Large-scale network reduction towards scale-free structure

Nicolas Martin, Paolo Frasca, Carlos Canudas de Wit

Large-scale network reduction towards scale-free structure

Nicolas Martin, Paolo Frasca, *Member, IEEE,* Carlos Canudas-de-Wit, *Fellow, IEEE*

Abstract—This paper deals with a particular problem of graph reduction. The reduced graph is aimed to have a particular structure, namely to be scale-free. To this end, we define a metric to measure the scale-freeness by measuring the difference between the degree distribution and the scale-free degree distribution. The reduction is made under constraints to preserve consistency with the initial graph. In particular, the reduced graph preserves the eigenvector centrality of the initial graph. We study the optimization problem and, based on the gained insights, we derive an algorithm allowing to find an approximate solution. We also show that, if the initial network is a flow network, it is possible to design the algorithm such that the output remains a flow network. Experimental results are then presented to optimally choose the parameters of the algorithm suggesting that, by tuning a parameter, it is possible to speed up the algorithm with a comparable efficiency. Finally, the algorithm is applied to an example of large physical network: the Grenoble urban traffic network.

Index Terms—Network theory, Network reduction, Scale-free network, Flow network

✦

1 INTRODUCTION

Many physical systems can be represented as networks and physical phenomena as processes over networks. Thus, 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 (with thousands of nodes) are common in several fields like transportation, power grid or biology among others. See [1] for a collection of such large networks. The analysis and control design of these large networks may remain out of our capacities. This complexity motivates network reduction methods also known as coarse-graining or summarization methods. Works on the reduction of graphs are profuse (see [2] for an extensive survey) and their objectives are different depending on the application. However, the different purposes of graph reduction methods have the following form: cutting the complexity (e.g. volume of data, redundancy, visualization) of a graph while preserving some properties (e.g. topological, dynamical, patterns). The techniques used differ but essentially there are four categories: partitioning (merging nodes in super-node and/or edges in super-edges), compression (exploiting redundancy in the patterns of the graph) and simplification (removing unimportant nodes and/or edges). This taxonomy, as well as the approaches, may differ according to the fields of study. For instance, the approach coming from the model reduction community considers the networks as representations of dynamical systems and aims to reduce networks by preserving a consistency in the dynamics or the control properties [3], [4], [5], [6]. In this work, we use the term *graph reduction* by analogy with the model reduction terminology.

The approach in our work combines several objectives: starting with a directed, weighted and strongly connected[1] initial graph, we aim to reduce it into a graph preserving some relevant properties such as connectivity and eigenvector centrality. See [7] for a similar approach that focuses on the eigenvector centrality preservation. Moreover, in the aim to provide useful properties to the reduced graph, we want it to be *scale-free*, whereas the initial graph is not necessarily such.

Scale-free networks are ubiquitous in a wide range of fields like biological networks [10], social networks [11], the internet network [12] or the world-wide web [13] among others. They are characterized by the presence of few nodes (the so-called hubs) with a large degree (number of connections) and a large number of nodes with small degree: by definition, their degree distribution is a power law. See Fig. 1(b) for an illustration. This type of structure appears naturally in certain growing networks due to preferential attachment processes [14]. These networks have first been emphasized by Price in 1965 [15], but their study came into fashion in 1999 with the work of Barabasi and his collaborators [9]. Interesting properties emerge from the definition of scale-free networks [16], such as small distances between nodes [17], robustness to random failures, easiness to disconnect, hyperbolic space embedding [18]. These properties allow to gain efficiency in some applications such as network navigation [19], vaccination in epidemiology [20], or control design [21]. In order to reap these advantages, we wonder in this paper how is it possible to find a network having a scale-free structure and abstracting an arbitrary network. However, we do not question here in which measure the computations made on the reduced network may be meaningful for the initial network. We focus only on the design of a reduction method whose output is a scale-free network.

The essential contribution of this work is the introduction of a graph partition method whose output graph has a degree distribution close to a desired scale-free degree distribution and such that the eigenvector centrality is preserved. Moreover, if the algorithm is applied on a flow network[2], the output will also be a flow network. The sum of all weights in the network is also preserved

- *N. Martin and P. Frasca are with Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, GIPSA-Lab, 38000 Grenoble, France*
  E-mail: {nicolas.martin;paolo.frasca}@gipsa-lab.fr
- *C. Canudas-de-Wit is with CNRS, GIPSA-lab*
  E-mail: carlos.canudas-de-wit@gipsa-lab.fr

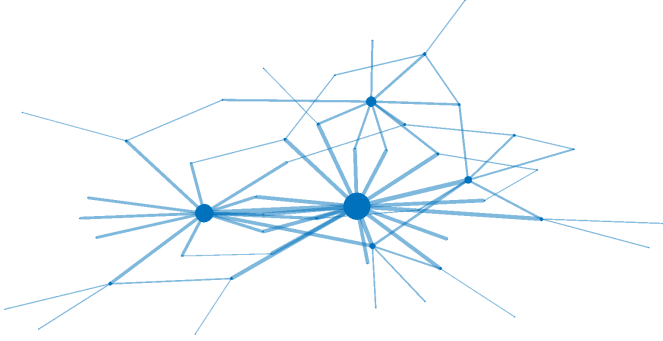1. A graph G is said strongly connected, if for all pair of nodes (u,v) in G there is a path from u to v.

2. A weighted network is said to be a flow network if for each node the amount of weights going in equals the amount of weights going out.
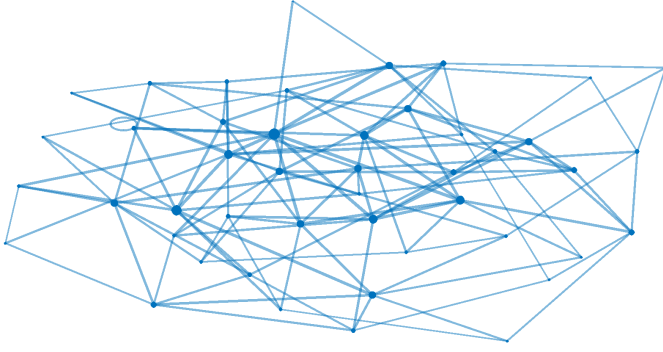
(a) Graph with homogeneous distribution. For instance, the Erdos-Renyi model [8] allows to generate such graphs.

(b) Scale-free graph. The distribution of degrees is heterogeneous. For instance, the Barabasi-Albert model [9] allows to generate scale-free graphs.
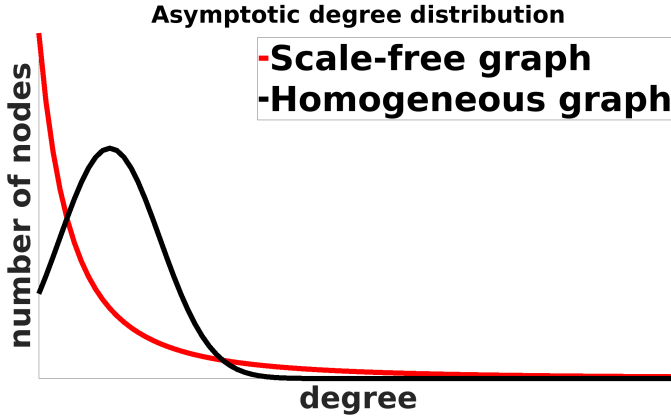
(c) Theoretical degree distributions: the homogeneous graph distribution is a Poisson law while the scale-free graph distribution is a power law. These continuous degree distributions are valid in the infinite graphs limit, hence the name *asymptotic*.

Fig. 1. Comparison between two types of graph: homogeneously distributed graph and scale-free graph.

through the algorithm. These three properties are particularly relevant in applications involving flows such as traffic networks, power networks or packet flow networks. The performance of the algorithm and the tuning of the parameters are studied on synthetic networks (Manhattan-like grids) and on a real network: the Grenoble urban traffic network. This large network (almost 20000 nodes) is an example of flow network as the number of cars is conserved through the nodes. The algorithm is applied on this network highlighting the efficiency of the method even with a large network.

2 PROBLEM FORMULATION

This section is devoted to introduce formally the problem of network reduction that we have introduced above. First, we introduce some preliminaries and the physical model used; then, we present the problem as a minimization problem. Finally, we give a formal definition for each element of the minimization problem.

2.1 Preliminaries: scale-free graph and partitions

In this section we give, first, some graph-theoretical definitions allowing to define scale-free graphs. Then, we define what we call a graph partition and some related notions.

Consider a directed and weighted graph G, represented by the triple (A, V, E) where A is the adjacency matrix, whose non-zeros values indicate the edges and their weights: $A_{i,j} = w > 0$ means that there is an edge $i \longrightarrow j$ and that a weight w is assigned to this edge. V is the set of vertices and E the set of edges. We may denote $G = (\cdot, V, E)$ if only the structure of G (and not the weights) is relevant. We denote by Γ_n the set of directed, weighted and strongly connected graphs with n nodes (or Γ if the number of nodes is not known or relevant). We also denote by Π_G the degree distribution of G, i.e. $\Pi_G(k)$ gives the number of nodes having degree k:

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

where $\deg(v)$ is the degree[3] of the node v.

We give now a definition of scale-free graph:

Definition 1 (Scale-free graph). A graph G is said to be scale-free if its degree distribution is proportional to a power law, which is:

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

where $\alpha > 0$ is called the scale-free coefficient (in most applications $\alpha \in [2; 3]$). We denote α-scale-free distribution a scale-free distribution with a coefficient α. In practice we call scale-free graph any graph whose degree distribution is *relatively* close to a power law.

A scale-free graph having an *heterogeneous* degree distribution, it is natural to contrast it with graphs having an *homogeneous* degree distribution. Figure 1 illustrates the differences between these two structures of graph.

In this work we will focus on graph partitioning. We choose to use this method as it preserves the structure of the network, which is: two parts of the partition are connected if and only if there is a connection between two nodes belonging to these parts. This property is meaningful when dealing with physical networks. We introduce now some notions about partitioning:

Definition 2 (Graph partition). A partition $S = \{S_1, S_2, ..., S_n\}$ of a graph $G = (\cdot, V, E)$, is a partition of the set of vertices V such that, for each part S_i, the subgraph $G_{|S_i}$ (in which only the nodes in S_i are considered) is weakly connected: each pair of nodes in $G_{|S_i}$ is linked regardless of the direction of the edges.

Definition 3 (Graph 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 obtained by replacing parts of S by super-nodes, and by linking super-nodes corresponding

---

3. as we consider directed graphs, the degree can be either in-degree or out-degree. As the computations and the results remain the same for both cases, the degree used is not precised

2

to parts which contains linked nodes. The graph G_1 is then said to be coming out of the partition S of G_0. Explicitly we have:

$$V_1 = \{1, \ldots, |S|\}$$
$$(i,j) \in E_1 \Leftrightarrow (S_i \times S_j) \bigcap E_0 \neq \emptyset \tag{2}$$

We will denote this relation by $G_0 \succ G_1$ or by $G_0 \overset{S}{\succ} G_1$ to emphasize the partition. Let us remark 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 the partition S of G_0.

Definition 4 (Merging). A merging is a particular partition where only two nodes are combined together. Let $V = \{1, \ldots, n\}$ be the set of vertices, the merging of the vertices v and w is denoted by $S_{v,w}$ and:

$$S_{v,w} = \{\{1\}, \{2\}, \ldots, \{v-1\}, \{v+1\}, \ldots$$
$$\ldots, \{w-1\}, \{w+1\}, \ldots \{n\}, \{v,w\}\}$$

2.2 Physical model

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

$$\begin{cases} x(k+1) = P^\top x(k) \\ x(0) = x_0 \end{cases} \tag{3}$$

where P is is the adjacency matrix normalized by rows, i.e. each element is divided by the sum of its row:

$$P_{i,j} = \frac{A_{i,j}}{\sum_k A_{i,k}}, \tag{4}$$

and $P^\top$ is the transpose of P. This normalization divides each weight coming out of a node i by the amount of weights coming out of i. Matrix P can be viewed as the transition matrix of a Markov chain associated to the graph. As we have specified that G is strongly connected, then the associated Markov chain is irreducible and it always exists [22] a stationary distribution $x^\star$:

$$x^\star = P^\top x^\star \tag{5}$$

We add the condition $\|x^\star\|_1 = 1$ to ensure the uniqueness [22]. The stationary distribution $x^\star$ is also known in graph theory as the eigenvector centrality.

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

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

This centrality gives a value at each node corresponding to the relative importance of the node in the graph. In particular, this value corresponds to the ratio of time spent by a random walker on each node of the graph. The PageRank algorithm used by Google to rank websites by their importance is based on this eigenvector centrality applied to the web network [23].

Let us note that strong connectivity is a necessary condition for computing the eigenvector centrality and so to develop the rest of our analysis. Traffic networks are always strongly connected (one can picture that it is always possible to reach any point from any point in a traffic network). However, in other cases it can be challenging to determine the strong connectivity of a network.

Some works bring efficient methods to answer this issue [24], [25]. To compare a graph G_0 with a graph G_1 issued from the partition S of G_0 based on their eigenvector centrality, we associate to any partition S an operator of projection σ_S. The projection corresponds to the sum of the components within each part of the partition. We give here a precise definition:

Definition 5 (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:

$$\sigma_S : \mathbb{R}^n \longrightarrow \mathbb{R}^{|S|}$$
$$x \longmapsto y : \forall i, \, y_i = \sum_{j \in S_i} x_j \tag{6}$$

This operator can be written as a matrix operation: $\sigma_S(x) = K_s x$ where:

$$(K_s)_{i,j} = \begin{cases} 1 & \text{if } j \in S_i \\ 0 & \text{else} \end{cases} \tag{7}$$

Now, we can compare the behavior of two graphs G_0 and G_1 such that $G_0 \overset{S}{\succ} G_1$ by looking at the vector Δ_{G_0,G_1} defined as:

$$\Delta_{G_0,G_1} = \Phi(G_1) - \sigma_S(\Phi(G_0)) \tag{8}$$

The $i-th$ entry of Δ_{G_0,G_1} represents how close the centrality of the nodes i in G_1 is to the sum of the centralities in the subset S_i in G_0. We denote $\delta_{G_0,G_1} = \|\Delta_{G_0,G_1}\|_2$ the *eigenvector distance* between the graphs G_0 and G_1.

2.3 Graph reduction as an optimization problem

The 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 degree distribution that is closest to a given scale-free distribution. We also want that G_1 preserves some physical properties of G_0 and that the eigenvector distance between G_0 and G_1 is null. 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.

$$\tilde{G} = \min_G \mathrm{J}_{\mathrm{SF}_\alpha}(G), \quad \text{subject to } G_0 \succ G$$
$$G \in \Psi \tag{9}$$
$$\delta_{G_0,G} = 0$$

where:

- $\mathrm{J}_{\mathrm{SF}_\alpha}$ is a scale-free cost function, indicating the α-scale-freeness of the graph.
- Ψ is the set of graphs respecting the physical properties imposed.

The constraint on the eigenvector centrality has been chosen because is shows the capacity of our method to deal with structural constraints and because the preservation of this centrality can be essential for some applications. For example, to investigate the most visited areas, the reduced network should have the same centrality as the initial one.

2.4 Scale-free cost function

We present here an intuitive scale-free cost function.

Definition 6 (Scale-free target distribution). The α-scale-free target distribution of size n is denoted $\Pi_{\alpha,n}^{SF}$ and is defined as:

$$\Pi_{\alpha,n}^{SF} = \frac{1}{\sum_{i=1}^{k_{cut}} i^\alpha} \begin{pmatrix} 1^\alpha \\ 2^\alpha \\ \vdots \\ k_{cut}^\alpha \end{pmatrix} \tag{10}$$

3

where k_{cut} is a cut-off calculated as the highest degree for which the number of nodes having this degree is higher than 1 in a α-scale-free graph of size n:

$$k_{cut} = \arg\max_{k} \left\{ \frac{k^{\alpha}}{\sum_{i=1}^{k} i^{\alpha}} \geq \frac{1}{n} \right\} \qquad (11)$$

We define now the scale-free cost function:

Definition 7 (Scale-free cost function). For any graph G, we define:

$$J_{SF_{\alpha}}(G) = \frac{\left\| \Pi_G - \Pi_{\alpha,|G|}^{SF} \right\|_2}{\left\| \Pi_{\alpha,|G|}^{SF} \right\|_2}, \qquad (12)$$

where $\Pi_{\alpha,|G|}^{SF}$ is the scale-free target distribution defined as in (10) and $|G|$ is the number of nodes in G.

Let us note that Π_G and $\Pi_{\alpha,|G|}^{SF}$ are not necessarily of the same size. In this case, zeros are added at the end of the smallest vector so that the sizes match.

Remark 1. It is possible to impose a scale-free distribution both on the in-degree and the out-degree with two different coefficients α_{in} and α_{out}. In this case the scale-free cost function is:

$$J_{SF_{\alpha}}(G) = \mu_{in} \left\| \Pi_{in} - \Pi_{\alpha_{in},|G|}^{SF} \right\|_2 / \left\| \Pi_{\alpha_{in},|G|}^{SF} \right\|_2 + \mu_{out} \left\| \Pi_{out} - \Pi_{\alpha_{out},|G|}^{SF} \right\|_2 / \left\| \Pi_{\alpha_{out},|G|}^{SF} \right\|_2 \qquad (13)$$

where $\mu_{in}, \mu_{out} > 0$ are two coefficients allowing to adjust the relative importance of a distribution with respect to the other one.

Remark 2. The mathematical results presented further are independent of the scale-free cost function chosen. Hence, any definition of the cost function could be used without questioning the validity of the results.

2.5 Physical property: flow network

In the the problem we treat, we assume that the initial network is a flow network [26] (also called transportation network) and we aim to preserve this property through the reduction. The preservation of this property has a strong physical meaning because some networks as electrical networks, water supply networks or generally every network representing transportation are flow networks by their nature. For instance, in electrical network this property corresponds to 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 graphs having this property:

$$\Psi = \left\{ G = (A, V, E), \forall k, \sum_i A_{ik} = \sum_j A_{kj} \right\} \qquad (14)$$

We also define a flow matrix as a matrix representing a flow network which is a matrix A such that $\forall k, \sum_i A_{ik} = \sum_j A_{kj}$.

We gave in this section specifications to problem (9). This particular case of the problem can be reformulated as follows. Given an initial network and a scale-free coefficient, we look for a reduced network that comes out of a partition of the initial network with the following three features: it minimises the scale-free cost function defined in (12), it is a flow network as defined in (14), and the eigenvector distance (8) between the initial and the reduced network is null.

3 ANALYSIS OF THE OPTIMIZATION PROBLEM

In this section, we will see how the weights of the reduced graph can be chosen such that: i) the eigenvector distance is null, ii) the reduced graph remains a flow graph and iii) the sum of all weights in the graph is preserved. These results will allow us to design an algorithm giving an approximation of the solution.

3.1 Cancelling eigenvector distance

We see here that with a certain choice of the weights of the reduced graph we can ensure a perfect consistency, in terms of eigenvector centrality, 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 choice of the weights of the graph G_1 coming out of the merging $S_{v,w}$ of G_0, such that the eigenvector distance between G_0 and G_1 is null, which is $\delta_{G_0,G_1} = 0$.

To do so, it is sufficient to take P_1, the normalized adjacency matrix of G_1 as:

$$P_1 = F P_0 H^{\top}, \qquad (15)$$

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

$$F_{i,j} = \begin{cases} 1 & \text{if } i < n-1 \wedge S_i = \{j\} \\ \beta_v & \text{if } i = n-1 \wedge j = v \\ \beta_w & \text{if } i = n-1 \wedge j = w \\ 0 & \text{else} \end{cases} \qquad (16)$$

$$H_{i,j} = \begin{cases} 1 & \text{if } j \in S_i \\ 0 & \text{else} \end{cases} \qquad (17)$$

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 eigenvector centrality of G_0.

Proof: We have to show three points: the matrix P_1 i) has a structure compatible with a graph coming out of the merging $S_{v,w}$ of the graph G_0, ii) is normalized and iii) has an eigenvector centrality equal to the projection of the eigenvector centrality of P_0.

i) By definition of a graph coming out of a partition (see (2)), there is an edge $i \longrightarrow j$ in the reduced graph if and only if there exists an edge $l \longrightarrow k$ in G_0 such that $l \in S_i$ and $k \in S_j$. We want to show that the position of the non-zeros values in P_1 respects this structure. We have:

$$P_{1_{i,j}} = \sum_{k=1}^{n} \sum_{l=1}^{n} F_{i,l} P_{0l,k} H_{j,k}$$

and we know that $F_{i,l} \neq 0 \Leftrightarrow l \in S_i$ and that $H_{j,k} \neq 0 \Leftrightarrow k \in S_j$, hence:

$$P_{1_{i,j}} = \sum_{k \in S_j} \sum_{l \in S_i} F_{i,l} P_{0l,k} H_{j,k}$$

It comes out that:

$$P_{1_{i,j}} \neq 0 \iff \exists (l,k) \in S_i \times S_j \quad \text{s.t.} \quad P_{0l,k} \neq 0 \qquad (18)$$

and this is what we were supposed to show.

ii) We want to show that the matrix P_1 is normalized in the sense that $\forall i \in [1,..,n-1], \sum_j P_{1_{i,j}} = 1$. We have:

$$\sum_{j=1}^{n-1} P_{1_{i,j}} = \sum_{j=1}^{n-1} \sum_{k \in S_j} \sum_{l \in S_i} F_{i,l} P_{0l,k} H_{j,k}$$

$$= \sum_{k \in S_{n-1}} \sum_{l \in S_i} F_{i,l} P_{0l,k} H_{n-1,k} + \sum_{j=1}^{n-2} \sum_{k \in S_j} \sum_{l \in S_i} F_{i,l} P_{0l,k} H_{j,k}$$

We note that for $j < n-1$ there exists an unique $k \in S_j$ and so an unique k such that $H_{j,k} = 1$. We denote it by k_j. We have then:

$$\sum_{j=1}^{n-1} P_{1i,j} = \sum_{k \in \{v,w\}} \sum_{l \in S_i} F_{i,l} P_{0l,k} + \sum_{j=1}^{n-2} \sum_{l \in S_i} F_{i,l} P_{0l,k_j}$$

We know that when j covers the set $\{1,...,n-2\}$ then k_j covers the set $\{1,..,n\} \setminus \{v,w\}$ and then:

$$\sum_{j=1}^{n-1} P_{1i,j} = \sum_{k \in \{v,w\}} \sum_{l \in S_i} F_{i,l} P_{0l,k} + \sum_{\substack{k=1 \\ k \notin \{v,w\}}}^{n} \sum_{l \in S_i} F_{i,l} P_{0l,k}$$

$$= \sum_{k=1}^{n} \sum_{l \in S_i} F_{i,l} P_{0l,k}$$

$$= \sum_{l \in S_i} F_{i,l} \sum_{k=1}^{n} P_{0l,k}$$

$$= \sum_{l \in S_i} F_{i,l} = 1 \qquad \text{(because } P_0 \text{ is normalized)}$$

Then, P_1 is the normalized adjacency matrix of a reduced graph G_1 coming out of the merging $S_{v,w}$ of G_0.

iii) Let $x_1^\star$ be the eigenvector centrality of G_1. We want to show $\sigma(x_0^\star) = x_1^\star$. We remark first that the definition of H is the same as the definition of K_s in (7) and so we have $\sigma(x) = Hx$ for all x. Moreover we have:

$$H^\top F = I_{n-1} + (\alpha_v - 1)e_{v,v}^{n-1} + \alpha_v e_{w,v}^{n-1} + (\alpha_w - 1)e_{w,w}^{n-1} + \alpha_w e_{v,w}^{n-1}$$

where $e_{i,j}^n$ is the square matrix of size n whose only non-zero entry is a 1 in (i, j), and I_n is the identity matrix of size n. It follows that

$$(x_0^{\star\top} H^\top F)_i = x_0^\star(i) \quad \forall i \notin \{v,w\}$$

$$(x_0^{\star\top} H^\top F)_v = x_0^\star(v) \alpha_v + x_0^\star(w) \alpha_v$$

$$= (x_0^\star(v) + x_0^\star(w)) \frac{x_0^\star(v)}{x_0^\star(v) + x_0^\star(v)} = x_0^\star(v)$$

$$(x_0^{\star\top} H^\top F)_w = x_0^\star(v) \alpha_w + x_0^\star(w) \alpha_w$$

$$= (x_0^\star(v) + x_0^\star(w)) \frac{x_0^\star(w)}{x_0^\star(v) + x_0^\star(v)} = x_0^\star(w)$$

Hence $x_0^{\star\top} = x_0^{\star\top} H^\top F$, and then:

$$x_0^{\star\top} P_0 H^\top = x_0^{\star\top} H^\top F P_0 H^\top$$

$$x_0^{\star\top} H^\top = x_0^{\star\top} H^\top P_1$$

$$H x_0^\star = P_1^\top H x_0^\star$$

$$\sigma(x_0^\star) = P_1^\top \sigma(x_0^\star)$$

Then, $\sigma(x_0^\star)$ is the eigenvector centrality of G_1. And by uniqueness of the eigenvector centrality $\sigma(x_0^\star) = x_1^\star$. $\qquad \square$

This result concerns only merging, which is a particular partition, but it can be extended to any partition. For simplicity, and since it is sufficient for the following, we have only shown the case of the merging.

3.2 Preservation of the flow graph property

The flow network property defined in Section 2.5 is another physical property that we consider. We see here that if the initial graph is a flow graph we can ensure that the reduced graph is a flow graph too.

Theorem 2. Let P be a normalized matrix as defined in (4), then it exists a diagonal matrix X such that XP is a flow matrix.

Namely, $X = \kappa \text{Diag}(x^\star)$, where $\kappa \neq 0$ and $x^\star$ is the eigenvector centrality associated with P.

Proof: Let $x^\star$ be the eigenvector centrality of P, i.e. $x^\star P = x^\star$ and $X = \kappa \text{Diag}(x^\star)$ for any $\kappa \neq 0$. We have that $\kappa x^\star P = \kappa x^\star$ and consequently $\mathbb{1}^\top XP = \mathbb{1}^\top X = (X \mathbb{1})^\top = (XP\mathbb{1})^\top$. Thus, the vector whose entries are the sum of the column of XP is equal to the vector which entries are the sum of the row of XP. Hence, XP is a flow matrix. $\qquad \square$

From Theorems 1 and 2 we have the following corollary:

Corollary 1. Let G_0 be a graph and P_0 its normalized adjacency matrix. Let S be any merging and F and H the merging matrices associated to S. Consider the graph G_1 defined by the matrix A_1 as:

$$A_1 = \kappa \text{Diag}(x_1^\star) F P_0 H^\top, \qquad (19)$$

where $x_1^\star$ is the eigenvector centrality of $P_1 := F P_0 H^\top$ and $\kappa \neq 0$. Then G_1 is a flow graph and $\delta_{G_0, G_1} = 0$.

We have shown in this section that for every 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 G_1 is a flow graph and such that the eigenvector distance between G_1 and G_0 is null. Let us 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 fix κ such that the sum of all weights in G_1 is equal to the sum of all weights in G_0, which is:

$$\kappa = \frac{|A_0|_0}{|\text{Diag}(l) F P_0 H^\top|_0} \qquad (20)$$

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 $G_1 = G_0^{(v,w)}$.

4 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 Problem (9).

4.1 Algorithm description

The results of Section 3 show that for any partition we can choose the weights of the resulting graph such that the two last constraints of Problem (9) are respected. Then, a solution to Problem 9 can be found by solving the new problem:
Given $G_0 = (A, V, E) \in \Gamma \cap \Psi$, find the partition S such that:

$$\tilde{G} = \min_{G} \quad J_{\text{SF}_\alpha}(G),$$
$$\text{where} \quad G_0 \overset{S}{\succ} G \qquad (21)$$

which is : find the best partition of G such that the graph coming out of this partition minimizes the scale-free cost function. The exploration of the set of partitions of a graph being computationally unfeasible, we propose an iterative algorithm in which at each step we look for the merging $S_{v,w}$ minimizing the scale-free cost function. Thus, we just need the result of Theorem 2 for merging. As looking for the best edge within the whole set of edges still requires relatively heavy computations, we will look for the best edge only within a random subset of edges. The effect of this random selection will be discussed later.
A description of the algorithm is presented in Algorithm 1: Therein the graph G_k is represented by (A_k, E_k, V_k). The inputs are

the initial graph $G_0 = (A_0, E_0, V_0) \in \Gamma_n \cap \Psi$, a scale-free coefficient $\alpha_{SF} > 0$ and an integer $n_{rand} \in \mathbb{N}$.

First, we compute the eigenvector centrality $x_0^\star$ of the initial graph (line 1), the normalized adjacency matrix P_0 (line 2) and the coefficient κ (line 3). A random subset $\bar{E}$ of n_{rand} edges is drawn (line 6). For each edge e, we compute the normalized adjacency matrix P_k^e according to (15) where F and H are the merging matrices of S_e (line 8).The scale-free cost function of the graph represented by P_k^e is computed (line 9). Among the n_{rand} edges tested, the best one is the one whose merging minimizes the scale-free cost function (line 11). The new graph is chosen (line 12). Finally the new centrality is computed as the projection of the current centrality onto the merging $S_{e_{best}}$ (line 13). We restart the algorithm with the new graph. When the algorithm stops we compute A_{end} thanks to 19

Algorithm 1 Merge to scale-free,Input: G_0, Parameters: n_{rand}, α_{SF}

Input: A_0, α_{SF}, n_{rand}
Output: A_{end}
1: $x_0^\star = \Phi(G_0)$
2: P_0 from (4)
3: κ from (20)
4: $k = 0$
5: **while** $\neg stop$ **do**
6:    $\bar{E} = \text{rand}(E_k, n_{rand})$;
7:    **for** $e \in \bar{E}$ **do**
8:       P_k^e from (15)
9:       $n_{SF}(e) = J_{SF_\alpha}(P_k^e)$
10:   **end for**
11:   $e_{best} = \text{argmin}_e \, n_{SF}(e)$
12:   $P_{k+1} = P_k^{e_{best}}$
13:   $x_{k+1}^\star = \sigma_{S_{e_{best}}}(x_k^\star)$
14:   k++
15: **end while**
16: A_{end} from (19) with $P_1 = P_k$

Clearly, this algorithm does not provide the global minimum of (9), 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 that decreases the scale-freeness cost function, or as a fixed number of iterations.

4.2 Algorithm complexity

We discuss in this section the complexity of the algorithm in terms of number of operations.

Proposition 1. Consider Algorithm 1 on an initial graph G_0 with N_v nodes, and with n_{rand} the size of the random subset of edges $\bar{E}$. The algorithm can be divided in two phases:

- An initial phase in which the eigenvector centrality is computed. The complexity of this phase is $\Theta(N_v^3)$ in the worst case, but can be lowered to $\Theta(N_v^2)$ in the case of sparse adjacency matrices.
- The reduction phase which consists in at the most N_v steps having each a complexity of $\Theta(N_v n_{rand})$

Overall the worst-case complexity is $\Theta(N_v^3)$.

Proof: For the first phase, the computation of Φ is the computation of an eigenvector which needs $\Theta(N_v^3)$ operations. However in [27] it is shown that if the maximum degree of the

graph is bounded the computation can be done in $\Theta(N_v^2)$.

As at each step of the algorithm the number of nodes decreases by 1, the total number of iterations can not be larger than N_v. We consider then that the number of steps is $\Theta(N_v)$. Let us denote C the total number of operations at each step, we have

$$C = C_{rand} + n_{rand}(C_{(15)} + C_{J_{SF_\alpha}}) + C_{\text{argmin}} + C_\sigma$$

The different functions can be detailed as follows:

- rand consists in picking n_{rand} values, so $C_{rand} = \Theta(n_{rand})$.
- (15) is the computation of the new adjacency matrix, which consists in the combination of the columns and row of the previous adjacency matrix: $\Theta(N_v)$ operations.
- J_{SF_α} can be decomposed as follow:

  - Update of the degree distribution from the adjacency matrix. The number of operation needed is the number of different degrees in the network. Thus, the number of operations is smaller than N_v.
  - Computation of the scale-free cost function: it is the norm of a difference of two vectors which sizes are always smaller than N_v. Thus, the number of operations is also smaller than N_v.

  Finally $C_{J_{SF_\alpha}} = \Theta(N_v)$.
- argmin require $\Theta(n_{rand})$ operations.
- Merge is equivalent to the first step of J_{SF_α}. As we have seen, we have then $C_{\text{Merge}} = \Theta(N_v)$
- σ consists in the combination of the coordinates of $x_k^\star$. Hence $C_\sigma = \Theta(N_v)$

Finally we have:

$$C = \Theta(n_{rand} + n_{rand}(N_v + N_v) + n_{rand} + N_v + N_v)$$
$$= \Theta(N_v n_{rand})$$

proving the statement. $\qquad\square$

This complexity is polynomial with respect to the size of the initial graph, whereas the naive way to find a partition of a graph by testing all possibilities would have had an exponential complexity. Even by improving the partitioning algorithm, the complexity is lower-bounded by the complexity of the eigenvector centrality which can not be lowered. However, this algorithm is not supposed to be run in real time, but once to find a reduced network which can be used then to different purposes. Thus a relatively high complexity is not crippling for the application.

The complexity of the reduction phase is linear with respect to n_{rand}. We will investigate in the next section the influence of this parameter on the performance of the algorithm.

5 EXPERIMENTAL RESULTS

In this section, we set up some experiments to emphasize the influence of n_{rand} on the algorithm. Precisely, we investigate the influence on the speed of convergence and on the variability between different outputs of the algorithm. Based on these results, we can choose a value for the parameter and show the result of the simulation on a synthetic network. Finally, we investigate how the algorithm modifies the topological properties of the initial network. For these simulations we will consider a synthetic family of networks: the Manhattan-like grids. The first subsection aims to introduce these graphs.

5.1 The Manhattan-like grid

Manhattan-like grids, lattice graphs or simply grids, is a family of directed network inspired by the topology of the urban network of Manhattan and other cities [28]. It consists simply in a grid of size $N \times N$ in which each of the N^2 intersections is a node. In our case, in view to get heterogeneous networks we add some random irregularities: some nodes are removed, some diagonal shortcuts are created and some edges are unidirectional. The weights on the edges are generated randomly while ensuring that the grid is a flow network. The advantages of running our algorithm over this type of network are the following: i) the degree distribution is far from a scale-free distribution, so it can shows the ability of the algorithm to get close to a scale-free distribution, ii) it is easy to build this type of network, even with an arbitrarily large size, making the results presented easily reproducible and iii) it is a good representation of some physical networks [29] as urban traffic networks or brain networks [30]. Figure 2 presents an example from this family of network.
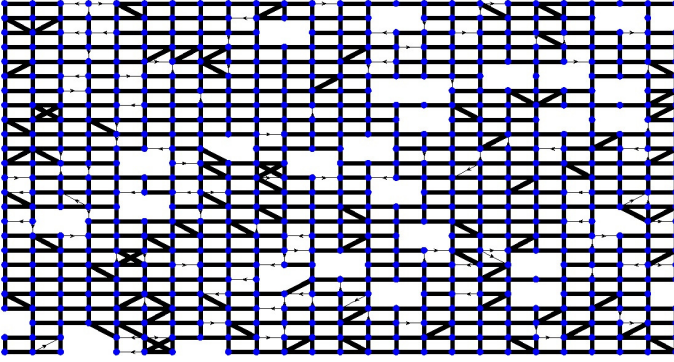

Fig. 2. Example of a Manhattan-like grid 25×25. Thicker edges represents double-way edges.

5.2 Influence of n_{rand} (size of the random subset of edges)

Before analyzing the properties of the output graph, we discuss how the size of the random subset of edges influences the output of the algorithm. We look at its effect on several points:

- The speed of convergence towards a scale-free network.
- The similarity of the properties of different graphs generated by the algorithm
- The similarity of different partitions generated by the algorithm

5.2.1 On the convergence of the algorithm

In Fig. 3 we observe the evolution of the scale-free cost function at each iteration of the algorithm for different values of n_{rand}. Let first remark that for every value of n_{rand} the error initially decreases, even in the case $n_{\mathrm{rand}} = 1$, where the edge to merge is randomly selected. This shows that a graph (at least this type of grids) naturally tends towards a scale-free structure when it is recursively merged. The figure shows that there is no significant advantage in having a large value of n_{rand}. Thus, increasing n_{rand} does not increase substantially the performance while it increases linearly the computation time as seen in Proposition 1.
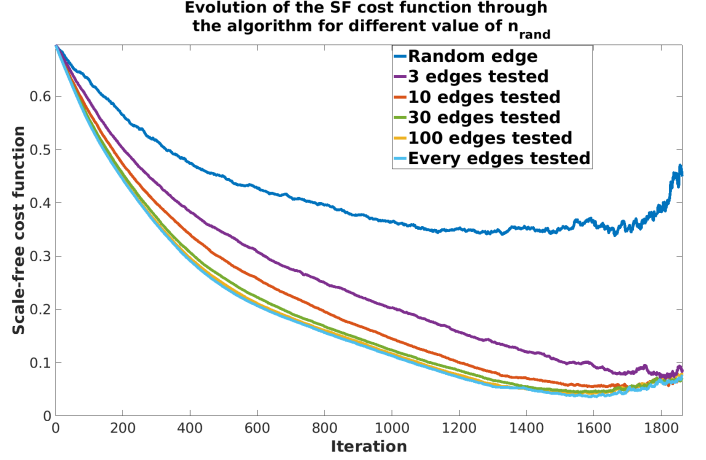

Fig. 3. Comparison of the evolution of the scale-free cost function through the algorithm for different values of n_{rand}. The initial graph is a Manhattan-like grid of size 45×45 and with 1962 nodes. The simulation is stopped when the size of the current graph is equal to 5% of the size of the initial graph.

5.2.2 On the reproducibility of the algorithm

We want now to examine if the graphs obtained via several instances of the algorithm are *close* to each other and how the value of n_{rand} influences this consistency. To answer this question, we have executed several times our algorithm with the same initial graph, and we compare topological properties of the graphs obtained. Precisely, we consider an initial Manhattan-like grid 25×25 and we run the algorithm until there is no more merging that increases the scale-freeness. We have executed the algorithm 50 times with $n_{\mathrm{rand}} = 3$, 50 times with $n_{\mathrm{rand}} = 30$ and once with $n_{\mathrm{rand}} = +\infty$ which is at each step all edges are tested (as the output is deterministic one instance is enough). We compute then four properties for each output graph: number of edges, number of nodes, radius (minimum eccentricity of any node), and scale-free cost function. Let us note that, to have a fair comparison, the radius is divided by the number of nodes in the graph. In Fig. 4 the results are presented in the form of histograms. We remark that the values are arranged around an expected value in an almost bell-shaped distribution. When n_{rand} is higher, this Gaussian behavior is more marked. It appears also that, in this case, the mean deviation is lower and the mean is closer to the case $n_{\mathrm{rand}} = +\infty$ (which can be considered as a reference value). As expected, a large value of n_{rand} reduces randomness.

In addition to the question of the consistency of the properties, we wonder if the partitions obtained with several simulations are close together. For this purpose, we compare the partitions obtained when $n_{\mathrm{rand}} = 3$ and $n_{\mathrm{rand}} = 30$ with the partition of reference obtained with $n_{\mathrm{rand}} = +\infty$. An useful way to compare two partitions is to use the normalized mutual information [31], [32] whose definition is quickly recalled here:

Let us consider two partitions $X = \{X_1,...,X_m\}$ and $Y = \{Y_1,...,Y_p\}$ over a graph[4] of size n. The joint distribution of X_i and Y_j is $P(X_i,Y_j) = \frac{1}{n}|X_i \cap Y_j|$ and the partial distributions are $P(X_i) = \frac{1}{n}|X_i|$ and $P(Y_j) = \frac{1}{n}|Y_j|$. The entropy of the partition X is defined as $H(X) = -\sum_i P(X_i)log(P(X_i))$ and translates how much the set $\{1,..,n\}$ is fragmented in X (if $X = \{\{1,...,n\}\}$, then the entropy is

---

4. We consider here partitions over a graph, but the definition remains the same for partitions over any set.
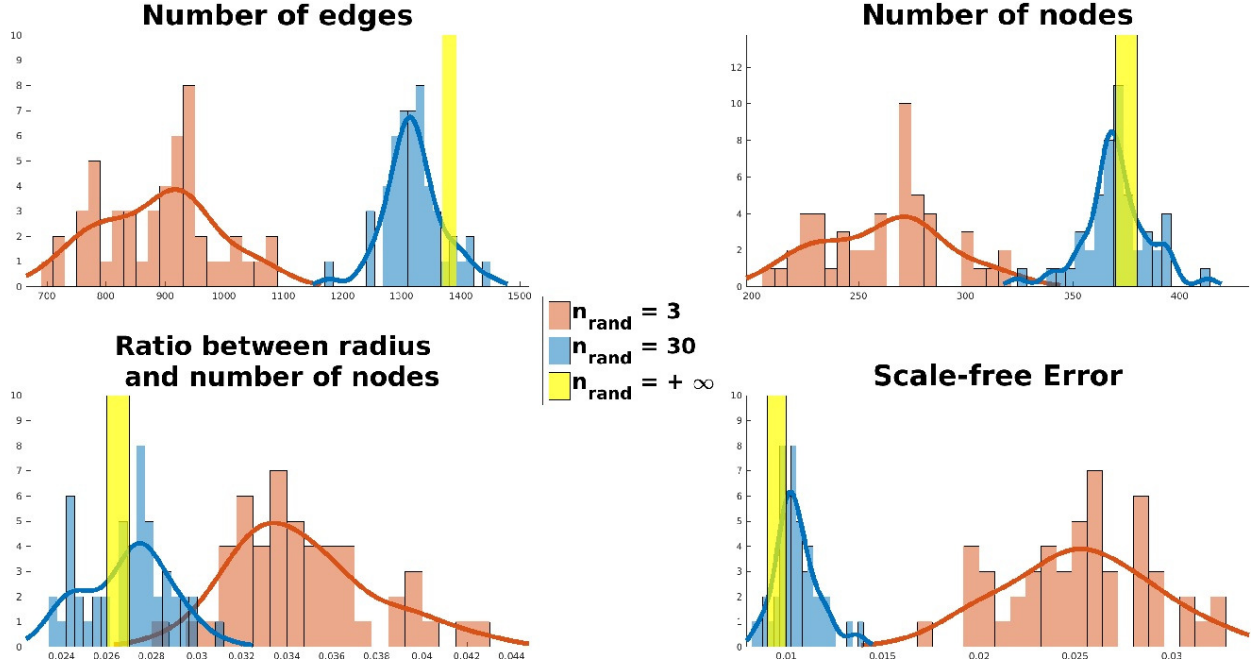
Fig. 4. Histograms of the properties of the different graphs obtained with different instances of the algorithm and with the same initial graph. For each property, the x-axis represents the values and the y-axis the frequency of apparition of each value. The red bars shows the values obtained when $n_{\mathrm{rand}} = 3$, the blue bars when $n_{\mathrm{rand}} = 30$, the yellow bar when $n_{\mathrm{rand}} = +\infty$. The superimposed lines fits the histograms.

null; if $X = \{\{1\}, ..., \{n\}\}$, then the entropy is maximal). The joint entropy is defined as $H(X,Y) = -\sum_{i,j} P(X_i, Y_j) log(P(X_i, Y_j))$ and translates how much the set $\{1,..,n\}$ is fragmented in the *intersection partition* $\{X_1 \cap Y_1, X_2 \cap Y_1, ..., X_m \cap Y_p\}$. Finally the mutual information is defined as $\tilde{I}(X,Y) = H(X) + H(Y) - H(X,Y)$. In a word, it translates how much the *intersection* of X and Y is more fragmented than X and Y are. To calculate the normalized mutual information, we add a normalization factor: $I(X,Y) = \frac{2\tilde{I}(X,Y)}{H(X)+H(Y)}$. Thanks to this tool coming from information theory, we are able to attribute a value $I(X,Y) \in [0;1]$ to measure the similarity between two partitions X and Y. In particular, if $X = Y$ then $I(X,Y) = 1$.

We consider a Manhattan-like grid of size 35×35. To test the consistency of the partitions, we compare 50 partitions obtained with $n_{\mathrm{rand}} = 3$ with the reference partition (obtained when $n_{\mathrm{rand}} = +\infty$). We also make the comparison between the 50 partitions obtained with $n_{\mathrm{rand}} = 30$ and the reference partition. Figure 5 shows the result obtained. Once again we observe a bell-shaped distribution of the values. We observe that the mean value is closer to 1 when n_{rand} is larger. It means that, as expected, when the random effect is reduced, the partitions obtained are closer to the reference partition. We have shown here that the partitions tend to be structured in the same way, and that this effect grows with n_{rand}.

In conclusion, we have seen that when n_{rand} increases: the run time increases linearly, the scale-free error decreases slowly and the variability of outputs decreases. In view of the numerical results, it appears that a relatively small value of the number of edges tested at each step ($n_{\mathrm{rand}} \approx 20$) is a good balance.

The choice of a small subset of edges tested being justified, we can exhibit the output of a simulation on a Manhattan-like grid.

5.3 Simulation on the Manhattan-like grid

In this section we consider an initial Manhattan-like grid and we apply the reduction algorithm to it. The tuning of the different parameters is presented in Table 1.
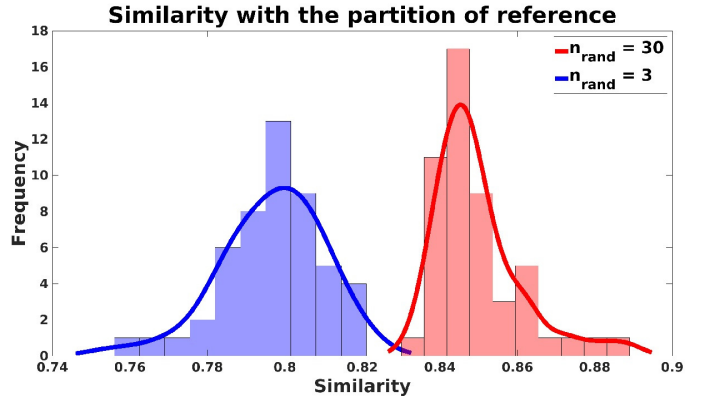

Fig. 5. Histograms of the normalized mutual information between the partition of reference and the 50 partitions obtained with $n_{\mathrm{rand}} = 3$ (in blue) and the 50 partitions obtained with $n_{\mathrm{rand}} = 30$ (in red).

TABLE 1
Parameters of the simulation on the Manhattan-like grid

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

The output of the simulation is presented in Fig. 6. It appears clearly that the algorithm achieves to get the graph very close to the scale-free target distribution. In the next section we wonder how the algorithm also modifies other topological properties.

5.4 Modification of topological properties

The main effect of the algorithm is to modify the degree distribution of an arbitrary graph. However, one can wonder how the other structural properties of the initial graph are changed. The direct comparison between the initial and the final graph may be

8

Initial graph: 3824 nodes

(a) The initial graph. Double-way links are thicker than one-way ones.

Reduced graph: 1062 nodes

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

Final partition

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

In-degree distribution

Distribution reduced graph

Target distribution

Distribution initial graph

Ratio

Degree

(d) In-degree distribution of the initial graph, of the reduced graph and the target distribution in log-log representation.
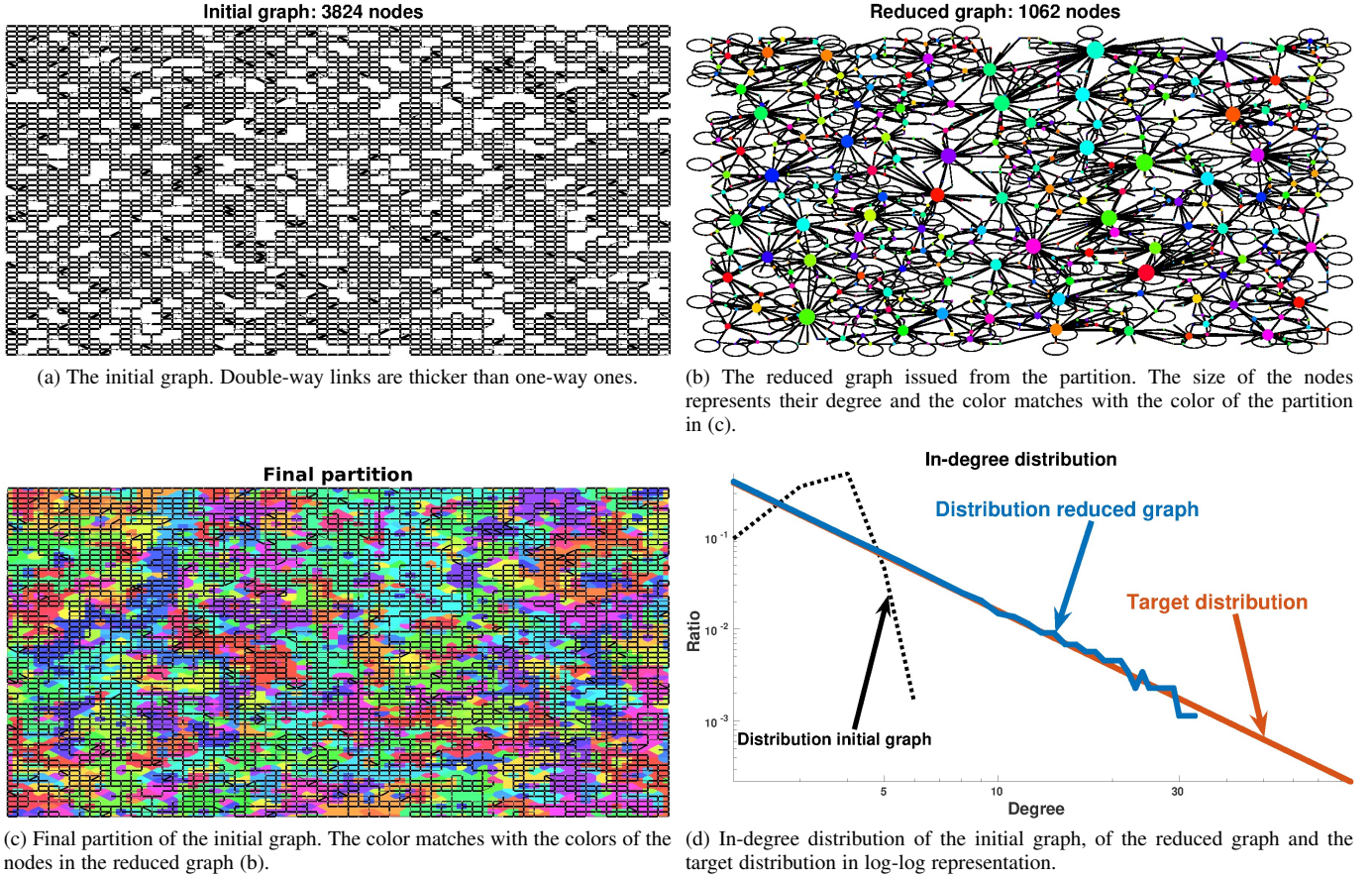
Fig. 6. Result of the Manhattan-like grid simulation. (a) shows the initial graph, (b) shows the final graph, (c) shows the final partition and in (d) the in-degree distributions. The last sub-figure emphasizes well the ability of the graph to drive the degree distribution highly close to the desired distribution

biased by their difference of size. Hence, to compare fairly the properties, we generate a small-scale version of the initial graph build the same way as the initial graph but with a size similar to the final graph. In Table 2 we show the structural properties of the initial graph, the mean values for 50 graphs resulting of the algorithm and the values for the small-scale version of the initial graph, named *reference graph* in the Table. The structural properties compared are the following: Number of nodes, number of edges, radius, diameter (greatest distance between any pair of node), number of hub (we defined a hub as a node connected with at least 5% of all nodes), clustering coefficient (measure the tendency that two linked nodes share a common neighbor), maximum in-degree and mean in-degree.

TABLE 2
Modification of the properties of the network through the reduction

	Initial graph	Reduced graphs	Reference graph
Num. of nodes	1165	516.42	508
Num. of edges	4426	2236.80	1911
Radius	29	9.16	20
Diameter	57	17.22	37
Num. of hub	0	35.14	0
Clust. coeff.	0.1	0.43	0.1
Max in-degree	6	23.3	6
Mean in-degree	3.80	4.33	3.76

The comparison seems fair as the reduced graphs and the reference graph have a similar size and because the initial graph and the reference graph are consistent (no hubs, same clustering coefficient, same max and mean degree).

One can remark that the characteristic distances (radius and diameter) of the reduced graph are significantly lower than the reference graph, while the clustering coefficient is significantly higher. Scale-free networks are known to exhibit ultra small-world property [17]. This property means that the characteristics distances scales as $\log(\log(n))$ and the clustering coefficient is relatively high. The values obtained confirm that the reduced graph endows this scale-free property. The other topological modifications, presence of hubs and higher maximum in-degree, can be explained directly by the power law degree distribution of scale-free graphs.

6 APPLICATION ON A REAL NETWORK

In this section, we will present the result of the algorithm on a physical example: the Grenoble urban traffic network. At first, we present how we obtained this network and why it is a meaningful example.

Situated in the south-east of France close to the Alps, Grenoble is the 16th largest city in France. The situation of the city, surrounded by three ranges of mountains, constraints the urban traffic network making Grenoble the fourth most congested city in France [33]. See Fig. 7 for a satellite picture of the city. In the framework of the ERC-Project ScaleFree-Back, the collection of the traffic data and the monitoring of the traffic condition over the whole city is studied and will be part of the *GTL-ville* experimentation

(following the *GTL* experimentation started in 2009 [34]). As large networks are hard to control, we wonder how an abstracted version of the network could help to design a control strategy and how the scale-freeness of this abstracting network can be an advantage. It is within this frame that the study of the reduction of the urban traffic of Grenoble takes place. The network, provided by TomTom, is an exhaustive representation of the Grenoble metropolis as it contains all roads and intersections within an area of about 60 km^2. In this graph, the nodes correspond to the roads and it exists an edge between two nodes n_i and n_j if there is an intersection linking road n_i to road n_j. The network possesses 19232 nodes and 49600 edges; see Fig. 8. While the structure of the graph comes from the real world, the weights are generated randomly, as in the Manhattan-like grid case, while ensuring the graph to be a flow graph. If these weights represent the mean flow of cars from a street to another, the network is naturally a flow network. The preservation of the flow network property is then essential to ensure that, in the reduced network, the number of cars coming into an area equals the number of car going out. The preservation of the sum of all weights ensured by (20) guarantees that the number of vehicles is the same in the initial and the reduced networks. In the context of traffic network, the computation of a scale-free network abstracting a physical network allows, for example, to identify areas highly connected (corresponding to hubs in the reduced networks) and then potentially vulnerable, or to design a boundary control to rules the in-flow and out-flow in the different areas of the network.

Fig. 7. Satellite picture of Grenoble. We can see the Vercors range in the west, the Chartreuse range in the north and the Belledonne range in the south-east constraining the development of the urban traffic network.

We apply the reduction algorithm on this network and we impose a scale-free distribution both for in-degree and out-degree to show the ability of the algorithm to drive both degree distributions towards different power laws. For example, as suggested in [35], a different scale-free coefficient for in-degree and out-degree distribution can be useful to improve controllability. The choice of the different parameters of the algorithm is presented in Table 3.

TABLE 3
Parameters of the simulation on the Grenoble urban traffic network

$|G|$	$\alpha_{SF,in}$	$\alpha_{SF,out}$	n_{rand}	μ_{in}	μ_{out}
19232	-2.5	-1.8	20	0.5	0.5

The result of the simulation is presented in Fig. 9. For such a large network the execution lasts for about 160 minutes. The

Fig. 8. The Grenoble urban traffic network.

ability of the algorithm to get close to the desired distribution is clear: the match is almost perfect for small degrees and more irregular for higher degrees. The irregularities are due to the small number of nodes having this degree. We observe also that for high degree, the in-degree and out-degree distributions tends to go away from their targets to get closer to each other. This fact shows the difficulty of the algorithm to obtain a scale-free network with different in-degree and out-degree coefficients.

7 CONCLUSION

We have formulated a problem of graph reduction towards a scale-free distribution as an optimization problem: we seek to optimize the scale-freeness of the graph under similarity constraints. We have shown that it is possible, for a type of partition called "merging", to compute the weights of the reduced graph such that these constraints are respected. Thus, we define a partition algorithm which takes advantage of these results and allows to find a sub-optimal solution. Experimental results brought strong clues on the choice of a free-parameter of the algorithm, and show that we can speed up the execution with almost the same efficiency. Finally, the algorithm is applied on a physical network: the Grenoble urban traffic network showing the ability of the algorithm to reduce large-scale network (about twenty thousand nodes), and to drive both in-degree and out-degree distributions towards different desired power laws in a reasonable time. Let us remark that the algorithm could be used to drive a graph towards any desired structure and still verify the similarity constraints.

REFERENCES

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

[2] Y. Liu, T. Safavi, A. Dighe, and D. Koutra, "Graph summarization methods and applications: A survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, p. 62, 2018.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] D. Gfeller and P. De Los Rios, "Spectral coarse graining of complex networks," *Physical review letters*, vol. 99, no. 3, p. 038701, 2007.

[8] P. Erdös and A. Rényi, "On random graphs, i," *Publicationes Mathematicae (Debrecen)*, vol. 6, pp. 290–297, 1959.

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

[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] S.-H. Yook, H. Jeong, and A.-L. Barabási, "Modeling the internet's large-scale topology," *Proceedings of the National Academy of Sciences*, vol. 99, no. 21, pp. 13 382–13 386, 2002.

[13] 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.

[14] D. d. S. Price, "A general theory of bibliometric and other cumulative advantage processes," *Journal of the Association for Information Science and Technology*, vol. 27, no. 5, pp. 292–306, 1976.

[15] ——, "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.

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

[17] R. Cohen, S. Havlin, and D. Ben-Avraham, "Structural properties of scale-free networks," *Handbook of graphs and networks*, p. 85, 2003.

[18] M. Boguná, F. Papadopoulos, and D. Krioukov, "Sustaining the internet with hyperbolic mapping," *Nature communications*, vol. 1, p. 62, 2010.

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

[20] R. Pastor-Satorras and A. Vespignani, *Epidemics and immunization in scale-free networks*. Wiley-Blackwell, 2005, ch. 5, pp. 111–130.

[21] 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.

[22] A. Blum, J. Hopcroft, and R. Kannan, "Foundations of data science," *Vorabversion eines Lehrbuchs*, 2016.

[23] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Technical Report 1999-66, November 1999. [Online]. Available: http://ilpubs.stanford.edu:8090/422/

[24] L. Fu, X. Wang, and P. Kumar, "Are we connected ? Optimal determination of source–destination connectivity in random networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 2, pp. 751–764, 2017.

[25] L. Fu, X. Fu, Z. Xu, Q. Peng, X. Wang, and S. Lu, "Determining source–destination connectivity in uncertain networks: Modeling and solutions," *IEEE/ACM Transactions on Networking*, vol. 25, no. 6, pp. 3237–3252, 2017.

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

[27] E. Valakevicius and M. Snipas, "A recursive algorithm for computing steady state probabilities," *IIIS*, 2010.

[28] D. Stanislawski, "The origin and spread of the grid-pattern town," *Geographical Review*, vol. 36, no. 1, pp. 105–120, 1946.

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

[30] V. J. Wedeen, D. L. Rosene, R. Wang, G. Dai, F. Mortazavi, P. Hagmann, J. H. Kaas, and W.-Y. I. Tseng, "The geometric structure of the brain fiber pathways," *Science*, vol. 335, no. 6076, pp. 1628–1634, 2012.

[31] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 09, p. P09008, 2005.

[32] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3, pp. 75–174, 2010.

[33] (2018) Tomtom traffic index. [Online]. Available: https://www.tomtom.com/trafficindex/

[34] C. Canudas-De-Wit, F. Morbidi, L. L. Ojeda, A. Y. Kibangou, I. Bellicot, and P. Bellemain, "Grenoble traffic lab: An experimental platform for advanced traffic monitoring and forecasting," *IEEE Control Systems*, vol. 35, no. 3, pp. 23–39, 2015.

[35] G. Lindmark and C. Altafini, "A driver node selection strategy for minimizing the control energy in complex networks," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 8309–8314, 2017.

[36] (2018) Simulation of Grenoble urban traffic network reduction. [Online]. Available: https://www.youtube.com/watch?v=tTxJmOReqKE

Nicolas Martin received the M.Sc. degree in applied mathematics from Institut National des Sciences Appliques (INSA) of Toulouse, France in 2016. In 2015, he made an internship at the European Southern Observatory (ESO) in Chile, Santiago. In 2016, he made an internship at the Centre National d'Etude Spatiale (CNES) in Toulouse, France. Since 2016 he is a Ph.D. student in GIPSA-lab at Grenoble, France, under the supervision of Carlos Canudas-de-Wit and Paolo Frasca.

Paolo Frasca (M'13, SM'18) received the Ph.D. degree in Mathematics for Engineering Sciences from Politecnico di Torino, Italy, in 2009. From 2013 to 2016, he has been an Assistant Professor at the University of Twente, the Netherlands. Since October 2016, he is a CNRS researcher at GIPSAlab, Grenoble, France. His research interests are in the theory of network systems and cyber-physical systems, with applications to robotic, sensor, infrastructural, and social networks.

Carlos Canudas-de-Wit was born in Villahermosa, Mexico, in 1958. He received the B.S. degree in electronics and communications from the Monterrey Institute of Technology and Higher Education, Monterrey, Mexico, in 1980, and the M.S. and Ph.D. degrees in automatic control from the Department of Automatic Control, Grenoble Institute of Technology, Grenoble, France, in 1984 and 1987, respectively. He is currently a Directeur de recherche (Senior Researcher) with CNRS, Grenoble, where he is the Leader of the NeCS Team, a joint team of GIPSA-Lab (CNRS) and INRIA, on networked controlled systems. Dr. Canudas-de-Wit is an IEEE-CSS Fellow and an IFAC Fellow. He is an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, the Automatica, the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, the Asian Journal of Control, and the IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS. He served as the President of the European Control Association from 2013 to 2015, and the IEEE Board of Governors of the Control System Society from 2011 to 2014. He holds the ERC Advanced Grant Scale-FreeBack from 2016 (19992002).
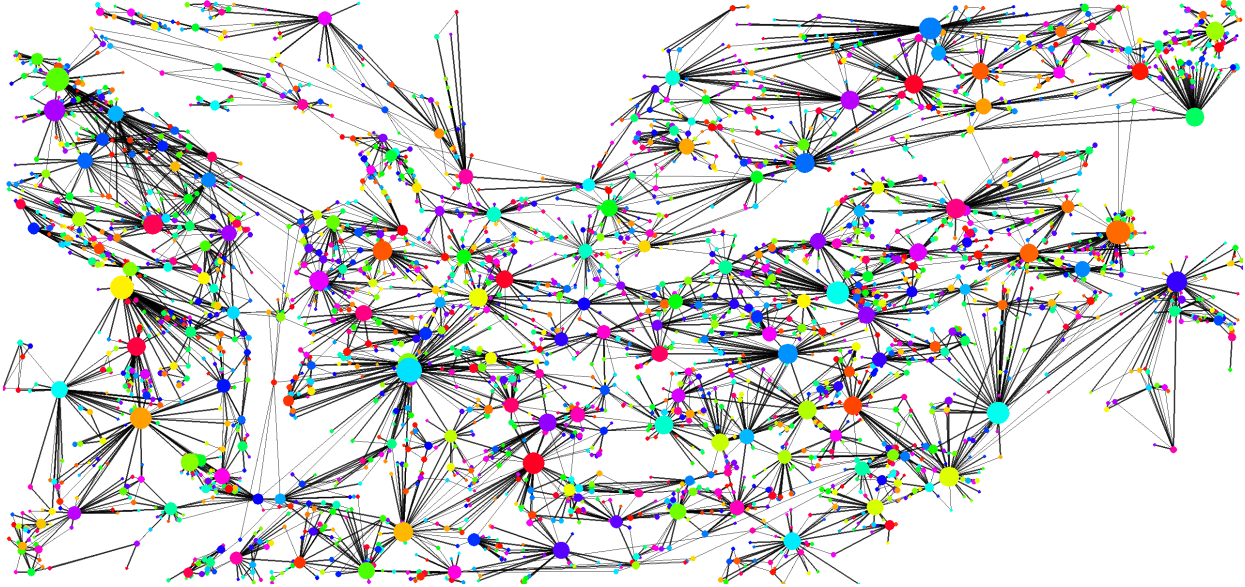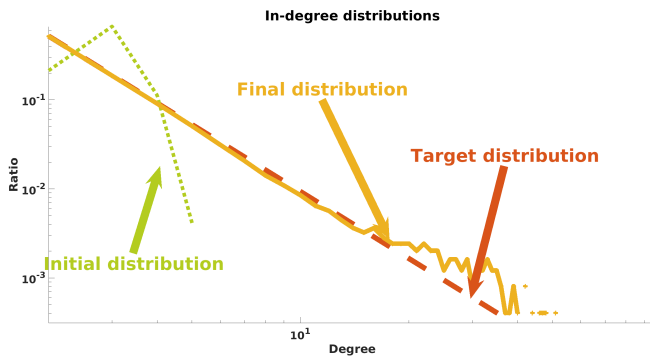
Final partition
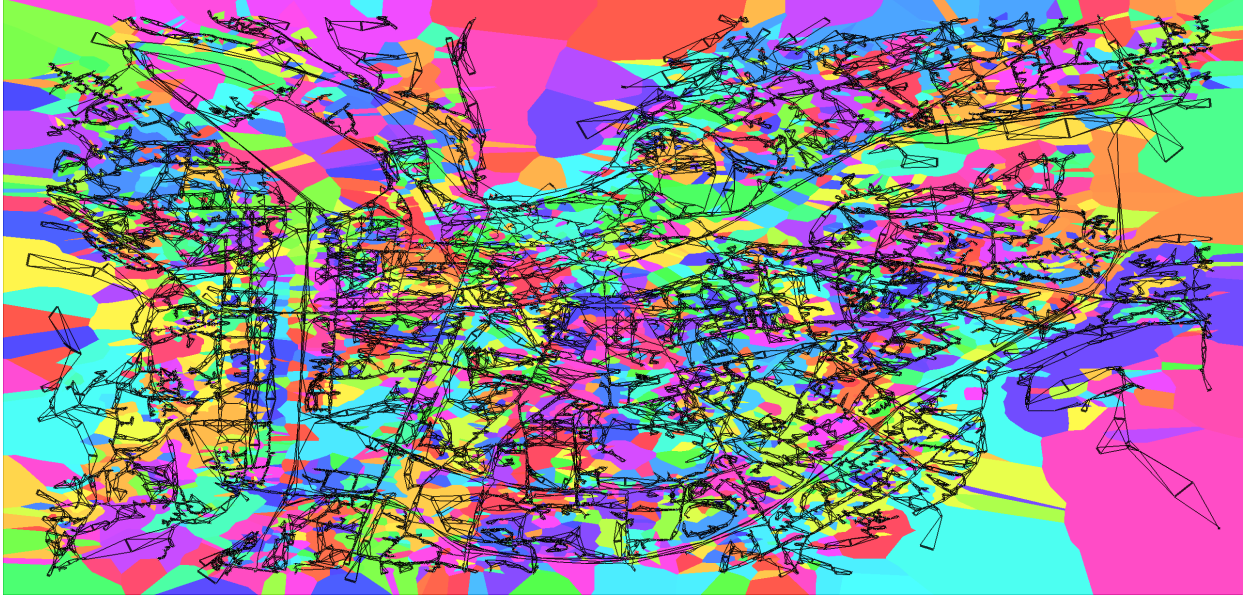
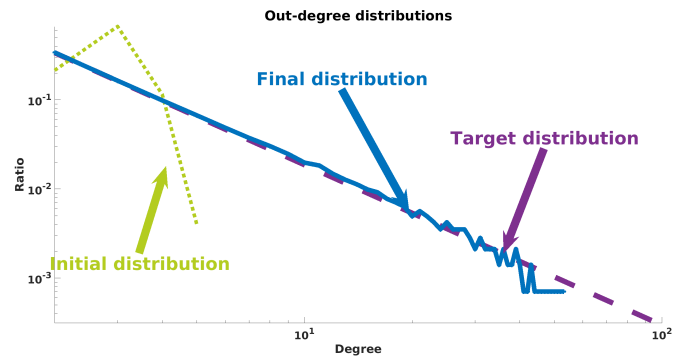(a) The initial graph. Background colors indicate the final partition obtained via the algorithm

Reduced graph : 3253 nodes

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

(c) Initial, final and target in-degree distributions in logarithmic scales.

(d) Initial, final and target out-degree distributions in logarithmic scales.

Fig. 9. Output of the Grenoble urban traffic network reduction. A video showing the evolution of the algorithm step-by-step of a similar simulation is available at [36]