Iranian Journal of Science & Technology, Transaction B, Vol. 28, No. B5
Printed in The Islamic Republic of Iran, 2004


# ALGEBRAIC GRAPH THEORY FOR THE FORMATION OF SUBOPTIMAL CYCLE BASES; AN EFFICIENT FORCE METHOD*

A. KAVEH** AND H. RAHAMI

Department of Civil Engineering, Iran University of Science and Technology, Tehran, I. R. of Iran
Email: alikaveh@iust.ac.ir

**Abstract–** An efficient algorithm is presented for the formation of suboptimal cycle bases of graphs corresponding to sparse cycle adjacency matrices, leading to the formation of highly sparse flexibility matrices. The algorithm presented employs concepts from the algebraic graph theory, together with a Greedy type algorithm to select cycles with small overlaps and uses a simple graph-theoretical method for controlling the independence of the selected cycles. Application of the present algorithm is extended to the formation of cycle bases corresponding to well conditioned flexibility matrices.

**Keywords–** Force method, flexibility matrices, statical bases, sparsity, cycle bases, graph theory, suboptimal

## 1. INTRODUCTION

Consider a frame structure S with M(S) members and N(S) nodes, which is $\gamma(S)$ times statically indeterminate. Select $\gamma(S)$ independent unknown forces as redundant forces. These unknown forces can be chosen from external reactions and/or internal forces of the structure. Denote these redundant forces by $\mathbf{q} = \{q_1, q_2,..., q_{\gamma(S)}\}$.

In order to obtain a statically determinate structure, the constraints corresponding to redundant forces should be removed. Such a structure is known as the *basic* (*primary or released*) *structure* of S. The rigidity of this basic structure is assumed to hold. Consider the external joint loads as $\mathbf{p} = \{p_1, p_2,..., p_n\}$, where n is the number of components for the applied nodal forces. The stress resultant distribution due to the given load **p** for a general linear analysis by the force method can be written as

$$\mathbf{r} = \mathbf{B}_0\,\mathbf{p} + \mathbf{B}_1\mathbf{q} \tag{1}$$

where $\mathbf{B}_0$ and $\mathbf{B}_1$ are rectangular matrices each having m rows, and n and $\gamma(S)$ columns, respectively. m being the number of independent components for member forces. $\mathbf{B}_0\,\mathbf{p}$ is known as a *particular solution* which satisfies equilibrium with the applied loads, and $\mathbf{B}_1\mathbf{q}$ are *complementary solutions* formed from a maximal set of independent self-equilibrating stress systems known as a *statical basis*.

Particular and complementary solutions are usually obtained from the same basic structure; however this is not a necessary requirement. A basic structure need not be selected as a determinate one. For redundant basic structures, one may obtain the necessary data either by analyzing it first for the loads **p** and bi-actions $q_i = 1$ ($i = 1,2, ... , \gamma(S)$), or by using the information available prior to the analysis [1].

Using the load-displacement relationship for each member, and collecting them in the diagonal of the unassembled flexibility matrix $\mathbf{F}_m$, one can write

$$\mathbf{u} = \mathbf{F}_m\,\mathbf{r} = \mathbf{F}_m\,\mathbf{B}_0\,\mathbf{p} + \mathbf{F}_m\,\mathbf{B}_1\mathbf{q} \tag{2}$$

in which **u** is the member distortions due to the internal forces **r**. Employing the contragradient principle, the displacements corresponding to **p** and **q** vectors are obtained as

*Received by the editors August 14, 2002 and in final revised form November 28, 2003
**Corresponding author

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{B}_0^t \\ \mathbf{B}_1^t \end{bmatrix} \mathbf{U} = \begin{bmatrix} \mathbf{B}_0^t \\ \mathbf{B}_1^t \end{bmatrix} [\mathbf{F}_m][\mathbf{B}_0 \;\; \mathbf{B}_1] \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix} \tag{3}$$

where $\mathbf{v}_0$ contains the displacements corresponding to the force components of $\mathbf{p}$, and $\mathbf{v}_1$ reflects the relative displacement of the released position (cuts) for the basic structure. By performing the multiplication, Eq. (3) results in

$$\begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{B}_0^t \mathbf{F}_m \mathbf{B}_0 & \mathbf{B}_0^t \mathbf{F}_m \mathbf{B}_1 \\ \mathbf{B}_1^t \mathbf{F}_m \mathbf{B}_0 & \mathbf{B}_1^t \mathbf{F}_m \mathbf{B}_1 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix} \tag{4}$$

Imposing the compatibility conditions as $\mathbf{v}_1 = \mathbf{0}$, the redundant forces are obtained as

$$\mathbf{q} = -(\mathbf{B}_1^t \mathbf{F}_m \mathbf{B}_1)^{-1} (\mathbf{B}_1^t \mathbf{F}_m \mathbf{B}_0) \mathbf{p} \tag{5}$$

Substituting in Eq. (1) yields

$$\mathbf{r} = \left[\mathbf{B}_0 - \mathbf{B}_1 (\mathbf{B}_1^t \mathbf{F}_m \mathbf{B}_1)^{-1} (\mathbf{B}_1^t \mathbf{F}_m \mathbf{B}_0)\right] \mathbf{p} \tag{6}$$

in which $\mathbf{G} = \mathbf{B}_1^t \mathbf{F}_m \mathbf{B}_1$ is known as the *flexibility matrix* of a structure.

Allied to a statical basis there is another set associated with the graph model S of a structure. This set consists of a maximal number of independent cycles of S known as its *cycle basis*. On each cycle, $\alpha = 3$ or 6 self-equilibrating stress systems can be formed depending on S being the model of a planar or a space structure, respectively. The cardinality of a cycle basis is equal to the first Betti number $b_1(S) = M(S) - N(S) + b_0(S)$, where M(S), N(S) and $b_0(S)$ are the numbers of members, nodes, and components of S, respectively.

The *sparsity coefficient* $\chi$ of a matrix is defined as its number of nonzero entries. A cycle basis $C = \{C_1, C_2, \ldots, C_{b_1(S)}\}$ is called *minimal* if it corresponds to a minimum value of

$$L(C) = \sum_{i=1}^{b_1(S)} L(C_i) \tag{7}$$

where $L(C_i)$ is the number of members of cycle $C_i$, known as its *length*, and $\mathbf{C}$ is the cycle-member incidence matrix of S. Obviously $\chi(\mathbf{C}) = L(C)$ and a minimal cycle basis corresponds to minimum $\chi(\mathbf{C})$. A cycle basis for which L(C) is near minimum is called a *subminimal* cycle basis of S.

A cycle basis corresponding to maximal sparsity of the $\mathbf{D}=\mathbf{C}\mathbf{C}^t$ is called an *optimal* cycle basis of S. If $\chi(\mathbf{D})$ does not differ considerably from its minimum value, then the corresponding basis is termed *suboptimal*.

$$\chi(\mathbf{D}) = \eta(S) + 2 \sum_{i=1}^{b_1(S)-1} \sigma_i(\mathbf{C}) \tag{8}$$

in which $\sigma_i(\mathbf{C})$ of row i of cycle basis incidence matrix $\mathbf{C}$ is the number of j such that

$$\text{a) } j \in \{i+1, i+2, \ldots, b_1(S)\}, \text{ b) } C_i \cap C_j \neq \phi \tag{9}$$

The cycle adjacency matrix $\mathbf{D} = \mathbf{C}\mathbf{C}^t$ of a graph is pattern equivalent to the flexibility matrix $\mathbf{G}$ of the corresponding structure constructed on the selected basis. In the force method, an optimal cycle basis is needed corresponding to the maximum sparsity of the $\mathbf{D}$ matrix.

Theoretically the application of the Greedy algorithm leads to the formation of the minimal cycle basis of a graph [1-3]. For generating a subminimal cycle basis of a graph, Kaveh's algorithm is the fastest known approach [4-5]. Graph theoretical algorithms for the formation of suboptimal cycle bases are due to Kaveh [6-7].

In this paper an algorithm is developed which uses some of the concepts from the algebraic graph theory, together with an expansion process for the formation of suboptimal cycle bases leading to highly sparse flexibility matrices. The admissibility condition of Kaveh [4] is used to control the independence of the cycles. Application of the present algorithm is extended to the formation of cycle bases corresponding to well conditioned flexibility matrices.

## 2. BASIC DEFINITIONS AND CONCEPTS

Some graph theoretical definitions are provided in this section. For further concepts and definitions the reader may refer to Kaveh [1, 5].

### *a) Basic definitions from graph theory*

A *graph* S consists of a set of N(S) elements called *nodes* (vertices or points) and a set of M(S) elements called *members* (edges or arcs) together with a relation of incidence which associates each member with a pair of nodes called *ends*. The connectivity properties of a skeletal structure can simply be transformed into that of a graph S; the joints and the members of the structure correspond to the nodes and the edges of S, respectively. Such a transformation is applied in the algorithm of this paper and the other existing algorithms of references [2-4]. Two or more members joining the same pair of nodes are known as *multiple members*, and a member joining a node to itself is called a *loop*. A graph with no loops but having multiple members is called a *simple graph*. If N(S) and M(S) are countable sets, then the corresponding graph S is finite. A graph is called *planar* if it can be drawn on a plane and its members intersect with each other only at the nodes. A graph $S_i$ is a *subgraph* of S if $N(S_i) \subseteq N(S)$, $M(S_i) \subseteq M(S)$, and each member of $S_i$ has the same end nodes as in S.

A *path* of S is a finite sequence $P_i = \{n_0, m_1, n_1, ..., m_p, n_p\}$ whose terms are alternately distinct nodes $n_i$ and distinct members $m_i$ of S for $1 \le i \le p$, and $n_{i-1}$ and $n_i$ are the two ends of $m_i$. The *length* of a path $P_i$ denoted by $L(P_i)$ is taken as the number of its members. $P_i$ is called the *shortest path* between the two end nodes $n_0$ and $n_p$, if for any other path is $P_j$ between these nodes $L(P_j) \le L(P_j)$. The *distance* between two nodes $n_1$ and $n_2$ in a graph S is taken as the length of the shortest path between them provided that the members of the path are contained in S and is denoted by $d_S(n_1,n_2)$ or $d_S(n_2,n_1)$.

Two nodes $n_i$ and $n_i$ are said to be *connected* in S if a path exists between these nodes. A graph S is called *connected* if all pairs of its nodes are connected. A *component* of S is a maximal connected subgraph, i.e. it is not a subgraph of any other connected subgraph of S. A graph is *2-connected* if it remains connected when one of its members is removed.

A *cycle* is a path $(n_0, m_1, n_1,..., m_p, n_p)$ for which $n_0 = n_p$ and $p \ge 3$. A *tree* T of S is a connected subgraph which contains no cycle. If a tree contains all the nodes of S it is called a *spanning tree* of S. A *shortest route tree* (SRT) rooted at a specified node $n_0$ of S is a spanning tree in which the distance between every node $n_i$ of T and $n_0$ is minimum.

Let a cycle set of members of a graph be defined as a set of members which form a cycle or form several cycles having no common member, but perhaps common nodes. The null set is also defined as a cycle set. A vector representing a cycle set is called a *cycle set vector*. It can be shown that the sum of two cycle set vectors of a graph is also a cycle set vector. Thus the cycle set vectors of a graph form a vector space over the field of integer modulo 2, [8]. The dimension of a *cycle space* is equal to the first Betti number of the graph $b_1(S)$.

### *b) Independence control*

Consider the following expansion process where in each step one cycle is selected

$$C_1 = C^1 \rightarrow C_1 \cup C_2 = C^2 \rightarrow C^2 \cup C_3 = C^3 \rightarrow \ldots \rightarrow C^{b_1(S)} = S \qquad (10)$$

A cycle $C_{k+1}$ is *admissible* if $b_1(C^{k+1}) = b_1(C^k) + b_1(C_{k+1}) = b_1(C^k) + 1$.

### *c) Formation of minimal cycles*

In order to form the shortest cycle on a trapper, form an SRT from one end of the trapper using the subgraph containing the previously selected cycles, until the other end of the trapper is reached. For the

formation of the shortest cycle on an unused member, discard the member and form an SRT from one end until the other end is reached. An SRT rooted from a specified node $n_0$ can be formed by the following simple algorithm: Label the selected root $n_0$ as "0" and the adjacent nodes as "1". Record the members incident to "0" as tree members. Repeat the process of labelling with "2" the unnumbered ends of all the members incident with nodes labelled as "1", again recording the tree members. This process terminates when each node of S is labelled and all the tree members are recorded.

The shortest cycle on an unused member containing the members of the previously selected cycles is formed by the same process as the previous formation with the difference being the use of the members of the previous cycles only.

***d) Definitions from algebraic graph theory***

Let S(N,M) be a graph with node set N, containing n nodes and the member set M. The adjacency matrix $\mathbf{A} = [a_{ij}]_{n\times n}$ of the labelled graph S is defined as

$$a_{ij} = \begin{cases} 1 & \text{if node } n_i \text{ is adjacent to } n_j \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

The degree matrix $\mathbf{D} = [d_{ij}]_{n\times n}$ is a diagonal matrix of the node degrees. $d_{ii}$ is equal to the degree of the ith node. Consider the following eigenproblem:

$$\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i \tag{12}$$

where $\lambda_i$ is the ith eigenvalue and $\mathbf{v}_i$ is the corresponding eigenvector. All the eigenvalues of **A** are real. It can be shown that matrix **A** is a positive semi-definite matrix with

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_n \tag{13}$$

## 3. ALGORITHM THE FORMATION OF MINIMAL CYCLE BASES

In this section, Horton's algorithms [2] for the formation of minimal cycle bases is briefly presented. For complete descriptions the reader may refer to the original paper.

***Horton's Algorithm***

**Step 1.** Find a minimum path P(x,y) between each pair of nodes x, y of S.

**Step 2.** For each node v and member {x,y} in the graph, create the cycle C(v,x,y) = P(v,x) + P(v,y) +{x,y} and calculate its length. Degenerate cases in which P(x,y) and P(v,y) have vertices other than v in common can be omitted.

**Step 3.** Order the cycles by their weights.

**Step 4.** Use the Greedy algorithm to find the minimum cycle basis from this set of cycles (A).

## 4. THE PRESENT METHOD FOR CYCLE BASIS SELECTION

In the following, definitions and concepts required for the present method are provided.

***a) Contraction***

An elementary contraction of a graph S is obtained by replacing a path containing all nodes of degree 2 with a new member. The contraction of S into S' is obtained by a sequence of elementary contractions. Naturally the first Betti number does not change in the process of a contraction, Kaveh [1]. This operation is performed in order to reduce the size of the graph and also because the number of members in an intersection of two cycles is unimportant; i.e. a single member is enough to render $C_i \cap C_j$ none-empty and hence produce a nonzero entry in **D**.

### *b) Cycle adjacency graph*

For a graph S with $b_1(S)=M(S)-N(S)+b_0(S)$, the number of elements in its cycle space given by $2^{b_1(S)}-1$. A subspace of this space containing only simple cycles is used by Kaveh [4], and a smaller subspace of S is suggested by Horton [2]. An algorithm for the formation of this subspace is described in Section 3a.

Consider a simple graph as shown in Fig. 1, with $b_1(S)=4$. Horton's algorithm forms 11 cycles, and for brevity only 7 cycles of length 3 and 4 are listed in the following:

| | | | | | | |
|---|---|---|---|---|---|---|
| C: | 1 | 5 | 7 | | | 1 |
| | 1 | 2 | 4 | | | 2 |
| | 1 | 7 | 9 | 4 | | 3 |
| | 5 | 6 | 9 | 4 | | 4 |
| | 2 | 7 | 9 | 3 | | 5 |
| | 7 | 3 | 5 | 6 | 7 | 6 |

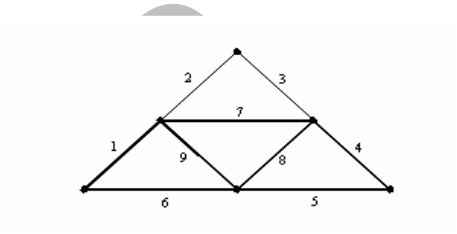



Fig. 1 A simple graph with $b_1(S)=4$

The cycle adjacency graph of S contains the nodes in a one-to-one correspondence with these cycles and two nodes are connected to each other if the corresponding cycles have at least one member in common. Naturally such a graph will not be simple and will have multiple members. The adjacency matrix **A*** of the new graph is constructed as

$$\mathbf{A}^* = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

Once **A*** is formed, the largest eigenvalue $\lambda_1$ with the corresponding eigenvector having all positive entries can easily be calculated. **A*** is real and symmetric and it can be shown that all entries of $\mathbf{A}^{*k}$ are positive. Thus it is primitive, and according to the Perron Frobenious theorem, $\lambda_1$ is real and positive and a simple root of the characteristic equation, corresponding to a unique eigenvector $\mathbf{v}_1$ with all entries positive.

Such an eigenvector can be obtained by the following simple algorithm: Let $\mathbf{v} = \{1,1,\ldots,1\}^t$, then the components of $\mathbf{A}^{*t}\mathbf{v}$ is the number of walks of length k beginning at an arbitrary node of S and ending at $n_i$. If $n_i$ is a good starting node, this number will be larger. Thus for k, one should obtain an average number defined as the *accessibility index* by Gould [9], which indicates how many walks go on average through a node. With a suitable normalization, $\mathbf{A}^{*k}\mathbf{v}$ converges to the largest eigenvector $\mathbf{v}_1$ of **A*** [10].

As an example, for the cycle adjacency matrix discussed in section 4.b, the largest eigenvector is obtained as

$$\mathbf{v}_1 = \begin{bmatrix} 0.3169 \\ 0.3169 \\ 0.3169 \\ 0.4179 \\ 0.4179 \\ 0.4179 \\ 0.4179 \end{bmatrix} \begin{array}{l} 1 \\ 3 \\ 4 \\ 2 \\ 7 \\ 5 \\ 6 \end{array}$$

*c) Algorithm*

This algorithm is simple and consists of the following steps:

**Step 1:** Contract S to S'.
**Step 2:** Form the cycle subspace using Horton's approach.
**Step 3:** Form the cycle adjacency matrix **A***.
**Step 4:** Calculate the largest eigenvector $\mathbf{v}_1$ of **A***.
**Step 5:** Put the entries of $\mathbf{v}_1$ in ascending order in vector **P**.
**Step 6:** Choose the first entry of **P** as the first cycle, remove it from **P**.
**Step 7:** Select the next admissible cycle from the new **P** starting from its first entry.
**Step 8:** Continue Step 7 until $b_1(S)$ admissible cycles, forming a suboptimal cycle basis, is constructed.

*d) Example*

Consider the graph shown in Fig. 2 for which $N \times b_1(S)=30$. Using the modified algorithm of Horton [2,5], thirty cycles are formed corresponding to a 30×30 cycle adjacency matrix. The smallest eigenvector is found and the cycles are ordered accordingly. Suppose the cycles from the end of the list are 3-sided and 4-sided ones. In such a case, the search space will be limited to the following cycles:

C:{(1,2,3),(7,8,9),(1,4,10),(6,9,10),(1,4,9,6), (2,4,7,5),(3,5,6,8),(3,5,7,10),(6,8,7,10),(2,3,4,10)}

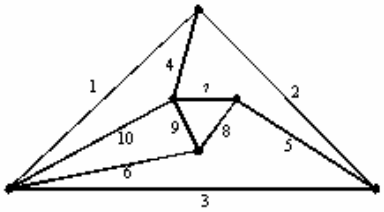



Fig. 2. A simple illustrative example and its member numbering

Thus **A*** becomes a 10×10 matrix. Typical entries of **A*** are as follows:

a(3,1)=a(1,3)=1 since cycle 1 has an overlap with cycle 3, while a(7,3)=a(3,7)=0 since cycle 3 has no overlap with cycle 7. The rest of the entries are found in a similar manner.

Forming $\lambda_1$ and $\mathbf{v}_1$ results in: $\mathbf{v}_1$={0.2648, 0.2648, 0.3008, 0.3008, 0.3324, 0.3324, 0.3362, 0.3362, 0.3404, 0.3404}. Ordering the entries in an ascending order results in a new $\mathbf{v}_1$ corresponding to the cycles with the following order:

P={2,1,4,3,5,8,6,7,10,9}

Therefore first cycle 2 is selected followed by cycles 1, 4 and 3. In this process, when a cycle is not admissible, the next cycle is selected. As an example, the next cycle 5 is not admissible and therefore cycle 8 is chosen. The selected basis consists of C:{(7,8,9), (1,2,3), (6.9,10), (1,4,10), (3,5,7,10)}, corresponding to $\chi(\mathbf{D})=5+2\times7=19$.

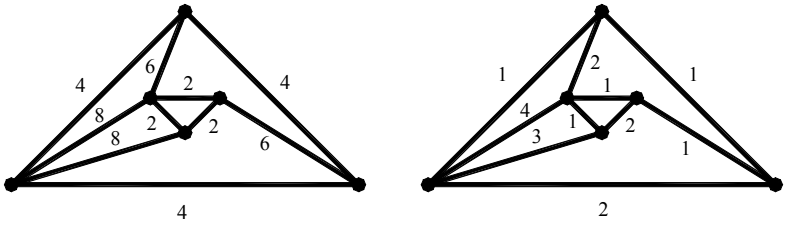
## 5. CONDITIONING OF THE FLEXIBILITY MATRICES

In order to improve the conditioning of the flexibility matrices, the selected cycle bases should have members of higher weights in the overlaps of the cycles. Weights are calculated from the mechanical properties of the members [11]. A basis with the above property results in small off-diagonal terms, leading to well-conditioned flexibility matrices.

As an example, consider the weighted graph as shown in Fig. 3a. **A*** is formed with a typical entry a(i,j) being the sum of the weights of the overlapping members of the cycles $C_i$ and $C_j$. Here the initial cycles are selected by Horton's algorithm. The cycles of the least possible length with maximal weights are then selected. However, since maximizing the weights is the main objective, $\lambda_1$ is obtained and the selection is made from the ordered cycles in $\mathbf{v}_1$. Using the algorithm similar to section 4.3, for the graph of Fig. 3a, the following cycles are selected:

C: {(6,7,8,10),(2,3,4,10),(3,5,6,8),(1,4,10),(6,9,10)}

corresponding to $\chi(\mathbf{D})=5+2\times9=23$, while the sum for the overlaps is 84, as illustrated in Fig. 3b.




(a) The weights of the members (b) The repetition of the members

Fig. 3. A weighted graph

A comparison of the results is made in Table 1 using Log $\lambda_{max}/\lambda_{min}$ as a condition number [11].

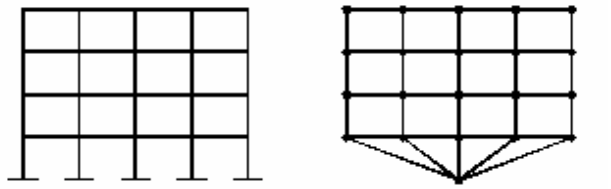
Table 1. Comparison of the results cycle bases with and without weight

| Cycle basis | $\chi(\mathbf{C})$ | $\chi(\mathbf{D})$ | Log $\lambda_{max}/\lambda_{min}$ |
|---|---|---|---|
| without weight | 16 | 19 | 0.6590 |
| with weight | 18 | 23 | 1.8246 |

## 6. EXAMPLES

In this section, some practical models are studied and $\chi(\mathbf{D})$ for the selected cycle bases are provided.

*Example 1*: A planar graph is considered as shown in Fig. 4a. The mathematical model is illustrated in Fig. 4b, where a ground node is used for the supports. For this model, $b_1(S)=16$. Using the present algorithm leads to the following cycle basis with four 3-sided and twelve 4-sided cycles corresponding to $\chi(\mathbf{D})=16+2\times24=64$.

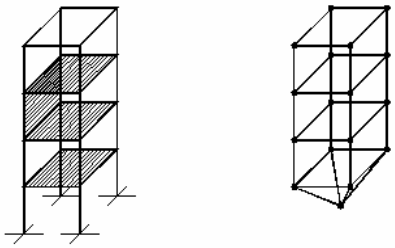


(a) A planar frame (b) The graph model G

Fig. 4. A planar frame and its graph model G

The above frame with 8 spans and 10 stories results in $\chi(\mathbf{D})=80+2\times142=364$, and a frame with 16 spans and 20 stories corresponds to $\chi(\mathbf{D})=320+2\times604=1528$.

*Example 2*: A simple space frame is considered as shown in Fig. 5a. Here the support nodes are identified as a single ground node (Fig. 5b). Using the present algorithm leads to a cycle basis with all 4-sided cycles except those shaded in Fig. 5a, corresponding to $\chi(\mathbf{D})=16+2\times28=72$.



(a) (b)

Fig. 5. A simple space frame and its graph model

Similar space frames are studied as 2×2×2 and 4×4×4 grids leading to $\chi(\mathbf{D}) = 28+2\times66 = 160$, and $\chi(\mathbf{D}) = 160+2\times479 = 1118$, respectively.

*Example 3*: A graph model in the form of a 3×3×1 grid is considered as shown in Fig. 6. Using Horton's algorithm leads to the formation of minimal cycle basis corresponding to $\chi(\mathbf{C})=132$ and $\chi(\mathbf{D}) = 33+2\times100 = 233$. For the cycle basis selected by the present algorithm $\chi(\mathbf{C})=132$ and $\chi(\mathbf{D})=33 +2\times77=187$.

*Example 4*: A graph model of a three story structure in the form of a 3×3×3 grid with a node at the middle of each cube connected to 8 corner node, Fig. 7. Using Horton's algorithm, 324 three-sided and 108 four-sided cycles are generated. Higher length cycles could also be generated, however, since an optimal cycle basis for a symmetric graph of Fig. 7 does not seem to have a single such cycle, higher length cycles are not selected. The weighted adjacency matrix **A*** is formed and using the present algorithm, a suboptimal cycle basis is generated. This basis consists of 270 cycles of length, three of which correspond to $\chi(\mathbf{C}) = 270\times3 = 810$ and $\chi(\mathbf{D}) = 270+ 2\times617 = 1504$ non-zero entries.

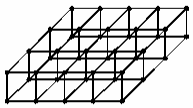


Fig. 6. A 3×3×1 cube-type graph with $b_1(S)=33$

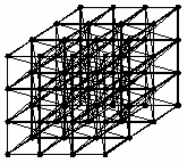


Fig. 7. The graph model of a space structure S

## 7. CONCLUSIONS

The present algorithm is very efficient and makes the fast and economical generation of subminimal cycle bases feasible. It leads to the formation of minimal cycle bases for most of the practical models.

The formation of α self-equilibrating stress system on each element of the selected cycle basis leads to the formation of highly sparse **G** matrices, making an efficient flexibility analysis of frame structures feasible. The application of this method is by no means limited to the analysis of a structure. It can be applied to the analysis of other systems such as hydraulic and electrical networks.

## REFERENCES


1. Kaveh, A. (2004). *Structural mechanics: graph and matrix methods*. Research Studies Press (John Wiley), Exeter, UK, 3rd ed.
2. Horton, J. D. (1987). A polynomial time algorithm to find the shortest cycle basis of a graph. *SIAM Journal of Computing*, *16*, 358-366.
3. Kaveh, A. & Roosta, G. R. (1994). Revised greedy algorithm for the formation of a minimal cycle basis of a graph. *Communications in Applied Numerical Methods*, *10*, 523-530.
4. Kaveh, A. (1976). Improved cycle bases for the flexibility analysis of structures. *Computer Methods in Applied Mechanics and Engineering, 9*, 267-272.
5. Kaveh, A. (1997). *Optimal structural analysis*. RSP (John Wiley), Exeter, UK.
6. Kaveh, A. (1988). Suboptimal cycle bases of graphs for the flexibility analysis of skeletal structures. *Computer Methods in Applied Mechanics and Engineering*, *71*, 259-271.
7. Kaveh, A. (1989). Suboptimal cycle bases of a graph for mesh analysis of networks. *International Journal of Networks*, *19*, 273-279.
8. Henderson, J. C. de C. & Maunder, E. W. A. (1969). A problem in applied topology. *Journal of the Institute of Mathematics and its Applications*, *5*, 254-269.
9. Gould, P. (1967). The geographical interpretation of eigenvalues. *Transactions of the Institute of British Geographer*, *42*, 53-58.
10. Straffing, P. D. (1983). Linear algebra in geography, eigenvectors of networks. *Mathematical Magazine*, *53*, 355-362.
11. Kaveh, A. (1991). Optimizing the conditioning of structural flexibility matrices. *Computers and Structures*, *41*, 489-494.