Acta Informatica 17, 215–219 (1982)

Representation of Graphs

Alon Itai[1] and Michael Rodeh[2]

[1] Computer Science Department, Technion – Israel Institute of Technology, Haifa, Israel
[2] IBM Israel Scientific Center, Technion City, Haifa, Israel

Summary. Given a formulation of a problem, a compact representation is required both for theoretical purposes – measuring the complexity of algorithms, and for practical purposes – data compression.

The adjacency lists method for representing graphs is compared to the information theoretic lower bounds, and it is shown to be optimal in many instances. For n-vertex labeled planar graphs the adjacency lists method requires $3n\log n+O(n)$ bits, a linear algorithm is presented to obtain a $3/2n\log n+O(n)$ representation while $n\log n+O(n)$ is shown to be the minimum.

1. Introduction

Algorithm design is concerned with finding efficient algorithms to solve problems. Each instance of a problem has some representation.

One measure of efficiency is the worst case time complexity; for each algorithm Al a function wt_{Al} is defined as follows: for each problem P let $t_{Al}(P)$ be the number of steps required by the algorithm Al, then $wt_{Al}(n)$ is the maximum of $t_{Al}(P)$ over all problems whose representation requires n bits. The representation method plays a crucial role in defining the complexity. In order to obtain meaningful complexity measures, it is required that the problems be represented compactly.

The knapsack problem may be solved in time linear in the length of a representation, whose length is an exponential function of the length of the usual more compact representation.

In Sect. 2 the space required to store adjacency lists is computed and then compared to the information theoretic lower bound. Section 3 is devoted to labeled planar graphs as a case study. It is shown that when using adjacency lists to represent labeled graphs with n vertices, $3n\log n+O(n)$ bits[1] may be required while the minimum is $n\log n+O(n)$ bits. An efficient algorithm is given for constructing a representation which requires at most $3/2\,n\log n+O(n)$ bits.

[1] All logarithms are to the base 2

0001-5903/82/0017/0215/$01.00

2. Representations of Labeled Graphs

Let G be a labeled undirected graph with no self loops and no parallel edges. A fairly compact representation is by adjacency lists: for each vertex i a list L_i is constructed. L_i consists of all the vertices adjacent to i with a larger label. Hence, each edge is represented once in the adjacency lists. To count the number of bits required we must be more specific. Each entry in the list L_i is an integer between 2 and n and can be represented by a block of $\lceil \log(n-1) \rceil$ bits. In each list the blocks are separated from one another by a "comma" and each list ends with a "period". Since a comma or a period must appear after every block, their locations are known in advance and they can be represented by one bit. To handle empty lists an additional Boolean vector of length $n-1$ is used: it contains zeros in those locations for which the corresponding L_i's are empty. The total length of the representation is $\lceil \log(n-1) \rceil + m + n - 1$.

Observe that the length of the representation depends only on m and n. Therefore, assume that all graphs having n vertices and m edges are to be represented. The number of such graphs is $\binom{\binom{n}{2}}{m}$. To represent all graphs $\log \binom{\binom{n}{2}}{m}$ bits are enough. Let $k = \binom{n}{2}$; $p = m/k$ and $q = 1-p$.

$$
\begin{aligned}
\log \binom{\binom{n}{2}}{m} &= \log \binom{p}{pk} \\
&= \log \frac{k!}{(pk)!\,(qk)!} \\
&\cong \log \frac{\sqrt{2\pi k}\cdot (k/e)^k}{\sqrt{2\pi pk}\,(pk/e)^{pk}\cdot \sqrt{2\pi qk}\,(qk/e)^{qk}} \\
&= -\log(\sqrt{2\pi pqk}\cdot (p^p q^q)^k) \\
&= -\tfrac{1}{2}\log(2\pi pqk) - k(p\log p + q\log q)
\end{aligned}
$$

Consider a graph G with n vertices and m edges as an object drawn at random out of the class $G_{n,m}$ of all graphs with n vertices and m edges [1]. All edges (i,j) have the same probability p to appear in G. G may be represented as a sequence of k bits m of which are 1. The probability of the i-th entry to be 1 is p. The entropy of every entry is $-p\log p - q\log q$ [2]. Consequently, the entropy of the space of k such random variables is $-k(p\log p + q\log q)$. The other expression, $-\frac{1}{2}\log(2\pi pqk)$ may be interpreted as the influence of the dependency between the edges (whose number is forced to be m).

Let us consider four special cases:

Case 1. $m = n^{1+\varepsilon}$ where $0 < \varepsilon < 1$. Therefore $p = 2\cdot n^{\varepsilon}/(n-1)$ and the entropy E tends to $1-\varepsilon$ as n approaches infinity. Thus, using adjacency lists causes a waste by a factor of $1/(1-\varepsilon)$.

Case 2. $m = p \cdot k$ where p is a constant $0 < p < 1$. In this case, using adjacency lists causes a waste of space by a factor of $c \cdot \log n$ for some constant $c > 0$.

Case 3. $m = cn \log n$ for some constant $c > 0$. Here $p = (2 \log n)/(n-1)$. In this family the adjacency lists representation tends to be optimal as n increases.

Case 4. Labeled trees. Using adjacency lists $n \log n + O(n)$ space suffices. On the other hand, the number of labeled trees is n^{n-2}. Therefore, $(n-2) \log n$ bits are required; thus, the adjacency lists representation is optimal in the limit.

3. Representation of Labeled Planar Graphs

Since trees are labeled planar graphs, labeled planar graphs require at least $(n-1) \log n$ space. First Tutte's result [5] on the enumeration of rooted non-separable inner-triangular graphs is used to prove that every labeled planar graph may be represented in $n \log n + O(n)$ bits.

However, having an existence proof on representations of graphs does not supply us with an effective procedure to construct such a representation. Therefore, a procedure is described for constructing a good representation. This procedure works in linear time and produces a representation which requires at most $3/2 n \log n + O(n)$ bits. This representation is more compact than adjacency lists which require $3n \log n + O(n)$ bits.

A planar graph is *triangular* if all its faces contain exactly 3 edges. Every planar graph is a subgraph of a triangular graph with the same number of vertices. Let GT be a triangular planar graph containing G (GT can be constructed in linear time). Let us sort the edges of GT in lexicographical order and construct a Boolean vector D of length $3n-6$ (the number of edges in GT) such that

$$D(i) = \begin{cases} 1 & \text{if the } i\text{-th edge of } GT \text{ is an edge of } G \\ 0 & \text{otherwise.} \end{cases}$$

A representation of G results from a representation of GT together with the vector D. We conclude that for a cost of $3n-6$ bits the problem of representing an arbitrary labeled planar graph is reduced to the problem of representing a labeled triangular graph.

Given a lebeled triangular graph GT, a choice of a face and an orientation for it induces a unique representation of the graph in the plane (for a more precise treatment, see [4]). Let us choose the face whose vertices $\{1, W_1, W_2\}$ form a minimum lexicographical triple. Draw the graph in the plane such that the face $(1, W_1, W_2)$ is the outermost face, where $(1, W_1, W_2)$ is arranged in clockwise order, $W_1 < W_2$. Let GT_1 be this planar representation.

Attaching direction to the edge $(1, W_1)$ (from 1 to W_1) and ignoring the labels of the vertices yields a non-separable inner triangular graph (that is, all inner faces must consist of three edges while the outermost face may contain an arbitrary number) with a designated edge in the outermost face (called a *root* by Tutte). In our case the outermost face consists of three edges, the number of inner faces is $2n-5$ and the root is the original edge $(1, W_1)$. Tutte found the generating function $q(x,z)$ for enumerating rooted non-separable inner-triangular graphs:

$$q(x,z)=\sum_{k=0}^{\infty}\sum_{j=0}^{\infty}\frac{2^{j+1}(2k+1)!\,(3j+2k)!\,x^{k+2}z^{k+2j}}{(k!)^2 j!\,(2j+2k+2)!} \tag{1}$$

where the exponent of x expresses the number of edges in the outermost face and the exponent of z corresponds to the number of inner faces. To find the number of triangular graphs, set $k+2=3$ and $k+2j=2n-5$. Therefore $k=1$ and $j=n-3$. Substituting in (1) yields

$$\frac{2^{n-2}\cdot 3!\,(3n-7)!}{(n-3)!\,(2n-2)!}=\frac{2^{n-2}\cdot 6\cdot\binom{3n-5}{n-3}}{(3n-5)\cdot(3n-6)}<2^{4n}$$

Therefore, every rooted non-separable triangular graph with n vertices may be represented by $4n$ bits.

Adding labels may produce up to $n!$ labeled triangular graphs. Therefore, $\log(n!\,4^n)+O(n)=n\log n+O(n)$ bits suffice to represent all labeled planar graphs. The note at the beginning of this section implies that this is also a lower bound.

The proof given above is not constructive in the sense that it is based on an enumeration theorem and no easily computable function to find the compact representation is provided. Since the compact representation is required to measure the complexity of problems and algorithms, an efficient function is imminent. Below is described a linear time algorithm to find a representation which is worse than the optimal by a factor of 3/2.

Recall the properties of the graph GT_1: it contains G as a subgraph, its outermost face is $(1, W_1, W_2)$ and it is triangular. The graph GT_1 will be represented using a vector A and a list L. Initially, $L=W_1, W_2$. We have thus wasted $2\log n$ bits to represent the outermost face. In the i-th step the graph GT_i is inner triangular and the outermost face contains the vertices $V_1, V_2, \ldots, V_k$ in clockwise order, and all the edges of this face have already been represented. Let V_1 be the smallest vertex in the outermost face and let $e_0, e_1, \ldots, e_m$ be the edges incident with V_1, ordered clockwise, such that $(V_2, V_1)=e_0$, $(V_1, V_k)=e_m$ and the other edges are not incident with the outermost face. Four exclusive cases may arise:

Case 1. V_1 is incident only with two edges of the outermost face, namely, (V_2, V_1) and (V_1, V_k). Since the graph is inner triangular the edge (V_2, V_k) necessarily exists. The two edges are deleted and (V_2, V_k) is added to form a new outermost face. Consequently, GT_{i+1} is obtained from GT_i by deleting V_1.

Case 2. The edge e_1 leads to V_3 (which is two edges away from V_1 on the outermost face). Replace the edge (V_1, V_2) and (V_2, V_3) by the edges (V_1, V_3), consequently V_2 is deleted to obtain GT_{i+1}.

Case 3. The edge e_1 leads to a vertex on the outermost face, V_j, $V_j \neq V_3$, V_k. Append V_j to L. The edge (V_j, V_2) necessarily exists. Delete the edge (V_1, V_2) to obtain GT_{i+1}. The vertex V_j becomes an articulation point and it separates the planar graph GT_{i+1} into two parts. First represent the part whose outermost face is $(V_1, V_j, V_{j+1}, \ldots, V_k)$. Then represent the part whose outermost face is $(V_j, V_2, V_3, \ldots, V_{j-1})$. Two additional edges have been accounted for at the cost of $\log n$ bits, hence $\frac{1}{2}\log n$ bits per edge.

Case 4. The edge e_1 leads from V_1 to some vertex V not on the outermost face. Append V to L. The edge (V_2, V) necessarily exists and at the cost of $\log n$ bits two edges have been accounted for.

Consequently L contains no more than $2\log n + 1/2 \cdot (3n-6)\log n < 3/2\, n\log n$ bits. To be able to decode L an additional vector A must be kept to designate which of the four cases occurred at each stage. The length of this vector is at most $3n-6$ and each entry requires 2 bits, hence $6n-12$ bits.

Given the adjacency lists of a graph, a planar representation can be found in linear time using Hopcroft and Tarjan's algorithm [3]. From a planar representation the vectors D and A and the list L can also be found in linear time. Conversely, a planar representation and the adjacency lists representation can be found from D, A and L in linear time.

References

1. Erdös, P., Spencer, J.: Probabilistic methods in combinatorics. Academic Press 1974
2. Gallager, R.G.: Information theory and reliable communication. John Wiley and Sons 1968
3. Hopcroft, J., Tarjan, R.: Efficient planarity testing. JACM **21**, 549–568 (1974)
4. Liu, C.L.: Introduction to combinatorial mathematics. McGraw-Hill 1968
5. Tutte, W.T.: The enumerative theory of planar maps. In: A survey of combinatorial theory. J.N. Srivastava, Havary, F., Rao, C.R., Rota, G.-C., Shrikhande, S.S. (eds.). North-Holland Publishing Company 1973

Received October 18, 1979 / December 29, 1981