Structural and Multidisciplinary Optimization (2023) 66:183
https://doi.org/10.1007/s00158-023-03640-7

RESEARCH PAPER

A guided evolution strategy for discrete sizing optimization of space steel frames

Aytaç Korucu[1] · **Oğuzhan Hasançebi**[1]

Received: 9 February 2023 / Revised: 21 June 2023 / Accepted: 21 July 2023 / Published online: 2 August 2023

Abstract

In this paper, a new design-driven hybrid optimization algorithm called guided evolution strategy (GES) is proposed for a reliable and rapid optimum design of space steel frames. The rationale behind the proposed GES algorithm is to improve convergence characteristics of the evolution strategies (ESs) optimization method by guiding search process according to the satisfaction/violation of strength constraints in a previous design. This is referred to as guided mutation, which is introduced as an auxiliary tool to a stochastic mutation scheme for accelerating the convergence speed of the optimization algorithm. The efficiency of the GES algorithm is investigated and quantified using design examples where sizing optimization of two space steel frames are achieved under strength and displacement constraints imposed according to ANSI/AISC 360-10 (Specification for structural steel buildings, ANSI/AISC 360-10, Illinois, 2010) and ASCE/SEI 7-10 (Minimum design loads for buildings and other structures, ASCE/SEI 7-10, Reston, 2010) design specifications. The solutions produced to these design examples with the GES algorithm are compared to those of some selected metaheuristic search techniques in terms of accuracy of the obtained solutions as well as speed of convergence to the optimum designs. It is shown that the GES algorithm has improved search abilities with respect to other employed techniques.

Keywords Structural optimization · Discrete sizing optimization · Space steel frames · Metaheuristic search techniques · Design-driven search methods · Guided evolution strategy

1 Introduction

In the past, sizing optimization problems of steel frames were usually handled using optimality criteria and mathematical programming optimization methods (Saka and Geem 2013). Nevertheless, the discrete nature of this problem raised various challenges and inconveniences that were difficult to surmount with these techniques. Nowadays, this problem is usually attempted using nature-inspired or metaphor-based optimization algorithms commonly referred to as metaheuristic search techniques. A vast amount of publications exists in the literature related to the use of various metaheuristic search techniques on the subject matter; such as Gholizadeh and Fattahi (2014), Kaveh et al. (2015), Aydoğdu et al. (2016), Gholizadeh and Baghchevan (2017), Gholizadeh and Milany (2018), Talatahari and Azizi (2020), Kazemzadeh Azad (2021), Talatahari et al. (2021, 2022), Degertekin and Tutar (2022), Nouhi et al. (2022), Sadrekarimi et al. (2023), etc.

A common strategy employed by almost all metaheuristic search techniques is that they explore the design space stochastically by making random moves based on some natural, biological, or physical principles (Saka 2007; Yang 2010). These techniques primarily differ from each other in the way a search direction is established in the design space without using or relying on gradient information. Indeed, a gradient-free search is advantageous in the sense that it prevents an algorithm to stagnate in a local optimum. On the other hand, random search strategies employed by these techniques cause them to converge to an optimum solution only after performing a large number of function evaluations. For optimization problems whose objective functions and constraints are expressed mathematically as explicit functions of design variables, this is not of much significance because the optimization time will change from a few seconds to a

Responsible Editor: Gang Li

✉ Oğuzhan Hasançebi
oguzhan@metu.edu.tr

[1] Department of Civil Engineering, Middle East Technical University, Ankara, Turkey

few minutes even for the cases in which millions of function evaluations are necessary. On the other hand, analysis of a structural system is a computationally expansive task, since a single analysis may take up to minutes for large-scale problems subjected to many loading conditions. Therefore, a time-efficient optimization of large-scale structures may not always be accomplished effectively with metaheuristic search techniques.

In order to reduce the computational burden to a manageable size and thus to improve the efficiency of a metaheuristic search technique, mainly two different approaches have been followed in the literature. In the first approach, the computational efficiency of a technique is enhanced by utilizing powerful parallel computing environments. In this approach, the optimization task is distributed among several processors that are connected to each other via network devices. This approach has been successfully implemented for structural design optimization problems in the literature (Hasançebi et al. 2011). It has been reported that the overall computation time required by metaheuristic search techniques can remarkably be reduced by virtue of this approach. However, extremely laborious and costly hardware environment configurations are required for parallel/distributed computing applications. Besides, the network scalability is not always proportional to the number of processors used, and may not be improved further with the inclusion of more processors due to increasing inter-processor communication, coordination, and synchronization overheads.

In the second approach, an attempt is made to improve computational efficiency of a metaheuristic search technique by employing efficient and intelligent search strategies in the stochastic nature of the algorithm, resulting in a reduced number of structural analyses required to locate an optimum solution. In this regard, information gathered during analysis and design stages of a previous solution is often utilized to guide an ongoing search process towards more promising and/or feasible regions of the design space. As a result, a rapid search is carried out through guidance from the evaluation of previous designs, and hence better solutions are attained using relatively fewer number of structural analyses. The structural optimization techniques which implement such intelligent search strategies in their algorithms are referred to as design-driven search methods.

When an appropriate metaheuristic optimization algorithm is employed, the design space is searched thoroughly based on some stochastic rules and moves, yet the search process is random and usually time-consuming. On the other hand, in a design-driven search algorithm the analysis and design data of previous solutions are utilized to generate improved solutions based on engineering analyses and guidance. Although, the convergence is rather rapid, the result produced might not guarantee a near-optimum solution since only a limited search can be accomplished with a design-driven search algorithm. In the present study, a new design-driven hybrid optimization algorithm called guided evolution strategy (GES) is proposed for a time-efficient optimum design of space steel frames. The proposed hybrid algorithm efficiently combines the useful features of these two search criteria to locate a good and reliable near-optimum solution using an affordable computation time and effort.

The evolution strategies (ESs) algorithm is a nature-inspired metaheuristic search technique whose underlying principles rest on the simulation of natural evolution. The technique has been developed in various forms in the literature. The first variant of the technique, known as $(1+1)-$ES, works on the basis of two individuals (designs), i.e., a parent and an offspring individual, at each iteration. The subsequent variants of the technique, referred to as $(\mu+\lambda)-$ES and $(\mu, \lambda)-$ES, employ design populations consisting of μ parent and λ offspring individuals, and are mainly intended to carry out a self-adaptive search in continuous designs spaces. In Hasançebi (2007), various discrete reformulations of the technique are presented and evaluated in the context of discrete sizing optimization of steel frames, and a new discrete reformulation is proposed.

The GES algorithm has some similarities and differences with the $(1+1)$ as well as (μ, λ) and $(\mu+\lambda)$ multi-membered variants of the ESs. Similar to the $(1+1)-$ES, it works on the basis of two individuals, and the selection scheme is implemented in the same way as in the $(1+1)-$ES. That is to say, if an offspring is better than its parent, it survives and becomes the parent at the next generation; otherwise, it dies out and the parent survives. On the other side, in the GES algorithm mutation of a parent individual is performed along the same line as the (μ, λ) and $(\mu+\lambda)$ discrete variants of the ESs, as formulated by Hasançebi (2007). However, in addition to this mutation operator named stochastic mutation, a so-called guided mutation is also incorporated into the algorithm to guide the search process according to satisfaction/violation of strength constraints in the parent design.

The sections of the paper are organized as follows. In Sect. 2, design-driven optimization methods used in structural optimization are overviewed. Besides, the three metaheuristic search techniques employed in this study for performance comparison with the proposed GES algorithm are briefly introduced in this section. Section 3 provides mathematical formulations regarding discrete sizing optimization problems of space steel frames under strength and displacement constraints imposed according to ANSI/AISC 360–10 (2010) and ASCE/SEI 7–10 (2010) design specifications. Geometric constraints between various steel members are also defined and emphasized in this section. In Sect. 4, the GES algorithm is introduced and its fundamental working principles are explained in detail. In Sect. 5, two design examples related to the sizing optimization of space

steel frames are studied using the GES algorithm as well as the aforementioned metaheuristic search techniques. This way, the performance of the GES algorithm is identified and quantified in terms of the solution accuracy and convergence speed in comparison to those of the metaheuristic search techniques. Finally, the important findings of the study are summarized and highlighted in Sect. 6.

2 Design-driven and metaheuristic search techniques: a review

In this section, the design-driven search methods used in structural optimization literature are overviewed first. Next, the three metaheuristic search techniques that are implemented in the study for numerical performance comparison with the proposed GES algorithm are briefly introduced with a particular emphasis on their reformulations and adaptations for structural optimization problems.

2.1 Design-driven search techniques

The fully stressed design (FSD) approach is one of the early optimization methods where a design-driven search strategy is implemented for structural optimization problems (Razani 1965; Gallagher and Zienkiewicz 1973). This method has long been used successfully for problems in which the constraints solely consist of stress limitations imposed on members. In this method, a search direction is determined based on the constraint (stress) ratios of structural members. In case a constraint (stress) ratio for a member happens to be well below a specified maximum value, a smaller cross-section is adopted for that member to enable a more effective use of material. If, however, a constraint (stress) ratio for a member is well above a specified maximum value, then a larger cross-section is assigned to the member to eliminate the associated constraint violation. In other words, in this approach, when a new (trial) solution is generated from the existing one, the design variables (i.e., cross-sectional areas of the members) are assigned such that the demand-to-capacity ratio (DCR) is adjusted to a value around unity for all the members. Although the FSD has proved to be a very effective method, it is limited to optimization problems with stress or strength constraints only; that is to say, displacement constraints cannot be dealt with this method.

Later, an improvement of the FSD method, known as fully utilized design (FUD), has been proposed to handle displacement constraints in addition to strength/stress constraints. In FUD, the calculated stress ratios by FSD are all prorated by a single factor determined based on the most violated displacement constraint in a structure. Although a feasible solution is obtained rapidly by the FUD, it has been shown that this solution in most cases corresponds to an overdesigned structure. Hence, Patnaik et al. (1998) proposed a modified (improved) version of the FUD, called modified fully utilized design (MFUD), to eliminate the problems associated with displacement constraints. Therein, the integrated force method (IFM) of structural analysis was integrated into the FUD to identify the significance of a violated displacement constraint and how effectively it could be remedied (Patnaik et al. 1991). However, the proposed method is limited to truss structures only.

In several studies, the concept of FSD has been combined with the global search abilities of metaheuristic search techniques to create effective hybrid algorithms. For instance, Ahrari and Atai (2013) and Ahrari et al. (2015) have proposed a so-called fully stressed design evolution strategy (FSD–ES) that combines useful features of the FSD and ES techniques to enhance computational efficiency of the latter. Later, Ahrari and Deb (2016) proposed an improved version of the FSD–ES, named FSD–ES-II, by introducing displacement constraints into the algorithm. A performance investigation of these hybrid methods (FSD–ES and FSD–ES-II) through some selected benchmark problems of structural optimization verified the efficiencies of the proposed methods. However, it is worth mentioning that these approaches are also limited to truss structures only.

Solving sizing optimization problems of steel frames with design-driven search methods has also been attempted in several studies in the literature. Murren and Khandelwal (2014) presents a design-driven harmony search algorithm (DDHS), in which constraint satisfaction and violation data from previous trial solutions are exploited to select smaller or larger sections for member groups. The effectiveness of the algorithm has been demonstrated using three benchmark planar steel frame problems. The principle of virtual work has been employed in some studies such as Elvin et al. (2009), Walls and Elvin (2010), and Kazemzadeh Azad and Hasançebi (2015). In these studies, the member groups which have the highest influence (participation factor) on nodal point displacements are identified to avoid displacement constraint violations more effectively.

2.2 Metaheuristic search techniques

In this study, the numerical performance of the GES algorithm is evaluated in comparison to three metaheuristic search techniques; namely, big bang–big crunch (BB–BC), particle swarm optimization (PSO), and evolution strategies (ES). It is important to emphasize that among hundreds of different metaheuristics, these techniques are deliberately selected and used for comparison based on their generality and robustness for the problems of interest.

The particle swarm optimization (PSO), introduced by Kennedy and Eberhart (1995), is one of the most common and widely accepted metaheuristic optimization techniques

employed by all disciplines of science and engineering. The method simulates the collective learning behavior of species like swarm, herd, and flock, and it has found fruitful applications in different areas of structural optimization. Many improvements of this technique have been accomplished in the literature. The PSO algorithm used in this study refers to the formulation of the technique by Hasançebi et al. (2010) and the computational steps of this algorithm are outlined in the relevant paper.

The BB–BC algorithm, developed by Erol and Keskin (2006), is inspired by the evolution of the universe theory. Many enhancements of this technique are developed in the literature. The one used in this study refers to a powerful and effective enhancement of the standard BB–BC algorithm, called the exponential big bang–big crunch algorithm (EBB–BC), which is proposed by Hasançebi and Kazemzadeh Azad (2012) for discrete sizing structural optimization problems in particular. A detailed formulation of this algorithm is presented in the relevant paper.

Evolution strategies (ESs), being one of three mainstreams of evolutionary algorithms, evolve a population of designs over generations using the concepts of natural evolution. The two multi-membered variants of this technique, known as (μ, λ)-ES and $(\mu+\lambda)$-ES, use a population of solutions while searching the design space. The way the selection is carried out identifies the only difference between these variants. In the (μ, λ)-ES, the parents are all left to die out, and the best μ individuals are chosen deterministically out of λ offspring in reference to their objective functions. In the $(\mu+\lambda)$-ES, however, the parents are also involved in the selection mechanism, and the best μ individuals are chosen from μ parents plus λ offspring. Although the (μ, λ) variant makes it easier to escape from mis-adapted strategy parameters and thus comes up with an increased promise for finding the optimum, it is not suitable much for the upper bound strategy (UBS), which is discussed in Sect. 4.4. Therefore, a discrete reformulation of the $(\mu+\lambda)$-ES proposed by Hasançebi (2007) is favored and employed in this study to accelerate convergence rate of the algorithm. It should be highlighted that the UBS is also implemented in conjunction with the EBB–BC and PSO algorithms to reduce the number of structural analyses required during an optimization process.

The successful applications of these methods in discrete sizing optimization of steel skeletal structures have been reported and verified in many publications of the authors, such as Hasançebi (2007), Hasançebi et al. (2010, 2011), and Hasançebi and Kazemzadeh Azad (2012). It should also be emphasized that most of the metaheuristic search techniques are initially developed and evaluated based on their performances on some mathematical test functions and/or benchmark optimization problems, which have a limited number of continuous design variables. It has been observed that some of these techniques might lead to non-optimum or very poor solutions when directly applied to practical structural optimization problems, which usually have a large number of discrete design variables. Therefore, not all metaheuristic approaches are indeed suitable for applications to discrete and large-scale structural optimization problems or they require an extensive reformulation or adaptation before being applied to such problems. An extensive survey of a wide range of metaheuristic search techniques employed in structural optimization problems is presented in some review papers such as Renkavieski and Parpinelli (2021) and Kashani et al. (2022).

3 Optimum design problem formulation

The sizing optimization problem of a space steel frame can be defined as finding the minimum weight of the frame subject to strength and displacement requirements imposed according to a chosen code of practice as well as geometric constraints between the connected members in the frame. It should be noted that in practical design applications of steel frames, members must be selected from a set of commercially available steel sections. Therefore, a section pool that consists of a number of available steel sections is prepared prior to the initialization of an optimization process. The steel sections are sorted in increasing order of a chosen sectional property; typically, the cross-sectional area. Each steel section in the pool is identified and referenced with a distinct sequence number that varies between 1 and the total number of sections in line with the sorted order of the sections in the pool. During an optimization process, selection and sizing of member groups are carried out in connection with the sequence numbers. That is to say, when a selection is made for a member group, the cross-sectional properties for the selected section become available from the section pool through the sequence number.

Accordingly, in a discrete sizing optimization problem of a space steel frame, the objective is to find a vector of design variables $\boldsymbol{I}$, which minimizes the weight (W) of the frame subject to a number of design constraints.

$$\boldsymbol{I}^{\mathrm{T}} = \left[I_1, I_2, \ldots, I_{N_g}\right] \tag{1}$$

$$W = \sum_{i=1}^{N_g} \gamma_i A_i \sum_{j=1}^{N_{m,i}} L_{j,i} \tag{2}$$

$$g_k \leq 0, \quad k = 1, 2, \ldots, N_c \tag{3}$$

In Eqs. (1)–(3), the design variable vector $\boldsymbol{I}$ holds sequence numbers of the sections assigned to N_g member groups from

the section pool; γ_i and A_i are the unit weight and area of the section assigned to a member group i, respectively; $N_{m,i}$ is the total number of members in member group i; $L_{j,i}$ is the length of the member j in member group i; g_k is the k-th design constraint; and N_c is the total number of design constraints. The design constraints consist of various requirements imposed by design codes and construction practices, and are defined in the following sub-sections.

3.1 Strength constraints

For steel members that are subjected to bending and axial loads, the following strength requirements are imposed:

$$g_1 = \begin{cases} \left(\frac{P_u}{\phi_a P_n}\right)_j + \frac{8}{9}\left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}}\right)_j - 1.0 \le 0 \; for \; \left(\frac{P_u}{\phi_a P_n}\right)_j \ge 0.2 \\ \left(\frac{P_u}{2\phi_a P_n}\right)_j + \left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}}\right)_j - 1.0 \le 0 \; for \; \left(\frac{P_r}{\phi_a P_c}\right)_j < 0.2 \end{cases} \tag{4}$$

For steel members that are subjected to shear, the following strength requirement is imposed:

$$g_2 = \left(\frac{V_u}{\phi_v V_n}\right)_j - 1 \le 0. \tag{5}$$

In Eqs. (4) and (5), P_u, M_u, and V_u are the required axial, flexural, and shear strengths calculated using the load combinations given in the ASCE/SEI 7–10 (2010) design load specification, respectively; P_n, M_n, and V_n are the nominal axial, flexural, and shear strengths calculated according to the formulations given in the ANSI/AISC 360–10 (2010) specification, respectively; ϕ_a, ϕ_b , and ϕ_v are the resistance factors for axial, flexural, and shear strengths, respectively, and they are all set to 0.9; the subscripts x and y represent the strong and weak axes of bending for member j, respectively.

3.2 Displacement constraints

The displacement constraints consist of limitations imposed on the deflection of beams and inter-story drift requirements. The maximum deflection in a steel beam is limited to span length divided by 360 under reduced live loads, as formulated in Eq. (6)

$$g_3 = \delta_j - \frac{L_j}{360} \le 0, \tag{6}$$

where δ_j and L_j are the calculated deflection in a beam j under reduced live loads and its span length, respectively.

In compliance with ASCE/SEI 7–10 (2010) specification, the inter-story drift in a steel frame is limited as follows.

$$g_4 = \frac{\Delta_i}{\Delta_a} - 1 \le 0 \text{ where } \Delta_i = \frac{C_d \delta_{xe}}{I_e}. \tag{7}$$

In Eq. (7), Δ_i and Δ_a represent the drift of the story i under seismic action and the allowable design story drift, respectively; δ_{xe} is the maximum difference between horizontal displacements of vertically aligned points at the top and bottom of a story along any of the edges of the structure and is determined by elastic analysis; C_d and I_e are the deflection amplification factor and the importance factor, respectively.

3.3 Geometric constraints

Steel members cannot be selected arbitrarily and connected together even though strength and displacement constraints are satisfied by all the members of a connection. Geometric constraints refer to a set of construction requirements that must be satisfied by the members that frame into the same joint. Unless these requirements are satisfied, solid connections cannot be produced, leading to impractical and sometimes insecure designs. Yet, despite their significance, geometric constraints are sometimes disregarded or partially considered in the previous studies. In the following, geometric constraints between various members of a connection are introduced, considering splice type connections between the column members in line.

Girders and columns that are connected together must satisfy certain geometric constraints depending on geometry of the connection. For a girder connected to the flange of a column member (Fig. 1), it is required that the flange width of the girder (b_f^g) does not exceed the flange width of the column (b_f^c). Accordingly, the associated geometric constraint is defined in Eq. (8).

$$g_5 = \frac{b_f^g}{b_f^c} - 1.0 \le 0 \tag{8}$$

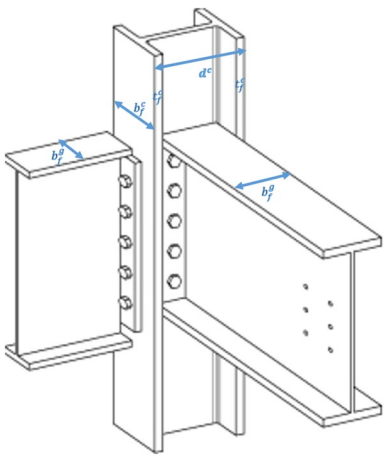

Fig. 1 Girder-to-column connection

For a girder connected to the web of a column member (Fig. 1), it is required that the flange width of the girder ideally does not exceed the clear web height of the column. Accordingly, the associated geometric constraint is defined in Eq. (9)

$$g_6 = \frac{b_f^g}{\left(d^c - 2t_f^c\right)} - 1.0 \le 0, \tag{9}$$

where b_f^g is the flange width of the girder; d^c is the total depth of the column; t_f^c is the flange thickness of the column; and $d^c - 2t_f^c$ is the clear web height of the column.

The column members in multi-story steel frames can be connected together using steel-plated bolted connections called column splices, as depicted in Fig. 2. For each splice connection, five geometric constraints can be defined as follows.

First, it is required that the depth of the upper column (d^{uc}) does not exceed the depth of the lower column (d^{lc}). Accordingly, the associated geometric constraint is defined in Eq. (10).

$$g_7 = \frac{d^{uc}}{d^{lc}} - 1.0 \le 0. \tag{10}$$

Second, it is required that the flange thickness of the upper column (t_f^{uc}) does not exceed the flange thickness of the lower column (t_f^{lc}). Accordingly, the associated geometric constraint is defined in Eq. (11).

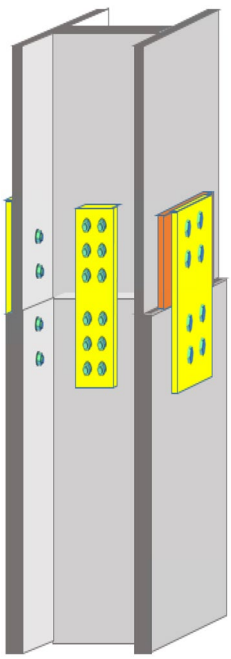

Fig. 2 Column-to-column connection (splice connection)

$$g_8 = \frac{t_f^{uc}}{t_f^{lc}} - 1.0 \le 0. \tag{11}$$

Third, it is required that the web thickness of the upper column (t_w^{uc}) does not exceed the web thickness of the lower column (t_w^{lc}). Accordingly, the associated geometric constraint is defined in Eq. (12).

$$g_9 = \frac{t_w^{uc}}{t_w^{lc}} - 1.0 \le 0. \tag{12}$$

Fourth, it is required that the flange width of the upper column (b_f^{uc}) does not exceed the flange width of the lower column (b_f^{lc}). Accordingly, the associated geometric constraint is defined in Eq. (13)

$$g_{10} = \frac{b_f^{uc}}{b_f^{lc}} - 1.0 \le 0. \tag{13}$$

Fifth, it is also required that the clear height of the upper column is not smaller than the clear height of the lower column. Accordingly, the associated geometric constraint is defined in Eq. (14)

$$g_{11} = \frac{d^{lc} - 2.t_f^{lc}}{d^{uc} - 2.t_f^{uc}} - 1.0 \le 0, \tag{14}$$

where t_f^{uc} is the flange thickness of the upper column; t_f^{lc} is the flange thickness of the lower column; d^{lc} is the depth of the lower column; and d^{uc} is the depth of the upper column. It should be noted that this requirement together with Eqs. (10) and (13) ensure that the flange of the upper column rests on the flange of the lower column.

It is important to mention that another connection type between column members in a steel frame is a cap plate connection, in which the plates welded to the ends of two columns are connected to each other by bolts. In the previous optimization works of steel frames in the literature, this connection type was usually assumed, in which case it would be sufficient to impose only Eqs. (10) and (13). However, in this study, all column-to-column geometric constraints given in Eqs. (10) through (14) are considered, assuming splice connections between column members.

3.4 Constraint handling with penalty function

Metaheuristic search techniques are fundamentally unconstrained optimization methods. During the past few decades, various strategies have been proposed for handling constraints with these techniques. One of the most common approaches implemented is to transform a constrained problem into an unconstrained one. Such a transformation is

performed by introducing a penalty function and integrating it into the objective function. The penalty function penalizes an infeasible solution, which violates problem constraints. For minimization problems, the violated constraints lead to an increase in the objective function value of that particular solution. The constraint-integrated objective function is referred to as penalized (constrained) objective function. Equation (15) presents the constrained objective function used in this study:

$$\phi = W\left[1 + \alpha_c\left(\sum_{k=1}^{N_c} \max\left(0, g_k\right)\right)\right], \tag{15}$$

where ϕ is the penalized objective function; W is the original objective function (i.e., the weight of a frame); N_c is the total number of design constraints, g_k is the k-th normalized design constraint; and finally, α_c is the penalty coefficient used to adjust the scale of penalization as a whole and is set to unity in this study.

4 Guided evolution strategy (GES)

In this study, a design-driven guided evolution strategy (GES) algorithm is proposed for a time-efficient optimum design of space steel frames. The rationale behind the GES algorithm is to improve convergence characteristics of the evolution strategies (ESs) technique by utilizing the information gathered during structural analysis and design stages.

Similar to (1 + 1) – ES, the GES algorithm works on the basis of two designs, i.e., a parent and an offspring individual at each iteration. A design vector is referred to as individual ($\boldsymbol{J}$) and it consists of a vector of sizing design variables ($\boldsymbol{I}$) and two sets of strategy parameters (p, $\boldsymbol{\psi}$), as formulated in Eq. (16). The strategy parameters are self-adaptive, which means that they are adjusted to the most suitable values automatically by the algorithm to achieve an effective search of the design space during an optimization process.

$$\boldsymbol{J} = \boldsymbol{J}(\boldsymbol{I}, p, \boldsymbol{\psi}). \tag{16}$$

In Eq. (16), p refers to the mutation probability parameter employed for design variables. It should be noted that this parameter determines the percentages of design variables that are mutated probabilistically when generating a new (offspring) individual. As explained in the following, in this algorithm design variables are mutated using geometric distributions. A geometric distribution is a discrete probability function and its shape is mainly controlled by a geometric distribution parameter (ψ), as shown in Fig. 3. In general, the higher the ψ parameter is, the flatter the distribution becomes, enabling large moves (step sizes) and explorative search in the design space. Oppositely, the low values of ψ parameter encourage small moves in favor of an exploitative search. It is crucial to mention that the geometric distribution parameters are not set to constant values; rather they are adjusted by the algorithm adaptively to emphasize either an explorative or exploitative search for the most successful mutation of design variables. For the mutation of each design variable I_i, a different geometric distribution ψ_i is defined and used, and thus the vector $\boldsymbol{\psi}$ in Eq. (16) represents the whole set of geometric distribution parameters employed by the algorithm, i.e., ($\psi_i = 1, \ldots, N_g$). The implementation steps of the GES algorithm are described in the following sub-sections.

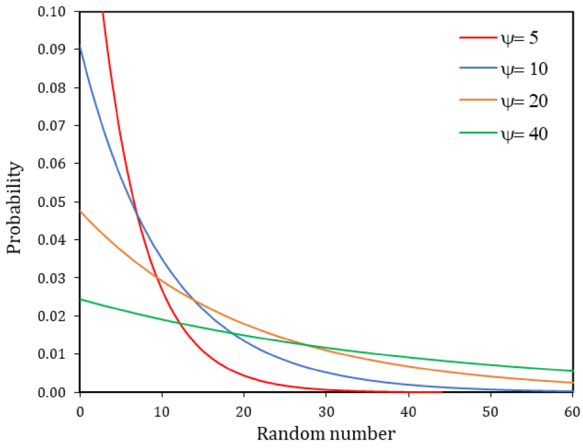

Fig. 3 Geometric distribution

4.1 Initialization

The algorithm is initiated with a generation of an initial parent individual. This individual is generated such that the sizing design variable vector $\boldsymbol{I}$ is initialized at random, and the strategy parameters (p, $\boldsymbol{\psi}$) are set to suitable initial values.

4.2 Evaluation of the parent individual

The parent individual is evaluated where structural analysis is carried out for the individual to obtain its force and deformation responses, and its objective function value is calculated from Eq. (15).

4.3 Mutation

Mutation is implemented to create an offspring individual from the parent one. The GES algorithm employs a stochastic mutation, similar to the (μ, λ) and ($\mu + \lambda$) discrete variants of the ESs, as developed by Hasançebi (2007). Indeed, a stochastic mutation is useful since it accommodates an extensive exploration of the design space, preventing entrapment

of the algorithm in a local optimum. However, in order to accelerate convergence speed of the algorithm, a so-called guided mutation is introduced as a supplementary tool to guide the search process using the information obtained during the evaluation stage of the parent individual. In order for the algorithm to take advantage of these two search features together during an optimization process, both mutation schemes are incorporated into the GES algorithm. Accordingly, a stochastic mutation is still followed when generating an offspring individual from the parent one, yet occasionally this conventional mutation scheme is replaced by the guided mutation scheme according to a predefined possibility called guided mutation ratio (GMR) in this study. Whether an offspring is generated based on a stochastic or guided mutation scheme is decided using Eq. (17).

$$M_{\text{type}} = \begin{cases} \text{Stochastic}, & \text{if } r > \text{GMR} \in [0,1] \\ \text{Guided}, & \text{if } r \leq \text{GMR} \in [0,1] \end{cases} \tag{17}$$

Equation (17) is implemented such that when a new offspring individual is generated, a uniform random number (r) is sampled anew in the range of [0, 1]. If $r \leq$ GMR, the offspring individual is produced using the guided mutation scheme; otherwise ($r >$ GMR) the conventional stochastic mutation is applied. In the following, these two mutation schemes are explained in detail.

4.3.1 Stochastic mutation

As mentioned previously, the stochastic mutation scheme is implemented along the same line as (μ, λ) and ($\mu+\lambda$) discrete variants of the ESs. Prior to mutation of design variables, the mutation probability parameter p is mutated first using a logistic normal distribution (Eq. 18) so that it is set to a different value between 0 and 1 for each offspring individual generated. A logistic normal distribution has a probability density curve similar to that of a normal distribution. However, unlike a normal distribution, it returns a value between 0 and 1 and thus ensures that the mutated value of the p parameter always remains between 0 and 1.

$$p' = \left(1 + \frac{1-p}{p} \cdot e^{-\gamma \cdot N(0,1)}\right)^{-1}. \tag{18}$$

In Eq. (18), p' is the mutated value of the p parameter; $N(0, 1)$ is a random number generated from a normal distribution with a mean of 0 and a standard deviation of 1, and it is sampled anew for each individual; and $\gamma = 1/\sqrt{2\sqrt{N_g}}$ is a learning rate constant for the p parameter, where N_g stands for the number of member groups (sizing design variables). It is important to mention that upper and lower bounds are also defined for the mutation probability parameter obtained from Eq. (18), as follows:

$$p_{\min} \leq p' \leq p_{\max}, \tag{19}$$

where the minimum mutation probability parameter p_{min} is imposed to ensure the mutability of the individual. In this study, this parameter is set to a value of $p_{min} = 1/N_g$ to facilitate mutation of at least one design variable probabilistically. Similarly, the maximum mutation probability parameter p_{max} is enforced to eliminate excessive mutation of the individual. In this study, p_{max} is not set to a static value; instead, it is decreased linearly from its initial value of p^0_{max} at iteration (generation) zero to its final value $p^{t_{max}}_{max}$ at the last iteration in order to promote a more exploitative search towards later stages of the optimization process (Eq. 20).

$$p^{t+1}_{\max} = p^0_{\max} - \left(p^0_{\max} - p^{t_{\max}}_{\max}\right)\frac{t}{t_{\max}}. \tag{20}$$

In Eq. (20), t_{max} is the maximum number of iterations to be performed during an optimization process, t is the current iteration number, and p^{t+1}_{max} is the maximum value of the mutation probability parameter at iteration $(t+1)$. In this study, p^0_{max} and $p^{t_{max}}_{max}$ are set to the following values: $p^0_{max} = 0.25$ and $p^{t_{max}}_{max} = 1/N_g$.

For each design variable I_i, a uniform random number r is then sampled in the range [0,1]. If the random number r is smaller than or equal to p', mutation of the design variable is performed; otherwise, the variable is not mutated. Before mutating the design variable, however, its associated geometric distribution parameter ψ_i must be mutated first using a lognormal distribution given in Eq. (21). It should be noted that the lognormal distribution yields a positive value only for a random variable. This way, it is ensured that the mutated value of a geometric distribution parameter remains always positive.

$$\psi_i' = \psi_i e^{\tau N_i(0,1)} \geq 1.0 \quad if\ r \leq p' \in [0,1] \tag{21}$$

In Eq. (21), ψ_i' is the mutated value of the ψ_i parameter; τ is a learning rate constant for the ψ_i parameter and it can be set to $\tau = 1/\sqrt{N_g}$; and a normally distributed random number $N_i(0, 1)$ is sampled anew for each design variable.

It is important to note that Eq. (21) may occasionally yield a too small or too big geometric distribution parameter due to mis-adapted strategy parameters, which adversely affect the search performance of the algorithm. Hence, an upper bound ψ_{max} and a lower bound ψ_{min} are also defined for this parameter to warrant an efficient search, as formulated in Eq. (22).

$$\psi_{\min} \leq \psi_i' \leq \psi_{\max} \tag{22}$$

Accordingly, if the mutated value of a geometric distribution parameter exceeds ψ_{max}, it is set to $\psi_i' = \psi_{\max}$; likewise, if it falls below ψ_{min}, it is set to $\psi_i' = \psi_{\min}$. In this study, ψ_{min} and ψ_{max} are set to the following values: $\psi_{min} = 2\sqrt{N_s}/5$ and $\psi_{max} = 5\sqrt{N_s}/2$, where N_s is the number of sections (values) in the discrete set used.

Finally, to mutate a design variable, two integer random numbers $(z_{i,1}, z_{i,2})$ are sampled according to a geometric distribution with the parameter ψ_i', and I_i is mutated using Eq. (23), where I_i' indicates the mutated value of the design variable.

$$I_i' = I_i + z_{i,1}(\psi_i') - z_{i,2}(\psi_i'). \tag{23}$$

4.3.2 Guided mutation

During the evaluation stage, a parent individual is analyzed under all design load combinations, and the maximum demand-to-capacity ratio (DCR) is established for each sizing design variable concerning the strength constraints. For a design variable, a DCR value below one indicates that related strength constraints are satisfied with the current section selected for the corresponding member or member group, and a smaller (lighter) section might be used to provide economy (i.e., the current section might be overdesign). Oppositely, for a sizing design variable, a DCR value above one implies that related strength constraints are not satisfied; accordingly, the current section is not adequate (under-designed) and a stronger section must be selected to eliminate the constraint violation.

In the guided mutation scheme, search directions are determined for the design variables in accordance with their associated DCR values in the parent individual. Accordingly, for design variables with DCR values larger than one, the type of search direction (SD_{type}) is labeled as SD_{inc}, implying that an increase in cross-section (i.e., stronger section) is necessary to remedy the constraint violations associated with these variables. Conversely, for design variables with DCR values smaller than one, the type of search direction is labeled as SD_{dec}, indicating that a decrease in cross-section (i.e., lighter section) might be useful in the sense of more effective use of material, as formulated in Eq. (24).

$$SD_{type} = \begin{cases} SD_{inc}, & \text{if DCR} > 1.0 \\ SD_{dec}, & \text{if DCR} \le 1.0 \end{cases} \tag{24}$$

In fact, at times when no global displacement constraint is imposed, ideally, DCR values for all member groups should be adjusted to 1.0 to ensure the most effective use of the material in a structure. However, the presence of displacement constraints might overwrite this rule to some extent because some displacement quantities might be quite sensitive to the cross-sectional properties of certain members, and thereby selection of larger sections for those members might be necessary to reduce displacements to the desired levels at the expense of having low DCR values for those particular members.

Another important issue is that not all design variables should be mutated at a time while generating an offspring individual from the parent one. This is because excessive mutation of an individual causes quite a different set of force distributions within structural members, reducing exploitative search characteristics of the algorithm. For this reason, mutation is implemented only on a selected number of design variables, and whether a design variable is mutated or not during the guided mutation scheme is decided upon using Eq. (25).

$$\text{Mutation} = \begin{cases} \text{Yes; } SD_{inc}, & \text{if DCR} > 1.0 \\ \text{Yes; } SD_{dec}, & \text{if DCR} < 1.0 \text{ and } r < (1 - \text{DCR})^2 \in [0, 1] \\ \text{No}, & \text{if DCR} < 1.0 \text{ and } r \ge (1 - \text{DCR})^2 \in [0, 1] \end{cases} \tag{25}$$

According to Eq. (25), the design variables with DCR values above one are strictly mutated in the increasing search direction (SD_{inc}) to have a stronger section and thereby to remedy the related constraint violation. On the other hand, design variables with DCR values below one are mutated probabilistically, where $(1 - \text{DCR})^2$ represents the mutation probability of the design variable. A uniform random r is generated anew in the range of [0, 1] and compared to the mutation probability of the design variable. If $r < (1 - \text{DCR})^2$, the design variable is mutated in the decreasing search direction (SD_{dec}) to have a lighter section; otherwise, no mutation is applied to the design variable. It should also be noted in Eq. (25) that the design variables with lower DCR values are more likely to be mutated as the mutation probability parameter $(1 - \text{DCR})^2$ approaches to one.

One point deserves particular attention. In discrete structural optimization, the sections that are used to size member groups in a structure are sorted in ascending order of cross-sectional areas in a section pool (discrete set). An increase in the cross-section in this list results in a heavier section, but not necessarily a stronger section. That is to say, the next section is larger in cross-sectional area but it may have a smaller moment of inertia about the strong or weak axis or both. On the other hand, in frame-type structures, multiple failure modes are usually available for structural members under the combined effect of axial force and bending moment. For example, one member may fail under shear; another member may fail under bending about the strong axis or weak axis, or local buckling mechanisms may be observed, etc. Therefore, an increase in member size in a

section pool may lead to a heavier yet weaker section; likewise, a decrease in member size might lead to both a lighter and stronger section.

In this study, an increasing or a decreasing search direction for a design variable is defined by preparing a so-called "Increasing Direction List (IDL)" in the former and "Decreasing Direction List (DDL)" in the latter. In this regard, an IDL consists of a predefined number of sections that have sectional properties larger than those of the section selected currently for the design variable, not only in terms of cross-sectional area but also the moments of inertias about strong and weak axes. Hence, to prepare an IDL for a design variable, sections very next to the currently assigned value of the variable in the list are scanned in the increasing direction of sequence number, and only those larger (heavier) sections which also have larger moment of inertias about both axes are included in the list. On the other hand, a DDL consists of a predefined number of sections that are lighter than the section selected currently for the design variable. As discussed above, sometimes a decrease in member size in a section pool might lead to a lighter yet stronger section, which is in fact in favor of the design variable. Accordingly, a DDL is prepared for a design variable such that a predefined number of sections very next to the currently assigned value of the variable in the decreasing direction of sequence number are included in the list without enforcing any requirement that they will also have smaller moment of inertias. Once an IDL or DDL is prepared for a design variable, mutation of the design variable is performed by stochastically switching to any of these sections in the corresponding list under equal probability.

The number of sections included in an IDL or DDL is referred to as "Guided Mutation Limit (GML)" in this study. Indeed, the GML is identical to the term "step size" in a traditional optimization algorithm. The larger this parameter is, the more explorative search can be achieved at the expense of decreased exploitative search capabilities of the algorithm. Indeed, at the beginning of the optimization process, a large step size is generally beneficial since it allows for an extensive exploration of the design space. However, exploitation of good solutions in favorable design regions requires a smaller step size for arriving at better solutions, especially towards the later stages of an optimization process.

4.4 Evaluation of the offspring individual

After an offspring individual is generated, its structural weight (W) is calculated first using Eq. (2) and compared with the objective function (ϕ) of the parent individual. If the former is lower than the latter, structural analysis is carried out for the offspring individual to obtain its force and deformation responses, and its objective function is calculated using Eq. (15). In the other case, the offspring is automatically eliminated without performing any structural analysis as it has no chance to defeat the parent individual anyway. This practice is known as the upper bound strategy (UBS) and is implemented in the GES algorithm to significantly reduce the number of structural analyses required during an optimization process (Kazemzadeh Azad et al. 2013).

4.5 Selection

The selection process is then implemented between the parent and offspring individuals such that whichever is better (having a lower objective function value) among the two survives and the other one dies out. Accordingly, the GES algorithm employs a strict elitism rule except during a stagnation escape period, which is discussed in the following section.

4.6 Stagnation control strategy

The GES algorithm utilizes a basic stagnation control strategy based on an uphill move to avoid entrapment at a local optimum. According to this strategy, if the parent individual is not improved over a predetermined number of iterations (t_{STAG}, which is set to 100 in this study) a stagnation escape period (SEP) is initiated by the algorithm at the next iteration. Before the SEP, the algorithm only allows for a transition to a better solution, and thus the parent individual (also called the elite design) represents the best solution obtained thus far in an optimization process. With the start of SEP, the elitism rule is suspended temporarily, and a transition to a non-improving offspring individual (called the uphill design) is allowed only once provided that it has an objective function value not more than a predefined ratio (α_{SEP}) of the objective function of the elite design;

$$\phi_{\mathrm{uphill}} \leq \alpha_{\mathrm{SEP}} \cdot \phi_{\mathrm{eli}}, \tag{26}$$

where ϕ_{uphill} and ϕ_{eli} are the objective function values of the uphill and elite designs, respectively, and α_{SEP} is set to 1.05 in this study.

Once an uphill design is established, it is referred to as the SEP design. Then, the elitism rule is reactivated and the SEP design is replaced and updated only by an improving offspring until it becomes better than the elite design. The maximum number of iterations performed within a SEP loop is referred to as t_{SEP}, which is also set to 100 in this study. While performing SEP iterations, if a SEP design better than the elite design is produced, SEP is terminated immediately, and this design becomes the new elite (parent) design of the optimization process. If, however, a SEP design better than the elite design cannot be located when t_{SEP} is reached,

another SEP loop is initiated but this time a new uphill move is allowed with reference to the last improved design (ϕ_{SEP}) of the previous SEP; that is, $\phi_{uphill} \leq \alpha_{\text{SEP}} \cdot \phi_{SEP}$.

4.7 Termination

The procedures described in Sects. 4.2 through 4.6 are implemented until a predetermined number of maximum iterations (t_{max}) is completed. A flowchart of the GES algorithm is shown in Fig. 4.

5 Numerical examples

In this section, the optimum designs of two moment-resisting space steel frames are studied numerically using the proposed GES algorithm as well as three metaheuristic search

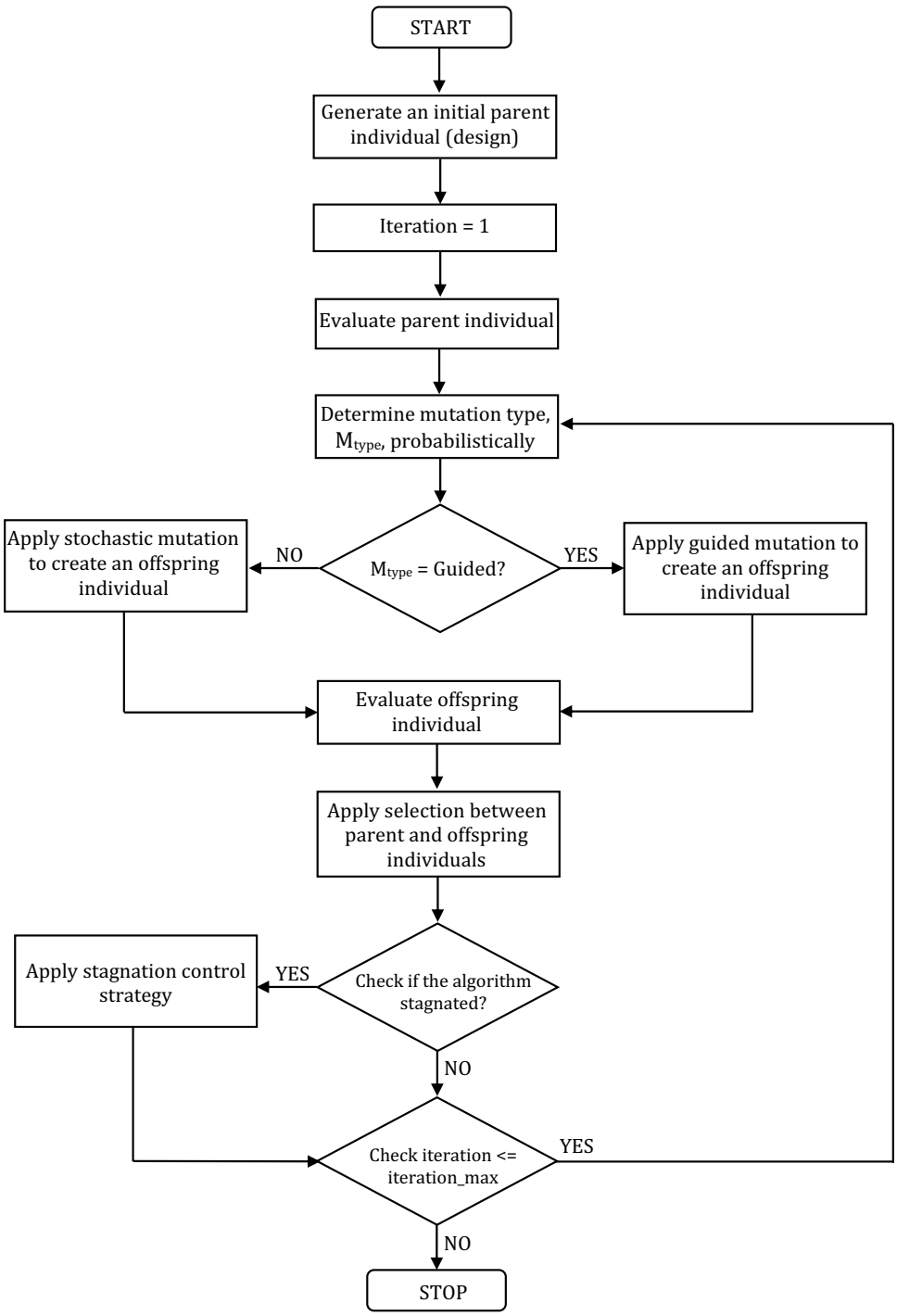

Fig. 4 Flowchart of the guided evolution strategy algorithm

techniques, namely big bang–big crunch (BB–BC), particle swarm optimization (PSO), and evolution strategies (ES). In these examples, performance of the GES algorithm is compared to those of the metaheuristic search techniques in terms of quality of the optimum solution as well as speed of convergence to an optimum solution (i.e., convergence rate).

5.1 Design loads and combinations

The investigated steel frames are subjected to the following load combinations as stated by ASCE/SEI 7–10 (2010) specification:

(1) 1.4D
(2) 1.2D + 1.6L
(3) 1.2D + 1.0L ± 1.0E
(4) 0.9D ± 1.0E

where *D*, *L*, and *E* represent the dead, live, and earthquake loads, respectively.

The design gravity loads on the investigated steel frames are assumed as given in Table 1. The earthquake loads are calculated using the seismic coefficients listed in Table 2, and are applied to the frames in accordance with the equivalent lateral load procedure defined in ASCE/SEI 7–10 (2010) specification. The amplified inter-story drift is restricted to 2% of story height. A section pool consisting of 297 wide-flange AISC sections is used to size the members of the steel frames according to the provisions of ANSI/AISC 360–10 specification. Moreover, the design constraints are enforced as defined in Sect. 3.

5.2 Parametrizations of the optimization techniques

The metaheuristic search techniques are implemented with their recommended parameter settings given in the relevant papers, and they are listed in Table 3. It is important to note that for all the techniques employed the constraints are handled using the penalty function integrated objective function given by Eq. (15), where the penalty coefficient α_c is set to 1.0 for all the numerical examples considered here.

Table 1 Design gravity loads assumed for the steel frames

Load type	Uniformly distributed load	
	Interior beams (kN/m)	Exterior beams (kN/m)
Dead load (D)	24	12
Live load (L)	12	6

Table 2 Seismic coefficients used for calculating seismic loads acting on the steel frames

Seismic coefficients	
0.2 S Spectral Accel, Ss	2.29
0.1 S Spectral Accel, S_1	0.869
Long-period transition period	8
Site class	D
Site coefficient, F_a	1
Site Coefficient, F_v	1.5
$S_{DS} = (2/3)\ F_a\ S_s$	1.5267
$S_{D1} = (2/3)\ F_v\ S_1$	0.869

On the other hand, the GES algorithm is newly proposed and it possesses two parameters, namely GML and GMR, which are critical for the performance of the algorithm. It is important to mention that tuning these parameters for the best performance of the algorithm is clearly a challenging task and thus extensive numerical experiments have been conducted in a PhD thesis study by Korucu (2022) to determine appropriate values of these parameters on various problems including those under high dominance of drift constraints. In the following, the results of this study are briefly summarized with a particular emphasis on the recommended value sets of the GML and GMR parameters. For the complete details of the parameter sensitivity analyses, the interested reader is referred to the electronic copy of this study, which can be accessed online through Turkey Council of Higher Education (2023).

The results reveal that a parameter value of GMR = 0.5 (50%), where probabilistically half of the individuals are mutated according to the guided mutation and the other half according to the stochastic mutation, works most

Table 3 Parameter value settings for the employed metaheuristic algorithms

Optimization techniques	Parameter values
EBB–BC	Population size (μ) = 50
	Big bang constant (α) = 0.25
PSO	Swarm size (μ) = 50
	Time step parameter (Δt) = 1.0
	Inertia weight w = 0.5
	Particle trust parameter c_1 = 1.5
	Swarm trust parameter c_2 = 1.5
($\mu+\lambda$)-ES	Parent population size (μ) = 50
	Child population size (λ) = 50
	Initial mutation probability parameter $p^{(0)} = 0.25$
	Initial geometric distribution parameter $\psi^{(0)} = 10$

successfully for all kinds of problems. Therefore, this parameter is set to 0.50 (50%) in this study.

On the other side, to implement the GML parameter, various strategies have been attempted. First, the parameter is set to various constant values during an optimization process. Next, a dynamical execution of this parameter is adopted based on both a linearly decreasing function (Eq. 27) and an exponentially decreasing function (Eq. 28)

$$\mathrm{GML}_{t+1} = \mathrm{GML}_{\max} - \left(\mathrm{GML}_{\max} - \mathrm{GML}_{\min}\right)\left(\frac{t}{t_{\max}}\right) \quad (27)$$

$$\mathrm{GML}_{t+1} = \mathrm{GML}_{\max} - \left(\mathrm{GML}_{\max} - \mathrm{GML}_{\min}\right)e^{-\frac{10t}{t_{\max}}}, \quad (28)$$

where t is the current iteration number; $t_{\max}$ is the maximum number of iterations to be performed in an optimization process, $\mathrm{GML}_{\max}$ is a specified maximum value of the GML parameter at $t = 0$; $\mathrm{GML}_{\max}$ is a specified minimum value of the GML parameter at $t = t_{\max}$ and GML_{t+1} is the updated value of the parameter at the next iteration $(t + 1)$.

It should be noted that in a linearly decreasing scheme, there is a smooth decrement in the value of the GML parameter, whereas the decrement is rather sharp and rapid in the case of exponentially decreasing scheme. Among the three different schemes tested (i.e., constant, linearly decreasing, or exponentially decreasing) the best performance of the GES algorithm is obtained with the exponentially decreasing scheme of the related parameter (i.e., Eq. 28).

The optimal values of $\mathrm{GML}_{\max}$ and $\mathrm{GML}_{\min}$ in Eq. (28) are also investigated in a number of combinations by setting $\mathrm{GML}_{\max}$ to $1.0N_s$, $0.75\ N_s$, $0.50\ N_s$, $0.25\ N_s$, $0.15\ N_s$, $0.10\ N_s$, and setting $\mathrm{GML}_{\min}$ to $0.1\ N_s$, $0.05\ N_s$, and $0.03\ N_s$, where N_s is the number of sections in a section pool used to size the structural members. For example, when a section pool consisting of 297 wide-flange sections ($N_s = 297$) is used to size the members as followed in this study, then $\mathrm{GML}_{\max}$ is set to 297, 223, 149, 74, 45, and 30, whereas $\mathrm{GML}_{\min}$ is set to 30, 15, and 9. Accordingly, a total of seventeen combinations are generated for exponentially decreasing scheme of the GML parameter, as summarized in Table 4.

Regarding the GML parameter, the sensitivity analysis has been conducted using several test examples. In each example, the GMR parameter is set to 0.5, and the GML parameter is implemented by performing ten independent runs with each combination listed in Table 4, and the related performance assessments are carried out using the average of these results.

The results indicate that for problems where the optimum design is mainly controlled by strength constraints, the best performance of the GES algorithm is obtained with the $(0.15N_s, 0.05N_s)$ combination. On the hand, for problems in which displacement limitations are the dominant constraint type, the $(0.75N_s, 0.03N_s)$ combination usually yields the most successful results. It is important to emphasize that initial values of mutation probability and geometric distribution parameters in the GES algorithm are also set to $p^{(0)} = 0.25$ and $\psi^{(0)} = 10$.

Table 4 $\mathrm{GML}_{\max}$ and $\mathrm{GML}_{\min}$ combinations tested for exponentially decreasing scheme of the GML parameter

Combination	$\mathrm{GML}_{\max}$	$\mathrm{GML}_{\min}$
$(1.0N_s, 0.10\ N_s)$	297	30
$(1.0N_s, 0.05\ N_s)$	297	15
$(1.0N_s, 0.03\ N_s)$	297	9
$(0.75N_s, 0.10\ N_s)$	223	30
$(0.75N_s, 0.05\ N_s)$	223	15
$(0.75N_s, 0.03\ N_s)$	223	9
$(0.50N_s, 0.10\ N_s)$	149	30
$(0.50N_s, 0.05\ N_s)$	149	15
$(0.50N_s, 0.03\ N_s)$	149	9
$(0.25N_s, 0.10\ N_s)$	74	30
$(0.25N_s, 0.05\ N_s)$	74	15
$(0.25N_s, 0.03\ N_s)$	74	9
$(0.15N_s, 0.10\ N_s)$	45	30
$(0.15N_s, 0.05\ N_s)$	45	15
$(0.15N_s, 0.03\ N_s)$	45	9
$(0.10N_s, 0.05\ N_s)$	30	15
$(0.10N_s, 0.03\ N_s)$	30	9

Finally, in both the design optimization examples the maximum number of structural analyses performed is set to 3000 for all the optimization algorithms employed here in consideration of the UBS approach. In addition, the material properties of the steel are taken as follows: yield strength (F_y) = 50 ksi (~ 345 MPa), modulus of elasticity (E) = 29,000 ksi (~ 200 GPa) and unit weight (γ) = 490 lb/ft^3 (~ 76.97 kN/m^3).

5.3 Example 1: 160-member Space Steel Frame

The first design example refers to a 4-story, 160-member space steel frame (Fig. 5), which consists of 96 beams and 64 columns. The stability of the frame in both directions is provided with moment-resisting connections. For practical fabrication requirements, the 160 members of the frame are collected under twelve member groups. The columns are grouped into four different sizing variables in a plan level as the corner (CG_1), side plane y–z (CG_2), side plane x–z (CG_3), and inner (CG_4) columns (Fig. 5b). The columns are also grouped every two stories along the height of the frame such that the column groups over the first two (story 1–2) and last two (story 3–4) stories are required to have the same cross-section, resulting in a total of eight column member

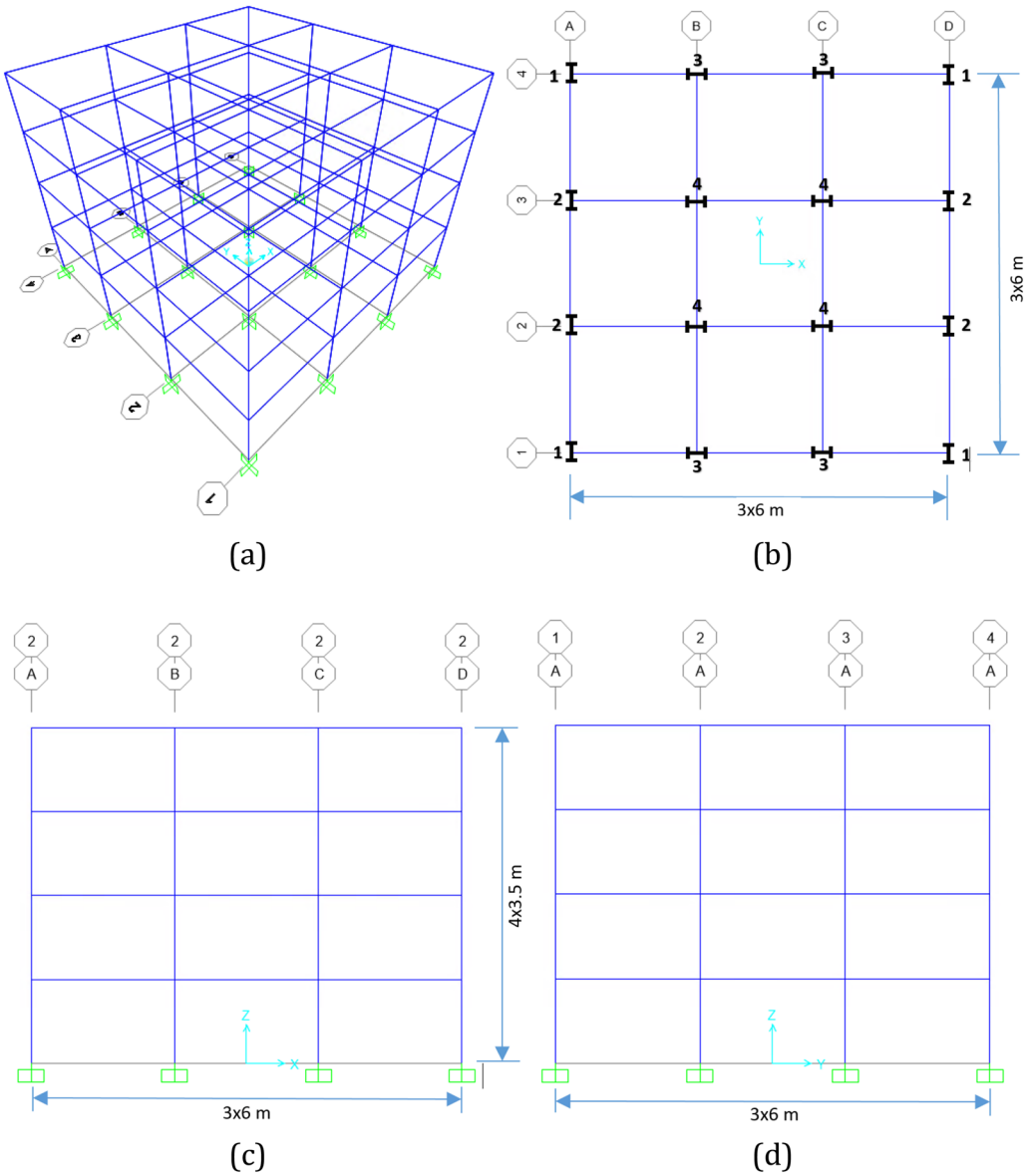

Fig. 5 160-member space steel frame: **a** 3-D view, **b** plan view, **c** side view in x–z plane, **d** side view in y–z plane

groups for the frame. Similarly, the beams are grouped into two distinct sizing variables in a plan level as inner (IB) and outer (OB) beams. The beams are also grouped every two stories along the height of the frame, yielding a total of four different beam member groups.

5.3.1 The first design case

In the study, the optimum design of the 160-member space steel frame is investigated under two different design cases. In the first design case, displacement constraints are not enforced and the frame is designed for the minimum weight under strength constraints only. Geometric constraints are also applied as formulated in Sect. 3.3.

Since metaheuristic search techniques have a stochastic nature, each run may end up with a different solution, especially for large and complex structural design optimization problems. Therefore, the frame is designed for the minimum weight by executing ten independent runs with each optimization technique to acquire statistically meaningful data required for performance evaluation. A comparison of the minimum weight designs (the best feasible designs) obtained using these techniques is carried out in Table 5, where the section designations assigned to all member groups are indicated along with the weights of the resulting designs. Among the results obtained with various optimization techniques shown in this table, the GES algorithm yields the least design weight (805.16 kN) for the frame,

Table 5 The optimum designs obtained for the 160-member space steel frame (under the first design case) using various optimization techniques

Stories	Group	PSO	EBB–BC	$(\mu+\lambda)$-ES	GES
1–2	IB	W24×76	W27×84	W24×68	W24×68
	OB	W16×40	W18×50	W16×40	W12×40
	CG_1	W24×55	W18×50	W18×86	W18×76
	CG_2	W30×99	W30×90	W18×106	W21×101
	CG_3	W33×221	W27×102	W24×207	W24×146
	CG_4	W21×182	W14×233	W27×235	W14×211
3–4	IB	W21×62	W21×62	W18×55	W18×55
	OB	W16×36	W14×34	W16×36	W16×40
	CG_1	W24×55	W18×50	W18×50	W18×50
	CG_2	W30×90	W30×90	W18×76	W21×101
	CG_3	W33×118	W27×102	W24×117	W24×104
	CG_4	W21×182	W14×145	W27×161	W14×132
Best design weight (kN)		861.14	837.18	843.86	805.16
Mean design weight (kN)		917.18	874.54	887.83	834.43
Worst design weight (kN)		974.07	909.61	942.23	849.90
Standard deviation		36.33	26.67	31.43	12.37
Coefficient of variation		3.96	3.05	3.54	1.48

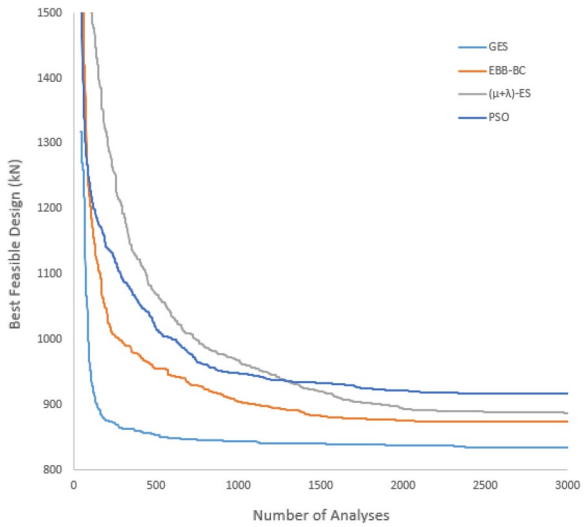

Fig. 6 Average convergence curves obtained for the 160-member space steel frame (under the first design case) using various optimization techniques

Table 6 The member groups' DCR values in the optimum designs of the 160-member space steel frame (under the first design case) produced by various optimization techniques

Stories	Group	PSO	EBB–BC	$(\mu+\lambda)$-ES	GES
1–2	IB	0.9941	0.8752	0.9995	0.9965
	OB	0.9857	0.9969	0.9961	0.9995
	CG_1	0.9873	0.9261	0.6234	0.7277
	CG_2	0.9239	0.9579	0.7499	0.9181
	CG_3	0.9187	0.9880	0.8034	0.8925
	CG_4	0.9988	0.9927	0.9151	0.9546
3–4	IB	0.9071	0.9021	0.9705	0.9642
	OB	0.9989	0.9962	0.9974	0.9998
	CG_1	0.7526	0.8491	0.7689	0.8138
	CG_2	0.8103	0.9104	0.7872	0.8608
	CG_3	0.9925	0.9802	0.8968	0.9970
	CG_4	0.9895	0.9630	0.9952	0.9969
Mean DCR		0.9383	0.9448	0.8753	0.9268
Max. DCR		0.9989	0.9969	0.9995	0.9998

and this design is obtained after implementing 696 structural analyses only at the best run of the algorithm, which takes a total computing time of 81 min with a PC having Intel i7-9750H, 6-core, 2.60 GHz processor, and 32 GB DDR4 RAM. The other techniques result in heavier design weights for the frame, namely 861.14 kN by PSO, 837.18 kN by EBB–BC, and 843.86 kN by $(\mu+\lambda)$-ES, even though they perform more structural analyses. A statistical evaluation of the independent runs for each technique is also provided in Table 5 in terms of the mean design weight, the worst design weight, standard deviation, and coefficient of variation.

The average convergence curves obtained with different optimization techniques during the optimum design process of the 160-member space steel frame under the first design case are plotted in Fig. 6. An average convergence curve shows variation of the best feasible design weight against the number of analyses based on the average of the ten optimization runs performed with each technique. The results presented in Table 5 and Fig. 6 evince that a more rapid, reliable, and effective design optimization process

Table 7 The x-direction inter-story drift values in the optimum designs of the 160-member space steel frame (under the first design case) produced by various optimization techniques

Stories	PSO	EBB–BC	$(\mu+\lambda)$-ES	GES
1	0.8325	0.9493	0.6177	1.0565
2	1.2863	1.4219	1.1844	1.6679
3	1.2807	1.5668	1.4419	1.7461
4	1.1962	1.4326	1.2285	1.2838
Average	1.1489	1.3426	1.1182	1.4386

Table 8 The y-direction inter-story drift values in the optimum designs of the 160-member space steel frame (under the first design case) produced by various optimization techniques

Stories	PSO	EBB–BC	$(\mu+\lambda)$-ES	GES
1	1.5181	1.4264	1.4630	1.4117
2	1.8294	1.6585	1.7779	1.7250
3	2.1832	1.9268	2.1782	2.1618
4	1.3717	1.2860	1.4821	1.4748
Average	1.7256	1.5744	1.7253	1.6933

is achieved with the GES algorithm in comparison to other techniques.

Table 6 presents the DCR (demand-to-capacity ratio) values for the member groups in the optimum designs of the 160-member space steel frame produced with different optimization techniques. Although not imposed, the inter-story drift ratios in the x- and y-directions are also given in Tables 7 and 8, respectively, for the completeness of the results.

Since only the strength constraints are applied in this design case (i.e., no displacement constraints are enforced), design variables are controlled by the associated DCR values. It is seen from Table 6 that the DCR values for the majority of design variables are close to their maximum value of 1.0. However, it is also worth mentioning that geometric constraints between the beam and column members presented in Sect. 3.3 are also considered. The fact that DCR values for some member groups are not assigned to values around 1.0 in the optimum designs can be attributed to the imposed geometric constraints. In order to satisfy these constraints, larger sections might be assigned to design variables, resulting in lower DCR values for some member groups.

5.3.2 The second design case

In this design case, the optimum design of 160-member space steel frame is investigated under both strength and displacement (deflection and inter-story drift) constraints. The geometric constraints are also imposed as formulated in Sect. 3.3.

The frame is designed for the minimum weight by executing ten independent runs with the GES algorithm as well as with each of the aforementioned metaheuristic search techniques. A comparison of the minimum weight designs (the best feasible designs) obtained using these optimization techniques is carried out in Table 9. Among the results obtained with various optimization techniques shown in this table, the GES algorithm yields the least design weight

Table 9 The optimum designs obtained for the 160-member space steel frame (under the second design case) using various optimization techniques

Stories	Group	PSO	EBB–BC	$(\mu+\lambda)$-ES	GES
1–2	IB	W30×90	W30×90	W30×99	W30×90
	OB	W21×50	W18×40	W14×30	W18×46
	CG_1	W21×62	W24×68	W33×118	W24×55
	CG_2	W27×102	W30×90	W30×99	W30×90
	CG_3	W40×244	W40×221	W40×221	W40×268
	CG_4	W40×328	W40×328	W40×298	W40×298
3–4	IB	W24×62	W24×62	W24×62	W24×62
	OB	W16×40	W24×55	W21×50	W18×46
	CG_1	W21×62	W24×55	W33×118	W24×55
	CG_2	W27×84	W30×90	W30×90	W30×90
	CG_3	W40×221	W40×221	W40×192	W40×221
	CG_4	W40×298	W40×298	W40×298	W40×298
Best design weight (kN)		1081.17	1079.11	1092.60	1073.76
Mean design weight (kN)		1140.99	1117.98	1162.90	1097.91
Worst design weight (kN)		1174.75	1142.92	1209.75	1149.05
Standard deviation		24.45	16.87	33.58	20.52
Coefficient of variation		2.14	1.51	2.89	1.87

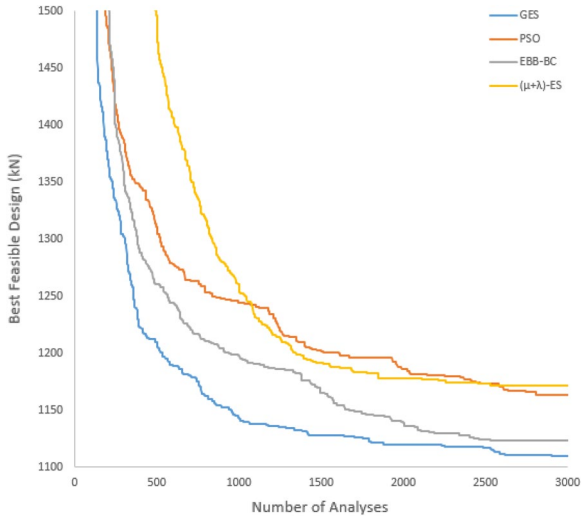

Fig. 7 Average convergence curves obtained for the 160-member space steel frame (under the second design case) using various optimization techniques

Table 10 The member groups' DCR values in the optimum designs of the 160-member space steel frame (under the second design case) produced by various optimization techniques

Stories	Group	PSO	EBB–BC	$(\mu+\lambda)$-ES	GES
1–2	IB	0.8106	0.8286	0.7810	0.8250
	OB	0.9970	0.9893	0.8738	0.8741
	CG_1	0.7056	0.7100	0.5609	0.9391
	CG_2	0.6477	0.7516	0.6729	0.7602
	CG_3	0.5973	0.6375	0.6510	0.5819
	CG_4	0.6152	0.6316	0.6671	0.6390
3–4	IB	0.9852	0.9948	0.9629	0.9868
	OB	0.7682	0.9659	0.9534	0.9191
	CG_1	0.6416	0.6298	0.4953	0.6622
	CG_2	0.6205	0.6402	0.6198	0.6395
	CG_3	0.4426	0.4455	0.5356	0.4306
	CG_4	0.4741	0.4653	0.4883	0.4793
Mean DCR		0.6921	0.7242	0.6885	0.7281
Max. DCR		0.9970	0.9948	0.9629	0.9868

(1073.76 kN) for the frame, and this design is obtained after implementing 1803 structural analyses only at the best run of the algorithm, which takes a total computing time of 240 min with a PC having Intel i7-9750H, 6-core, 2.60 GHz processor, and 32 GB DDR4 RAM. Relatively heavier design weights are produced for the frame with the other optimization techniques using a higher number of structural analyses, namely 1081.17 kN by PSO, 1079.11 kN by EBB–BC, and 1092.60 kN by $(\mu+\lambda)$-ES. For each optimization technique, the optimization statistics of the independent runs are also provided in Table 9, indicating that GES algorithm demonstrates a superior performance than the metaheuristic search techniques.

Average convergence curves obtained with different optimization techniques during the optimum design process of the 160-member space steel frame under the second design case are plotted in Fig. 7. It can be observed from this figure that the GES algorithm has improved convergence characteristics compared to the other implemented techniques.

In the optimum designs of the 160-member space steel frame (under the second design case), the computed DCR values for member groups, the inter-story drift ratios in the x-direction, and the inter-story drift ratios in the y-direction are given in Tables 10, 11, and 12, respectively. It is seen from these tables that mainly inter-story drift limitations (especially in the y-direction) are the dominant constraints in this design case.

Table 11 The x-direction inter-story drift constraint values in the optimum designs of the 160-member space steel frame (under the second design case) produced by various optimization techniques

Stories	PSO	EBB–BC	$(\mu+\lambda)$-ES	GES
1	0.4030	0.4435	0.4665	0.4269
2	0.8242	0.8739	0.9397	0.8378
3	0.9991	0.9145	0.9604	0.9918
4	0.9946	0.8365	0.8021	0.9490
Average	0.8052	0.7671	0.7922	0.8014

Table 12 The y-direction inter-story drift constraint values in the optimum designs of the 160-member space steel frame (under the second design case) produced by various optimization techniques

Stories	PSO	EBB–BC	$(\mu+\lambda)$-ES	GES
1	0.7947	0.8199	0.8347	0.8028
2	0.9794	0.9972	0.9875	0.9911
3	0.9945	0.9808	0.9950	0.9910
4	0.7838	0.7261	0.7387	0.7296
Average	0.8881	0.8810	0.8890	0.8786

5.4 Example 2: 584-member space steel frame

The second design example is an 8-story space steel frame (Fig. 8), which is composed of 584 structural members collected under 24 member groups. The frame is subjected to both strength and displacement constraints in addition to geometric limitations between the beam and column members. The member grouping is applied in both plan and elevation levels. At elevation level, the structural members are grouped every two stories along the height of the frame. At the plan level, the columns are collected under 4 different groups as the corner (CG_1), side plane y–z (CG_2), side

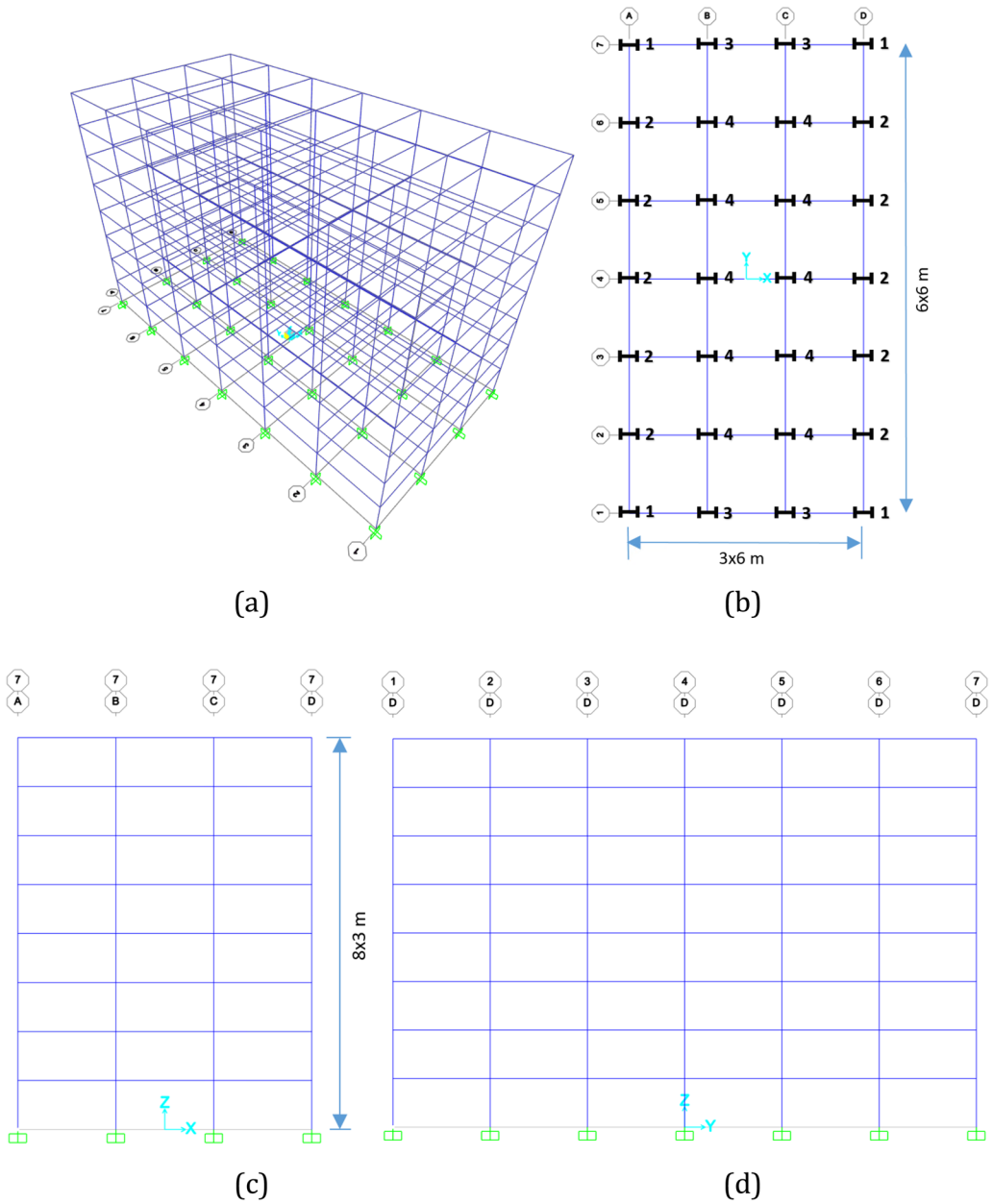

Fig. 8 584-member space steel frame: **a** 3-D view, **b** plan view, **c** side view in x–z plane, **d** side view in y–z plane

plane x–z (CG_3), and inner (CG_4) columns (Fig. 8b), and the beams are collected under two groups as inner (IB) and outer (OB) beams. Therefore, a total of sixteen column groups and eight beam groups (24 sizing design variables in all) are defined for the 584-member space steel frame example.

The frame is designed for the minimum weight by executing ten independent runs with the GES algorithm as well as with each of the aforementioned metaheuristic search techniques. A comparison of the minimum weight designs (the best feasible designs) obtained using these optimization techniques is carried out in Table 13. Among the results obtained with various optimization algorithms shown in this table, the GES algorithm yields the least design weight (4771.93 kN) for the frame, and this design is obtained after implementing 1970 structural analyses only at the best run of the algorithm, which takes a total computing time of 492 min with a PC having Intel i7-9750H, 6-core, 2.60 GHz processor, and 32 GB DDR4 RAM. The design weights that the other algorithms obtained are 5004.58 kN by PSO, 4877.17 kN by EBB–BC, and 4884.44 kN by $(\mu+\lambda)$-ES.

For each optimization technique, the optimization statistics of the independent runs are also provided in Table 13, indicating a superior performance of the GES algorithm in comparison to other methods. The average convergence

Table 13 The optimum designs obtained for the 584-member space steel frame using various optimization techniques

Stories	Group	PSO	EBB–BC	$(\mu+\lambda)$-ES	GES
1–2	IB	W33×118	W33×118	W30×99	W36×135
	OB	W24×68	W24×76	W24×76	W24×76
	CG_1	W33×201	W33×201	W30×173	W30×173
	CG_2	W40×244	W33×241	W40×221	W36×230
	CG_3	W33×221	W36×245	W30×326	W40×249
	CG_4	W36×359	W36×328	W36×439	W40×268
3–4	IB	W40×149	W33×118	W30×108	W33×118
	OB	W24×62	W24×76	W24×68	W24×68
	CG_1	W33×201	W33×201	W30×173	W30×173
	CG_2	W40×244	W33×221	W40×221	W36×230
	CG_3	W33×221	W33×221	W30×326	W40×215
	CG_4	W36×300	W36×328	W36×393	W40×268
5–6	IB	W30×99	W30×90	W30×90	W27×84
	OB	W24×62	W24×62	W24×55	W24×55
	CG_1	W33×201	W33×118	W30×132	W30×173
	CG_2	W40×192	W33×221	W40×221	W33×201
	CG_3	W33×221	W33×201	W30×235	W40×199
	CG_4	W36×245	W36×280	W36×300	W40×268
7–8	IB	W24×55	W21×50	W24×55	W24×62
	OB	W14×30	W14×34	W18×35	W18×35
	CG_1	W33×201	W33×118	W30×90	W30×90
	CG_2	W40×192	W33×201	W40×192	W33×201
	CG_3	W33×118	W33×201	W30×173	W40×199
	CG_4	W36×230	W33×221	W33×201	W40×192
Best design weight (kN)		5004.58	4877.17	4884.44	4771.93
Mean design weight (kN)		5481.44	5272.89	5163.69	5017.01
Worst design weight (kN)		6522.44	5462.96	5540.25	5221.91
Standard deviation		421.88	153.87	187.26	153.18
Coefficient of variation		7.70	2.92	3.63	3.05

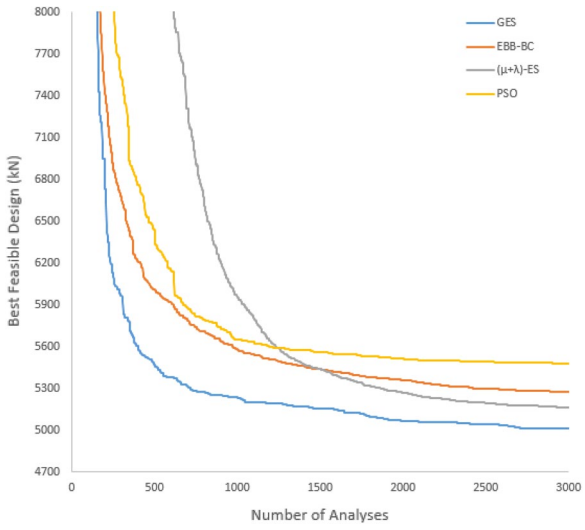

Fig. 9 Average convergence curves obtained for the 584-member steel frame design example using various optimization techniques

curves obtained with different optimization techniques during the optimum design process of the 584-member space steel frame are plotted in Fig. 9. It can be observed from this figure that the GES algorithm outperforms the other techniques in terms of having a faster convergence rate towards the optimum solution.

In the optimum designs of the 584-member steel frame, the computed DCR values for member groups, the inter-story drift ratios in the *x*-direction, and the inter-story drift ratios in the *y*-direction are given in Tables 14, 15, and 16, respectively. It is seen from these tables that like the second design case of the previous example, mainly inter-story drift constraints (especially in the *y*-direction) are the dominant constraints in this example as well.

Table 14 The member groups' DCR values in the optimum designs of the 584-member space steel frame produced by various optimization techniques

Stories	Group	PSO	EBB–BC	$(\mu+\lambda)$-ES	GES
1–2	IB	0.5840	0.5831	0.6616	0.5361
	OB	0.6869	0.5420	0.5613	0.5505
	CG_1	0.5360	0.5702	0.6286	0.6456
	CG_2	0.6791	0.6235	0.7079	0.6493
	CG_3	0.8375	0.8247	0.6323	0.8434
	CG_4	0.7741	0.8161	0.6138	0.9446
3–4	IB	0.5472	0.6191	0.6861	0.6404
	OB	0.9735	0.6158	0.6476	0.6698
	CG_1	0.4129	0.4623	0.4853	0.4774
	CG_2	0.4859	0.5808	0.5623	0.5454
	CG_3	0.7475	0.7054	0.4723	0.7646
	CG_4	0.7630	0.6749	0.5654	0.8345
5–6	IB	0.6572	0.6664	0.6780	0.6920
	OB	0.9546	0.7800	0.8851	0.9056
	CG_1	0.3781	0.5775	0.5066	0.4181
	CG_2	0.5979	0.5134	0.4925	0.5219
	CG_3	0.6402	0.6149	0.5200	0.6336
	CG_4	0.7615	0.6545	0.6002	0.6731
7–8	IB	0.9074	0.9188	0.8418	0.7682
	OB	0.8749	0.6857	0.9465	0.9652
	CG_1	0.2792	0.4803	0.5157	0.4797
	CG_2	0.4400	0.3866	0.4153	0.3723
	CG_3	0.6048	0.4654	0.5110	0.4815
	CG_4	0.5649	0.5367	0.5830	0.6335
Mean DCR		0.6537	0.6207	0.6133	0.6519
Max. DCR		0.9735	0.9188	0.9465	0.9652

Table 15 The *x*-direction inter-story drift constraint values in the optimum designs of the 584-member space steel frame produced by various optimization techniques

Stories	PSO	EBB–BC	$(\mu+\lambda)$-ES	GES
1	0.3406	0.3066	0.3011	0.2906
2	0.6953	0.5585	0.5909	0.5655
3	0.8503	0.6766	0.6987	0.7036
4	0.8960	0.7094	0.7447	0.7557
5	0.8755	0.7398	0.7770	0.7705
6	0.8466	0.7386	0.7572	0.7567
7	0.9078	0.7684	0.7553	0.7502
8	0.9324	0.7892	0.7026	0.7052
Average	0.7931	0.6609	0.6659	0.6622

Table 16 The *y*-direction inter-story drift constraint values in the optimum designs of the 584-member space steel frame produced by various optimization techniques

Stories	PSO	EBB–BC	$(\mu+\lambda)$-ES	GES
1	0.8033	0.8224	0.7367	0.8344
2	0.9793	0.9950	0.9896	0.9881
3	0.9911	0.9977	0.9967	0.9934
4	0.9667	0.9326	0.9555	0.9642
5	0.9967	0.9917	0.9953	0.9923
6	0.9328	0.8955	0.8882	0.9449
7	0.9915	0.9919	0.9993	0.9829
8	0.9154	0.8581	0.7103	0.6866
Average	0.9471	0.9356	0.9089	0.9233

6 Conclusions

In this paper, a time-effective optimum design of space steel frame structures is investigated using a new design-driven hybrid optimization technique called guided evolution strategy (GES). The rationale behind the proposed technique is to improve and facilitate the convergence characteristics of an evolution strategy by guiding the search process based on satisfaction/violation of strength constraints in a previous design. Conventionally, a stochastic mutation scheme is still followed by the GES algorithm in order to take advantage of a randomized search process of the design space, yet this conventional mutation scheme is supplemented by the guided mutation scheme, which makes use of the design-driven information obtained during the search process to accelerate the convergence speed of the algorithm.

The effectiveness of the optimization process with the GES algorithm has been investigated and compared to some selected metaheuristic search techniques using two numerical examples related to discrete sizing design optimization of space steel frames. The upper bound strategy (UBS), which significantly reduces the number of structural analyses during an optimization process, is implemented for all the techniques for an objective comparison of the performances. The maximum number of iterations (structural analyses) is limited to 3000 in all the examples. The performance of the GES algorithm is compared to those of the metaheuristic search techniques in terms of the quality of the optimum solution as well as the speed of convergence to an optimum solution. The GES algorithm has produced the best feasible solutions to these two design examples.

The optimization statistics of the independent runs as well as convergence curves plotted using the average performances of the techniques manifest the success of the GES algorithm. Despite the fact that the GES algorithm provides guidance related to the strength constraints of the member groups only, a superior performance of this algorithm is also observed for problems where the steel frames are under high inter-story drift domination design constraints. The success of the algorithm can be attributed to the fact that when GMR

is set to 0.50 (50%), half of the individuals that are treated according to the guided mutation scheme rapidly evolve designs that satisfy strength constraints effectively, whereas the other is half mutated according to the stochastic mutation scheme mainly handles geometric and inter-story constraints for a problem under consideration. This way, a fast and effective search of the design space is accomplished in all stages of an optimization process.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00158-023-03640-7.

Acknowledgements The authors would like to thank Dr. Hasan Eser from Middle East Technical University for providing the finite element models of the steel frames used in this study.

Declarations

Conflict of interest The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Replication of results The data for producing the presented results will be made available upon request.

References

Ahrari A, Atai AA (2013) Fully stressed design evolution strategy for shape and size optimization of truss structures. Comput Struct 123:58–67

Ahrari A, Deb K (2016) An improved fully stressed design evolution strategy for layout optimization of truss structures. Comput Struct 164:127–144

Ahrari A, Atai AA, Deb K (2015) Simultaneous topology, shape and size optimization of truss structures by fully stressed design based on evolution strategy. Eng Optim 47(8):1063–1084

ANSI, AISC 360-10 (2010) Specification for structural steel buildings. ANSI, AISC 360-10, Illinois

ASCE, SEI 7–10 (2010) Minimum design loads for buildings and other structures. ASCE, SEI 7–10, Reston

Aydoğdu İ, Akın A, Saka MP (2016) Design optimization of real-world steel space frames using artificial bee colony algorithm with levy flight distribution. Adv Eng Softw 92:1–14

Bäck T, Schütz M (1995) Evolutionary strategies for mixed-integer optimization of optical multilayer systems. In: McDonnel JR, Reynolds RG, Fogel DB (eds) Proceedings of the fourth annual conference on evolutionary programming. MIT Press, Cambridge, pp 33–51

Cai J, Thierauf G (1993) Discrete structural optimization using evolution strategies. In: Topping BHV, Khan AI (eds) Neural networks and combinatorial optimization in civil and structural engineering. Civil-Comp, Edinburgh, pp 95–100

Degertekin SO, Tutar H (2022) Optimized seismic design of planar and spatial steel frames using the hybrid learning based Jaya algorithm. Adv Eng Softw 171:103172

Elvin A, Walls R, Cromberge D (2009) Optimising structures using the principle of virtual work. J S Afr Inst Civil Eng 51(2):11–19

Erol OK, Eksin I (2006) A new optimization method: big bang–big crunch. Adv Eng Softw 37(2):106–111

Gallagher RH, Zienkiewicz OC (1973) Optimum structural design: theory and applications. Wiley, London

Gholizadeh S, Baghchevan A (2017) Multi-objective seismic design optimization of steel frames by a chaotic meta-heuristic algorithm. Eng Comput 33(4):1045–1060

Gholizadeh S, Fattahi F (2014) Design optimization of tall steel buildings by a modified particle swarm algorithm. Struct Des Tall Spec Build 23(4):285–301

Gholizadeh S, Milany A (2018) An improved fireworks algorithm for discrete sizing optimization of steel skeletal structures. Eng Optim 50(11):1829–1849

Hasançebi O (2007) Discrete approaches in evolution strategies based optimum design of steel frames. Struct Eng Mech 26(2):191–210

Hasançebi O, Kazemzadeh Azad S (2012) An exponential big bang-big crunch algorithm for discrete design optimization of steel frames. Comput Struct 110–111:167–179

Hasançebi O, Çarbaş S, Doğan E, Erdal F, Saka MP (2010) Comparison of non-deterministic search techniques in the optimum design of real size steel frames. Comput Struct 88(17–18):1033–1048

Hasançebi O, Bahçecioğlu T, Kurç Ö, Saka MP (2011) Optimum design of high-rise steel buildings using an evolution strategy integrated parallel algorithm. Comput Struct 89(21–22):2037–2051

Kashani AR, Camp CV, Rostamian M, Azizi K, Gandomi AH (2022) Population-based optimization in structural engineering: a review. Artif Intell Rev 55:345–452

Kaveh A, Bakhshpoori T, Azimi M (2015) Seismic optimal design of 3D steel frames using cuckoo search algorithm. Struct Des Tall Special Build 24(3):210–227

Kazemzadeh Azad S (2021) Design optimization of real-size steel frames using monitored convergence curve. Struct Multidisc Optim 63(1):267–288

Kazemzadeh Azad S, Hasançebi O (2015) Computationally efficient discrete sizing of steel frames via guided stochastic search heuristic. Comput Struct 156:12–28

Kazemzadeh Azad S, Hasançebi O, Kazemzadeh Azad S (2013) Upper bound strategy for metaheuristic-based design optimization of steel frames. Adv Eng Softw 57:19–32

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks. IEEE Press, vol 4, pp 1942–1948

Korucu A (2022) Developing a structural optimization software for efficient and practical optimum design of real-world steel structures. PhD Thesis, Middle East Technical University, Ankara

Lamberti L, Pappalettere C (2011) Metaheuristic design optimization of skeletal structures: a review. Comput Technol Rev 4:1–32

Murren P, Khandelwal K (2014) Design-driven harmony search (DDHS) in steel frame optimization. Eng Struct 59:798–808

Nouhi B, Jahani Y, Talatahari S, Gandomi AH (2022) A swarm optimizer with modified feasible-based mechanism for optimum structure in steel industry. Decis Anal J 5:9

Patnaik SN, Berke L, Gallagher RH (1991) Integrated force method versus displacement method for finite element analysis. Comput Struct 38(4):377–407

Patnaik SN, Gendy AS, Berke L, Hopkins DA (1998) Modified fully utilized design (MFUD) method for stress and displacement constraints. Int J Numer Methods Eng 41(7):1171–1194

Razani R (1965) Behavior of fully stressed design of structures and its relationship to minimum-weight design. AIAA J 3(12):2262–2268

Rechenberg I (1965) Cybernetic solution path of an experimental problem. Royal Aircraft Establishment Library Translation 1122

Rechenberg I (1973) Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution [Evolution strategy: optimization of technical systems according to the principles of biological evolution]. Frommann-Holzboog Verlag, Stuttgart

Renkavieski C, Parpinelli RS (2021) Meta-heuristic algorithms to truss optimization: Literature mapping and application. Expert Syst Appl 182:115197

Rudolph G (1994) An evolutionary algorithm for integer programming. In: Davidor Y, Schwefel H-P, Manner R (eds) Proceedings of third conference on parallel problem solving from nature. Springer, Heidelberg, pp 139–148

Sadrekarimi N, Talatahari S, Azar BF, Gandomi AH (2023) A surrogate merit function developed for structural weight optimization problems. Soft Comput 27(3):1533–1563

Saka MP (2007) Optimum design of steel frames using stochastic search techniques based on natural phenomena: a review. In: Topping BHV (ed) Civil engineering computations: tools and techniques. Saxe-Coburg Publications, Stirlingshire, pp 105–147

Saka MP, Geem ZW (2013) Mathematical and metaheuristic applications in design optimization of steel frame structures: An extensive review. Math Probl Eng 2013:271031

Schwefel H-P (1965) Kybernetische Evolution als Strategie der Experimentellen Forschung in der Strömungstechnik. Diplomarbeit, Technische Universität, Berlin

Schwefel H-P (1977) Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie. Vol. 26, Interdisciplinary system research. Birkhäuser Verlag, Basel

Schwefel H-P (1981) Numerical optimization of computer models. Wiley, Chichester

Tabak EI, Wright PM (1981) Optimality criteria method for building frames. J Struct Div 107(7):1327–1342

Talatahari S, Azizi M (2020) Optimal design of real-size building structures using quantum-behaved developed swarm optimizer. Struct Des Tall Spec Build 29(11):e1747

Talatahari S, Jalili S, Azizi M (2021) Optimum design of steel building structures using migration-based vibrating particles system. Struct 33:1394–1413

Talatahari S, Veladi H, Azizi M, Moutabi-Alavi A, Rahnema S (2022) Optimum structural design of full-scale steel buildings using drift-tribe-charged system search. Earthq Eng Eng Vib 21(3):825–842

Turkey Council of Higher Education (2023) Thesis Center, https://tez.yok.gov.tr/UlusalTezMerkezi/tarama.jsp. Accessed 25 Apr 2023

Walls R, Elvin A (2010) Optimizing structures subject to multiple deflection constraints and load cases using the principle of virtual work. J Struct Eng 136(11):1444–1452

Yang XS (2010) Nature-inspired metaheuristic algorithms. Luniver Press, Beckington