



BANDWIDTH OPTIMIZATION USING CBO AND ECBO

A. Kaveh* and Sh. Bijari

Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of
Science and Technology, Tehran, Iran

Received: 2 December 2014; **Accepted:** 10 February 2015

ABSTRACT

In this paper two recently developed meta-heuristic optimization methods, known as Colliding Bodies Optimization (CBO) and Enhanced Colliding Bodies Optimization (ECBO), are used for optimum nodal ordering to minimize bandwidth of sparse matrices. The CBO is a simple optimization algorithm which is inspired by a collision between two objects in one-dimension. Each agent is modeled as a body with a specified velocity and mass. A collision happens between pairs of bodies and the new positions of the colliding bodies are updated based on the collision laws. The enhanced colliding bodies optimization (ECBO) utilizes memory to save some best so-far-solution to improve the performance of the CBO without increasing the computational cost. This algorithm utilizes a mechanism to escape from local optima. The bandwidth of some graph matrices, which have equivalent pattern to structural matrices, is minimized using these approaches. Comparison of the obtained results with those of some existing methods shows the robustness of these two new meta-heuristic algorithms for bandwidth optimization.

Keywords: Bandwidth reduction; ordering; colliding bodies optimization; enhanced colliding bodies optimization; optimization.

1. INTRODUCTION

The solution of simultaneous equations is required by the analysis of many problems in structural engineering. Such non-singular systems of linear algebraic equations are in the form $Ax = b$ arises from finite element method. These types of equations usually involve a positive definite, symmetric, and sparse matrix coefficient A . For large structures a great deal of the computational cost and memory are dedicated to the solution of these equations. Hence some suitable specified patterns for the solutions of the corresponding equations have been provided, like banded form, profile form and partitioned form. These patterns are often attained by nodal ordering of the corresponding models.

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

In finite element model (FEM) analysis, for the case of one degree of freedom per node, performing nodal ordering is equivalent to reordering the equations. In a more general problem with m degree of freedom per node, there are m coupled equations produced for each node. In this case re-sequencing is usually performed on the nodal numbering of the graph models, to reduce the bandwidth, profile or wavefront, because the size of these problems are m fold smaller than those for m degree of freedom numbering. In this article, the mathematical model of a FEM is considered as an element clique graph, and nodal ordering is carried out to reduce the bandwidth of the corresponding matrices, Kaveh [1,2,3].

There is an important rule for nodal ordering in the solution of sparse systems. It can be achieved by permuting the rows and columns of a matrix by proper renumbering of the nodes of the associated graph. One important subject in nodal ordering is bandwidth optimization. In fact, for sparse matrices the size can be measured by the bandwidth of such matrices. These problems have created significant interest during recent years because it has practical relevance for a considerable range of global optimization applications. Since the nature of the problem of nodal ordering is NP-Complete, many approximate algorithms and heuristics are proposed, examples of which can be found in Papademetriou [4], Cuthill and McKee [5], Kaveh [1], Gibbs et al. [6].

Meta-heuristics algorithms are recent generation of the optimization methods to solve complex problems. These techniques explore the feasible region based on both randomization and some specified rules through a group of search agents. The rules are usually inspired from Laws of natural phenomena, Kaveh [7].

As a newly developed type of meta-heuristic method, colliding bodies optimization (CBO) is introduced and applied to structural problems by Kaveh and Mahdavi [8,9]. The CBO is multi-agent technique inspired by a collision between two objects in one-dimension. Each agent is modeled as a body with a specified velocity and mass. A collision happens between pairs of bodies and the new positions of the colliding bodies are updated based on the collision laws. The enhanced colliding bodies optimization is introduced by the Kaveh and Ilchi Ghazaan [10] and it utilizes memory to save some best so-far-solution to improve the CBO performance without increasing the computer execution time. This algorithm utilizes a mechanism to escape from local optima.

The rest of this paper is organized as follows: in Section 2 the bandwidth problem is presented, the CBO and ECBO algorithms are briefly demonstrated in Section 3. In order to show the performance of these techniques on bandwidth reduction, Section 4 contains six examples. The last section concludes the paper.

2. PROBLEM DEFINITION

Let $G(N,M)$ be a graph with members set $M(|M|=m)$ and nodes set $N(|N|=n)$. A labeling As of G assigns the set of integers $\{1,2,3,...,n\}$ to the nodes of graph G . $As(i)$ is the label or the integer assigned to node i and each node has different label. The bandwidth of node i for this assignment, $bw(i)$, is the maximum difference of $As(i)$ and $As(j)$, where $As(j)$ is the label of nodes adjacent to node i or the number assigned to its adjacent nodes. That is

$$bw_{As}(i) = \max \{|As(i) - As(j)| : j \in N(i)\} \quad (1)$$

Where $N(i)$ is the set of adjacent nodes of node i . The bandwidth of the graph G with respect to the assignment, $As(i)$, is then

$$BW_{As}(G) = \max \{bw(i) : i \in G\} \quad (2)$$

The minimum value of BW_{As} over all possible assignments is the bandwidth of the graph:

$$BW(G) = \min \{BW_{As}(G) : \forall As(i)\} \quad (3)$$

Therefore, in the bandwidth minimization problem, one searches an assignment $As(i)$ that minimizes $BW(G)$. Such an assignment holds all the non-zero elements of the matrix onto a band, which is as close as possible to the main diagonal, Kaveh and Sharafi [11,12].

In this paper, the aim is to find an optimal assignment for nodal ordering of a graph to reduce the bandwidth of the associated matrix employing CBO and ECBO algorithms. The algorithms for bandwidth reduction are based on reordering or assigning new labels to the nodes of the graph to obtain an optimal bandwidth.

Each permutation of rows and columns of an $n*n$ sparse matrix associated to graph G , leads to a new reordering called the assigned set. If the initial ordering of the graph is $\{1,2,3,\dots,n\}$, each permutation of this list will be a new assigning set. The purpose is to find the optimal assigning list in order to achieve the best bandwidth.

3. CBO AND ECBO ALGORITHMS

This section contains the Colliding Bodies Optimization algorithm and its enhanced version. First, a brief description of standard CBO is provided, and then the ECBO is presented, Kaveh and Ilchi Ghazaan [10].

3.1 Colliding bodies optimization

The collision is a natural phenomenon and the Colliding Bodies Optimization algorithm was developed based on this occurrence by Kaveh and Mahdavi [8,9]. In this method, one object collides with other object and they move towards a minimum energy level. The CBO utilizes simple formulation, does not require any internal parameter, and does not use memory for saving the best solutions so far.

This technique is a population-based meta-heuristic algorithm. Each solution candidate X_i is considered as a colliding body (CB) and it has a specified mass defined as:

$$m_k = \frac{\frac{1}{fit(k)}}{\sum_{i=1}^n \frac{1}{fit(i)}} \quad k = 1, 2, \dots, n \quad (4)$$

where $fit(i)$ represents the objective function value of the i th CB and n is the number of colliding bodies.

In order to select pairs of objects for collision, CBs are sorted according to their mass in a decreasing order and they are divided into two equal groups: (i) stationary group and (ii) moving group. Moving objects collide to stationary objects to improve their positions and push stationary objects towards better positions (see Fig. 1).

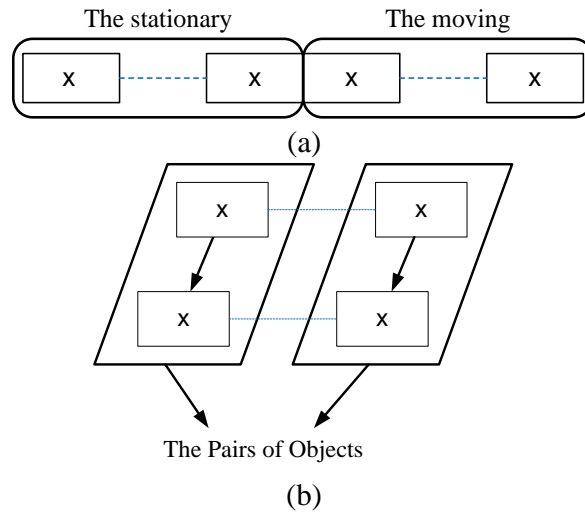


Figure 1. (a) The sorted CBs in an ascending order. (b) The pairs of objects for the collision

The velocity of the stationary bodies before collision is zero so

$$v_i = 0, \quad i = 1, 2, \dots, \frac{n}{2} \quad (5)$$

The velocity of each moving body before collision is

$$v_i = x_{i-\frac{n}{2}} - x_i, \quad i = \frac{n}{2} + 1, \frac{n}{2} + 2, \dots, n \quad (6)$$

The velocity of each stationary CB after the collision (v'_i) is specified by

$$v'_i = \frac{\left(m_{i+\frac{n}{2}} + \varepsilon m_{i+\frac{n}{2}} \right) v_{i+\frac{n}{2}}}{m_i + m_{i+\frac{n}{2}}}, \quad i = 1, \dots, \frac{n}{2} \quad (7)$$

The velocity of each moving CB after the collision (v'_i) is defined by

$$v'_i = \frac{\left(m_i - \varepsilon m_{i-\frac{n}{2}} \right) v_i}{m_i + m_{i-\frac{n}{2}}}, \quad i = \frac{n}{2} + 1, \frac{n}{2} + 2, \dots, n \quad (8)$$

Here ε is the coefficient of restitution (COR) that decreases linearly from unit to zero. Thus, it is stated as

$$\varepsilon = 1 - \frac{iter}{iter_{max}} \quad (9)$$

where $iter$ is the current iteration number and $iter_{max}$ is the total number of iteration for optimization process.

New positions of CBs are updated according to their velocities after the collision and the positions of stationary CBs. Therefore, the new position of each stationary CB is

$$x_i^{new} = x_i + rand \circ v'_i, \quad i = 1, \dots, \frac{n}{2} \quad (10)$$

where x_i^{new} , x_i and v'_i are the new position, previous position and the velocity after the collision of the i th CB, respectively. $rand$ is a random vector uniformly distributed in the range of $[-1, 1]$ and the sign “ \circ ” denotes an element-by-element multiplication. The new position of each moving CB is calculated by

$$x_i^{new} = x_{i-\frac{n}{2}} + rand \circ v'_i, \quad i = \frac{n}{2} + 1, \frac{n}{2} + 2, \dots, n \quad (11)$$

The process of optimization is terminated if the maximum number of analyses have been evaluated. For further details, the reader may refer to Kaveh and Mahdavi [8,9].

3.2 Enhanced colliding bodies optimization

A modified version of the CBO is Enhanced Colliding Bodies Optimization, which

improves the CBO to get more reliable solutions. The introduction of memory can increase the convergence speed of ECBO with respect to standard CBO. Furthermore, changing some components of colliding bodies will help ECBO to escape from local optima. The steps of ECBO are as follows:

Step 1: Initialization

The initial positions of all CBs are determined randomly in an m -dimensional search space.

$$x_i^0 = x_{\min} + \text{random} \circ (x_{\max} - x_{\min}), \quad i = 1, 2, \dots, n \quad (12)$$

where x_i^0 is the initial solution vector of the i th CB. Here, x_{\min} and x_{\max} are the bounds of design variables; *random* is a random vector which each component is in the interval $[0, 1]$; n is the number of CBs.

Step 2: Defining mass

The value of mass for each CB is evaluated according to Eq. (4).

Step 3: Saving

Considering a memory which saves some historically best CB vectors and their related mass and objective function values can make the algorithm performance better without increasing the computational cost, Kaveh and Ilchi [13,14]. Here a Colliding Memory (CM) is utilized to save a number of the best-so-far solutions. Therefore in this step, the solution vectors saved in CM are added to the population, and the same numbers of current worst CBs are deleted. Finally, CBs are sorted according to their masses in a decreasing order.

Step 4: Creating groups

CBs are divided into two equal groups: (i) stationary group and (ii) moving group. The pairs of CBs are defined according to Fig. 1.

Step 5: Criteria before the collision

The velocity of stationary bodies before collision is zero (Eq. (5)). Moving objects move toward stationary objects and their velocities before collision are calculated by Eq. (6).

Step 6: Criteria after the collision

The velocities of stationary and moving bodies are evaluated using Eqs. (7) and (8), respectively.

Step 7: Updating CBs

The new position of each CB is calculated by Eqs. (10) and (11).

Step 8: Escape from local optima

Meta-heuristic algorithms should have the ability to escape from the trap when agents get close to a local optimum. In ECBO, a parameter like *Pro* within $(0, 1)$ is introduced and it is specified whether a component of each CB must be changed or not. For each colliding body *Pro* is compared with rn_i ($i = 1, 2, \dots, n$) which is a random number uniformly distributed within $(0, 1)$. If $rn_i < Pro$, one dimension of the i th CB is selected randomly and its value is regenerated as follows:

$$x_{ij} = x_{j,\min} + \text{random} \cdot (x_{j,\max} - x_{j,\min}) \quad (13)$$

where x_{ij} is the j th variable of the i th CB. $x_{j,min}$ and $x_{j,max}$ respectively, are the lower and upper bounds of the j th variable. In order to protect the structures of CBs, only one dimension is changed. This mechanism provides opportunities for the CBs to move all over the search space thus providing better diversity.

Step 9: Terminating condition check

The optimization process is terminated after a fixed number of iterations. If this criterion is not satisfied go to Step 2 for a new round of iteration.

4. NUMERICAL EXAMPLES

In this section, six examples are considered. The first two examples are from Kaveh [2] and are used for examining the correctness of codes for both meta-heuristic algorithms and the four-step algorithm. The third example is the grid model of a fan with 1D beam elements. The fourth example is a FEM for shear wall, and an H-shaped finite element grid is presented in the fifth example. At the last example, a grid model of a shear wall with two irregular openings is considered. Two algorithms, namely the Colliding Bodies Optimization and Enhanced Colliding Bodies Optimization, are applied for bandwidth reduction of their matrices. The results are then compared to those of the four-step algorithm of Kaveh [15] and those of Kaveh and Sharafi [11,12] in Table 1.

Example 1: The graph model of a truss structure with 24 nodes is shown in Fig. 2. The performance of the CBO and ECBO algorithms are tested on this model and the results are provided in Table 1.

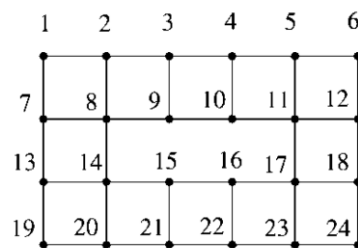


Figure 2. Graph model of a truss structure

Example 2: This is the model of a grid with uniform valency distribution, as shown in Fig. 3, having 28 nodes. The performance of the CBO and ECBO algorithms are tested on this model and the results are provided in Table 1.

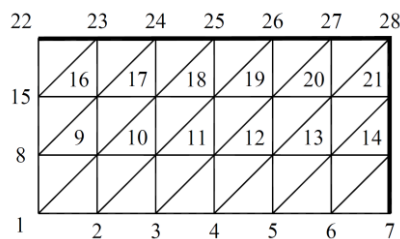


Figure 3. Graph model of a grid

Example 3: The graph model of a fan with 1575 nodes is considered, as shown in Fig. 4. Similar to the previous examples, the results of the algorithms are represented in Table 1, where the results can easily be compared.

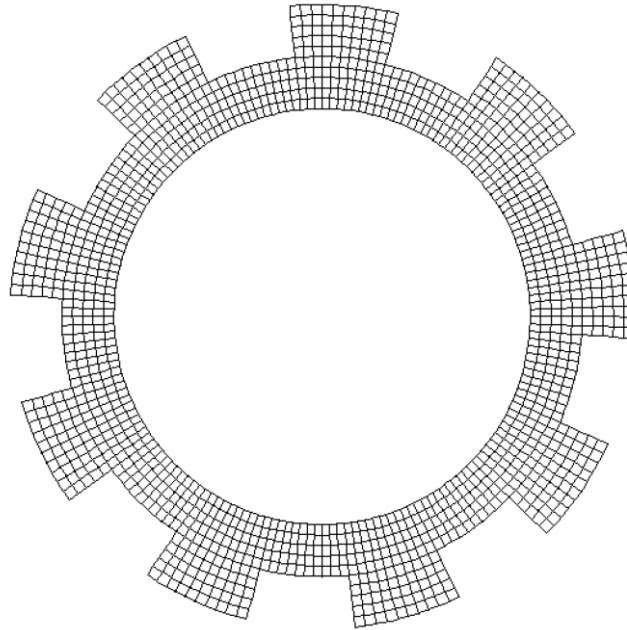


Figure 4. The graph model of a fan

Example 4: The FEM of a shear wall with 550 nodes is considered, as shown in Fig. 5. The performance of the CBO and its enhanced version is tested on this model, and the results are given in Tables 1. Quality of the results is provisioned in this table.

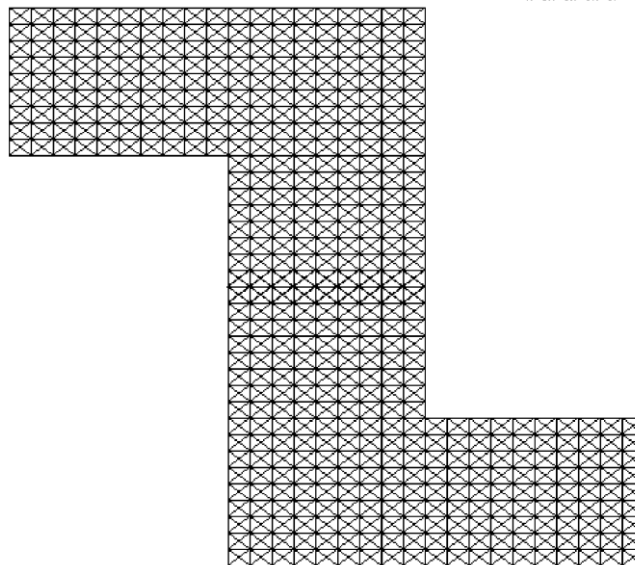


Figure 5. FEM of a shear wall

Example 5: An H-shape finite element grid with 4949 nodes is considered, as shown in Fig. 6. The element clique graph of this model includes 4949 nodes and 9688 beam elements (edges). The performance of the CBO and ECBO algorithms are examined on this model and the results are provided in Table 1.

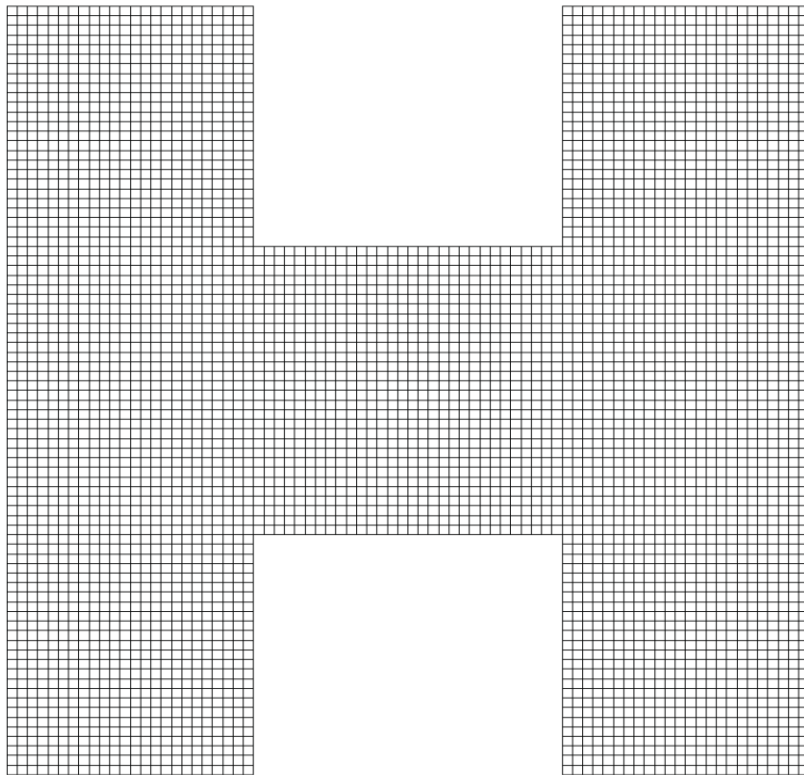


Figure 6. An H-shaped finite element grid

Example 6: A finite element grid model of a shear wall with two irregular openings is considered, as shown in Fig. 7. It has 235 nodes. The performance of the CBO and ECBO algorithms are tested on this model and the results are shown in Table 1.

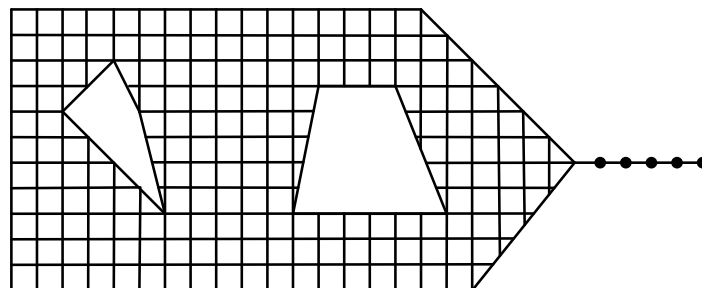


Figure 7. Finite element grid model of a shear wall

Table 1: Comparison of the results of different algorithms

	Four-step Algorithm of Kaveh [1,15]	CBO Algorithm	ECBO Algorithm	Results of Refs. [11,12]		
				4-step	ACO	CSS
Example 1	4	4	4	4	---	---
Example 2	4	4	4	---	---	---
Example 3	18	18	18	23	23	21
Example 4	28	28	28	29	29	---
Example 5	57	57	57	66	60	58
Example 6	13	13	13	---	---	---

5. CONCLUDING REMARKS

The main aim of this paper has been to show the performance and robustness of CBO and ECBO for bandwidth reduction of matrices as a discrete optimization problem. From Table 1, it can be observed that the obtained results from these two algorithms are quite satisfactory as compared to the well-known graph theoretical method, four-step algorithm. CBO and its enhanced version improve the bandwidth values previously attained by CSS and ACO algorithms, and these values are the best results so far.

REFERENCES

1. Kaveh A. Applications of topology and matroid theory to the analysis of structures, Ph.D. thesis, Imperial College of Science and Technology, London University, UK, 1974.
2. Kaveh A. *Structural Mechanics: Graph and Matrix Methods*, Research Studies Press, 3rd edition, Somerset, UK, 2004.
3. Kaveh A. *Optimal Structural Analysis*, John Wiley, 2nd edition, Chichester, UK, 2006.
4. Papademetriou CH. The NP-completeness of bandwidth minimization problem, *Computing Journal*, **16**(1976) 177-92.
5. Cuthill E, McKee J. Reducing the bandwidth of sparse symmetric matrices, *Proceedings of the 24th National Conference ACM*, Bradon System Press, NJ, 1969, pp. 157-72.
6. Gibbs NE, Poole WG, Stockmeyer PK. An algorithm for reducing the bandwidth and profile of a sparse matrix, *SIAM Journal of Numerical Analysis*, **12**(1976) 236-50.
7. Kaveh A. *Advances in Metaheuristic Algorithms for Optimal Design of Structures*, Springer International Publishing, Switzerland, 2014.
8. Kaveh A, Mahdavi VR. Colliding bodies optimization: A novel meta-heuristic method, *Computers and Structures*, **139**(2014) 18-27.
9. Kaveh A, Mahdavi VR. Colliding bodies optimization method for optimum design of truss structures with continuous variables, *Advances in Engineering Software*, **70**(2014) 1-12.
10. Kaveh A, Ilchi Ghazaan M. Enhanced colliding bodies optimization for design

- problems with continuous and discrete variables, *Advances in Engineering Software*, **77**(2014) 66-75.
11. Kaveh A, Sharafi P. Nodal ordering for bandwidth reduction using ant system algorithm, *Engineering Computations*, No. 3, **26**(2009) 313-23.
 12. Kaveh A, Sharafi P. Ordering for bandwidth and profile minimization problems via Charged System Search algorithm, *Iranian Journal of Science and Technology, Transactions in Civil Engineering*, No. C1, **36**(2012) 39-52.
 13. Kaveh A, Ilchi Ghazaan M Computer codes for colliding bodies optimization and its enhanced version, *International Journal of Optimization in Civil Engineering*, No. 3, **4**(2014)321-32.
 14. Kaveh A, Ilchi Ghazaan M. Enhanced colliding bodies algorithm for truss optimization with dynamic constraints, *Journal of Computing in Civil Engineering, ASCE*, 2014, [10.1061/\(ASCE\)CP.1943-5487.0000445](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000445), 04014104.
 15. Kaveh A. Ordering for bandwidth reduction, *Computers and Structures*, **24**(1986) 413-20.