amaral, H. E. Stanley, and Y. Åberg, "The web of human sexual contacts," *Nature*, vol. 411, no. 6840, pp. 907–908, 2001.

[12] A.-L. Barabási, R. Albert, and H. Jeong, "Scale-free characteristics of random networks: the topology of the world-wide web," *Physica A: statistical mechanics and its applications*, vol. 281, no. 1, pp. 69–77, 2000.

[13] D. Price, "Statistical studies of networks of scientific papers," in *Statistical Association Methods for Mechanized Documentation: Symposium Proceedings*, vol. 269. US Government Printing Office, 1965, p. 187.

[14] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.

[15] M. E. Newman, "The structure and function of complex networks," *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.

[16] R. Pastor-Satorras and A. Vespignani, "Immunization of complex networks," *Phys. Rev. E*, vol. 65, p. 036104, Feb 2002. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.65.036104

[17] M. Boguná and D. Krioukov, "Navigating ultrasmall worlds in ultra-short time," *Physical review letters*, vol. 102, no. 5, p. 058701, 2009.

[18] J. C. Nacher and T. Akutsu, "Dominating scale-free networks with variable scaling exponent: heterogeneous networks are not difficult to control," *New Journal of Physics*, vol. 14, no. 7, p. 073005, 2012.

[19] J. Hopcroft and R. Kannan, "Foundations of data science," 2014, ch. 5.

[20] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows*. Elsevier, 2014.

[21] F. Travostino, J. Mambretti, and G. Karmous-Edwards, *Grid networks: enabling grids with advanced communication technology*. John Wiley & Sons, 2006.