Multiobjective optimal design of reinforced concrete frames using two metaheuristic algorithms

Mehdi Babaei* and Masoud Mollayi

Department of Civil Engineering, Faculty of Engineering, University of Zanjan, Zanjan, Iran
** Corresponding Author : mbabaei@znu.ac.ir*

Submitted ***:*** 18/03/2020
Revised ***:*** 21/01/2021
Accepted ***:*** 09/02/2021

ABSTRACT

Genetic algorithm (GA) and differential evolution (DE) are metaheuristic algorithms that have shown a favorable performance in the optimization of complex problems. In recent years, only GA has been widely used for single-objective optimal design of reinforced concrete (RC) structures; however, it has been applied for multiobjective optimization of steel structures. In this article, the total structural cost and the roof displacement are considered as objective functions for the optimal design of the RC frames. Using the weighted sum method (WSM) approach, the two-objective optimization problem is converted to a single-objective optimization problem. The size of the beams and columns are considered as design variables, and the design requirements of the ACI-318 are employed as constraints. Five numerical models are studied to test the efficiency of the GA and DE algorithms. Pareto front curves are obtained for the building models using both algorithms. The detailed results show the accuracy and convergence speed of the algorithms.

Keywords: Reinforced concrete frame; Weighted sum method; Multi-objective optimization; Differential evolution; Genetic algorithm; Optimum cost; Displacement.

INTRODUCTION

Due to the diversity of materials for reinforced concrete (RC) structures compared to steel structures, the optimization of RC structures is more complex than that of steel structures. Some variables in optimizing the RC frames include the compressive strength of concrete, the strength of steel bars, size of the cross-sections, the amount and percentage of reinforcements, and the other related parameters. The three parameters of the required concrete volume, the weight of the required reinforcing bars, and the formwork cost are important in calculating the cost of RC structures. It should be noted that labor costs must also be added to the above costs; however, labor costs can be included in the formwork costs. Therefore, the task of optimal design of the RC frame is more complicated.

In recent years, metaheuristic algorithms were developed to optimize problems with continuous or discrete design variables, and with complex, non-differentiable, or even implicit functions where the classic methods cannot be applied to. These algorithms were applied to the optimal design of many types of structures as well as steel and RC structures. Kaveh and Zarandi (2019) developed three metaheuristic algorithms for the optimal design of steel-concrete composite bridges. Farahmand-Tabar and Babaei (2021) applied a multi verse optimizer (MVO) for simultaneous size and shape optimization of trusses. Sarma and Adeli (1998) studied the cost optimization of concrete structures. Rajeev and Krishnamoorthy (1998) employed a genetic algorithm-based methodology for design optimization of the RC frames.

Chan and Wang (2006) conducted a study on nonlinear stiffness design optimization of tall RC structures under service loads. An integrated genetic algorithm with direct search was developed for the optimal design of the RC frames by Kwak and Kim (2009). Minimum cost design of RC beams using continuum type optimality criteria studied by Adamu et al. (1994). Kaveh and Sabzi (2011) conducted a comparative study of two metaheuristic algorithms for the optimal design of reinforced concrete frames. Akin and Saka (2015) developed a harmony search algorithm for the optimal design of reinforced concrete plane frames according to the ACI 318-05 provisions (2005). Optimal seismic design of RC shear walls-frame structures is studied by Kaveh and Zakian (2014).

Colliding bodies optimization (CBO) and big-bang big-crunch (BB-BC) algorithms were developed for the optimal design of the RC frames (Kaveh and Mahdavi, 2015; Kaveh and Sabzi, 2012). Boscardin et al. (2019) optimized RC building frames with an automated grouping of columns. Optimal design of RC 3D structures considering frequency constraints was performed by Kaveh and Behnam (2013) using charged system search (CSS). Esfandiari et al. (2018) developed multicriterion decision-making (DM) and Particle Swarm Optimization (PSO) algorithm, called DMPSO, for optimum design of the 3D RC frames.

To solve an engineering optimization problem, a parametric study is the simplest method to find the optimal layout, while it needs a huge evaluation of many different designs, which are impossible for large-scale problems. Using parametric study method, practical optimal topologies for the RC frames have been obtained and reported in the literature for 5-story and 10-story buildings (Babaei, 2015). In another study, using this approach, the number and location of trusses for steel structures with core and outrigger-belt truss system optimized considering weight and roof deflection as the objective functions (Babaei, 2017).

A few studies have been carried out for multiobjective optimization of the RC frames using evolutionary algorithms, while many studies carried out for steel structures (Babaei and Sanaei, 2016). Kaveh (2017) applied the enhanced colliding bodies (ECB) algorithm, Camp and Huq (2013) employed the big-bang big-crunch (BB-BC) algorithm to optimize the cost and CO_2 emission of the RC frames. Babaei and Mollayi (2016) developed a none-dominated sorting genetic algorithm (NSGA-II) to the multiobjective optimal design of the RC frames, where the total weight and the roof displacement were the objective functions. Leyva et al. utilized NSGA-II to earthquake design of RC buildings (Leyva et al., 2018). Zou et al. (2007) carried out a study on multiobjective optimization of the RC frames for performance-based design.

Multiobjective optimization methods and algorithms are more complicated and time-consuming, and the efficiency of these algorithms decreases when applying to large-scale optimization problems. There are, however, other methods that convert multiobjective optimization problem (MOP) into single-objective optimization problem (SOP) and easily get the optimum solutions set and trade-offs by iterating the main algorithm which is developed for the SOP's (Deb, 2001; D'Errico, 2015; Marler and Arora, 2004; Sanaei and Babaei, 2012; Sanaei and Babaei, 2011). Among them, the weighted sum method (WSM) is the most famous one. This article aims to determine the trade-off between the cost and the roof displacement of the RC frame structures using the WSM, and to obtain the optimal Pareto front through the feasible objective space using GA and DE.

In the next section, the optimization procedure for the RC frames is developed by defining design variables, objective functions, and design constraints. Section 3 describes the methods of optimization including GA and DE, briefly. Numerical examples including five RC frames are optimized in section 4 using the algorithms and the results are compared together and discussed. Finally, section 5 concludes the paper.

OPTIMIATION PROCESS

Objectıve Functıons

In this paper, the cross-sections of the beams and columns are considered as design variables and then two conflicting objective functions are presented in terms of the design variables. To avoid complexity, discrete cross-sections with acceptable and practical reinforcement ratios are defined according to the code requirements, as addressed in section 2.4. The first objective function is the total structural cost and the second one is the maximum lateral displacement. For this purpose, the total cost is calculated based on three parameters: the volume of required concrete, the amount of the reinforcement bars, and the formwork and labor costs. Therefore, to minimize the cost, the first objective function can be presented as follows:

$$F_c = f(P_s, P_c, P_f) \tag{1}$$

where F_c is the cost function to be minimized, P_s is the cost of steel bars, P_c is the cost of concrete, and P_f is the formwork and labor cost.

The second objective function is the maximum lateral displacement of the roof, shown as δ_{roof}. Therefore, the two-objective optimization problem could be defined as follows:

$$Minimize \ \ F_1 = \sum_{element} C_s L A_s + C_c Lbh + C_f Lp \tag{2}$$

$$Minimize \ \ F_2 = \delta_{roof} \tag{3}$$

where C_c is the cost of concrete per unit volume, C_s the cost of steel bars per unit weight, C_f is the formwork cost per unit area, L is the length of the structural elements, and p is the surface area of the structural elements.

COMPOSITE OBJECTIVE FUNCTION

As stated earlier, genetic algorithm is a single-objective optimizer algorithm and, therefore, an approach is required to solve multiobjective optimization problems. One of the most common methods to solve MOPs is the weighted sum method (WSM), in which objective functions are combined to form a single function using a simple linear combination (Deb, 2001; Marler and Arora, 2004). In this method, each of the objective functions is assigned a weight between 0 and 1, so that the sum of the weights is equal to 1. To find out the Pareto front within the objective space, different weights are applied, and the related optimal solutions are obtained and compared. The composite objective function could be shown as follows:

$$F = \alpha . F_1 + (1 - \alpha) . F_2 \tag{4}$$

where α is the weight of the first objective function.

Since two objective functions of cost and displacement have different dimensions and units, therefore, they cannot be added to each other in the above equation. In other words, the composite objective function should be normalized to form a dimensionless single objective function to be minimized.

Therefore, the maximum and the minimum values of cost and displacement are initially calculated and both objective functions are normalized using the following equation:

$$F = \alpha(\frac{f_1 - f_{1\min}}{f_{1\max} - f_{1\min}}) + (1-\alpha)(\frac{f_2 - f_{2\min}}{f_{2\max} - f_{2\min}}) \tag{5}$$

where f_1 and f_2 are the normalized objective functions; f_{1min} is the minimum amount of the first objective function; f_{1max} is the maximum amount of the first objective function; f_{2min} is the minimum amount of the second objective function, and f_{2max} is the maximum amount of the second objective function. Note that f_{1min} and f_{2max} are calculated by assigning the weakest cross-sections for all of the structural members, while f_{1max} and f_{2min} are obtained by considering the strongest cross-sections for the elements. To find out the Pareto optimal solutions, different values for α should be selected, and the Pareto front is obtained by connecting them in the objective space.

PENALTY FUNCTION

The objective function should be optimized so that all provisions such as the constraints have to be satisfied. On the other hand, the provisions related to structural deformation must also be met. From the codes and practical point of view, columns of the lower floors should have larger or same dimensions as those of the upper floors. Moreover, the diameters and number of reinforcing bars in the upper floor elements should be equal or less than those in the lower floors. Other requirements related to the ACI provisions must also be satisfied.

One of the most famous methods to satisfy the constraints in optimization problems is the penalty approach. In this method, if a constraint is not satisfied by a solution, the objective functions (cost or displacement) will be subjected to penalties. Using this method, in addition to the simplification of the objective function, provisions are applied so that the structural cost or displacement will be increased and solutions having penalties will automatically have a small chance to be selected as optimal solutions within the next generations. Many methods have been proposed to introduce a penalty function; however, in this study, the penalty function is used as follows:

$$F_p = F(1+C_i)^2 \tag{6}$$

$$C_i = \alpha_i V_i \tag{7}$$

$$\Phi = \sum_{i=1}^{n} C_i \tag{8}$$

where F_p is the penalized objective function, C_i is the penalty function for each constraint, α_i is the penalty coefficient for each constraint, V_i is the penalty for each constraint, and Φ is the cumulative amount of the penalties.

Penalty coefficient values are obtained according to the problem statement and using a trial and error procedure. These coefficients are unique values for each structure.

CODE REQUIERMENTS

According to the ACI (2005), columns must be evaluated using the interaction diagram between axial loads and bending moments. If the position of the applied load combination is inside or on the interaction diagram, the element will remain safe; otherwise, a stronger cross-section should be used for the element. In this case, the element is penalized by the related constraint. There are different methods for calculating penalties.

In this paper, the distance of the points from the reference coordinate is used. For this purpose, the distance of one point from the origin is calculated and the allowable value of that point (i.e., the location of the connecting line between that point and the origin, with interaction diagram) is also obtained. Finally, the penalty will be calculated from the distance between the desired point and the allowed point on its respective distance. In other words, if the related point for internal forces is out of the interaction diagram then the objective function is penalized.

Similar code requirements apply to the beams. Since there are no axial loads on the beams, the interaction ratio is not necessary for beams. In this case, only provisions for bending moments are applied.

An additional provision that must be met according to the codes is to maintain the reinforcement ratios within the allowable range. In other words, the lower and upper limits for reinforcement ratios of the columns should be between $0.01 \le \rho \le 0.08$. This amount is limited to $0.01 \le \rho \le 0.06$ in regions with higher seismic risk and for frames with moderate or high ductility levels. Given that the amount of reinforcement is doubled at the overlaps through the column, therefore, half of the above upper bound should be considered as the maximum allowable amount for reinforcement ratios in the columns.

According to the ACI, the longitudinal reinforcement should also have a distance from the margins which would be considered as the cover for the beams and columns. These values are calculated according to the reinforcement diameter. The number of reinforcements available in the sections is calculated according to this distance and the distance between reinforcements.

Based on the above code requirements, in total 252 sections are considered for beams with a different layout for the reinforcing bars, where the cross-sectional sizes are as 300×300, 300×400, 300×500, 350×350, 350×450, 350×550, 400×400, 400×500, 400×600, 450×450, 450×550, 450×650, 500×500, 500×600, 500×700, 550×550, 550×650, 550×700, 600×600, 600×650, and 600×700. For columns, 202 sections are assumed with different layouts of reinforcing bars as 300×300, 350×350, 400×400, 450×450, 500×500, 550×550, 600×600, 650×650, and 700×700.

The other restrictions, that must be addressed based on the codes, are the coefficients related to crack due to bending or axial forces, the code proposes reduction coefficients for the moment of inertia values of beams and columns. These coefficients are 0.35 and 0.7 for beams and columns, respectively.

OPTIMIZATION METHODS

Genetıc Algorıthm

Developed by Holland (1975), Genetic Algorithm (GA) is one of the metaheuristic algorithms, which has been welcomed by many researchers over the past decades. This algorithm is used for solving optimization problems in many areas as well as engineering problems. GA has been developed to solve SOPs; however, in this article, using WSM it is indirectly utilized to obtain solutions for MOPs. In this section a brief description for GA is addressed; the detailed structure of the algorithm is available through the literature.

GA starts with initialization to generate a uniformly distributed random population. The algorithm, then, creates new generations during iterations using *selection*, *crossover*, and *mutation* operators. In the selection stage, parents are randomly selected to create the next generation based on their fitness evaluations. The crossover operator creates children (solutions) using the selected parents and in the mutation stage, some of these children are reformed. The new generation is now created and the last three operators should be repeated. The number of parents chosen for crossover and mutation operations depends on the problem type and the size of the population itself.

It is proven in the literature that during iterations the algorithm finds near-optimal solutions. Famous selection methods include roulette wheel, tournament selection, and random selection. By trial and error, the best method can be obtained according to the type of problem for choosing the most efficient ones. In this article, the evaluation showed that the tournament method had the best performance. Considering the design requirements, in this paper, discrete design variables are considered for the beams and columns. Figure 1 displays a pseudocode of GA, which has been developed to solve frame optimization problems.

```
Parameter Definition
    Set initial parameters: MaxIt, Npop, Pc,Pm, gamma, mu
Initialization
    Initiate random population vector
        $X_i^0 = (x_{i,1}^{(0)}, x_{i,2}^{(0)}, \ldots, x_{i,npop}^{(0)}),\ (i = 1\ to\ npop)$
    Evaluate the fitness function of each target
For it=1 to MaxIt  do
    Crossover
        Choose pc*npop members of population to create offspring set.
            alpha=unifrnd(-gamma,1+gamma,size(x1));
            y1=alpha.*x1+(1-alpha).*x2;
            y2=alpha.*x2+(1-alpha).*x1;
             y1(1,1:nVarB)=max(y1(1,1:nVarB),1);
             y1(1,1:nVarB)=min(y1(1,1:nVarB),VarMaxB);
             y1(2,1:nVarC)=max(y1(2,1:nVarC),1);
             y1(2,1:nVarC)=min(y1(2,1:nVarC),VarMaxC);
            y1=round(y1);
            y2(1,1:nVarB)=max(y2(1,1:nVarB),1);
             y2(1,1:nVarB)=min(y2(1,1:nVarB),VarMaxB);
             y2(2,1:nVarC)=max(y2(2,1:nVarC),1);
             y2(2,1:nVarC)=min(y2(2,1:nVarC),VarMaxC);
             y2=round(y2);
    Mutation
        Choose pm*npop members of population to create Mutation set.
            nmub=ceil(mu*nVarB);
            nmuc=ceil(mu*nVarC);
            jb=randsample(nVarB,nmub);
            jc=randsample(nVarC,nmuc);
            sigmab=0.1*(VarMaxB-VarMinB);
            sigmac=0.1*(VarMaxC-VarMinC);
            y=x;
            y(1,jb)=(x(1,jb))'+sigmab*randn(size(jb));
            y(2,jc)=(x(2,jc))'+sigmac*randn(size(jc));
            y(1,1:nVarB)=max(y(1,1:nVarB),VarMinB);
            y(2,1:nVarC)=max(y(2,1:nVarC),VarMinC);
            y(1,1:nVarB)=min(y(1,1:nVarB),VarMaxB);
            y(2,1:nVarC)=min(y(2,1:nVarC),VarMaxC);
            y=round(y);
    Selection
        Create new set including npop, popc(crossover) and popm (mutation)
        Evaluate all members in the new set
        Sort the members
        Choose the best members of new set size of npop
    Set it=it+1
End for
```

Figure 1. The pseudocode for GA.

DIFFERENTIAL EVOLUTION

Differential Evolution (DE), for the first time, was suggested by Storn and Price (Storn and Price, 1995; Storn and Price, 1997) as a simple approach for global optimization of problems with continuous variables. During the past two decades, DE algorithms have been developed more and they have emerged as one of the most competitive evolutionary algorithms. DE and its variants have been applied to solve optimal design problems in different fields successfully (Ho-Huu et al., 2015; Babaei and Mollayi, 2020). Similar to GA, differential evolution is a population-based algorithm, and unlike GA, however, it uses real number decision variables; therefore, no encoding is needed to convert strings from real to binary.

```
Parameter Definition
    Set initial parameters: MaxIt, Npop, F,Cr
Initialization
    Initiate random population vector
    $X_i^0 = (x_{i,1}^{(0)}, x_{i,2}^{(0)}, ..., x_{i,npop}^{(0)}),\ (i = 1\ to\ npop)$
    Evaluate the fitness function of each target
For it=1 to MaxIt  do
    Mutation
        Create mutant vectors, using the mutation strategies according to
        the equations.
             beta1=0.2;
        beta2=0.8;
        beta=(it/MaxIt)+beta1;
        if beta >beta2
            beta=beta2;
        end
        if unifrnd(0,1)>beta
            if unifrnd(0,1)<0.5
        U=x1+F.*(best-x1)+F.*(x2-x3);  % Current to best/1
    else
         U=x1+F.*(best-x1)+F.*(best-x2); % Current to best/2
    end
else
    if unifrnd(0,1)<0.5
        U=x4+F.*(x2-x3);                              % rand/1
    else
        U=x4+F.*(x2-x3)+F.*(x5-x6);                   % rand/2
    end
end
best= best member of populations.
x1, x2,x3,x4,x5,x6= random member of population.
```

Crossover

```
Cerate offspring vector. For each target vector u
Pcr=((MaxIt-it)/MaxIt)*Pcrmax;
    if Pcr<Pcrmin
        Pcr=Pcrmin;
    end

jb1=randi([1 nVarB]);
jc1=randi([1 nVarC]);
jb2=find(rand(1,nVarB)<Pcr);
jc2=find(rand(1,nVarC)<Pcr);
Jb=[jb1 jb2];
```

```
        Jc=[jc1 jc2];
        x(1,Jb)=U(1,Jb);
        x(2,Jc)=U(2,Jc)
        V=x;
    Evaluate the offspring to compare with parents.
    Selection
        Apply selection for each offspring vector, if mutant vector is better than
        the main member, mutant forward to new population, otherwise the main
        member sent to new population to create next generation.
    Set it=it+1
End for
```

Figure 2. The pseudocode for DE.

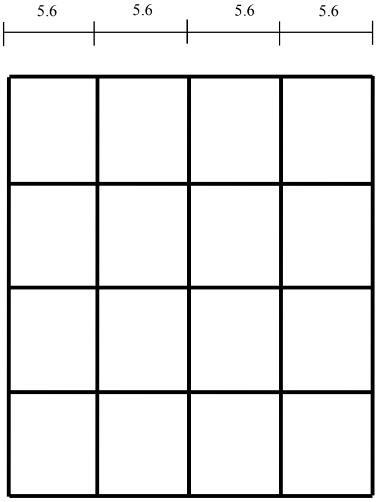
The first step in DE is random initialization. After population initialization, DE starts with three basic steps of mutation, crossover, and selection. The iterations stop when a termination criterion is satisfied, which is usually the number of iterations or the same solutions through the last populations. Details of the operations of DE could be found in the literature (Storn and Price, 1995; Storn and Price, 1997; Ho-Huu et al., 2015; Babaei and Mollayi, 2020). The pseudocode for the developed DE is given in Figure 2, which utilizes four different mutation strategies to increase the efficiency of the algorithm.

NUMERICAL EXAMPLES

In this study, five building models are optimized using the GA and DE algorithms; however, only three of them are reported in this manuscript, because of the limitations of the journal. The models are 2D RC frames extracted from 3D practical buildings. For this purpose, based on the research carried out by Babaei (2015), two different practical topologies with spans lengths of 5.6m and 7.5m are considered for the building plan to locate two and three vehicles between columns, respectively. The plan area of each building is about 529 m^2; and 5, 10, and 14-story structures are considered to evaluate the RC frames, as shown in Figures 3 and 4. The height of each floor is set 3.5m, for residential applications. The soil type under the foundation is of type (II); the building site is in Tehran, which is a city with very high seismicity risk. The dead and live loads of the floors are 5 kN/m^2 and 3 kN/m^2, while they are assumed as 5 kN/m^2 and 1.5 kN/m^2 for the roof, respectively. According to the ACI design code (2005), the following load combinations are applied:

$$U = 1.2D + 1.6L$$
$$U = 1.2D + 1.0L \pm 1.0E$$
$$U = 0.9D \pm 1.0E \qquad (9)$$

The concrete compressive strength is set $f_c^{'} = 28\,MPa$ and the yielding stress for reinforcing bars is considered as $f_y = 420\,MPa$.



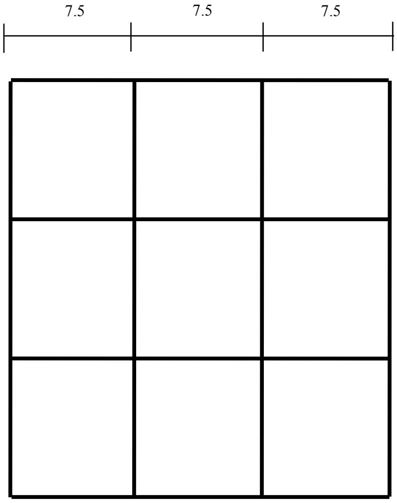


Figure 3. The first building models group.

Figure 4. The second building models group.

Different methods can be considered as a termination criterion for the algorithm, including convergence rate or the number of iterations. Based on the evaluations implemented in this work, the maximum iteration number is selected as the best method to stop the algorithm. It should be noted that the cost of the unit volume of concrete is assumed as 28.5 USD; the cost of reinforcing bars is considered as 46 USD per kN, and the formwork cost is considered as 4.3 USD per unit area.

THREE-BAY 5-STORY FRAME

In this model, a three-bay 5-story frame has been evaluated. The spans are 7.5 meters and the height of each story is 3.5 meters. Similar to the previous example, beams are grouped into three types, and columns are grouped into four types. The structure is subjected to the same loads as considered in the previous example. The topology, loading, and the grouping of the beams and columns are shown in Figure 5.

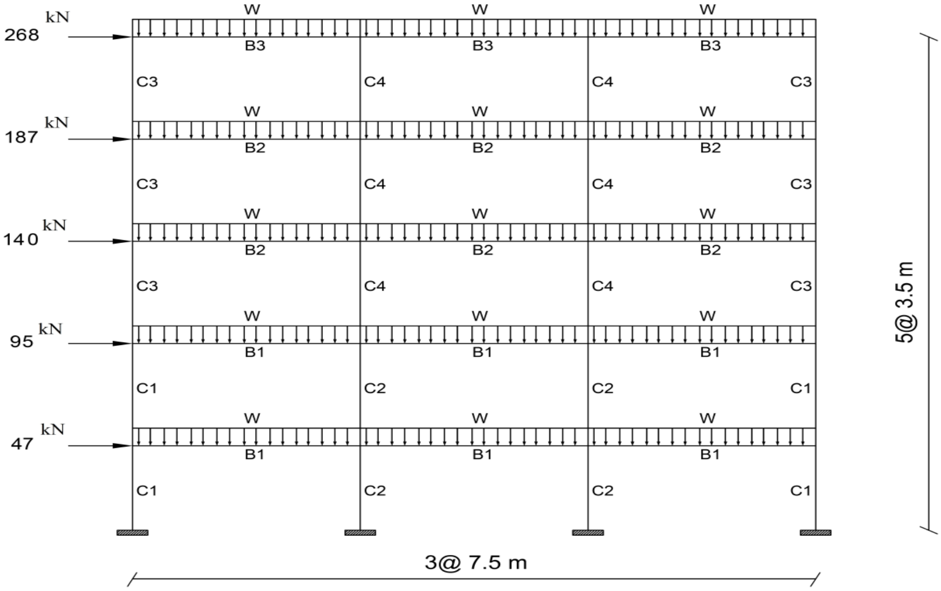


Figure 5. Topology, loading, and members grouping scheme of the three-bay 5-story frame.

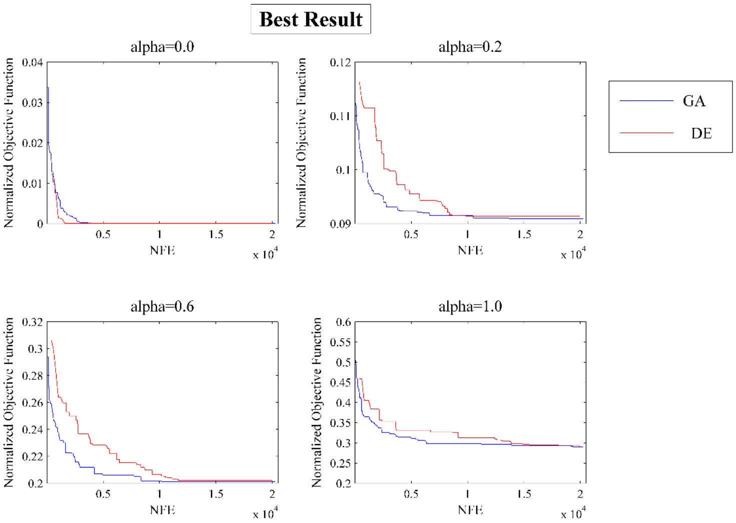


Figure 6. Convergence histories obtained for the three-bay 5-story frame.

Figure 6 shows the convergence history of the normalized objective function for the best solutions of the selected scenarios obtained using the algorithms, where the normalized objective function is composed of the normalized cost and displacement functions. The results of GA and DE are given for comparison. The Pareto front is obtained for this building model using GA and DE as shown in Figure 7. Comparing these trade-offs, the results obtained by GA are better than the results of DE.

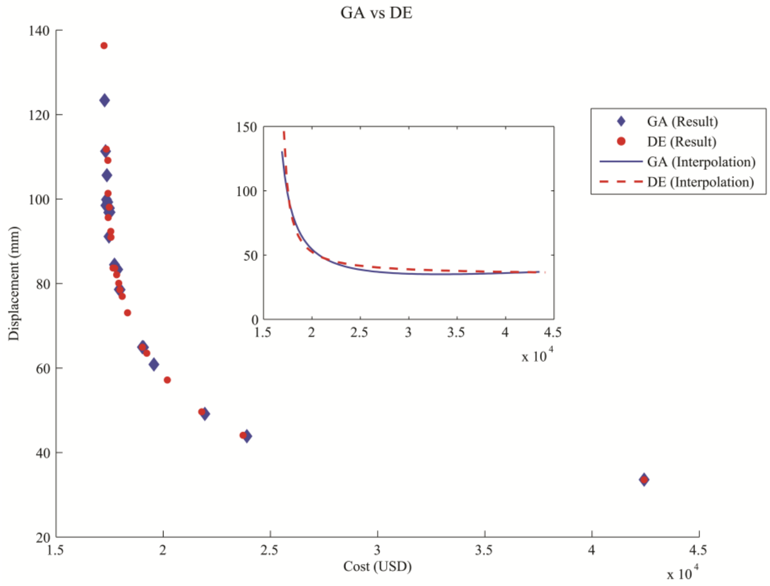


Figure 7. Pareto front obtained for the three-bay 5-story frame.

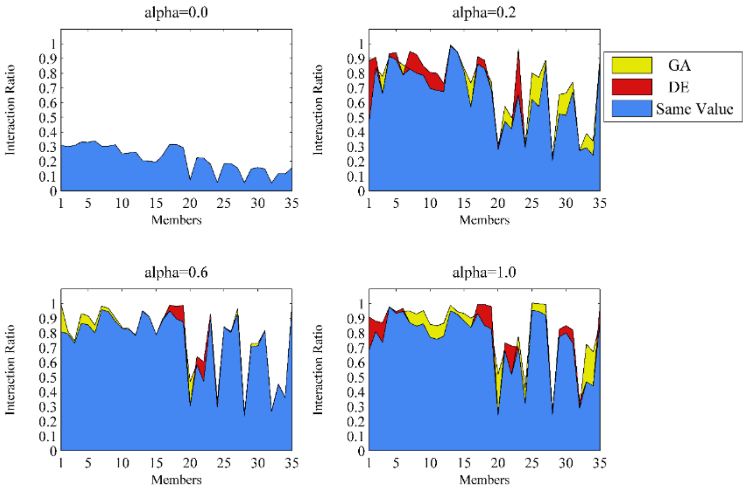


Figure 8. Maximum stress ratio of members for the selected scenarios of the three-bay 5-story frame.

Figure 8 displays the maximum stress ratio of the members obtained using GA and DE for four of the selected scenarios within the Pareto front. Although there are small differences between the ratios obtained by GA and DE, the best solutions obtained by genetic algorithm have more stress ratios. In other words, GA found better frames in terms of the normalized objective function, as can be seen in the figure. On the other hand, as expected, the interaction ratios are increased when the importance of the cost function is increased by raising the alpha.

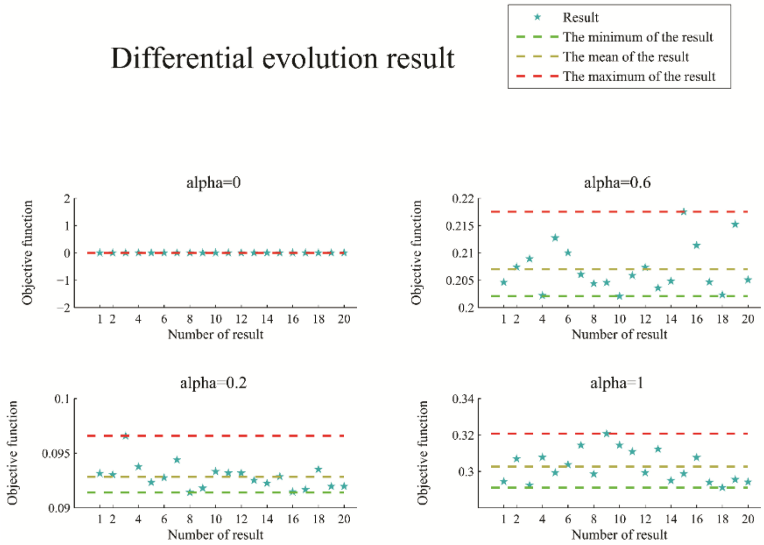


Figure 9. Optimal objective function obtained by DE for three-bay 5-story frame (100 independent runs).

Results of 100 independent runs of four different scenarios are illustrated in Figures 9 and 10, obtained by DE and GA, respectively. Table 2 shows the minimum, maximum, mean, and SD values of these runs. In this example, the convergence speed and the optimal designs of GA are slightly better than the optimal solutions of DE. The best designs obtained by GA are 0.5%, 0.4% and 0.4% optimal than the designs of DE for the three scenarios (alpha=0.2, 0.6 and 1.0), respectively. The SDs obtained for 100 independent runs of DE and GA are considerably small, by less than 3.0% and 2.0%, respectively, which prove the robustness of both algorithms, especially DE.

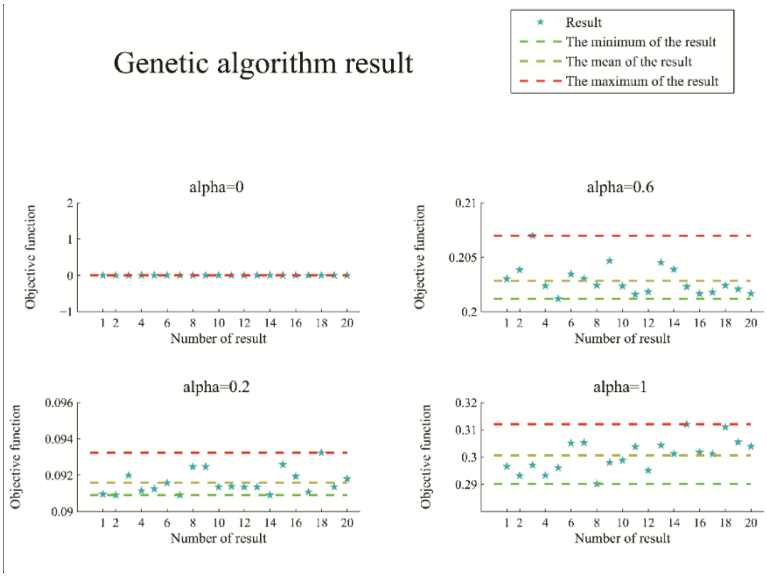


Figure 10. Optimal objective function obtained by GA for three-bay 5-story frame (100 independent runs).

Table 1: The obtained values for three-bay 5-story frame (100 independent runs).

DE					GA				
Alpha	0	0.2	0.6	1.0	Alpha	0	0.2	0.6	1.0
Min	1.05E-06	0.091399	0.20205	0.29116	Min	1.05E-06	0.090898	0.201181	0.290075
Mean	1.05E-06	0.092849	0.207032	0.302519	Mean	1.05E-06	0.091592	0.202840	0.300603
Max.	1.05E-06	0.09657	0.21752	0.32065	Max.	1.05E-06	0.093236	0.206975	0.311966
Standard deviation	2.17E-22 (0%)	0.001059 (1.1%)	0.003870 (1.9%)	0.008724 (3.0%)	Standard deviation	2.17E-22 (0%)	0.000659 (0.7%)	0.001393 (0.7%)	0.005838 (2.0%)
NFE=20030.					NFE=20000.				

FOUR-BAY 10-STORY FRAME

The topology, the members grouping numbers, and the applied loads of a four-bay 10-story frame are shown in Figure 11. In this example, type I is related to the beams of the first to third floors; type II is for beams of four to six floors; type III is for the beams of the seventh to ninth floors, and type IV is related to the beams of the roof level. The columns are grouped so that, type I and type II are for the first three floors of internal and external columns, respectively. Similarly, the fourth to sixth floors are assumed as of two types of columns and the four upper floors of the structure are grouped into two other types, including the internal and external columns.

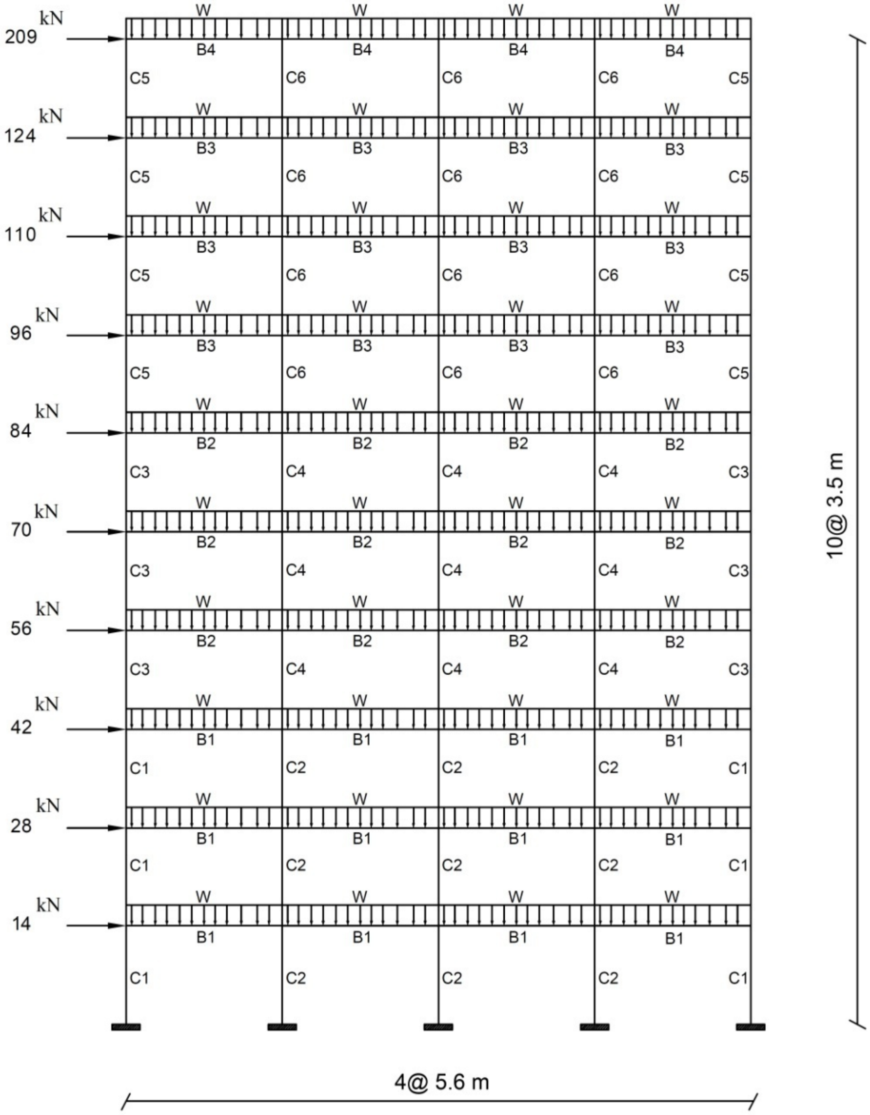


Figure 11. Topology, loading, and members grouping scheme of the four-bay 10-story frame.

The convergence histories of the selected scenarios are drawn in Figure 12, where the results of GA and DE are given for comparison. As shown in the figure, the optimization procedure using GA is faster than DE in many cases, and GA could find the best solutions in lower iterations. The Pareto front is obtained and shown in Figure 13, comparing GA and DE results, which shows the efficiency of GA. In general, the results obtained using GA are better than the results of DE.

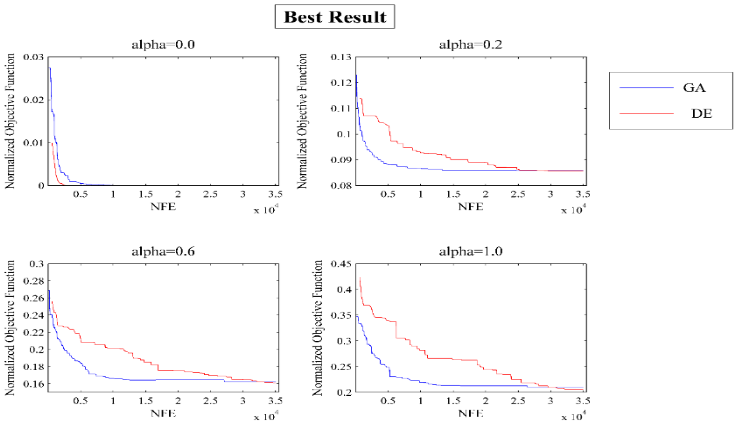


Figure 12. Convergence histories obtained for the four-bay 10-story frame.

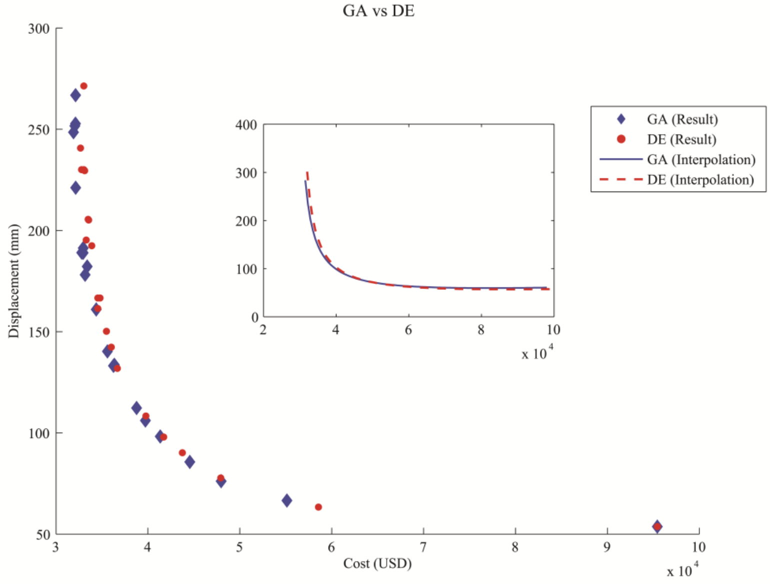


Figure 13. Pareto front obtained for the four-bay 10-story frame.

Figure 14 illustrates the interaction ratios of beams and columns obtained using GA and DE for the selected scenarios selected from the first rank of the Pareto front. The ratios of the best solutions obtained by GA are bigger than those obtained by DE. It means that GA found better cross-sections compared to DE. It is also clear from Figure 13 that the results of GA are better, especially when the importance of the cost function is increased by raising the alpha.

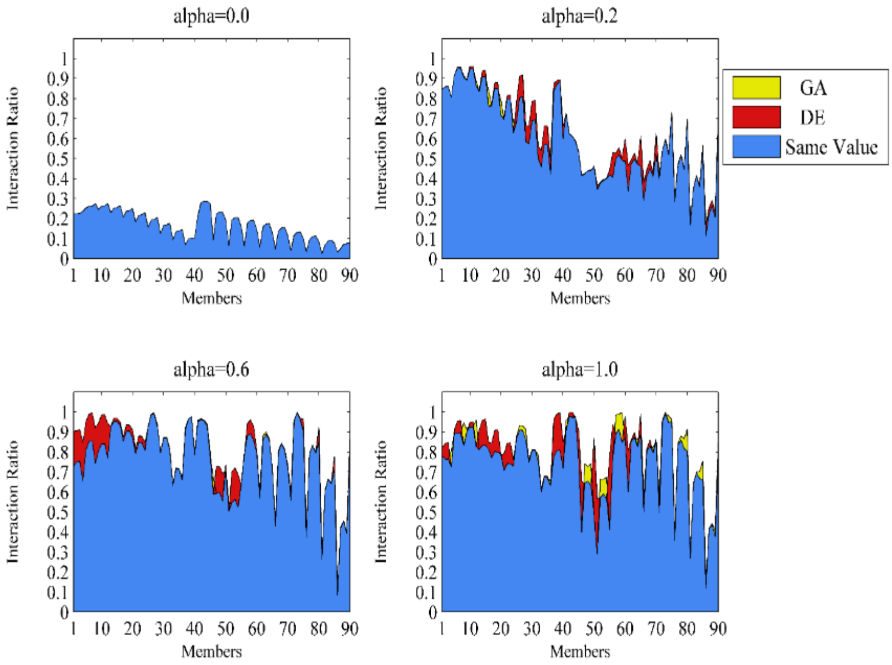


Figure 14. Maximum stress ratio of members for the selected scenarios of the four-bay 10-story frame.

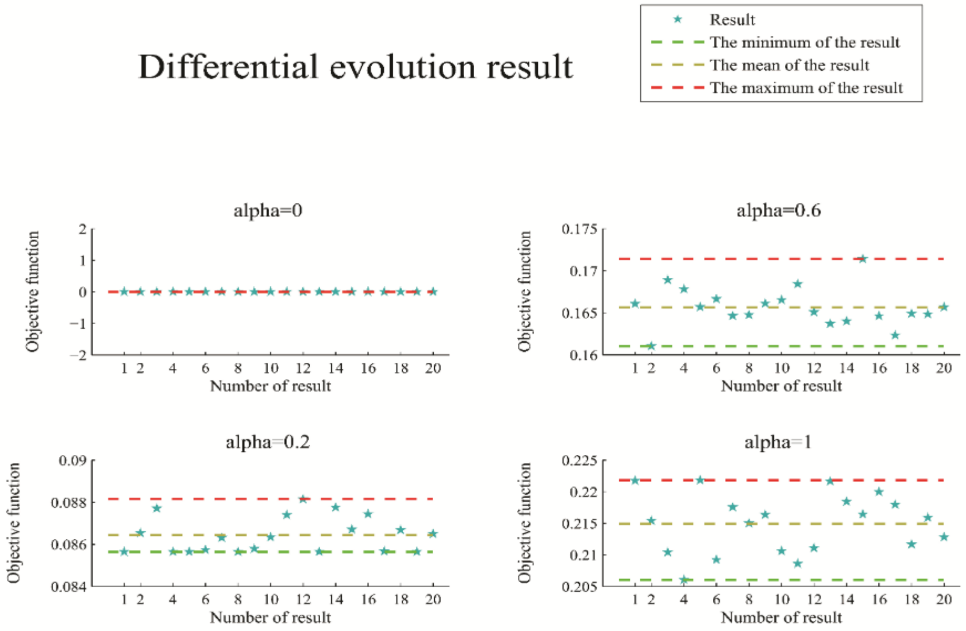


Figure 15. Optimal objective function obtained by DE for four-bay 10-story frame (100 independent runs).

Results of 100 independent runs of four different scenarios using DE and GA are depicted in Figures 15 and 22, respectively. Although the convergence speed of GA is better than DE, however, the best designs of DE are 0.1%, 0.5%, and 1.7% optimal than designs of GA for the three scenarios (alpha=0.2, 0.6, and 1.0), respectively. Table 3 displays the related values of these runs, where the resultant SDs are considerably small (less than 2.3%).

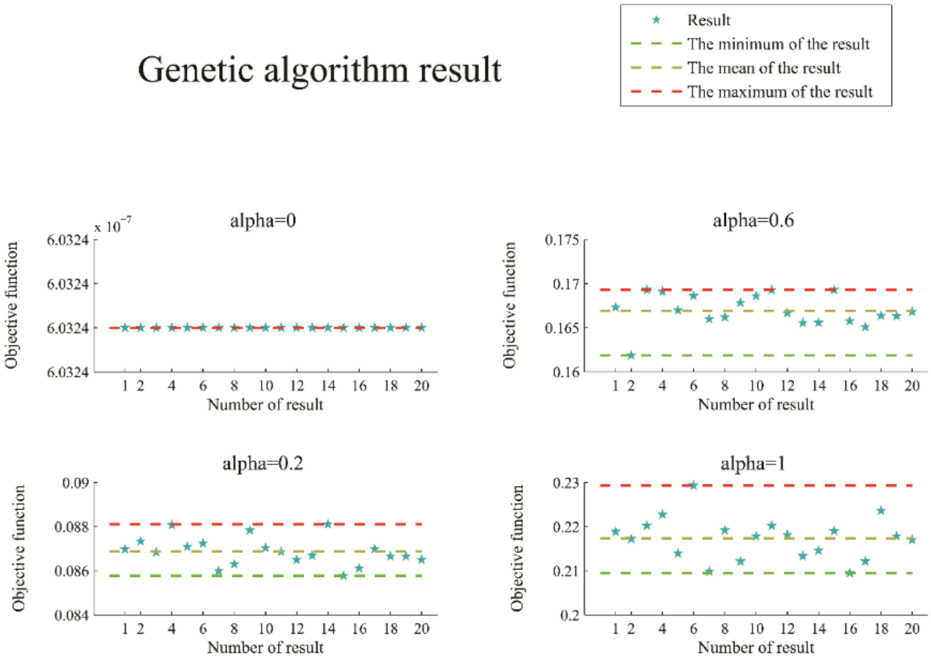


Figure 16. Optimal objective function obtained by GA for four-bay 10-story frame (100 independent runs).

Table 2: The obtained values for four-bay 10-story frame (100 independent runs).

DE					GA				
Alpha	0	0.2	0.6	1.0	Alpha	0	0.2	0.6	1.0
Min	6.03E-07	0.08565	0.16105	0.20605	Min	6.03E-07	0.085773	0.16188	0.20949
Mean	6.03E-07	0.086431	0.165662	0.214935	Mean	6.03E-07	0.086873	0.166930	0.217305
Max.	6.03E-07	0.08815	0.17139	0.22182	Max.	6.03E-07	0.088106	0.1693	0.22929
Standard deviation	1.08E-22 (0%)	0.000752 (0.9%)	0.002052 (1.3%)	0.004695 (2.3%)	Standard deviation	1.08E-22 (0%)	0.000563 (0.6%)	0.001636 (1.0%)	0.004839 (2.3%)
NFE=35010.					NFE=35030.				

FOUR-BAY 14-STORY FRAME

According to the design codes, buildings up to 50 meters in height could be analyzed using the static analysis method. Therefore, a four-bay 14-story building model is tested by the algorithm as the last case study to show the efficiency of the method. The topology, loading, and the grouping of the elements for this model are given in Figure 17. The columns are grouped into eight groups and the beams are grouped into five groups. The trade-off between two objective functions is displayed in Figure 18, and the convergence histories for the selected scenarios are shown in Figure 19. Figure 20 illustrates the interaction ratios for the beams and columns obtained for the optimal solutions using GA and DE. Based on the results, GA shows adequate efficiency to find global optima, in terms of the trade-off between objective functions, robustness, and stress ratios of the elements.

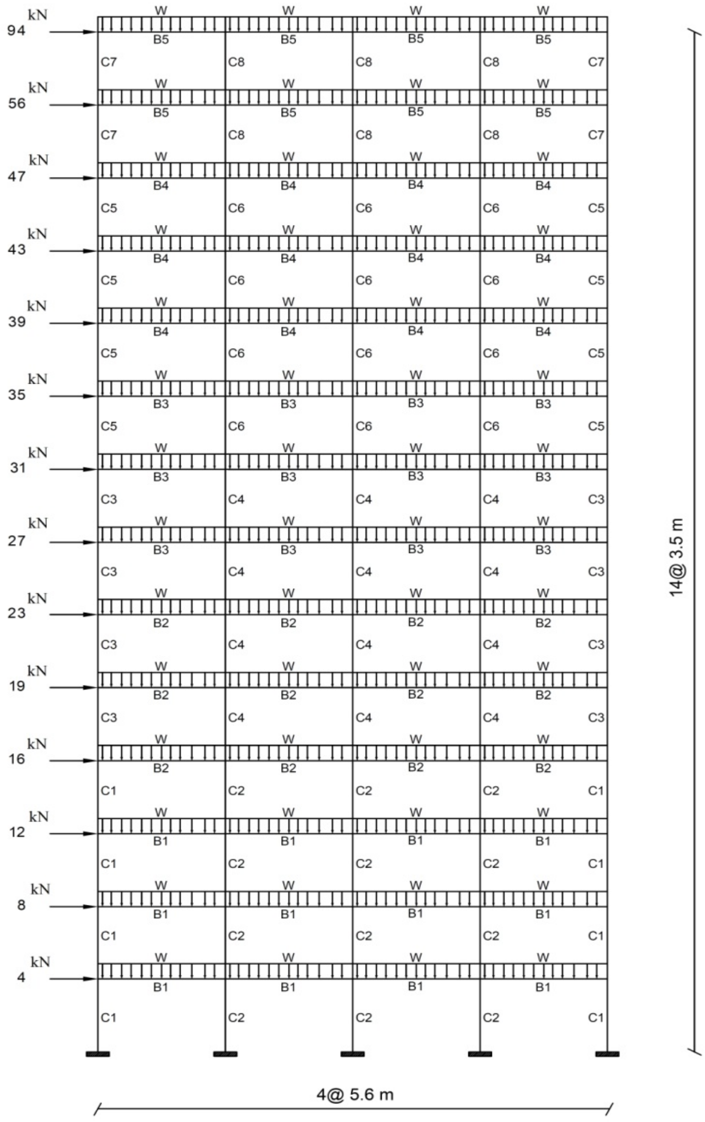


Figure 17. Topology, loading, and member grouping scheme of the four-bay 14-story frame.

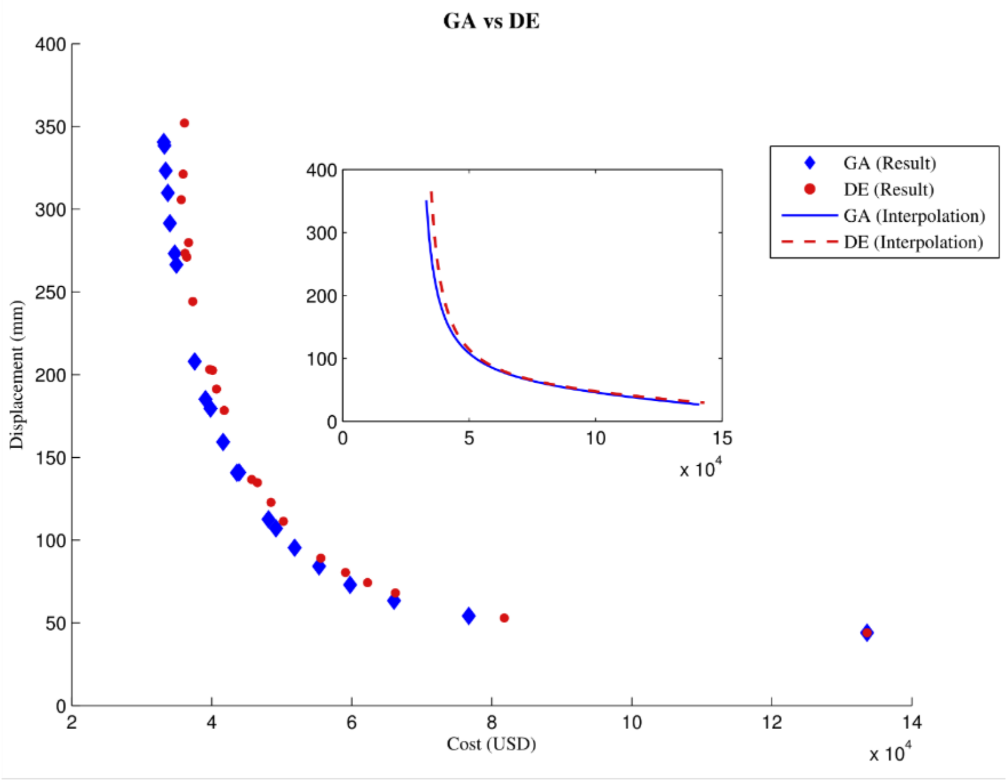


Figure 18. Pareto front obtained for the four-bay 14-story frame.

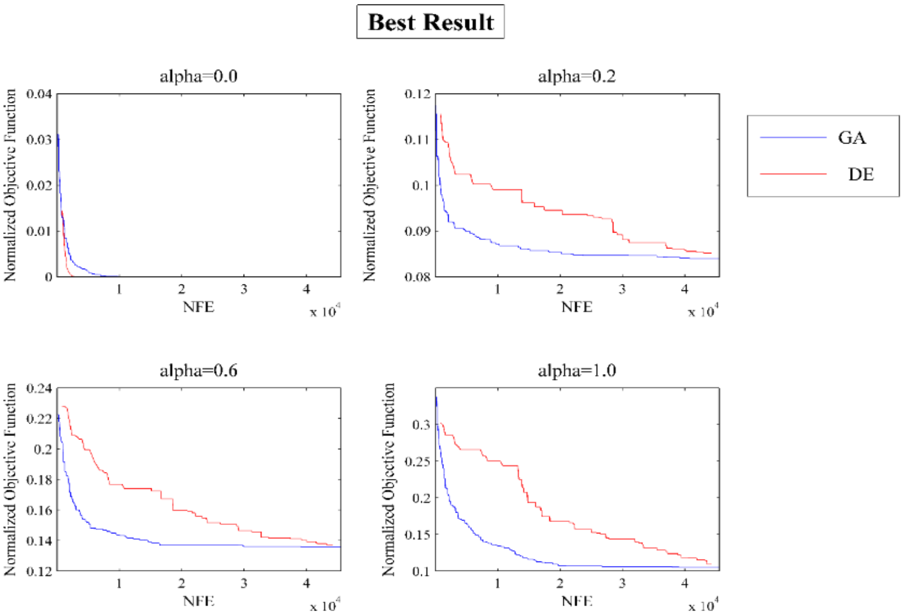


Figure 19. Convergence histories obtained for the four-bay 14-story frame.

In this example, the convergence speed of GA is higher than DE and the results of 100 independent runs show that it is capable to find the best designs which are 1.4%, 1.0%, and 4.2% optimal than designs of DE for the three scenarios (alpha=0.2, 0.6, and 1.0), respectively, as shown in Figures 21 and 22. Table 5 displays the related results, where the maximum SD's are obtained 2.8%, and 2.1% for 100 independent runs of DE and GA.

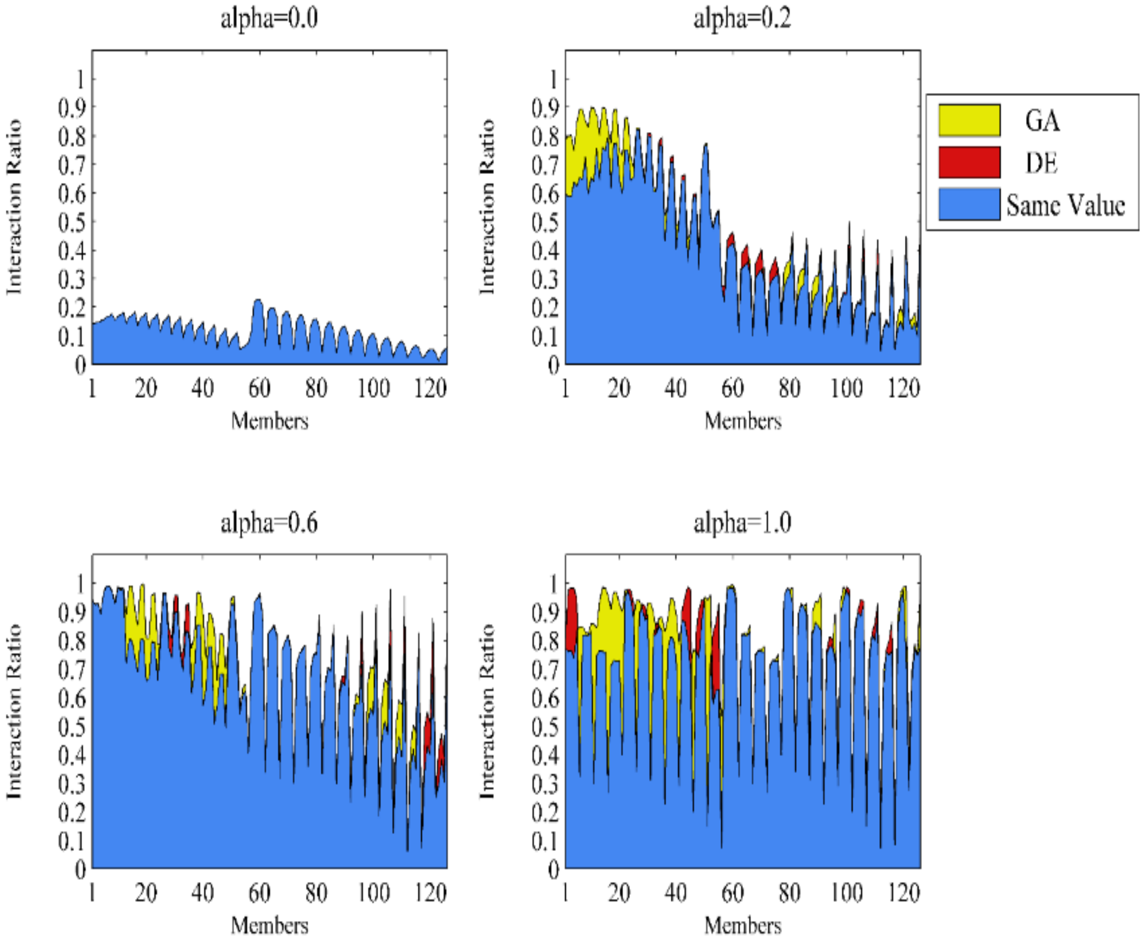


Figure 20. Maximum stress ratio of members for the selected scenarios of the four-bay 14-story frame.

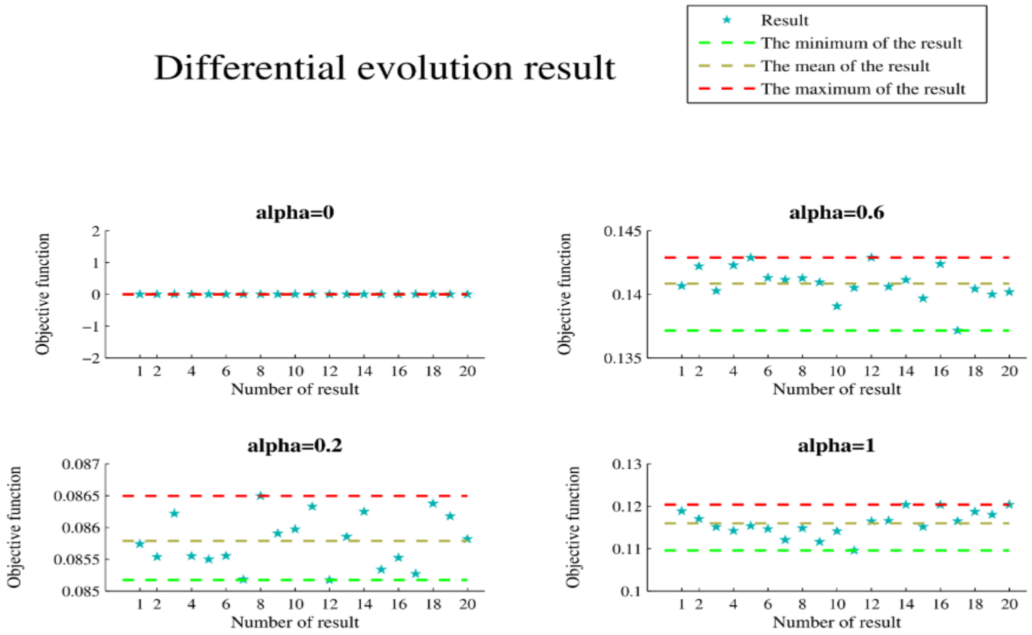


Figure 21. Optimal objective function obtained by DE for four-bay 14-story frame (100 independent runs).

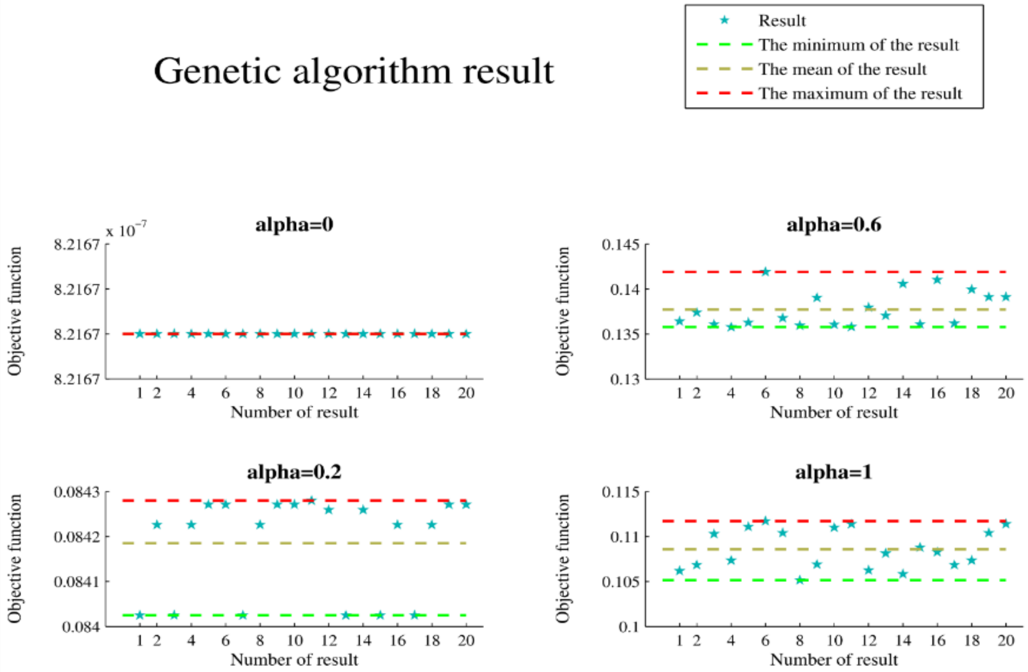


Figure 22. Optimal objective function obtained by GA for four-bay 14-story frame (100 independent runs).

Table 3: The obtained values for four-bay 14-story frame (100 independent runs).

DE					GA				
Alpha	0	0.2	0.6	1.0	Alpha	0	0.2	0.6	1.0
Min	8.22E-07	0.085174	0.13715	0.10957	Min	8.22E-07	0.084025	0.13577	0.10515
Mean	8.22E-07	0.085787	0.140849	0.11546	Mean	8.22E-07	0.084185	0.137720	0.108572
Max.	8.22E-07	0.086496	0.14289	0.12038	Max.	8.22E-07	0.08428	0.14189	0.11171
Standard deviation	2.17E-22 (0%)	0.000370 (0.4%)	0.001210 (0.9%)	0.003105 (2.8%)	Standard deviation	2.17E-22 (0%)	9.71E-05 (0.1%)	0.001750 (1.3%)	0.002176 (2.1%)
NFE=44240.					NFE=46840.				

CONCLUSION

In this study, optimal design of reinforced concrete moment resisting frames is investigated considering two objective functions and using two different algorithms: genetic algorithm and differential evolution. Since the single-objective algorithms are utilized to solve two-objective optimization problem, the objective functions are converted to a single combined function using the weighted sum method (WSM). Two objective functions of the total structural cost and the maximum roof displacement are considered as objective functions for optimization. Following normalization of objective functions, the final composite objective function is obtained and considered as the main objective to be minimized.

The cross-sectional area of the beams and columns are assumed as design variables. A list of limited number of cross-sections is utilized as discrete design variables for structural members. To test the efficiency of GA and DE five 2D RC frames are investigated and the results of optimal frames are given and compared. To illustrate the trade-off between the objective functions, Pareto front is obtained. Four scenarios are extracted from the Pareto front to investigate the efficiency of the methods in more details using the results of 100 independent runs of the algorithms. The main findings are listed as below.

1. The results of both algorithms are extremely similar. In almost all cases there is only 0%-2% difference between the optimal results of DE and GA.
2. Considerably small SDs (by less than 5.2%) obtained from the 100 individual runs prove the efficiency and robustness of the algorithms.

3. The trade-off curves between the two objective functions obtained for all examples using both algorithms are very smooth and approximately the same.
4. In most cases, although the speed of convergence of GA is higher than DE, however, the final optimal designs of DE are slightly better than GA.
5. The stress ratios are decreased when the displacement objective function is more important (case: alpha=0.0). In this case, the best designs of DE are approximately identical to the best solutions of GA.
6. The most difference between the optimal results of DE and GA are obtained when the importance of the cost objective function is increased. In this case, the stress ratios are going to approach the maximum possible amount (case: alpha=1.0).
7. As a future work, the application of the approach could be evaluated for tall RC buildings with or without shear walls.

Conflict of Interest The authors declare that they have no conflict of interest.

REFERENCES

Adamu, A.; Karihaloo, B. & Rozvany, G. (1994). Minimum cost design of reinforced concrete beams using continuum-type optimality criteria. Structural Optimzation. 7(1-2), 91-102.

Akin, A. & Saka, M. (2015). Harmony search algorithm based optimum detailed design of reinforced concrete plane frames subject to ACI 318-05 provisions. Comput. Struct. 147, 79-95.

American Concrete Institute (ACI). (2005). Building code requirements for structural concrete and commentary, ACI 318-05.

Babaei, M. (2015). Exploring practical optimal topology for reinforced concrete moment resisting frame structures. American Journal of Civil Engineering. 3(4), 102-106. https://doi.org/10.11648/j.ajce.20150304.12

Babaei, M. (2017). Multi-objective optimal number and location for steel outrigger-belt truss system. International Journal of Engineering Science and Technology. 12(10), 2599-2612.

Babaei, M. & Mollayi, M. (2016). Multi-objective optimization of reinforced concrete frames using NSGA-II algorithm. Engineering Structures and Technologies. 8(4), 157-164. https://doi.org/10.3846/2029882X.2016.1250230

Babaei, M. & Mollayi, M. (2020). An improved constrained differential evolution for optimal design of steel frames with discrete variables. Mechanics Based Design of Structures and Machines. 48(6), 697-723. DOI: 10.1080/15397734.2019.1657890

Babaei, M. & Sanaei, E.(2016). Multi-objective optimal design of braced frames using hybrid genetic and ant colony optimization algorithm. Frontiers of Structural and Civil Engineering. 10(4), 472-480. https://doi.org/10.1007/s11709-016-0368-4

Boscardin, J.T.; Yepes, V. & Kripka, M. (2019). Optimization of reinforced concrete building frames with automated grouping of columns. Automation in Construction. 104, 331-340. https://doi.org/10.1016/j.autcon.2019.04.024

Camp, C. V. & Huq, F. (2013). CO2 and cost optimization of reinforced concrete frames using a big bang-big crunch algorithm. Engineering Structures. 48, 363-372 (2013)

Chan, C.M. & Wang, Q. (2006). Nonlinear stiffness design optimization of tall reinforced concrete buildings under service loads. Journal of structural engineering. 132(6), 978-990.

Deb K. (2001). Multi-objective Optimization using Evolutionary Algorithm. John Wiley & Sons

D'Errico, F. (2015) Multi-objective Optimization in Engineering Design, in Material Selections by a Hybrid Multi-criteria Approach. Springer.

Esfandiari, M.J.; Urgessa, G.S.; Sheikholarefin, S. & Dehghan Manshadi, S.H. (2018). Optimum design of 3D reinforced concrete frames using DMPSO algorithm. Advances in Engineering Software. 115, 149-160. https://doi.org/10.1016/j.advengsoft.2017.09.007

Farahmand-Tabar, S. & Babaei, M. Simultaneous Shape and Size Optimization of Truss Structures using Multi-Verse Optimizer. Engineering with Computers. under review

Ho-Huu, V.; Nguyen-Thoi, T.; Nguyen-Thoi, M. & Le-Anh, L. (2015). An improved constrained differential evolution using discrete variables (D-ICDE) for layout optimization of truss structures. Expert Systems with Applications. 42(20), 7057-7069

Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. Ann Arbor, University of Michigan Press

Kaveh, A. (2017). Cost and CO_2 Emission Optimization of Reinforced Concrete Frames Using Enhanced Colliding Bodies Optimization Algorithm. In: Applications of Metaheuristic Optimization Algorithms in Civil Engineering. Springer.

Kaveh, A. & Behnam, A.F.(2013). Design optimization of reinforced concrete 3D structures considering frequency constraints via a charged system search. Scientia Iranica. 20 (3), 387-396. https://doi.org/10.1016/j.scient.2012.11.017

Kaveh, A. & Mahdavi, V.R. (2015). Optimal Design of Concrete Structures Using Colliding Bodies Optimization Colliding Bodies Optimization. Springer. https://doi.org/10.1007/978-3-319-19659-6_8

Kaveh, A. & Sabzi, O. (2011). A comparative study of two meta-heuristic algorithms for optimum design of reinforced concrete frames. International Journal of Civil Engineering. 9(3), 193-206.

Kaveh, A. & Sabzi, O. (2012). Optimal design of reinforced concrete frames Using big bang-big crunch algorithm. International Journal of Civil Engineering. 10(3), 189-200.

Kaveh, A. & Zakian, P. (2014). Optimal seismic design of reinforced concrete shear wall-frame structures. KSCE Journal of Civil Engineering. 18(7), 2181-2190.

Kaveh, A. & Zarandi, M. (2019). Optimal Design of Steel-Concrete Composite I-girder Bridges Using Three Meta-Heuristic Algorithms. Periodica Polytechnica. Civil Engineering. 63(2), 317-337 .

Kwak, H.-G. & Kim, J. (2009). An integrated genetic algorithm complemented with direct search for optimum design of RC frames. Computer-Aided Design. 41(7),490-500.

Leyva, H.A.; Bojórquez, E.; Bojórquez, J.; Reyes-Salazar, A.; Castorena, J.H.; Fernández, E. & Barraza, M.A. (2018). Earthquake design of reinforced concrete buildings using NSGA-II. Advances in Civil Engineering. Article ID 5906279, 11 pages. https://doi.org/10.1155/2018/5906279.

Marler, R.T. & Arora, J.S. (2004). Survey of multi-objective optimization methods for engineering. Structural and multidisciplinary optimization. 26(6), 369-395.

Rajeev, S. & Krishnamoorthy, C.S. (1998). Genetic algorithm-based methodology for design optimization of reinforced concrete frames. Comput Aided Civil Infrastruct Eng. 13, 63–74. https://doi.org/10.1111/0885-9507.00086

Sanaei, E. & Babaei, M. (2011). Cellular automata in topology optimization of continuum structures. International Journal of Engineering, Science and Technology. 3(4), 27-41. https://doi.org/10.4314/ijest.v3i4.68538

Sanaei, E. & Babaei, M. (2012). Topology optimization of structures using cellular automata with constant strain triangles. International Journal of Civil Engineering. 10(3), 179-188

Sarma, K.C. & Adeli, H. (1998). Cost optimization of concrete structures. Journal of Structural Engineering. 124(5), 570-578.

Storn, R. & Price, K. (1995). Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, 1–12.

Storn, R. & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization. 11(4), 341-359

Zou, X. K.; Chan, C. M.; Li G. & Wang, Q. (2007). Multiobjective optimization for performance-based design of reinforced concrete frames. Journal of Structural Engineering. 133(10), 1462-1474.