SIAM J. Comput.
Vol. 30, No. 3, pp. 838–846

A FAST GENERAL METHODOLOGY FOR INFORMATION-THEORETICALLY OPTIMAL ENCODINGS OF GRAPHS*

XIN HE†, MING-YANG KAO‡, and HSUEH-I LU§

Abstract. We propose a fast methodology for encoding graphs with information-theoretically minimum numbers of bits. Specifically, a graph with property π is called a π-*graph*. If π satisfies certain properties, then an n-node m-edge π-graph G can be encoded by a binary string X such that (1) G and X can be obtained from each other in $O(n \log n)$ time, and (2) X has at most $\beta(n)+o(\beta(n))$ bits for any continuous superadditive function $\beta(n)$ so that there are at most $2^{\beta(n)+o(\beta(n))}$ distinct n-node π-graphs. The methodology is applicable to general classes of graphs; this paper focuses on planar graphs. Examples of such π include all conjunctions over the following groups of properties: (1) G is a planar graph or a plane graph; (2) G is directed or undirected; (3) G is triangulated, triconnected, biconnected, merely connected, or not required to be connected; (4) the nodes of G are labeled with labels from $\{1, \ldots, \ell_1\}$ for $\ell_1 \leq n$; (5) the edges of G are labeled with labels from $\{1, \ldots, \ell_2\}$ for $\ell_2 \leq m$; and (6) each node (respectively, edge) of G has at most $\ell_3 = O(1)$ self-loops (respectively, $\ell_4 = O(1)$ multiple edges). Moreover, ℓ_3 and ℓ_4 are not required to be $O(1)$ for the cases of π being a plane triangulation. These examples are novel applications of small cycle separators of planar graphs and are the only nontrivial classes of graphs, other than rooted trees, with known polynomial-time information-theoretically optimal coding schemes.

Key words. data compression, graph encoding, planar graphs, triconnected graphs, biconnected graphs, triangulations, cycle separators

AMS subject classifications. 05C10, 05C30, 05C78, 05C85, 68R10, 65Y25, 94A15

PII. S0097539799359117

1. Introduction.

Let G be a graph with n nodes and m edges. This paper studies the problem of *encoding* G into a binary string X with the requirement that X can be *decoded* to reconstruct G. We propose a fast methodology for designing a coding scheme such that the bit count of X is information-theoretically optimal. Specifically, a function $\beta(n)$ is *superadditive* if $\beta(n_1)+\beta(n_2) \leq \beta(n_1+n_2)$. A function $\beta(n)$ is *continuous* if $\beta(n + o(n)) = \beta(n) + o(\beta(n))$. For example, $\beta(n) = n^c \log^d n$ is continuous and superadditive, for any constants $c \geq 1$ and $d \geq 0$. The continuity and superadditivity are closed under additions. A graph with property π is called a π-*graph*. If π satisfies certain properties, then we can obtain an X such that (1) G and X can be computed from each other in $O(n \log n)$ time, and (2) X has at most $\beta(n)+o(\beta(n))$ bits for any continuous superadditive function $\beta(n)$ so that there are at most $2^{\beta(n)+o(\beta(n))}$ distinct n-node m-edge π-graphs. The methodology is applicable to general classes of graphs; this paper focuses on planar graphs.

*Received by the editors July 22, 1999; accepted for publication (in revised form) January 31, 2000; published electronically August 9, 2000. A preliminary version appeared in *Proceedings of the 7th Annual European Symposium on Algorithms*, Lecture Notes in Comput. Sci. 1643, Springer-Verlag, New York, 1999, pp. 540–549.

http://www.siam.org/journals/sicomp/30-3/35911.html

†Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY 14260 (xinhe@cse.buffalo.edu).

‡Department of Computer Science, Yale University, New Haven, CT 06250 (kao-ming-yang@cs.yale.edu). This author's research was supported in part by NSF grant CCR-9531028.

§Institute of Information Science, Academia Sinica, Taipei 115, Taiwan, R.O.C. (hil@iis.sinica.edu.tw). Part of this work was performed at the Department of Computer Science and Information Engineering, National Chung-Cheng University, Chia-Yi 621, Taiwan, R.O.C. This author's research was supported in part by NSC grant NSC-89-2213-E-001-034.

A *conjunction* over k groups of properties is a boolean property $\pi_1 \wedge \cdots \wedge \pi_k$, where π_i is a property in the ith group for each $i = 1, \ldots, k$. Examples of suitable π for our methodology include every conjunction over the following groups:

F1. G is a planar graph or a plane graph.
F2. G is directed or undirected.
F3. G is triangulated, triconnected, biconnected, merely connected, or not required to be connected.
F4. The nodes of G are labeled with labels from $\{1, \ldots, \ell_1\}$ for $\ell_1 \leq n$.
F5. The edges of G are labeled with labels from $\{1, \ldots, \ell_2\}$ for $\ell_2 \leq m$.
F6. Each node of G has at most $\ell_3 = O(1)$ self-loops.
F7. Each edge of G has at most $\ell_4 = O(1)$ multiple edges.

Moreover, ℓ_3 and ℓ_4 are not required to be $O(1)$ for the cases of π being a plane triangulation. For instance, π can be the property of being a directed unlabeled biconnected simple plane graph. These examples are novel applications of small cycle separators of planar graphs [12, 11]. Note that the rooted trees are the only other nontrivial class of graphs with a known polynomial-time information-theoretically optimal coding scheme, which encodes a tree as nested parentheses using $2(n-1)$ bits in $O(n)$ time.

Previously, Tutte proved that there are $2^{\beta(m)+o(\beta(m))}$ distinct m-edge plane triangulations where $\beta(m) = (\frac{8}{3} - \log_2 3)m + o(m) \approx 1.08m + o(m)$ [17] and that there are $2^{2m+o(n)}$ distinct m-edge n-node triconnected plane graphs that may be nonsimple [18]. Turán [16] used $4m$ bits to encode a plane graph G that may have self-loops. Keeler and Westbrook [10] improved this bit count to $3.58m$. They also gave coding schemes for several families of plane graphs. In particular, they used $1.53m$ bits for a triangulated simple G, and $3m$ bits for a connected G free of self-loops and degree-1 nodes. For a simple triangulated G, He, Kao, and Lu [5] improved the bit count to $\frac{4}{3}m + O(1)$. For a simple G that is triconnected and thus free of degree-1 nodes, they [5] improved the bit count to at most $2.835m$ bits. This bit count was later reduced to at most $\frac{3\log_2 3}{2}m + O(1) \approx 2.378m + O(1)$ by Chuang et al. [2]. These coding schemes all take linear time for encoding and decoding, but their bit counts are not information-theoretically optimal. For labeled planar graphs, Itai and Rodeh [6] gave an encoding of $\frac{3}{2}n \log n + O(n)$ bits. For unlabeled general graphs, Naor [14] gave an encoding of $\frac{1}{2}n^2 - n \log n + O(n)$ bits.

For applications that require query support, Jacobson [7] gave a $\Theta(n)$-bit encoding for a connected and simple planar graph G that supports traversal in $\Theta(\log n)$ time per node visited. Munro and Raman [13] improved this result and gave schemes to encode binary trees, rooted ordered trees, and planar graphs. For a general planar G, they used $2m + 8n + o(m + n)$ bits while supporting adjacency and degree queries in $O(1)$ time. Chuang et al. [2] reduced this bit count to $2m + (5 + \frac{1}{k})n + o(m + n)$ for any constant $k > 0$ with the same query support. The bit count can be further reduced if only $O(1)$-time adjacency queries are supported, or if G is simple, triconnected, or triangulated [2]. For certain graph families, Kannan, Naor and Rudich [8] gave schemes that encode each node with $O(\log n)$ bits and support $O(\log n)$-time testing of adjacency between two nodes. For dense graphs and complement graphs, Kao, Occhiogrosso, and Teng [9] devised two compressed representations from adjacency lists to speed up basic graph search techniques. Galperin and Wigderson [4] and Papadimitriou and Yannakakis [15] investigated complexity issues arising from encoding a graph by a small circuit that computes its adjacency matrix.

Section 2 discusses the general encoding methodology. Sections 3 and 4 use the methodology to obtain information-theoretically optimal encodings for various classes of planar graphs. Section 5 concludes the paper with some future research directions.

2. The encoding methodology. Let $|X|$ be the number of bits in a binary string X. Let $|G|$ be the number of nodes in a graph G. Let $|S|$ be the number of elements, counting multiplicity, in a multiset S.

FACT 1 (see [1, 3]). *Let $X_1, X_2, \ldots, X_k$ be $O(1)$ binary strings. Let $n = |X_1| + |X_2| + \cdots + |X_k|$. Then there exists an $O(\log n)$-bit string χ, obtainable in $O(n)$ time, such that given the concatenation of $\chi, X_1, X_2, \ldots, X_k$, the index of the first symbol of each X_i in the concatenation can be computed in $O(1)$ time.*

Let $X_1 + X_2 + \cdots + X_k$ denote the concatenation of $\chi, X_1, X_2, \ldots, X_k$ as in Fact 1. We call χ the *auxiliary binary string* for $X_1 + X_2 + \cdots + X_k$.

A graph with property π is called a π-*graph*. Whether two π-graphs are *distinct* or *indistinct* depends on π. For example, let G_1 and G_2 be two topologically non-isomorphic plane embeddings of the same planar graph. If π is the property of being a planar graph, then G_1 and G_2 are two indistinct π-graphs. If π is the property of being a planar embedding, then G_1 and G_2 are two distinct π-graphs. Let α be the number of distinct n-node π-graphs. Clearly it takes $\lceil \log_2 \alpha \rceil$ bits to differentiate all n-node π-graphs. Let $\mathrm{index}_\pi(G)$ be an $\lceil \log_2 \alpha \rceil$-bit indexing scheme of the α distinct π-graphs.

Let G_0 be an input n_0-node π-graph. Let $\lambda = \log\log\log(n_0)$. The encoding algorithm $\mathrm{encode}_\pi(G_0)$ is merely a function call $\mathrm{code}_\pi(G_0, \lambda)$, where the recursive function $\mathrm{code}_\pi(G, \lambda)$ is defined as follows:

```
function code_π(G, λ)
{
  if |G| = O(1) or |G| ≤ λ then
    return index_π(G)
  else
  {
    compute π-graphs G_1, G_2, and a string X, from which G can be recovered;
    return code_π(G_1, λ) + code_π(G_2, λ) + X;
  }
}
```

Clearly, the code returned by algorithm $\mathrm{encode}_\pi(G_0)$ can be decoded to recover G_0. For notational brevity, if it is clear from the context, the code returned by algorithm $\mathrm{encode}_\pi(G_0)$ (respectively, function $\mathrm{code}_\pi(G, \lambda)$) is also denoted $\mathrm{encode}_\pi(G_0)$ (respectively, $\mathrm{code}_\pi(G, \lambda)$).

Function $\mathrm{code}_\pi(G, \lambda)$ *satisfies the separation property* if there exist two constants c and r, where $0 \le c < 1$ and $r > 1$, such that the following conditions hold:

P1. $\max(|G_1|, |G_2|) \le |G|/r$.

P2. $|G_1| + |G_2| = |G| + O(|G|^c)$.

P3. $|X| = O(|G|^c)$.

Let $f(|G|)$ be the time required to obtain $\mathrm{index}_\pi(G)$ and G from each other. Let $g(|G|)$ be the time required to obtain G_1, G_2, X from G, and vice versa.

THEOREM 2.1. *Assume that function $\mathrm{code}_\pi(G, \lambda)$ satisfies the separation property and that there are at most $2^{\beta(n)+o(\beta(n))}$ distinct n-node π-graphs for some continuous superadditive function $\beta(n)$.*

1. $|\mathrm{encode}_\pi(G_0)| \le \beta(n_0) + o(\beta(n_0))$ *for any n_0-node* π-graph G_0.
2. *If $f(n) = 2^{n^{O(1)}}$ and $g(n) = O(n)$, then G_0 and $\mathrm{encode}_\pi(G_0)$ can be obtained from each other in $O(n_0 \log n_0)$ time.*

Proof. The theorem holds trivially if $n_0 = O(1)$. For the rest of the proof we assume $n_0 = \omega(1)$, and thus $\lambda = \omega(1)$. Many graphs may appear during the execution of $\text{encode}_\pi(G_0)$. These graphs can be organized as nodes of a binary tree T rooted at G_0, where (i) if G_1 and G_2 are obtained from G by calling $\text{code}_\pi(G, \lambda)$, then G_1 and G_2 are the children of G in T, and (ii) if $|G| \le \lambda$, then G has no children in T. Further consider the multiset S consisting of all graphs G that are nodes of T. We partition S into $\ell+1$ multisets $S(0), S(1), S(2), \ldots, S(\ell)$ as follows. $S(0)$ consists of the graphs G with $|G| \le \lambda$. For $i \ge 1$, $S(i)$ consists of the graphs G with $r^{i-1}\lambda < |G| \le r^i\lambda$. Let $G_0 \in S(\ell)$, and thus set $\ell = O(\log \frac{n_0}{\lambda})$.

Define $p = \sum_{H \in S(0)} |H|$. We first show

$$|S(i)| < \frac{p}{r^{i-1}\lambda} \tag{1}$$

for every $i = 1, \ldots, \ell$. Let G be a graph in $S(i)$. Let $S(0, G)$ be the set consisting of the leaf descendants of G in T; for example, $S(0, G_0) = S(0)$. By condition P2, $|G| \le \sum_{H \in S(0,G)} |H|$. By condition P1, no two graphs in $S(i)$ are related in T. Therefore $S(i)$ contains at most one ancestor of H in T for every graph H in $S(0)$. It follows that $\sum_{G \in S(i)} |G| \le \sum_{G \in S(i)} \sum_{H \in S(0,G)} |H| \le p$. Since $|G| > r^{i-1}\lambda$ for every G in $S(i)$, inequality (1) holds.

Statement 1. Suppose that the children of G in T are G_1 and G_2. Let $b(G) = |X| + |\chi|$, where χ is the auxiliary binary string for $\text{code}_\pi(G_1, \lambda) + \text{code}_\pi(G_2, \lambda) + X$. Let $q = \sum_{i \ge 1} \sum_{G \in S(i)} b(G)$. Then $|\text{encode}_\pi(G_0)| = q + \sum_{H \in S(0)} |\text{code}_\pi(H, \lambda)| \le q + \sum_{H \in S(0)} (\beta(|H|) + o(\beta(|H|)))$. By the superadditivity of $\beta(n)$, $|\text{encode}_\pi(G_0)| \le q + \beta(p) + o(\beta(p))$. Since $\beta(n)$ is continuous, Statement 1 can be proved by showing $p = n_0 + o(n_0)$ and $q = o(n_0)$ below.

By condition P3, $|X| = O(|G|^c)$. By Fact 1, $|\chi| = O(\log |G|)$. Thus, $b(G) = O(|G|^c)$, and

$$q = \sum_{i \ge 1} \sum_{G \in S(i)} O(|G|^c). \tag{2}$$

Now we regard the execution of $\text{encode}_\pi(G_0)$ as a process of growing T. Let $a(T) = \sum_{H \text{ is a leaf of } T} |H|$. At the beginning of the function call $\text{encode}_\pi(G_0)$, T has exactly one node G_0, and thus $a(T) = n_0$. At the end of the function call, T is fully expanded, and thus $a(T) = p$. By condition P2, during the execution of $\text{encode}_\pi(G_0)$, every function call $\text{code}_\pi(G, \lambda)$ with $|G| > \lambda$ increases $a(T)$ by $O(|G|^c)$. Hence

$$p = n_0 + \sum_{i \ge 1} \sum_{G \in S(i)} O(|G|^c). \tag{3}$$

Note that

$$\sum_{i \ge 1} \sum_{G \in S(i)} |G|^c \le \sum_{i \ge 1} (r^i\lambda)^c p/(r^{i-1}\lambda) = p\lambda^{c-1} r \sum_{i \ge 1} r^{(c-1)i} = p\lambda^{c-1} O(1) = o(p). \tag{4}$$

By (3) and (4), we have $p = n_0 + o(p)$, and thus $p = O(n_0)$. Therefore $\sum_{i \ge 1} \sum_{G \in S(i)} |G|^c = o(n_0)$. By (2) and (3), $p = n_0 + o(n_0)$ and $q = o(n_0)$, finishing the proof of Statement 1.

Statement 2. By conditions P1 and P2, $|H| = \Omega(\lambda)$ for every $H \in S(0)$. Since $\sum_{H \in S(0)} |H| = p = n_0 + o(n_0)$, $|S(0)| = O(n_0/\lambda)$. Together with (1), we know

$|S(i)| = O(\frac{n_0}{r^i\lambda})$ for every $i = 0, \ldots, \ell$. By the definition of $S(i)$, $|G| \leq r^i\lambda$ for every $i = 0, \ldots, \ell$. Therefore G_0 and $\text{encode}_\pi(G_0)$ can be obtained from each other in time

$$\frac{n_0}{\lambda} O\left(f(\lambda) + \sum_{1\leq i\leq \ell} r^{-i} g(r^i\lambda)\right).$$

Clearly $f(\lambda) = 2^{\lambda^{O(1)}} = 2^{o(\log\log n_0)} = o(\log n_0)$. Since $\ell = O(\log n_0)$ and $g(n) = O(n)$, $\sum_{1\leq i\leq \ell} r^{-i} g(r^i\lambda) = \sum_{1\leq i\leq \ell} \lambda = O(\lambda \log n_0)$, and Statement 2 follows. □

Sections 3 and 4 use Theorem 2.1 to encode various classes of graphs G. Section 3 considers plane triangulations. Section 4 considers planar graphs and plane graphs.

3. Plane triangulations. A *plane triangulation* is a plane graph, each of whose faces has size exactly 3. For any plane triangulation P with n nodes, m edges, and f faces, Euler's formula ensures that $n - m + f = 2$ even if P contains self-loops and multiple edges. One can then obtain $m = 3n - 6$. Therefore every n-node plane triangulation, simple or not, has exactly $3n - 6$ edges.

In this section, let π be an arbitrary conjunction over the following groups of properties of a plane triangulation G: F2, F6, and F7, where ℓ_3 and ℓ_4 are not required to be $O(1)$. Our encoding scheme is based on the next fact.

FACT 2 (see [12]). *Let H be an n-node m-edge undirected plane graph, each of whose faces has size at most d. We can compute a node-simple cycle C of H in $O(n + m)$ time such that*

- *C has at most $2\sqrt{dn}$ nodes; and*
- *the numbers of H's nodes inside and outside C are at most $2n/3$, respectively.*

Let G be a given n-node π-graph. Let G' be obtained from the undirected version of G by deleting the self-loops. Clearly each face of G' has size at most 4. Let C' be a cycle of G' having size at most $4\sqrt{n}$ guaranteed by Fact 2. Let C consist of the edges of G corresponding to the edges of C' in G'. Note that C is not necessarily a directed cycle if G is directed. Since G' does not have self-loops, $2 \leq |C| \leq 4\sqrt{n}$. If $\ell_4 \geq 2$, then $|C|$ can be 2. Let G_{in} (respectively, G_{out}) be the subgraph of G formed by C and the part of G inside (respectively, outside) C. Let x be an arbitrary node on C.

G_1 is obtained by placing a cycle C_1 of three nodes outside G_{in} and then triangulating the face between C_1 and G_{in} such that a particular node y_1 of C_1 has degree strictly lower than the other two. Clearly this is feasible even if $|C| = 2$. The edge directions of $G_1 - G_{\text{in}}$ can be arbitrarily assigned according to π.

G_2 is obtained from G_{out} by (1) placing a cycle C_2 of three nodes outside G_{out} and then triangulating the face between C_2 and G_{out} such that a particular node y_2 of C_2 has degree strictly lower than the other two, and (2) triangulating the face inside C by placing a new node z inside of C and then connecting it to each node of C by an edge. Note that (2) is feasible even if $|C| = 2$. Similarly, the edge directions of $G_2 - G_{\text{out}}$ can be arbitrarily assigned according to π.

Let u be a node of G. Let v be a node on the boundary $B(G)$ of the exterior face of G. Define $\text{dfs}(u, G, v)$ as follows. Let w be the counterclockwise neighbor of v on $B(G)$. We perform a depth-first search of G starting from v such that (1) the neighbors of each node are visited in the counterclockwise order around that node, and (2) w is the second visited node. A numbering is assigned the first time a node is visited. Let $\text{dfs}(u, G, v)$ be the binary number assigned to u in the above depth-first search. Let $X = \text{dfs}(x, G_1, y_1) + \text{dfs}(x, G_2, y_2) + \text{dfs}(z, G_2, y_2)$.

LEMMA 3.1.

1. *G_1 and G_2 are π-graphs.*
2. *There exists a constant $r > 1$ with* $\max(|G_1|, |G_2|) \le n/r$.
3. $|G_1| + |G_2| = n + O(\sqrt{n})$.
4. $|X| = O(\log n)$.
5. *G_1, G_2, X can be obtained from G in $O(n)$ time.*
6. *G can be obtained from G_1, G_2, X in $O(n)$ time.*

Proof. Statements 1–5 are straightforward by Fact 2 and the definitions of G_1, G_2, and X. Statement 6 is proved as follows. It takes $O(n)$ time to locate y_1 (respectively, y_2) in G_1 (respectively, G_2) by looking for the node with the lowest degree on $B(G_1)$ (respectively, $B(G_2)$). By Fact 1, it takes $O(1)$ time to obtain $\mathrm{dfs}(y_1, G_1, x)$, $\mathrm{dfs}(y_2, G_2, x)$, and $\mathrm{dfs}(y_2, G_2, z)$ from X. Therefore x and z can be located in G_1 and G_2 in $O(n)$ time by depth-first traversal. Now G_{in} can be obtained from G_1 by removing $B(G_1)$ and its incident edges. The cycle C in G_{in} is simply $B(G_{\mathrm{in}})$. Also, G_{out} can be obtained from G_2 by removing $B(G_2)$, z, and their incident edges. The C in G_{out} is simply the boundary of the face that encloses z and its incident edges in G_2. Since we know the positions of x in G_{in} and G_{out}, G can be obtained from G_{in} and G_{out} by fitting them together along C by aligning x. The overall time complexity is $O(n)$. □

THEOREM 3.2. *Let G_0 be an n_0-node π-graph. Then G_0 and* $\mathrm{encode}_\pi(G_0)$ *can be obtained from each other in $O(n_0 \log n_0)$ time. Moreover,* $|\mathrm{encode}_\pi(G_0)| \le \beta(n_0) + o(\beta(n_0))$ *for any continuous superadditive function $\beta(n)$ such that there are at most $2^{\beta(n)+o(\beta(n))}$ distinct n-node π-graphs.*

Proof. Since an n-node π-graph has $O(n)$ edges, there are at most $2^{O(n \log n)}$ distinct n-node π-graphs. Thus, there exists an indexing scheme $\mathrm{index}_\pi(G)$ such that $\mathrm{index}_\pi(G)$ and G can be obtained from each other in $2^{|G|^{O(1)}}$ time. The theorem follows from Theorem 2.1 and Lemma 3.1. □

4. Planar graphs and plane graphs. In this section, let π be an arbitrary conjunction over the following groups of properties of G: F1, F2, F3, F6, and F7. Clearly an n-node π-graph has $O(n)$ edges.

Let G be an input n-node π-graph. For the cases of π being a planar graph rather than a plane graph, let G be embedded first. Note that this is only for the encoding process to be able to apply Fact 2. At the base level, we still use the indexing scheme for π-graphs rather than the one for embedded π-graphs. As shown below, the decoding process does not require the π-graphs to be embedded.

Let G' be obtained from the undirected version of G by (1) triangulating each of its faces that has size more than 3 such that no additional multiple edges are introduced, and then (2) deleting its self-loops. Let C' be a cycle of G' guaranteed by Fact 2. Let C consists of (a) the edges of G corresponds to the edges of C' in G', and (b) the edges of C' that are added into G' by the triangulation. (C is not necessarily a directed cycle of a directed G.) Let G_C be the union of G and C. Let G_{in} (respectively, G_{out}) be the subgraph of G_C formed by C and the part of G_C inside (respectively, outside) C. Let $C = x_1 x_2 \cdots x_\ell x_{\ell+1}$, where $x_{\ell+1} = x_1$. By Fact 2, $\ell = O(\sqrt{n})$.

LEMMA 4.1. *Let H be an $O(n)$-node $O(n)$-edge graph. There exists an integer k with $n^{0.6} \le k \le n^{0.7}$ such that H does not contain any node of degree k or $k - 1$.*

Proof. Assume for a contradiction that such a k does not exist. It follows that the sum of degrees of all nodes in H is at least $(n^{0.6} + n^{0.7})(n^{0.7} - n^{0.6})/4 = \Omega(n^{1.4})$. This contradicts the fact that H has $O(n)$ edges. □

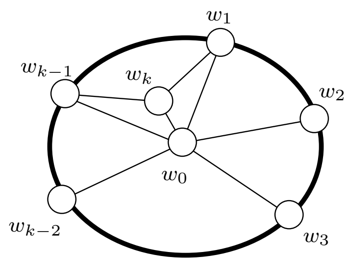

FIG. 1. *A k-wheel graph W_k.*

Let W_k, with $k \geq 3$, be a *k-wheel graph* defined as follows. As shown in Figure 1, W_k consists of $k+1$ nodes $w_0, w_1, w_2, \ldots, w_{k-1}, w_k$, where $w_1, w_2, \ldots, w_k, w_1$ form a cycle. w_0 is a degree-k node incident to each node on the cycle. Finally, w_1 is incident to w_{k-1}. Clearly W_k is triconnected. Also, w_1 and w_k are the only degree-4 neighbors of w_0 in W_k. Let k_1 (respectively, k_2) be an integer k guaranteed by Lemma 4.1 for G_{in} (respectively, G_{out}). Now we define G_1, G_2, and X as follows.

G_1 is obtained from G_{in} and a k_1-wheel graph W_{k_1} by adding an edge (w_i, x_i) for every $i = 1, \ldots, \ell$. Clearly for the case of π being a plane graph, G_1 can be embedded such that W_{k_1} is outside G_{in}, as shown in Figure 2(a). Thus, the original embedding of G_{in} can be obtained from G_1 by removing all nodes of W_{k_1}. The edge directions of $G_1 - G_{\text{in}}$ can be arbitrarily assigned according to π.

G_2 is obtained from G_{out} and a k_2-wheel graph W_{k_2} by adding an edge (w_i, x_i) for every $i = 1, \ldots, \ell$. Clearly for the case of π being a plane graph, G_2 can be embedded such that W_{k_2} is inside C, as shown in Figure 2(b). Thus, the original embedding of G_{out} can be obtained from G_2 by removing all nodes of W_{k_2}. The edge directions of $G_2 - G_{\text{out}}$ can be arbitrarily assigned according to π.

Let X be an $O(\sqrt{n})$-bit string which encodes k_1, k_2, and whether each edge (x_i, x_{i+1}) is an original edge in G, for $i = 1, \ldots, \ell$.

LEMMA 4.2.

1. *G_1 and G_2 are π-graphs.*
2. *There exists a constant $r > 1$ with* $\max(|G_1|, |G_2|) \leq n/r$.
3. $|G_1| + |G_2| = n + O(n^{0.7})$.
4. $|X| = O(\sqrt{n})$.
5. *G_1, G_2, X can be obtained from G in $O(n)$ time.*
6. *G can be obtained from G_1, G_2, X in $O(n)$ time.*

Proof. Since W_{k_1} and W_{k_2} are both triconnected, and each node of C has degree at least 3 in G_1 and G_2, statement 1 holds for each case of the connectivity of the input π-graph G. Statements 2–5 are straightforward by Fact 2 and the definitions of G_1, G_2, and X. Statement 6 is proved as follows. First of all, we obtain k_1 from X. Since G_{in} does not contain any node of degree k_1 or $k_1 - 1$, w_0 is the only degree-k_1 node in G_1. Therefore it takes $O(n)$ time to identify w_0 in G_1. w_{k_1} is the only degree-3 neighbor of w_0. Since $k_1 > \ell$, w_1 is the only degree-5 neighbor of w_0. w_2 is the common neighbor of w_0 and w_1 that is not adjacent to w_{k_1}. From now on, w_i, for each $i = 3, 4, \ldots, \ell$, is the common neighbor of w_0 and w_{i-1} other than w_{i-2}. Clearly, $w_1, w_2, \ldots, w_\ell$ and thus $x_1, x_2, \ldots, x_\ell$ can be identified in $O(n)$ time. G_{in} can now be obtained from G_1 by removing W_{k_1}. Similarly, G_{out} can be obtained from G_2 and X by deleting W_{k_1} after identifying $x_1, x_2, \ldots, x_\ell$. Finally, G_C can be recovered by fitting G_{in} and G_{out} together by aligning $x_1, x_2, \ldots, x_\ell$. Based on X, G can then be

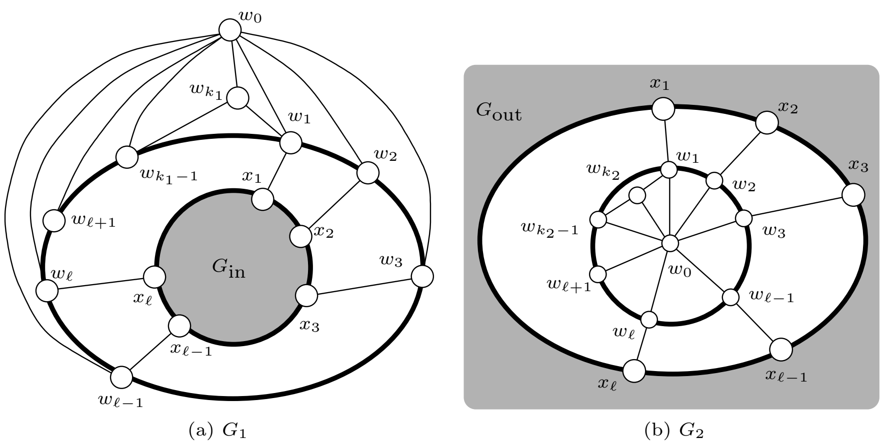

FIG. 2. *G_1 and G_2. The gray area of G_1 is G_{in}. The gray area of G_2 is G_{out}.*

obtained from G_C by removing the edges of C that are not originally in G. ∎

Remark. In the proof for statement 6 of Lemma 4.2, identifying the degree-k_1 node (and the k_1-wheel graph W_{k_1}) does not require the embedding for G_1. Therefore the decoding process does not require the π-graphs to be embedded. This is different from the proof of Lemma 3.1.

THEOREM 4.3. *Let G_0 be an n_0-node π-graph. Then G_0 and $\text{encode}_\pi(G_0)$ can be obtained from each other in $O(n_0 \log n_0)$ time. Moreover, $|\text{encode}_\pi(G_0)| \le \beta(n_0) + o(\beta(n_0))$ for any continuous superadditive function $\beta(n)$ such that there are at most $2^{\beta(n)+o(\beta(n))}$ distinct n-node π-graphs.*

Proof. Since there are at most $2^{O(n \log n)}$ distinct n-node π-graphs, there exists an indexing scheme $\text{index}_\pi(G)$ such that $\text{index}_\pi(G)$ and G can be obtained from each other in $2^{|G|^{O(1)}}$ time. The theorem follows from Theorem 2.1 and Lemma 4.2. ∎

5. Concluding remarks. For brevity, we left out F4 and F5 in sections 3 and 4. One can verify that Theorems 3.2 and 4.3 hold even if π is a conjunction over F1 through F7 including F4 and F5.

The coding schemes given in this paper require $O(n \log n)$ time for encoding and decoding. An immediate open question is whether one can encode some graphs other than rooted trees in $O(n)$ time using information-theoretically minimum number of bits. It would be of significance to determine whether the tight bound of the number of distinct π-graphs for each π is indeed continuous superadditive.

REFERENCES

[1] T. C. BELL, J. G. CLEARY, AND I. H. WITTEN, *Text Compression*, Prentice-Hall, Englewood Cliffs, NJ, 1990.

[2] R. C.-N. CHUANG, A. GARG, X. HE, M.-Y. KAO, AND H.-I LU, *Compact encodings of planar graphs via canonical orderings and multiple parentheses*, in Automata, Languages and Programming, 25th Colloquium, K. G. Larsen, S. Skyum, and G. Winskel, eds., Lecture Notes in Comput. Sci. 1443, Springer-Verlag, Aalborg, Denmark, 1998, pp. 118–129.

[3] P. ELIAS, *Universal codeword sets and representations of the integers*, IEEE Trans. Inform. Theory, IT-21 (1975), pp. 194–203.

[4] H. GALPERIN AND A. WIGDERSON, *Succinct representations of graphs*, Inform. and Control, 56 (1983), pp. 183–198.

[5] X. He, M.-Y. Kao, and H.-I Lu, *Linear-time succinct encodings of planar graphs via canonical orderings*, SIAM J. Discrete Math., 12 (1999), pp. 317–325.

[6] A. Itai and M. Rodeh, *Representation of graphs*, Acta Inform., 17 (1982), pp. 215–219.

[7] G. Jacobson, *Space-efficient static trees and graphs*, in Proceedings of the 30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, NC, IEEE Computer Society Press, Los Alamitos, CA, 1989, pp. 549–554.

[8] S. Kannan, M. Naor, and S. Rudich, *Implicit representation of graphs*, SIAM J. Discrete Math., 5 (1992), pp. 596–603.

[9] M. Y. Kao, N. Occhiogrosso, and S. H. Teng, *Simple and efficient compression schemes for dense and complement graphs*, J. Combin. Optim., 2 (1999), pp. 351–359.

[10] K. Keeler and J. Westbrook, *Short encodings of planar graphs and maps*, Discrete Appl. Math., 58 (1995), pp. 239–252.

[11] R. J. Lipton and R. E. Tarjan, *A separator theorem for planar graphs*, SIAM J. Appl. Math., 36 (1979), pp. 177–189.

[12] G. L. Miller, *Finding small simple cycle separators for* 2*-connected planar graphs*, J. Comput. System Sci., 32 (1986), pp. 265–279.

[13] J. I. Munro and V. Raman, *Succinct representation of balanced parentheses, static trees and planar graphs*, in Proceedings of the 38th Annual Symposium on Foundations of Computer Science, Miami Beach, FL, IEEE Computer Society Press, Los Alamitos, CA, 1997, pp. 118–126.

[14] M. Naor, *Succinct representation of general unlabeled graphs*, Discrete Appl. Math., 28 (1990), pp. 303–307.

[15] C. H. Papadimitriou and M. Yannakakis, *A note on succinct representations of graphs*, Inform. and Control, 71 (1986), pp. 181–185.

[16] G. Turán, *On the succinct representation of graphs*, Discrete Appl. Math., 8 (1984), pp. 289–294.

[17] W. T. Tutte, *A census of planar triangulations*, Canad. J. Math., 14 (1962), pp. 21–38.

[18] W. T. Tutte, *A census of planar maps*, Canad. J. Math., 15 (1963), pp. 249–271.