Vertex Labeling and Routing in Recursive Clique-Trees, a New Family of Small-World Scale-Free Graphs*

F. COMELLAS
Universitat Politècnica de Catalunya, Spain

G. FERTIN
IRIN, Université de Nantes, France

A. RASPAUD
LaBRI, Université Bordeaux 1, France

Abstract

We present a new category of graphs, recursive clique-trees $K(q,t)$ (with $q \geq 2$ and $t \geq 0$), which have small-world and scale-free properties and allow a fine tuning of the clustering and the power-law exponent of their discrete degree distribution. This family of graphs is a generalization of recent constructions with fixed degree distributions. We first compute the relevant characteristics of those graphs: diameter, clustering and power law exponent. Then we propose a labeling of the vertices of $K(q,t)$, for any $q \geq 2$ and $t \geq 0$, that allows to determine a shortest path routing between any two vertices of $K(q,t)$, only based on the above mentioned labels.

Keywords

real-life networks, small-world, scale-free, labeling, routing protocol, shortest paths

1 Introduction

The World Wide Web, transportation and communication networks, biological or social networks, and many other real life networks have in common three characteristics : (i) a strong local clustering (the neighbors of a given vertex are themselves highly connected to each other), (ii) a small diameter and (iii) a distribution

*The authors thank a Picasso bilateral grant, references HF2001-0056 (Spain) and 04295SG (France). Additional support for the first author provided by the Ministry of Science and Technology, Spain, and the European Regional Development Fund (ERDF) under projects TIC2001-2171 and TIC2002-00155. Additional support for the two last authors provided by Action Spécifique CNRS "Algorithmique des Grands Graphes".

of degrees according to a power law (see for example [1, 2, 3, 12, 15]). Networks that satisfy conditions (i) and (ii) are said to be *small-world*, while they are *scale-free* if they satisfy condition (iii). In the recent past, there have been many attempts to give a general model for such networks, see for instance [19, 4].

While most of the work for these graphs is based on stochastic techniques and computer simulations [17, 19], this approach has shown to be quite limited in the sense that it does not always allow to compute the main network parameters ; moreover, the communication aspects of those stochastic networks are difficult to be considered. On the other hand, the use of simple deterministic models, with the help of standard graph theory concepts, allows a quick and exact determination of the relevant parameters of the graph, that may be compared with experimental data from real and simulated networks. Besides, much information can be derived from deterministic networks in terms of communication. Previous work in this context considered deterministic small-world graphs [10] comparable to those obtained stochastically by Watts and Strogatz, vertex replacement and vertex addition methods to produce small-world and scale-free graphs from a low diameter "backbone" graph [11], specific recursive scale-free constructions with fixed degree distributions [6, 13, 18] and scale-free trees (without clustering) [16].

We have recently introduced a recursive graph construction [9], (which we call *recursive clique-trees*) that provides scale-free small-world graphs with an adjustable diameter and clustering and such that the parameter of the power law associated to the degree distribution, γ, takes values between 2 and $1+\frac{\ln 3}{\ln 2} = 2.58496$ (it has been shown that the values for the scaling exponent of most real networks [5] are precisely in this range). Those networks are in fact a generalization of [14, 13], in which the diameter, the clustering and the parameter γ associated to the power law distribution were fixed.

In this paper, we describe the construction of recursive clique-trees, and to make this paper self-contained we recall results from [9], that is the main characteristics of this family of graphs, in terms of its relevant parameters (diameter, clustering, power law exponent). We then continue the study of these graphs by giving a labeling of the vertices of recursive clique-trees, that is optimal in length and allows to determine a shortest path between any two vertices, only based on the above mentioned labels, providing in this way a new deterministic model for many real life networks.

2 Recursive Clique-Trees: Definition and Properties

Definition 2.1 *The recursive clique-tree $K(q,t)$ ($t \geq 0$, $q \geq 2$) is the graph constructed as follows: For $t = 0$, $K(q,0)$ is the complete graph K_q (or q-clique). For $t \geq 1$, $K(q,t)$ is obtained from $K(q,t-1)$ by (i) adding for each of its existing subgraphs isomorphic to a q-clique a new vertex and (ii) joining it to all the vertices of this subgraph.*

Then, at $t=1$, $K(q,1)$ results in the complete graph with $q+1$ vertices, K_{q+1}, and at $t=2$ we add $q+1$ new vertices, each of them connected to all the vertices of one of the q-cliques K_q (subgraphs of K_{q+1}), and so on (see Fig. 1 for the case $q=3$).

This construction produces a complex growing graph with a tunable parameter q which controls all its relevant characteristics. In the particular case $q=2$, we obtain the same graph as in [13]. However our family is infinite as q can take any natural value starting from 2.

Notice that although we call the graph a recursive clique-tree, the graph contains numerous cycles and hence is not a tree in the strict sense. Recursive clique-trees $K(q,t)$ are a natural generalization of trees (if one considers the case $q=1$, then we obtain binomial trees). Similar and much more general constructions in which new vertices are joined to every vertex of a given q-clique have been considered, in another context, in [7] and termed *k-trees*.

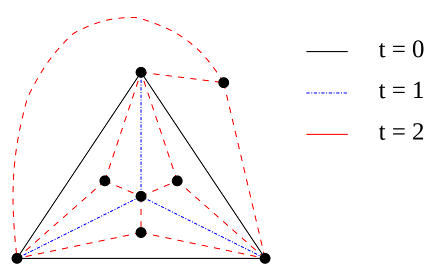

Figure 1: First stages of a growing recursive clique-tree $K(3,t)$.

2.1 Recursive construction

There exists another interesting way to construct $K(q,t)$, which clearly shows the recursive structure of such graphs. We call *native clique* of $K(q,t)$ its initial q-clique at $t=0$. For any step $t\geq 1$, $K(q,t)$ is constructed as follows: consider a $(q+1)$-clique, then every subgraph of it isomorphic to a q-clique (there are $q+1$ such cliques) is a native clique of a $K(q,t-1)$; see figure 2.

2.2 Size and order

Table 1 gives the number of new edges added to the tree at each step and the total number of K_q at this step.

Therefore we can easily compute the total number of edges at step t:

$$|E_t| = \frac{q(q-1)}{2} + q\sum_{i=0}^{t-1}(q+1)^i = \frac{q(q-1)}{2} + (q+1)^t - 1$$

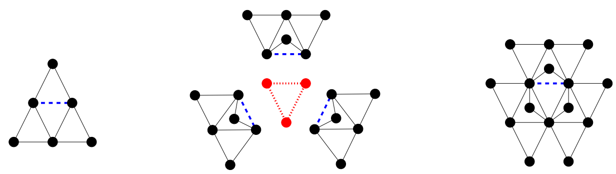

Figure 2: Recursive construction of $K(2,t)$. We *glue* a native clique of a $K(2,t-1)$ on each 2-clique of K_3 to construct $K(2,t)$. In this figure we obtain $K(2,3)$ from 3 copies of $K(2,2)$.

Step	New edges	Number of K_q
0	$\frac{q(q-1)}{2}$	1
1	q	$q+1$
2	$q(q+1)$	$q(q+1)+(q+1)=(q+1)^2$
3	$q(q+1)^2$	$q(q+1)^2+(q+1)^2=(q+1)^3$
$\cdots$	$\cdots$	$\cdots$
i	$q(q+1)^{i-1}$	$(q+1)^i$
$i+1$	$q(q+1)^i$	$q(q+1)^i+(q+1)^i=(q+1)^{i+1}$
$\cdots$	$\cdots$	$\cdots$

Table 1: Number of new edges and of q-cliques in $K(q,t)$.

The distribution of vertices and degrees is given in Table 2. Adding up the number of vertices gives the value of N_t, the number of vertices of $K(q,t)$ at step t:

$$N_t = \sum_{j=1}^{t}(q+1)^j + (q+1) = \frac{(q+1)^t - 1}{q} + q$$

From Table 2 we also see that the maximum degree at step i is $\Delta_i = \frac{q-q^i}{1-q} + q = \frac{q^i-1}{q-1} + q - 1$ and we can compute the average degree for a clique-tree $K(q,t)$: $\overline{k}_t = 2|E_t|/N_t = \frac{q\left(q^2-q+2\,(q+1)^t-2\right)}{(q+1)^t-1+q^2}$ (for $q=2$, it is $\dfrac{4}{1+3^{1-t}}$; more generally, $\overline{k}_t \sim 2q$ when t gets large).

2.3 Degree distribution

Here, we recall that $K(q,t)$ is a scale-free network, that is the number of vertices of a given degree k, $N(k',t)$, is of the form $k^{1-\gamma}$, where γ is a constant. More precisely, γ ranges in $]2;2.58496]$ depending on the value of q.

Step	Number of vertices	Degree
1	$q+1$	q
2	$q+1$ $q+1$	$2q$ q
3	$q+1$ $q+1$ $(q+1)^2$	q^2+2q $2q$ q
4	$q+1$ $q+1$ $(q+1)^2$ $(q+1)^3$	q^3+q^2+2q q^2+2q $2q$ q
$\cdots$	$\cdots$	$\cdots$
i	$q+1$ $q+1$ $\cdots$ $(q+1)^{i-2}$ $(q+1)^{i-1}$	$q^{i-1}+q^{i-2}+\cdots+q^2+2q$ $q^{i-2}+\cdots+q^2+2q$ $2q$ q
$\cdots$	$\cdots$	$\cdots$

Table 2: Distribution of degrees and vertices in $K(q,t)$.

Theorem 2.2 *[9] The γ coefficient associated to the power law in recursive clique-trees $K(q,t)$ satisfies $\gamma \approx 1 + \frac{\ln(q+1)}{\ln q}$ when t gets large.*

Proof. Since the degree spectrum is discrete (cf. Table 2), we use a cumulative distribution and we compute $P_{cum}(k) = \sum_{k' \geq k} N(k',t)/N_t$. Here k and k' are points of the discrete degree spectrum. For a degree $k = q^{t-l} + q^{t-l-1} + \cdots + q + q = q(\frac{q^{t-l}-1}{q-1} + 1)$ with $l \leq t-1$, there are $(q+1)^{l-1}$ vertices with this exact degree. Since $\sum_{k' \geq k} N(k',t) = \sum_{p=1}^{l-1}(q+1)^p + (q+1) = \frac{(q+1)^l-1}{q} + q$, and $N_t = \frac{(q+1)^t-1}{q} + q$, we have $(q(\frac{q^{t-l}-1}{q-1}+1))^{1-\gamma} = \frac{\frac{(q+1)^l-1}{q}+q}{\frac{(q+1)^t-1}{q}+q} = \frac{(q+1)^l-1+q^2}{(q+1)^t-1+q^2}$. Therefore, for t large $(q^{t-l})^{1-\gamma} = (q+1)^{l-t}$ and $\gamma \approx 1 + \frac{\ln(q+1)}{\ln q}$ so that $2 < \gamma < 2.58496$. □

2.4 Clustering distribution

Definition 2.3 *The clustering coefficient C_v of a vertex v is the ratio between (i) the total number of existing connections between the k neighbors of v and (ii) $k(k-1)/2$, the number of all possible connections between them.*

Definition 2.4 *The clustering of a graph is the average value of the clustering coefficient over all its vertices.*

We next give the clustering of a clique-tree and some insights on its computation (details can be found in [9]):

Theorem 2.5 *[9] The clustering parameter of $K(q,t)$ is $\overline{C}_t = \frac{S_t}{N_t}$, where*

$$S_t = 2 \cdot \frac{(q+1)(q-1)(\Delta_t - \frac{q}{2})}{\Delta_t(\Delta_t - 1)} + \sum_{i=1}^{t-1} 2 \cdot \frac{(q+1)^{t-i}(q-1)(\Delta_i - \frac{q}{2})}{\Delta_i(\Delta_i - 1)}$$

Proof. The proof is by induction on t. We compute the clustering of a given vertex x by keeping track of the number of neighbors, and edges among them, which x has at any iteration step. For this we count the number of q-cliques that contain x as each of them will be connected to a newly introduced vertex at a given iteration step. □

We can verify that for $t \geq 7$ and for any $q \geq 3$, $\overline{C}_t \geq \frac{3q-2}{3q-1}$. Thus the clustering is high, and, similarly to the γ coefficient of the power law, is tunable by choosing the right value of q: in particular, $\overline{C}_t$ ranges from $\frac{4}{5}$ (in the case $q = 2$) to a limit of 1 when q gets large.

2.5 Diameter

Computing the exact diameter of $K(q,t)$ can be done analytically, and gives the result shown below.

Theorem 2.6 *[9] The diameter of $K(q,t)$, denoted $Diam(K(q,t))$, is the following : $Diam(K(q,t)) = 2(\lfloor \frac{t-2}{q} \rfloor + 1) + f(q,t)$ where $f(q,t) = 0$ if $t - \lfloor \frac{t-2}{q} \rfloor q \leq \lceil \frac{q+1}{2} \rceil$, and 1 otherwise.*

Proof. It can be easily seen that (i) $Diam(K(q,1)) = 1$; (ii) if $2 \leq t \leq \lceil \frac{q+1}{2} \rceil$ then $Diam(K(q,t)) = 2$; and (iii) if $\lceil \frac{q+1}{2} \rceil + 1 \leq t \leq q$, $Diam(K(q,t)) = 3$.

Now, we note that for any step $t > q$, the diameter always lies between pair of vertices that have just been created at this step. We will call *outervertices* such vertices. Then, we note that, by construction, no two outervertices created at the same step t can be connected. Hence, if v_t is an outervertex created at step t, then v_t has been connected to a q-clique composed of vertices created at pairwise different steps $t_1 < t_2 < \ldots < t_q$ and consequently $t_1 \leq t - q$.

Now we want to know the maximum distance between two outervertices u_t and v_t. Let us call *initial $(q+1)$-clique* the $(q+1)$-clique obtained after step 1. The idea is, starting from u_t, to reach the initial $(q+1)$-clique by jumps from u_t to u_{t-q}, then to u_{t-2q}, etc. Hence it takes at most about $\frac{t}{q}$ jumps to go from u_t to $K(q,1)$.

Then it takes at most as many jumps to go from $K(q,1)$ to v_t and we conclude that the diameter cannot be bigger than (roughly) $\frac{2t}{q}$. □

When t gets large, then $Diam(K(q,t)) \sim \frac{2t}{q}$, while $N_t \sim q^{t-1}$, thus the diameter clearly grows logarithmically with the number of vertices.

3 Labeling of $K(q,t)$ and Routing by Shortest Paths

In this section, we describe a way to label the vertices of $K(q,t)$, for any $q \geq 2$ and $t \geq 0$, such that a routing by shortest paths between any two vertices of $K(q,t)$ can be deduced from the labels. We note that a more general result on shortest paths routing of graphs with given treewidth is given in [8]. However, here we address the more specific case of recursive clique-trees $K(q,t)$.

In what follows, $L(v)$ will denote the label of v, for any vertex $v \in V(K(q,t))$.

3.1 Labeling Protocol

The idea here is to assign to any vertex v created at step $t \geq 2$ a label of length $t-1$, in the form of a word of $t-1$ digits, each digit being an integer between 1 and $q+1$ (the vertices obtained at step $t=0$ and $t=1$, i.e. the vertices of the initial $(q+1)$-clique $K(q,1)$, are assigned a special label). More precisely, the labeling of any vertex v of $K(q,t)$ is done thanks to the following rules:

- Label the vertices of the initial $(q+1)$-clique $K(q,1)$ arbitrarily, with labels $1', 2' \ldots (q+1)'$.
- At any step $t \geq 2$, when a new vertex v is added and joined to all vertices of a clique K_q:
  - If v is the first vertex that is generated from the q-clique it is connected to (we recall that v is then called an *outervertex*):
    1. If v is connected to q vertices of the initial $(q+1)$-clique, then $L(v) = l$, where l' is the only vertex of the initial $(q+1)$-clique that does not belong to this q-clique.
    2. If not, then v is connected to $w_1, w_2 \ldots w_q$, where at least one of the w_is is not a vertex of the initial $(q+1)$-clique. Thus, any such vertex has a label $L(w_i) = s_{1,i}s_{2,i} \ldots s_{k,i}$. W.l.o.g., let w_1 be the vertex not belonging to the initial $(q+1)$-clique with the longest label. In that case, we give to vertex v the label $L(v)$ defined as follows: $L(v) = \alpha \cdot L(w_1)$, where $1 \leq \alpha \leq q+1$ is the only integer not appearing as first digit in the labels of $w_1, w_2 \ldots w_q$, that is $\alpha = \{1, 2 \ldots q, q+1\} / \cup_{i=1}^{q} s_{1,i}$ (the fact that α is unique will be proved by Property 1 below).

– If v is a *twin* vertex (that is, v is generated from a q-clique that has already generated at least one vertex v_0 in a previous step), then $L(v) = \beta\beta\ldots\ldots\beta L(v_0)$, where β is the first element of $L(v_0)$. The number of βs in $L(v)$ is determined by the fact that we want $|L(v)| = t-1$.

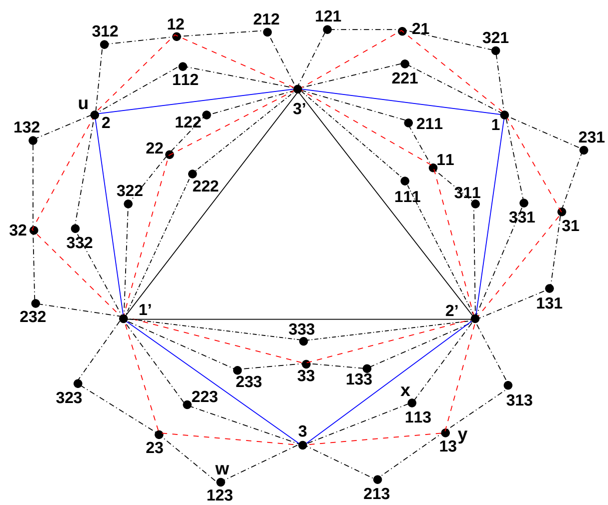

Figure 3: How to label the vertices of $K(2,4)$

Such a labeling is illustrated in Figure 3. In this figure, we label the vertices of $K(q,t)$, for $q=2$ and up to $t=4$. We see that vertex u, created at step 1, has label $L(u)=2$ because it is not connected to vertex $2'$ of the initial 3-clique. Vertex w is an outervertex not connected to any vertex of the initial 3-clique: its label is first composed of the only digit not appearing as first digit of its neighbors (in this case, 1), concatenated with the longest label of its neighbors (in this case, 23). Vertex x is a twin of vertex y, and is thus assigned label 113, since $L(y)=13$ and $t=4$.

Thus, we see that for any $t \geq 2$, any vertex v_t created at step t has a unique label, and that for any vertex v created at step $t \geq 2$, $L(v) = s_1 s_2 \ldots s_{t-1}$ is of length $t-1$, where each digit s_j satisfies $1 \leq s_j \leq q+1$; while the vertices created at step 0 and 1 have length 1 (these are the l', $1 \leq l \leq q+1$).
We also note that since for any step $t \geq 1$, the number of vertices that are added to the recursive clique-tree is equal to $(q+1)^{t-1}$, the labeling we propose is optimal in the sense that each label $L(v_t)$ of a vertex created at step t is a $(q+1)$-ary word of length $t-1$. Globally, any vertex of $K(q,t)$ is assigned a label of length $O(\log_{q+1}(t-1))$; since there are $N_t = \frac{(q+1)^t-1}{q} + q = (q+1)^{t-1}(1+\frac{1}{q}) + \frac{q^2-1}{q}$ vertices in $K(q,t)$, we can see that, overall, the labeling is optimal as well.

Next, we give three properties about the above labeling. Property 1 ensures that our labeling is deterministic. Property 2 is a tool to prove Property 3, the latter being important to show that our routing protocol is valid and of shortest paths.

Property 1 *In $K(q,t)$, for any $(q+1)$-clique induced by vertices $w_1, w_2 \dots w_{q+1}$, every integer $1 \leq i \leq q+1$ appears exactly once as the first digit of the label of a w_j.*

Proof. By induction on t. When $t = 1$, the property is true by construction. Suppose now that the property is true for any $t' < t$, and let us then show it is true for t. Any $(q+1)$-clique in $K(q,t)$ is composed of exactly one vertex v created at a given step t_1, and q vertices $w_1, w_2, \dots w_q$ created at steps strictly less than t_1. If $t_1 < t$, then the property is true by induction hypothesis. If $t_1 = t$, then we have two cases: (1) v is a twin vertex. In that case, by definition, the first digit of $L(v)$ is the same as the first digit of $L(v_0)$, where v_0 is the outervertex created (at a step strictly less than t) thanks to the q-clique formed by $w_1, w_2, \dots w_q$. Since by induction hypothesis the property is true for the $(q+1)$-clique formed with $w_1, w_2, \dots w_q, v_0$, it is also true for the $(q+1)$-clique formed with $w_1, w_2, \dots w_q, v$; (2) v is an outervertex. In that case, since $w_1, w_2, \dots w_q$ induce a q-clique, this means that this q-clique did not exist at step $t-1$. In other words, one of the w_is, say w_1, has been created at step $t-1$, based on q vertices $w_2, w_3 \dots w_q$ and x. By induction hypothesis, each integer $1 \leq i \leq q+1$ appears exactly once as first digit of the labels of $w_1, w_2, w_3 \dots w_q, x$. However, by construction, the first digit of $L(v)$ is the first digit of $L(x)$. Thus we conclude that each integer $1 \leq i \leq q+1$ also appears exactly once as first digit of the labels of $w_1, w_2, w_3 \dots w_q, v$, and the result is proved by induction. □

Property 2 *Let v_t be a vertex of $K(q,t)$ created at step $t \geq 2$. Among the vertices $w_1, w_2, \dots w_q$ forming the q-clique that generated v_t, let $w_1, w_2 \dots w_k$, $k \leq q$, be the vertices that do not belong to the initial $(q+1)$-clique. Then $L(v_t)$ is a superstring of $L(w_i)$ for all $1 \leq i \leq k$.*

Proof. By induction on t. When $t = 2$, any vertex v_2 created at step 2 is connected to vertices of the initial $(q+1)$-clique only. Thus the result is true. Now suppose the result is true for any $2 \leq t' \leq t-1$, $t \geq 3$, and let us prove it is then true for t. For this, we consider a vertex v_t created at step t, and the q-clique $\mathcal{C}$ it is connected to. We distinguish two cases:

- v_t is a twin: then the result is true by construction of the labels ; indeed, $L(v_t)$ is by construction a superstring of $L(v_{t'})$, $t' < t$, where $v_{t'}$ is the outervertex generated from $\mathcal{C}$. Since $t' < t$, by induction hypothesis, $L(v_{t'})$ is a superstring of the labels of all the vertices of $\mathcal{C}$ that are not in the initial $(q+1)$-clique.

- v_t is an outervertex: then v_t is a neighbor of w_p, where w_p was created at step $t-1$ (since v_t is not a twin). However, w_p was created itself thanks to a q-clique, say $\mathcal{C}'$, composed of vertices $x_1, x_2 \ldots x_q$. W.l.o.g., suppose that $k \leq q$ such vertices, $x_1, x_2 \ldots x_k$ do not belong to the initial $(q+1)$-clique. By induction hypothesis, $L(x_i) \subseteq L(w_p)$ for any $1 \leq i \leq k$. Hence, in $\mathcal{C}$, w_p is the vertex not belonging to the initial $(q+1)$-clique that has the longest label. By construction of $L(v_t)$, we have that $L(w_p) \subseteq L(v_t)$, thus we also conclude that $L(x_i) \subseteq L(v_t)$ for any $1 \leq i \leq k$. Thus $L(v_t)$ is a superstring of the labels of any vertex of $\mathcal{C}$ that does not belong to the initial $(q+1)$-clique, and the result is proved by induction.

□

Property 3 *Let v_t be a vertex of $K(q,t)$ created at step $t \geq 2$. For any $1 \leq i \leq q+1$, if $i \notin L(v_t)$, then v_t is a neighbor of a vertex v' of the initial $(q+1)$-clique, such that $L(v') = i'$.*

Proof. By induction on t. When $t = 2$, any vertex v_2 constructed at step 2 is assigned label i, where i' is the only vertex of the initial $(q+1)$-clique v_t is not connected to ; thus, by construction, the property is satisfied.
Now we suppose that the property is true for any $2 \leq t' \leq t-1$, $t \geq 3$, and we will show it then holds for t as well. As for the previous property, we consider a vertex v_t created at step t, and we distinguish two cases:

- v_t is a twin: let $v_{t'}$ be the outervertex generated from the same clique $\mathcal{C}$ as v_t. Thus $t' < t$. By induction, the property is true for $v_{t'}$; since by construction of the labels, we have $L(v_{t'}) \subseteq L(v_t)$ and $L(v_t)$ contains exactly the same digits as $L(v_{t'})$, we conclude that the property holds for v_t as well.

- v_t is not a twin (and thus is an outervertex): then v_t is connected to a vertex w_{t-1} that was created at step $t-1$. However, w_{t-1} was created itself thanks to a q-clique $\mathcal{C}'$ composed of vertices $x_1, x_2 \ldots x_q$. Among those q vertices, only one, say x_p, does not belong to $\mathcal{C}$. W.l.o.g., suppose that $k \leq q$ such vertices, $x_1, x_2 \ldots x_k$ do not belong to the initial $(q+1)$-clique. Now suppose that $i \notin L(v_t)$; then i appears as the first digit of one of the $L(x_j)$s, $1 \leq j \leq q-1$, or of $L(w_{t-1})$ (by Property 1). However, $L(x_j) \subseteq L(w_{t-1}) \subseteq L(v_t)$ for any $1 \leq j \leq k$ (by Property 2). Thus, neither w_{t-1} nor any vertex among the x_js, $1 \leq j \leq k$ contains the digit i in its label. Hence, only a vertex y from the initial $(q+1)$-clique can have i in its label, and thus $L(y) = i'$. Hence it suffices to show that v_t and y are neighbors to prove the property. The only case for which this would not happen is when $y = x_p$; we will show that this is not possible. Indeed, by construction of the labels, the first digit of $L(v_t)$ is the only integer not appearing as first digit of the labels of the vertices of $\mathcal{C}$, that is $w_{t-1}, x_1, x_2 \ldots x_{p-1} x_{p+1} \ldots x_q$. However, the fact that we suppose $y = x_p$ means that no vertex of $\mathcal{C}$ contains i in its label. Thus this would mean that the first digit of $L(v_t)$ is i, a contradiction. Thus, v_t is connected to y with $L(y) = i'$, and the induction is proved.

□

3.2 Routing Protocol

Now we describe the routing protocol between any two vertices u and v of $K(q,t)$, with labels respectively equal to $L(u)$ and $L(v)$. We note that since $K(q,0)$ (resp. $K(q,1)$) is isomorphic to the complete graph K_q (resp. K_{q+1}), we can assume $t \geq 2$. The routing protocol is special here in the sense that the routing is done both from u and v, until they reach a common vertex. Hence, the routing strategy will be used simultaneously from u and from v. In order to find a shortest path between any two vertices u and v, the routing protocol is as follows:

1. Compute the longest common suffix $LCS(L(u), L(v))$ of $L(u)$ and $L(v)$.

2. If $LCS(L(u), L(v)) = \emptyset$:

   (a) Simultaneously from u and v (say, from u): let $u = u_0$ and go from u_i to u_{i+1}, $i \geq 0$ where u_{i+1} is the neighbor of u_i with shortest label.

   (b) Stop when u_k is a neighbor of the initial $(q+1)$-clique.
   Let $\overline{L}(u_k)$ (resp. $\overline{L}(v_{k'})$) be the integers not present in $L(u_k)$ (resp. $L(v_{k'})$), and let $S = \overline{L}(u_k) \cap \overline{L}(v_{k'})$.

      i. If $S \neq \emptyset$, pick any $l \in S$, and close the path by taking the edge from u_k to l', and the edge from l' to $v_{k'}$.

      ii. If $S = \emptyset$, we route from u_k to any neighbor l_1' (belonging to the initial $(q+1)$-clique) of u_k, and we do similarly from $v_{k'}$ to a neighbor l_2' (belonging to the initial $(q+1)$-clique) of $v_{k'}$. Then, we take the edge from l_1' to l_2' and thus we close the path from u to v.

3. If $LCS(L(u), L(v)) \neq \emptyset$, then we call *least common clique* of u and v, or $LCC(u,v)$, the clique indicated by the longest common suffix $LCS(L(u), L(v))$. We simultaneously route from u and v to (respectively) u_k and $v_{k'}$, going each time to the neighbor with $LCS(L(u), L(v))$ as label suffix, and having the shortest label. Similarly as above, we stop at u_k (resp. $v_{k'}$), where u_k (resp. $v_{k'}$) is the first of the u_is (resp. of the v_js) to be a neighbor of $LCC(u,v)$. Then there are two subcases, depending on $S = \overline{L}(u_k) \cap \overline{L}(v_{k'})$.

   (a) If $S \neq \emptyset$, we close the path by going to any vertex w with label $l \cdot LCS(L(u), L(v))$, $l \in S$.

(b) If $S = \emptyset$, then we route from u_k (resp. $v_{k'}$) to any neighbor w_1 (resp. w_2) in $LCC(u,v)$, and we close the path by taking the edge (w_1,w_2), which exists since both vertices w_1 and w_2 belong to the same clique $LCC(u,v)$.

We now give the main ideas for the validity of the above routing protocol. Take any two vertices u and v. By construction of $L(u)$ and $L(v)$, the longest common suffix $LCS(L(u),L(v))$ indicates to which $(q+1)$-clique u and v have to go. We can consider this as a way for u and v to reach their least common ancestor in the tree of cliques induced by the construction of $K(q,t)$, or the "*least common clique*". In case (2) this least common clique is the initial $(q+1)$-clique ; thus, u and v have to get back to it. In case (3), the shortest path does not go through the initial $(q+1)$-clique, and the least common clique of u and v, say $LCC(u,v)$, is indicated by the longest common suffix $LCS(L(u),L(v))$. In other words, the length of $LCS(L(u),L(v))$ indicates the depth of $LCC(u,v)$ in the tree of cliques induced by the construction of $K(q,t)$. In that case, the routing is similar as in Case (2), except that the initial $(q+1)$-clique has to be replaced by the clique $LCC(u,v)$. Hence, the idea is to adopt the same kind of routing, considering only neighbors which also have $LCS(L(u),L(v))$ as suffix in their labels.
When this least common ancestor is determined, one can see, still by construction, that the shortest route to reach this clique (either from u or v) is to go to the neighbor which has smallest label, since the length of the label indicates at which step the vertex was created. Besides, the earlier the neighbor w was created, the smaller the distance from w to the least common clique is.
When we have reached, from u (resp. from v), a vertex u_k (resp. $v_{k'}$) that is a neighbor of the least common clique, the last thing we need to know is whether u_k and $v_{k'}$ are neighbors. Thanks to Property 3, we know that looking at $L(u_k)$ and $L(v_{k'})$ is sufficient to answer this question. More precisely :

- In case 2(b)-i, u_k and $v_{k'}$ share a neighbor in the initial $(q+1)$-clique (by Property 3). All those common neighbors have label l', where $l \in S$. Hence, if we pick any $l \in S$, then there exists and edge between u_k and l', as well as an edge between l' and $v_{k'}$.

- In case 2(b)-ii, u_k and $v_{k'}$ do not share a neighbor in the initial $(q+1)$-clique. Hence, taking a route from u_k (resp. $v_{k'}$) to any neighbor l'_1 (resp. l'_2) belonging to the initial $(q+1)$-clique, we can finally take the edge from l'_1 to l'_2 (which are neighbors, since they both belong to the initial $(q+1)$-clique) in order to close the path from u to v.

- In case 3(a), u_k and $v_{k'}$ share a neighbor in $LCC(u,v)$. Hence we can close the path by going to any vertex w with label $l \cdot LCS(L(u),L(v))$, $l \in S$, since w is a neighbor of both u_k and $v_{k'}$.

- In case 3(b), u_k and $v_{k'}$ do not share a neighbor in $LCC(u,v)$. Hence we route from u_k (resp. $v_{k'}$) to any neighbor w_1 (resp. w_2) in $LCC(u,v)$, and we close the path by taking the edge (w_1,w_2). (w_1,w_2) exists since both vertices w_1 and w_2 belong to the same clique $LCC(u,v)$.

Hence we conclude that our labeling of $v(K(q,t))$ allows a routing between any two vertices u and v, and that it is of shortest paths ; besides, this labeling is optimal in the length of the labels.

4 Discussion

The recursive clique-trees $K(q,t)$ that we study in this paper are recursively constructed graphs which have both small-world and scale-free characteristics. Moreover, it is possible to obtain different clustering parameters and power-law exponents by choosing adequately the value of q. They are actually a deterministic tunable generalization of the scale-free growing networks introduced in [14] in which one vertex is created per unit time and connects to both the ends of a randomly chosen edge. These networks were further generalized and called "pseudofractal" graphs in [13], corresponding to the particular case $q=2$.

We have also introduced a way to label the vertices, such that (i) the length of the label is optimal and (ii) a shortest path routing can be derived from the labels of the vertices. We are now studying the possibility that this labeling, or a slight variant, could also be used for computing the distance between two vertices, thus making it an optimal *distance labeling*.

We finally mention that modifications of our construction, for example by choosing that in different steps the new vertices added are attached to cliques of different sizes, would allow a richer structure and even more flexibility in the control of the clustering and power-law exponent.

References

[1] L. Adamic and B. Huberman, Power-law distribution of the world wide web, *Science* **287** (2000), 2115.

[2] R. Albert, H. Jeong, A.-L. Barabási, Diameter of the world wide web, *Nature* **401** (1999) 130-131.

[3] E. Almaas, R. V. Kulkarni, and D. Stroud, Characterizing the structure of small-world networks, *Phys. Rev. Lett.* **88** (2002), 098101.

[4] A.-L. Barabási and R. Albert, Emergence of scaling in random networks, *Science* **286** (1999) 509–512.

[5] A.-L. Barabási, Z. Dezö, E. Ravasz, S.-H. Yook, and Z. Oltvai, Scale-free and hierarchical structures in complex networks, In *Proc. 18 Sitges Conference "Statistical Mechanics of Complex Networks"* (2003).

[6] A.-L. Barabási, E. Ravasz, and T. Vicsek, Deterministic scale-free networks, *Physica A* **299** [3–4] (2001) 559–564.

[7] H.L. Bodlaender, Some classes of graphs with bounded treewidth, *Bulletin of the EATCS* **36** (1988) 116–126.

[8] S. Chaudhuri and C. D. Zaroliagis, Shortest paths in digraphs of small treewidth. Part I: Sequential algorithms, *Algorithmica* **27** [3] (2000) 212–226.

[9] F. Comellas, G. Fertin, and A.Raspaud, Recursive graphs with small-world scale-free properties, *submitted* (2002).

[10] F. Comellas, J. Ozón, and J.G. Peters, Deterministic small-world communication networks, *Inform. Process. Lett.* **76**, 83–90 (2000).

[11] F. Comellas and M. Sampels, Deterministic small-world networks, *Physica A* **309** (2002) 231–235.

[12] F. Comellas and J. Ozón, On the universality of small-world graphs *Electronic Notes in Discrete Math.* **10** (2001).

[13] S.N. Dorogovtsev, A.V. Goltsev, and J.F.F. Mendes, Pseudofractal scale-free web, *Phys. Rev. E* **65** (2002), 066122.

[14] S.N. Dorogovtsev, J.F.F. Mendes, and A.N. Samukhin, Size-dependent degree distribution of a scale-free growing network, *Phys. Rev. E* **63** (2001), 062101.

[15] M. Faloutsos, P. Faloutsos and C. Faloutsos, On power-law relationships of the internet topology. In *Proc. ACM SIGCOMM Conf.* (1999), pp. 251–260.

[16] S. Jung, S. Kim, and B. Kahng, Geometric fractal growth model for scale-free networks. *Phys. Rev. E* **65** 056101 (2002).

[17] S. R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal, Stochastic models for the web graph, In *Proc. 41st Symp. Foundations of Computer Science* (2000), IEEE, pp. 57–65.

[18] E. Ravasz and A.-L. Barabási, Hierarchical organization in complex networks. *Phys. Rev. E* **67** 026112 (2003).

[19] D.J. Watts and S.H. Strogatz, Collective dynamics of 'small-world' networks, *Nature* **393**, 440–442 (1998).

Francesc Comellas is with Departament de Matemàtica Aplicada IV, EPSC, Universitat Politècnica de Catalunya, Avda. Canal Olímpic s/n, 08860 Castelldefels, Barcelona, Catalonia, Spain. E-mail: comellas@mat.upc.es

Guillaume Fertin is with IRIN, Université de Nantes, 2 rue de la Houssinière, BP 92208, 44322 Nantes Cedex 3, France. E-mail: fertin@irin.univ-nantes.fr

André Raspaud is with LaBRI, Université Bordeaux 1, 351 Cours de la Libération, 33405 Talence Cedex, France. E-mail: raspaud@labri.fr