

## Vertex labeling and routing in self-similar outerplanar unclustered graphs modeling complex networks

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

2009 J. Phys. A: Math. Theor. 42 425001

(<http://iopscience.iop.org/1751-8121/42/42/425001>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 171.66.16.155

The article was downloaded on 03/06/2010 at 08:13

Please note that [terms and conditions apply](#).

# Vertex labeling and routing in self-similar outerplanar unclustered graphs modeling complex networks

Francesc Comellas and Alicia Miralles

Departament de Matemàtica Aplicada IV, EPSC, Universitat Politècnica de Catalunya,  
c/ Esteve Terradas 5, 08860 Castelldefels, Barcelona, Catalonia, Spain

E-mail: [comellas@ma4.upc.edu](mailto:comellas@ma4.upc.edu) and [almirall@ma4.upc.edu](mailto:almirall@ma4.upc.edu)

Received 17 June 2009, in final form 31 August 2009

Published 30 September 2009

Online at [stacks.iop.org/JPhysA/42/425001](http://stacks.iop.org/JPhysA/42/425001)

## Abstract

This paper introduces a labeling and optimal routing algorithm for a family of modular, self-similar, small-world graphs with clustering zero. Many properties of this family are comparable to those of networks associated with technological and biological systems with low clustering, such as the power grid, some electronic circuits and protein networks. For these systems, the existence of models with an efficient routing protocol is of interest to design practical communication algorithms in relation to dynamical processes (including synchronization) and also to understand the underlying mechanisms that have shaped their particular structure.

PACS numbers: 02.10.Ox, 05.45.Df, 89.20.Ff, 89.75.Da, 89.75.–k

(Some figures in this article are in colour only in the electronic version)

## 1. Introduction

The study of networks associated with complex systems has led to the introduction of new mathematical models which describe their main properties by focusing on three main aspects: the networks have a relatively small average distance and diameter (comparable to that of random networks with a similar number of nodes and links); they present a relatively large local clustering (nodes have many mutual neighbors) and their degrees follow a power law. These properties have been considered a distinctive feature for many complex networks, such as the Internet, telephone networks, transportation systems, and some biological and social networks; see [1, 16]. However, relevant real-life networks which describe technological and biological systems, such as some electronic circuits and protein networks, do not match the last two characteristics, as they have a low clustering (related sometimes to geographical or topological restrictions) and their degree distributions do not follow a power law. These systems can be modeled with networks which have exponential distributions [2]. We recall that

even the original Watts and Strogatz small world network model [16] follows an exponential degree distribution. Very often, this distribution and clustering are related to modular systems, which can be associated with self-similar graphs [14, 15]. Finally, in some cases, these networks are also planar [12, 17–20].

In this paper, we provide an optimal labeling and routing for a family of small-world, self-similar, unclustered and outerplanar graphs (a planar graph is called outerplanar if it has an embedding where all vertices lie on the boundary of the exterior). A labeling and routing scheme which performs well for scale-free networks has been given in [4]. Our labeling and routing is optimal for the family of deterministic graphs considered here, which have exponential degree distributions. A deterministic approach has proven useful to complement and enhance more common probabilistic and simulation techniques. Deterministic models have the strong advantage that it is often possible to compute analytically many graph properties, which can then be compared with experimental data from real and simulated networks.

To make the paper self-contained we recall from [5] the relevant parameters of the graphs as models for complex systems (i.e. diameter, average distance, clustering, degree distribution and degree correlation). We continue the study of the graphs by introducing a vertex labeling which is optimal in length and allows us to find from the labels a shortest path, and an optimal routing, between any two vertices. Therefore, this family of graphs, with its labeling and routing protocol, is a good mathematical model for many real-life systems and provides a new tool to produce practical algorithms in relation to dynamical processes (synchronization, cover time, etc) for technological and biological networks and can help to understand underlying mechanisms associated with their topologies.

## 2. Generation and properties of the graph $M(t)$

In this section, we recall the definition and main properties of a family of modular, self-similar and outerplanar graphs  $M(t)$  which have the small-world property. These graphs were introduced by the authors in [5].  $M(t)$  is constructed iteratively by introducing, at each step  $t$ , new vertices and edges in a deterministic way. The main difference with most iterative constructions is that at each step, we introduce a subgraph structure instead of a single vertex.

In our iterative graph construction, a *generating edge* is any edge whose endvertices have been introduced at different iteration steps and the initial edge of the construction. All other edges are known as *passive edges*. A generating edge becomes passive after its use in the construction process.

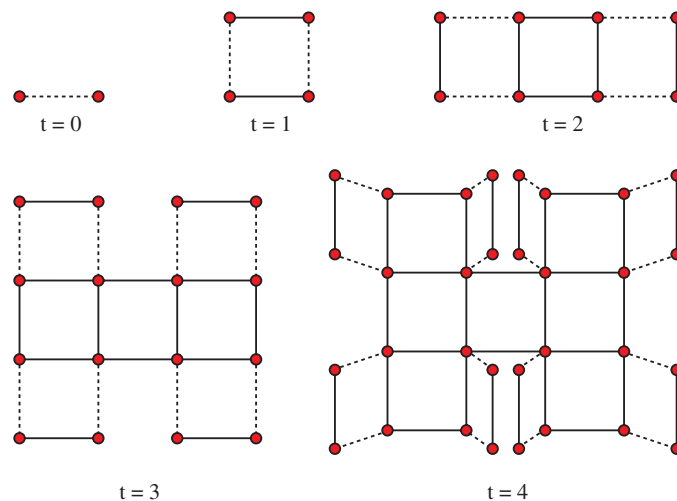
**Definition 2.1.** *The graph  $M(t)$ ,  $t \geq 0$ , is constructed as follows.*

*For  $t = 0$ ,  $M(0)$  has two vertices and a generating edge connecting them.*

*For  $t \geq 1$ ,  $M(t)$  is obtained from  $M(t-1)$  by adding to every generating edge in  $M(t-1)$  a path of length three,  $P_4$ , by identifying the two final vertices of the path with the endvertices of the generating edge.*

Therefore, at  $t = 0$ ,  $M(0)$  is  $K_2$ , at  $t = 1$  the graph is a cycle  $C_4$  with two generating edges and two passive edges, at  $t = 2$  is the ladder graph  $K_2 \square P_4$  and so on; see figure 1. From this figure we see that the graph  $M(t)$ ,  $t > 1$ , can also be constructed recursively from two copies of  $M(t-1)$ , by identifying, vertex to vertex, the initial edge of each  $M(t-1)$  with two opposite edges of  $C_4$ .

*Order and size of  $M(t)$ .* The order and size of  $M(t)$ ,  $t \geq 0$ , are  $|V(t)| = 2^{t+1}$  and  $|E(t)| = 3 \times 2^t - 2$ .



**Figure 1.** Graphs  $M(t)$  produced at iterations  $t = 0, 1, 2, 3$  and  $4$ . Active edges are represented with dashed lines.

*Planarity and outerplanarity.* A graph is planar if it can be drawn on the plane with no edges crossing. A planar graph is called outerplanar if it has an embedding in the plane such that the vertices are on a ring and the edges are drawn inside without any intersection. By construction,  $M(t)$  is outerplanar as the introduction at each iteration of a path connected to a generating edge, which becomes passive, adds two new vertices to the graph which can be placed at the exterior face with no crossing edges; see figure 1. The two new generating edges also lie in this exterior face keeping the outerplanarity of the graph. Outerplanarity is an important feature for its relation to the development of efficient algorithms [3].

*Modularity.* From the recursive construction process, we see that the graphs are modular as two copies of  $M(t - 1)$  are joined to form  $M(t)$  using only two new edges. It is possible to quantify this modularity with the function  $Q$  introduced by Newman and Girvan in [11] and an algorithm to find communities in a network by maximizing it; see [6, 8, 10, 11]. We have performed this analysis for  $M(t)$ ,  $3 \leq t \leq 16$ , with the specific Mathematica functions which generate optimal partitions with respect to  $Q$  and are ready to use in the standard distribution (version 7.0.1). The results are shown in figure 2, and we see that very quickly  $Q$  attains values near 1. For example, the maximum value of  $Q$  obtained for  $M(16)$  is 0.993499. Therefore this family of graphs is highly modular. In figure 3, we show the communities in  $M(5)$  which produce the maximum possible value  $Q = 0.726007$ .

Thanks to the deterministic nature of the graphs  $M(t)$ , we can give exact values for the relevant topological properties of this graph family, namely, degree distribution, degree correlations, diameter and average distance.

*Degree distribution.* The degree distribution of  $M(t)$  is found from the following facts: initially, at  $t = 0$ , the graph has two vertices of degree 1. When a new vertex  $i$  is added to the graph at iteration  $t_i$ , this vertex has degree 2 and it is connected to only one generating edge. From the construction process, every vertex of a generating edge, in the next step increases its degree in one unit and becomes the endvertex of a new generating edge. As the degree spectrum is discrete, we use the technique described by Newman in [9] to find the cumulative degree distribution, which is given next.

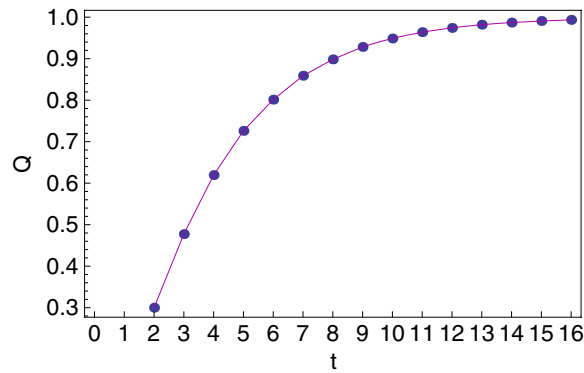


Figure 2. Values of the modularity  $Q$  for graphs  $M(t)$ ,  $t = 2, \dots, 16$ .

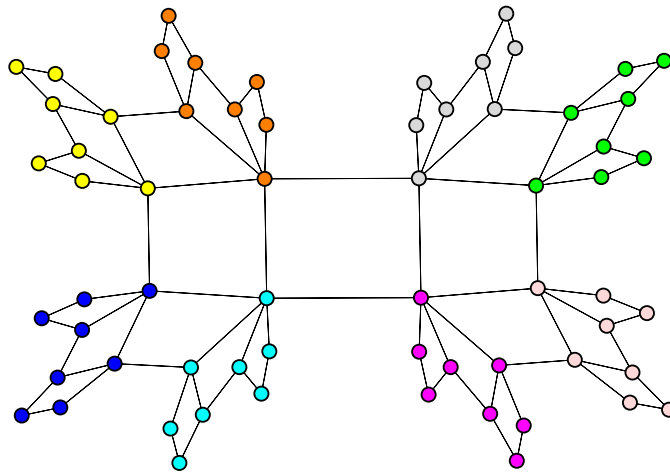


Figure 3. A decomposition in communities of  $M(5)$  which maximizes the modularity function  $Q$ .

**Proposition 2.2** [5]. *The cumulative degree distribution of the graph  $M(t)$  follows an exponential distribution  $P_{\text{cum}}(k) = 2^{2-k}$ .*

Some small-world networks, including the Watts–Strogatz model, have exponential degree distributions [2]. Research on networks associated with electronic circuits show that many of them are almost planar, modular and have a small clustering coefficient and in some cases their degree distributions can be associated with exponentials [7, 9].

*Correlation coefficient.* We have obtained the Pearson correlation coefficient,  $r(t)$ , for the degrees of the endvertices of the edges of  $M(t)$ .

**Proposition 2.3** [5]. *The correlation coefficient of the graph  $M(t)$  is*

$$r(t) = \frac{12 \cdot 2^t t^2 - 10 \cdot 2^t t + 2^{1+t} - 14 \cdot 4^t + 8t^2 + 28t + 12}{18 \cdot 2^t t^2 + 2^{1+t} t - 32 \cdot 4^t + 32 \cdot 2^t + 4 \cdot t^2 + 20t}.$$

Note that, for large values of  $t$ ,  $r(t) \sim \frac{7}{16} = 0.4375$ . Therefore this family of graphs has the degrees of the endvertices positively correlated and the graph is assortative. Social

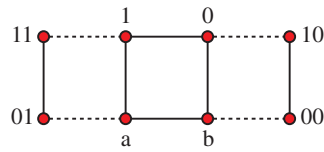


Figure 4. Labeling process at steps  $t = 0$ ,  $t = 1$  and  $t = 2$ .

networks are usually assortative as it is also the case of some technical networks such as the power grid and electronic circuits; see [9, 13].

*Diameter.* Computing the exact diameter of  $M(t)$  can be done analytically and gives the result shown below.

**Proposition 2.4 [5].** *The diameter of  $M(t)$  is  $D(t) = 2t$ ,  $t \geq 1$ .*

Therefore, as for  $t$  large,  $t \sim \ln|V_t|$  we have in this limit that  $D(t) \propto \ln|V_t|$ .

*Average distance.* The average distance of  $M(t)$  is defined as

$\bar{d}(t) = 2 \sum_{u,v \in V(t)} d(u, v) / (|V(t)|(|V(t)| - 1))$ , where  $d(u, v)$  is the distance between vertices  $u$  and  $v$ .

The modular recursive construction of  $M(t)$  allows us to obtain the exact value of  $\bar{d}(t)$ .

**Proposition 2.5 [5].** *The average distance of  $M(t)$  is  $\bar{d}(t) = \frac{2+2^t(2t-1)}{2^{t+1}-1}$ .*

Note that for a large iteration step  $\bar{d}(t) \simeq t \sim \ln|V_t|$ , which shows a logarithmic scaling of the average distance with the order of the graph. The diameter shows a similar behavior and thus the graph is small-world.

### 3. Labeling of $M(t)$

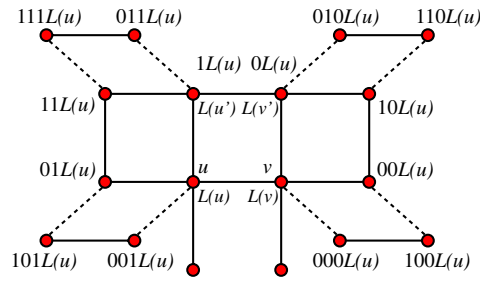
In this section, we describe a way to label the vertices of  $M(t)$  for  $t \geq 0$ , such that a routing by the shortest paths between any two vertices of  $M(t)$  can be deduced from the labels. Our labeling process is inspired by the relationship between  $M(t)$  and binary trees. Note that  $M(t)$  can be obtained from a rooted complete binary tree by replacing each of its vertices by a cycle of length 4, and identifying edges between two cycles which correspond to adjacent vertices in the binary tree.

In what follows, for any vertices  $u, v \in V(t)$ ,  $L(v)$  is the label of  $v$ ,  $xL(v)$  denotes the concatenation of digit  $x$  and  $L(v)$  and  $d(u, v)$  is the distance (shortest path) between  $u$  and  $v$ .

Note that as in each step  $M(t)$  is obtained from  $M(t - 1)$  by adding to every generating edge in  $M(t - 1)$  a path  $P_4$  of length three, the shortest path between any two existing vertices is not modified in the next iteration and therefore their distance is the same in  $M(t)$  as in  $M(t - 1)$ .

Our labeling protocol is based on assigning to any vertex  $v$ , introduced at step  $t \geq 1$ , a label of length  $t$  as a word of  $t$  binary digits 0 and 1. The initial vertices of step  $t = 0$  are given a special label. To be precise, the labeling of any vertex  $v$  in  $M(t)$  is performed according to the following rules.

- Label the two vertices of the initial graph  $M(0)$ , arbitrarily, with labels  $a$  and  $b$ , and label the two vertices introduced at step  $t = 1$  with 0 and 1; see figure 4.



**Figure 5.** Labeling process at steps  $t + 1$ ,  $t + 2$  and  $t + 3$  from a generating edge  $(u, v)$  where vertex  $u$  has been introduced at step  $t$  and vertex  $v$  was added in a former step.

- At any step  $t \geq 2$ , when two new vertices  $u'$  and  $v'$  are added and joined to the endvertices  $u$  and  $v$ , respectively, of a generating edge in  $M(t - 1)$ :
  - (i) If  $L(u) \notin \{a, b\}$  and  $L(v) \in \{a, b\}$  then  $L(u') = 1L(u)$  and  $L(v') = 0L(u)$ .
  - (ii) If  $L(u), L(v) \notin \{a, b\}$  and  $\text{length}(L(u)) > \text{length}(L(v))$  then  $L(u') = 1L(u)$  and  $L(v') = 0L(u)$ ; see figure 5.

We prove below that for  $t \geq 1$ , any vertex  $v$  introduced at step  $t$  has a unique label  $L(v) = s_t \dots s_2 s_1$  with length  $t$ , where each digit  $s_i$  is 0 or 1. We also note that since for any step  $t \geq 1$ , the number of vertices that are added to the graph  $M(t)$  is equal to  $2^t$ , this labeling is optimal in the sense that each label  $L(v)$  of a vertex introduced at this step  $t$  is a binary word of length  $t$ .

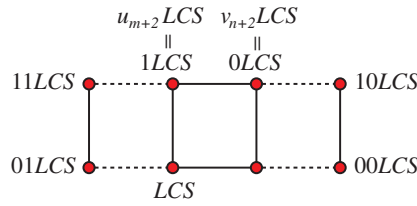
Next, we prove two properties about this labeling: lemma 3.1 ensures that our labeling is deterministic. Because of this, and to facilitate the reading of some proofs, we will refer, when needed, to vertices by their labels. Lemma 3.2 is used to prove proposition 4.1 and theorem 4.3 which show that our routing protocol, introduced in the following section, is valid and of the shortest paths.

**Lemma 3.1.** *Each vertex of  $M(t)$  has a unique label.*

**Proof.** By induction on  $t$ . When  $t$  is 0, 1 or 2, this is true by construction; see figure 4. Assume now that the lemma is true for  $t' \leq t - 1$ . Thus, each generating edge in  $M(t - 1)$  has one endvertex with a label of length  $t - 1$  while the other endvertex, which was introduced in a former step, has a shorter label. Therefore, from the labeling rules, the labels of the vertices introduced at step  $t - 1$  are those used to produce the labels of the vertices introduced at step  $t$ : a vertex introduced at step  $t - 1$ ,  $v_{t-1}$ , with label  $L(v_{t-1})$ , generates two new different labels,  $0L(v_{t-1})$  and  $1L(v_{t-1})$ . As all the labels in  $M(t - 1)$  are different, this is also true for  $M(t)$ . □

**Lemma 3.2.** *Let  $u$  be a vertex of  $M(t)$  introduced at step  $t \geq 1$ . Then  $d(\mathbf{0}^{k-1} \mathbf{0} 1 L(u), u) = 1$  and  $d(\mathbf{0}^k \mathbf{0} 0 L(u), u) = 2, k \geq 0$ .*

**Proof.** We prove the first equality by induction on  $k$ . For  $k = 0$  and  $k = 1$  it is true by construction. Note now that if vertex  $1L(u)$  is a neighbor of  $u$  introduced at step  $\tau$ , then vertices  $01L(u), 001L(u), \dots$  are also neighbors of  $u$  introduced, respectively, at steps  $\tau + 1, \tau + 2, \dots$  (see figure 5). Therefore, by induction, if  $\mathbf{0}^{k-1} \mathbf{0} 1 L(u)$  is a neighbor of  $u$ , at the next step, vertex  $\mathbf{0}^k \mathbf{0} 1 L(u)$  will also be a neighbor of  $u$ . The proof for the second equality follows a similar argument. □



**Figure 6.** LCS is the longest common suffix of two vertices. We can see that the structure  $M_{LCS}$  produced from the vertex with label LCS is isomorphic to  $M(2)$ .

**4. Routing by the shortest paths in  $M(t)$**

Now we describe the routing protocol between any two vertices  $u$  and  $v$  of  $M(t)$ , which have labels, respectively,  $L(u)$  and  $L(v)$ . The way to find the shortest path routing between  $u$  and  $v$  is particular, in the sense that the routing is done both from  $u$  and  $v$ , until they attain a common vertex. However, to find the full routing there is only need to know the labels of the source and destination vertices. First, we compute the longest common suffix, LCS, of  $L(u)$  and  $L(v)$  and we write  $L(u)$  and  $L(v)$  as,  $L(u) = u_1 \dots u_m u_{m+1} u_{m+2} \text{LCS}$  and  $L(v) = v_1 \dots v_n v_{n+1} v_{n+2} \text{LCS}$ . If the vertex with label LCS has been introduced at some step  $t' < t$ , then vertices  $u_f = u_{m+1} u_{m+2} \text{LCS}$  and  $v_f = v_{n+1} v_{n+2} \text{LCS}$  were introduced at step  $t' + 2$  and together with LCS belong to a structure isomorphic to  $M(2)$ ; see figure 6. We call this structure  $M_{LCS}$ .

The point is to find the shortest paths between  $u$  and  $u'_f \in M_{LCS}$ , and between  $v$  and  $v'_f \in M_{LCS}$ , where  $L(u'_f) \subset L(u_f)$  and  $L(v'_f) \subset L(v_f)$ . The shortest path between  $u'_f$  and  $v'_f$  can be seen from figure 6.

From the labeling rules, the shortest route to reach  $M_{LCS}$  (either from  $u$  or  $v$ ) consists of going, at each step, to the neighbor with the shortest label (the label's length corresponds to the step in which the vertex was introduced).

To find  $u'$ , the neighbor ancestor with the shortest label of the vertex with label  $L(u) = u_1 \dots u_m u_{m+1} u_{m+2} \text{LCS}$ , we have to consider the following two cases.

- If  $u_1 = 1$ ,  $u'$  is the vertex with label  $L(u') = u_2 \dots u_m u_{m+1} u_{m+2} \text{LCS}$ . From lemma 3.2,  $d(u, u') = 1$ .
- If  $u_1 = \dots = u_k = 0$  and  $u_{k+1} = 1$  with  $k + 1 \leq m$ ,  $u'$  is the vertex with label  $L(u') = u_{k+2} \dots u_m u_{m+1} u_{m+2} \text{LCS}$ . Again, from lemma 3.2,  $d(u, u') = 1$ .

The following proposition provides a routing by the shortest path between two vertices  $u$  and  $v$  with a label different from  $b$  and such that  $L(u)$  is a suffix of  $L(v)$ . It gives also the distance between them.

**Proposition 4.1.** *Let  $u$  and  $v$  be two vertices of  $M(t)$  which verify:*

- $L(v) = s_1 \dots s_{n+1} L(u)$  or  $L(v) = s_1 \dots s_{n+1}$  and  $L(u) = a$  ( $s_i \in \{0, 1\}$ ).
- $s_{i_1} = \dots = s_{i_k} = 1, i_1 < \dots < i_k$  with  $1 \leq k \leq n$  and  $s_j = 0$  if  $j \in \{1, \dots, n\} \setminus \{i_1, \dots, i_k\}$ . (If  $s_1 = \dots = s_n = 0$  then  $k = 0$  in what follows.)

*Then, the distance,  $d(u, v)$ , and the routing between  $u$  and  $v$  are as follows.*

- If  $s_{n+1} = 1$ ,  $d(u, v) = 1 + k$  and the routing is

$$v, s_{i_1+1} \dots s_{n+1} L(u), s_{i_2+1} \dots s_{n+1} L(u), \dots, s_{i_k+1} \dots s_{n+1} L(u), u.$$



- If  $s_{n+1} = 0$ ,  $d(u, v) = 2 + k$  and
  - If  $L(u)$  contains one or more digits 1, i.e.  $L(u) = 0^{.l}. 01\alpha$ ,  $0 \leq l \leq \text{length}(L(u)) - 1$ , where  $\alpha$  is any sequence of binary digits, the routing is
 
$$v, s_{i_1+1} \dots s_{n+1}L(u), s_{i_2+1} \dots s_{n+1}L(u), \dots, s_{i_k+1} \dots s_{n+1}L(u), \alpha, u.$$
  - If  $L(u)$  is an all 0's sequence, the routing is
 
$$v, s_{i_1+1} \dots s_{n+1}L(u), s_{i_2+1} \dots s_{n+1}L(u), \dots, s_{i_k+1} \dots s_{n+1}L(u), b, u.$$
  - If  $L(u) = a$ , the routing is
 
$$v, s_{i_1+1} \dots s_{n+1}L(u), s_{i_2+1} \dots s_{n+1}L(u), \dots, s_{i_k+1} \dots s_{n+1}L(u), b, a.$$

**Proof.** Apply the routing protocol to  $L(v)$ . Every time that we find the sequence  $0^{.k}. 01$ ,  $k \geq 0$ , we consider lemma 3.2 and we add one unit to the distance. The protocol is used again after removing this sequence of digits from the label:

$$v, s_{i_1+1} \dots s_{n+1}L(u), s_{i_2+1} \dots s_{n+1}L(u), \dots$$

When we reach  $s_{n+1}$  we have to consider the following two cases (lemma 3.2):

- If  $s_{n+1} = 1$ , we add one unit to the distance and the routing is
 
$$v, s_{i_1+1} \dots s_{n+1}L(u), s_{i_2+1} \dots s_{n+1}L(u), \dots, s_{i_k+1} \dots s_{n+1}L(u), u.$$
- If  $s_{n+1} = 0$ , we add two units to the distance. To determine the final part of the routing, we consider these cases:
  - If  $L(u)$  contains 1 one or more times, we can write  $L(u) = 0^{.l}. 01\alpha$  where  $\alpha$  is a sequence of binary digits and the routing is
 
$$v, s_{i_1+1} \dots s_{n+1}L(u), s_{i_2+1} \dots s_{n+1}L(u), \dots, s_{i_k+1} \dots s_{n+1}L(u), \alpha, u.$$
 Note that from lemma 3.2,  $d(s_{i_k+1} \dots s_{n+1}L(u), \alpha) = 1$  and  $d(\alpha, u) = 1$ .
  - If  $L(u)$  is a sequence of 0's, from the construction—see figure 4—, vertices  $s_{i_k+1} \dots s_{n+1}L(u)$  and  $u$  are both connected to the initial vertex  $b$ . The routing is
 
$$v, s_{i_1+1} \dots s_{n+1}L(u), s_{i_2+1} \dots s_{n+1}L(u), \dots, s_{i_k+1} \dots s_{n+1}L(u), b, u.$$
  - If  $L(u) = a$ , the routing from  $v$  to  $a$  is
 
$$v, s_{i_1+1} \dots s_{n+1}, s_{i_2+1} \dots s_{n+1}, \dots, s_{i_k+1} \dots s_{n+1}, b, a.$$
 Note that in this case (see figure 4), vertex  $s_{i_k+1} \dots s_{n+1}$  is a sequence of 0's and it is also connected to vertex  $b$ . □

Finally, note that the routing between any vertex  $w$  with label  $L(w) = s_1 \dots s_n$  to  $b$  goes from  $w$  to  $a$  (see the former proposition) and then to  $b$ .

**Example 4.2.** If  $u$  and  $v$  are the vertices with labels  $L(v) = 00111001111000$  and  $L(u) = 11000$ , the routing is

00111001111000, 11001111000, 1001111000, 001111000, 111000, 11000

and  $d(v, u) = 5$ .

With the results obtained so far, we can now give the routing between any two vertices of  $M(t)$ .

In what follows, we denote by  $u \rightarrow v$  the (shortest) path from vertex  $u$  to vertex  $v$ , generated according to the rule of the neighbor ancestor which has the shortest label, and  $u \leftrightarrow v$  will denote the shortest path between two vertices of  $M_{LCS}$ .

**Theorem 4.3.** Let  $u$  and  $v$  be two vertices of  $M(t)$  with labels  $L(u) = u_1 \dots u_m u_{m+1} u_{m+2} \text{LCS}$  and  $L(v) = v_1 \dots v_n v_{n+1} v_{n+2} \text{LCS}$ .

The routing between  $u$  and  $v$  is given by

$$\begin{cases} u \rightarrow u_{m+1}u_{m+2}\text{LCS} \leftrightarrow v_{n+1}v_{n+2}\text{LCS} \leftarrow v & \text{if } u_{m+1} = v_{n+1} = 1 \\ u \rightarrow \text{LCS} \leftarrow v & \text{otherwise.} \end{cases}$$

**Proof.** As  $u_{m+1}u_{m+2}\text{LCS} \subset L(u)$  and  $v_{n+1}v_{n+2}\text{LCS} \subset L(v)$ , from proposition 4.1 we can compute the distances  $d(u, u_{m+1}u_{m+2}\text{LCS})$ ,  $d(v, v_{n+1}v_{n+2}\text{LCS})$  and find the routing  $u \rightarrow u_{m+1}u_{m+2}\text{LCS}$  and  $v \rightarrow v_{n+1}v_{n+2}\text{LCS}$ . Because LCS is the longest common suffix of  $L(u)$  and  $L(v)$ , we have that  $u_{m+2} \neq v_{n+2}$ . Consider  $u_{m+2} = 1$  and  $v_{n+2} = 0$  (the other case is similar). We have the following cases (see figure 6):

- If  $u_{m+1} = v_{n+1} = 1$ , the routing between  $u$  and  $v$  is  $u \rightarrow 11\text{LCS}, 1\text{LCS}, 0\text{LCS}, 10\text{LCS} \leftarrow v$ .
- If  $u_{m+1} = 0$  and  $v_{n+1} = 1$ , the routing between  $u$  and  $v$  is  $u \rightarrow 01\text{LCS}, \text{LCS}, 1\text{LCS}, 0\text{LCS}, 10\text{LCS} \leftarrow v$ .
- If  $u_{m+1} = 0$  and  $v_{n+1} = 0$  or if  $u_{m+1} = 1$  and  $v_{n+1} = 0$ , the routing goes through vertex LCS. □

## 5. Conclusion

We have provided a labeling system and produced a routing algorithm for a family of graphs which are outerplanar, modular, have an exponential degree hierarchy and are also small-world. At the same time the graphs have clustering zero. This combination of modularity, small clustering coefficient and small-world properties can be found in real networks such as some social and technical networks and those related to several biological systems (metabolic networks, protein interactome, etc) [7, 9]. The graphs are also outerplanar and many algorithms which are NP-complete for general graphs perform polynomial in outerplanar graphs [3]. This should be useful, together with the labeling proposed here, to find new efficient algorithms for dynamical processes (communication, hub location, synchronization, etc) for complex networks with similar properties. Finally, we should emphasize that the deterministic character of the graphs, in contrast to more usual probabilistic approaches, should facilitate the exact determination of other network parameters, the development of these algorithms and also help to understand the underlying mechanisms that have led some networks to their particular structure.

## Acknowledgments

This research was supported by the Ministerio de Educación y Ciencia, Spain, and the European Regional Development Fund under project MTM2008-06620 and partially supported by the Catalan Research Council under grant 2009SGR1387.

## References

- [1] Barabási A-L and Albert R 1999 Emergence of scaling in random networks *Science* **286** 509–12
- [2] Barrat A and Weigt M 2000 On the properties of small-world network models *Eur. Phys. J. B* **13** 547–60  
Barrat A and Weigt M 2008 *Discrete Appl. Math.* doi:10.1016/j.dam.2008.04.018
- [3] Brandstädt A, Le V B and Spinrad P J 1999 *Graph Classes: A Survey (SIAM Monographs on Discrete Mathematics and Applications)* (Philadelphia, PA)
- [4] Carmi S, Cohen R and Dolev D 2006 Searching complex networks efficiently with minimal information *Europhys. Lett.* **74** 1102–8

- [5] Comellas F and Miralles A 2009 Modeling complex networks with self-similar outerplanar unclustered graphs *Phys. A* **388** 2227–33
- [6] Clauset A, Newman M E J and Moore C 2004 Finding community structure in very large networks *Phys. Rev. E* **70** 066111
- [7] Ferrer i Cancho R, Janssen C and Solé R V 2001 Topology of technology graphs: small world patterns in electronic circuits *Phys. Rev. E* **64** 046119
- [8] Fortunato S and Barthélemy M 2007 Resolution limit in community detection *Proc. Natl Acad. Sci. USA* **104** 36–41
- [9] Newman M E J 2003 The structure and function of complex networks *SIAM Rev.* **45** 167–256
- [10] Newman M E J 2004 Fast algorithm for detecting community structure in networks *Phys. Rev. E* **69** 066133
- [11] Newman M E J and Girvan M 2004 Finding and evaluating community structure in networks *Phys. Rev. E* **69** 026113
- [12] Ozik J, Hunt B-R and Ott E 2004 Growing networks with geographical attachment preference: emergence of small worlds *Phys. Rev. E* **69** 026108
- [13] Solé R V and Valverde S 2004 Information theory of complex networks: on evolution and architectural constraints *Lecture Notes Phys.* **650** 189–207
- [14] Song C, Havlin S and Makse H A 2005 Self-similarity of complex networks *Nature* **433** 392–5
- [15] Song C, Havlin S and Makse H A 2006 Origins of fractality in the growth of complex networks *Nature Phys.* **2** 275–81
- [16] Watts D J and Strogatz S H 1998 Collective dynamics of ‘small-world’ networks *Nature* **393** 440–2
- [17] Zhang Z, Rong L and Comellas F 2006 Evolving small-world networks with geographical attachment preference *J. Phys. A: Math. Gen.* **39** 3253–61
- [18] Zhang Z, Rong L and Guo C 2006 A deterministic small-world network created by edge iterations *Physica A* **363** 567–72
- [19] Zhang Z, Zhou S, Wang Z and Shen Z 2007 A geometric growth model interpolating between regular and small-world networks *J. Phys. A: Math. Theor.* **40** 11863–76
- [20] Zhang Z, Zhou S, Shen Z and Guang J 2007 From regular to growing small-world networks *Physica A* **385** 765–72