Filtering graphs to check isomorphism and extracting mapping by using the Conductance Electrical Model[☆]

Manuel Igelmo[a], Alberto Sanfeliu[a,*]

[a]Institut de Robòtica i Informàtica Industrial, CSIC-UPC
Llorens i Artigas 4-6, 08028 Barcelona, Spain

Abstract

This paper presents a new method of filtering graphs to check exact graph isomorphism and extracting their mapping. Each graph is modeled by a resistive electrical circuit using the Conductance Electrical Model (CEM). By using this model, a necessary condition to check the isomorphism of two graphs is that their equivalent resistances have the same values, but this is not enough, and we have to look for their mapping to find the sufficient condition. We can compute the isomorphism between two graphs in $O(N^3)$, where N is the order of the graph, if their star resistance values are different, otherwise the computational time is exponential, but only with respect to the number of repeated star resistance values, which usually is very small. We can use this technique to filter graphs that are not isomorphic and in case that they are, we can obtain their node mapping. A distinguishing feature over other methods is that, even if there exits repeated star resistance values, we can extract a partial node mapping (of all the nodes except the repeated ones and their neighbors) in $O(N^3)$. The paper presents the method and its application to detect isomorphic graphs in two well know graph databases, where some graphs have more than 600 nodes.

Keywords: graph isomorphism, graph matching, Conductances Equivalent Model, Star Method, graph filter

[☆] This work was partially funded by CICYT DPI2013-42458-P.

[*] Corresponding author

Email addresses: migelmo@xtec.cat (Manuel Igelmo), sanfeliu@iri.upc.edu (Alberto Sanfeliu)

URL: www.iri.upc.edu (Manuel Igelmo), www.iri.upc.edu (Alberto Sanfeliu)

1. Introduction

It is known that the a graph is a powerful and flexible structure which allow modeling many types of objects and systems, due to this, graphs are used in many fields such as chemistry, biochemistry, transport, telephony, computer networks, voice recognition, computer vision, etc. [1]; in many cases the graphs have a high number of nodes and/or edges [2].

In the field of Pattern Recognition, the process of evaluating the similarity of two graphs is referred as graph matching. In this area we can differentiate between two type of the methods: exact and inexact graph matching. The stringent way of defining the exact graph matching is the graph isomorphism, meanwhile the inexact graph matching looks for the best mapping between the graphs through minimizing a matching cost. There are numerous works that deal with the state of the art on the graph matching, such as [3], [4], [5], [6] and [7]. Other papers ([8] and [9] among others) perform comparisons between different methods.

Two graphs are isomorphic when any node renumbering preserves adjacencies (unweighted graphs) or weights (weighted graphs). The graph as a data structure, has the great drawback that the comparison between them requires computationally prohibitive calculation time [10], i.e. exponential time complexity with respect to the number of nodes. That is why there is a vast and extensive literature[1] to find reasonably quick ways to decide when two graphs are identical, i.e. isomorphic, and also if applicable, to extract the mapping between their nodes. Moreover, it is known that the graph isomorphism problem belongs to **NP**, but not known to belong to either one of the following subsets: **P** and **NP**-complete [12] (see also [13]).

As we have already commented the graph isomorphism (exact matching) is an open problem, in contrast to other related graphs problems whose computa-

[1] In [11] (published in 2013), the authors assert that there are a few hundreds of algorithms published on the subject.

tional complexity has been shown to be **NP**-complete such as graph homomorphism, subgraph isomorphism and maxim common subgraph[2] of graphs whose proof can be found in [14], [10] and [15] respectively, so that all efforts are being dumped in search in polynomial time suboptimal solutions for these problems.

The foregoing is for graphs in general, but there are subsets of graphs for which there has been shown subexponential solutions to the problem of isomorphism, such as planar graphs [16, 17, 18, 19], rooted trees [20], graphs of bounded degree [21], interval graphs [22], circular graphs chords [23, 24] and arcs [24], graphs of bounded genus [25, 26, 27], graphs of bounded eigenvalue [28] and graphs of bounded treewidth [29].

There are other approaches to the problem of graph isomorphism for general case. Many of them use a tree search of solutions, these algorithms use brute force but with pruning to nonviable solutions and backtracking techniques. These differ essentially in the criteria for pruning, thus they have the algorithms of Ullmann [30], SD [31] and VF [32, 33].

Other methods use the Theory of Groups seeking a canonical labeling of graphs allowing to discern whether they are isomorphic through their respective canonical equality [34]. These techniques also make use of a search tree and automorphisms of graphs. However, as is affirmed in [35], in terms of computational complexity, the theoretical state of canonical labeling is still unsolved. All these algorithms have been computationally implemented giving rise to (in chronological order) “nauty” [36], “saucy” [37], “Bliss” [38, 39], “Traces” [11] and “conauto” [40].

Other inexact methods can also be applied to match graphs, not to solve the isomorphism problem, which find a cost to map one graph to another one. There is an extensive literature on this topic which have already been mentioned ([3], [4], [5], [6] and [7]). We are not going mention these methods, because is out of the scope of this article.

[2] The maximum common subgraph problem is reducible to the problem of clique and this is **NP**-complete.

In this paper we present a completely new method for filtering graph isomorphism and at the same time, extract their node mapping that neither derives nor inspired by any of the aforementioned methods. This method can be applied to attributed graphs with only one numeric attribute (weight) in each edge, and for connected and undirected graphs. It also serves to unweighted graph if these are taken to each edge a unit weight. It can not be applied to graphs with symbolic labels.

Our method, denominated the Star Method (hereinafter SM) is based on the Conductance Electrical Model (hereinafter CEM) [41]. It models weighted graphs where its weighted edges are transformed in conductances values (S) (we use conductances instead of resistance values (Ω)). The method can be also applied to unweighed graphs, where the value of the edge weight is equal to 1 in this case. By assuming serial connection of an ideal diode with a resistor, the method can be extended to directed graphs. Unfortunately this extension brings nonlinearities making the analysis much more complex (in terms of the circuit).

Using an electric model, we can apply the theories, methods and procedures that are well known in Electrical Circuit Theory (see among others [42]). In the literature, we have only find a work, [43], that uses also electrical circuit representation, but oriented to define a resistance distance to match graph models.

Although the method that we propose is oriented to solve exact graph isomorphism in an efficient way, from the point of view of computational time complexity, reducing from exponential to cubic time complexity in most of the cases, we present the work as a filtering technique to eliminate the graphs that are not isomorphic, and detecting and extracting the node mapping of the graphs that are isomorphic. The reason is that in this way, the method can be applied to solve problems where checking isomorphism is the key issue. The proposed method uses the Conductance Electrical Model (CEM) and has two filtering processes. The first filtering process eliminates the graphs when the equivalent resistances do not match. The second filtering process, either detect that there exist an isomorphism and in this case extract the correct node mapping, or detect the graphs that are not isomorphic. The important difference is that the first process is cubic, $O(N^3)$, and the second process can be quadratic or exponential, but in this case only with respect to the star resistance values that are identical. That implies that in most cases, the graph isomorphism can be done in cubic time complexity, making this filtering process very efficient.

In order to compare our method with other well known methods, Table 1 shows a comparison using the following features: best time complexity case; worst time complexity case; if the method uses tree search; if it is an iterative method; if it can be obtained a partial matching; and if the method has a closed form.

We have selected these features to show that our method has some strengths. First, the best time complexity case is the same than the other methods. Second, the worst time complexity case is better than the other methods $max(O(N^3, O(J!))$, because $J \ll N$. Third, it has a closed form, it is not probabilistic, not iterative neither recursive. Fourth, it can be obtained a partial mapping in $O(N^3)$ time complexity, a feature that no other exact methods have. Finally, we have to underline that our method is based on a well known electrical circuit theory.

	Best case	*Worst case*	*Search tree?*	*Iterative?*	*Partial mapping?*	*Closed form?*
Ullmann [30]	$O(N^3)$	$O(N!N^3)$	yes	yes	no	no
SD [31]	$O(N^3)$	$O(N!N)$	yes	yes	no	no
VF [32]	$O(N^3)$	$O(N!N)$	yes	yes	no	no
Nauty [36]	$O(N^2 \log N)$	exponential	yes	yes	no	no
SM (this paper)	$O(N^3)$	$\max(O(N^3), O(J!))$	no	no	yes	yes

Table 1: Comparison of features of some methods. Pay special attention to the column "Partial mapping?".

The rest of the paper is devoted to present the proposed model and method (CEM and SM) for filtering, analyze its characteristics and present experiments to verify the performance of them.

2. Filtering graphs to check isomorphism by using SM

The Figure 1 shows a block diagram of the SM using CEM, that will be used for the description of the method. The graphs modeled as CEM are characterized by having one numerical attribute in each edge (we will call them weights and they can be any non-negative value) and no attributes in their nodes. We can treat also unweighted graphs by assigning value 1 to the attribute of all edges. Hereinafter the two graphs modeled by the CEM will be denoted by g and h, and we will assume that both have the same order N.

2.1. First filter phase: Obtaining CEM and equivalent resistances

Consider two undirected and connected graphs (weighted or unweighted[3]) g and h both of order N and size M. These graphs come characterized by their adjacency matrices A^g and A^h respectively (input and 1st line of block A of Figure 1). The CEM consists of modeling each graph by an electric circuit composed exclusively for M resistors[4] with the same topology as the graph (edges are replaced by resistors). The value of each resistor in the circuit is defined by a step function (ϕ).

Definition 1. *The step function is:*

$$\phi(\omega_{ij}) = c_{ij} \qquad \text{(S)} \tag{1}$$

where ω_{ij} is the weight of the edge connecting the nodes i and j of the graph, and c_{ij} is the conductance [5] (in siemens) that connects nodes i and j in the CEM.

It is important to make clear that the CEM weights are transformed into conductances (S) instead of resistances (Ω). In this way, when two nodes are

[3] In this case we consider that they have unit weights in all edges, i.e., for a undirected graph we always consider in this paper that if nodes i and j are connected, then $\omega_{ij} = 1$.

[4] In what follows we will always explicitly distinguish between resistor (device) and resistance (opposition to the passage of electric current measured in Ω).

[5] In what follows we will use the letter c for conductances instead of the usual g, since the latter will be used to represent graphs.

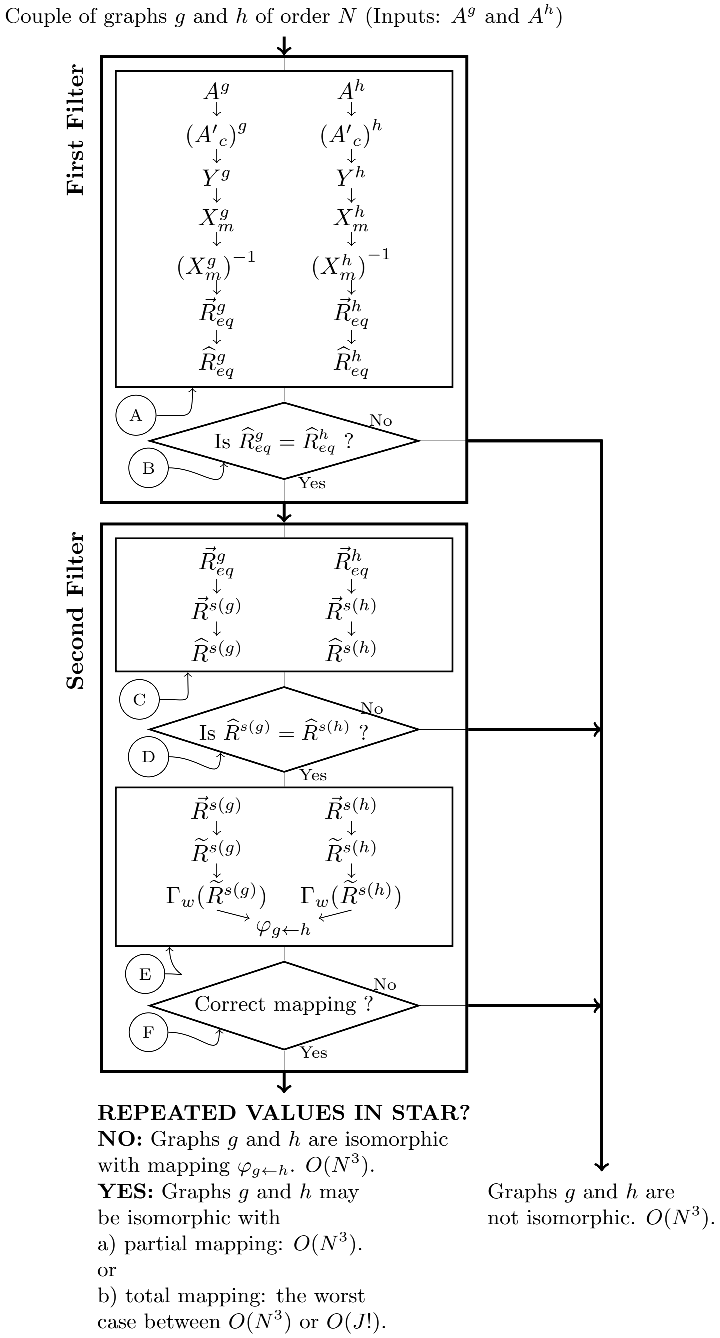

Couple of graphs g and h of order N (Inputs: A^g and A^h)
First Filter
$A^g \to (A'_c)^g \to Y^g \to X_m^g \to (X_m^g)^{-1} \to \vec{R}_{eq}^g \to \widehat{R}_{eq}^g$
$A^h \to (A'_c)^h \to Y^h \to X_m^h \to (X_m^h)^{-1} \to \vec{R}_{eq}^h \to \widehat{R}_{eq}^h$
A
Is $\widehat{R}_{eq}^g = \widehat{R}_{eq}^h$?
B
No
Yes
Second Filter
$\vec{R}_{eq}^g \to \vec{R}^{s(g)} \to \widehat{R}^{s(g)}$
$\vec{R}_{eq}^h \to \vec{R}^{s(h)} \to \widehat{R}^{s(h)}$
C
Is $\widehat{R}^{s(g)} = \widehat{R}^{s(h)}$?
D
No
Yes
$\vec{R}^{s(g)} \to \widetilde{R}^{s(g)} \to \Gamma_w(\widetilde{R}^{s(g)})$
$\vec{R}^{s(h)} \to \widetilde{R}^{s(h)} \to \Gamma_w(\widetilde{R}^{s(h)})$
$\varphi_{g\leftarrow h}$
E
Correct mapping ?
F
No
Yes
REPEATED VALUES IN STAR?
NO: Graphs g and h are isomorphic with mapping $\varphi_{g\leftarrow h}$. $O(N^3)$.
YES: Graphs g and h may be isomorphic with
a) partial mapping: $O(N^3)$.
or
b) total mapping: the worst case between $O(N^3)$ or $O(J!)$.
Graphs g and h are not isomorphic. $O(N^3)$.

Figure 1: Block diagram of the filtering isomorphic graphs by SM using CEM. The maximum number of repeated resistances (if any) in each star is denoted by J. The letters inscribed in circles from “A” to “F” are not part of the block diagram and are only used for references in the text.

not connected in the graph, the corresponding value in the adjacency matrix will be zero.

The decision to choose the step function, depends strongly on the physical meaning of the weights of the graph and, consequently, depends on the context of the problem, in other words, the step function is a design parameter. Moreover the step function has to accomplish with:

1. The step function must be injective. This restriction is absolutely necessary if we want to recover the graph model, i.e., ϕ^{-1} should exist.
2. The step function $\phi(0) = 0\,\mathrm{S}$. As explained above, if two nodes are not connected, the corresponding value in the adjacency matrix is zero consequently the conductance must be zero.

The characterization of the circuit of graph g is given by the adjacency matrix of conductances.[6]

$$(A')^g = \begin{pmatrix} 0 & c_{12} & c_{13} & \dots & c_{1N} \\ c_{12} & 0 & c_{23} & \dots & c_{2N} \\ c_{13} & c_{23} & 0 & \dots & c_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{1N} & c_{2N} & c_{3N} & \dots & 0 \end{pmatrix} \; (\mathrm{S}) \qquad (2)$$

and similarly for the graph h (2^{nd} line of block A of Figure 1).

However in the Circuit Theory field, they do not use this matrix, instead they use the Indefinite Admittance Matrix (IAM) and for this work we will use

[6] This matrix should not be confused with the adjacency matrix of the graph. Therefore, in this work, the notation $(A')^g$ (adjacency matrix of conductances) is used to distinguish it from A^g (adjacency matrix).

the IAMs matrices of graphs g and h. For graph g the IAM is:

$$Y^g = \begin{pmatrix} \sum\limits_{\substack{j=1\\ j\neq 1}}^{N} c_{1,j} & -c_{1,2} & \cdots & -c_{1,N} \\ -c_{2,1} & \sum\limits_{\substack{j=1\\ j\neq 2}}^{N} c_{2,j} & \cdots & -c_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ -c_{N,1} & -c_{N,2} & \cdots & \sum\limits_{\substack{j=1\\ j\neq N}}^{N} c_{N,j} \end{pmatrix} \text{(S)} \tag{3}$$

and similarly for the graph h (3rd line of block A of Figure 1).

Note that $c_{ij} = c_{ji}$, since we are using a pure resistive circuit for modeling the graphs, and therefore c_{ij} or c_{ji} is used when needed.

We use the IAM, as the CEM of each graph (g and h), and we will apply the Circuit Theory field methods to solve the graph isomorphism. Specifically, our method (SM) is based on the computation of all equivalent resistances of each graph represented by their CEM (see Appendix A relating to 4th, 5th and 6th lines of block A of Figure 1).

Note that for a graph with N nodes, there are $N(N-1)/2$ equivalent resistances if the circuit has only resistors, since $r_{eq_{ij}} = r_{eq_{ji}}$.

For a given circuit is trivial that the value of an equivalent resistance does not depend on the numbering of the nodes. Indeed, suppose that the node i is numbered with u, and node j with v then $r'_{eq_{uv}} = V_{uv}/I_s$. Although the numbering have changed, the electrical circuit is the same, then the voltages (since they are potential differences) have to be identical, $V_{ij} = V_{uv}$. Therefore their equivalent resistances fulfill

$$r'_{eq_{uv}} = V_{uv}/I_s = V_{ij}/I_s = r_{eq_{ij}} \tag{4}$$

In summary, if two nodes of a circuit are renumbered (without changing the circuit), then the value of the equivalent resistances are the same.

At this point we have $N(N-1)/2$ equivalent resistances extracted from the graph model as shown in the Appendix A and especially Table 5. Let us show three different formats to represent the all equivalent resistances of a graph that will be used in the text.

1. By a square matrix R^g_{eq} of order N,

$$R^g_{eq} = \begin{pmatrix} 0 & r^g_{eq_{12}} & \cdots & r^g_{eq_{1,N}} \\ r^g_{eq_{12}} & 0 & \cdots & r^g_{eq_{2,N}} \\ \vdots & \vdots & \ddots & \vdots \\ r^g_{eq_{1,N}} & r^g_{eq_{2,N}} & \cdots & 0 \end{pmatrix} (\Omega) \tag{5}$$

2. By a column vector $\vec{R}^g_{eq}$ with $N(N-1)/2$ elements. The sequence of the elements is such that the equivalent resistances are assigned according to the numbering of the nodes: for each i from 1 to $N-1$ for each j from $i+1$ to N. The vector is

$$\begin{aligned} \vec{R}^g_{eq} = (&r^g_{eq_{12}}, r^g_{eq_{13}}, r^g_{eq_{14}}, \ldots, r^g_{eq_{1,N-1}}, r^g_{eq_{1,N}}, \\ &r^g_{eq_{23}}, r^g_{eq_{24}}, \ldots, r^g_{eq_{2,N-1}}, r^g_{eq_{2,N}}, \\ &\vdots \\ &r^g_{eq_{N-2,N-1}}, r^g_{eq_{N-2,N}}, \\ &r^g_{eq_{N-1,N}})^t \qquad (\Omega) \end{aligned} \tag{6}$$

3. By using the set $\widehat{R}^g_{eq}$ (7th line of block A of Figure 1) consisting of all ordered pairs of the form $(r^g_{eq_z}, f_z)$ where $r^g_{eq_z}$ is a value of the equivalent resistance and f_z is the absolute frequency of repetition of that value (if the value $r^g_{eq_z}$ is not repeated then $f_z = 1$).[7] The compact form of $\widehat{R}^g_{eq}$ is

$$\widehat{R}^g_{eq} = \big\{(r^g_{eq_z}, f_z)|k = 1, \ldots, L\big\} \tag{7}$$

[7] Note that z does not correspond to any node numbering.

where L $(1 \leq L \leq N(N-1)/2)$ is the ordered pairs number, the frequencies must satisfy

$$f_1 + f_2 + \cdots + f_L = N(N-1)/2 \tag{8}$$

Note that the first and second representation contains the same information. In contrast, the third loses node information, because we can not recover the node to which belongs the equivalent resistance.

Also note that the equivalent resistance between nodes i and j does not depend of the chosen reference node m, due that equation (47) does not take into account the chosen reference node m.

Using the previous results (specially (4)) and based on the well known method on equivalent resistances of the electrical circuit theory field, where is proved that two identical electrical circuits have the same values of the equivalent resistances (see [42]), we can assert that *a necessary condition for detecting isomorphism between two graphs, is that if g and h are isomorphic graphs of order N then the sets $\widehat{R}^g_{eq}$ and $\widehat{R}^h_{eq}$ are the same*, that is

$$g \cong h \Rightarrow \widehat{R}^g_{eq} = \widehat{R}^h_{eq} \tag{9}$$

where the sign $\cong$ denotes isomorphic graphs. Equation (9) is only *a necessary condition* that must accomplish two isomorphic graphs, however there are graphs that are not isomorphic and have the same equivalent resistances set $(\widehat{R}_{eq})$, we will call them *co-resistance graphs* (see subsection of co-resistance graphs). Let us shown where we can find these co-resistance graphs. Because we have obtained the equivalent resistances, R^g_{eq}, doing forward linear operations (from A^g to R^g_{eq}), we can go backwards and recovering the original nodes from R^g_{eq} to A^g. Because there are $N(N-1)/2$ equivalent resistances, the number of graph permutations that can be recovered going backwards is $[N(N-1)/2]!$, and $N!$ are isomorphic, then $[N(N-1)/2]! - N!$ can not be isomorphic. Take into account that many of these potential co-resistance graphs will not ever being a graph because they will not accomplish the constraint imposed by an adyacency matrix (weights must be non-negative).

As summary, the expression (9) is *a necessary condition*, and we have to look for *the sufficient condition* to assure that two graphs are isomorphic. However, expression (9) allows us to filter (question B of Figure 1) many graphs that will not be isomorphic because they will not accomplish with condition (9).

In the second filter, we will explain *the sufficient condition* and how to obtain the node mapping of the graphs.

2.2. Second filter phase: Approximation of the equivalent resistances by a star circuit and validation process

Let us consider a star circuit with $N+1$ nodes and one resistor per each one of the branches with resistance value r_k where $k = 1, \dots, N$. Just for convenience and without loss of generality, we assume that the last node $(N + 1)$ is the root and the order of the rest of the nodes is arbitrary. The numbering of each resistor is the same of its leaf node (see Figure 2).

Our purpose is to get the resistances of the star circuit of Figure 2 taking into account the equivalent resistances of the original circuit computed in the first filter phase. That means, we want to obtain the r_k, $k = 1, \dots, N$, star resistances, from $N(N - 1)/2$ equivalent resistances of the original circuit, and in case that exist an isomorphism, then get the mapping between the graphs.

2.2.1. Obtaining the star resistances

In order to obtain the r_k values which will form the vector $\vec{R}$ (2^{nd} line of block C of Figure 1), we should minimize the Mean Square Error (henceforth MSE) between the equivalent resistances of the star circuit ($r'_{eq_{ij}}$) and the equivalent resistances of the original circuit ($r_{eq_{ij}}$). For this purpose and thereafter we consider the root node in star is hidden.

It is easy to see that in the star the equivalent resistance between nodes i and j is $r'_{eq_{ij}} = r_i + r_j$ (two resistors in series). The proposed approach is to

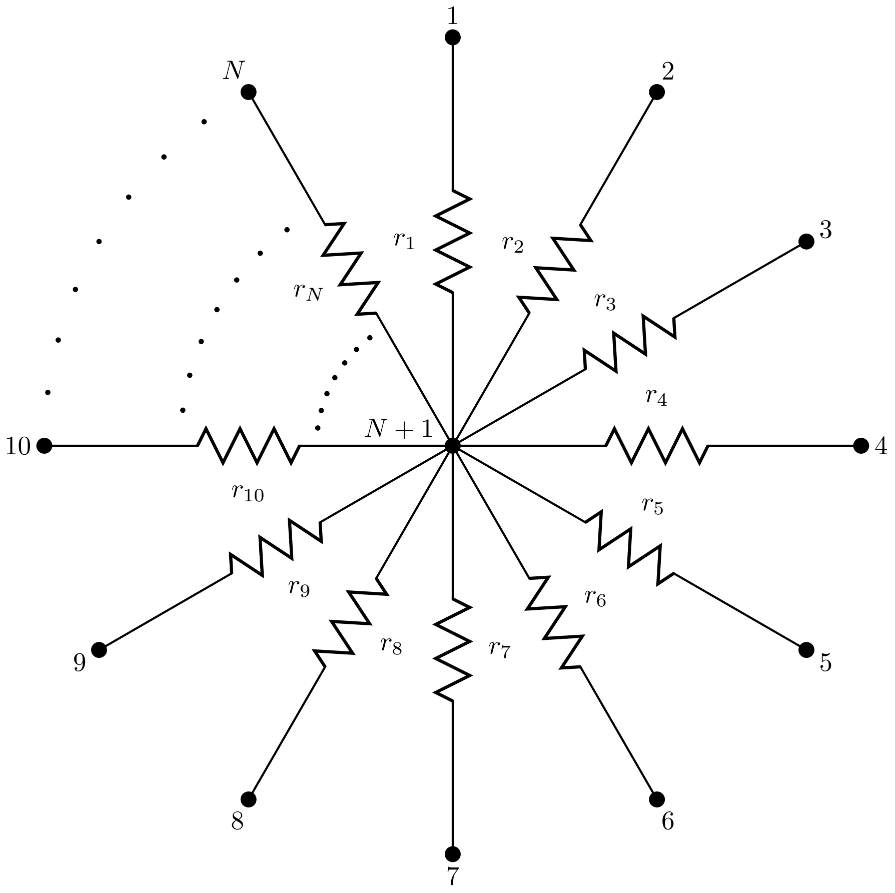


Figure 2: Generic star resistor circuit of $N+1$ nodes and N resistors.

replace $r'_{eq_{ij}}$ by $r_{eq_{ij}}$, therefore we can write the following matricial expression for all the equivalent resistances.

$$\begin{pmatrix} 1100\cdots00 \\ 1010\cdots00 \\ 1001\cdots00 \\ \vdots\vdots\vdots\vdots\ddots\vdots\vdots \\ 1000\cdots10 \\ 1000\cdots01 \\ 0110\cdots00 \\ 0101\cdots00 \\ \vdots\vdots\vdots\vdots\ddots\vdots\vdots \\ 0100\cdots10 \\ 0100\cdots01 \\ 0011\cdots00 \\ \vdots\vdots\vdots\vdots\ddots\vdots\vdots \\ 0010\cdots10 \\ 0010\cdots01 \\ 0001\cdots00 \\ \vdots\vdots\vdots\vdots\ddots\vdots\vdots \\ 0001\cdots10 \\ 0001\cdots01 \\ \vdots\vdots\vdots\vdots\ddots\vdots\vdots \\ \vdots\vdots\vdots\vdots\ddots\vdots\vdots \\ 0000\cdots11 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ \vdots \\ r_{N-1} \\ r_N \end{pmatrix} = \begin{pmatrix} r_{eq_{12}} \\ r_{eq_{13}} \\ r_{eq_{14}} \\ \vdots \\ r_{eq_{1,N-1}} \\ r_{eq_{1,N}} \\ r_{eq_{23}} \\ r_{eq_{24}} \\ \vdots \\ r_{eq_{2,N-1}} \\ r_{eq_{2,N}} \\ r_{eq_{34}} \\ \vdots \\ r_{eq_{3,N-1}} \\ r_{eq_{3,N}} \\ r_{eq_{45}} \\ \vdots \\ r_{eq_{4,N-1}} \\ r_{eq_{4,N}} \\ \vdots \\ \vdots \\ r_{eq_{N,N-1}} \end{pmatrix} \tag{10}$$

and in a more compact way, it can be rewritten as

$$K\vec{R} = \vec{R}_{eq} \tag{11}$$

where K is the coefficient matrix, $\vec{R}_{eq}$ is the independent vector and $\vec{R}$ is the vector of the unknowns of the expression (10). This is a system of $N(N-1)/2$ linear equations with N unknowns, and thus, due that always $N(N-1)/2 > N$ for $N > 3$, the system will be overdetermined.[8] This system generally has no solution unless there are enough linearly dependent equations.

In any basic treaty of Numerical Analysis, we can find that the approximate solution that minimizes the MSE is given by expression

$$\vec{R} = (K^t K)^{-1} K^t \vec{R}_{eq} \tag{12}$$

The matrix $(K^t K)^{-1} K^t$ with N rows and $N(N-1)/2$ columns is known as the pseudoinverse of Moore-Penrouse (hereinafter simply called by the pseudoinverse) and is designated by K^+. This matrix is

$$K^+ = (K^t K)^{-1} K^t \tag{13}$$

Then we can rewrite (12) as

$$\vec{R} = K^+ \vec{R}_{eq} \tag{14}$$

It has to be noted that the pseudoinverse is the same for all graphs with the same order, regardless of any other considerations (the pseudoinverse only changes if N is changed).

The above result shows the general way of getting $\vec{R}$, but formula (14) requires to compute the inverse of a matrix product and is computationally expensive. The coefficients of $\vec{R}$ can be computed using a straightforward formula and the time complexity can be reduced from $O(N^3)$ to $O(N^2)$ in this step. Using the results shown in [44] and the derivation shown in Appendix B of this paper, the values r_k of $\vec{R}$ with $k = 1, \ldots, N$ can be computed as follows:

$$r_k = \tfrac{1}{(N-1)(N-2)} \left((N-1)\Psi(k) - \Psi_t\right) \tag{15}$$

[8] Note that for $N = 3$, the system is not overdetermined and we consider it a degenerate case.

where

$$\Psi(k) = \sum_{j=1}^{N} r_{eq_{kj}} \text{ for } k = 1, 2, \ldots, N \tag{16}$$

and

$$\Psi_t = \sum_{i=1}^{N-1} \left(\sum_{j=i+1}^{N} r_{eq_{ij}} \right) \tag{17}$$

At this point (2^{nd} line of block C of Figure 1), we have obtained the N star resistance values (r_k) represented by vector $\vec{R}$, that minimizes MSE equivalent resistances between the original circuit ($r_{eq_{ij}}$) and the star circuit ($r'_{eq_{ij}}$). In what follows the letter s is reserved to denote a star graph (or star circuit), and we will use the notation $s(g)$ to denote the star graph (or star circuit) that comes when we apply SM to the graph g.

2.2.2. Obtaining the mapping of the isomorphic graphs

Now we can test the *sufficient condition* of graph isomorphism, filtering out the graphs that are not isomorphic and obtaining the node mapping of the graphs that are isomorphic. In order to do the mapping between both graphs, we use the r_k of both graphs as it is explained below.

Let us present three different types of representations of the N values of r_k that we need for the mapping procedure. These representations are for a circuit s of order $N + 1$ and star topology (root is $N + 1$), and they are:

1. By a column vector $\vec{R}^s$ with N elements. The element label is assigned according to the numbering of the nodes of the star as follows:

$$\vec{R}^s = (r_1^s, r_2^s, \ldots, r_N^s)^t \qquad (\Omega) \tag{18}$$

2. By the set $\widehat{R}^s$ consisting of all L ordered pairs of the form (r_z^s, f_z) where r_z^s is a resistant value and f_z is the absolute frequency of repetition of that

value (if the value r_z^s is not repeated then $f_z = 1$).[9] Then

$$\widehat{R}^s = \{(r_z^s, f_z) | z = 1, \ldots, L\} \tag{19}$$

and the frequencies must satisfy ($1 \leq L \leq N$)

$$f_1 + f_2 + \cdots + f_L = N \tag{20}$$

3. By the sequence $\widetilde{R}^s$ of N ordered pairs of the form (r_k, k) which are ordered from the lowest to the highest value of r_k ($k = 1, \ldots, N$), where k is a non-root node of the star and r_k is the resistive value whose non-root node is k.

Note that in $\widehat{R}^s$ the node information that corresponds to the star resistance value is lost, so it will be impossible to obtain $\vec{R}^s$ or $\widetilde{R}^s$ from $\widehat{R}^s$. The following definition uses the third representation.

Definition 2. *The $\Gamma_w(\widetilde{R}^s)$ function obtains the value of the second component (node of the star) of the pair that occupies the w-th position in the sequence $\widetilde{R}^s$.*

Now we can describe how we proceed with the mapping between two graphs.

Star case: We have already explained that *the necessary condition* for graph isomorphism is that both graph have to have the identical set of equivalent resistances with repetitions, but in general this condition is not enough. However, for the case of a star this condition is also *the sufficient condition.* Two undirected graphs s_1 and s_2 of order $N + 1$ with star topology, they will be isomorphic if, and only if, the set of weights with repetitions of the two graphs coincide, that is

$$s_1 \cong s_2 \Leftrightarrow \widehat{R}^{s_1} = \widehat{R}^{s_2} \tag{21}$$

General case: One way to do the mapping of both graphs is to look for the

[9] See note 7.

canonical graphs of both graphs and do the mapping between them. However, in general does not exist this canonical graph, but in our case we have transformed a graph in a star, and we have their r_k values. Then we are able to get a canonical graph of each one of them, by ordering the star resistances using these values in increasing order. In this way we can look for the isomorphism, matching one to one the values of both canonicals (see the "Isomorphism mapping compatibility").

With this in mind, and taken into account the previous definitions, that $\widehat{R}^g_{eq} = \widehat{R}^h_{eq}$ and hence $\widehat{R}^{s(g)} = \widehat{R}^{s(h)}$,[10] and the previous two cases, then the mapping (3rd and 4th lines of block E of Figure 1) between the nodes of g on h is done as follows

$$\varphi_{h\leftarrow g}\left(\Gamma_w(\widetilde{R}^{s(g)})\right) = \Gamma_w(\widetilde{R}^{s(h)}) \tag{22}$$

and h on g is

$$\varphi_{g\leftarrow h}\left(\Gamma_w(\widetilde{R}^{s(h)})\right) = \Gamma_w(\widetilde{R}^{s(g)}) \tag{23}$$

for $w = 1, \ldots, N$ in both cases. it is obvious that $\varphi_{h\leftarrow g} = \varphi^{-1}_{g\leftarrow h}$.

2.2.3. Validation process

As a co-resistance may have occurred (see co-resistance subsection), when both stars have the same r_k values and in the same order (see example 3 below), is necessary to validate (question F of Figure 1) the mapping given by (22) or alternatively by (23). Then we apply an algorithm to detect if there exists the correct mapping between both graphs. The pseudocode presented in algoritm 1 shows the validation algorithm where it has been sufficiently commented (its operation is shown in the example 1). The time complexity of this pseudocode is polynomial of order $O(N^2)$. At this point the second filtering ends.

The output of the two phases in the filtering process has three outputs: isomorphic graph, not isomorphic, or possibly isomorphic graphs with almost complete (but partial) mapping nodes, due to possible repetitions of the branches

[10] In case that $\widehat{R}^g_{eq} = \widehat{R}^h_{eq}$ but $\widehat{R}^{s(g)} \neq \widehat{R}^{s(h)}$ (question D of Figure 1), then the graphs are not isomorphic (they are co-resistance), and filtering ends here (see example 3 below).

Algorithm 1 Validation algorithm of mapping obtained by the SM.

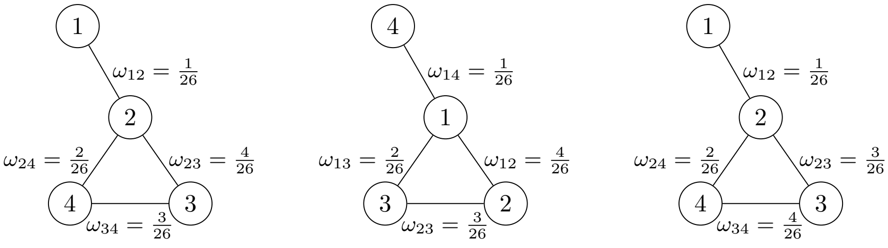
```
Require: two matrices and a function, these are
    1) Adjacency matrices of $g$ and $h$, $A^g$ and $A^h$ respectively. The element of row $i$ and
    column $j$ is denotated by $A^g(i,j)$ and $A^h(i,j)$ respectively (weight between nodes $i$ and
    $j$ if they are adjacent or zero if they are not). $N$ is the orden of any of these matrices
    (graphs order).
    2) The function map$(h,g,k)$. This function gives the node in the graph $h$ corresponding
    to node $k$ of the graph $g$ according to the mapping obtained by SM. This is $\varphi_{h\leftarrow g}(k)$.
Ensure: $FalsePos$. If $FalsePos$ is true then map function does not correspond to a valid
    isomorphism.
$FalsePos \leftarrow$ false; $i \leftarrow 1$; $j \leftarrow 2$
while not $FalsePos$ and $i < N$ and $j \leq N$ do
    if $A^g(i,j) == A^h(\text{map}(h,g,i), \text{map}(h,g,j))$ then
        $j \leftarrow j+1$
        if $j > N$ then
            $i \leftarrow i+1$
            $j \leftarrow i+1$
        end if
    else
        $FalsePos \leftarrow$ true
    end if
end while
return $FalsePos$
```

of the stars.

Let us going to present several examples to shown the filtering phases.

2.3. Example 1

Let us illustrate the matching between graph g (Figure 3a) with graphs h (Figure 3b) and q (Figure 3c). The first pair (g and h) are isomorphic and the second pair (g and q) are not isomorphic.


1
$\omega_{12} = \frac{1}{26}$
2
$\omega_{24} = \frac{2}{26}$
$\omega_{23} = \frac{4}{26}$
4
3
$\omega_{34} = \frac{3}{26}$
4
$\omega_{14} = \frac{1}{26}$
1
$\omega_{13} = \frac{2}{26}$
$\omega_{12} = \frac{4}{26}$
3
2
$\omega_{23} = \frac{3}{26}$
1
$\omega_{12} = \frac{1}{26}$
2
$\omega_{24} = \frac{2}{26}$
$\omega_{23} = \frac{3}{26}$
4
3
$\omega_{34} = \frac{4}{26}$

(a) Graph g.

(b) Graph h.

(c) Graph q.

Figure 3: The graph g is isomorphic to the graph h, however it is not isomorphic to the graph q.

The adjacency matrices corresponding to g, h and q are respectively

$$A^g = \frac{1}{26}\begin{pmatrix} 0&1&0&0 \\ 1&0&4&2 \\ 0&4&0&3 \\ 0&2&3&0 \end{pmatrix}, \ A^h = \frac{1}{26}\begin{pmatrix} 0&4&2&1 \\ 4&0&3&0 \\ 2&3&0&0 \\ 1&0&0&0 \end{pmatrix} \text{ and } A^q = \frac{1}{26}\begin{pmatrix} 0&1&0&0 \\ 1&0&3&2 \\ 0&3&0&4 \\ 0&2&4&0 \end{pmatrix}$$

Then we apply the corresponding CEM IAMs, the results are

$$Y^g = \frac{1}{26}\begin{pmatrix} 1&-1&0&0 \\ -1&7&-4&-2 \\ 0&-4&7&-3 \\ 0&-2&-3&5 \end{pmatrix}, \ Y^h = \frac{1}{26}\begin{pmatrix} 7&-4&-2&-1 \\ -4&7&-3&0 \\ -2&-3&5&0 \\ -1&0&0&1 \end{pmatrix}$$

and

$$Y^q = \frac{1}{26}\begin{pmatrix} 1&-1&0&0 \\ -1&6&-3&-2 \\ 0&-3&7&-4 \\ 0&-2&-4&6 \end{pmatrix}$$

where identity has been used for step function (1).

In all three cases, we have arbitrarily taken the last node ($m = 4$), as reference node. Eliminating the 4th row and 4th column, then we obtain the MDAs matrices. These matrices are respectively

$$X^g = \frac{1}{26}\begin{pmatrix} 1&-1&0 \\ -1&7&-4 \\ 0&-4&7 \end{pmatrix}, \ X^h = \frac{1}{26}\begin{pmatrix} 7&-4&-2 \\ -4&7&-3 \\ -2&-3&5 \end{pmatrix}$$

and

$$X^q = \frac{1}{26}\begin{pmatrix} 1&-1&0 \\ -1&6&-3 \\ 0&-3&7 \end{pmatrix}$$

where subscript m is omitted for clarity.

Their inverses are

$$(X^g)^{-1} = \begin{pmatrix} 33&7&4 \\ 7&7&4 \\ 4&4&6 \end{pmatrix}, \ (X^h)^{-1} = \begin{pmatrix} 26&26&26 \\ 26&31&29 \\ 26&29&33 \end{pmatrix}$$

and

$$(X^q)^{-1} = \begin{pmatrix} 33 & 7 & 3 \\ 7 & 7 & 3 \\ 3 & 3 & 5 \end{pmatrix}$$

For each matrix, Table 5 is applied and the respective equivalent resistances are obtained, these are shown below.

$$\vec{R}^g_{eq} = (26, 31, 33, 5, 7, 6)^t \tag{24}$$
$$\vec{R}^h_{eq} = (5, 7, 26, 6, 31, 33)^t \tag{25}$$
$$\vec{R}^q_{eq} = (26, 32, 33, 6, 7, 5)^t$$

From which, the sets $\widehat{R}^g_{eq}$, $\widehat{R}^h_{eq}$ and $\widehat{R}^q_{eq}$ are build, which are (the values of these sets have been ordered by courtesy).

$$\widehat{R}^g_{eq} = \{(5,1),(6,1),(7,1),(26,1),(31,1),(33,1)\}$$
$$\widehat{R}^h_{eq} = \{(5,1),(6,1),(7,1),(26,1),(31,1),(33,1)\}$$
$$\widehat{R}^q_{eq} = \{(5,1),(6,1),(7,1),(26,1),(32,1),(33,1)\}$$

As it can be seen, $\widehat{R}^g_{eq} \neq \widehat{R}^q_{eq}$ so the pair of graphs g and q are not isomorphic and the filtering process finish here for these pair of graphs. For the pairs g and h, the process continue. Because $\widehat{R}^g_{eq} = \widehat{R}^h_{eq}$, we have to compute the r_k of each graph, extract the mapping and do the validation process. We compute the pseudoinverse using equation (67). The pesudoinversa for $N = 4$ is

$$K^+ = \frac{1}{6} \begin{pmatrix} 2 & 2 & 2 & -1 & -1 & -1 \\ 2 & -1 & -1 & 2 & 2 & -1 \\ -1 & 2 & -1 & 2 & -1 & 2 \\ -1 & -1 & 2 & -1 & 2 & 2 \end{pmatrix} \tag{26}$$

For both graphs we obtain the same pseudoinverse. Multiplying the pseudoinverse (26) by the vector $\vec{R}^g_{eq}$ (24) and $\vec{R}^h_{eq}$ (25), we obtain the resistances values of the star, i.e.

$$\vec{R}^{s(g)} = (r_1^{s(g)}, r_2^{s(g)}, r_3^{s(g)}, r_4^{s(g)})^t = (27, 1, 3, 5)^t \tag{27}$$

$$\vec{R}^{s(h)} = (r_1^{s(h)}, r_2^{s(h)}, r_3^{s(h)}, r_4^{s(h)})^t = (1, 3, 5, 27)^t \tag{28}$$

The same result can be reached in a straightforward way using equation (15), and the Ψ^g_t and Ψ^h_t are

$$\Psi^g_t = 26 + 31 + 33 + 5 + 7 + 6 = 108 \tag{29}$$

$$\Psi^h_t = 5 + 7 + 26 + 6 + 31 + 33 = 108 \tag{30}$$

Both values coincide because g and h are isomorphic and we will denote them in this example as Ψ_t. The star $s(g)$ has the following values

$$\begin{array}{l}
r_1^{s(g)} = \frac{1}{6}(3\Psi^g(1) - \Psi_t) = \frac{1}{6}(3(26 + 31 + 33) - 108) = 27 \\
r_2^{s(g)} = \frac{1}{6}(3\Psi^g(2) - \Psi_t) = \frac{1}{6}(3(26 + 5 + 7) - 108) = 1 \\
r_3^{s(g)} = \frac{1}{6}(3\Psi^g(3) - \Psi_t) = \frac{1}{6}(3(31 + 5 + 6) - 108) = 3 \\
r_4^{s(g)} = \frac{1}{6}(3\Psi^g(4) - \Psi_t) = \frac{1}{6}(3(33 + 7 + 6) - 108) = 5
\end{array} \tag{31}$$

and the star $s(h)$ has the values

$$\begin{array}{l}
r_1^{s(h)} = \frac{1}{6}(3\Psi^h(1) - \Psi_t) = \frac{1}{6}(3(5 + 7 + 26) - 108) = 1 \\
r_2^{s(h)} = \frac{1}{6}(3\Psi^h(2) - \Psi_t) = \frac{1}{6}(3(5 + 6 + 31) - 108) = 3 \\
r_3^{s(h)} = \frac{1}{6}(3\Psi^h(3) - \Psi_t) = \frac{1}{6}(3(7 + 6 + 33) - 108) = 5 \\
r_4^{s(h)} = \frac{1}{6}(3\Psi^h(4) - \Psi_t) = \frac{1}{6}(3(26 + 31 + 33) - 108) = 27
\end{array} \tag{32}$$

The results (31) and (32), coincides with those previously obtained in (27) and (28) respectivaly, but in an efficient way.

We can see that $\widehat{R}^{s(g)} = \widehat{R}^{s(h)}$ (recall that $\widehat{R}^g_{eq} = \widehat{R}^h_{eq}$) and the resistance sequences of $s(g)$ and $s(h)$ are

$$\widetilde{R}^{s(g)} = ((1,2),(3,3),(5,4),(27,1))$$
$$\widetilde{R}^{s(h)} = ((1,1),(3,2),(5,3),(27,4))$$

Note that the first components of the ordered pairs of both sequences coincide and that the second component (node numbering of each graph) is used for extracting the mapping. Applying equation (22) for w equal 1, 2, 3 and 4, we have

$$\varphi_{h\leftarrow g}\left(\Gamma_1(\widetilde{R}^{s(g)})\right) = \Gamma_1(\widetilde{R}^{s(h)}) \Rightarrow \varphi_{h\leftarrow g}(2) = 1 \tag{33}$$
$$\varphi_{h\leftarrow g}\left(\Gamma_2(\widetilde{R}^{s(g)})\right) = \Gamma_2(\widetilde{R}^{s(h)}) \Rightarrow \varphi_{h\leftarrow g}(3) = 2 \tag{34}$$
$$\varphi_{h\leftarrow g}\left(\Gamma_3(\widetilde{R}^{s(g)})\right) = \Gamma_3(\widetilde{R}^{s(h)}) \Rightarrow \varphi_{h\leftarrow g}(4) = 3 \tag{35}$$
$$\varphi_{h\leftarrow g}\left(\Gamma_4(\widetilde{R}^{s(g)})\right) = \Gamma_4(\widetilde{R}^{s(h)}) \Rightarrow \varphi_{h\leftarrow g}(1) = 4 \tag{36}$$

The resulting mapping can be checked by inspecting graphs g and h in subfigures 3a and 3b respectively. In order to finish, the validation process should be applied, for this example, the validation process (see algorithm 1) is as follows:

1. Begin algorithm with (in this example) $N \leftarrow 4$
2. $FalsePos \leftarrow$ **false**, $i \leftarrow 1$ and $j \leftarrow 2$ (1st line)
3. **while** condition is met (2nd line): 1st iteration
    - (a) As map$(h, g, 1)$ and map$(h, g, 2)$ are 4 and 1 respectively then $A^h(4, 1)$ is $\frac{1}{26}$, the latter coincides with $A^g(1, 2)$ therefore the **if** condition is met in 3rd line
    - (b) $j \leftarrow 3$ (4th line)
    - (c) No $j > N$ then the **if** condition is not met in 5th line
4. **while** condition is met (2nd line): 2nd iteration
    - (a) As map$(h, g, 1)$ and map$(h, g, 3)$ are 4 and 2 respectively then $A^h(4, 2)$ is 0, the latter coincides with $A^g(1, 3)$ therefore the **if** condition is met in 3rd line
    - (b) $j \leftarrow 4$ (4th line)
    - (c) No $j > N$ then the **if** condition is not met in 5th line

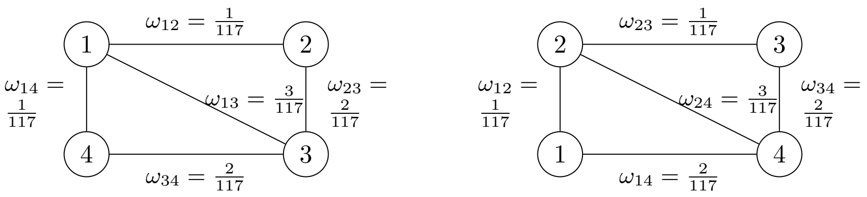
5. **while** condition is met (2nd line): 3rd iteration
    (a) As map$(h, g, 1)$ and map$(h, g, 4)$ are 4 and 3 respectively then $A^h(4,3)$ is 0, the latter coincides with $A^g(1,4)$ therefore the **if** condition is met in 3rd line
    (b) $j \leftarrow 5$ (4th line)
    (c) It holds that $j > 4$ then the **if** condition is met in 5th line. Therefore $i \leftarrow 2$ and $j \leftarrow 3$ (6th and 7th lines)
6. **while** condition is met (2nd line): 4th iteration
    (a) As map$(h, g, 2)$ and map$(h, g, 3)$ are 1 and 2 respectively then $A^h(1,2)$ is $\frac{4}{26}$, the latter coincides with $A^g(2,3)$ therefore the **if** condition is met in 3rd line
    (b) $j \leftarrow 4$ (4th line)
    (c) No $j > N$ then the **if** condition is not met in 5th line
7. **while** condition is met (2nd line): 5th iteration
    (a) As map$(h, g, 2)$ and map$(h, g, 4)$ are 1 and 3 respectively then $A^h(1,3)$ is $\frac{2}{26}$, the latter coincides with $A^g(2,4)$ therefore the **if** condition is met in 3rd line
    (b) $j \leftarrow 5$ (4th line)
    (c) It holds that $j > 4$ then the **if** condition is met in 5th line. Therefore $i \leftarrow 3$ and $j \leftarrow 4$ (6th and 7th lines)
8. **while** condition is met (2nd line): 6th iteration
    (a) As map$(h, g, 3)$ and map$(h, g, 4)$ are 2 and 3 respectively then $A^h(2,3)$ is $\frac{3}{26}$, the latter coincides with $A^g(3,4)$ therefore the **if** condition is met in 3rd line
    (b) $j \leftarrow 5$ (4th line)
    (c) It holds that $j > 4$ then the **if** condition is met in 5th line. Therefore $i \leftarrow 4$ and $j \leftarrow 5$ (6th and 7th lines)
9. **while** condition is not met (2nd line)
10. End algorithm with result **false** for $FalsePos$

2.4. Example 2

This example (see Figure 4) shows how the SM is consistent when the graph automorphism exists.

The matrix R^g_{eq} and R^h_{eq} are

$$R^g_{eq} = \begin{pmatrix} 0 & 51 & 27 & 51 \\ 51 & 0 & 42 & 78 \\ 27 & 42 & 0 & 42 \\ 51 & 78 & 42 & 0 \end{pmatrix}$$


$\omega_{12} = \frac{1}{117}$
1
2
$\omega_{14} = \frac{1}{117}$
$\omega_{13} = \frac{3}{117}$
$\omega_{23} = \frac{2}{117}$
4
3
$\omega_{34} = \frac{2}{117}$
$\omega_{23} = \frac{1}{117}$
2
3
$\omega_{12} = \frac{1}{117}$
$\omega_{24} = \frac{3}{117}$
$\omega_{34} = \frac{2}{117}$
1
4
$\omega_{14} = \frac{2}{117}$

(a) Graph g with automorphism between nodes 2 and 4.

(b) Graph h with automorphism between nodes 1 and 3.

Figure 4: The graphs g and h are isomorphic and also present automorphisms (nodes 2 and 4 of g and nodes 1 and 3 of h).

and

$$R^h_{eq} = \begin{pmatrix} 0 & 51 & 78 & 42 \\ 51 & 0 & 51 & 27 \\ 78 & 51 & 0 & 42 \\ 42 & 27 & 42 & 0 \end{pmatrix}$$

Note that because of the symmetries of the graphs (automorphism), there are equivalent resistance with repeated values (in this example $42\,\Omega$ and $51\,\Omega$). The values of the star resistances are

$$\vec{R}^{s(g)} = (r_1^{s(g)}, r_2^{s(g)}, r_3^{s(g)}, r_4^{s(g)})^t = (16, 37, 7, 37)^t$$
$$\vec{R}^{s(h)} = (r_1^{s(h)}, r_2^{s(h)}, r_3^{s(h)}, r_4^{s(h)})^t = (37, 16, 37, 7)^t$$

Where it can be shown that due to the automorphism, there exist more than one valid isomorphism mapping. Nodes 1 and 3 of g correspond to the nodes 2 and 4 of h respectively. In turn, node 2 and 4 of g correspond to nodes 3 and 1 of h. These results can be verified by comparing the subfigures 4a and 4b. This result is “natural” and does not indicate any abnormality.

3. Characteristics of the SM

We will analyze in the following subsections the isomorphism mapping compatibility, time complexity and co-resistances of the SM using CEM.

3.1. Isomorphism mapping compatibility

We are going to show in this subsection, that there exists an isomorphism mapping compatibility between the equivalent resistances of the original graphs and the star resistances, in such a way that we can use this compatibility to do the matching between two graphs with the star resistances. This isomorphism mapping compatibility is used for doing the node assignment with the star resistances to look for the isomorphism between two graphs.

Let us consider that the graphs g and h are connected, undirected and has order N, with numbering going from 1 to N. We have already seen that these graphs can be modeled as pure resistive circuits. Let be $\varphi(\cdot)$ any permutation of the nodes of g that can map one to one, the nodes of the isomorphic graph h. The equivalent resistances extracted from both graphs accomplish the following equation:

$$r^{g}_{eq_{i,j}} = r^{h}_{eq_{\varphi(i),\varphi(j)}} \tag{37}$$

for $i = 1, 2, \ldots, N-1$ and $j = i+1, 2, \ldots, N$.

Let us define $Q_1 = ((N-1)(N-2))^{-1}$ and $Q_2 = N-1$. The star resistances of the graphs g and h are given by the equation (15), so

$$r^{s(g)}_{k} = Q_1 \left(Q_2 \sum_{w=1}^{N} r^{g}_{eq_{kw}} - S^{g}_{t} \right) \text{ for } k = 1, 2, \ldots, N \tag{38}$$

$$r^{s(h)}_{l} = Q_1 \left(Q_2 \sum_{w=1}^{N} r^{h}_{eq_{lw}} - S^{h}_{t} \right) \text{ for } l = 1, 2, \ldots, N \tag{39}$$

In addition, $S^{g}_{t} = S^{h}_{t}$, because the graphs g and h are isomorphic, and they will be denoted by S_t. Then, the expressions (38) and (39) can be rewritten as

$$r^{s(g)}_{k} = Q_1 \left(Q_2 \sum_{w=1}^{N} r^{g}_{eq_{kw}} - S_t \right) \text{ for } k = 1, 2, \ldots, N \tag{40}$$

$$r^{s(h)}_{l} = Q_1 \left(Q_2 \sum_{w=1}^{N} r^{h}_{eq_{lw}} - S_t \right) \text{ for } l = 1, 2, \ldots, N \tag{41}$$

Using (37) we can rewrite the equation (41) as a function of the equivalent resistance of the graph g, i.e.

$$r_l^h = Q_1 \left(Q_2 \sum_{w=1}^{N} r^g_{eq_{\varphi^{-1}(l),w}} - S_t \right) \text{ for } l = 1, 2, \ldots, N \tag{42}$$

By making the change of variable $l = \varphi(k)$ in equation 42 we obtain,

$$r^h_{\varphi(k)} = Q_1 \left(Q_2 \sum_{w=1}^{N} r^g_{eq_{kw}} - S_t \right) \tag{43}$$

for $k = \varphi^{-1}(1), \varphi^{-1}(2), \ldots, \varphi^{-1}(N)$. However, the order in which the N equations of the above expression (43) are obtained is irrelevant, and this expression can be rewritten as

$$r^h_{\varphi(k)} = Q_1 \left(Q_2 \sum_{w=1}^{N} r^g_{eq_{kw}} - S_t \right) \text{ for } k = 1, 2, \ldots, N \tag{44}$$

Then we can realize that the rights sides of the equations (40) and (44) are identical, so we obtain that

$$r^g_k = r^h_{\varphi(k)} \text{ for } k = 1, 2, \ldots, N \tag{45}$$

This is an important conclusion, because it shows that when two graphs are isomorphic, we can do the mapping of the stars of both graphs in the same way that we do the mapping of the graphs (remember that we can go backwards from the stars mapping to the mapping of original graph). We use this result for doing the mapping between star resistances in the second filtering phase.

3.2. Time complexity of the complete method

We are going analyze the time complexity of the two phases:

- *First filter phase: Obtaining CEM and the equivalent resistances* We have seen that this phase requires to do the following steps for both graphs g and h:
  - Apply the step function to the graph edges and obtain $(A')^g$ and $(A')^h$.

- Obtain Y^g from $(A')^g$ and Y^h from $(A')^h$.
- Obtain X_m^g from Y^g and X_m^h from Y^h.
- Compute $(X_m^g)^{-1}$ and $(X_m^h)^{-1}$.
- Compute de equivalents resistances from $(X_m^g)^{-1}$ and $(X_m^h)^{-1}$ using Table 5 and obtain $\vec{R}_{eq}^g$ and $\vec{R}_{eq}^h$.
- Obtain $\widehat{R}_{eq}^g$ and $\widehat{R}_{eq}^h$.
- Check if $\widehat{R}_{eq}^g$ and $\widehat{R}_{eq}^h$ are equal.
- If is true then continue with the next phase, otherwise STOP (the graphs are not isomorphic).

Each one of these operations require at most $O(N^2)$, except the inverse of X_m that requires $O(N^3)$, then the time complexity will be $O(N^3)$.

- *Second filter phase: Approximation of the equivalent resistances by the star circuit and validation process* The operations are the following:

  - Compute the star resistances using (15). This operation is $O(N^2)$.
  - Obtain the sets $\widehat{R}^{s(g)}$ and $\widehat{R}^{s(h)}$ and check if they are equal.
  - If this is true then continue, otherwise STOP (the graphs are not isomorphic).
  - Compute the sequences $\widetilde{R}^{s(g)}$ and $\widetilde{R}^{s(g)}$ and look for the mapping using equation (22) or alternatively (23). This operation is $O(N^2)$.
  - Do the validation process. If all the r_k have different values among them, then the validation process will have a $O(N^2)$ time complexity. Otherwise, if there are some values of r_k that are identical, then in the worst case we have to validate all possible combinations, and the time complexity will be exponential with respect to the maximum number of repetitions in the star resistances.

  This phase can be $O(N^2)$ if the r_k are different in each star or if there is not mapping, or exponential with respect to the number of repetitions of resistances in each star in the worst case.

In conclusion, the filtering phase will eliminate most of the non-isomorphic graphs and it will detect when two graphs are isomorphic in $O(N^3)$, except when there were repeated values in the star resistances. In this last case, the time complexity can be in the worst case exponential with respect to the number of repetitions in the star resistances. However, we can get the complete (or partial in case of repetitions) node mapping in $O(N^3)$. The partial mapping is obtained of all the nodes that have been assigned through the validation process, except the ones that are repeated and the nodes that have edges with the repeated ones.

3.3. Co-resistances graphs

As it has been shown before, two isomorphic graphs should have the same set of equivalent resistances, but the reverse is not true, there can be two graphs with the same equivalent resistances that are not isomorphic, and those will be the "co-resistance" graphs. We will use the following definition.

Definition 3. *Two non-isomorphic graphs g and h are co-resistance when $\widehat{R}^g_{eq} = \widehat{R}^h_{eq}$.*

We explained in subsection "First filter" that for a graph g of order N, there could be at most $[N(N-1)/2]! - N!$ co-resistance graphs, although most of them will not meet the conditions for been a graph (for example, they have negative weights).

3.4. Example 3

Let us going to show an example where a pair of graphs are co-resistances. Let be graphs g and h:

$$A^g = \frac{1}{556}\begin{pmatrix} 0\,1\,2\,3 \\ 1\,0\,4\,5 \\ 2\,4\,0\,6 \\ 3\,5\,6\,0 \end{pmatrix} \text{ and } A^h = \frac{1}{75268}\begin{pmatrix} 0 & 39 & 167 & 461 \\ 39 & 0 & 233 & 723 \\ 167 & 233 & 0 & 201 \\ 461 & 723 & 201 & 0 \end{pmatrix}$$

Taking a visual inspection of these matrices, we can see that they are not isomorphic (not even match the sets of weights). However, the equivalent resistances

$$R^g_{eq} = \begin{pmatrix} 0 & 132 & 115 & 104 \\ 132 & 0 & 75 & 68 \\ 115 & 75 & 0 & 59 \\ 104 & 68 & 59 & 0 \end{pmatrix}$$

and

$$R^h_{eq} = \begin{pmatrix} 0 & 132 & 115 & 104 \\ 132 & 0 & 59 & 68 \\ 115 & 59 & 0 & 75 \\ 104 & 68 & 75 & 0 \end{pmatrix}$$

are identical ($\widehat{R}^g_{eq} = \widehat{R}^h_{eq}$), then the two graphs are co-resistance. If we continue with the method, we will realize that star resistances are

$$\begin{aligned} \vec{R}^{s(g)} &= \tfrac{1}{3}\,(250, 136, 97, 70)^t \\ \vec{R}^{s(h)} &= \tfrac{1}{3}\,(250, 112, 97, 94)^t \end{aligned}$$

that means that $\widehat{R}^{s(g)}$ and $\widehat{R}^{s(h)}$ sets do not match and at this point it would be detected that both graphs are not isomorphic and and filtering ends here.

4. Experiments

Although we have proved that the method works from the theoretical point of view, we have included this section to show the behavior of the method in different well known databases. First we wanted to know the constant of the cubic time complexity, and we found that it is a low constant, 10^{-6}, independent of the computer power (the time has been normalized). Second we wanted to find co-resistances, false positives and repetitions in the star, that are predicted by the theory. However, we did not find these issues in the two databases although they have a big number of graphs and some of the graphs has large number of nodes.

4.1. Corroboration of the time complexity

In order to confirm that the time complexity of the SM is $(O(N^3))$,[11] we have proceed to apply the method in a graph of order N up to the star resistance computation. This process was repeated from $N = 90$ to 2110 in increments of one by one. The processing time, T_N (s), was normalized to the duration of the $N = 90$ ($\tau_N = T_N/T_{90}$) to obtain the plot of Figure 5.

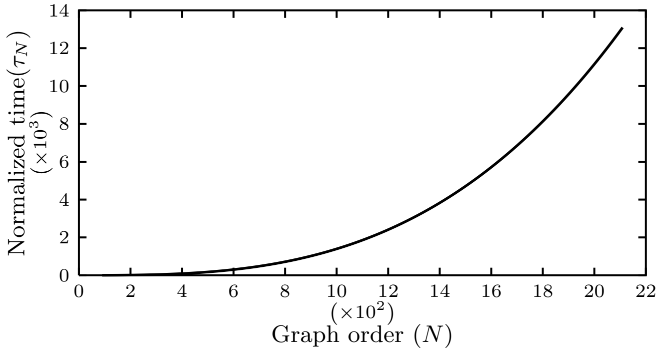


Figure 5: Normalized duration (τ_N) versus the order of the graph (N) with the SM. The time complexity is polynomial of order three ($O(N^3)$).

From these 2019 values were extracted the regression curve, this is

$$\tau_N = 1.36242934289419 \cdot 10^{-6} N^{3.0040699854}$$

corroborating the previously predicted time complexity.

4.2. "Letter" database

In the "Letter" database [45], the nodes have two numeric labels corresponding to the cartesian location of the node on a plane. Because our method works for graphs with weighted edges, we eliminate the coordinates of the nodes and put as the edge weight, the euclidean distance between the two nodes. During this process, those graphs with more than one connected component were dis-

[11] The processing time for the SM is deterministic and only depends on the number of nodes (not on the number of branches or the weights assigned)

carded. A total of 1708 graphs were considered. All the graphs are different (no two are alike even isomorphic).

In order to have isomorphic graphs, we generate some isomorphic graphs, “Isomorphic generated”,and the total of them per order of the graph is shown in Table 2. We tested 23 389 graphs.

		For each isomorphic graph			Couple of graphs		
N	Not isomorphic	generated	maximum	total	isomorphic	not isomorphic	total
3	167	6	6	1 002	2 505	498 996	501 501
4	411	24	24	9 864	113 436	48 530 880	48 644 316
5	678	10	120	6 708	30 510	22 950 300	22 980 810
6	337	13	720	4 381	26 286	9 568 104	9 594 390
7	106	11	5 040	1 166	5 830	673 365	679 195
8	8	11	40 320	88	440	3 388	3 828
9	1	12	362 880	12	12	0	12

Table 2: For each order and each isomorphic graph, the isomorphic graphs are indicated under the item “Isomorphic generated”. N is the order of the graph

We apply the SM method for all the graphs with the same order (see Table 2) and the total pairwise comparisons was 82 404 106. The Table 3 shows number of nodes per graph, the total number of isomorphic and non-isomorphic graphs taken into account (the ground-truth) and the number of isomorphic and non-isomorphic graphs detected by the method. Moreover we include in the table, the efficiency of the method (η) defined as

$$\eta = 1 - \frac{\text{pairs of graphs with partial mapping}}{\text{all pairs of graphs}}$$

In all cases the method worked, and non co-resistance graphs were detected. Nor even, there were found repeated values in the branches of the star resistances.

	Ground-truth		Filter output			
N	couple isomorphic	couple not isomorphic	couple isomorphic	couple not isomorphic	partial mapping	η (%)
3	2 505	498 996	2 505	498 996	0	100
4	113 436	48 530 880	113 436	48 530 880	0	100
5	30 510	22 950 300	30 510	22 950 300	0	100
6	26 286	9 568 104	26 286	9 568 104	0	100
7	5 830	673 365	5 830	673 365	0	100
8	440	3 388	440	3 388	0	100
9	12	0	12	0	0	100

Table 3: Results of the filtering applied to “Letter” database. N is the order of the graph

4.3. “Web” database

The “Web” database [45] contains 2340 directed graphs, with multiple edges, some of them not connected. The minimum order is 43 and the maximum 834. The number of graphs in the database of order between 43 and 834 varies, could be $0, 1, 2, \ldots$ or 25. The order of the graph is very sparse,[12] no two graphs are identical, neither isomorphic. The graphs were modified, in order that all nodes were connected and they were transformed in undirect graphs. During the process, those graphs with a single representative for a given order were discarded. At the end of this process, 2239 graphs were obtained with a minimum order of 57 and maximum of 635. For a given order, the number of representatives was $2, 3, \ldots$ or 25 graphs. For each graph, three isomorphic graphs were generated (four isomorphic graphs if the original graph is taken into account) and the total number of graphs, isomorphic or not, was 8956.

For a given order, the number of total graph pairs was 189 626, where 176 192 were ground-truth pairs of non-isomorphic graphs and 13 434 were ground-truth isomorphic graphs. All these pairs of graphs were checked using the SM method, and the output of the filter process can be seen in Table 4.

In all cases, the SM method did not detect any co-resistance graphs. Neither

[12] This is the reason that we do not attach a table with a breakdown by order of graphs as was done in the “Letter” database .

	Ground-truth		Filter output			
N	couple isomorphic	couple not isomorphic	couple isomorphic	couple not isomorphic	partial mapping	η (%)
57-99	1 884	24 992	1884	24 992	0	100
100-199	7 392	120 816	7392	120 816	0	100
200-299	3 252	27 952	3252	27 952	0	100
300-399	762	2 240	762	2 240	0	100
400-499	120	160	120	160	0	100
500-599	12	16	12	16	0	100
600-635	12	16	12	16	0	100

Table 4: Results of the filtering applied to "Web" database. N is the order of the graph. Because dispersion order graph couple are grouped in ranges.

there were found repetitions in the branches of the star resistances.

In both experiments, the SM method detected all the isomorphic graphs and rejected all the non isomorphic graphs.

5. Conclusions

We have presented a new method (SM) for filtering non-isomorphic graphs based on the CEM, detecting isomorphic graphs and in this case, obtaining the complete or partial node mapping between both graphs. The time complexity for detecting non-isomorphic graphs of ordern N is in most of the cases $O(N^3)$. The time complexity for detecting graph isomorphism if the values of the star resistances are different, is $O(N^3)$, but in case that there exist repeated star resistance values, the worst time complexity can be exponential with respect to the number of repeated star values (this number is much less that the graph N order). The method can extract the complete or the partial node mapping, depending on the restrictions on the time complexity.

The method has been validated using two well know databases, the "Letter" and "Web" databases.

The method has some issues that should be highlighted:

1. The detection and mapping of a graph isomorphism (excluding co-resistances) are clearly separated, which allows doing the filter process in two phases.

2. The filter performs an early detection of non-isomorphic graphs.
3. If there are repeated values in the star resistances, but there is a graph isomorphism, at least partial extraction of the node mapping can be done in $O(N^3)$.
4. The filtering process is not probabilistic, not iterative neither recursive, so the computation complexity is deterministic and only depends on N.
5. We do no need to calculate the pseudoinverse, we can compute the star resistance values by a sum of finite number of terms.
6. If we have to compare repeatedly unknown graphs with respect to a graph database preset beforehand, then we can pre-compute and store in the computer memory, the equivalent and star resistances of all graphs of the database.
7. The weakness of this method is on the matrix inverse computation of the first filter for very big graphs, and this can affect the comparison between $\widehat{R}^g_{eq}$ and $\widehat{R}^h_{eq}$. This can be solved partially in different ways, for example increasing the computer numerical resolution and/or allowing a tolerance error to match both equivalent resistances.

A. Equivalent resistance and reference node

First of all the formal definition of equivalent resistance is as follow.

Definition 4. *The equivalent resistance between nodes i and j ($r_{eq_{ij}}$) is the quotient (Ohm's law) between the voltage of the node i referred to node j and the current absorbed by the circuit when an independent current source I_s is connected from node j to node i. Applying the Ohm's law, the equivalent resistance is:*

$$r_{eq_{ij}} = \frac{V_{ij}}{I_s} \quad (\Omega) \tag{46}$$

In order to compute (46) we will use the node analysis method.[13] This

[13] There are two methods (one is the dual of the other) for the systematic analysis of circuits:

method requires to fix an arbitrary reference node m,[14] eliminate it from the matrix (3) and renumbering the rest of the nodes (the numbering of the nodes will change). Hence the equation (46) can be rewritten as,

$$r_{eq_{ij}} = \frac{V_{ij}}{I_s} = \frac{V_{pq}}{I_s} \tag{47}$$

In order to preserve the order of the renumbering nodes and being able to recover the original node numbering once we apply the method backwards (to recover the original graph), we will do the following node renumbering assignment. In the forward node renumbering assignment (once we fix the reference node), the new node p will be renumber as

$$p = \begin{cases} i; & 1 \leq i < m \\ i-1; & m < i \leq N \end{cases} \tag{48}$$

When we go backwards, that means we want to recover the node renumbering of the original graph, then we will do the following node renumbering:

$$i = \begin{cases} m; & \text{for reference node} \\ p; & 1 \leq p < m \\ p+1; & m \leq p \leq N-1 \end{cases} \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} \text{otherwise} \tag{49}$$

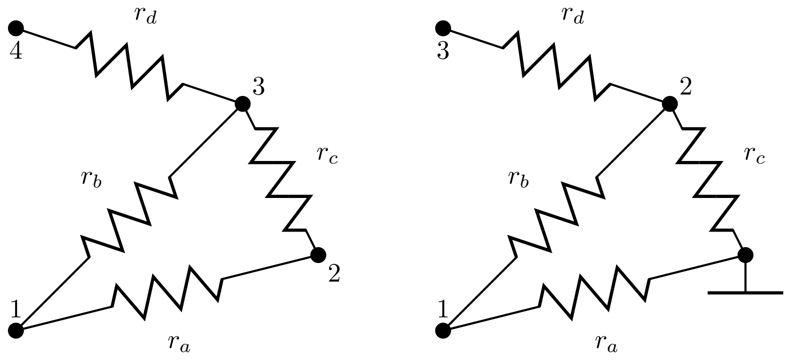
It is clear that once this transformation is done, the number of the reference node, m, is saved. A simple example of the above can be seen in Figure 6.

Now, we can again rewrite the equation (47) as follows:

$$r_{eq_{ij}} = \frac{V_{pq}}{I_s} = \frac{V_p - V_q}{I_s} \tag{50}$$

mesh method and node method. In principle, it can be used any of them interchangeably, but the method of meshes suffers from a strong constraint: it can only be used on planar circuits. For this reason we will use the node method.

[14] Theoretically, the results do not depend on the choice of reference node, but in practice (when digital computers are used) the appropriate selection of the reference node can minimize the rounding errors, usually it is selected the reference node that has more connections.



(a) Electrical circuit before selecting the reference node ($m = 2$). Note how are remunerated nodes 3 and 4 in Figure 6b.

(b) The same electrical circuit of Figure 6a after selecting the reference node (node 2 of the Figure 6a).

Figure 6: The same electrical circuit before and after selecting an arbitrary reference node (for example $m = 2$).

where p and q are the new number of the nodes i and j respectively. In order to calculate (50), we have to calculate V_p and V_q using $\vec{I} = X_m\vec{V}$, then the $\vec{V}$ is

$$\vec{V} = X_m^{-1}\vec{I} \tag{51}$$

where the X_m matrix, square of order $N - 1$, is called Definite Admittance Matrix (DAM). DAM is computed (4th line of block A of Figure 1) by removing the row m and column m of the matrix Y (eq. (3)). The vector $\vec{V}$ represents the voltages of the nodes referred to the reference node ($\vec{V}$ are the unknowns). The vector $\vec{I}$ is the vector of the electrical currents accessing to each node and they can be positive (when it is an incoming electrical current) or negative (when it is an outgoing electrical current). $\vec{I}$ is the data vector. It is known that due X_m is a DAM, it will always be invertible (5th line of block A of Figure 1). In what follows we will represent the matrix X_m^{-1} as:

$$X_m^{-1}=\begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1,N-1} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{N-1,1} & \alpha_{N-1,2} & \cdots & \alpha_{N-1,N-1} \end{pmatrix} \tag{52}$$

The matrix X_m^{-1} is symmetric ($\alpha_{ji} = \alpha_{ij}$), due that represents a circuit with only resistors.

Let us now compute the equivalent resistances (6th line of block A of Figure 1) between nodes i and j, once we have fixed node m. In order to abbreviate the expressions we will use $Z = N - 1$.

(i) For the case $i < j < m$ (this implies that $p = i$ and $q = j$ by (49)) we have from equation (51) and expression (52) that

$$\begin{pmatrix} V_1 \\ \vdots \\ V_p \\ \vdots \\ V_q \\ \vdots \\ V_Z \end{pmatrix} = \begin{pmatrix} \alpha_{11} & \cdots \alpha_{1p} & \cdots \alpha_{1q} & \cdots \alpha_{1Z} \\ \vdots & \ddots \vdots & \ddots \vdots & \ddots \vdots \\ \alpha_{p1} & \cdots \alpha_{pp} & \cdots \alpha_{pq} & \cdots \alpha_{pZ} \\ \vdots & \ddots \vdots & \ddots \vdots & \ddots \vdots \\ \alpha_{q1} & \cdots \alpha_{qp} & \cdots \alpha_{qq} & \cdots \alpha_{qZ} \\ \vdots & \ddots \vdots & \ddots \vdots & \ddots \vdots \\ \alpha_{Z1} & \cdots \alpha_{Zp} & \cdots \alpha_{Zq} & \cdots \alpha_{ZZ} \end{pmatrix} \begin{pmatrix} 0 \\ \vdots \\ I_s \\ \vdots \\ -I_s \\ \vdots \\ 0 \end{pmatrix} \tag{53}$$

from where we obtain

$$V_p = I_s(\alpha_{pp} - \alpha_{pq}) \tag{54}$$

$$V_q = I_s(-\alpha_{qq} + \alpha_{qp}) \tag{55}$$

Incorporating the above results in the formula (50) we obtain

$$r_{eq_{ij}} = \frac{V_p - V_q}{I_s} = \frac{I_s(\alpha_{pp} - \alpha_{pq} - \alpha_{qp} + \alpha_{qq})}{I_s} \tag{56}$$

and due that $\alpha_{qp} = \alpha_{pq}$ then $r_{eq_{ij}}$ can be rewritten as

$$r_{eq_{ij}} = \alpha_{pq} - 2\alpha_{pq} + \alpha_{pq} \tag{57}$$

and using (48) we finally obtain

$$r_{eq_{ij}} = \alpha_{ii} - 2\alpha_{ij} + \alpha_{jj} \tag{58}$$

(ii) For the case $i < j = m$ (this implies that $p = i$ by (49) and j is the reference node) we have from equation (51) and expression (52) that

$$\begin{pmatrix} V_1 \\ \vdots \\ V_p \\ \vdots \\ V_Z \end{pmatrix} = \begin{pmatrix} \alpha_{11} & \cdots & \alpha_{1p} & \cdots & \alpha_{1Z} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \alpha_{p1} & \cdots & \alpha_{pp} & \cdots & \alpha_{pZ} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \alpha_{Z1} & \cdots & \alpha_{Zp} & \cdots & \alpha_{ZZ} \end{pmatrix} \begin{pmatrix} 0 \\ \vdots \\ I_s \\ \vdots \\ 0 \end{pmatrix} \tag{59}$$

As V_q is zero for being the reference node then

$$r_{eq_{ij}} = \frac{V_p - V_q}{I_s} = \frac{I_s \alpha_{pp}}{I_s} = \alpha_{pp} \tag{60}$$

and using (48) we finally obtain

$$r_{eq_{ij}} = \alpha_{ii} \tag{61}$$

(iii) For the case $i < m < j$ (this implies that $p = i$ and $q = j - 1$ using (49)) and applying the same expressions as before, we obtain

$$r_{eq_{ij}} = \alpha_{ii} - 2\alpha_{i,j-1} + \alpha_{j-1,j-1} \tag{62}$$

(iv) For the case $i = m < j$ (this implies that $q = j - 1$ using (49) and i is de reference node) we have from the equation (51) and the expression (52) that

$$\begin{pmatrix} V_1 \\ \vdots \\ V_q \\ \vdots \\ V_Z \end{pmatrix} = \begin{pmatrix} \alpha_{11} & \cdots & \alpha_{1q} & \cdots & \alpha_{1Z} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \alpha_{q1} & \cdots & \alpha_{qq} & \cdots & \alpha_{qZ} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \alpha_{Z1} & \cdots & \alpha_{Zq} & \cdots & \alpha_{ZZ} \end{pmatrix} \begin{pmatrix} 0 \\ \vdots \\ -I_s \\ \vdots \\ 0 \end{pmatrix} \tag{63}$$

As V_p is zero for being the reference node, then

$$r_{eq_{ij}} = \frac{V_p - V_q}{I_s} = \frac{-(-I_s \alpha_{qq})}{I_s} = \alpha_{qq} \tag{64}$$

and using (48), we finally obtain

$$r_{eq_{ij}} = \alpha_{j-1,j-1} \tag{65}$$

(v) And finally for the case $m < i < j$ (this implies that $p = i - 1$ and $q = j - 1$ using (49)), we can apply the same expressions from (53) to (57), but using (48), we obtain

$$r_{eq_{ij}} = \alpha_{i-1,i-1} - 2\alpha_{i-1,j-1} + \alpha_{j-1,j-1} \tag{66}$$

All the above results (formulas (58), (61), (62), (65) and (66)) are summarized in the Table 5 in compact format.

Item	Case	Calculation of $r_{eq_{i,j}}$
(i)	$i < j < m$	$\alpha_{ii} - 2\alpha_{ij} + \alpha_{jj}$
(ii)	$i < j = m$	α_{ii}
(iii)	$i < m < j$	$\alpha_{ii} - 2\alpha_{i,j-1} + \alpha_{j-1,j-1}$
(iv)	$i = m < j$	$\alpha_{j-1,j-1}$
(v)	$m < i < j$	$\alpha_{i-1,i-1} - 2\alpha_{i-1,j-1} + \alpha_{j-1,j-1}$

Table 5: Summary of the resulting equations of the equivalent resistances using the node numbering of the original graph (before renumbering the nodes due to the selection of the reference node).

B. Efficient computation of the equivalent resistances

As already it is seen computing the r_k values of the star ($\vec{R}$) can be obtained by formula (12); or alternatively, by successively applying the formulas (13) and (14). This result shows the general way of getting $\vec{R}$, but formula (12) requires computing a matrix inverse, three products of matrices and a transpose matrix, consequently it is computationally expensive. Instead of doing these operations we can compute r_k using a straightforward formula and the time complexity can be reduced from $O(N^3)$ to $O(N^2)$ in this step. We will obtain the straightforward formula using the results presented in [44] and the following derivation. We shown in [44] that K^+ can be computed as follows:

$$K^+ = \frac{1}{(N-1)(N-2)} \left((N-1)K^t - \mathbb{1}_{N,N(N-1)/2}\right) \tag{67}$$

where $\mathbb{1}_{m,n}$ is a matrix of order $N \times N(N-1)/2$ which coefficients are all ones. If we take one element k_{ij} of matrix K, then the element k_{ij}^+ of the equation (67) can be rewritten as follows

$$k_{ij}^+ = \frac{1}{(N-1)(N-2)} \left((N-1)k_{ji} - 1\right) \tag{68}$$

for $i = 1, \dots, N$ and $j = 1, \dots, N(N-1)/2$. We can simplify even more the equation (68) by analyzing the MSE minimization procedure. It should be noted that for obtaining any value r_k of $\vec{R}$, we have to multiply the vector $\vec{R}_{eq}$ by the k row of matrix K^+; then, except for the constant $((N-1)(N-2))^{-1}$, which will be denoted as Q_1 in what follows, the equivalents resistances of $\vec{R}_{eq}$ will be $N-1$ values multiplied by minus unity (first case) and the rest $((N-1)(N-2)/2$ values) multiplied by $N-2$ (second case), allowing to extract the common factor of these constants. We can rewrite equation (12) using the equation (68), and taken into account that the equivalent resistances ($r_{eq_{ij}}$) for the first case are those that meet $i = k$ or $j = k$ and for the second case are those that meet $i \neq k$ and $j \neq k$. Then we can write

$$r_k = Q_1 \left((N-2) \sum_{\substack{j=1 \\ j \neq k}}^{N} r_{eq_{kj}} - \sum_{\substack{i=1 \\ i \neq k}}^{N-1} \left(\sum_{\substack{j=i+1 \\ j \neq k}}^{N} r_{eq_{ij}} \right) \right) \tag{69}$$

for $k = 1, 2, \dots, N$ and where we have to take into account that $r_{eq_{ij}} = r_{eq_{ji}}$. In the first summation of the expression (69), the variable j is different from k to avoid the sum of the term $r_{eq_{kk}}$, but this restriction may be obviated as this term is always zero, then we can rewrite the equation as

$$r_k = Q_1 \left((N-2) \sum_{j=1}^{N} r_{eq_{kj}} - \sum_{\substack{i=1 \\ i \neq k}}^{N-1} \left(\sum_{\substack{j=i+1 \\ j \neq k}}^{N} r_{eq_{ij}} \right) \right) \tag{70}$$

for $k = 1, 2, \dots, N$. In the expression (70) we denote by $\Psi(k)$ the first summation, by $\bar{\Psi}(k)$ the double summation and by Ψ_t the sum of all equivalent resistances,[15] this is

$$\Psi_t = \sum_{i=1}^{N-1} \left(\sum_{j=i+1}^{N} r_{eq_{ij}} \right) \tag{71}$$

then

$$\Psi_t = \Psi(k) + \bar{\Psi}(k) \text{ for } k = 1, 2, \ldots, N \tag{72}$$

Given the above definitions and the relationship given by (72) then (70) can be written by simple manipulations, as

$$r_k = \tfrac{1}{(N-1)(N-2)} \left((N-1)\Psi(k) - \Psi_t \right) \tag{73}$$

for $k = 1, 2, \ldots, N$. This last equation (73) coincides with the advanced in (15).

The improvement is substantial since for the calculation of the star resistances is not necessary to obtain the pseudoinverse and multiplying matrices, since the calculation is straightforward using the expression (15) and it is not necessary to use no (12), no (67) and no (68).

References

[1] H. Bunke, A. Sanfeliu, Syntactic and structural pattern recognition: theory and applications, Vol. 8 of Computer science series, World Scientific Publishing, 1990.

[2] A. Sanfeliu, R. Alquézar, J. Andrade, J. Climent, Serratosa, F., J. Vergés, Graph-based representations and techniques for image processing and image analysis, Pattern Recognition 35 (3) (2002) 639–650.

[3] D. Conte, P. Foggia, C. Sansone, M. Vento, Thirty years of graph matching in pattern recognition, International Journal of Pattern Recognition and Artificial Intelligence 18 (3) (2004) 265–298. doi:10.1142/S0218001404003228.

[15] This expression (71) was already advanced in (17).

[4] B. Gallagher, Matching structure and semantics: A survey on graph-based pattern matching, AAAI FS 6 (2006) 45–53.

[5] H. Bunke, Graph matching: Theoretical foundations, algorithms, and applications, in: International Conference on Vision Interface, 2000, pp. 82–84.

[6] H. Bunke, C. Irniger, M. Neuhaus, Graph matching–challenges and potential solutions, in: F. Roli, S. Vitulano (Eds.), Image Analysis and Processing – ICIAP 2005, Vol. 3617 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2005, pp. 1–10. doi:10.1007/11553595_1.

[7] V. Vento, A long trip in the charming world of graphs for pattern recognition, Pattern Recognition 48 (2) (2015) 291 – 301. doi:10.1016/j.patcog.2014.01.002.

[8] H. Bunke, M. Vento, Benchmarking of graph matching algorithms, in: Proceedings 2nd IAPR-TC15 Workshop on Graph-based Representations, Handhorf, 1999, pp. 109–114.

[9] P. Foggia, C. Sansone, M. Vento, A performance comparison of five algorithms for graph isomorphism, in: in Proceedings of the 3rd IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition, 2001, pp. 188–199.

[10] S. A. Cook, The P versus NP problem, in: Clay Mathematical Institute; The Millennium Prize Problem, 2000.

[11] B. D. McKay, A. Piperno, Practical graph isomorphism, II, Journal of Symbolic Computation 60 (0) (2014) 94 – 112. doi:10.1016/j.jsc.2013.09.003.

[12] S. A. Cook, The complexity of theorem-proving procedures, in: Proceedings of the Third Annual ACM Symposium on Theory of Computing, STOC '71, ACM, New York, NY, USA, 1971, pp. 151–158. doi:10.1145/800157.805047.

[13] M. R. Garey, D. S. Johnson, Computers and Intractability; A Guide to the Theory of NP-Completeness, W. H. Freeman & Co., New York, NY, USA, 1990.

[14] L. A. Levin, Universal sorting problems, Problems of Information Transmission 9 (1973) 265–266.

[15] R. M. Karp, Reducibility among combinatorial problems, 50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art (2009) 219.

[16] J. Hopcroft, R. Tarjan, A V^2 algorithm for determining isomorphism of planar graphs, Information Processing Letters 1 (1) (1971) 32–34. doi:10.1016/0020-0190(71)90019-6.

[17] J. Hopcroft, R. Tarjan, Isomorphism of planar graphs (working paper), in: R. E. Miller, J. W. Thatcher, J. D. Bohlinger (Eds.), Complexity of Computer Computations, The IBM Research Symposia Series, Springer US, 1972, pp. 131–152. doi:10.1007/978-1-4684-2001-2_13.

[18] J. E. Hopcroft, J.-K. Wong, Linear time algorithm for isomorphism of planar graphs (preliminary report), in: Proceedings of the Sixth Annual ACM Symposium on Theory of Computing, STOC '74, ACM, New York, NY, USA, 1974, pp. 172–184. doi:10.1145/800119.803896.

[19] J. P. Kukluk, L. B. Holder, D. J. Cook, Algorithm and experiments in testing planar graphs for isomorphism, Journal of Graph Algorithms and Applications 8 (3) (2004) 313–356.

[20] A. V. Aho, J. E. Hopcroft, D. J. Ullmann, The Design and Analysis of Computer Algorithms, 1st Edition, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1974.

[21] E. M. Luks, Isomorphism of graphs of bounded valence can be tested in polynomial time, Journal of Computer and System Sciences 25 (1) (1982) 42 – 65.

[22] G. S. Lueker, K. S. Booth, A linear time algorithm for deciding interval graph isomorphism, J. ACM 26 (2) (1979) 183–195. doi:10.1145/322123.322125.

[23] J. Spinrad, Recognition of circle graphs, Journal of Algorithms 16 (2) (1994) 264–282. doi:10.1006/jagm.1994.1012.

[24] W. Hsu, $O(M{\cdot}N)$ algorithms for the recognition and isomorphism problems on circular-arc graphs, SIAM Journal on Computing 24 (3) (1995) 411–439. doi:10.1137/S0097539793260726.

[25] G. Miller, Isomorphism testing for graphs of bounded genus, in: Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing, STOC '80, ACM, New York, NY, USA, 1980, pp. 225–235. doi:10.1145/800141.804670.

[26] I. S. Filotti, J. N. Mayer, A polynomial-time algorithm for determining the isomorphism of graphs of fixed genus, in: Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing, STOC '80, ACM, New York, NY, USA, 1980, pp. 236–243. doi:10.1145/800141.804671.

[27] M. Grohe, Isomorphism testing for embeddable graphs through definability, in: Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing, STOC '00, ACM, New York, NY, USA, 2000, pp. 63–72. doi:10.1145/335305.335313.

[28] L. Babai, D. Y. Grigoryev, D. M. Mount, Isomorphism of graphs with bounded eigenvalue multiplicity, in: Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing, STOC '82, ACM, New York, NY, USA, 1982, pp. 310–324. doi:10.1145/800070.802206.

[29] H. L. Bodlaender, Polynomial algorithms for graph isomorphism and chromatic index on partial k-trees, Journal of Algorithms 11 (4) (1990) 631 – 643. doi:10.1016/0196-6774(90)90013-5.

[30] J. R. Ullmann, An algorithm for subgraph isomorphism, Journal of the Association for Computing Machinery 23 (1) (1976) 31–42.

[31] D. C. Schmidt, L. E. Druffel, A fast backtracking algorithm to test directed graphs for isomorphism using distance matrices, J. ACM 23 (3) (1976) 433–445. doi:10.1145/321958.321963.

[32] L. P. Cordella, P. Foggia, C. Sansone, M. Vento, Performance evaluation of the VF graph matching algorithm, in: Image Analysis and Processing, 1999. Proceedings. International Conference on, 1999, pp. 1172–1177. doi:10.1109/ICIAP.1999.797762.

[33] L. P. Cordella, P. Foggia, C. Sansone, M. Vento, An improved algorithm for matching large graphs, in: 3rd IAPR-TC15 workshop on graph-based representations in pattern recognition, 2001, pp. 149–159.

[34] B. D. McKay, Computing automorphisms and canonical labellings of graphs, in: D. A. Holton, J. Seberry (Eds.), Combinatorial Mathematics, Vol. 686 of Lecture Notes in Mathematics, Springer Berlin Heidelberg, 1978, pp. 223–232. doi:10.1007/BFb0062536.

[35] A. Piperno, Search space contraction in canonical labeling of graphs (preliminary version), CoRR abs/0804.4881.

[36] B. D. McKay, Practical graph isomorphism, Congressus Numerantium 30 (1981) 45–87.

[37] P. T. Darga, M. H. Liffiton, K. A. Sakallah, I. L. Markov, Exploiting structure in symmetry detection for CNF, in: Proceedings of the 41st Annual Design Automation Conference, DAC '04, ACM, New York, NY, USA, 2004, pp. 530–534. doi:10.1145/996566.996712.

[38] T. Junttila, P. Kaski, Engineering an efficient canonical labeling tool for large and sparse graphs, in: D. Applegate, G. S. Brodal, D. Panario, R. Sedgewick (Eds.), Proceedings of the Ninth Workshop on Algorithm

Engineeringand Experiments and the Fourth Workshop on Analytic Algorithms and Combinatorics, SIAM, 2007, pp. 135–149.

[39] T. Junttila, P. Kaski, Conflict propagation and component recursion for canonical labeling, in: A. Marchetti-Spaccamela, M. Segal (Eds.), Theory and Practice of Algorithms in (Computer) Systems, Vol. 6595 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2011, pp. 151–162. doi:10.1007/978-3-642-19754-3_16.

[40] J. L. López-Presa, A. Fernández Anta, Fast algorithm for graph isomorphism testing, in: J. Vahrenhold (Ed.), Experimental Algorithms, Vol. 5526 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 221–232. doi:10.1007/978-3-642-02011-7_21.

[41] M. Igelmo, A. Sanfeliu, M. Ferrer, A Conductance Electrical Model for representing and matching weighted undirected graphs, in: Proceedings of the 2010 20th International Conference on Pattern Recognition, ICPR '10, IEEE Computer Society, Washington, DC, USA, 2010, pp. 958–961. doi:10.1109/ICPR.2010.240.

[42] R. A. DeCarlo, P. Lin, Linear Circuit Analysis, Prentice Hall, 1995.

[43] D. J. Klein, M. Randić, Resistance distance, Journal of Mathematical Chemistry 12 (1) (1993) 81–95. doi:10.1007/BF01164627.

[44] M. Igelmo, A. Sanfeliu, Compact form of the pseudo–inverse matrix in the approximation of a star graph using the Conductance Electrical Model (CEM), in: G. Gimel'farb, E. Hancock, A. Imiya, A. Kuijper, M. Kudo, S. Omachi, T. Windeatt, K. Yamada (Eds.), Structural, Syntactic, and Statistical Pattern Recognition, Vol. 7626 of Lecture Notes in Computer Science, Springer, Berlin Heidelberg, 2012, pp. 539–547. doi:10.1007/978-3-642-34166-3_59.

[45] K. Riesen, H. Bunke, IAM graph database repository for graph based pattern recognition and machine learning, in: N. Vitoria Lobo, T. Kasparis,

F. Roli, J. T. Kwok, M. Georgiopoulos, G. C. Anagnostopoulos, M. Loog (Eds.), Structural, Syntactic, and Statistical Pattern Recognition, Vol. 5342 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2008, pp. 287–297. doi:10.1007/978-3-540-89689-0_33.