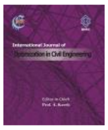
INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2020; 10(4):611-627



OPTIMAL DESIGN OF STEEL MOMENT FRAME STRUCTURES USING THE GA-BASED REDUCED SEARCH SPACE (GA-RSS) TECHNIQUE

B. Kamali Janfada and M. R. Ghasemi[1*, †]
Department of Civil Engineering, University of Sistan and Baluchestan, Zahedan, Iran

ABSTRACT

This paper proposes a GA-based reduced search space technique (GA-RSS) for the optimal design of steel moment frames. It tries to reduce the computation time by focusing the search around the boundaries of the constraints, using a ranking-based constraint handling to enhance the efficiency of the algorithm. This attempt to reduce the search space is due to the fact that in most optimization problems the optimal solution lies on or near the boundaries of the feasible region. All the analyses/optimization steps have been implemented in MATLAB and the method has been validated by optimizing three moment-frame benchmark problems. According to the results, the algorithm performs fit and needs relatively fewer analyses than other metaheuristic algorithms to reach a global optimum solution.

Keywords: structural optimization; steel frame structures; genetic algorithm; reduced search space; constraint handling.

Received: 25 July 2020; Accepted: 27 October 2020

1. INTRODUCTION

Since steel moment frames are vital practical issues in engineering design problems, their safe and optimal design is very important because their design variables and constraints are numerous and the search space is large [1, 2]. Design variables are discrete in nature because they are selected from a list of standard w-shaped beam and column sections [3, 4].

Major optimization algorithms are either classical or heuristic/metaheuristic. Mathematical programming (classical), are not suitable for solving large engineering problems because they start the search from a single selected continuous point causing their

[*] Corresponding author: Department of Civil Engineering, University of Sistan and Baluchestan, Zahedan, Iran

[†] E-mail address: mrghasemi@eng.usb.ac.ir (M.R. Ghasemi)

final solution to get trapped in a local optimum which depends on that initial point. Besides, since they need the gradient information of the objective function/constraints and thus, a large problem size makes reaching for the optimal solution difficult [1, 5]. However, metaheuristic algorithms including Genetic Algorithm (GA) [6-9], Ant Colony Optimization algorithm (ACO) [10, 11], Particle Swarm Optimization (PSO) [12], Harmony Search (HS) [13-15], Teaching–Learning-Based Optimization (TLBO) [16], Big Bang-Big Crunch (BB-BC) [17], Shuffled Shepherd Optimization Algorithm (SSOA) [18],¬¬ Billiards-inspired Optimization Algorithm (BOA) [19], hybrid algorithms [1, 5] and so on are random search-based; therefore, they are more suitable for moment frame optimization problems because they can handle discrete variable type problems as well.

Many researchers have employed meta-heuristic algorithms for optimal design of steel frame structures. Kripakaran et al. [20] used an alternative method, combined with the genetic algorithm, to carry out the optimal design of steel moment resisting frames. Kaveh and Zakian [21] employed charged system search (CSS) and improved harmony search algorithms in order to design of steel frames. Kaveh and Bakhshpoori [22] performed optimum design of two-dimensional steel frames by means of Cuckoo search (CS) algorithm with Levy flights. Flager et al. [23] presented the Fully Constrained Design (FCD) method for discrete sizing optimization of steel structures. Kaveh et al. [24] employed the non-dominated sorting genetic algorithm (NSGA-II) to minimize construction cost and reducing seismic damage of steel frame structures. Mahallati Rayeni et al. [25] developed an improved Multi-Objective Evolutionary Algorithm (IMOEA) in order to design planar steel frames. Kaveh and Ghazaan [26] performed optimum seismic design of 3D irregular steel frames using four metaheuristic algorithms. Also in the literature [27], seven population-based meta-heuristic algorithms were employed for size optimization of two-dimensional steel frame structures.

Since metaheuristic algorithms need numerous objective function/constraint evaluations, some researchers have tried, through a number of studies, to reduce the required computational time [8, 28, 29]. This paper has used a novel search-space reduction technique and shown, by comparing its results with those of other similar researches, that it is quite efficient in finding the problem’s optimal solution.

2. FRAME OPTIMIZATION PROBLEMS

Optimization of steel moment frames, with the following formulation, is aimed to yield a least-weight structure design so that the constraints are satisfied:

$$f(x) = \sum_{i=1}^{n} \gamma_i A_i L_i \tag{1}$$

where γ_i, A_i and L_i are the material density, sectional area, and length of member i, respectively, and n is the number of members. The AISC frame design is based on the following constraints [30]:

Member normalized tension:

$$v_i^{\alpha} = \left|\frac{\sigma_i}{\sigma_i^{\alpha}}\right| - 1 \leq 0 \qquad i = 1,2,\dots,n \tag{2}$$

Maximum normalized lateral displacement:

$$v^{\Delta} = \frac{\Delta_T}{H} - R \leq 0 \tag{3}$$

Inter-story displacements:

$$v_j^d = \frac{d_j}{h_j} - R_I \leq 0 \qquad j = 1,2,\dots,n_s \tag{4}$$

where σ_i^{α} and σ_i are the existing and allowable stress in member i, respectively, R is the maximum allowable drift, Δ_T is the structure's maximum lateral displacement, H is the total structure height, d_j is the inter-story drift, h_j is the height of story j, R_I is the allowable inter-story drift index (= 1/300 according to AISC) and n_s is the number of stories. The LRFD interaction constraints relationships are as follows:

$$v_i^I = 1 - \frac{P_u}{2\phi_c P_n} - \left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}}\right) \leq 0 \quad \text{For } \frac{P_u}{\phi_c P_n} < 0.2 \tag{5}$$

$$v_i^I = 1 - \frac{P_u}{\phi_c P_n} - \frac{8}{9}\left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}}\right) \leq 0 \quad \text{For } \frac{P_u}{\phi_c P_n} \geq 0.2 \tag{6}$$

where P_u and P_n are the required and nominal axial resistance (tension or compression), respectively, ϕ_c is the resistance factor (= 0.9 for tension and 0.85 for compression), (M_{ux}, M_{uy}) and (M_{nx}, M_{ny}) are the required and nominal flexural strength around the x and y axes, respectively (M_{uy}= 0 for 2D frames), and ϕ_b (= 0.9) is the flexural strength reduction factor. The effective length factor k is needed to find the Euler stresses; it equals 1 for beams and braced members, but for columns, use is made of the following approximate relation with an accuracy of -1% to + 2% of the exact solution [31]:

$$K = \sqrt{\frac{1.6G_A G_B + 4(G_A + G_B) + 7.5}{G_A + G_B + 7.5}} \tag{7}$$

where G_A and G_B are the column stiffness ratios at both ends.

3. GENETIC ALGORITHM

Genetic algorithms are metaheuristic methods that work based on the "natural evolution" mechanism and "survival of the fittest" principle and use such operators as the mutation and crossover inspired by the biological evolution [28, 32]. Since they are inherently developed to solve unconstrained problems, a challenge in their application is how to handle constraints for which many methods have been proposed by different researchers [33-36].

3.1. Constraint treatment

To control and handle constraints in optimization problems, this paper has used the extended balanced ranking method (E-BRM) [37] the explanation of which first requires the definition of the general form of a constrained optimization problem as follows:

$$\text{Optimize } f(\vec{x})$$
$$\text{Subject to: } f(\vec{x}) = \begin{cases} g_j(\vec{x}) \le 0 & j = 1,2,\dots,q \\ h_j(\vec{x}) = 0 & j = q+1,\dots,m \\ x_i^{Low} \le x_i \le x_i^{Up} & i = 1,2,\dots,\text{nvar} \end{cases} \tag{8}$$

where vector $\vec{x}$ is a solution with nvar design variable,$f(\vec{x})$ is the objective function to be optimized, $g_j(\vec{x})$ and $h_j(\vec{x})$ are the unequal and equal constraints, respectively, x_i^{Low} and x_i^{Up} are, respectively, the lower and upper bounds of variable i (from the set of design variables) q is the number of unequal constraints and m is the total number of constraints; feasible solutions (FS) are those that satisfy these constraints and infeasible solutions (IS) refer to those that do not do so [37].

The constraint violation is found as follows:

$$v_j(\vec{x}) = \begin{cases} \max\{0, g_j(\vec{x})\} & if\ 1 \le j \le q \\ \max\{0, |h_j(\vec{x})| - \varepsilon\} & if\ q+1 \le j \le m \end{cases} \tag{9}$$

where ε is a small value used to convert equal constraints into unequal ones [37].

The penalty function for infeasible solutions is [37]:

$$p(\vec{x}) = \sum_{j=1}^{m} |v_j(\vec{x})|^{\beta} \tag{10}$$

where β is defined as follows:

$$\beta = 2 + \left(1 - f(\sigma)\right) \tag{11}$$

where $f(\sigma)$ is a penalty-balancing function for non-violated constraints to direct the search towards the feasible space [37]:

$$f(\sigma) = \begin{cases} 0 & if\ count(\text{FS}) = 0 \\ \sigma & \text{otherwise} \end{cases} \tag{12}$$

$\text{count}(\text{FS})$ is the number of feasible solutions, σ varies in the [0,1] interval [37]:

$$\sigma = \frac{\hat{C}}{m\ \mathrm{Npop}} \tag{13}$$

$\hat{C}$ is the number of non-violated constraints in the present population and Npop is the population size.

The fitness function for feasible and infeasible solutions is as follows [37]:

$$\mathrm{eval}(\vec{x}) = \begin{cases} \mathrm{rank}(f(\vec{x}), \mathrm{FS}) & x \text{ is feasible} \\ \mathrm{rankWeighted}(\vec{x}, \sigma) + \sqrt{\Delta} + \Psi & x \text{ is infeasible} \end{cases} \tag{14}$$

$\mathrm{rank}(f(\vec{x}), \mathrm{FS})$ is ranking of solution $\vec{x}$ among feasible ones, sorted based on the objective function value.

Another relationship related to infeasible solutions is:

$$\mathrm{rankWeighted}(\vec{x}, \sigma) = \frac{\mathrm{rank}(p(\vec{x}), \mathrm{IS})(1 - f(\sigma)) + \mathrm{rank}(f(\vec{x}), \mathrm{IS}) f(\sigma)}{2} \tag{15}$$

where $\mathrm{rank}(p(\vec{x}), \mathrm{IS})$ and $\mathrm{rank}(f(\vec{x}), \mathrm{IS})$ are ranking of solution x based, respectively, on the values of the penalty and objective functions among infeasible solutions; here,$f(\sigma)$ plays the role of giving weight to two ranking criteria (penalty and objective functions) [37].

$\sqrt{\Delta}$ and Ψ are the integration parameters defined as follows:

$$\sqrt{\Delta} = \sqrt{\mathrm{count}(\mathrm{FS})\,\frac{\mathrm{count}(\mathrm{IS})}{\mathrm{Npop}}} \tag{16}$$

$$\Psi = \begin{cases} 0 & if\,(count(\mathrm{IS}) + \sqrt{\Delta}) > count(FS) \\ \frac{\varphi}{\mathrm{count}(\mathrm{IS}) - 1}(\mathrm{rank}(f(\vec{x}), \mathrm{IS}) - 1) & otherwise \end{cases} \tag{17}$$

where $count(IS)$ is the number of infeasible solutions in the population and φ is:

$$\varphi = \mathrm{count}(\mathrm{FS}) - (\mathrm{count}(\mathrm{IS}) + \sqrt{\Delta}) \tag{18}$$

Users need not adjust any parameter manually because they are handled automatically in the proposed technique. In short, E-BRM is aimed to use the potential of the infeasible solutions and direct the search towards the feasible space.

3.2. Mutation operator

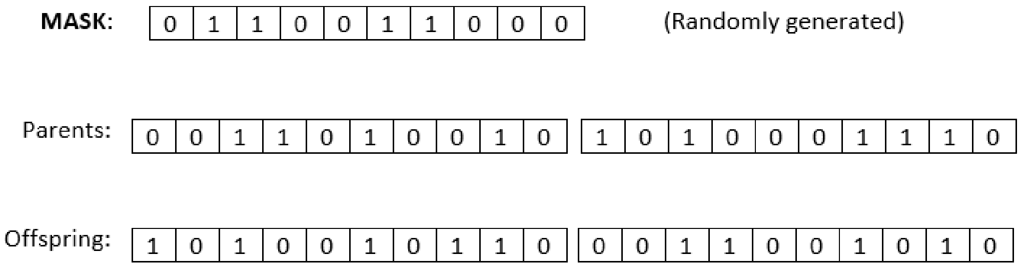
"Mutation" is an important GA operator that plays a vital role in keeping diversity in the population and its absence may cause some search space regions not to be explored [38]. This paper has used the Gaussian mutation to mutate variable i through the following equation:

$$x_i' = x_i + N(0, \sigma_n) \tag{19}$$

where x_i' and x_i are the mutated and primary variables, respectively and $N(0, \sigma_n)$ is a normally distributed random number with 0 mean and σ_n standard deviation [39].

3.3. Crossover operator

“Crossover” is another major operator that combines the characteristics of two parent chromosomes to form two offspring chromosomes [40]; this paper has used the mask and uniform crossovers with equal probabilities for a more effective search of the design space. In the former, 0 and 1 are first used randomly to from a parent chromosome and the offspring ones are then selected from it [25] (Fig. 1).



MASK: 0 1 1 0 0 1 1 0 0 0 (Randomly generated)

Parents: 0 0 1 1 0 1 0 0 1 0 | 1 0 1 0 0 0 1 1 1 0

Offspring: 1 0 1 0 0 1 0 1 1 0 | 0 0 1 1 0 0 1 0 1 0

Figure 1. Schematic of mask crossover

and in the latter, offspring chromosomes O_1 and O_2 are generated from parents P_1 and P_2 as follows [41]:

$$O_1 = \alpha P_1 + (1 - \alpha) P_2 \tag{20}$$

$$O_2 = \alpha P_2 + (1 - \alpha) P_1 \tag{21}$$

where α is the integration factor the value of which in this paper is 0.25.

4. REDUCED SEARCH SPACE (RSS) TECHNIQUE

Since it is very likely, in many constrained optimization problems, that the optimal solution may lie near or on the search space feasible-infeasible boundary region, it would be reasonable if the algorithm emphasized the search more on the boundaries of the feasible regions [42-44]. The method proposed in this paper considers a boundary region as a new feasible one for each constraint that forms the boundaries of the feasible space. As iterations go on, these regions get narrower and the search is focused further within that updated feasible space the details of which are given in Subsections 4.1-4.3.

4.1. External boundary

To include infeasible solutions around constraints’ boundaries, parameter δ_{out} is defined with an initial value $\delta_{out}(0)$ specified after the first iteration; this value is the maximum violation among the top 10% of the population for all constraints. If this value is zero,

$\delta_{out}(0)$ will be taken equal to 1 and δ_{out} is updated until the number of analyses is less than NFE_{δ}; after that it will become zero so as to focus only on non-violated solutions:

$$\delta_{out}(t) = \begin{cases} \delta_{out}(0) * \left(1 - \frac{nfe}{NFE_{\delta}}\right)^{z} & 0 < nfe \leq NFE_{\delta} \\ 0 & nfe > NFE_{\delta} \end{cases} \tag{22}$$

where nfe is the current number of function evaluations and z is a control parameter to reduce δ_{out}.

$$Z = \frac{-20 - \ln(\delta_{out}(0))}{\ln(0.05)} \tag{23}$$

4.2. Internal boundary

To reduce the search space from inside the feasible region, use is made of parameter δ_{in}. In minimization problems, since feasible solutions with higher objective values are considered less fit, $\delta_{in}(0)$ is taken a large value for each constraint so that the whole feasible region is initially considered. The internal boundary decreases in each iteration according to the following relation:

$$\delta_{in}(t) = \delta_{in}(0) \times (1 - \frac{nfe}{NFE_{max}})^{r} \tag{24}$$

where $\delta_{in}(0)$ is set equal to the minimum constraint value among the top 20% of the population; If all the top 20% have violated the constraint, the minimum $\delta_{in}(0)$ is taken equal to 5. NFE_{max} is the highest number of function evaluations and r is a δ_{in} reduction control parameter.

$$r = \frac{-20 - \ln(\delta_{in}(0))}{\ln(0.05)} \tag{25}$$

4.3. Constraint violation

The width of the boundary region for each constraint is found as follows:

$$-\delta_{in} \leq g_j(\vec{x}) \leq \delta_{out} \tag{26}$$

Any $\vec{x}$ solution outside this region is infeasible and its constraint violation is found as follows:

$$V_1 = -\delta_{in} - g(\vec{x}) \tag{27}$$

$$V_2 = \begin{cases} \frac{g(\vec{x})}{10} & g(\vec{x}) > 0 \ and \ g(\vec{x}) < \delta_{out} \\ g(\vec{x}) & g(\vec{x}) \geq \delta_{out} \\ 0 & otherwise \end{cases} \tag{28}$$

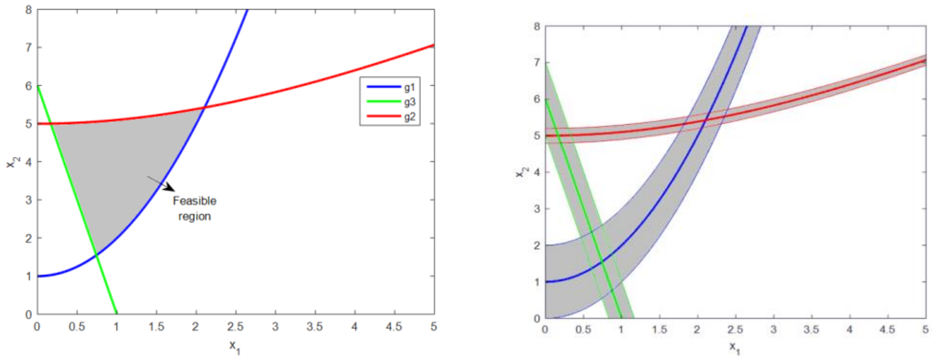
For solutions lying in the constraint’s outer boundary region, a slight violation ($\frac{g(\vec{x})}{10}$) is considered so as to prevent the final solution to be infeasible. The total value of each constraint violation is as follows:

$$V(\vec{x}) = \max(0, \max(V_1, V_2)) \tag{29}$$

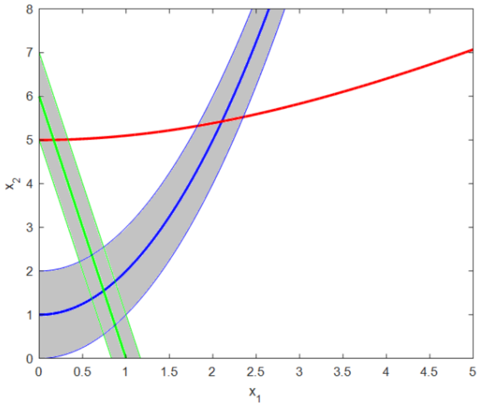
4.4. Elimination of inactive constraints

Among all constraints of any optimization problem, some might be inactive [42]. After passing a predetermined number of iterations (IT_δ), if some constraints of the current optimum solution lie outside the internal boundary, a large value is assigned to the corresponding constraint’s δ_{in} to cover the entire feasible search space and δ_{out} is set to zero; IT_δ is taken equal to 0.1 times the maximum number of iterations.

An example of the RSS performance is shown in Fig. 2; 2(a) shows the total search space and the feasible region formed by the intersection of each constraint’s acceptable regions and 2(b) depicts the boundary region for each constraint $(g_j(\vec{x}))$ made using Eq. (26) after the first iteration. The shaded area identifies the new feasible area (as mentioned before, a slight violation is considered for solutions lying in the outer boundary region). Then, after a predetermined number of iterations, these boundary regions are removed for constraints for which the superior solution lies far from the boundaries $(g_j(\vec{x}) \ll 0)$ (constraint $g2$ in the present example). Fig. 2(c) shows the new reduced search space.



(a) Original feasible region (b) New feasible region for all constraints



(c) New feasible region for all constraints
Figure 2. Example of RSS technique

5. DESIGN EXAMPLES

To check the validity of the proposed method, three benchmark frame structures are optimized and the results are compared with those of other previous studies. The structural analyses and algorithm coding are done in MATLAB and percent crossover and mutation are 80 and 30, respectively.

5.1. Two-bay three-story frame

Fig. 3 shows the configuration and loading of a 2-bay 3-story frame optimized based on the AISC-LRFD criteria. The steel elasticity modulus E is 200 GPa (29000 ksi), yield stress (F_y) is 248.2 MPa (36 ksi), the beams' unbraced length factor was 0.167, beams were all selected from the W-shaped sections of AISC standard list, columns were selected only from W10 sections [7, 25, 45]. The population size in each cycle is 30.

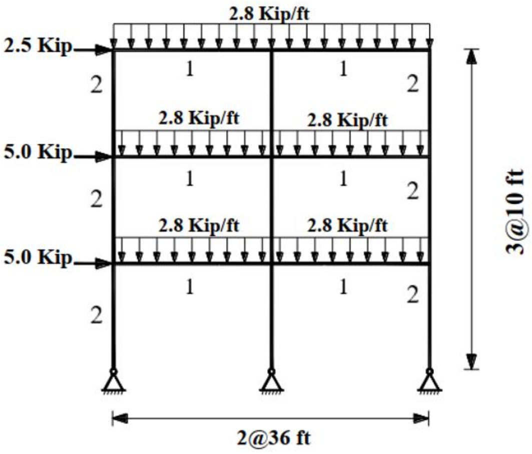


Figure 3. 2-bay 3-story steel frame structure

Fig. 4 shows the convergence history of the mentioned frame optimization. The optimum design with a minimum frame weight of 83.587 KN was obtained by standard GA after 492 analyses while the GA-RSS has done it within 195 analyses. Number of analyses required to meet a converged solution for the GA-RSS algorithm was found significantly less than those carried by Pezeshk et al. [7]. They were also less than those found by DDHS [45] and IMOEA [25]. The average weight of the GA-RSS designs over the 10 independent runs was 84.163 KN, with a standard deviation of 1.82 KN while the average weight of the standard GA designs was 84.451 KN, with a standard deviation of 2.73 KN.

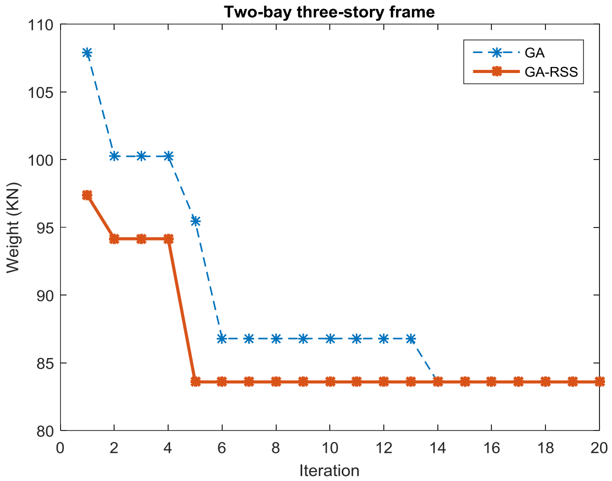


Figure 4. Comparison of the best-weight convergence curves of GA-RSS and standard GA obtained in the two-bay three-story frame problem

Table 1 compares the optimization results of this study with other results in the literature and reveals that the convergence speed has improved in the GA-RSS compared to other algorithms.

Table 1: Optimal design comparison for the 2-bay 3-story steel frame structure

Element group	Optimal W-shapes sections				
	Pezeshk et al.	Murren and Khandelwal	Mahallati et al.	Present study	
	GA [7]	DDHS [45]	IMOEA [25]	GA	GA-RSS
1	W24X62	W24X62	W24X62	W24X62	W24X62
2	W10X60	W10X60	W10X60	W10X60	W10X60
Weight (KN)	83.587	83.587	83.587	83.587	83.587
No. of required analyses	1800	270	250	492	195

5.2. One-bay ten-story frame

Fig. 5 shows the configuration and member grouping of a 1-bay 10-story 30-member frame. Beams are selected from among all 267 W-shaped sections, columns are limited to W12 and W14 sections (66 W-shaped). This frame is designed following the AISC-LRFD specification and uses inter-story drift constraints, the unbraced length for each beam member is specified as one-fifth of the span length [6, 8, 12]. E and F_y are the same as in 5.1, but the population size is 100.

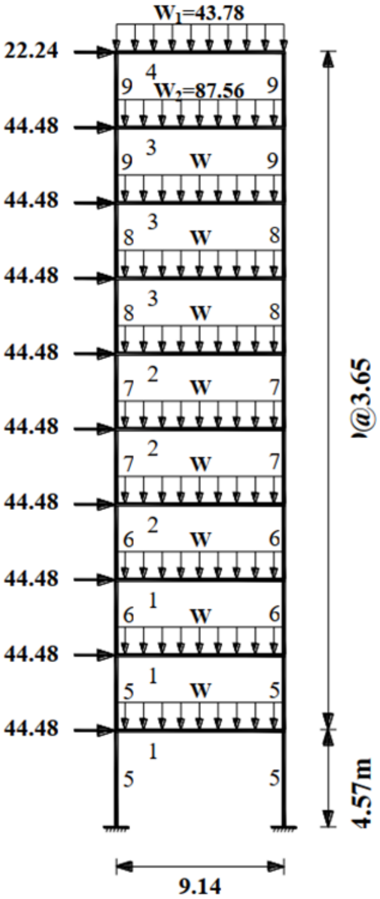


Figure 5. 1-bay 10-story steel frame structure

Fig. 6 shows the convergence history for the GA-RSS and standard GA. The latter has computed the optimum design to be 285.37 KN within 2300 frame analyses while the former has done it within 2190 analyses and yielded an optimum design of 281.72 KN. GA-RSS algorithm with a 4.78% reduction in the number of analyses caused as well a 1.28% improvement in the optimal solution. The average weight of the GA-RSS designs over the 10 independent runs was 287.34 KN, with a standard deviation of 2.85 KN while the average weight based on the standard GA was 292.41 KN, with a standard deviation of 12.35 KN.

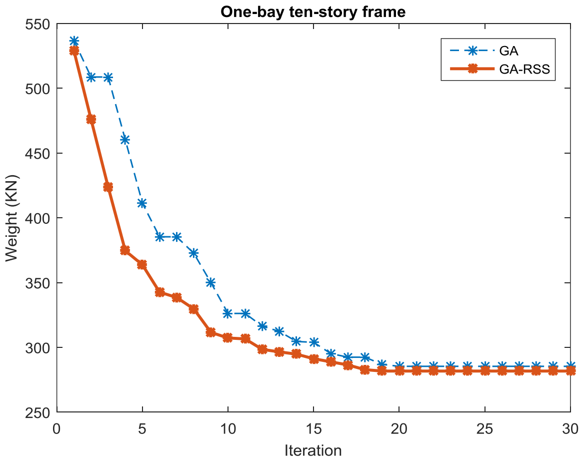


Figure 6. Comparison of the best-weight convergence curves of GA-RSS and standard GA obtained in the one-bay ten-story frame problem

Table 2 compares the optimization results of this study with those of other researches and reveals that the algorithm has found the optimal design with fewer analyses than the GA [7] and IACO [11] and lesser weight than GA [7] and GSU-PSO [5].

Table 2: Optimal design comparison for the 1-bay 10-story steel frame structure

Element group	Optimal W-shapes sections				
	Pezeshk et al.	Kaveh and Talatahari	Khajeh et al.	Present study	
	GA [7]	IACO [11]	GSU-PSO [5]	GA	GA-RSS
1	W33 × 118	W33 × 118	W33 × 118	W33 × 118	W33 × 118
2	W30 × 90	W30 × 90	W30 × 99	W30 × 99	W30 × 99
3	W27 × 84	W24 × 76	W27 × 84	W27 × 84	W27 × 84
4	W24 × 55	W14 × 30	W18 × 40	W16 × 45	W18 × 40
5	W14 × 233	W14 × 233	W14 × 233	W14 × 233	W14 × 233
6	W14 × 176	W14 × 176	W14 × 176	W12 × 190	W14 × 176
7	W14 × 159	W14 × 145	W14 × 145	W14 × 132	W14 × 132
8	W14 × 99	W14 × 90	W12 × 106	W14 × 99	W14 × 99
9	W12 × 79	W12 × 65	W12 × 65	W14 × 61	W14 × 61
Weight (KN)	289.72	274.99	287.18	285.37	281.72
No. of required analyses	3000	2500	1920	2300	2190

5.3. Three-bay fifteen-story frame

Fig. 7 shows a schematic view of the member grouping and loading of a 3-bay 15-story frame; here, the design constraints are the AISC combined strength constraints and displacement.

Beam and column element groups are selected from all 267 W-shaped sections of the AISC standard list [1, 2, 25] and E, F_y and population size are the same as in 5.2.

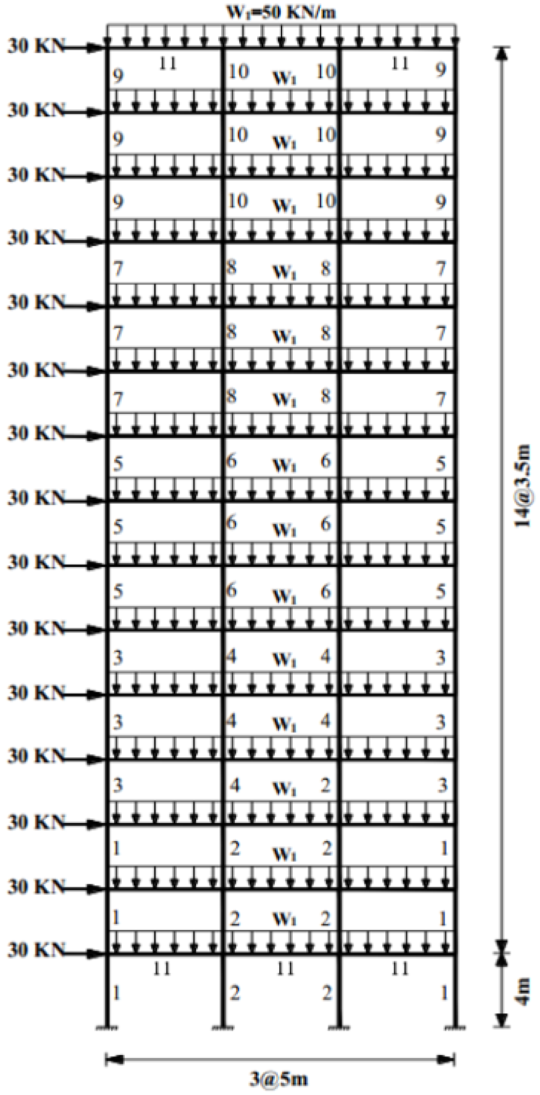


Figure 7. 1-bay 10-story steel frame structure

Fig. 8 shows the convergence history for the GA-RSS and standard GA. The latter has computed the optimum design to be 411.13 KN within 7250 frame analyses while the former has done it within 6150 analyses and yielded an optimum design of 405.33 KN. GA-RSS algorithm with a 15.17% reduction in the number of analyses caused a 1.41% improvement in the optimal solution. The average weight of the GA-RSS designs over the 10 independent runs was 426.89 KN, with a standard deviation of 14.29 KN while the average weight of the standard GA designs was 430.97 KN, with a standard deviation of

25.03 KN. Table 3 compares the optimum results gained from this study with those of other researches. The optimum design based on GA-RSS is 4.93% lighter than the optimum solution of HPSACO [1], 2.9% lighter than the optimum design of ICA [2] and 4.79% lighter than that of IMOEA [25].

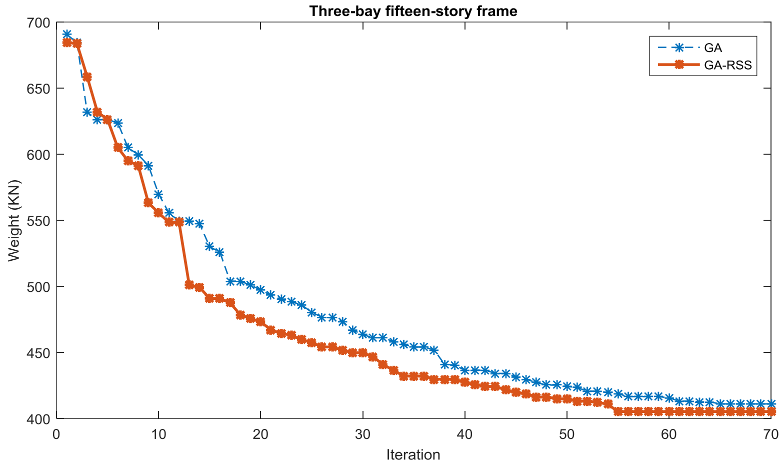


Figure 8. Comparison of the best-weight convergence curves of GA-RSS and standard GA obtained in the three-bay fifteen-story frame problem

Table 3: Optimal design comparison for the 3-bay 15-story steel frame structure

Element group	Optimal W-shapes sections				
	Kaveh and Talatahari	Kaveh and Talatahari	Mahallati et al.	Present study	
	HPSACO [1]	ICA [2]	IMOEA [25]	GA	GA-RSS
1	W21 × 111	W24 × 117	W33 × 118	W33 × 130	W33 × 118
2	W18 × 158	W21 × 147	W36 × 160	W24 × 146	W36 × 160
3	W10 × 88	W27 × 84	W18 × 86	W18 × 86	W14 × 90
4	W30 × 116	W27 × 114	W14 × 120	W33 × 118	W24 × 104
5	W21 × 83	W14 × 74	W21 × 68	W18 × 76	W24 × 76
6	W24 × 103	W18 × 86	W30 × 90	W30 × 90	W18 × 86
7	W21 × 55	W12 × 96	W10 × 60	W21 × 48	W14 × 48
8	W26 × 114	W24 × 68	W24 × 68	W21 × 166	W12 × 58
9	W10 × 33	W10 × 39	W10 × 33	W14 × 34	W14 × 30
10	W18 × 46	W12 × 40	W12 × 40	W10 × 39	W16 × 40
11	W21 × 44	W21 × 44	W21 × 50	W21 × 44	W21 × 44
Weight (KN)	426.36	417.46	425.72	411.13	405.34
No. of required analyses	6800	6000	6500	7250	6150

6. CONCLUSIONS

The GA is a random search algorithm that works based on the principle of the evolution of living things in the nature and uses such biological techniques as the crossover and mutation. It is often used to solve very complex and nonlinear problems, but despite all its benefits, its computation time for frame structure optimization problems is very lengthy.

This paper introduced a GA-based reduced search space (GA-RSS) technique to improve the speed of convergence and quality of the optimal solution of a moment frame problem. It creates a boundary region for each constraint to limit the search space and focus the search in this region. Solutions lying in these regions are considered better than the rest and lead the population towards the global optimal solution. To apply the design constraints to the optimization problem, use was made of the Extended Balanced Ranking Method (E-BRM) where the solutions were sorted based on the values of the objective and penalty functions and, hence, the potential of the infeasible solutions was used to find the feasible ones.

To examine the efficiency of the proposed algorithm, three frame design examples were solved and the numerical results were compared with some other metaheuristic algorithms concluding that the proposed algorithm could be justified as robust in finding reasonable solutions through significantly less analyses. The proposed technique, showed that still there is a possibility to reduce number of structural analyses required for optimization, compared to the results reported by others in the literature yet unveiling maybe a slightly modified optimum performance. Although the proposed method's main benefit is its search focus in the boundaries of the feasible space where the optimal solution is more probable, it may not be effective in cases where the optimal solution lies in the central regions of the search space.

REFERENCES

1. Kaveh A, Talatahari S. *Hybrid Algorithm of Harmony Search, Particle Swarm and Ant Colony for Structural Design Optimization*, in *Harmony Search Algorithms for Structural Design Optimization* 2009, Springer, pp. 59-198.
2. Kaveh A, Talatahari S. Optimum design of skeletal structures using imperialist competitive algorithm, *Comput Struct* 2010; **88**(21-22): 1220-9.
3. Talatahari S, Gandomi AH, Yang XSh, Deb S. Optimum design of frame structures using the Eagle Strategy with Differential Evolution, *Eng Struct* 2015; **91**: 16-25.
4. Ghasemi MR, Hinton E, Bulman S. *Performance of Genetic Algorithms for Optimization of Frame Structures*, London, Springer London, 1998.
5. Khajeh A, Ghasemi M, Ghohani Arab H. Hybrid particle swarm optimization, grid search method and univariate method to optimally design steel frame structures, *Int J Optim Civil Eng* 2017; **7**(2): 173-91.
6. Kaveh A, Abdietehrani A. Design of frames using genetic algorithm, force method and graph theory, *Int J Numer Meth Eng* 2004; **61**(14): 2555-65.
7. Pezeshk S, Camp C, Chen D. Design of nonlinear framed structures using genetic optimization, *J Struct Eng* 2000; **126**(3): 382-8.

8. Safari D, Maheri MR, Maheri A. Optimum design of steel frames using a multiple-deme GA with improved reproduction operators, *J Construct Steel Res* 2011; **67**(8): 1232-43.
9. Camp C, Pezeshk S, Cao G. Optimized design of two-dimensional structures using a genetic algorithm, *J Struct Eng* 1998; **124**(5): 551-9.
10. Camp CV, Bichon BJ, Stovall SP. Design of steel frames using ant colony optimization, *J Struct Eng* 2005; **131**(3): 369-79.
11. Kaveh A, Talatahari S. An improved ant colony optimization for the design of planar steel frames, *Eng Struct* 2010; **32**(3): 864-73.
12. Doğan E, Saka MP. Optimum design of unbraced steel frames to LRFD–AISC using particle swarm optimization, *Adv Eng Soft* 2012; **46**(1): 27-34.
13. Degertekin SO. Optimum design of steel frames using harmony search algorithm, *Struct Multidisc Optim* 2008; **36**(4): 393-401.
14. Maheri MR, Narimani M. An enhanced harmony search algorithm for optimum design of side sway steel frames, *Comput Struct* 2014; **136**: 78-89.
15. Hasançebi O, Erdal F, Saka MP. Adaptive harmony search method for structural optimization, *J Struct Eng* 2010; **136**(4): 419-31.
16. Toğan V. Design of planar steel frames using teaching–learning based optimization, *Eng Struct* 2012; **34**: 225-32.
17. Kaveh A, Talatahari S. Size optimization of space trusses using Big Bang–Big Crunch algorithm, *Comput Struct* 2009; **87**(17): 1129-40.
18. Kaveh A, Biabani Hamedani K, Zaerreza A. A set theoretical shuffled shepherd optimization algorithm for optimal design of cantilever retaining wall structures, *Eng Comput* 2020.
19. Kaveh A, Khanzadi M, Rastegar Moghaddam M. Billiards-inspired optimization algorithm; a new meta-heuristic method, *Struct* 2020; **27**: 1722-39.
20. Kripakaran P, Hall B, Gupta A. A genetic algorithm for design of moment-resisting steel frames, *Struct Multidisc Optim* 2011; **44**(4): 559-74.
21. Kaveh A, Zakian P. Optimal design of steel frames under seismic loading using two meta-heuristic algorithms, *J Construct Steel Res* 2013; **82**: 111-30.
22. Kaveh A, Bakhshpoori T. Optimum design of steel frames using Cuckoo Search algorithm with Lévy flights, *Struct Des Tall Special Build* 2013; **22**(13): 1023-36.
23. Flager F, Soremekun G, Adya A, Shea K, Haymaker J, Fischer M. Fully Constrained Design: A general and scalable method for discrete member sizing optimization of steel truss structures, *Comput Struct* 2014; **140**: 55-65.
24. Kaveh A, Fahimi-Farzam M, Kalateh-Ahani M. Optimum design of steel frame structures considering construction cost and seismic damage, *Smart Struct Syst* 2015; **16**(1): 1-26.
25. Mahallati Rayeni A, Ghohani Arab H, Ghasemi M. Optimization of steel moment frame by a proposed evolutionary algorithm, *Int J Optim Civil Eng* 2018; **8**(4): 511-24.
26. Kaveh A, Ghazaan MI. Optimum seismic design of 3D irregular steel frames using recently developed metaheuristic algorithms, *J Comput Civil Eng* 2018; **32**(3): 04018015.
27. Kaveh A, Biabani Hamedani K, Hosseini SM, Bakhshpoori T. Optimal design of planar steel frame structures utilizing meta-heuristic optimization algorithms. in Structures. 2020. Elsevier.

28. Toğan V, Daloğlu AT. An improved genetic algorithm with initial population strategy and self-adaptive member grouping, *Comput Struct* 2008; **86**(11-12): 1204-18.
29. Foley CM, Schinler D. Automated design of steel frames using advanced analysis and object-oriented evolutionary computation, *J Struct Eng* 2003; **129**(5): 648-60.
30. Construction AIOS. Load And Resistance Factor Design, 2001.
31. Dumonteil P. Simple equations for effective length factors, *Eng J AISC* 1992; **29**(3): 111-15.
32. Nanakorn P, Meesomklin K. An adaptive penalty function in genetic algorithms for structural design optimization, *Comput Struct* 2001; **79**(29-30): 2527-39.
33. Takahama T, Sakai S. Solving difficult constrained optimization problems by the ε constrained differential evolution with gradient-based mutation, in Constraint-Handling in Evolutionary Optimization, Springer, 2009, pp. 51-72.
34. Coello CAC. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, *Comput Method Appl Mech Eng* 2002; **191**(11-12): 1245-87.
35. Mezura-Montes E, Coello CAC. Constraint-handling in nature-inspired numerical optimization: past, present and future, *Swarm Evolut Computat* 2011; **1**(4): 173-94.
36. Jiao R, Zeng S, Li C. A feasible-ratio control technique for constrained optimization, *Inform Sci* 2019; **502**: 201-17.
37. de Castro Rodrigues M, Guimarães S, de Lima BSLP, *E-BRM:* A constraint handling technique to solve optimization problems with evolutionary algorithms, *Appl Soft Comput* 2018; **72**: 14-29.
38. Issa HK, Mohammad FA. Effect of mutation schemes on convergence to optimum design of steel frames, *J Construct Steel Res* 2010; **66**(7): 954-61.
39. Kazemzadeh Azad S, Jayant Kulkarni A. Structural optimization using a mutation-based genetic algorithm, Int J Optim Civil *Eng* 2012; **2**(1): 81-101.
40. Tang KZ, Sun TK, Yang JY. An improved genetic algorithm based on a novel selection strategy for nonlinear programming problems, *Comput Chemic Eng* 2011; **35**(4): 615-621.
41. Herrera F, Lozano M, Sánchez AM. A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study, *Int J Intell Syst* 2003; **18**(3): 309-38.
42. Sallam KM, Sarker RA, Essam DL. Reduced search space mechanism for solving constrained optimization problems, *Eng Applicat Artifi Intell* 2017; **65**: 147-58.
43. Schoenauer M, Michalewicz Z. Boundary Operators for Constrained Parameter Optimization Problems, in ICGA, 1997.
44. Bonyadi MR, Michalewicz Z. On the edge of feasibility: a case study of the particle swarm optimizer, *in 2014 IEEE Congress on Evolutionary Computation (CEC)* 2014, IEEE.
45. Murren P, Khandelwal K. Design-driven harmony search (DDHS) in steel frame optimization, *Eng Struct* 2014; **59**: 798-808.