Discrete Applied Mathematics 8 (1984) 289-294
North-Holland

ON THE SUCCINCT REPRESENTATION OF GRAPHS

György TURÁN
Automata Theory Research Group, Hungarian Academy of Sciences, Szeged, Somogyi u. 7. H-6720, Hungary

Received 17 November 1982
Revised 24 June 1983

It is shown that unlabeled planar graphs can be encoded using $12n$ bits, and an asymptotically optimal representation is given for labeled planar graphs.

Introduction

The 'good' representation of data is a general and important problem of computation. A representation (or encoding) can be good from several different points of view. An important aspect is its suitability for efficient algorithms. In the case of graphs the adjacency list is better than the adjacency matrix in several cases. Another possible aspect is the succinctness of the representation. A succinct encoding can be desirable e.g. when we have to keep several instances of data of a given type in the memory.

The typical problem considered here can be put as follows. Let $\mathcal{G}$ be a class of graphs (with labeled or unlabeled) vertices, $\mathcal{G}_n$ be the class of graphs on n vertices belonging to $\mathcal{G}$, represented originally by adjacency lists. Find a *succinct* representation of $\mathcal{G}$ *efficiently*.

An encoding can be considered succinct if the length of the encodings of graphs G in $\mathcal{G}_n$ is not too large compared to $\log_2 |\mathcal{G}_n|$ (the minimal number of bits necessary). The encoding is efficient if it is polynomial time computable.

In Section 1 we give some formal definitions. In Section 2 we give an encoding of unlabeled planar graphs optimal within a constant factor and an asymptotically optimal encoding of labeled planar graphs. This improves a recent result of Itai and Rodeh [3]. In Section 3 we mention some further results and problems.

1. Definitions

A graph is denoted by $G=(V,E)$, it is assumed to be undirected, without loops and multiple edges. The standard representation of graphs is defined to be the adjacency lists representation. Classes of graphs are denoted by $\mathcal{G}$, the class of graphs on n vertices belonging to $\mathcal{G}$ is denoted by $\mathcal{G}_n$.

Definition 1. Let $\mathscr{G}$ be a class of labeled graphs. A *representation* (or *encoding*) of $\mathscr{G}$ is a pair of mappings (CODE, DECODE) satisfying

(1) CODE : $\mathscr{G} \to \{0,1\}^*$.

(2) DECODE(CODE(G)) is the standard representation of G for every $G \in \mathscr{G}$.

(3) CODE and DECODE are polynomial time computable (in the number of vertices of G).

The *length* of an encoding is the function

$$l(n) = \max_{G \in \mathscr{G}_n} |\mathrm{CODE}(G)|.$$

For unlabeled graphs the above definition must be slightly modified.

Definition 2. Let $\mathscr{G}$ be a class of labeled graphs closed under isomorphism. A *representation* (or *encoding*) of the *unlabeled graphs* in $\mathscr{G}$ is a pair of mappings (CODE, DECODE) satisfying:

(1) CODE : $\mathscr{G} \to \{0,1\}^*$.

(2) If $G_1 \in \mathscr{G}$, then DECODE(CODE(G_1)) is the standard representation of $G_2 \in \mathscr{G}$ such that $G_1 \cong G_2$.

(3) CODE and DECODE are polynomial time computable (in the number of vertices of G).

The *length* of an encoding is defined the same way as above.

(It can be noted here that encoding as defined here is not related to the more common theme of canonical encoding of graphs in the context of the graph isomorphism problem.)

2. Planar graphs

We consider here the representation of unlabeled and labeled planar graphs. We begin with the unlabeled case as this will be used for the labeled case. The number of unlabeled planar graphs on n vertices is known to be bounded by c^n for some c, so it is desirable to have a representation the length of which is linear in n. Our encoding of unlabeled graphs is similar to the encoding of labeled planar graphs constructed by Cori [1] as a tool for enumeration results. The length of his encoding is $6n \log_2 n$.

Theorem 1. *There is a representation of unlabeled planar graphs satisfying*

$$l(n) \leq 12\, n.$$

Proof. Let G be a planar graph on n vertices. We can assume that we have a planar embedding of G, or equivalently an appropriate cyclic ordering of the edges incident to v for each vertex v. We also assume G to be connected as otherwise it is easy to see how to modify the construction.

Consider a rooted spanning tree T of G with root v_0 and a distinguished edge $(v_0, v) \in T$. Direct every edge of T away from the root. For each v we define an order on the edges incident to v and belonging to T:

(a) If v is the root, the first edge is the distinguished edge, other edges follow according to the planar embedding in counterclockwise order.

(b) If v is not the root, the first edge is the only edge entering v, other edges follow according to the planar embedding in counterclockwise order.

Take the usual preorder traversal of T. This determines a cyclic sequence of edges of length $2n-2$ (each edge traversed twice), or alternatively a cyclic sequence of vertices of length $2n-2$. Here, the number of occurences of a vertex v equals $\deg_T(v)$. Beginning with the distinguished edge in the cyclic order, for each v index the occurences of v in the cyclic sequence with indices $1, 2, \ldots, \deg_T(v)$.

For the purposes of coding this cyclic sequence can be replaced by a sequence of pluses and minuses by replacing each edge by a + (−) if it leads away from (towards) the root. (This is the usual encoding of unlabeled trees, see e.g. in [5]. The +, − serve as a pair of brackets.) This replacement is given later in Step I.

Now take an arbitrary edge $e = (v, w) \notin T$. We define an indexing of e. Let $v^{(i)}$ be the copy of v corresponding to the angle of T around v containing the edge e, and $w^{(j)}$ be the corresponding copy of w. (This is illustrated in Fig. 1. Broken lines indicate the edges of T, arrows the tree-traversal. The angle containing e is dark.) Then we index e with i and j, it becomes $e = (v^{(i)}, w^{(j)})$.

The construction described is illustrated for a small graph on Fig. 2. (The edges of T are the thick edges.)

The cyclic sequence corresponding to T is

$$1\ 2\ 3\ 2\ 4\ 2\ 1\ 5\ 6\ 5\ 7\ 8\ 7\ 5.$$

The indexing of the cyclic sequence is

$$1^{(1)}\ 2^{(1)}\ 3^{(1)}\ 2^{(2)}\ 4^{(1)}\ 2^{(3)}\ 1^{(2)}\ 5^{(1)}\ 6^{(1)}\ 5^{(2)}\ 7^{(1)}\ 8^{(1)}\ 7^{(2)}\ 5^{(3)}.$$

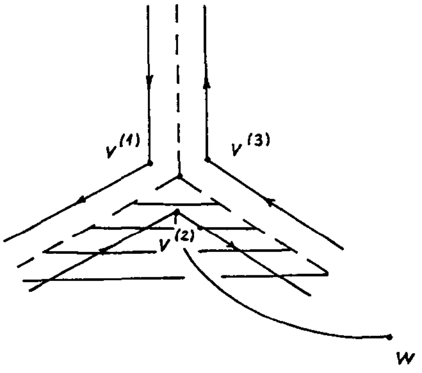

Fig. 1.

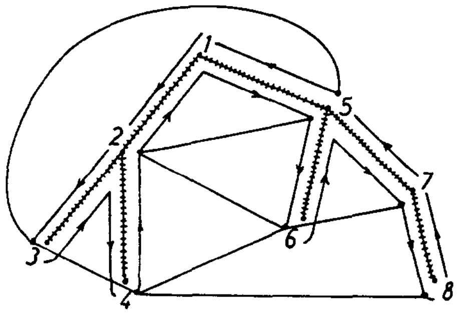

Fig. 2.

The indexing of edges not belonging to T is

$(2,5)$	becomes	$(2^{(3)},5^{(1)})$,
$(2,6)$	becomes	$(2^{(3)},6^{(1)})$,
$(3,4)$	becomes	$(3^{(1)},4^{(1)})$,
$(3,5)$	becomes	$(3^{(1)},5^{(3)})$,
$(4,6)$	becomes	$(4^{(1)},6^{(1)})$,
$(4,8)$	becomes	$(4^{(1)},8^{(1)})$,
$(6,7)$	becomes	$(6^{(1)},7^{(1)})$.

Now we need only one observation to finish the construction. Draw the vertices of the cyclic sequence corresponding to T as a regular $(2n-2)$-gon and draw each edge not belonging to T as a diagonal connecting the indexed vertices corresponding to this edge. In the case of our example this is shown by Fig. 3.

The diagonals are *non-intersecting*. This follows from the planarity of G. The graph obtained by drawing the diagonals is isomorphic to the graph G' obtained on the following way: replace each vertex v by $\deg_T(v)$ new vertices and form a new face corresponding to T. See Fig. 4.

Consider now the cyclic sequence of vertices corresponding to the traversal of T. Replacing diagonal $(v^{(i)}, w^{(j)})$ by a pair of brackets $\cdots(v^{(i)}\cdots)w^{(j)}\cdots$ we get a correctly bracketed sequence (see e.g. in Lovász [5]). Our example becomes:

$$1^{(1)}2^{(1)}((3^{(1)}2^{(2)})((4^{(1)}((2^{(3)}\ 1^{(2)})5^{(1)}))(6^{(1)}5^{(2)})7^{(1)})8^{(1)}7^{(2)})5^{(3)}.$$

We constructed a bracketed sequence of length $2n-2$. From the construction it follows that G can be reconstructed from this sequence. The last part of the construction is to represent this sequence as a sequence of 0's and 1's. This is done in two steps.

Step I. We represent the sequence as a sequence over the alphabet $+, 1, (,)$. To do this, keep the brackets, and replace each $v^{(i)}$ by a $+$ if edge $(w^{(j)}, v^{(i)})$ leads away

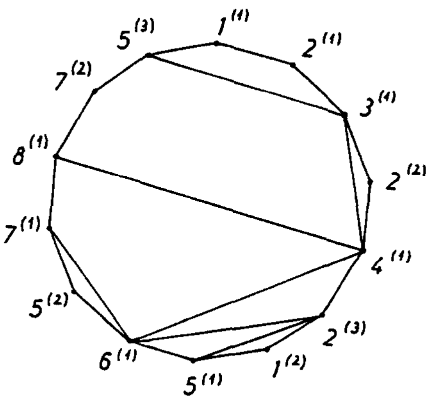

Fig. 3.

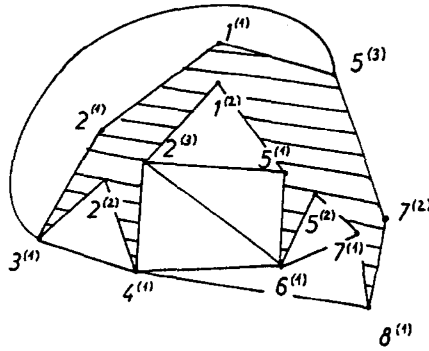

Fig. 4.

from the root ($w^{(j)}$ precedes $v^{(i)}$ in the sequence), otherwise replace $v^{(i)}$ by a $-$. Thus $1^{(1)}$ is replaced by a $-$. Thus our example becomes:

$$-+((+-)((+((--)+))(+-)+)+-)-.$$

Step II. Finally replace $+,-,(,)$ by the four binary sequences of length 2.

To summarize, the steps of the encoding algorithm are the following.

(1) Choose a spanning tree T.

(2) Construct the cyclic sequence of vertices corresponding to the traversal of T.

(3) Index edges not belonging to T.

(4) Replace the cyclic sequence by a sequence of $+$, $-$ and bracket the sequence using the indexing.

(5) Replace $+,-,(,)$ by $00, 01, 10, 11$.

The length of the final sequence can be estimated as follows:

(a) the cyclic sequence of length $2n-2$ needs $4n-4$ bits,

(b) the number of edges not belonging to T is at most $(3n-6)-(n-1)=2n-5$, each edge needs 1 pair of brackets, thus these edges need at most $8n-20$ bits.

Hence the length of the representation is at most $12n-24$.

If G is not connected, adding a final $-$ to the code of every connected component, the component will be identifiable and the length bound remains valid.

It is easy to see how to decode the final sequence; first we reconstruct T using the $+,-$ signs, and then reconstuct each edge using the bracketing. Both coding and decoding are obviously polynomial.

If we allow loops and multiple edges, the natural modification of the representation will still be linear in the number of edges. □

Now we turn to labeled planar graphs. As it is remarked in [3] as well, from the enumeration of labelled trees it follows that $(n-2)\log_2 n$ bits are necessary for their representation. This lower bound can be attained asymptotically.

Theorem 2. *There is a representation of labeled planar graphs satisfying*

$$l(n) = n\lceil \log_2 n\rceil + 12\, n.$$

Proof. This follows directly from Theorem 1. Represent G as an unlabeled planar graph using $\leq 12\,n$ bits. Padding with 0's obtain a sequence of length $12\,n$ (as we had a correct bracketing it can be detected where the padding begins). In this representation there is specified an ordered spanning forest. The planar embedding determines an ordering of the vertices, i.e. a permutation of the original vertex labels (in our example the permutation is the identity). To get a representation of G it suffices to write down the permutation (taking $n\lceil \log_2 \mathrm{n}\rceil$ bits) and concatenate it to the initial $12\,n$ bits. We do not need commas because the length depends only on n (and n can be calculated easily given the code). □

3. Remarks

A problem that is perhaps of some theoretical interest is the encoding of general unlabeled graphs. Here the adjacency matrix gives an $\binom{n}{2}$ representation, while the enumeration of unlabeled graphs [2] gives a lower bound of $\binom{n}{2} - n \log_2 n + O(n)$. It can be shown that there is an encoding into $\binom{n}{2} - \frac{1}{8} n \log_2 n + O(n)$ bits using the idea of the proof of Ramsey's theorem. As to 'save bits' means here to find efficiently regularity in the graph to be encoded it may be possible that some other kind of Ramsey argument could improve this result.

Another related problem is the *encoding of subgraphs*. Suppose we want to encode the *spanning trees* of a graph so that the length of the codes is not too large compared to the logarithm of the number of spanning trees. (We do not give a formal definition here, it is similar to Definition 1.) Using the following two results:

(a) spanning trees of a multigraph can be *enumerated* in polynomial time (see e.g. in Lovász [5]),

(b) spanning trees can be *listed* without repetition spending only a polynomial amount of time between the listing of two trees (see Read–Tarjan [6]),

it can be shown that in fact an *optimal* encoding can be given in this case. The optimal encoding of the tree is its *rank* in the listing that can be computed efficiently using the listing algorithm and cutting of irrelevant branches of the search tree calling the enumerating algorithm each time a subtree is cut off.

The existence of results analogous to (a) and (b) above for *matchings in planar graphs* gives a similar result for the corresponding encoding problem.

The case of encoding matchings in general (or in bipartite) graphs is different. Results analogous to (b) exist [4], but probably there is no analogue of (a) as indicated by the results of Valiant [8]. A good encoding would be given if we had an approximate counting algorithm for matching, but even this is doubtful (see Stockmeyer [7]).

Acknowledgement

I am grateful to Professor L. Lovász for suggesting the problem of encoding planar graphs.

References

[1] R. Cori, Sur la rationalite de certaines series generatrices, Discrete Math. 3 (1972) 315–332.
[2] F. Harary and E. Palmer, Graphical Enumeration (Academic Press, New York, 1973).
[3] A. Itai and M. Rodeh, Representation of graphs, Acta Inform. 17 (1982) 215–219.
[4] A. Itai and M. Rodeh, Some matching problems, in: S. Salomaa and M. Steinby, eds., Automata, Languages and Programming, Lecture Notes in Computer Science 52 (Springer, Berlin, 1975) 258–268.
[5] L. Lovász, Combinatorial Problems and Exercises (Akadémiai Kiadó, Budapest, 1979).
[6] R.C. Read and R.E. Tarjan, Bounds on backtrack algorithms for listing cycles, paths, and spanning trees, Networks 5 (1975) 237–252.
[7] L. Stockmeyer, The complexity of approximate counting, Proc. 15th ACM STOC (1983) 118–126.
[8] L.G. Valiant, The complexity of computing the permanent, Theoret. Comp. Sci. 8 (1979) 189–202.