Laboratoire Bordelais de Recherche en Informatique
UMR 5800 - Université Bordeaux I, 351, cours de la Libération,
33405 Talence CEDEX, France

Research Report RR-1268-01

A Note on Implicit Representation of Graphs

by Cyril Gavoille

December, 2001

A Note on Implicit Representation of Graphs

Cyril Gavoille*

Abstract

An implicit representation for a given family of graphs is a node labeling scheme of all the n-node graphs of the family by binary strings of length $O(\log n)$ such that adjacency between any two nodes can be decided from their labels only. In this note, we design a $3 \log n$-bit labeling scheme for planar graphs constructible in $O(n)$ time and with constant time adjacency test, improving by a $\log n$ additive factor the result of Kannan, Naor and Rudich [KNR88]. More generally, we show that bounded k-arboricity graphs support an asymptotic $k \log n$-bit labeling scheme with $O(k)$ adjacency test. For large k, we construct a $O(k \log(n/k))$-bit labeling scheme with constant adjacency test, and show that the label length is asymptotically optimal.

Keywords: compact graph representation, implicit representation, partial k-trees, k-decomposable graphs, arboricity, degeneracy

1 Introduction

There are two basic ways of representing an n-node m-edge graph: matrix or edge list representation. The first one uses $O(n^2)$ bits and answers in constant time whether any two nodes are adjacent or not. The list representation uses $O(n + m)$ memory words[1], whereas such a data-structure does not support in the worst-case constant time adjacency queries.

So matrix representation is time-efficient but not space-efficient for sparse graphs, like planar graphs that have at most $3n-6$ edges. However, for several graph families, there are ways to compact the representation of a graph. For instance, Chang, Lin and Lu showed in [CLL01] that planar graphs support a $2m + 2n + o(n)$ bits representation with constant time queries. Although this representation is rather compact (this is $8n + o(n)$ bits for $m = 3n - o(n)$ bits), it does not allow *label-based* graph representation. A label-based representation allows retrieving useful information about arbitrary functions or substructures

*Laboratoire Bordelais de Recherche en Informatique, Université Bordeaux I. *gavoille@labri.fr*

[1] In this paper we assume that in constant time one can compute standard arithmetic operations on words of length $O(\log n)$ bits.

in a graph in a *localized* manner, i.e., using only local pieces of information available to, or associated with, the nodes under inspection, and not having to search for additional global information. It is clearly a desirable property for a graph in the framework of distributing computing where individual processor element of a network want to communicate with its neighbors but has no enough local memory resources to store all the underlying topology of the network (see [GP01] for an overview).

In this spirit, Kannan, Naor, and Rudich [KNR88] have proposed to label each node x of a planar graph by a pair $(x, P(x))$ where $P(x)$ is a set of at most three neighbors of x such that x and y are adjacent if and only if either $y \in P(x)$ or $x \in P(y)$. The nodes being represented by a unique integer taken from $\{1, \ldots, n\}$, this yields to a representation of every planar graph with local labels of $4 \log n$ bits and supporting constant time adjacency queries. A similar $4 \log n$-bit labeling scheme has been obtained by Chrobak and Eppstein [EC91]. This naturally leads to the following definition.

Definition 1 *A graph family $\mathcal{F}$ has a $f(n)$-*adjacency labeling scheme *if there exists a pair (L, A) of functions, called respectively* labeling *and* adjacency decoder*, such that for every n-node graph G of $\mathcal{F}$, and for all nodes x, y of G,*

1. *$L(x, G)$ is a binary string of length at most $f(n)$;*
2. *$A(L(x, G), L(y, G)) = 1$ if and only if x and y are adjacent in G.*

As defined in [KNR88], a graph family $\mathcal{F}$ has an *implicit representation* if $\mathcal{F}$ has an $O(\log n)$-adjacency labeling scheme[2]. Several graph families support an implicit representation: some intersection graph families, like interval graphs, paths graphs (intersection of paths in a tree), circle graphs (intersection of chords in a circle), permutation graphs, circular-arc graphs, graphs with bounded interval number ... (see Golumbic for definitions of these families [Gol80]), but also graphs with bounded arboricity [KNR88]. The *arboricity* of a graph G is defined[3] to be

$$k \;=\; \max_{H} \left\lceil \frac{|E(H)|}{|V(H)| - 1} \right\rceil$$

where H range over all possible induced subgraphs of G containing at least two nodes. Arboricity is related to *degeneracy* of graphs. A graph is *d-degenerated* if it has a node of degree at most d whose deletion leave the graph d-degenerated. The degeneracy is the smallest integer d that makes the graph d-degenerated. It is easy to see that every graph of arboricity k and degeneracy d satisfies $d/2 \leqslant k \leqslant d$. Moreover, $k \leqslant \Delta/2 + 1$ if the graph is of maximum degree Δ. Many families of graphs have bounded arboricity: bounded

[2] In this paper all the logarithms are in based two.

[3] Usually, there is no "ceiling" in the definition of arboricity. However, by the Nash-Williams' Theorem quoted hereafter, there is no restrictions to define integral arboricity.

degree graphs, bounded genus graphs, bounded tree-width graphs, geometric neighborhood graphs [MTV91], bounded pagenumber graphs, and bounded degeneracy graphs.

The Nash-Williams' Theorem [CMW$^+$94] states that the edges of a k-arboricity graph can be decomposed into k forests. Finding a decomposition in exactly k forests is non-trivial. There is an $O(nm \log n)$ algorithm [GW92] using matroid technique, that is an $O(kn^2 \log n)$ time algorithm since the number of edges m is at most $k(n-1)$. A simple $O(kn)$ algorithm exists for decomposition into $2k-1$ forests [AMZ97].

In [KNR88] it is shown that graphs of arboricity bounded by k have a $(k+1)\log n$-adjacency labeling scheme. Combining several recent results we improve in this note implicit representations based on arboricity.

In Section 2 we show that graphs with large arboricity k support a $k \log(n/k)$-bit labeling scheme with constant time queries, and we show that the bound on the label length is tight. In Section 3 we show a labeling with $k \log n$-bit labels and supporting $O(k)$ adjacency queries, providing a $3 \log n$-bit labeling for planar graphs and a $2 \log n$-bit labeling for series-parallel graphs. Finally, in section 4, we show that k-decomposable graphs (or partial k-trees) have a $k \log n$-bit labeling scheme and that for this family we prove a $\Omega(k + \log n)$ lower bound on the label length. We leave in Section 5 as an open problem the design of a $3 \log n$-bit labeling scheme for all genus 1 graphs, i.e., all the graphs having an embedding on the torus.

2 A Scheme for Large Arboricity Graphs

In this section we consider the family of n-node graphs whose arboricity is bounded by some function $k(n)$.

Let G be such a graph, and w.l.o.g. assume that the node set is $\{1, \ldots, n\}$. By the Nash-Williams' Theorem, the edges of G can be decomposed into $k = k(n)$ forests. Let assume that the forest decomposition is given, namely $F_1, \ldots, F_k$. Clearly, two nodes x and y are adjacent in G if and only if there is a forest F_i in which x is adjacent to y. We assume that each tree of each F_i is rooted at some arbitrary node.

As done in the scheme of Kannan, Naor and Rudich [KNR88] presented previously, we associate to each node x of G the set $P(x) = \bigcup_{i=1}^{k} \{y \text{ parent of } x \text{ in } F_i\}$. The label of x is set to $L(x, G) = (x, P(x))$. The adjacency decoder is defined by the following two membership queries: $A((x, P(x)), (y, P(y))) = 1$ if and only if $y \in P(x)$ or $x \in P(y)$.

Observe that $P(x) \subset \{1, \ldots, n\}$ and that $|P(x)| \leqslant k$. Let $B = \lceil \log \binom{n}{k} \rceil$ be the number of bits for coding a k-element subset of $\{1, \ldots, n\}$. The label $L(x, G) = (x, P(x))$ might be coded by a binary string of $\lceil \log n \rceil + B$ bits.

In [BM99], Brodnik and Munro have showed that every k-element subset of $\{1, \ldots, n\}$ can be encoded into a data-structure (polynomially-time constructible) of $B + O(B/\log^{(3)} n)$ bits supporting constant time membership queries under a standard word-RAM computational model. So, the label of x can be encoded with $\log n + B + o(B)$ bits such that the adjacency decoder A can be computed in constant time. We have $(n/k)^k \leqslant \binom{n}{k} \leqslant (ne/k)^k$, thus $B = k\log(n/k) + O(k)$. On the other hand $B \geqslant \log n$ for every $k \notin \{0, n\}$, thus the label length for x is at most $k\log(n/k) + O(k + \log n) + o(k\log(n/k)$.

Disregarding the $o(k\log(n/k)$ term when n is large enough, we have:

Theorem 1 *The family of bounded k-arboricity graphs have an asymptotic $k\log(n/k) + O(k + \log n)$-adjacency labeling scheme with constant time decoder. Moreover, all the labels are polynomially-time constructible.*

Actually the label length upper bound given in Theorem 1 is tight since:

Theorem 2 *For all n and k, every $f(n,k)$-adjacency labeling scheme on the family of bounded k-arboricity graphs requires $f(n,k) > \frac{1}{2}k\log(n/k) - \frac{1}{2}k - 1$.*

Proof. As seen in the introduction, since every graph is of maximum degree $n-1$, one can restrict our attention to arboricity $k \leqslant \lfloor (n-1)/2 \rfloor + 1 = \lceil n/2 \rceil$. Let $p = \lfloor n/2 \rfloor$ and $q = \lceil n/2 \rceil$. We consider the family $\mathcal{F}$ of labeled graphs defined as follows. The node set is $X \cup Y$ with $X = \{1, \ldots, p\}$ and $Y = \{p+1, \ldots, p+q\}$. We have $|X \cup Y| = p + q = n$, and $k \leqslant q \leqslant \lceil n/2 \rceil$. The edges link the nodes of X to the nodes of Y only (so the graphs of $\mathcal{F}$ are bipartite) such a way that the degree of every node in X is exactly k.

Consider any $G \in \mathcal{F}$. We can partition the edges of G such that, for each $x \in X$, the k edges incident to x belong to a distinct forest. Since G is bipartite, every cycle in some forest would implies that two edges incident to a certain $x_0 \in X$ belongs to this cycle, contradicting the edge assignment of x_0's edges. It follows that G is of arboricity bounded by k.

Every graph in $\mathcal{F}$ has n nodes, and the number of graphs in $\mathcal{F}$ is $|\mathcal{F}| = \binom{q}{k}^p$. Let (L, A) be any $f(n,k)$-adjacency labeling scheme on $\mathcal{F}$, for some function $f(n,k)$. Consider the set $\mathcal{L} = \{(L(1,G), \ldots, L(n,G)) \mid G \in \mathcal{F}\}$. We have, $|\mathcal{L}| \geqslant |\mathcal{F}|$. Indeed, if $|\mathcal{L}| < |\mathcal{F}|$ then there exists two distinct graphs $G, H \in \mathcal{F}$ such that $L(v,G) = L(v,H)$ for all node $v \in \{1, \ldots, n\}$, that is impossible. On the other hand, if all the labels are binary strings of length at most $f(n,k)$ bits, then $|\mathcal{L}| \leqslant S^n$ where $S = \sum_{i=0}^{f(n,k)} 2^i = 2^{f(n,k)+1} - 1$ is the number of binary strings of length at most $f(n,k)$. Thus we have

$$|\mathcal{F}| = \binom{q}{k}^p \leqslant |\mathcal{L}| < 2^{(f(n,k)+1)n} . \tag{1}$$

Because $\binom{q}{k} \geqslant (q/k)^k$, we have

$$\log \binom{q}{k}^p = \left\lfloor \frac{n}{2} \right\rfloor \log \binom{\lceil n/2 \rceil}{k} \geqslant \frac{1}{2} nk \log\left(\frac{n}{2k}\right) .$$

This implies that $f(n,k) > \frac{1}{2}k\log(n/k) - \frac{1}{2}k - 1$, that completes the proof. □

3 An Improved Labeling Scheme

In this section we consider the family of n-node graphs whose arboricity is bounded by some constant k. The next result is more compact than Theorem 1.

Theorem 3 *The family of bounded k-arboricity graphs have an asymptotic $k \log n$-adjacency labeling scheme with a $O(k)$ time decoder. Moreover, all the labels can be constructed in $O(kn^2 \log n)$ time, and in the case of planar graphs ($k = 3$) the time can be reduced to $O(n)$.*

Proof. Let G be a graph of arboricity at most k. By the Nash-Williams' Theorem, the edges of G can be decomposed into k forests. Firstly, let assume that the forest decomposition is given, and let $F_1, \ldots, F_k$ be these forests, each one spanning the nodes of G (so eventually F_i's may contain zero degree node). Clearly, x and y are adjacent in G if and only if there is a forest F_i in which x and y are adjacent. For each $i \in \{1, \ldots, k\}$, we apply the following construction: For each connected component of F_i pick an arbitrary node x_i^j and form a tree T_i rooted at a new node linked to all the x_i^j's node of F_i.

In [AR01, KM01], it is showed that n-node trees have an $\log n + O(\sqrt{\log n}\,)$-adjacency labeling scheme with constant time decoder (under a standard word-RAM computational model). Moreover the time to construct the n labels is $O(n)$. So, we associate such a binary label $\ell(x, T_i)$ to each node x of T_i as done in [AR01, KM01]. The length of the label is $\log |V(T_i)| + O(\sqrt{\log |V(T_i)|}\,)$ that is $\log n + O(\sqrt{\log n}\,)$ bits since T_i has $n+1$ nodes. Finally, to each node x of G we associate the k-tuple $L(x, G) = (\ell(x, T_1), \ldots, \ell(x, T_k))$ of length $k \log n + O(k\sqrt{\log n}\,)$. To test adjacency it suffices to test adjacency in each component of the labels. Observe that once the forests have been constructed it takes $O(kn)$ time to set all the labels, and $O(k)$ time to decode adjacency.

There is an $O(kn^2 \log n)$ time algorithm to find a decomposition of G into k forests [GW92]. For planar graphs ($k = 3$) there is an $O(n \log n)$ time algorithm [GL98]. We can improved this result by the use of *realizer* of plane triangulation[4], notion introduced by [Sch90] for grid embedding of planar graphs (a similar linear time algorithm can be founded in [Epp94]). A *realizer* of a plane triangulation is a particular partition of the interior edges in three sets of directed edges: T_1, T_2, T_3. The three sets are computable in $O(n)$ time [Epp94], and [Sch90] proved that each T_i is a directed tree including all interior nodes and exactly one exterior node distinct for each i. So, three edge-disjoint forests can be constructed in $O(n)$ time for every plane triangulation. Since in linear time one can triangulate and embed on the plane any planar graph G, it follows that the partition of the edges of G into three forests can be done in $O(n)$. This completes the proof. □

[4]A plane triangulation is a planar graph embedded on the plane whose all its faces are triangles.

4 An Improved Scheme for k-Decomposable Graphs

A graph G is *k-decomposable* if either G has $k+1$ or fewer nodes or there is a subgraph S of G with at most k nodes such that $G-S$ augmented by S with completely connected nodes is k-decomposable [AP87]. In [KNR88], it was showed that k-decomposable graphs have an asymptotic $k\log_{3/2} n \approx 1.7k\log n$-bit labeling scheme, observing that S can be chosen such that $G-S$ can be partitioned in two sets of nodes, A and B, with no edges linking A to B and with $|A|,|B| \leqslant 2|V(G)|/3$. The scheme we proposed here is based on a characterization of k-decomposable graphs in terms of partial k-trees and uses $k\log n$-bit labels.

The family of *k-trees* is the set of all the graphs obtained as follows: a complete graph with k nodes is a k-tree, and a k-tree with $n+1$ nodes is obtained by adding a node connected with k nodes of a complete subgraph of an n-node k-tree [BP69]. This definition clearly implies that k-trees are k-degenerated, and thus have arboricity bounded by k. Actually, this bound is exact for every k-tree with n nodes and n large enough, since the number of edges is $m = kn - k(k+1)/2$, and thus the arboricity must be at least $m/(n-1) \geqslant k - O(k^2/n)$.

Arnborg and Proskurowski showed in [AP87] that k-decomposable graphs are exactly the families of *partial k-trees*, i.e., a subgraph of a k-tree with the same number of nodes. Thus if we are able to find in t time a k-tree H whose G is a partial k-tree then one can apply Theorem 3, and one can construct in $O(kn+t)$ time a $k\log n$-adjacency labeling scheme for G. Unfortunately, the embedding of G into a k-tree is NP-complete in general [ACP87], even for graphs of degree bounded by 9 [BT97]. However Bodlaender has constructed a linear time algorithm to embed, if possible, G into a k-tree [Bod96]. His algorithm has a hidden constant at least exponential in k. An alternative solution is to use the algorithm of Arnborg, Corneil, and Proskurowski [ACP87] that runs in $O(n^{k+2})$ time. The smallest k such that G is a partial k-tree is also called the *tree-width* of G (see [Bod98] for a survey).

Theorem 4 *For every constant k, the family of partial k-trees (or the family of tree-width at most k graphs) has an asymptotic $k\log n$-adjacency labeling scheme computable in linear time and supporting constant time queries. Moreover, for every $k \leqslant n/2$, every $f(n,k)$-adjacency labeling scheme on the family of partial k-trees requires $f(n,k) = \Omega(k+\log n)$.*

Proof. The upper bound holds from the above discussion. Let us prove the lower bound. Consider the family $\mathcal{K}$ of all subgraphs of a complete graphs of $k+1$ nodes. Every graph of $\mathcal{K}$ is a partial k-tree, and its arboricity is at most k. By Theorem 2, every $f_1(n_1,k_1)$-adjacency labeling scheme on $\mathcal{K}$ requires $f_1(n_1,k_1) = \Omega(k_1\log(n_1/k_1))$ that is $\Omega(k)$ for $n_1 = k+1$ and $k_1 = k$. Consider now the family $\mathcal{T}$ of labeled n-node trees, a sub-family of partial k-trees. By the Cayley's formula, $|\mathcal{T}| = n^{n-2}$. Using Eq. 1 in the proof of Theorem 2, for every $f_2(n,k)$-adjacency labeling scheme on $\mathcal{T}$, we must have:

$$|\mathcal{T}| = n^{n-2} \ < \ 2^{(f_2(n,k)+1)n} \quad \Rightarrow \quad f_2(n,k) = \Omega(\log n)\ .$$

Therefore, every $f(n,k)$-adjacency labeling scheme on the family of n-node k-trees requires $f(n,k) \geqslant \min\{f_1, f_2\} = \Omega(k + \log n)$ completing the proof. $\Box$

An application of Theorem 4 is that series-parallel graphs (that are partial 2-trees), have a $2\log n$-adjacency labeling scheme computable in linear time.

5 Open Problem

We show in this note how to construct in linear time a $3\log n$-adjacency labeling scheme for planar graphs, and more generally a $k\log n$-bit labeling for bounded k-arboricity graphs. To conclude we leave as an open problem the problem to prove or disprove that the family of graphs having an embedding on a torus (genus 1 graphs) support an asymptotic $3\log n$-adjacency labeling scheme. Theorem 3 gives only a $6\log n$-bit labeling since bounded g genus graphs are $\lfloor 6 + 12(g-1)/n \rfloor$-degenerated (they have at most $3n + 6(g-1)$ edges). Thus, genus 1 graphs are 6-degenerated and of arboricity at most 6.

Acknowledgments: The author is thankful to André Raspaud for fruitful discussions about arboricity.

References

[ACP87] Stefan Arnborg, Derek G. Corneil, and Andrzej Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM Journal Algebraic Discrete Math.*, 8:277–284, 1987.

[AMZ97] Srinivasa R. Arikati, Anil Maheshwari, and Christos D. Zaroliagis. Efficient computation of implicit representations of sparse graphs. *Discrete Applied Mathematics*, 78:1–16, 1997.

[AP87] Stefan Arnborg and Andrzej Proskurowski. Characterization and recognition of partial 3-trees. *SIAM Journal Algebraic Discrete Math.*, 7(2):302–314, April 1987.

[AR01] Stephen Alstrup and Theis Rauhe. Improved labeling scheme for ancestor queries. In 13^{th} *Symposium on Discrete Algorithms (SODA)*. ACM-SIAM, January 2001. To appear.

[BM99] Andrej Brodnik and J. Ian Munro. Membership in constant time and almost-minimum space. *SIAM Journal on Computing*, 28(5):1627–1640, May 1999.

[Bod96] Hans Leo Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25:1305–1317, 1996.

[Bod98] Hans Leo Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1-2):1–45, 1998.

[BP69] L. W. Beineke and R. E. Pippert. The number of labeled k-dimensional trees. *Journal of Combinatorial Theory*, 6:200–205, 1969.

[BT97] Hans Leo Bodlaender and Dimitrios M. Thilikos. Treewidth for graphs with small chordality. *Discrete Applied Mathematics*, 79:45–61, 1997.

[CLL01] Yi-Ting Chiang, , Ching-Chi Lin, and Hsueh-I Lu. Orderly spanning trees with applications to graph encoding and graph drawing. In 12^{th} *Symposium on Discrete Algorithms (SODA)*, pages 506–515. ACM-SIAM, January 2001.

[CMW+94] Boliong Chen, Makoto Matsumoto, Jian Fang Wang, Zhong Fu Zhang, and Jian Xun Zhang. A short proof of nash-williams' theorem for arboricity of a graph. *Graphs and Combinatorics*, 10(1):27–28, 1994.

[EC91] David Eppstein and Marek Chrobak. Planar orientations with low out-degree and compaction of adjacency. *Theoretical Computer Science*, 86(2):243–266, September 1991.

[Epp94] David Eppstein. Arboricity and bipartite subgraph listing algorithms. *Information Processing Letters*, 51(4):207–211, 1994.

[GL98] Roberto Grossi and Elena Lodi. Simple planar graph partition into three forests. *Discrete Applied Mathematics*, 84:121–132, 1998.

[Gol80] Martin Charles Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, Harcourt Brace Jovanovich, Academic Press edition, 1980.

[GP01] Cyril Gavoille and David Peleg. Compact and localized distributed data structures. Research Report RR-1261-01, LaBRI, University of Bordeaux, 351, cours de la Libération, 33405 Talence Cedex, France, August 2001. To appear in Journal of Distributed Computing for the PODC 20-Year Special Issue.

[GW92] Harold N. Gabow and Herbert H. Westermann. Forests, frames, and games: algorithms for matroid sums and applications. *Algorithmica*, 7:465–497, 1992.

[KM01] Haim Kaplan and Tova Milo. Short and simple labels for small distances and other functions. In 7^{th} *International Workshop on Algorithms and Data Structures (WADS)*, volume 2125 of Lecture Notes in Computer Science, pages 32–40. Springer, August 2001.

[KNR88] Sampath Kannan, Moni Naor, and Steven Rudich. Implicit representation of graphs. In 20^{th} *Annual ACM Symposium on Theory of Computing (STOC)*, pages 334–343, Chicago, IL, May 1988.

[MTV91] Gary L. Miller, S.-H. Teng, and S. A. Vavasis. A unified geometric approach to graph separators. In 32^{nd} *Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 538–547. IEEE Computer Society Press, 1991.

[Sch90] Walter Schnyder. Embedding planar graphs on the grid. In 1^{st} *Symposium on Discrete Algorithms (SODA)*, pages 138–148. ACM-SIAM, 1990.