A Method for Reducing Space Complexity of Incidence Matrices of Traceable Graphs and its Application in Parallel Computing

Kenan Kalajdzic <kenan@unix.ba>

Written in March 2012 (original idea 1998)

Abstract

We present a simple method to reduce the space complexity of an incidence matrix of a traceable undirected graph by $O(n^2)$, where n is the number of vertices of the graph. Large-scale parallel applications, which make use of the incidence matrix representation, could benefit from the reduced memory, storage and bandwidth requirements achieved by utilizing the presented method.

1 Introduction and basic definitions

Let $G = (V, E)$ be an undirected graph with vertices $V = \{v_1, v_2, \ldots, v_n\}$ and edges $E = \{e_1, e_2, \ldots, e_m\}$. The relationship between the vertices and edges of G can be conveniently expressed through an incidence matrix.

The incidence matrix M of the graph G is an $n \times m$ matrix, whose elements m_{ij} are defined as follows:

$$m_{ij} = \begin{cases} 1 & \text{if edge } e_j \text{ is incident on vertex } v_i \\ 0 & \text{if edge } e_j \text{ is not incident on vertex } v_i \end{cases}$$

Consider the example graph G_1 shown in Figure 1. The incidence matrix M_1 of this graph has the size of 8×13 and is defined as follows:

$$M_1 = \begin{bmatrix} 0&0&0&1&0&1&1&0&1&0&0&1&0 \\ 0&1&0&0&1&0&0&1&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&1&0 \\ 1&0&0&0&0&0&0&0&1&0&1&0&0 \\ 0&0&0&0&0&1&0&1&0&1&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&1 \\ 1&0&1&1&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&1&0&0&1&1&0&1 \end{bmatrix}$$

Looking at the matrix M_1, it is hard to observe any regular structure. In the general case, the vertices and edges of G_1 are named arbitrarily, which makes the distribution of 0s and 1s in the matrix M_1 irregular.

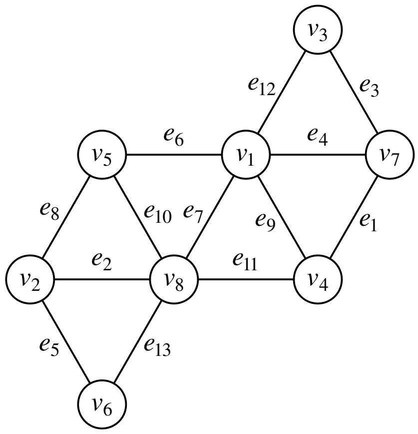
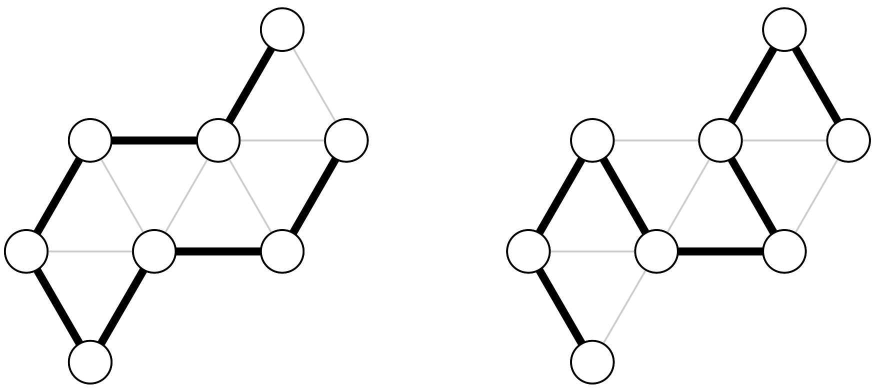


Figure 1: Graph G_1 with 8 vertices and 13 edges

Let us assume that G is a traceable graph. This means that there exists at least one simple path P which connects all the vertices of G. Such a path is called *Hamiltonian* or *traceable*. In the forthcoming discussion we demonstrate how it is possible to reorder the vertices and edges of any traceable graph to produce a corresponding incidence matrix with a partly regular structure.



a. b.

Figure 2: Two different Hamiltonian paths of the graph G_1

For the purpose of illustration, let us consider the example graph G_1. It is easy to see that G_1 is traceable and contains multiple Hamiltonian paths (Figure 2).

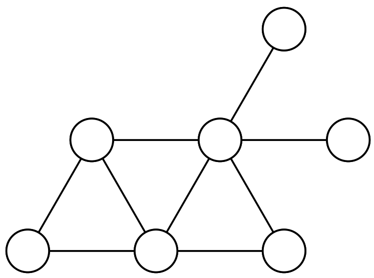


Figure 3: A connected graph without a Hamiltonian path

A slightly different graph is shown in Figure 3. Despite being connected, this graph is not traceable, meaning it is not possible to find a simple path which connects all its vertices. The method, which we describe in the following section, is therefore not applicable to this graph.

2 Description of the method

Let P_1 be a Hamiltonian path of G_1. For the purpose of discussion, we assume P_1 is the path shown in Figure 2b. Once we have chosen the path P_1, we continue with the following procedure:

1. Beginning with the start vertex, which we name v'_1, we continue naming all the vertices along the path P_1 sequentially, so that their indices are increasing as we traverse P_1 from the start to the end.

2. Subsequently, we name all the edges of P_1, so that the edge connecting vertices v'_i and v'_{i+1} is named e'_i, $i = 1, 2, \ldots, n-1$.

3. Finally, we name the remaining edges $e'_n, e'_{n+1}, \ldots, e'_m$ in an arbitrary fashion.

The result is shown in Figure 4. The corresponding incidence matrix M'_1 defines the relationships between the renamed vertices and edges of G_1:

$$M'_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

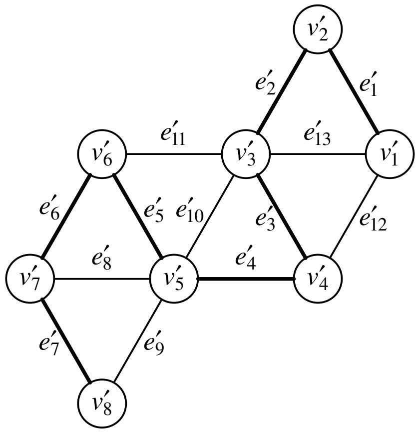


Figure 4: Graph G_1 with renamed vertices and edges

As a result of renaming the vertices and edges along the path P_1, the seven leftmost columns of M_1' have a regular form, in which the 1s are distributed diagonally. The rightmost six columns of M_1' have a rather arbitrary distribution of 0s and 1s due to the way in which we named the remaining edges of G_1. We can now conveniently define the matrix M_1' as:

$$M_1' = \begin{bmatrix} L_1' & R_1' \end{bmatrix} \ , \ L_1' = \begin{bmatrix} 1&0&0&0&0&0&0 \\ 1&1&0&0&0&0&0 \\ 0&1&1&0&0&0&0 \\ 0&0&1&1&0&0&0 \\ 0&0&0&1&1&0&0 \\ 0&0&0&0&1&1&0 \\ 0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&1 \end{bmatrix} \ , \ R_1' = \begin{bmatrix} 0&0&0&0&1&1 \\ 0&0&0&0&0&0 \\ 0&0&1&1&0&1 \\ 0&0&0&0&1&0 \\ 1&1&1&0&0&0 \\ 0&0&0&1&0&0 \\ 1&0&0&0&0&0 \\ 0&1&0&0&0&0 \end{bmatrix}$$

2.1 Generalization of the presented method

The method described in the previous section can be used to transform an $n \times m$ incidence matrix M of a traceable graph G into an equivalent incidence matrix M', with two submatrices L' and R', such that:

$$M' = \begin{bmatrix} L' & R' \end{bmatrix}$$

The left submatrix L' is an $n \times (n-1)$ matrix, whose elements l'_{ij} are defined as follows:

$$l'_{ij} = \begin{cases} 1 & i = j \text{ or } i = j+1 \qquad j \in \{1, 2, \ldots, n-1\} \\ 0 & \text{otherwise} \end{cases}$$

The right submatrix R' contains the rightmost $m - n + 1$ columns of M'. Since, however, the locations of the 1s in the matrix R' are generally not predictable, a further analysis of R' in the context of reducing space complexity of M is not of an interest to us.

2.2 An alternative interpretation

The process of converting M into M' can be interpreted in a different way. Renaming the vertices and edges of the graph G corresponds to reordering of the rows and columns of its incidence matrix M. Matrix M' is derived from the matrix M through a specific permutation of the rows and columns of M. In an $n \times m$ incidence matrix M there is a total of $n!m!$ such permutations. If the graph G is traceable, then we can always find at least $(m - n + 1)!$ permutations which produce M' with the left submatrix L'.

3 Application in parallel computing

Even though incidence matrices are fairly inefficient in terms of space complexity, some applications may still use them as a convenient representation in solving problems which are modeled using graphs.

Consider a parallel MPI-based application consisting of p processes which are executing on a cluster, and assume the root process calls `MPI_BCAST()` to distribute a huge $n \times m$ incidence matrix M to all non-root processes. In this situation, the root process may first perform conversion from M to M' as described in Section 2 and inform all the non-root processes about it. Each process could then easily derive the left submatrix L', so that root would only have to broadcast the right submatrix R'. The total savings while transferring the matrix M would be $O(pn^2)$.

Since, however, M is sparse, it is likely that a carefully designed parallel application would use a more space-efficient data structure for storing and distributing the contents of M. In this case, the conversion from M to M' would yield savings of $O(2pn)$.