ASIAN JOURNAL OF CIVIL ENGINEERING (BUILDING AND HOUSING) VOL. 10, NO. 1 (2009)
PAGES 113-130

FORMATION OF STATICAL BASIS FOR EFFICIENT FORCE METHOD BY ANT COLONY OPTIMIZATION

A. Kaveh* [a,b] and M. Daei[a]

[a]*Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran*

[b]*Institute for Mechanics of Materials and Structures, Vienna University of Technology, Karlsplatz 13, A-1040 Wien, Austria*

Abstract

An efficient algorithm is presented for the formation of statical basis, corresponding to highly sparse flexibility matrices for structures. This is achieved by applying a modified ant colony optimization algorithm for the formation of localized self-equilibrating systems. The efficiency of the present algorithm is illustrated through simple truss examples.

Keywords: Statical basis; self-equilibrating stress systems (S.E.Ss); self-stress matrix; flexibility matrix; sparsity; ant colony system

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.

Four different approaches are adopted for the force method of structural analysis, which are classified as:

1. Topological force methods,
2. Algebraic force methods,
3. Mixed algebraic-topological force methods,
4. Integrated force method.

Topological methods have been developed by Henderson [1] and Henderson and Maunder [2] for rigid-jointed skeletal structures. Development of general combinatorial approaches and methods suitable for computer programming are due to Kaveh [3-4]. Algebraic methods have been developed by Denke [5], Robinson [6], Topçu [7], Kaneko et al. [8], and mixed algebraic-topological methods have been used by Gilbert et al. [9],

*E-mails address of the corresponding author: alikaveh@iust.ac.ir (A. Kaveh)

Coleman and Pothen [10-11]. The integrated force method has been developed by Patnaik [12-13], in which the equilibrium equations and compatibility conditions are satisfied simultaneously in terms of the force variables.

The force method of structural analysis requires the formation of a maximal set of independent self-equilibrating stress systems (*S.E.Ss*), known as a *statical basis* [14-15]. The elements of this basis form the columns of an $m \times \gamma(S)$ matrix, B_1, known as *S.E.Ss-member incidence matrix* referred to as *self-stress matrix* in this paper.

The main difficulty in the application of the force method is the formation of a self-stress matrix B_1 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.

In this paper, the ant colony system (*ACS*) which is a variation of the ant colony optimization (*ACO*) is applied to overcome this problem.

Heuristic algorithms, such as ant colony algorithms, have found many applications in optimization problems in the last decade. The power of these algorithms lies in their capability to converge to a good solution which does not depend on the specific search space to which they are applied. In this work, ant colony system is employed for the formation of sparse self-stress matrices. Though the method is quite general, however, for simplicity only pin-jointed truss structures are used as illustrative examples.

2. Formulation of the Force Method

Consider a structure S with M members and N nodes, which is $\gamma(S)$ times statically indeterminate. Select $\gamma(S)$ independent unknown forces as redundant. These unknown forces which can be selected from external reactions and or internal forces of the structure are denoted by

$$q = \{q_1, q_2, \ldots, q_{\gamma(S)}\} \tag{1}$$

In order to obtain a statically determinate structure, known as the *basic* (released or primary) structure of S, the constraints corresponding to redundant forces are removed. Consider the joints loads as

$$p = \{p_1, p_2, \ldots, p_n\} \tag{2}$$

where n is the number of the entries of the applied load vector.

The stress resultant distribution r due to the given load p for a linear analysis by the force method can be written as

$$r = B_0 p + B_1 q \tag{3}$$

where B_0 and B_1 are rectangular matrices each having m rows and n and $\gamma(S)$ columns, respectively. Here, m is the number of independent components for member forces.

Since the *overall flexibility matrix* G of a structure is equal to $B_1^t F_m B_1$, for the sparsity of G one can select a statical basis corresponding to *sparse self-stress matrix* B_1.

The main objective of this paper is to find statical bases to ensure the formation of well-conditioned flexibility matrices. For a discrete or discretized structure S, which is assumed to be statically indeterminate, let r denote the m-dimensional vector of generalized independent element forces and p be the n-vector of the nodal loads. The equilibrium conditions of the structure can then be expressed as $Ar = p$, where A is an $n \times m$ *equilibrium matrix*.

For the formation of a S.E.S. no applied load is required, thus the above equilibrium conditions can be expressed as

$$AB_1 = 0 \tag{4}$$

This equation shows the linear dependence of the columns of the matrix A, which is an $n \times m$ matrix with rank n. There are $m - n = t$ independent columns of B_1 which satisfy Eq. (4), thus forming a set of S.E.Ss.

There are many sets of S.E.Ss (statical bases) which satisfy Eq. (4). However, the main 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

$$B_1 = [S_1, S_2, ..., S_g, ..., S_t] \tag{5}$$

Suppose the first S.E.S. S_1 is found, then it can be normalized by the following equation

$$e_1^t S_1 = 1 \tag{6}$$

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

$$e_1^t S_2 = 0 \qquad 7(a)$$

$$e_2^t S_2 = 1 \qquad 7(b)$$

where $e_2 = \{0 1 0 ... 0 ... 0\}$ is an $m \times 1$ vector with 1 in the second position. Similar conditions can be written analogously for the remaining S.E.Ss.

3. The Mathematical Model for Optimization

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 S.E.Ss $S_1, S_2, \ldots, S_{\gamma(S)}$.

The first S.E.S. S_1 is arbitrary and therefore should be chosen as simple as possible. Then the second S.E.S. S_2 is selected satisfying the following conditions:

$$AS_2 = 0 \qquad 8(a)$$

$$e_1^t S_2 = 0 \qquad 8(b)$$

$$e_2^t S_2 = 1 \qquad 8(c)$$

The above relationships can be expressed as

$$\begin{bmatrix} & A & \\ I_2 & & 0 \end{bmatrix} S_2 = \begin{bmatrix} 0 \\ - \\ e_2 \end{bmatrix}. \qquad (9)$$

where $\bar{e}_2 = \{0 \quad 1\}$. S_2 should be find such that $Z = |S_2|$ becomes minimized. Here, $|S_2|$ denotes the cardinality of S_2 and it is equals to the number of non-zero entries of S_2.

This can now be generalized for the g^{th} S.E.S. S_g, after all the previous S.E.Ss up to g–1 are obtained. This can be stated as follows:

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

$$\begin{bmatrix} & A & \\ I_g & & 0 \end{bmatrix} S_g = \begin{bmatrix} 0 \\ - \\ e_g \end{bmatrix} \qquad (10)$$

where $\bar{e}_g = \{0\,0\,...\,0\,...1\}$ is a $g \times 1$ vector, with 1 in the g^{th} position.

Therefore, by performing a series of operations similar to Eq. (10), t S.E.Ss forming a statical basis $B_1 = [S_1, S_2, ..., S_g, ..., S_t]$ will be formed. It should be noted that for the last S.E.S., i.e. the t^{th} 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 $n \times m$ matrix A, and $m - n = t$ orthogonalising equations, thus forming $n + t = m$ equations (with the number of variables being equal to m), leads to a unique solution for the last S.E.S..

The numbering of the members in the structureis important of and can be recognized by considering the additional equations used in the normalizing and orthogonalising. Here, the ant colony system is applied to choose the ordering of the members such that the resulting S.E.Ss correspond to highly sparse B_1 matrices.

4. Optimization by Ant Colony Systems

A meta-heuristic algorithm based on the ants behavior was developed in early 1990s by Dorigo [16] and developed by other researchers, e.g. Dorigo and Gambardella [17]. This algorithm is called *ant colony optimization* (*ACO*) because it was motivated by ants social behavior. Ant colony system (*ACS*) is a variation of the *ACO* which has proven to behave more robustly and provide far better results for some optimization problems. In this work, *ACS* is chosen as a suitable tool for finding sparse statical bases. A brief description of *ACO* is given in the next section, when describing the process of adapting *ACS* to the problem of finding sparse statical basis.

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 do evaporate and loose 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 Figure 1. First random decision when moving towards the food is shown in Figure 1(a), the second group returning to the nest is illustrated in Figure 1(b) and the number of dashed lines in Figure 1(c) is approximately proportional to the amount of pheromone deposited by ants.

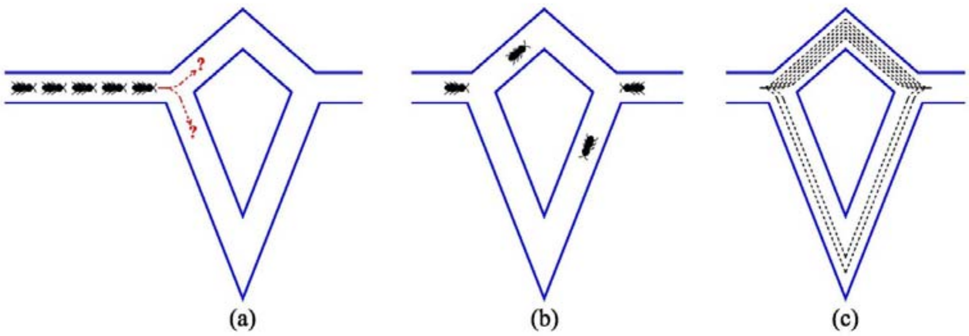


Figure 1. Ant technique to find optimum solution; (a) First random decision when moving towards the food (b) Second group returning to the nest (c) The amount of pheromone deposited

5. ACS for the Formation of Sparse Statical Basis

According to the proposed mathematical modeling, the pattern of numbering affects the results of selected S.E.Ss. This can be found out by considering the additional equations which are used in the process of normalization and orthogonalization. Taking $e_1^t S_1 = 1$, it is required to have a force equal to unity in the first element for S_1. This element is called the *Generator* of S_1. And according to $e_1^t S_2 = 0$ and $e_2^t S_2 = 1$, the first element in S_2, which is the generator of S_1, is zero, while the second element is one. This second element is known as the generator for the second column in the second S.E.S. S_2. Therefore, for the g^{th} S.E.S. 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 pin-jointed truss shown in Figure 2. This truss has 29 members and 12 nodes, therefore its degree of static indeterminacy (*DSI*) is $\gamma(S) = 29 - 2\times 12 + 3 = 8$.

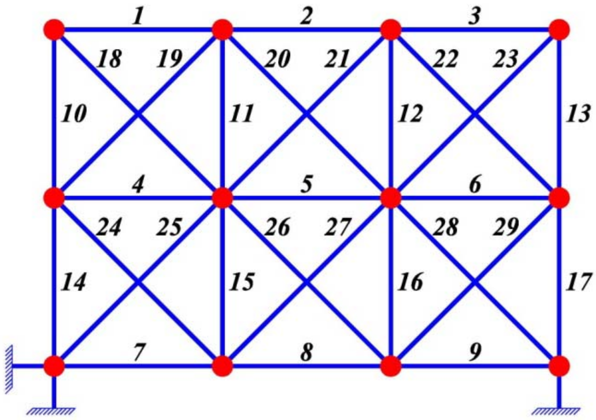


Figure 2. A pin-jointed truss with 29 members and 12 nodes

First, the generators are chosen based on members numbering as $(1 \to 2 \to 3 \to 4 \to 5 \to 6 \to 7 \to 8)$. For this selection, six S.E.Ss of the type presented in Figure. 3 are obtained and the two last S.E.Ss are illustrated in Figure. 4.

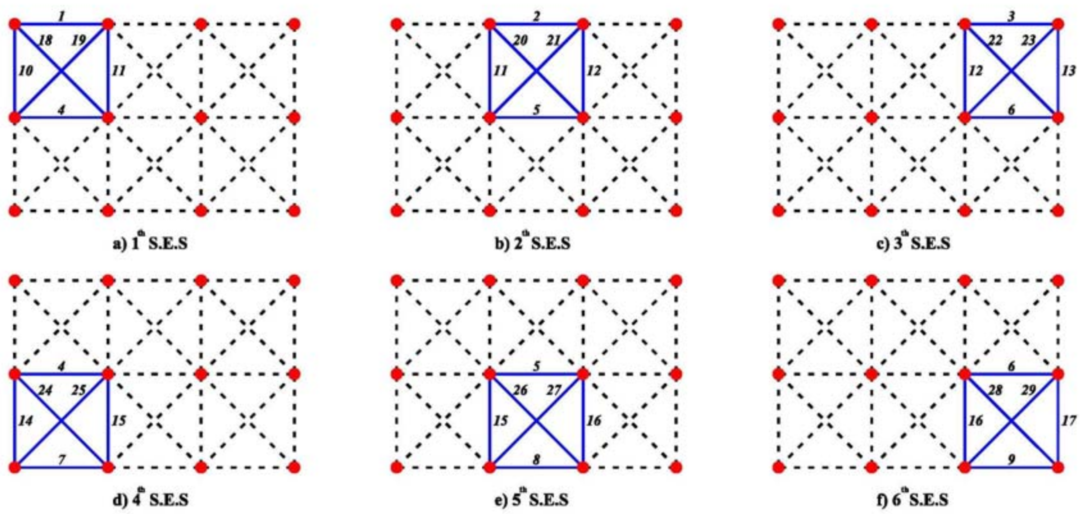


Figure 3. The first six S.E.Ss based on $(1 \to 2 \to 3 \to 4 \to 5 \to 6)$ as generators

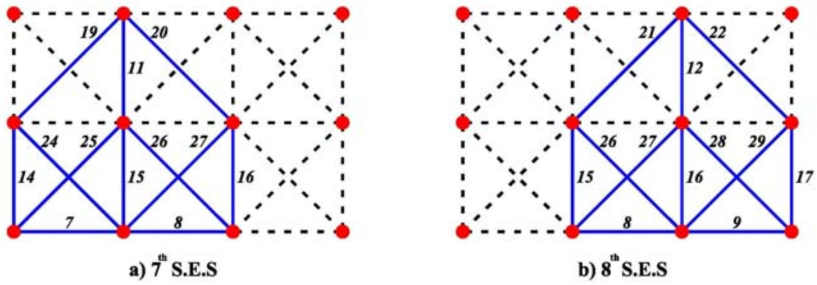


Figure 4. The last two S.E.Ss based on $(1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8)$ as generators

In order to show the effect of different set of members as generators, another sequence of members is chosen in the second attempt as $(1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 11 \rightarrow 12)$. The reason for doing this change is to get better results for the *7^{th}* and *8^{th}* S.E.Ss, while the first six S.E.Ss are identical to the previous set of member numbering (Figure. 3). The new results for the *7^{th}* and *8^{th}* S.E.Ss are shown in Figure. 5. Obviously, this system is sparser than the previous statical basis, resulting in a sparser flexibility matrix.

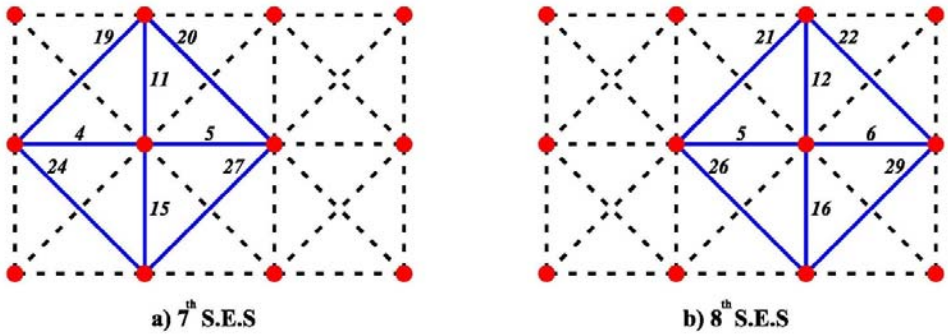


Figure 5. The new result for the last two S.E.Ss based on
$(1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 11 \rightarrow 12)$ as generators

Selection of a different set of generators as well as using a different order of members in this sequence can alter the results. In the previous set of generators, if the position of the member 2 is exchanged by the position of member 11, i.e. if the sequence $(1 \rightarrow 11 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 2 \rightarrow 12)$ is used, then the *7^{th}* and *8^{th}* S.E.Ss will be as those illustrated in Figure 6, which is unsatisfactory.

Therefore, each sequence of members as generators can be considered as a tour for an ant travel, and the best ant search for generators is the one which leads to sparser statical basis. The ant colony system which is applied to find the optimum solution is explained in the following section.

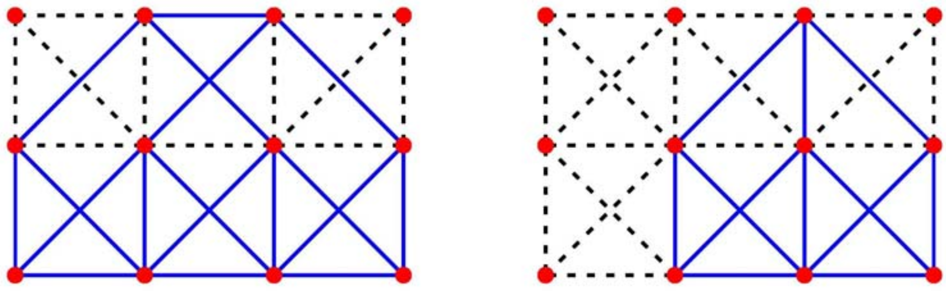


Figure 6. The new result for the last two S.E.Ss based on $(1 \to 11 \to 3 \to 4 \to 5 \to 6 \to 2 \to 21)$ as generators

6. ACS Algorithm for the Formation of Sparse Statical 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 statical basis, different sequence of members as generators is supposed as a tour for an ant to travel, therefore the cooperative ant agents search to find the best set of generators resulting in sparse statical basis.

Since both the member numbering, and its order in the generator sequence are important, thus the pheromone amount is specified by two parameters (τ_{ij}), where the *i* factor is as generator order in the set of generators and the *j* factor shows member number. As an example, τ_{25} shows the amount of pheromone for selection of member number 5 as the second generator in the generators set.

First *m* artificial ants are initially positioned on *m* members 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:

$$j = \begin{cases} \arg\max_{u \in L_k(r)} \left(\tau_{ru} \cdot \eta_{ru}^{\beta} \right) & \text{if } q \le q_0 \\ J & \text{Otherwise} \end{cases} \tag{11}$$

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.

$$P_{rs}^{k} = \begin{cases} \dfrac{\tau_{rs} \cdot \eta_{rs}^{\beta}}{\sum_{u \in L_k(r)} \tau_{ru} \cdot \eta_{ru}^{\beta}} & \text{if } s \in L_k(r) \\ 0 & \text{Otherwise} \end{cases} \tag{12}$$

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

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

$$\tau_{rs} \leftarrow (1-\xi)\cdot\tau_{rs} + \xi\cdot\tau_0 \tag{13}$$

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

Once all the ants complete their own tour, the pheromone is updated for all the members 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:

$$\tau_{rs} \leftarrow (1-\rho)\cdot\tau_{rs} + \rho\cdot\Delta\tau_{rs} \tag{14}$$

where

$$\Delta\tau_{rs} = \begin{cases} (D_{gb})^{-1} & if \quad (r,s)\in global \quad best \quad tour \\ 0 & Otherwise \end{cases} \tag{15}$$

D_{gb} is the sparsity coefficient of the globally best tour (number of nonzero elements in the selected statical basis) and $0<\rho<1$ is the pheromone decay parameter.

7. Finding a Lower Bound for Optimal Statical Basis Selection

Heuristic optimization algorithms like ant colony optimization, seek good feasible solutions to optimization problems in circumstances where the complexities of the problem or the limited time available for solution do not allow exact solution, although these algorithms often show the capability of leading to the best optimal solution.

In this section, an integer programming formulation is presented to evaluate the efficiency of the suboptimal solution which is obtained by the proposed *ACS* algorithm.

In spite of the previous algebraic relations for controlling the independence of the S.E.Ss, in this formulation, the independence control of selected S.E.Ss is performed using a graph theoretical approach as follow:

As before, the columns of the matrix B_1 is shown by S_i,

$$B_1 = [S_1, S_2, \ldots, S_g, \ldots, S_\gamma]. \tag{16}$$

For each column of this matrix, the non-zero entries form a subgraph which corresponds to one S.E.S.. The underlying subgraph of a S.E.S. is called a *γ-cycle* [12] and denoted by C_k.

The sequence of expansions is considered, where in each step one subgraph C_i is selected until the entire structure is formed as

$$C^{1} \to C^{2} \to C^{3} \to \ldots \to C^{\gamma(S)} \tag{17}$$

Where $C^{(k-1)} = \bigcup_{i=1}^{k-1} C_i$. In this sequence, in the k^{th} step a subgraph C_k is added to $C^{(k-1)}$ and it is called *admissible* if the following relationship holds:

$$\gamma(C^k) = \gamma(C^{(k-1)} \cup C_k) = \gamma(C^{(k-1)}) + 1 \tag{18}$$

This means that the degree of statically indeterminacy (γ) must only be increased by unity in each step of expansion.

In what follows, the different constants and variables which are applied in the proposed integer programming formulation are defined and then the mathematical model is presented.

Consider a structure S with M members and N nodes, and the corresponding graph model $G=(V,E)$ with N nodes (vertices) and M members (edges)

$$\begin{aligned} V &= \{1, 2, \ldots, v, \ldots, N\} \\ E &= \{1, 2, \ldots, e, \ldots, M\} \end{aligned} \tag{19}$$

For finding the k^{th} S.E.S., which is added to $C^{(k-1)}$ in the expansion process of Eq. (17), the two parameters as below are defined:

$$\forall i \in V : M_{SG}(i) = 1 \quad \text{if edge } i \text{ is in subgraph } C^{(k-1)}, \text{and } 0 \text{ otherwise} \qquad 20(a)$$

$$\forall i \in E : N_{SG}(i) = 1 \quad \text{if vertex } i \text{ is in subgraph } C^{(k-1)}, \text{and } 0 \text{ otherwise} \qquad 20(b)$$

The two parameters, M_{SG} and N_{SG}, determine the nodes and members that exist in the previous selected S.E.Ss and it means the nodes and members which belong to the subgraph $C^{(k-1)}$. The total number of members in this subgraph is shown by M_t and the total number of nodes by N_t.

The nodes and members which belong to the k^{th} S.E.S., C_k, are denoted by M_{id} and N_{id} parameters as below:

$$\forall i \in V : M_{id}(i) = 1 \quad \text{if edge } i \text{ is in } C_k, \text{and } 0 \text{ otherwise} \qquad 21(a)$$

$$\forall i \in E : N_{id}(i) = 1 \quad \text{if vertex } i \text{ is in } C_k, \text{and } 0 \text{ otherwise} \qquad 21(b)$$

The following is an integer linear programming formulation for finding the k^{th} S.E.S. with minimum length. This procedure is repeated $(\gamma-1)$ times for finding the 2^{th} up to γ^{th} S.E.Ss. The first S.E.S. is arbitrary and therefore should be selected as simple as possible. In this formulation, $Deg(i)$ shows the degree of vertex i and $Adj(i,j)$ denotes the j^{th} adjacent node of vertex i.

$$Min \ \sum_{i=1}^{M} M_{id}(i) \tag{22(a)}$$

$$Such\ that \ [A]\,[B_1]=0 \tag{22(b)}$$

$$-\mathrm{U}\times \mathrm{M_{id}(i)} \le \mathrm{S(i)} \le \mathrm{U}\times \mathrm{M_{id}(i)} \qquad \forall \mathrm{i}\in \mathrm{E} \tag{22(c)}$$

$$N_{id}(i) \le \sum_{j=1}^{Deg(i)} M_{id}(Adj(i,j)) \le Deg(i)\times N_{id}(i) \qquad \forall \mathrm{i}\in \mathrm{V} \tag{22(d)}$$

$$\mathrm{M_t} + \sum_{i=1}^{M} M_{id}(i)[1-M_{SG}(i)] - 2\,(N_t + \sum_{i=1}^{N} N_{id}(i)[1-N_{SG}(i)]\,) + 3 = k \tag{22(e)}$$

$$\sum_{i=1}^{M} M_{id}(i) - 2\sum_{i=1}^{N} N_{id}(i) + 3 = 1 \tag{22(f)}$$

The equilibrium condition is expressed in the constraint 22(b). By constraint 22(c), If $S(i)$, which shows the force in member *i,* is non-zero, then its corresponding parameter in the subgraph, $M_{id}(i)$ will be 1. In this formulation, U is an upper bound for the forces in structure, which is assumed as a big number. Constraint 22(d) ensures that if a node is in the selected S.E.S., then its corresponding parameter in the subgraph, $N_{id}(i)$ will be 1. The independence control is checked by constraint 22(e) and constraint 22(f) showing that each S.E.S should be have degree of statical indeterminacy equal to unity.

Based on this formulation, for a structure with M members and N nodes, there is $(2M+4N+2)$ constraints and $(2M+N)$ variables, therefore its computational time that is obtained with LINGO solver, is extremely high for large examples. However, the corresponding result is only used to evaluate the efficiency of the suboptimal solution which is obtained by the proposed *ACS* algorithm.

8. Numerical Results

In this section, four examples are presented to verify the high performance of the proposed *ACS* algorithm, and provide a measure of its efficiency. This algorithm is coded by MATLAB, and is run on a personal computer PentiumI 4 CPU 3.40GHz, 1.00 GB of RAM.

Example 1: Consider a truss as shown in Figure 7. This structure has 60 members and 25 nodes, therefore its degree of static indeterminacy is $\gamma(S) = 60 - 2\times 25 + 3 = 13$.

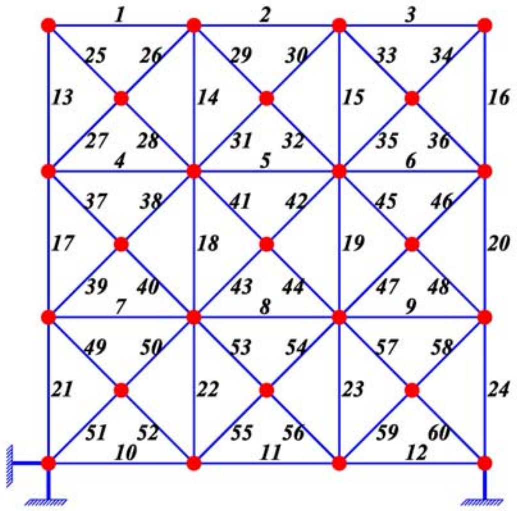
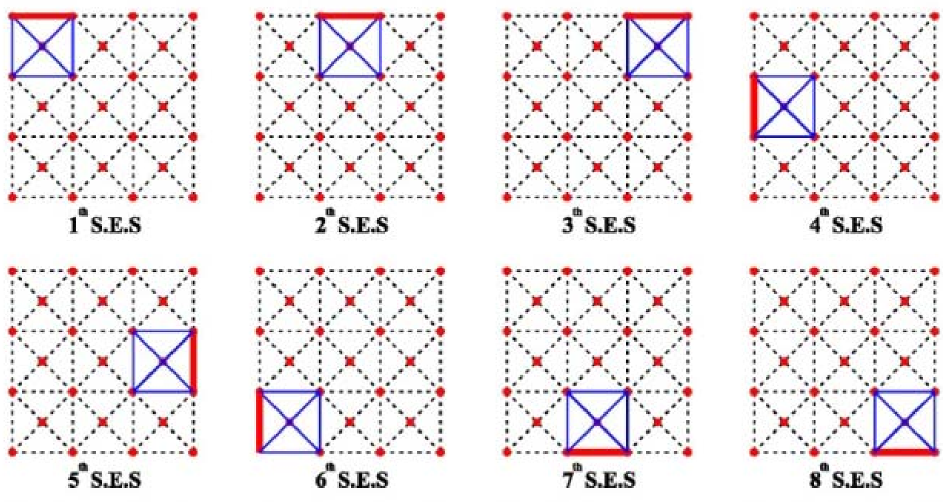


Figure 7. A truss with 60 members and 25 nodes

The set of $(1 \rightarrow 2 \rightarrow 3 \rightarrow 17 \rightarrow 20 \rightarrow 21 \rightarrow 11 \rightarrow 12 \rightarrow 14 \rightarrow 15 \rightarrow 7 \rightarrow 9 \rightarrow 5)$ is chosen by the best ant as generators. The resulted sparse S.E.Ss are shown in Figure. 8. In this figure, the generator of each system is shown in bold red line. The total number of non-zero entries is 136 and the elapsed run time is less than one second.



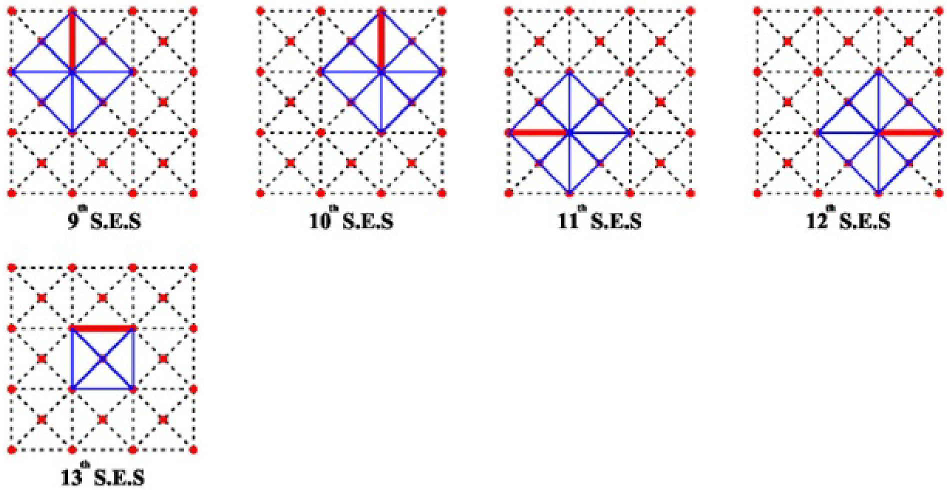


Figure 8. The sparse statical basis selected by *ACS*

The same S.E.Ss are selected by the proposed integer programming which is solved by LINGO. Since the concept of independence control is different, the order of above S.E.Ss is changed to 1, 2, 3, 4, 6, 9, 10, 5, 13, 11, 7, 12, 8.

Example 2: Consider a truss with a cut-out at the middle as shown in Figure 9. This structure has 40 members and 16 nodes, therefore the corresponding degree of static indeterminacy is $\gamma(S) = 40 - 2 \times 16 + 3 = 11$.

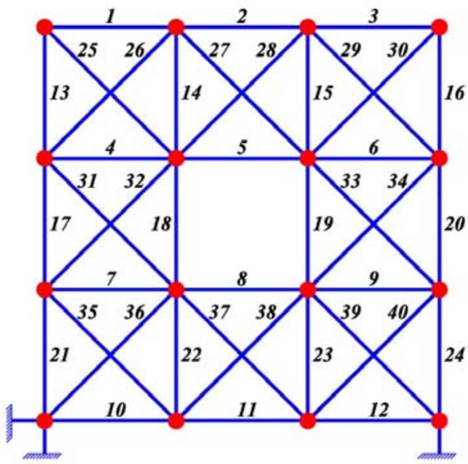


Figure 9. A truss with a cut-out

The set $13 \rightarrow 3 \rightarrow 10 \rightarrow 35 \rightarrow 17 \rightarrow 11 \rightarrow 12 \rightarrow 2 \rightarrow 40 \rightarrow 29 \rightarrow 19$ is chosen by the best ant as generators. The resulted sparse SESs are shown in Figure. 10. In this figure, the generator of each system is shown in bold line. The integer programming chooses the same statical basis, however, the order of S.E.Ss is as 1, 5, 3, 8, 6, 2, 7, 4, 10, 9, 11. The total number of non-zero entries is 90. The forces in the selected SESs are shown in Figure. 11.

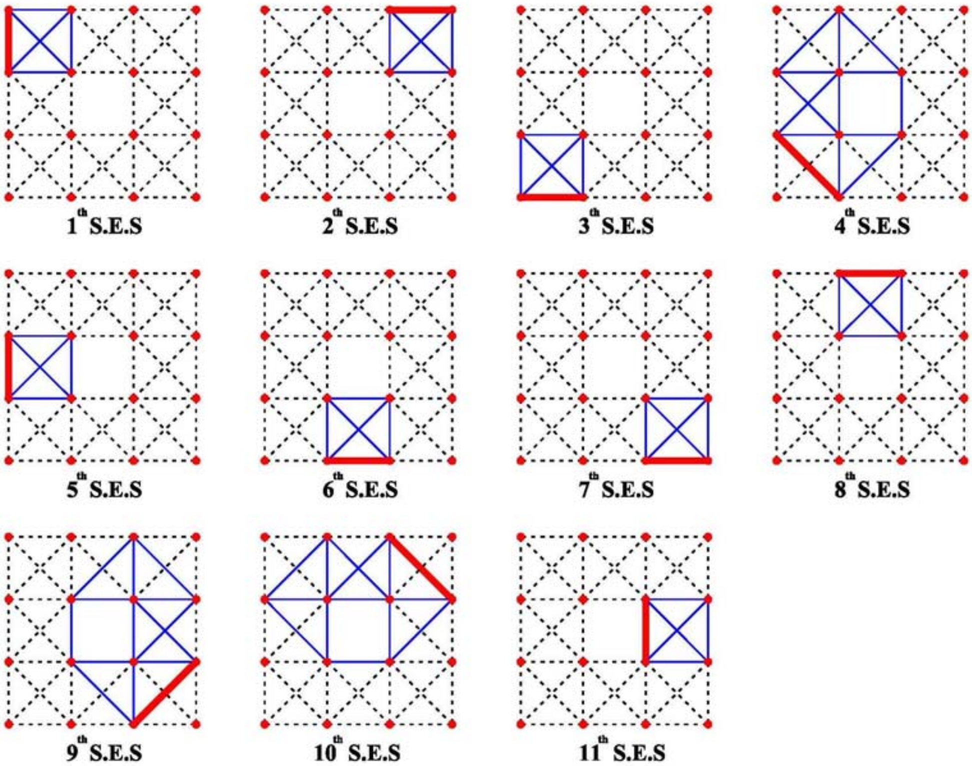


Figure 10. The sparse statical basis selected by *ACS*

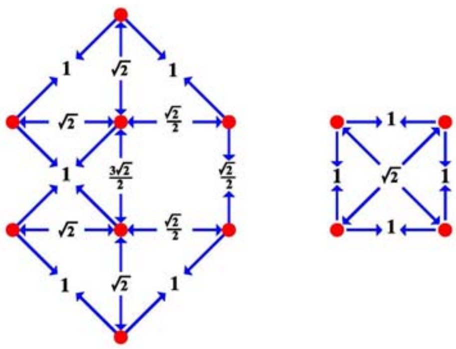


Figure 11. The forces in the selected S.E.Ss

Example 3: Consider a beam type truss with 10 bays as shown in Figure. 12. This structure has 92 members and 33 nodes, therefore its degree of static indeterminacy is equal to $\gamma(S) = 92 - 2 \times 33 + 3 = 29$.

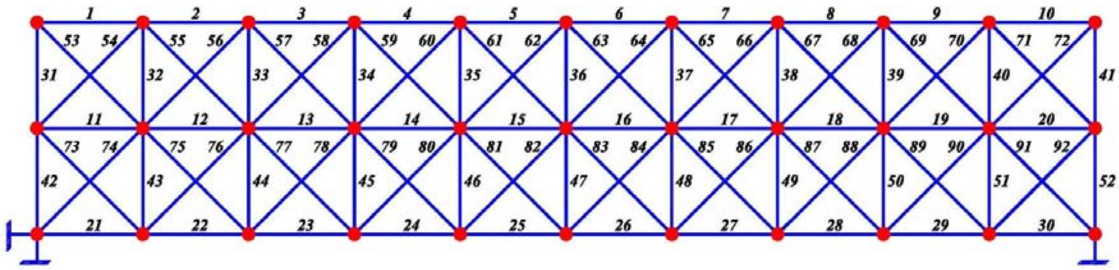


Figure 12. A beam type truss with 10 bays

The proposed algorithm selected 20 S.E.Ss with 6 entries and 9 S.E.Ss with 8 entries. These two types of S.E.Ss are shown in Figure. 13. For this sparse basis, the total number of non-zero entries is 192.

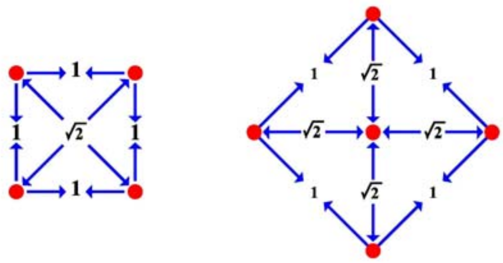


Figure 13. Forces in two types of sparse S.E.Ss for the beam type truss

Example 4: In order to present the elapsed run time of the algorithm, a unit block consisting of 4 nodes and 6 members is considered and sample structures are composed of equal number of such a unit in x and y directions, as shown in Figure. 14.

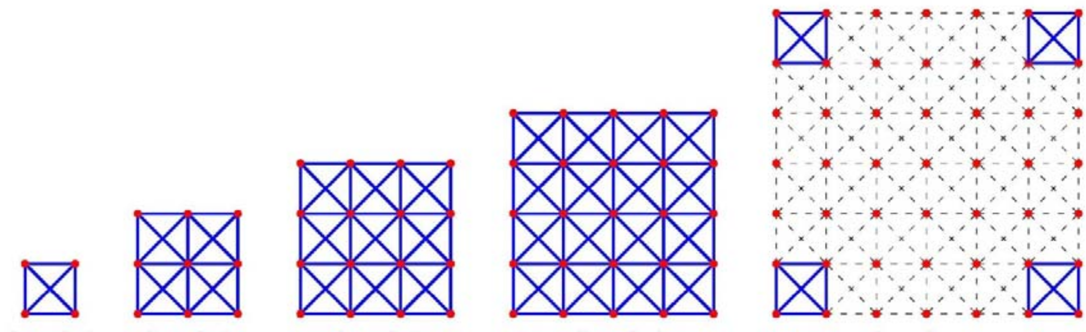


Figure 14. The geometry of sample structures

Furthermore, the following parameters values are considered in the proposed *ACS* algorithm: $\beta = 2$, $\xi = 0.1$, $\rho = 0.1$ and $q_0 = 0.5$. For this type of structure, the sparse statical basis consist of two types, the square shape with diagonal members and the diamond shape with diagonal members. These two types of S.E.Ss are shown in Figure 15.

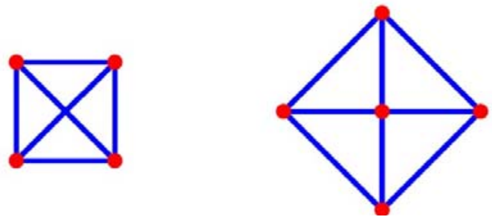


Figure 15. The underlying subtrusses of two types of localized S.E.Ss: square shape and diamond shape

Figure 16 shows the variation of elapsed run time with versus the number of structure nodes, for these samples.

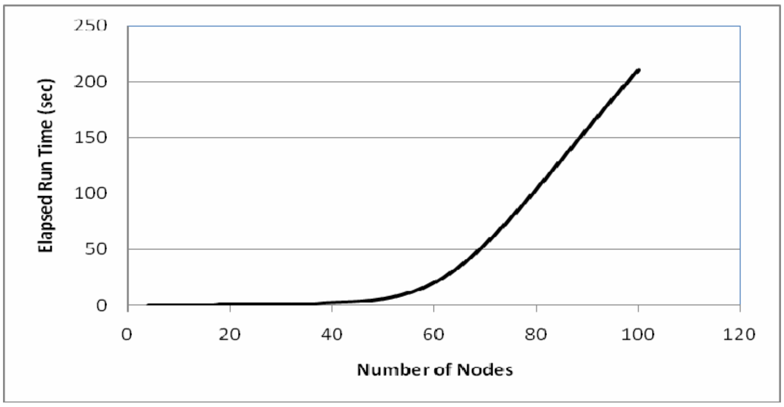


Figure 16. The variation of elapsed run time versus the number of nodes

For $n=10$ (the structure with 121 nodes and 420 members), a statical basis is selected by the best ant leading to the formation of a flexibility matrix G with 1733 nonzero entries. The pattern of G is shown in Figure 17.

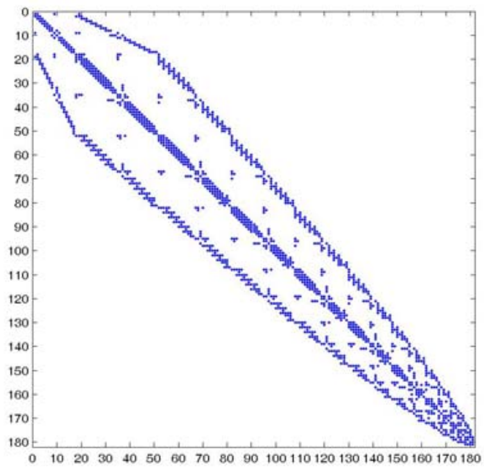


Figure 17. The pattern of the sparse flexibility matrix G,
Obtained by the proposed *ACS* algorithm

9. Conclusions

In this paper, an ant colony system is developed for the formation of sparse statical basis leading to sparse self-stress matrices, and correspondingly highly sparse flexibility matrices.

In the present method, numbering the members of the structure is not important and the selected sparse statical bases do not depend on numbering pattern of the structures. An integer programming formulation is also presented to evaluate the efficiency of the solutions obtained by the proposed ACS algorithm. Though the method of this paper is quite general, however, for simplicity only pin-jointed truss structures are used as illustrative examples.

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

References

1. Henderson JC de C. Topological aspects of structural analysis, *Aircraft Engineering*, **3**(1960)137-41.
2. Henderson JC de C, Maunder EAW. A problem in applied topology, *Journal of the Institute of Mathematics and its Applications*, No. 2, **5**(1969)254-69
3. Kaveh A. Application of topology and matroid theory to the analysis of structures, Ph.D. Thesis, London University, Imperial College, March 1974.
4. Kaveh A. Improved cycle bases for the flexibility analysis of structures, *Computer Methods in Applied Mechanics and Engineering*, **9**(1976)267-72.
5. Denke PH. A general digital computer analysis of statically indeterminate structures, NASA-TD-D-1666, 1962.
6. Robinson J. *Integrated Theory of Finite Element Methods*, Wiley, New York, 1973.
7. Topçu A. A contribution to the systematic analysis of finite element structures using the force method (in German), Doctoral Dissertation, Essen University, 1979.
8. Kaneko I, Lawo M, Thierauf G. On computational procedures for the force methods, *International Journal for Numerical Methods in Engineering*, **18**(1982)1469-95.
9. Gilbert JR, Heath MT. Computing a sparse basis for the null space, *SIAM Journal of Algebraic Discrete Methods*, **8**(1987)446–59.
10. Coleman TF, Pothen A. The null space problem I; Complexity, *SIAM Journal Algebraic Discrete Methods*, No. 4, **7**(1986)527–37.
11. Coleman TF, Pothen A. The null space problem II; Algorithms, *SIAM Journal Algebraic Discrete Methods*, No. 4, **8**(1987)544–61.
12. Patnaik SN. An integrated force method for discrete analysis, *International Journal for Numerical Methods in Engineering*, No. 2, **6**(1973)237-51.
13. Patnaik SN, Berke L, Gallagher, RH. Integrated force method versus displacement method for finite element analysis, *Computers and Structures*, No. 4, **38**(1991)377-407.
14. Kaveh A. A combinatorial optimization problem; Optimal generalized cycle bases, *Computer Methods in Applied Mechanics and Engineering*, **20**(1979)39–52.

15. Kaveh A. *Structural Mechanics: Graph and Matrix Methods*, third ed., Research Studies Press (Wiley), Somerset, UK, 2004.
16. Dorigo M. Optimization, Learning and Natural Algorithms (in Italian). PhD thesis, Dipartimento di Elettronica e Informazione, Politecnico di Milano, IT, 1992.
17. Dorigo M, Gambardella LM. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, No. 1, **1**(1997)53-66.