IJST, Transactions of Civil Engineering, Vol. 37, No. C1, pp 17-32
Printed in The Islamic Republic of Iran, 2013

PLATE BENDING FINITE ELEMENT ANALYSIS BY THE FORCE METHOD USING ANT COLONY OPTIMIZATION*

A. KAVEH** AND M. S. MASSOUDI

Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran-16, I. R. of Iran
Email: alikaveh@iust.ac.ir

Abstract– The formation of null basis for equilibrium matrix is the most important part of the finite element analysis when the force method is utilized. For an optimal analysis, the selected null basis matrices should be sparse and banded leading to sparse, banded and well-conditioned flexibility matrices. In this paper, an efficient algorithm is developed for the formation of null basis of triangular and rectangular plate bending finite element models, corresponding to highly sparse flexibility matrices. This is achieved by applying a modified ant colony system. An integer linear programming formulation is also presented to evaluate the quality of the results obtained by the proposed ant colony system algorithm. The efficiency of the present algorithm is illustrated through some examples.

Keywords– Finite elements, triangular and rectangular elements, force method, ant colony system (ACS), flexibility matrix, null basis matrix, sparsity

1. INTRODUCTION

The force method of structural analysis in which the member forces are used as unknowns is appealing to engineers since the properties of members of a structure most often depend on the member forces rather than joint displacements. This method was used extensively until 1960. The advent of the digital computer and the amenability of the displacement method for computation attracted most researchers. As a result, the force method and some of the advantages it offers in non-linear analysis and optimization has been neglected.

Five different approaches are adopted for the force method of structural analysis which can be classified as

1. Topological (graph theoretical) force methods
2. Algebraic force methods
3. Mixed algebraic-graph theoretical force methods
4. Integrated force method
5. Meta-heuristic force methods

Topological methods have been developed by Henderson [1], Maunder [2], and Henderson and Maunder [3] for rigid-jointed skeletal structures using the cycle bases of their topological models. In this type of force method, the cycle basis of the structural model is selected using ideas from topology such as manifold embedding and collapsible complex embedding [3]. Graph theoretical methods suitable for computer programming are due to Kaveh [4-6], where the use of the Greedy algorithm has been suggested

*Received by the editors April 25, 2010; Accepted June 15, 2011.
**Corresponding author

for selection of the minimal cycle basis. These methods are generalized to cover different types of skeletal structures such as rigid-jointed frames, pin-jointed planar trusses, and ball-jointed space trusses [7]. Applications are extended to finite element models by Cassell [8] and Kaveh et. al. [9], Kaveh and Koohestani [10], and Kaveh and Nasiri [11].

Algebraic methods have been developed by Denke [11], Robinson [12], Topçu [13] Kaneko et al. [14], Soyer and Topçu [15], and Kaveh and Rahami [16]. Mixed algebraic-topological methods have been used by Gilbert et al. [17], Coleman and Pothen [18-19], Pothen [20] and Heath et al. [21]. The integrated force method has been developed by Patnaik [22-23], in which member forces are used as variables; the equilibrium equations and the compatibility conditions are satisfied simultaneously in terms of these variables. Meta-heuristics algorithms have recently been applied to the force methods by Kaveh and Daei [24-25] and Kaveh and Malakouti [26].

The force method of structural analysis requires the formation of a maximal set of independent self-equilibrating stress systems (SESs), known as a *null basis* [24, 25]. The elements of this basis form the columns of an $m \times \gamma(S)$ matrix, B_1, known as the *self-stress matrix*. The main problem in the application of the force method is the formation of a self-stress matrix corresponding to a sparse flexibility matrix $G = B_1^t F_m B_1$, where F_m contains the flexibility matrices of the individual members of the structure in a block diagonal form. The graph theoretical methods for the force method are very efficient for skeletal structures and, in particular, for rigid-jointed frames. For a general structure, the underlying graph or hypergraph of a SES has not yet been completely defined, and further research is needed. Algebraic methods, on the other hand, are formulated in a more general form to cover different types of structures such as skeletal structures and finite element models (FEM). The main drawbacks of these methods are the large storage requirements and the higher number of operations.

Heuristic algorithms, such as ant colony algorithms, have found many applications in optimization problems in the last decade. The essence of these algorithms lies in the fact that their capability to converge to a good solution does not depend on the specific search space to which they are applied. In this paper, the ant colony system (ACS) which is a variation of the ant colony optimization (ACO) is applied to the formation of null bases of triangular and rectangular plate bending finite element models corresponding to highly sparse and banded flexibility matrices. An integer linear programming formulation is also used to evaluate the quality of the results obtained by the proposed ACS algorithm [24]. The efficiency of the present method is illustrated through simple examples.

2. FORMULATION OF THE FORCE METHOD

Consider a structure *S* which is $\gamma(S)$ times statically indeterminate. Then $\gamma(S)$ independent unknown forces should be selected as redundants. These unknown forces can be chosen from external reactions and/or internal forces of the structure. These redundants are denoted by a vector as $q = \{q_1, q_2, ..., q_{\gamma(S)}\}^t$. In order to obtain a statically determinate structure, known as the *basic* (released or primary) *structure* of *S*, the constraints corresponding to redundants are removed. Consider the joint loads vector as $p = \{p_1, p_2, ..., p_n\}^t$, where *n* is number of entries of the applied nodal load vector and let *r* denote the *m*-dimensional vector of generalized independent element forces. The equilibrium conditions of the structure can then be expressed as

$$Ar = p \tag{1}$$

where *A* is an *n*×*m equilibrium matrix*. The element forces can be written as

$$r = \mathrm{B}_0\,\mathrm{p} + \mathrm{B}_1\mathrm{q} \tag{2}$$

where B_0 is an *m*×*n* matrix such that $A \times B_0$ is an *n*×*n* identity matrix and B_1 is an $m \times \gamma(S)$ matrix such that $A \times B_1$ is an $n \times \gamma(S)$ zero matrix. B_1 and B_0 always exist for a structure and in fact many of them

can be found for a structure. Each column of B_1 is known as a *self-equilibrating stress system* (SES) or a *null vector*. A maximal set of SESs (null vectors) is known as a *statical basis* (*null basis*). B_1 is called a *self-equilibrating stress matrix* (*null basis matrix*).

Minimizing the complementary potential energy requires that *r* minimize the quadratic form

$$\frac{1}{2} r^t F_m r \tag{3}$$

subjected to the constraint as in Eq. (1). F_m is an $m \times m$ block diagonal element flexibility matrix. Using Eq. (2) it can be seen that *q* must satisfy the following equation:

$$(B_1^t F_m B_1) q = -(B_1^t F_m B_0) p \tag{4}$$

where $G = B_1^t F_m B_1$ is the *overall flexibility matrix* of the structure. Computing the redundant forces *q* from Eq. (4), *r* can be found using Eq. (2). The structure of *G* is again important and its sparsity, bandwidth and conditioning govern the efficiency of the force method. For the sparsity of *G* one can search for a sparse B_1 matrix which is often referred to as the *sparse null basis* problem [27].

3. TRIANGULAR AND RECTANGULAR PLATE BENDING ELEMENTS

The element nodal forces and moments for these elements are illustrated in Fig. 1.

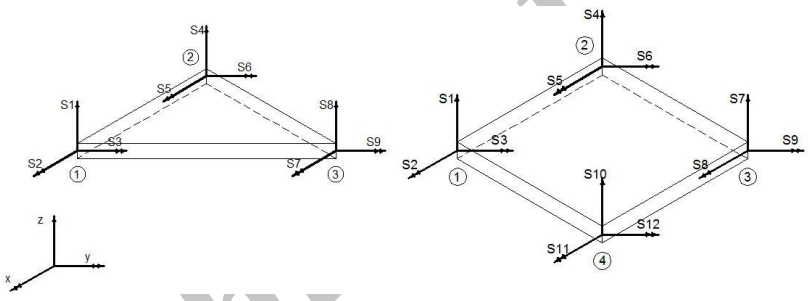

Fig. 1. Element nodal forces and moments for triangular and rectangular elements

The system of independent element forces for a rectangular finite element contains four symmetric moments F_1, F_3, F_5, F_7 and four anti-symmetric moments F_2, F_4, F_6, F_8 and a set of four forces. This system is shown in Fig. 2.

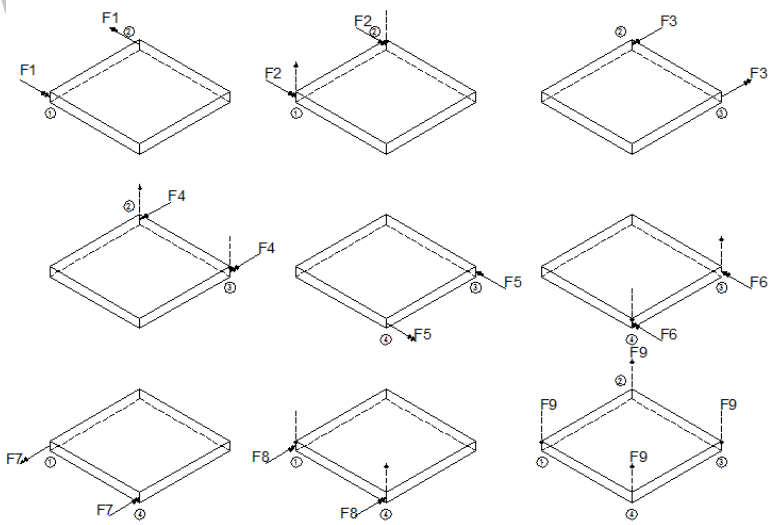

Fig. 2. Independent element forces and moments for a rectangular finite element

The system of independent element forces for a triangular finite element can be defined as three symmetric moments F_1, F_3, F_5 and three anti-symmetric moments F_2, F_4, F_6. This system is illustrated in Fig. 3.

These forces can be related to element nodal forces for rectangular and triangular elements by a 12×9 and 9×6 transformation matrix, respectively [28].

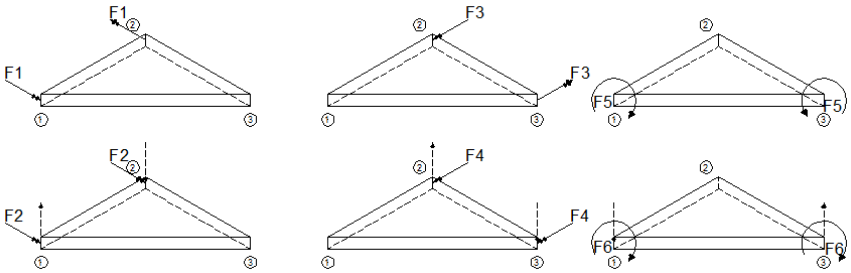

Fig. 3. Independent element forces and moments for a triangular finite element

4. MATHEMATICAL MODELING FOR OPTIMIZATION PROBLEM

Since the *overall flexibility matrix* of the structure G is $B_1^t F_m B_1$, for the sparsity of G one should select a null basis corresponding to sparse B_1 matrix, which is often referred to as the sparse null basis problem. The main objective of this paper is to find sparse self-stress matrices to simplify the solution and to ensure the formation of well-conditioned flexibility matrices.

For a SES (null vector), no applied load is required, thus the equilibrium conditions can be expressed as

$$AB_1 = 0 \tag{5}$$

This equation shows that the columns of the matrix A, which is an $n\times m$ matrix with rank of n are linearly dependent. There are $m-n = t$ independent columns of B_1 which will satisfy this equation, thus forming a set of SESs as a basis.

It must be noted that there are many sets of SESs which have independent columns and satisfy the above equation. However, the problem is to find a set corresponding to highly sparse B_1 matrix.

Let us denote the columns of matrix B_1 by S_i as

$$B_1 = [S_1, S_2, ..., S_g, ..., S_t] \tag{6}$$

Suppose the first null vector S_1 is found, then it can be normalized by the following equation:

$$e_1^t S_1 = 1 \tag{7}$$

where e_1={10...0...0} is an $m\times 1$ unit vector with 1in the first entry position. The second column S_2 can be normalized and must be independent of S_1 and these conditions are expressed as

$$\begin{aligned} e_1^t S_2 &= 0 \\ e_2^t S_2 &= 1 \end{aligned} \tag{8}$$

where e_2={010...0...0} is an $m\times 1$ unit vector with 1 in the second entry position. It is obvious that the conditions analogous to these relationships can be formed for the subsequent null vectors.

In this section, first the mathematical programming is employed for selecting the column S_2 and then extended for the formation of the complete set of the SESs. The first null vector, S_1, is arbitrary. Now we find the second null vector, S_2, satisfying the following equations:

$$\begin{aligned} AS_2 &= 0 \\ e_1^t S_2 &= 0 \\ e_2^t S_2 &= 1 \end{aligned} \tag{9}$$

or more concisely

$$\begin{bmatrix} & A & \\ I_2 & & 0 \end{bmatrix} S_2 = \begin{bmatrix} 0 \\ \bar{e}_2 \end{bmatrix} \tag{10}$$

where $\bar{e}_2 = \{01\}$ is a 2×1 unit vector, with 1 in the gth position which minimizes the function $Z = |S_2|$. Here, $|S_2|$ denotes the cardinality of S_2 and it is equal to the number of non-zero entries of S_2.

This can be generalized for the gth null vector S_g, after all the previous null vectors up to $g-1$ have been obtained. The problem can now be stated as follows:

Minimize the objective function of the form $Z = |S_g|$ satisfying

$$\begin{bmatrix} & A & \\ I_g & & 0 \end{bmatrix} S_g = \begin{bmatrix} 0 \\ \bar{e}_g \end{bmatrix} \tag{11}$$

where $\bar{e}_g = \{0\ 0...0...1\}$ is a g×1 unit vector, with 1 in the gth entry position.

By performing the above series of operations, t null vectors $B_1 = [S_1, S_2, ..., S_g, ..., S_t]$ are generated consecutively one after another. It should be noted that for the last null vector (the tth system ($t = m-n$)), there is no choice. This is because the number of equations is equal to the number of variables, i.e. there are n original equations in the n×m matrix A, and $m-n = t$ orthogonalising equations, thus forming $n+t = m$ equations (with the number of variables being equal to m), leading to a unique solution for the last null vector. A point to notice is the numbering of the members in the structure. The importance of numbering pattern can be recognized by considering the additional equations used in the normalization and orthogonalization. Here, the ant colony system will be applied to choose the members such that the resulting null vectors lead to sparse B_1 matrices.

5. OPTIMIZATION BY ANT COLONY SYSTEMS

A meta-heuristic algorithm based on the ants' behavior was developed in the early 1990s by Dorigo and Gambardella [29]. This algorithm was called ant colony optimization because it was motivated by social behavior of ants. Ant colony system is a variation of the ACO which has proven to behave more robustly and provide far better results for certain problems. In this work, ACS is chosen as a suitable tool for finding sparse null vectors. A brief description of ACO is given in the next section when describing the process of adapting ACS to the problem of finding sparse null basis. This method is also applied extensively in structural optimization, an example of which can be found in the work of Kaveh and Masoudi [30].

The building blocks of these algorithms are cooperative agents called ants. These agents have simple capabilities, which make their behavior similar to real ants. Real ants are capable of finding the shortest path from food source to their nest or vice versa by smelling pheromones which are chemical substances they leave on the ground while walking. Each ant probabilistically prefers to follow a direction rich in

pheromone. Since pheromones evaporate and lose strength over time, the final result is that more ants tend to pass over the shortest path and this path is visited more often as the amount of pheromone being laid increases. As an illustrative example, consider the sketch shown in Fig. 4. The number of dashed lines in Fig. 4(c) is approximately proportional to the amount of pheromone deposited by ants.

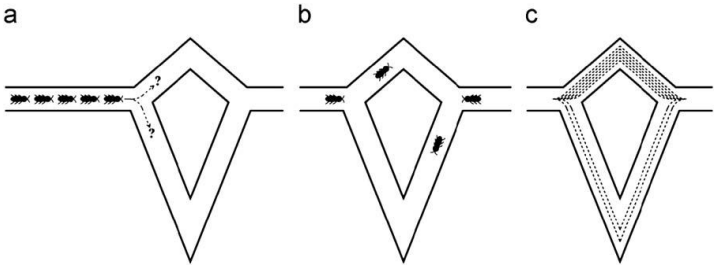

Fig. 4. Ant technique to find an optimum solution

6. THE EFFECT OF GENERATOR SEQUENCE AND EDGE ORDERING ON THE SPARSITY OF NULL BASIS

According to the proposed mathematical modeling, the numbering of the members affects the results of the selected null vectors. This can be found out by considering the additional equations which are used in the process of normalization and orthogonalisation. Taking $e_1^t S_1 = 1$, it is necessary to have a force equal to unity in the first entry of S_1 vector. This entry is called the *generator* of S_1. Since $e_1^t S_2 = 0$ and $e_2^t S_2 = 1$, the first entry in S_2 vector, which is the generator of S_1, must be zero, while the second entry must be equal to one. This second entry is known as the generator for the second column in null basis matrix, i.e. the second null vector S_2. Therefore, for the gth null vector, S_g, the forces in the previous generators are zero while in its generator position it is equal to one.

As an example, consider a finite element model as shown in Fig. 5. This model is divided into 8 rectangular elements and its degree of static indeterminacy is equal to 27.

The numbering of the edges of the interface graph corresponding to the rectangular elements is shown in the discretized structure in Fig. 6, Refs. [31-32].

First, every set of four edges of interface graph, corresponding to two elements of the FEM with common edges, contains two SESs with two non-zero entries. The set of four edges corresponding to the common edges of the two elements i and j (i < j) has two edges m_i and n_i (m < n), and r_j and s_j (r < s). The two SESs with two edges obtained from this set and consequently two null vectors can be extracted. A null vector with non-zero entries (−1,1) at rows (m,r) and another null vector with non-zero entries (1,1) at rows (n,s) are formed. It is obvious that these null vectors will be among the columns of a minimal null basis matrix, because there are just two non-zero entries in the corresponding null vectors.

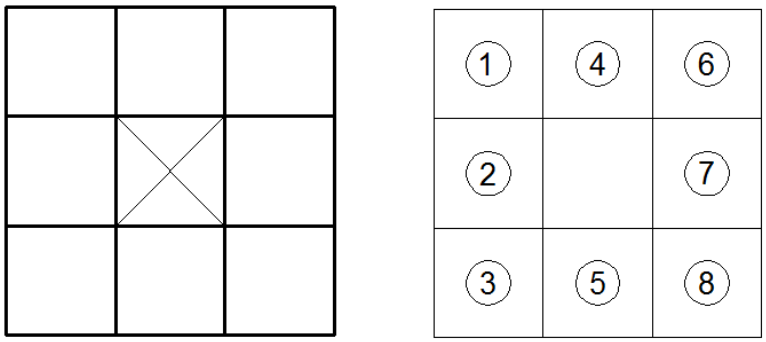

Fig. 5. A finite element model with 8 elements and 27 degrees of static indeterminacy

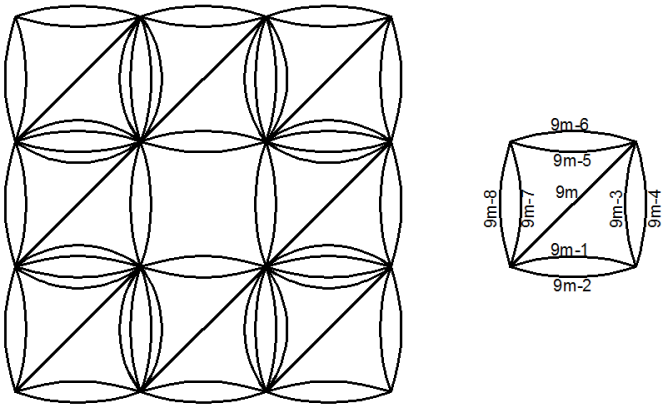

Fig. 6. The interface graph and the pattern for numbering of its edges

Since for the remaining null vectors, the forces in the previous generators should be zero, after removal of the generating edges of all the double edges from the interface graph, the remaining null vectors should be selected from the remaining interface graph.

After deleting these generators, the degree of statical indeterminacy for the remaining interface graph will be obtained as follows:

$$DSI = 27 - 16 = 11$$

This means that 11 SESs should be extracted from the remaining subgraph. After generating these null vectors, the previously selected null vectors should be added to them and all vectors must be gathered in a single matrix.

First, the set of generators is chosen as

$$(17 \to 44 \to 34 \to 25 \to 59 \to 40 \to 35 \to 68 \to 11 \to 36 \to 23)$$

In this case, the null vectors corresponding to this sequence of generators have 175 non-zero entries. The members of these 11 null vectors are illustrated in the following matrix:

$$M = \begin{bmatrix}
(1)\{10\ \ 11\ \ 14\ \ 15\ \ 16\ \ 17\ \ 18\ \ 19\ \ 20\ \ 23\ \ 24\ \ 25\ \ 26\ \ 39\ \ 40\ \ 43\ \ 44\ \ 45\} \\
(2)\{32\ \ 33\ \ 39\ \ 40\ \ 41\ \ 42\ \ 43\ \ 44\ \ 45\ \ 50\ \ 51\ \ 54\ \ 55\ \ 59\ \ 61\ \ 62\ \ 68\ \ 69\ \ 70\ \ 71\} \\
(3)\{30\ \ 31\ \ 34\ \ 35\ \ 36\ \ 48\ \ 49\ \ 52\ \ 53\ \ 54\} \\
(4)\{3\ \ 4\ \ 5\ \ 6\ \ 7\ \ 8\ \ 9\ \ 14\ \ 18\ \ 23\ \ 24\ \ 25\ \ 26\ \ 27\ \ 30\ \ 31\ \ 32\ \ 33\ \ 35\ \ 36 \\
\qquad 41\ \ 42\ \ 43\ \ 45\ \ 48\ \ 49\ \ 50\ \ 51\ \ 52\ \ 53\ \ 56\ \ 59\ \ 60\ \ 68\ \ 69\ \ 70\ \ 71\} \\
(5)\{41\ \ 42\ \ 55\ \ 56\ \ 59\ \ 60\ \ 63\ \ 68\ \ 69\ \ 72\} \\
(6)\{14\ \ 15\ \ 23\ \ 24\ \ 30\ \ 31\ \ 36\ \ 39\ \ 40\ \ 41\ \ 42\ \ 45\ \ 48\ \ 49\ \ 54\ \ 55\ \ 56\ \ 61\ \ 62\ \ 63\} \\
(7)\{3\ \ 4\ \ 7\ \ 8\ \ 9\ \ 31\ \ 35\ \ 36\ \ 48\ \ 49\ \ 52\ \ 53\ \ 54\} \\
(8)\{32\ \ 33\ \ 41\ \ 42\ \ 50\ \ 51\ \ 54\ \ 56\ \ 60\ \ 63\ \ 68\ \ 69\ \ 72\} \\
(9)\{1\ \ 2\ \ 5\ \ 6\ \ 9\ \ 10\ \ 11\ \ 14\ \ 15\ \ 18\} \\
(10)\{5\ \ 6\ \ 14\ \ 23\ \ 24\ \ 32\ \ 33\ \ 36\ \ 41\ \ 42\ \ 45\ \ 55\} \\
(11)\{1\ \ 2\ \ 5\ \ 6\ \ 9\ \ 10\ \ 14\ \ 19\ \ 20\ \ 23\ \ 24\ \ 27\}
\end{bmatrix}$$

For example, the interface subgraph corresponding to the first selected null vector (first row of above matrix) is illustrated in Fig. 7.

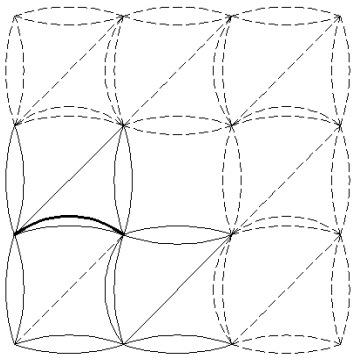

Fig. 7. The interface subgraph corresponding to the first selected null vector

In this figure, the bold line shows the corresponding generator.

In order to show the effect of choosing different sets of generators, the sequence of generators in the second attempt is changed to

$$(32 \to 5 \to 16 \to 68 \to 10 \to 56 \to 39 \to 72 \to 3 \to 26 \to 31)$$

In this sequence of selection, the null vectors have all together 161 non-zero entries. Obviously, this system is sparser than the previous null basis, resulting in a more sparse flexibility matrix. Therefore, in the next section each sequence of edges as generators is considered as a tour for ant travel, and the best ant search for the generators is the one leading to the sparsest possible null basis. An efficient algorithm based on the ant colony system is presented in the following section for finding an optimum solution.

7. ACS ALGORITHM FOR THE FORMATION OF SPARSE NULL BASIS

In order to apply the ACO algorithm to a specific problem, it is necessary to represent it as a set of different paths for ants to travel. In the problem of finding sparse null basis, different sequence of generators is considered as a tour for an ant to travel, therefore the cooperative ant agents search to find the best generator sequence resulting in a sparse null basis. Since both the edge numbering and its order in the generator sequence are important, the pheromone amount is specified by two indices (τ_{ij}), where the index *i* is the generator order in the set of generators, and the index *j* shows the edge number. As an example, τ_{25} shows the amount of pheromone for selection of the edge number 5 as the 2nd generator in the generators set. In our algorithm, first *m* artificial ants are initially positioned on *m* edges of elements as primary generators, and then ACS algorithm is applied as follows:

An ant *k* chooses the *r*th generator by applying the rule of the following equation:

$$j = \begin{cases} \arg\max_{u \in L_k(r)} (\tau_{ru} . \eta_{ru}^{\beta}) & if \quad q <= q_0 \\ J & otherwise \end{cases} \tag{12}$$

Where q is a random number uniformly distributed in [0..1], q_0 is a parameter $0 \le q_0 \le 1$, and J is a random variable selected according to the probability distribution given in the following equation:

$$P_{rs}^{k} = \begin{cases} \dfrac{\tau_{rs} \eta_{rs}^{\beta}}{\sum_{u \in L_k(r)} \tau_{ru} \eta_{ru}^{\beta}} & if \quad S \in L_k(r) \\ 0 & otherwise \end{cases} \tag{13}$$

$L_k(r)$ is the set of generators that remain, to be chosen by ant *k* as the *r*th generator and τ_{rs} is the amount of pheromone deposited on the generator number *s* as a candidate for being the *r*th generator. It is assumed that there is an equal amount of pheromone τ_0, deposited initially on each generator. η_{rs} is the corresponding heuristic value which remains constant throughout the iterations and, unlike pheromone amount, is not modified. Moreover, β is a parameter for controlling the relative importance between τ and η.

After an ant chooses one edge as a generator, the local updating rule on that chosen generator is performed in order to shuffle the solution and prevent focusing on a specific solution. The local updating rule modifies the amount of pheromone by

$$\tau_{rs} \leftarrow (1-\xi)\tau_{rs} + \xi\tau_0 \tag{14}$$

where $0 < \xi < 1$ is a parameter for adjusting the pheromone previously deposited on τ_{rs}.

Once all the ants complete their own tours, the pheromone will be updated for all the edges according to the global updating rule. This pheromone updating is intended to allocate a greater amount of pheromone to shorter tours. The rule is given by the following equation:

$$\tau_{rs} \leftarrow (1-\rho)\tau_{rs} + \rho\Delta\tau_{rs} \tag{15}$$

where

$$\Delta\tau_{rs} = \begin{cases} (D_{gb})^{-1} & \text{if } (r,s) \in \text{global best tour} \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

Here, D_{gb} is the sparsity coefficient of the globally best tour (number of non-zero elements in the selected null basis) and $0 < \rho < 1$ is the pheromone decay parameter. The best ant tries to find the sparse null basis.

As an example, consider a finite element model as shown in Fig. 8. This model is divided into 11 rectangular elements. The degree of static indeterminacy of this model is equal to 42 ($9\times11-3\times20+3=42$).

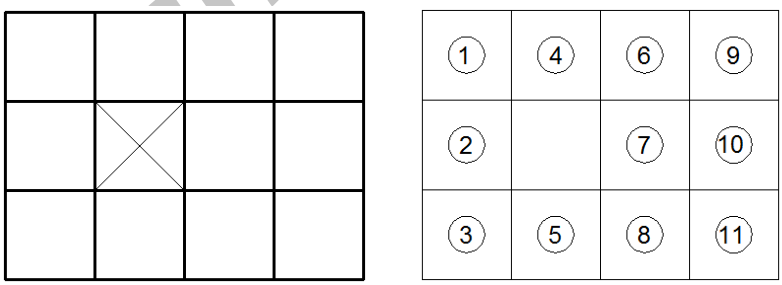

Fig. 8. A finite element model with 11 elements and 42 degree of static indeterminacy

The interface graph corresponding to this FEM is illustrated in Fig. 9.

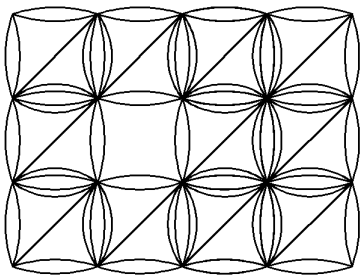

Fig. 9. The interface graph corresponding to the considered FEM

After generating the 13 null vectors corresponding to sets of four edges in the interface graph and removing the generators, the remaining interface subgraph is obtained (Fig. 10). The degree of indeterminacy for this subgraph is equal to 16 ($42-2\times13=16$).

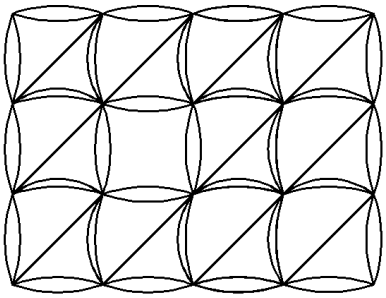

Fig. 10. The remaining interface graph after the removal of the generators

The number of the remaining edges in the interface subgraph is equal to 73 ($9\times11-2\times13=73$). Thus, ACS algorithms should select 16 edges as generators among these 73 edges. The null basis matrix which is obtained by this sequence of generators has minimum number of non-zero entries.

The best tour which is obtained by ants is

$$(98\rightarrow96\rightarrow86\rightarrow70\rightarrow3\rightarrow89\rightarrow69\rightarrow60\rightarrow75\rightarrow56\rightarrow55\rightarrow44\rightarrow39\rightarrow34\rightarrow24\rightarrow18)$$

The interface subgraphs corresponding to this sequence of generators are illustrated in Fig. 11. In this figure, the bold lines show the generators.

The total number of non-zero entries of the null basis matrix is 224, having the pattern as shown in Fig. 12a. In this figure, the comparison between this method, graph-theoretical approach and the algebraic method are illustrated.

It should be noted that the ACS algorithm can improve the results for problems which have one or more cut-outs. In other words, in the problems which have no cut-outs, ACS and graph-theory approaches result in the identical null basis matrix.

Due to the nature of the present method (ACS algorithm), its computational time is quite high for large examples. Therefore, in the following section, a new combined method is described.

8. GRAPH THEORY FOR DECREASING COMPUTATIONAL TIME OF THE ACS ALGORITHM

In the previous section, ACS algorithm has been described and by solving an example, it has been concluded that for plates with one or more cut-outs, ACS algorithm concludes in good results in comparison to the other methods. However, the computational time of ACS algorithm becomes very high for large scale examples. In order to decrease the computational time, graph theory can be used and an approach consisting of the ACS and graph theory will be introduced.

In this hybrid method, the null vectors corresponding to the cycles of generalized associate digraph will be formed via graph theory and for each cut-out, an ACS algorithm should be used for formation of three null vectors. In other words, after removing the generators corresponding to null vectors which are achieved by graph theory method, ACS can extract three remaining null vectors for each cut-out.

As an example, for the FEM illustrated in Fig. 9, by using graph theory and ACS simultaneously, the null basis matrix is obtained having 254 non-zero entries, with the pattern as shown in Fig. 12d.

The null basis matrix obtained in this section has more non-zero entries than the pure ACS (previous section), but the computational time for this method is decreased for large examples. In addition, the null basis matrices, which are calculated by using the graph theory and ACS simultaneously, are sparser than the results of the pure graph theory method.

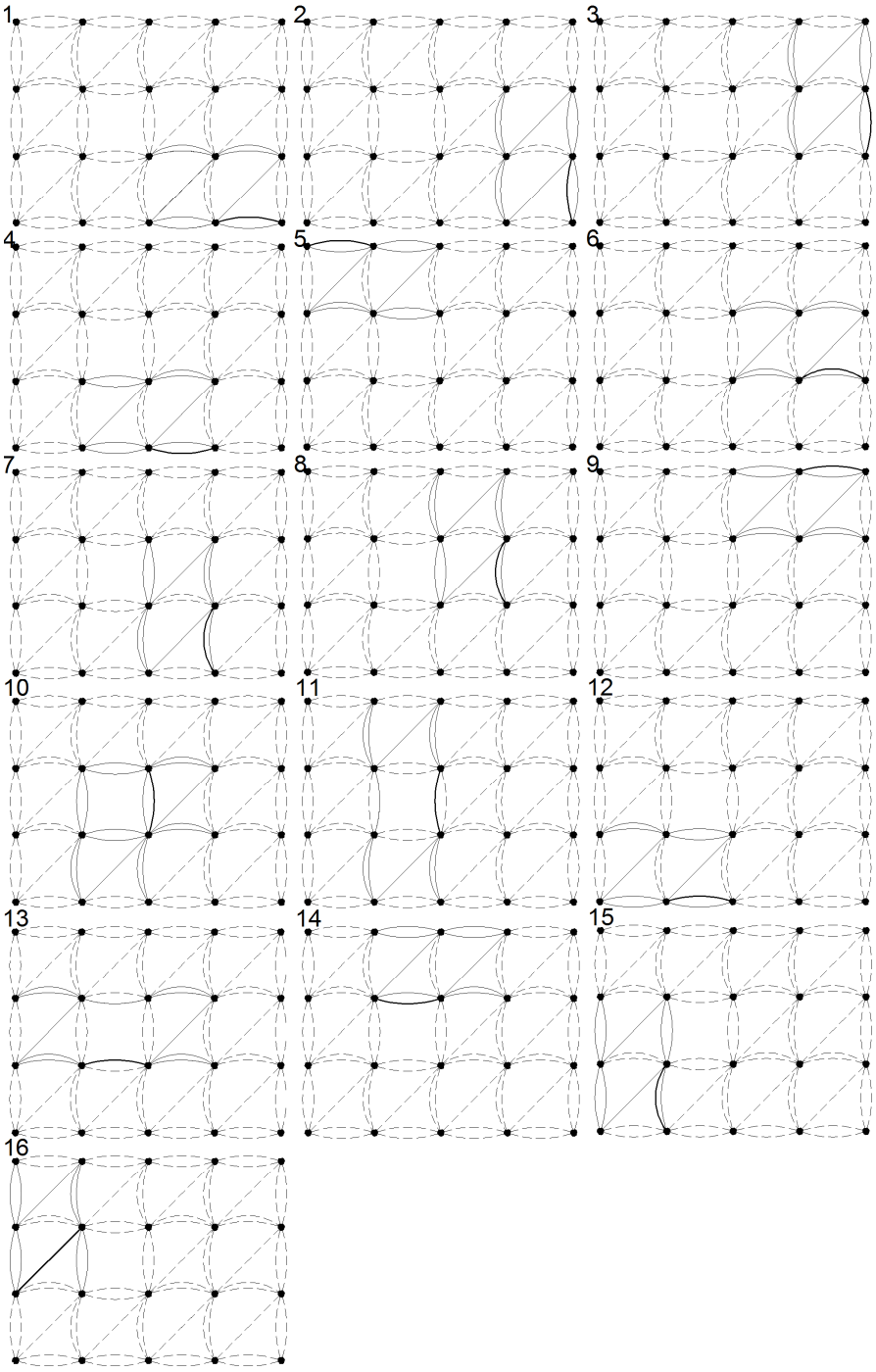

Fig. 11. The interface subgraphs corresponding to the selected sparse null vectors

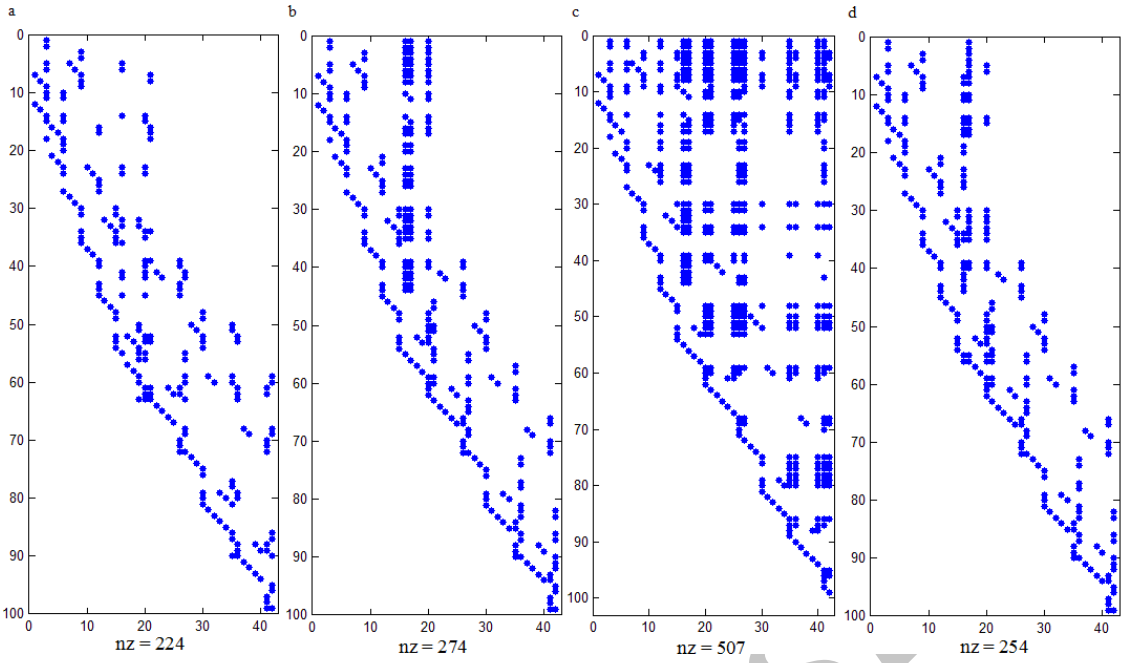

Fig. 12. Pattern of the null basis matrix resulted by, a) the best ant, b) graph-theoretical method, c) *LU* factorization method and d) using graph theory and ACS simultaneously

9. NUMERICAL RESULTS

In this section, the performance of the proposed algorithm is illustrated by solving some examples. All of the models are assumed to be supported in a statically determinate fashion. The null basis matrices for each model are calculated using the *LU* factorization [31] approach, Kaveh-Massoudi [32] graph theoretical method, and the present method. The results are compared for their sparsity. These algorithms are coded by MATLAB.

Example 1. In this example, a FEM with rectangular plate bending element and containing an opening is considered, as shown in Fig. 13.

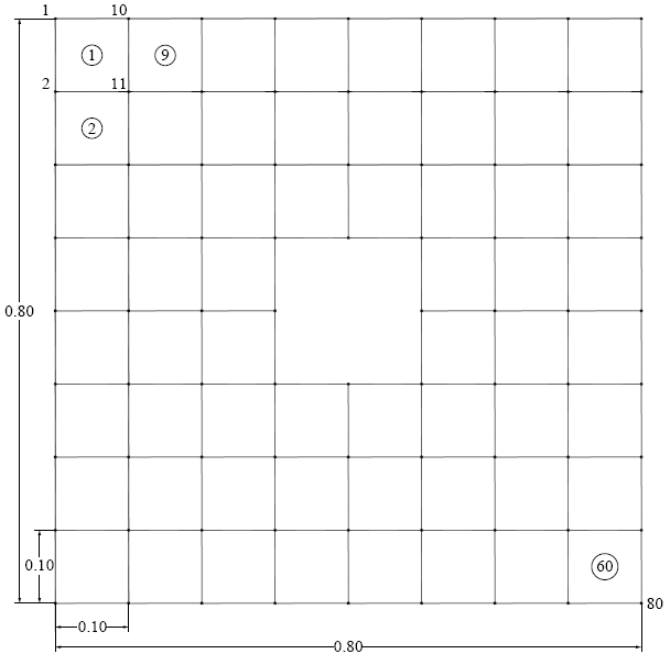

Fig. 13. A rectangular FEM with an opening and its numbering

Pattern of the null basis matrix for three methods and the comparison of the results are shown in Fig. 14. The sparsity, computational time and norm of AB_1 matrix are compared with *LU* factorization and graph theoretical method, Table 1.

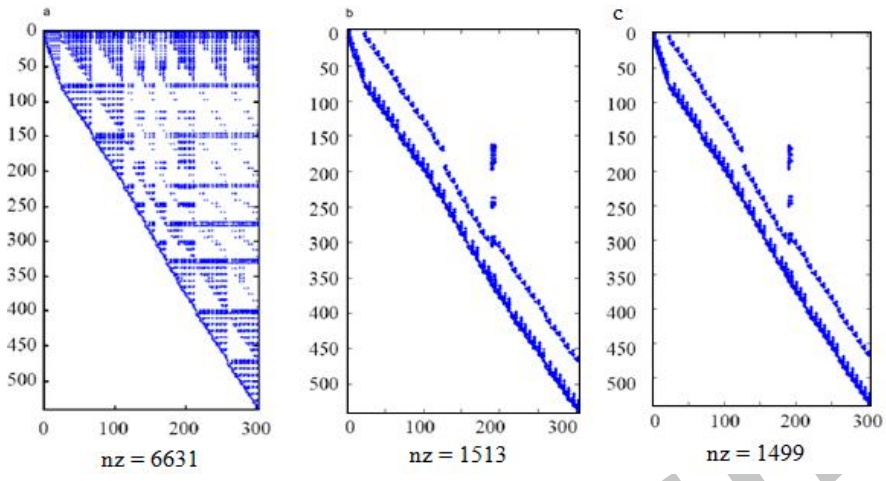

Fig. 14. Pattern of the null basis matrix resulted by, a) *LU* factorization method, b) graph-theoretical method and c) the present method

Table 1. Comparison of the sparsity, computational time and accuracy of the present algorithm versus LU factorization and graph-theoretical method

	Number of non-zero entries(nz)	$\frac{Time}{LU\ Time}$	$\| AB_1 \|_{fro}$
LU factorization	6631	1	5.37e-12
Graph-theoretical	1513	0.6123	4.81e-14
Present algorithm (hybrid method)	1499	8.4074	2.22e-16

Example 2. In this example, a FEM with triangular plate bending element which contains an opening is considered, as shown in Fig. 15. Pattern of the null basis matrix for three methods and the comparison of the results are shown in Fig. 16. The sparsity, computational time and norm of AB_1 matrix are compared with *LU* factorization and graph theoretical method, Table 2.

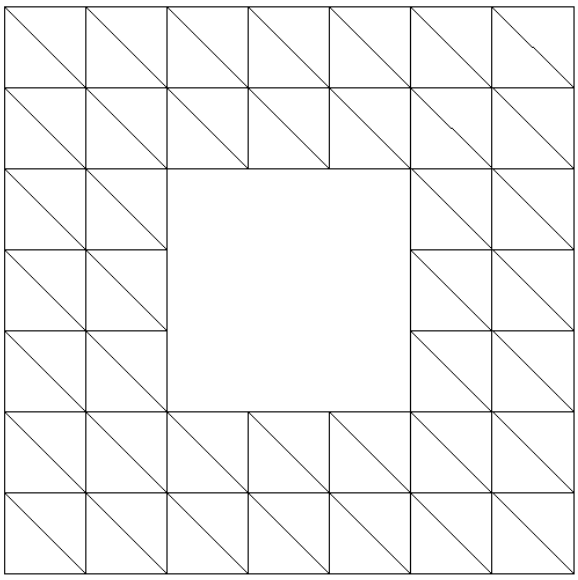

Fig. 15. A triangular FEM with an opening

10. CONCLUSION

In this paper, an ant colony system is developed for the formation of sparse null basis leading to sparse self-stress matrices, and correspondingly highly sparse flexibility matrices for triangular and rectangular plate bending finite element models.

The quality of the present method is compared to the *LU* factorization method and graph-theoretical method is shown by some examples. Since *LU* factorization is the primary step of the Turn-back and REDUC algorithms, naturally the present method is more efficient than these methods.

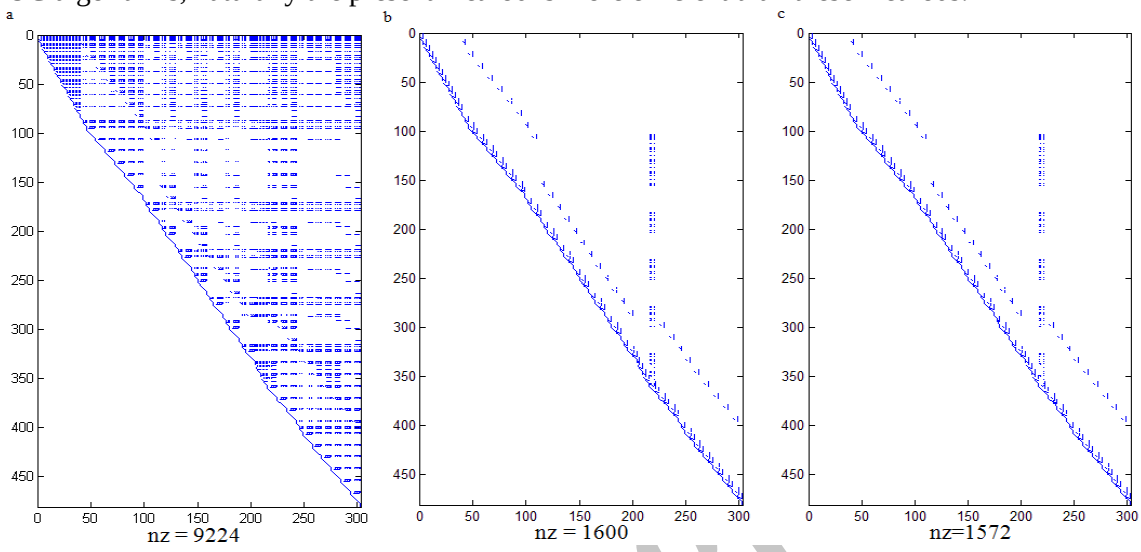

Fig. 16. Pattern of the null basis matrix resulted from, a) *LU* factorization method, b) graph-theoretical method and c) the present method

Table 2. Comparison of the sparsity, computational time and accuracy of the present algorithm versus *LU* factorization and graph-theoretical method

	Number of non-zero entries(nz)	$\frac{Time}{LU\ Time}$	$\| AB_1 \|_{fro}$
LU factorization	9224	1	3.12e-13
Graph-theoretical	1600	0.7534	8.52e-14
Present algorithm (hybrid method)	1572	10.3421	3.59e-14

The time for the formation of the null basis matrices is, in general, higher compared to the graph theoretical methods, however since the meta-heuristics are rapidly progressing and becoming more and more efficient, one hopes such algorithms will become superior in the very near future.

Acknowledgement: The first author is grateful to the Iran National Science Foundation for the support.

REFERENCES

1. Henderson, J. C de C. (1960). Topological aspects of structural analysis. *Aircraft Engineering*, Vol. 32, pp. 137–141.
2. Maunder, E. A. W. (1971). Topological and linear analysis of skeletal structures. Ph.D. Thesis, London University, Imperial College, UK.
3. Henderson, J. C de C. & Maunder, E.A.W. (1969). A problem in applied topology. *Journal of the Institute of Mathematics and its Applications*, Vol. 5, pp. 254-269.
4. Kaveh, A. (1974). Application of topology and matroid theory to the analysis of structures. Ph.D. Thesis, London University, Imperial College, UK.

5. Kaveh, A. (1992). Planar drawing of space structures. *Iranian Journal of Science ad Technology*, Vol. 14, pp. 23-32.
6. Kaveh, A. (1976). Improved cycle bases for the flexibility analysis of structures. *Computer Methods in Applied Mechanics and Engineering*, Vol. 9, pp. 267-272.
7. Kaveh, A. (1979). A combinatorial optimization problem; optimal generalized cycle bases, *Computer Methods in Applied Mechanics and Engineering*, Vol. 20, pp. 39-52.
8. Cassell, A. C. (1976). An alternative method for finite element analysis; a combinatorial approach to the flexibility method, *Proceedings of the Royal Society of London*, Vol. A352, pp. 73–89.
9. Kaveh, A., Koohestani, K. & Taghizadeh, N. (2007). Efficient finite element analysis by graph-theoretical force method. *Finite Elements in Analysis and Design*, Nos. 6-7, Vol. 43, pp. 543-554.
10. Kaveh, A. & Naseri, N. (2010). A new four-node quadrilateral plate bending element for highly sparse and banded flexibility matrices. *Acta Mechanica*, Nos. 3-4, Vol. 209, pp. 295-309.
11. Denke, P. H. (1962). A general digital computer analysis of statically indeterminate structures NASA-TD-D-1666.
12. Robinson, J. (1973). *Integrated theory of finite element methods*. Wiley, New York.
13. Topçu, A. (1979). A contribution to the systematic analysis of finite element structures using the force method (in German). Doctoral Dissertation, Essen University, Germany.
14. Kaneko, I., Lawo, M. & Thierauf, G. (1982). on computational procedures for the force methods. *International Journal for Numerical Methods in Engineering*, Vol. 18, pp. 1469–1495.
15. Soyer, E. & Topçu, A. (2001). Sparse self-stress matrices for the finite element force method. *International Journal for Numerical Methods in Engineering*, Vol. 50, pp. 2175–2194.
16. Kaveh, A. & Rahami, H. (2004). Algebraic Graph theory for suboptimal cycle bases of graphs for an efficient force method. *Iranian Journal of Science and Technology, Transaction B: Technology*, Vol. 28, pp. 529-536.
17. Gilbert, J. R. & Heath, M. T. (1987). Computing a sparse basis for the null space. *SIAM Journal of Algebraic Discrete Methods*, Vol. 8, pp. 446–459.
18. Coleman, T. F. & Pothen, A. (1986). The null space problem I; Complexity. *SIAM Journal of Algebraic Discrete Methods*, No. 4, Vol. 7, pp. 527–537.
19. Coleman, T. F. & Pothen, A. (1987). The null space problem II; Algorithms, *SIAM Journal of Algebraic Discrete Methods*, No. 4, Vol. 8, pp. 544–561.
20. Pothen, A. (1989). Sparse null basis computation in structural optimization. *Numerische Mathematik*, Vol. 55, pp. 501–519.
21. Heath, M. T., Plemmons, R. & Ward, J. (9184). Sparse orthogonal schemes for structural optimization using the force method. *SIAM Journal on Scientific and Statistical Computing*, Vol. 5, pp.514–532.
22. Patnaik, S. N. (1973). An integrated force method for discrete analysis. *International Journal for Numerical Methods in Engineering*, No. 2, Vol. 6, 237–251.
23. Patnaik, S. N., Berke, L. & Gallagher, R. H. (1991). Integrated force method versus displacement method for finite element analysis. *Computers and Structures*, No. 4, Vol. 38, pp. 377–407.
24. Kaveh, A. & Daei, M. (2009). Efficient force method for the analysis of finite element models comprising of triangular elements using ant colony optimization. *Finite Elements in Analysis and Design*, No. 10, Vol. 45, pp. 710-720.
25. Kaveh, A. & Daei, M. (2009). Formation of statical basis for efficient force method by ant colony optimization. *Asian Journal of Civil Engineering*, No. 1, Vol. 10, pp. 79-96.
26. Kaveh, A. & Malakouti Rad, S. (2010). Hybrid genetic algorithm and particle swarm optimization for the force method-based simultaneous analysis and design. *Iranian Journal of Science and Technology, Transaction B: Technology*, Vol. 34, pp. 15-34.

27. Kaveh, A. (2004). *Structural mechanics: Graph and matrix methods*. 3rd edition, Research Studies Press (Wiley), Somerset, UK.
28. Przemieniecki, J. S. (1968). *Theory of matrix structural analysis*. McGraw-Hill, New York.
29. Dorigo, M. & Gambardella, L.M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. Evol. Comput. No. 1, Vol. 1, pp. 53–66.
30. Kaveh, A. & Masoudi, M. S. (2012). Cost optimization of a composite floor system using ant colony system. *Iranian Journal of Science and Technology, Transactions of Civil Engineering, Vol. 36 No. C2*, pp. 139-148.
31. Kaveh, A. & Koohestani, K. (2007). An efficient graph theoretical method for plate bending finite element analysis via force method. *Engineering Computations*, No. 7, Vol. 24, pp. 679-698.
32. Kaveh, A. & Massoudi, M. S. (2011). Efficient finite element analysis by graph-theoretical force method; rectangular and triangular plate bending elements. *Scientia Iranica*, No. 18, Vol. 5, pp. 1045–1053.