Arabian Journal for Science and Engineering (2022) 47:5003–5015
https://doi.org/10.1007/s13369-021-06338-4

RESEARCH ARTICLE-CIVIL ENGINEERING

Proposing a Novel Oriented Genetic Algorithm for Optimum Seismic Design of Steel Moment Resisting Frames

Mostafa Rastegaran[1] · Seyed Bahram Beheshti Aval[1] · Ehsan Noroozinejad Farsangi[2] · Ishagh Faryabi[1]

Received: 20 August 2021 / Accepted: 24 October 2021 / Published online: 27 November 2021

Abstract

This paper proposes a new variant of the genetic algorithm (GA), called the oriented genetic algorithm (OGA), for optimum seismic design of steel special moment resisting frames. Since GA is mainly based on random operators, its computational burden is usually high. To overcome this issue, OGA takes advantage of the violation values of design constraints of the problem, such as elements' demand-to-capacity ratios, strong column-weak beam requirements, and story drift ratios, to direct the search procedure. OGA is applied to a set of steel frames with different geometrical properties to demonstrate, and the results are compared to those of GA. The numerical results indicate that OGA significantly reduces the total number of function evaluations (NFE) required to obtain the optimum solutions. Also, the convergence history of OGA is compared with Particle Swarm Optimization and Ant Colony Optimization algorithms. It is shown that for a specified NFE limit, OGA gives better-optimized results.

Keywords Genetic algorithm · Optimization · Orienting operator · Steel special moment resisting frames · Seismic

1 Introduction

With every new idea, there is always an optimization problem, and engineers are constantly seeking approaches to optimize their work efficiency. Optimization is basically defined as changing the initial concept and improving it using available information. Optimum seismic design of structures having minimum weight while satisfying the required constraints is one of the major concerns in structural engineering.

The traditional gradient-based optimization methods possess some disadvantages, such as getting trapped in local optima, high computational complexity, and sensitivity to the initial design point when solving the most practical optimization problems [1–3]. To cope with these issues, different meta-heuristic algorithms have been developed by inspiring natural phenomena and physical laws in recent decades. Meta-heuristic algorithms are simple to implement and do not rely on any gradient computations. Genetic algorithm (GA) [4], particle swarm optimization (PSO) [5], ant colony optimization (ACO) [6], bat algorithm (BA) [7], gray wolf algorithm (GWA) [8], and charged system search (CSS) [9] are some of the well-known meta-heuristic algorithms.

The application of meta-heuristic algorithms in dealing with the optimum seismic design of structures has received considerable attention in recent years. De Castro utilized GA to obtain the minimum weight design of frame structures [10]. Camp et al. [11, 12] solved the optimization problem of steel frame and low-weight steel frames using ACO. Morsali and Behnamfar [13] developed a method for seismic damaged-based optimized design of SMRF's using PSO. Kaveh and Bakhshpoori [14, 15] applied Cuckoo Search Algorithm (CSA) for optimum design of steel frames. Maheri and Narimani [16] developed an enhanced harmony search algorithm (EHS) to find the optimum design of side-sway steel frames. Kaveh et al. [17] performed optimum seismic design of steel frames using the simplified dolphin echolocation algorithm. Farshchin et al. [18] employed school-based optimization (SBO) to carry out an optimum design of steel frames. Gholizadeh et al. [19] proposed improved black hole and multiverse algorithms to solve the optimization problem of planar frame structures. Togan [20] employed teaching–learning based optimization (TLBO) to obtain the minimum weight of planar steel frames according

✉ Ehsan Noroozinejad Farsangi
noroozinejad@kgut.ac.ir

[1] Faculty of Civil Engineering, K.N. Toosi University of Technology, Tehran, Tehran, Iran

[2] Faculty of Civil and Surveying Engineering, Graduate University of Advanced Technology, Kerman, Iran

to AISC-LRFD specifications. The feature-sign search algorithm is used to reduce the uncertainty of idealized boundary conditions in the identification of structural damage [21]. Hasancebi et al. [22] utilized an evolution strategy for optimum seismic design of large steel space frame structures according to ASD-AISC requirements. A combination of PSO and artificial neural network is used to find optimized parameters of the tunneling process according to monitoring surface settlement [23]. Mangal and Cheng [24] developed GA-HJ hybrid approach by integrating Hooke and Jeeves (HJ) based local search algorithm in GA and utilized it in optimization of steel reinforcement in RC building frames considering constructability limitations.

Although meta-heuristic algorithms are robust tools in dealing with the optimization problem of frame structures, they induce a high computational burden. This drawback is mainly attributed to the random stochastic nature of solution updating rules utilized in the meta-heuristic algorithm. Similarly, GA, due to its inherent randomness in finding the optimum solutions, usually requires significant computational efforts with substantial NFEs to achieve the optimum solutions, especially in dealing with the optimum seismic design of structures that contain a high number of design variables.

In the present study, a new variant of GA, termed oriented GA (OGA), is proposed to efficiently tackle the optimum seismic design of steel special moment resisting frames (SMRFs). OGA uses a novel operator named orienting operator, which directs the algorithm to an optimum solution. GA and OGA are applied to perform the optimum seismic design of different SMRFs, and comparisons are made between the results.

The present manuscript is organized as follows: The optimization problem, including a definition of the objective function and intended design constraints and also the method of handling constraints, are described in Sect. 2. Fundamental concepts of genetic algorithm and its operators and the presentation of the novel orienting operator, and how to consider it in GA are described in Sect. 3. In Sect. 4, the application of the OGA on 2- to 5-story SMRFs is demonstrated. In this section, the optimal values of GA parameters are determined by parameter tuning, and then the performance of OGA and GA with optimal parameters are compared. Finally, in Sect. 5, the conclusions and limitations are drawn, and suggestions for future studies are presented.

2 Problem Formulation

Structural optimization aims to find the design variables so that some specific objectives such as weight, construction costs, life-cycle costs, repair costs after an earthquake, and specific performance objectives of the structure to be optimized minimized; subject to certain design constraints. Considering weight as the objective function, the problem formulation can be expressed as follows [8]:

$$\begin{aligned}&\text{Find}: X=\left\{x_1,\ldots,x_i,\ldots,x_{n_{DV}}\right\}^T\\&\text{To minimize}: W(X)=\sum_{i=1}^{n_e}\rho_i L_i A_i\\&\text{Subject to}: g_k(X)\le 0, k=1,2,\ldots,n_{cons}\end{aligned}\tag{1}$$

Where design variable x_i is steel section assigned to the element i, n_{DV} is the number of design variables, n_e is the total number of beam and column elements, $W(X)$ indicates the total weight of the structure, ρ_i represent the material density of the element i, L_i is the length of element i, A_i the cross-sectional area of element i, $g_k(X)$ is the k'th design constraint, and n_{cons} is the total number of design constraints.

2.1 Design Constraints

As stated previously, the present study deals with the optimum seismic design of SMRFs. In this respect, design constraints taken into account are present as follows:

Strength constraint: According to the AISC 360–10 [25], demand-to-capacity ratio constraint should be satisfied for each structural element as follows:

$$g_{1,i}=\begin{cases}\frac{P_r}{P_c}+\left(\frac{M_{rx}}{M_{cx}}+\frac{M_{ry}}{M_{cy}}\right)-1 \text{ for } \frac{P_r}{P_c}<0.2\\ \frac{P_r}{P_c}+\frac{8}{9}\left(\frac{M_{rx}}{M_{cx}}+\frac{M_{ry}}{M_{cy}}\right)-1 \text{ for } \frac{P_r}{P_c}\ge 0.2\end{cases}, i=1,2,\ldots,n_e\tag{2}$$

where P_r is the required axial strength, P_c is the available axial strength, M_r is the required flexural strength, M_c is the available flexural strength, and n_e is the total number of elements.

Inter-story drift constraint: According to ASCE07-10 [26], inter-story drift constrain is applied as follows:

$$g_{2,i}=\frac{d_i}{d_{allow,i}}-1,\quad i=1,2,..,n_s\tag{3}$$

where d_i is the inter-story drift of floor i, $d_{allow,i}$ is the allowable inteory drift of floor i specified by the code, and n_s is the total number of stories.

Strong column-weak beam criterion: To avoid the formation of plastic hinges in the columns that may give rise to structural instability, the strong column-weak beam criterion as in AISC 341–10 [27] should be satisfied for each beam-to-column connection of the SMRF as follows:

$$g_{3,i}=\frac{\sum M^*_{pb}}{\sum M^*_{pc}}-1,\quad i=1,2,\ldots,n_{con}\tag{4}$$

where $\sum M^*_{pb}$ is the sum of the projections of the expected flexural strengths of the beams at the plastic hinge locations to the column centerline, $\sum M^*_{pc}$ is the sum of the projections of the nominal flexural strengths of the columns above and below the joint to the beam centerline with a reduction for the axial force in the column, and n_{con} is the total number of beam-to-column connections.

Width-to-thickness limitations: According to AISC 341–10 [27], to prevent local buckling before the occurrence of yielding in elements under compression, the width-to-thickness ratios for the flanges and webs of W-sections are limited as follows:

$$g_{4,i} = \frac{b_f/t_f}{0.6\sqrt{E/F_y}} - 1, \quad i = 1, 2, \ldots, n_e \tag{5}$$

$$g_{5,i} = \begin{cases} \frac{h/t_w}{2.45\sqrt{E/F_y}(1-0.93C_a)} - 1 & for\ C_a \le 0.125 \\ \frac{h/t_w}{max\left(0.77\sqrt{E/F_y}(2.93-C_a), 1.49\sqrt{E/F_y}\right)} - 1 & for\ C_a > 0.125 \end{cases} \quad i = 1, 2, \ldots, n_e \tag{6}$$

where b_f is the flange width, t_f is the thickness of the flange, E denotes the modulus of elasticity, F_y indicates the specified yield stress of the steel, h is the web depth, t_w is the web thickness, n_e is the total number of elements, and C_a is defined as follows:

$$C_a = \frac{P_u}{\phi_c P_y} \tag{7}$$

where P_u is the required axial strength, ϕ_c is the Resistance factor for compression, and P_y is the Nominal axial yield strength of a member.

Geometric constraints: In practical designs, the beam flange width should be smaller or equal to the column flange width at each beam-to-column connection. Also, the ctiodth of the upper-story column should be equal to or smaller than that of the lower-story coln at each column-to-column connection. Thus, two geometric constraints should be satisfied as follows:

$$g_{6,i} = \frac{b_{fb}}{b_{fc}} - 1, \quad i = 1, 2, \ldots, n_{cb} \tag{8}$$

$$g_{7,i} = \frac{d_{c,upper}}{d_{c,lower}} - 1, \quad i = 1, 2, \ldots, n_{cc} \tag{9}$$

where b_{fb} and b_{fc} are the beam flange width and column flange width, respectively, n_{cb} is the number of beam-to-column connections, $d_{c,upper}$ and $d_{c,lower}$ are the depth of the upper and lower columns, respectively, and n_{cc} is the number of column-to-column conneions.

The control of geometric constraints does not need any structural analysis. Geometric constraints are first evaluated to prevent excessive structural analysis. If geometric constints are not satisfied, the other constraints are not evaluated, and the structure is removed from other steps of the optimization algorithm.

2.2 Constraint Handling

As GA is used to optimize unconstrained problems, the constrained problem should be converted to an unconstrained problem by an appropriate constraint handling approach. Two approaches may be applied [28]: (a) Death Penalty Approach in which members of the population that have not satisfied constraints (infeasible designs) are completely removed from the population, and no information is extracted from the infeasible population. (b) Penalty Function Approach where the objective function is penalized proportionally to the degree of unsatisfied constraints. The Death Penalty Approach has some disadvantages. First, problems in which design space is strictly under the control of constraints, production of a generation in which all of its members satisfy all the constraints requires very high computational costs [28]. Moreover, because of the possibility of approaching optimum value from any direction, usually search in both feasible and infeasible design spaces is more efficient than restricting the search to just feasible design space [28]. Considering the disadvantages of the death penalty method, the penalty function approach is used here to dealithints. The penalty function is considered as follows:

$$\begin{aligned} &\mathrm{F = W + V} \\ &\mathrm{W} = \sum_{i=1}^{n_e} \mathrm{w_i L_i} \\ &\mathrm{V} = \sum_{k=1}^{n_e} \mathrm{v_{1,k}} + \sum_{k=1}^{n_e} \mathrm{v_{2,k}} + \sum_{k=1}^{n_e} \mathrm{v_{3,k}} + \ldots \\ &\quad = \sum_{j=1}^{n_{cons}} \sum_{k=1}^{n_e} \mathrm{v_{j,k}} \\ &\mathrm{v_{j,k}} = \begin{Bmatrix} 0 & if\ \mathrm{c_{j,k}} \ge 0 \\ \left|\mathrm{c_{j,k}}\right| \times \alpha & if\ \mathrm{c_{j,k}} < 0 \end{Bmatrix} \\ &\alpha = \mathrm{n_e} \times \mathrm{mean_{Pool}} \times \mathrm{R} \end{aligned} \tag{10}$$

where n_e is the total number of beam and column elements, w_i represents the weight of the element i per unit length, n_{cons} is the number of constraints, $v_{j,k}$ is the violation of constraint j in element k, $c_{j,k}$ is the value of constraint j in element k,

α is the penalty multiplier, $mean_{Pool}$ is the mean weight of variables pool (mean weight per unit length of selectable sections), and R is the penalty constant determined by the designer (here is assumed 1000).

3 Oriented GA

In the present section, GA is first introduced, and then the oriented GA (OGA) is presented in detail.

3.1 Fundamentals of Genetic Algorithm

GA is an evolutionary algorithm that aims to find the global optimum solution by using a population of potential solutions and their combinations. GA also searches the entire design space in multiple directions to prevent trapping in local minima and tries to achieve a global solution [29, 30]. The search process in GA is explained as follows:

1. Initial population: Design variables (genes) are selected randomly from the search space (e.g., W-sections in the design of steel structures). Putting the genes together, a member of the population (chromosome) is formed. Then this process is repeated n_{pop} times to form an initial population with n_{pop} random members.

2. Fitness Evaluation: All population members are analyzed at this stage, and the objective function values are obtained. Considering design constraints, a fitness value (value of the penalized objective function) is allocated to each population member. In this way, the population can be sorted and classified based on members' fitness.

3. Selection of appropriate members to produce next generation: Following the evaluation of members' fitness in the previous stage, some members (parents) should be selected to produce the next generation. Researchers have suggested different strategies to select parents. Usually, members' selection chance/probability for production of the next generation has a direct relationship with members' fitness. In other words, members with better fitness have a much higher chance of being selected. Different methods such as Roulette-Wheel, Uniformly Random, Tournament selection, etc., can be used to select parents for next-generation operation [1]. In the present study, Roulette-Wheel, by which parents with better features have a greater chance of being selected, is chosen.

4. *Crossover:* n_{cross} number of parents are selected for crossover, which n_{cross} is determined by the designer through crossover percentage parameter (p_{cross}):

$$n_{cross} = 2*\text{round}\left(p_{cross} \times n_{pop}/2\right) \tag{11}$$

where n_{cross} is the number of selected parents for crossover, and p_{cross} is the crossover percentage.

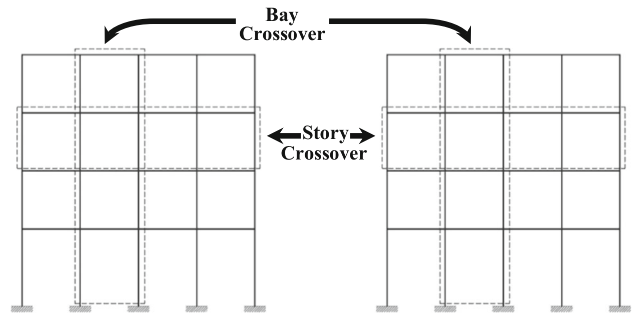

Fig. 1 Schematic representation of related bay-story crossover

Selected members are randomly paired together to share some of their genes, to generate n_{cross} new children for the next generation. There are different crossover methods: single point, double point, uniform, mixed-random, and related bay-story. The operation of a single point, double point, and uniform crossover operators are described in Kaya [31] and Hasançebi [32]. Mixed-random crossover randomly selects (with uniform distribution) one of the single point, double point, or uniform crossover in each step where crossover is required.

The "related bay-story" crossover, called "boosted or object homologous geometric" crossover by [33, 34], selects randomly one bay and one story from the first parent and substitutes corresponding elements in the second parent (Fig. 1). In all kinds of crossover operations, care should be taken to avoid substituting beams with columns or vice versa. For instance, if the two following parents (for a 3story-3bay structure) are considered, double point crossover acts as Fig. 2.

5. *Mutation*: m_u genes of each of n_m chromosomes (n_m of members selected for mutation, in the way described in stage 3) are changed randomly, and new n_m children are produced. For mutation operation, two parameters should be determined by the designer: mutation percentage (p_m) and mutation rate (m_r) mutation percentage determines the number of parents for mutation, and mutation rate determines the number of genes (m_u) that change in a chromosome. For example, if the mutation rate = 20% in a 3story-2bay frame structure, two columns of the selected parent ($\text{round}[20\% \times n_{col}] = \text{round}[20\% \times 9] = 2$) and one beam of the selected parent ($\text{round}[20\% \times n_{beam}] = \text{round}[20\% \times 6] = 1$) alter.

6. *Termination*: New generated members (from crossover and mutation) are analyzed, and their merits are evaluated. Objective function values and degree of constraints violation are criteria of merits for generated members. Children with higher merit would substitute weaker parents. Stages 3 to 6 are repeated until the convergence condition is satisfied or the algorithm reaches the predetermined number of iterations. The values of n_{pop}, n_{cross}, n_m, and m_u are determined by the designer.

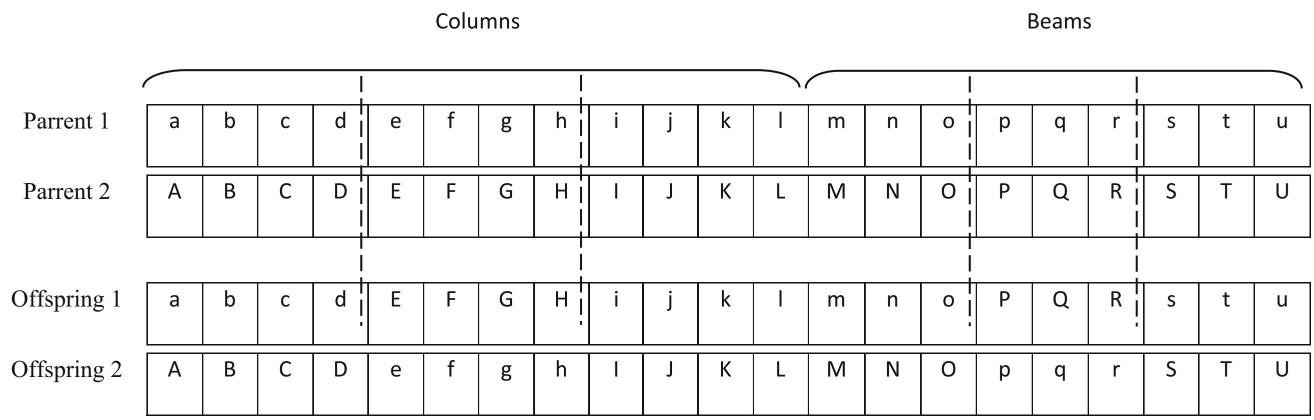

Fig. 2 Sample of double point crossover for a 3story-3bay structure

In GA, the crossover is responsible for approaching the local optimum solutions and mutation searches the entire design space to overcome trapping in local optimums [35].

3.2 Oriented Genetic Algorithm (OGA)

As shown in the *Numerical results* section, NFE in GA is relatively high and in practice, GA's performance and efficiency diminish when applied to the large-scale structures. Moreover, due to the random nature of the offsprings generation process in GA, many researchers such as Azad and Hasançebi [36] and Gen and Cheng [37] named it as a "blind algorithm." Various attempts have been made to improve the performance of this algorithm. Most of these efforts have focused on changing operators, which in pieces of the literature such as Gero et al. Gero, García and del Coz Díaz [38], Shi et al. Shi, Liang, Lee, Lu and Wang [39], and Toğan and Daloğlu [40] have been named as modified, improved, enhanced, intelligent, etc.

In this study, by adding an operator called "Orienting" to GA, the efficiency of the optimization process is considerably increased. In generating offsprings by crossover and mutation operators, genes change randomly; thus, generated offsprings can gain good or poor properties of their parents. Usually, in two statuses, the number of generated offsprings with worse properties would be higher than the number of offsprings with better properties: 1) when the number of design variables is large; 2) when early stages of the optimization process have passed. Thus, it is intended that with some changes in GA, the properties of offsprings be improved compared to their parents; as a result, better generations (generations with more optimized objective function and less violated constraints) and solutions would be achieved over time.

The number of parents selected for orienting operator is affected by two designer-determined parameters. These parameters are "Orienting Percentage" (O_p) and "Orienting Diversity Rate" (O_r).

To select parents for orienting, at first, members with at least $O_{div} = round(n_e \times O_r)$ different genes are selected. These selected members are sorted based on their fitness, and following that, the $n_{Ori,max} = O_p \times n_{pop}$ first best chromosomes are chosen for orienting; then Orienting operator is applied as *Orienting* section.

To evaluate the necessity of applying "Diversity Rate": At first, a structure (chromosome) including few members (genes) that do not satisfy the constraints is chosen (initial structure). Next, some other structures (chromosomes) are created with random change in a few genes of the initial structure; in this way, the initial population is built from members with relatively similar genes. Then Orienting operator is applied to this initial population, and the resulted population is recorded after 20 iterations of Orienting. The mentioned operation was repeated several times and applied to multiple different initial structures. Comparison of the resulted population after 20 iterations (with a similar initial population) shows that in more than 73% of analyses with different initial structures; the obtained population after 20 iterations converges to a unique chromosome (in other words, more than 73% of the resulted population with similar initial structure converges to a unique chromosome). Therefore, given that similar parents usually orient to a unique solution; if similar parents are selected, then extra calculations are made. Thus, with the application of diversity rate, different parents are selected for Orienting operation.

Orienting is applied as follows: based on the designer's opinion, the designer sorts constraints according to their effect on the value of the objective function. Constraints with more effect on objective function value are checked firstly, and subsequently, constraints with lower effect are checked. Based on the authors' experiences, using constraints out of proper order only affects the performance of the Orienting operator; and if the designer has not sorted the constraints according to their effect, NFE increases.

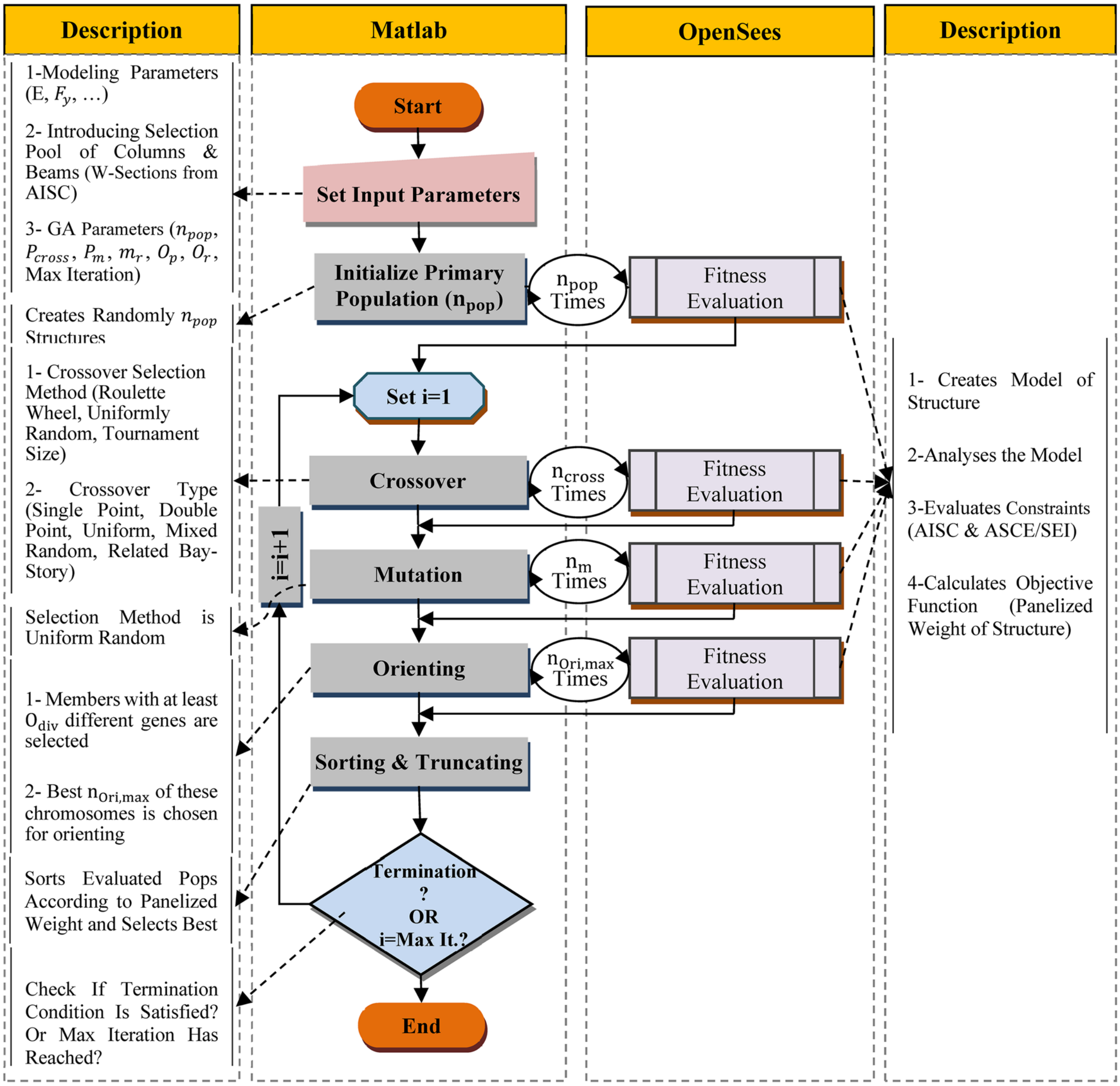

Fig. 3 Genetic algorithm optimization process

Parents are selected, as stated earlier, and "Orienting" modifies them as follows:

(a) If the selected parent has members (genes) with dissatisfied first constraint (capacity ratio): for dissatisfied genes, "Orienting" operator substitutes a section randomly with equal or higher characteristic (for beams higher moment of inertia and columns larger section area) than the current gene's section. Thus, generated offsprings might have a similar or heavier weight than their parents; but because of probable satisfaction of its first constraints, they may have a better merit function.

(b) In the case of satisfaction of the first constraint for all members and dissatisfaction of the second constraint (drift ratio), a section with the same or larger area than the current section is randomly dedicated to columns of the relevant story.

(c) If both the first and second constraints of all members are satisfied and the third constraint (strong column-weak beam principle) is not satisfied for some joints of structure, one of the three following cases is selected randomly:

(c.1) left or right beam of the dissatisfied joint is selected randomly.
(c.2) both left and right beams of the dissatisfied joint are selected.
(c.3) columns corresponding to the dissatisfied joint are selected.

Note: If c.1 or c.2 is selected, a section with the same or lower moment of inertia than the current section is assigned to beam from variables pool; and if c.3 is selected, a section with the same or higher area of the current section is dedicated to columns of the relevant story.

Table 1 Specifications of Analyzed SMRFs

Frame ID	No. of stories	No. of bays	First story height	Others stories height	Bay Width (from left to right)	No. of design search space
1	2	2	4	3	5, 5	1.05E + 06
2	3	3	4	3	5, 6, 5	3.66E + 15
3	4	4	4	3	5, 6, 6, 5	3.83E + 21
4	5	5	4	3	5, 6, 6, 6, 5	3.33E + 23

Table 2 Material Properties

Elastic Modulus (E)	1.99e11	N/m^2
Yield Stress (F_y)	3.44e8	N/m^2
Effective Yield Stress (F_{ye})	3.79e8	N/m^2
Tensile Strength (F_u)	4.48e8	N/m^2
Effective Tensile Stress (F_{ue})	4.92e8	N/m^2
Shear Modulus (G)	77.2e9	N/m^2

Table 3 Selection pool for beams and columns

Items	No. of sections	Sections selection pool
Columns	20	W5X19, W6X12, W6X25, W8X15, W8X28, W8X40, W8X67, W10X100, W12X45, W14X53, W14X82, W14X311, W14X455, W14X730, W18X119, W18X143, W18X175, W18X211, W24X250, W40X372
Beams	15	W5X16, W6X16, W8X15, W8X21, W8X40, W8X58, W10X112, W12X50, W12X136, W18X106, W21X132, W24X162, W24X250, W24X370, W27X539

(d) If all three constraints of all members are satisfied, and the fourth constraint is not satisfied, then a section with a higher moment of inertia than the current section is assigned to dissatisfied members from the variables pool.

(e) If all four constraints of all members are satisfied, the same or smaller section property (moment of inertia for beams and area for columns) is dedicated to all members, randomly.

Eventually, generated offsprings from "Orienting" operation are merged with other population of the current generation (parents, crossover & mutation offsprings) and then sorting & truncating is applied.

Population is sorted base on their fitness value, and the first n_{pop} are selected, and the rest are set aside.

The summary of described algorithm is shown in Fig. 3.

4 Numerical Results

In this section, by running GA for different values of affecting parameters and considering convergence rate, parameter tuning is carried out and optimum values for GA parameters are suggested. Then OGA is applied to several SMRFs. Finally, a comparison has been made between this study's results and some previous studies to point out the superiority of OGA and tuned GA.

4.1 Steel Frames

To determine optimum values of GA parameters ($n_{pop}, p_{cross}, p_m, m_r$ and crossover mode), GA is applied to some SMRFs (Table 1).

A uniform dead and live load of $6.0\,kN/m^2$ and $2.0\,kN/m^2$ was applied on all floors, respectively. A live load of 1.5 kN/m^2 was applied on roofs. The tributary width of the uniform loads on frames was considered 5 m. For all beams and columns, material properties are set as Table 2.

Beam and columns are modeled as "*Elastic Beam-Column Element*" in OpenSees. Sections of all members can be selected from a list of AISC W-Sections; 15 and 20 selectable sections for beams and columns have been considered here, respectively (Table 3).

Considering the symmetry of frames and the number of selectable beams and columns (Table 3), a total number of searches in design space is indicated in Table 1.

4.2 Grouping

Regarding construction conditions, members of beams and columns can be grouped, and for each group, a section would be assigned. Changing the grouping of beams and columns causes different results [41]. Since the determination of optimum parameters of GA is one of the goals of this research, for generality, no grouping is considered for beams and columns.

It should be noted that in the case of grouping, NFE decreases, and calculations speed up.

4.3 Determination of Optimal GA Parameters (Parameter Tuning)

Range of parameters; $n_{pop}, p_{cross}, p_m, m_r$ and crossover mode are effective parameters on GA. To obtain optimum

Table 4 Values of parameters for determination of optimum GA parameters

Parameters	Values (Modes)				
	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5
nPop Ratio (n_{pop}/n_{DV})	1	1.25	1.5	1.75	2
Crossover pct. (p_c)	0.5	0.6	0.7	0.8	0.9
Mutation pct. (p_m)	0.03	0.06	0.09	0.12	0.15
Mutation Rate (m_r)	0.01	0.05	0.10	0.15	0.20
Crossover Mode	Single Point	Double Point	Uniform	Mixed-Random	Related Bay-Story

values for these parameters, earlier described GA is run for different frames of Table 1 and parameter values of Table 4.

For shorthand and better perception, every parameter value is assigned to a mode; for example, in the case of "nPop Ratio," Mode 1 corresponds to $1 \times n_{DV}$, Mode 2 corresponds to $1.25 \times n_{DV}$, … and in the case of "crossover percentage," Mode 1 corresponds to 0.5, Mode 2 corresponds to 0.6, … and other parameters in the same way.

As it can be seen in Table 4, there are five effective parameters. For each parameter, five different modes are considered. Thus, for each frame, 5^5 different states are available. Due to the inherent randomness of GA, the algorithm is repeated 500 times for each state (totally $5^5 \times 500$ GA runs for each frame). *Optimum GA parameters.* Optimization results of frames mentioned in Table 1, with parameter values mentioned in Table 4 are shown in Fig. 4; In which the given data refers to the occurrence percentage of optimized weight with minimum NFE, for different modes of parameters.

For instance, in a 5story-5bay frame, the third mode of n_{pop} (namely $n_{pop} = 1.50 \times n_{DV}$) in 34.85% of runs has the minimum NFE to achieve optimum result among other modes of n_{pop} and the second mode of n_{pop} (namely $n_{pop} = 1.25 \times n_{DV}$) with 27.28% has the next minimum NFE to find the optimized result. Also, in this frame, the fourth mode of crossover percentage ($p_{cross} = 0.7$) with 28.25% has minimum NFE to achieve optimized results among all modes of parameters.

The summary of optimized modes for all parameters of analyzed frames is shown in Table 5.

As it can be seen in Table 5, for frames analyzed with n_{pop} between 1.25 ~ 1.5 n_{DV} and selection of crossovermode as "mixed random crossover" or "uniform crossover" and p_{cross} in the range of 0.7 ~ 0.8 and p_m in the range of 0.06 ~ 0.09 and m_r in the range of 0.05 ~ 0.10, optimized results can be obtained with minimum NFE.

Also applying various O_p and O_r values on frames of Table 1, $O_p = 0.2 \sim 0.3$ and $O_r = 0.15 \sim 0.20$ are suggested for the efficient performance of OGA.

As seen in Fig. 4, usually occurrence percentage of minimum weight with optimum parameter value (for example, fourth mode of crossover in 5story-5bay frame) has relatively little difference with the second optimum mode of the parameter (for example, the third mode of crossover in the 5story-5bay frame; namely %28.25-%25.67 = %2.58). Thus, choosing even the next optimized value of parameters has little effect on the required NFE for optimization.

4.4 Comparison of OGA with GA

To evaluate the efficiency of OGA, the NFE required to obtain optimized weight is compared. OGA and GA (with optimized parameter values) are applied on mentioned frames of Table 1, and the results are compared (Table 6). As it can be seen in Table 6, in the case of applying OGA, the average of NFE reduction to obtain optimum weight is at least 21.73% less than GA (even with using optimum values of GA parameters); and for a 5bay-5story frame, the reduction in NFE reaches 48.47%.

In Fig. 5, convergence history for some considered frames is illustrated. There are few iterations in 3bay-3story and 5bay-5story frames in which GA has a relatively better-penalized objective function than OGA. But despite those iterations, the trends in Fig. 5 make it clear that in all frames, OGA has an obvious superiority over GA (in the case of convergence rate). In other words, OGA optimizes the frame to a lower weight in a shorter time.

As stated in the section *Improving GA with Orienting Operator*, the most important reason for increasing efficiency, is the improvement of offsprings by the "Orienting" operator. Figure 6 compares improved offsprings (relative to their parents) produced by Orienting, crossover (single point, double point, uniform, mixed random), and mutation operators. It can be observed that more than 65% of Orienting offsprings and about 30% of mixed-random offsprings are better than their parents. As expected, mutation operator generates few offsprings better than their parents (only about 8%). Also, the mixed-random and uniform crossover operators have the highest percentage of improved offsprings between all utilized crossover operators.

In Fig. 7, the average number of the best and worst offsprings produced by each operator during all iterations of

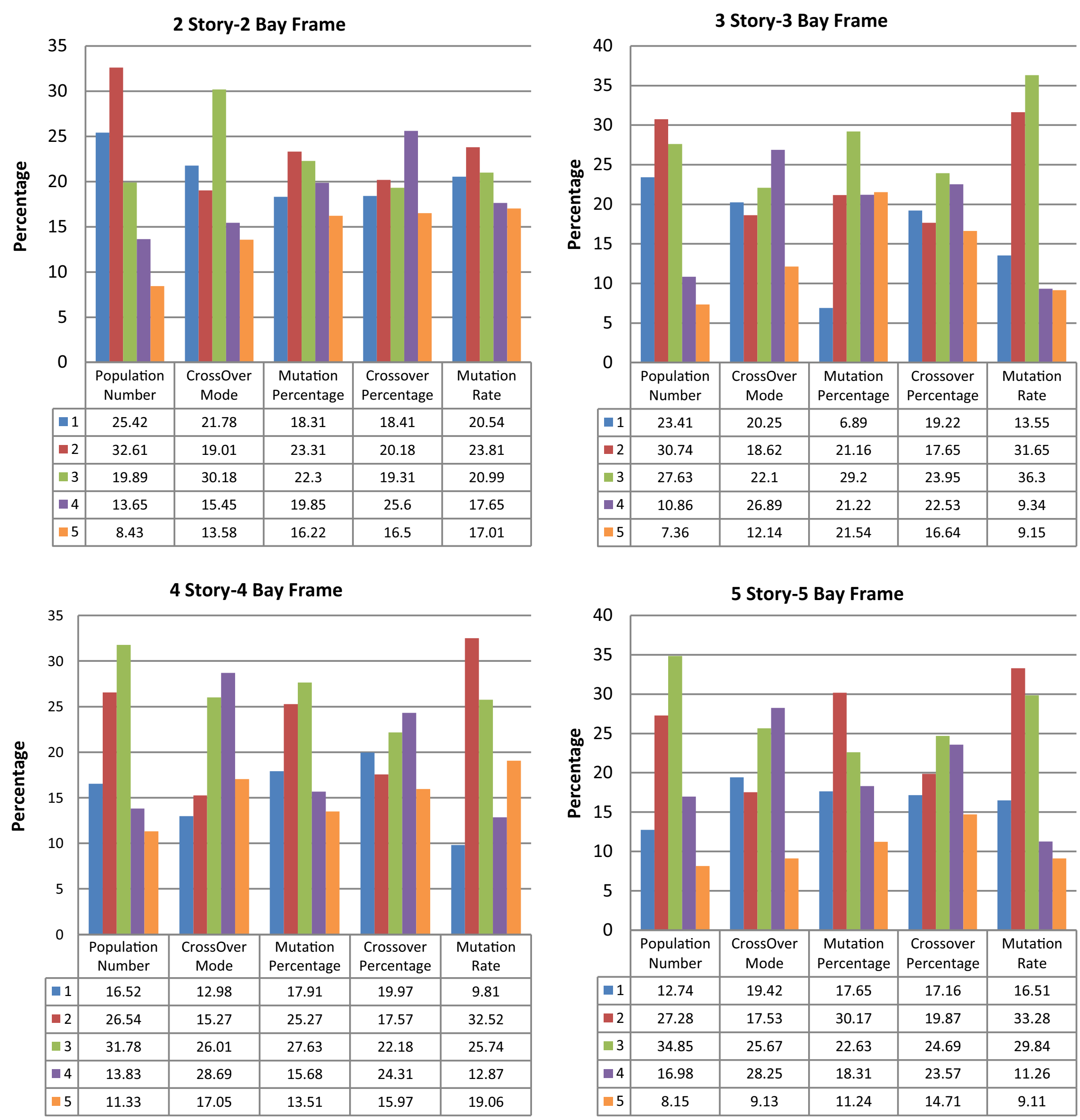

Fig. 4 Percentage of optimized frames, for different values of GA parameters

Table 5 Optimized modes of GA parameters (occurrence of minimum weight with least NFE)

Parameters	Frames			
	2*2	3*3	4*4	5*5
Population Number (n_{pop})	2	2	3	3
Crossover Mode	3	4	4	4
Mutation Percentage (p_m)	2	3	3	2
Crossover Percentage (p_{cross})	4	3	4	3
Mutation Rate (m_r)	2	3	2	2
	Dominant Mode			

several runs is presented. In each iteration, the origin of the best and worst offsprings are recorded; in another word, it is recorded that the best and worst offsprings in each iteration are created by which operator, and then at the end of each run, the number of roles of operators in creation of the best and worst offsprings are determined. This operation is repeated for different runs, and finally, the average number of roles of each operator in creating the best and worst offsprings is determined. As can be seen, a large proportion of offsprings of the Orienting operator have better features than their parents or are the same as their parents; while in other operators (especially mutation), a relatively large percentage of generated offsprings have worse features than their parents.

Considering the prominent role of Orienting operator in generating better offsprings, why are crossover and mutation operators not eliminated? Or why the role of Orienting oper-

Table 6 Comparison of NFE, in OGA and GA

Statistics	Frames							
	2*2		3*3		4*4		5*5	
	GA	OGA	GA	OGA	GA	OGA	GA	OGA
Minimum NFE	172	132	2,248	1,436	3,336	2,016	4,320	2,226
Average NFE	1,026	803	5,135	3,667	7,061	4,712	9,576	5,998
Standard Deviation	613	121	2,272	251	3,483	275	4,243	309
NFE Reduction (max)	23.26%		36.12%		39.57%		48.47%	
NFE Reduction (Ave)	21.73%		28.59%		33.27%		37.36%	

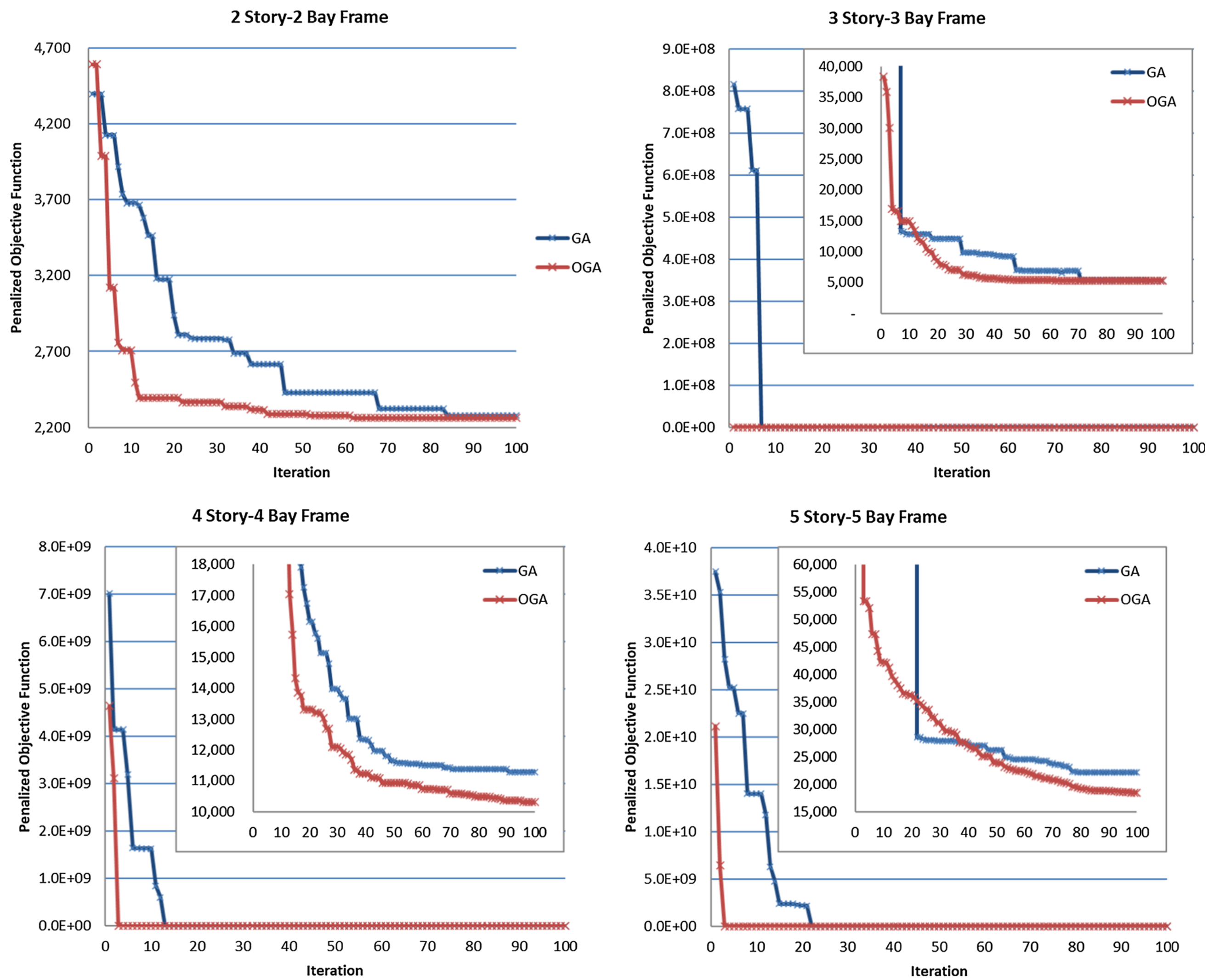

Fig. 5 Typical convergence history of considered frames

ator is not increased? To answer these questions, it should be noted that offsprings of crossover and mutation operators are necessary to prevent the algorithm from trapping in local optima. In the case of reducing these operators' role, the algorithm may be trapped in local minima before converging to global result, or the progress toward it may be extremely slow.

4.5 Comparison of OGA with PSO and ACO

A typical convergence history diagram of the 5bay-5story frame is shown in Fig. 8; in which the convergence rate of GA, OGA, ACO, and SPO algorithms are compared. As shown, the convergence rate of the OGA is much faster than the other algorithms. It should be noted that the OGA does not claim

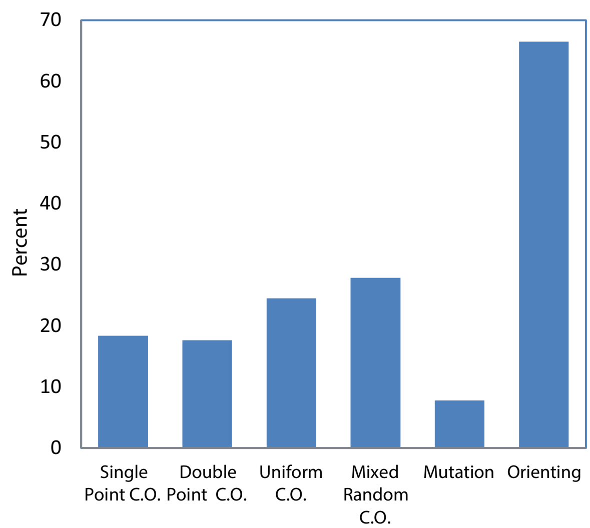

Fig. 6 Percentage of improved offsprings produced by different operators

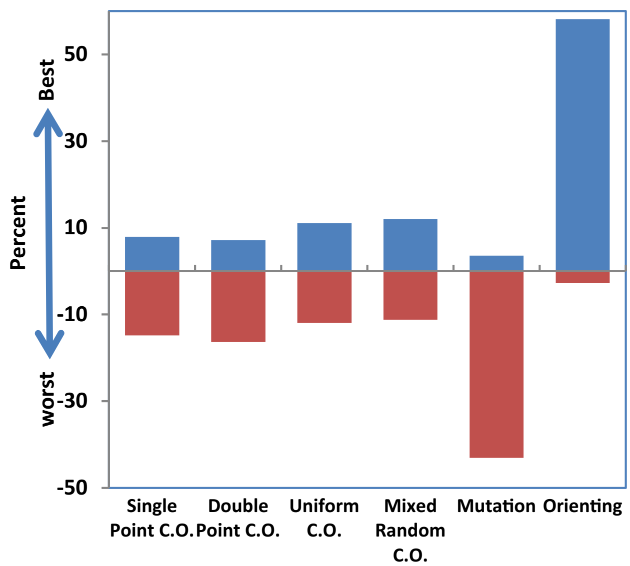

Fig. 7 Role of various operators in the generation of best and worst offspring

more final optimized results than other algorithms, but it is approved that for a given NFE limit, OGA achieves a more optimal result; for example, if the number of iterations is limited to 85 iterations, OGA usually results in more optimized results. But if there is no limit to NFE, other algorithms may give more optimal results.

5 Conclusion

As mentioned earlier, the purpose of this paper is to reduce the computational burden of optimization problems with an emphasis on SMRFs.

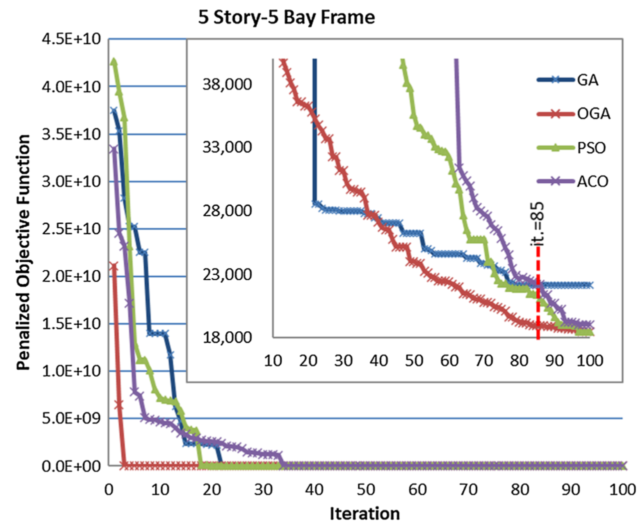

Fig. 8 Comparative typical convergence history of 5bay-5story frame for different optimization algorithms

In this paper, GA with various parameter values was applied to some SMRFs, and optimum values for parameters of GA were determined, aiming to reduce NFE. For analyzed frames, selection of n_{pop} in the range of 1.25 ~ 1.5 n_{DV}, crossovermode as "mixed random crossover" or "uniform crossover," p_{cross} in the range of 0.7 ~ 0.8, p_m in the range of 0.06 ~ 0.09 and m_r in the range of 0.05 ~ 0.10 were determined as optimum values for GA parameters.

It should further be noted that during case study investigations for this paper, among different kinds of crossover operators (single point, double point, uniform, mixed-random, and related bay-story), mixed-random and uniform crossover operators had better performance in GA and in more than 44% of optimized states with minimum NFE, these two kinds of crossover operators were utilized.

Due to the inherent blindness of crossover and especially mutation operators in generating better offsprings, a novel operator called "Orienting" was introduced. This operator has tried to generate better offsprings in each generation by adding a local search engine to the GA algorithm.

The presented novel operator has two affecting parameters: O_p and O_r; which their values are suggested in the range of $O_p = 0.2 \sim 0.3$ and $O_r = 0.15 \sim 0.20$. Considering that most offsprings of Orienting operator are better than their parents, GA's efficiency is highly increased and NFE is reduced considerably. In the case of applying OGA, the average NFE to obtain minimum weight is at least 21.73% less than that of GA (even with using optimum values of GA parameters); even in the 5bay-5story frame, this average efficiency increases to 37.36%.

Despite the significant improvement of GA through adding Orienting operator, it should be noted that elimination or reduction in the role of crossover and mutation operators or excessive increase in the role of Orienting operator, not only reduces the efficiency of GA and increases the probability of trapping but also increases NFE. Thus, Orienting operator beside other operators (crossover and mutation) makes GA a more efficient algorithm.

Also, in comparison of OGA with PSO and ACO, it is shown that OGA gives more optimized results when the NFE limit is considered. It should be noted that the OGA does not claim more final optimized results than other algorithms, but it is approved that for a given NFE limit, OGA achieves a more optimal result; But if there is no limit to NFE, other algorithms may give more optimal results. The efficiency of OGA over these algorithms in fields other than SMRFs can be a subject of research.

Declaration

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Akbari, J.; Ayubirad, M.S.: Seismic optimum design of steel structures using gradient-based and genetic algorithm methods. Int. J. Civil Eng. **15**(2), 135–148 (2017)
2. Casapulla, C.; Mousavian, E.; Zarghani, M.: A digital tool to design structurally feasible semi-circular masonry arches composed of interlocking blocks. Comput. Struct. **221**, 111–126 (2019)
3. Mousavian, E.; Casapulla, C.: Quantifiable feasibility check of masonry assemblages composed of interlocking blocks. Adv. Eng. Softw. **149**, 102898 (2020)
4. Holland, J.H.; Reitman, J.S.: Cognitive systems based on adaptive algorithms. In: Pattern-directed inference systems. (1978). Elsevier. pp. 313–329
5. Eberhart, R.; Kennedy, J.: A new optimizer using particle swarm theory. In: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. (1995). IEEE
6. Dorigo, M.; Birattari, M.: Ant colony optimization. Encyclopedia of machine learning. Springer, New York (2010)
7. Yang, X.-S.: A new metaheuristic bat-inspired algorithm, in Nature inspired cooperative strategies for optimization (NICSO 2010). (2010), Springer. pp. 65–74
8. Mirjalili, S.; Mirjalili, S.M.; Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
9. Kaveh, A.; Talatahari, S.: A novel heuristic optimization method: charged system search. Acta Mech. **213**(3–4), 267–289 (2010)
10. de Castro, L.C.L.B.; Partridge, P.W.: Minimum weight design of framed structures using a genetic algorithm considering dynamic analiysis. Latin Am. J. Solids Struct. **3**(2), 107–123 (2006)
11. Camp, C.V.; Bichon, B.J.; Stovall, S.P.: Design of low-weight steel frames using ant colony optimization. In: Structures 2004: Building on the Past, Securing the Future. (2004). pp. 1–11
12. Camp, C.V.; Bichon, B.J.; Stovall, S.P.: Design of steel frames using ant colony optimization. J. Struct. Eng. **131**(3), 369–379 (2005)
13. Morsali, V.; Behnamfar, F.: Seismic damage-based design of steel moment frames. Eur. J. Environ. Civil Eng., p. 1–18 2019.
14. Kaveh, A.; Bakhshpoori, T.: Optimum design of steel frames using Cuckoo Search algorithm with Lévy flights. Struct. Design Tall Spec. Build. **22**(13), 1023–1036 (2013)
15. Kaveh, A.; Bakhshpoori, T.; Azimi, M.: Seismic optimal design of 3D steel frames using cuckoo search algorithm. Struct. Design Tall Spec. Build. **24**(3), 210–227 (2015)
16. Maheri, M.R.; Narimani, M.: An enhanced harmony search algorithm for optimum design of side sway steel frames. Comput. Struct. **136**, 78–89 (2014)
17. Kaveh, A.; Vaez, S.R.H.; Hosseini, P.: Simplified dolphin echolocation algorithm for optimum design of frame. Smart Struct. Syst. **21**(3), 321–333 (2018)
18. Farshchin, M.; Maniat, M.; Camp, C.V.; Pezeshk, S.: School based optimization algorithm for design of steel frames. Eng. Struct. **171**, 326–335 (2018)
19. Gholizadeh, S.; Razavi, N.; Shojaei, E.: Improved black hole and multiverse algorithms for discrete sizing optimization of planar structures. Eng. Optim. **51**(10), 1645–1667 (2019)
20. Toğan, V.: Design of planar steel frames using teaching–learning based optimization. Eng. Struct. **34**, 225–232 (2012)
21. Vahedi, M.; Khoshnoudian, F.; Fallahian, M.; Shadan, F.: Structural damage identification incorporating uncertain boundary flexibility by ensemble couple sparse coding classification method. Eur. J. Environ. Civil Eng., 1–26 (2019)
22. Hasançebi, O.; Bahçecioğlu, T.; Kurç, Ö.; Saka, M.: Optimum design of high-rise steel buildings using an evolution strategy integrated parallel algorithm. Comput. Struct. **89**(21–22), 2037–2051 (2011)
23. Nikakhtar, L.; Zare, S.; Nasirabad, H.M.; Ferdosi, B.: Application of ANN-PSO algorithm based on FDM numerical modelling for back analysis of EPB TBM tunneling parameters. Eur. J. Environ. Civil Eng. (2020), p. 1–18
24. Mangal, M.; Cheng, J.C.: Automated optimization of steel reinforcement in RC building frames using building information modeling and hybrid genetic algorithm. Autom. Constr. **90**, 39–57 (2018)
25. AISC360–10, *Specification for Structural Steel Buildings (ANSI/AISC 360–10).* American Institute of Steel Construction, Chicago-Illinois, 2010.
26. ASCE7–10, *Minimum Design Loads for Buildings and Other Structures (ASCE/SEI 7–10).* American Society of Civil Engineers, Reston, Virginia, 2010.
27. AISC341–10, *Seismic Provisions for Structural Steel Buildings,(ANSI/AISC 341–10).* 2010, American Institute of Steel Construction, Chicago, IL.
28. Woldesenbet, Y.G.; Yen, G.G.; Tessema, B.G.: Constraint handling in multiobjective evolutionary optimization. IEEE Trans. Evol. Comput. **13**(3), 514–525 (2009)
29. Kwak, H.-G.; Kim, J.: An integrated genetic algorithm complemented with direct search for optimum design of RC frames. Comput. Aided Des. **41**(7), 490–500 (2009)
30. Hasançebi, O.; Çarbaş, S.; Doğan, E.; Erdal, F.; Saka, M.: Comparison of non-deterministic search techniques in the optimum design of real size steel frames. Comput. Struct. **88**(17), 1033–1048 (2010)
31. Kaya, M.: The effects of two new crossover operators on genetic algorithm performance. Appl. Soft Comput. **11**(1), 881–890 (2011)
32. Hasançebi, O.; Erbatur, F.: Evaluation of crossover techniques in genetic algorithm based optimum structural design. Comput. Struct. **78**(1–3), 435–448 (2000)
33. Safari, D.; Maheri, M.R.; Maheri, A.: Optimum design of steel frames using a multiple-deme GA with improved reproduction operators. J. Constr. Steel Res. **67**(8), 1232–1243 (2011)
34. Foley, C.M.; Schinler, D.: Automated design of steel frames using advanced analysis and object-oriented evolutionary computation. J. Struct. Eng. **129**(5), 648–660 (2003)

35. Gupta, K.; Gupta, M.K.: Optimization of Manufacturing Processes. Springer, Berlin (2020)
36. Azad, S.K.; Hasançebi, O.: Computationally efficient discrete sizing of steel frames via guided stochastic search heuristic. Comput. Struct. **156**, 12–28 (2015)
37. Gen, M.; Cheng, R.: Genetic algorithms and engineering optimization, Vol. 7. Wiley, New York (2000)
38. Gero, M.B.P.; García, A.B.; del Coz Díaz, J.J.: A modified elitist genetic algorithm applied to the design optimization of complex steel structures. J. Construct. Steel Res. **61**(2), 265–280 (2005)
39. Shi, X.; Liang, Y.; Lee, H.; Lu, C.; Wang, L.: An improved GA and a novel PSO-GA-based hybrid algorithm. Inf. Process. Lett. **93**(5), 255–261 (2005)
40. Toğan, V.; Daloğlu, A.T.: An improved genetic algorithm with initial population strategy and self-adaptive member grouping. Comput. Struct. **86**(11), 1204–1218 (2008)
41. Aravelli, A., *Multi-objective Design Optimization of Engineering Systems: Uncertainty Approach and Practical Applications.* 2014.