Structural and Multidisciplinary Optimization (2024) 67:203
https://doi.org/10.1007/s00158-024-03898-5

RESEARCH PAPER

An enhanced guided stochastic search with repair deceleration mechanism for very high-dimensional optimization problems of steel double-layer grids

Saeid Kazemzadeh Azad[1] · **Saman Aminbakhsh**[1] · **Amir H. Gandomi**[2,3]

Received: 6 September 2023 / Revised: 4 September 2024 / Accepted: 27 September 2024

Abstract
Finding reasonably good solutions using a fewer number of objective function evaluations has long been recognized as a good attribute of an optimization algorithm. This becomes more important, especially when dealing with very high-dimensional optimization problems, since contemporary algorithms often need a high number of iterations to converge. Furthermore, the excessive computational effort required to handle the large number of design variables involved in the optimization of large-scale steel double-layer grids with complex configurations is perceived as the main challenge for contemporary structural optimization techniques. This paper aims to enhance the convergence properties of the standard guided stochastic search (GSS) algorithm to handle computationally expensive and very high-dimensional optimization problems of steel double-layer grids. To this end, a repair deceleration mechanism (RDM) is proposed, and its efficiency is evaluated through challenging test examples of steel double-layer grids. First, parameter tuning based on rigorous analyses of two preliminary test instances is performed. Next, the usefulness of the proposed RDM is further investigated through two very high-dimensional instances of steel double-layer grids, namely a 21,212-member free-form double-layer grid, and a 25,514-member double-layer multi-dome, with 21,212 and 25,514 design variables, respectively. The obtained numerical results indicate that the proposed RDM can significantly enhance the convergence rate of the GSS algorithm, rendering it an efficient tool to handle very high-dimensional sizing optimization problems.

Keywords Structural optimization · High-dimensional optimization · Steel double-layer grids · Discrete sizing · Guided stochastic search · Optimization algorithm

1 Introduction

Over the past few decades, several algorithms have emerged in the literature regarding structural optimization for the optimal design of steel skeletal structures. Usually, contemporary approaches rely on gradient-free evolutionary optimization methodologies (Saka and Topping 2007; Lamberti and Pappalettere 2011; Kashani et al. 2022). This is due to the shortcomings of conventional structural optimization methods, i.e., mathematical programming (Erbatur and Al-Hussainy 1992) and optimality criteria (Tabak and Wright 1981; Saka 1990), such as dependency on gradient information and inefficiency in handling high-dimensional discrete optimization problems. In this context, Babaei and Sheidaii (2014) employed particle swarm optimization for the automated optimum design of double-layer latticed domes. The authors used parametric mathematical functions for the generation of geometrical configurations of the dome structures, as well as the handling of geometry variables. The usefulness of the approach was illustrated using sizing and geometry optimization examples of steel double-layer grids, specifically a 1479-member double-layer ribbed dome and a 1488-member double-layer lamella dome, under a single gravity load case.

Responsible editor: Mehmet Polat Saka.

✉ Amir H. Gandomi
gandomi@uts.edu.au

[1] Department of Civil Engineering, Atilim University, Ankara, Turkey

[2] Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW, Australia

[3] University Research and Innovation Center (EKIK), Óbuda University, 1034 Budapest, Hungary

Published online: 27 November 2024

Considering the effect of uncertainties in applied loads, Mashayekhi et al. (2012) proposed a two-stage optimization algorithm for reliability-based topology optimization of double-layer grids. The authors used the method of moving asymptotes (MMA), together with the well-known ant colony optimization (ACO) algorithm, developing the so-called MMA-ACO technique. In this approach, first, structural stiffness was maximized using the MMA, where more effective members that increase the structural stiffness were identified. The obtained results were then employed to guide the ACO algorithm. The numerical results revealed the efficiency of the two-stage MMA-ACO algorithm in the reliability-based topology optimization of double-layer grids. Later, Torkzadeh et al. (2015) performed simultaneous sizing, geometry, and topology optimization of steel double-layer grids, by using a modified genetic algorithm based on fuzzy inference system. The authors compared their optimum designs to those already achieved by the MMA-ACO algorithm (Mashayekhi et al. 2012).

Handling the excessive computational effort is the most obvious challenge in large-scale structural optimization using gradient-free evolutionary algorithms. The use of artificial neural networks (ANNs) for approximate structural response prediction (Hajela and Berke 1991; Kaveh and Servati 2002), together with evolutionary algorithms has long been recognized as an efficient method to reduce the computational cost of large-scale structural optimization problems. Rajasekaran (2001) integrated a cellular genetic algorithm with ANNs for the optimum design of large-scale three-dimensional reticulated structures and illustrated its applicability using different test examples, which included the barrel vault structure for the platform shelter roof of Thirumyilai railway station (Ramaswamy 1997). In the proposed approach, structural analyses were carried out up to two or three levels of optimization to gather sufficient data for training the ANNs. Next, the structural analysis stage was replaced by ANNs for computational efficiency.

Papadrakakis et al. (1999) developed a hybrid optimization technique, which was based on the combination of evolution strategies and ANNs, for large-scale sizing optimization of space truss structures. In their approach, the structural analysis phase was replaced by an ANN prediction to enhance the computational efficiency. The performance of the proposed hybrid method was assessed by using challenging test examples of double-layer grids, which included an 8000-member cylindrical double-layer space truss roof with 48 design variables. It was concluded that the developed hybrid optimization methodology could find promising solutions for both large and computationally expensive optimization instances, with a considerably lower computational cost, when compared to the standard evolution strategies and conventional approaches based on mathematical programming.

Gholizadeh et al. (2013) combined evolutionary algorithms with ANNs for sizing and geometry optimization of double-layer grids subjected to gravity and earthquake loads. For optimization, a variant of the particle swarm optimization algorithm known as the quantum-behaved particle swarm optimization technique was employed. Besides, to reduce the computational effort required for time history analysis, radial basis function neural networks were used to predict the structural responses of double-layer grids over the optimization iterations. The numerical results demonstrated the computational efficiency of the proposed technique to be applied in the optimum design of steel double-layer grids.

Kaveh and Moradveisi (2020) used an enhanced variant of colliding bodies optimization method for the simultaneous sizing and geometry optimization of double-layer grids considering both material and geometrical nonlinearities. To achieve minimum weight designs, member groups that are associated with cross-sectional areas of members were considered as sizing variables, while the height of the double-layer grid was treated as a geometry variable. The authors investigated simultaneous sizing and geometry optimization of different double-layer grid instances with up to 1520 members and 28 design variables.

Another methodology to handle the optimization of large-scale structural systems is to divide the associated large structural optimization problems into smaller problems that can be solved in parallel. Research work on decomposition methods for structural optimization, such as substructuring techniques, has been well documented in the literature (Spengemann and Thierauf 1991; Patnaik et al. 1994). Papadrakakis et al. (2003) studied the efficiency of parallel domain decomposition algorithms integrated with evolution strategies and genetic algorithms in the optimum design of large-scale space truss systems. For a computationally efficient sizing optimization, the configuration of a 12,974-member long-span multi-layer space truss structure was decomposed into 12 subdomains, where the computational time for the optimization algorithms was reported for both serial and parallel computing environments. Similarly, configuration of a large-scale double-layer space truss structure was partitioned into 48 subdomains for computationally efficient sizing, geometry, and topology optimization. The study highlighted the computational advantages of domain decomposition algorithms, as well as parallel computing strategies in large-scale structural optimization problems.

To overcome the difficulty of solving complex optimization instances, Patnaik et al. (1997) implemented a cascade optimization strategy, where different optimizers were used in a successive manner to locate a reasonably good solution. In this type of methodology, each optimization stage was initiated using the solution obtained in the preceding stage, with some pseudo-random perturbations. Lagaros

et al. (2005) developed a non-dominant cascade evolutionary algorithm for multi-objective structural optimization and demonstrated its efficiency using real-scale transmission tower and pedestrian truss bridge test examples. Later, Kaveh and Ilchi Ghazaan (2016) combined cascade optimization strategy with an enhanced variant of colliding bodies optimization technique to solve optimization problems of large-scale truss towers with up to 2386 structural elements and 220 design variables.

Despite the widespread use of evolutionary algorithms in the literature on structural optimization, their need for excessive computational work in large-scale problems has encouraged researchers to seek more computationally efficient alternatives. Design-driven methods (Flager et al. 2014; Murren and Khandelwal 2014; Ahrari and Deb 2016), which use domain knowledge to guide the optimization process, have been found to be effective means to achieve this goal. A recent case study on design-driven optimization of a stadium space frame roof structure, with 1955 members aggregated into 34 groups, has been provided by Flager et al. (2014). Using a design-driven method, which is called fully constrained design (FCD), the authors reported a cost saving of 19% compared to conventional design methods for the investigated stadium roof structure. The study also revealed that the total required time by the proposed FCD technique, including setup, was about 20% less than that of conventional practice.

Guided stochastic search (GSS) (Kazemzadeh Azad et al. 2014a) is a discrete sizing optimization technique that involves both design-oriented and probabilistic search features. Through the advantageous formulation of the integrated force method (IFM) (Patnaik et al. 1991), particularly in providing sensitivity information for handling displacement constraints, two different versions of the method, i.e., GSS_A and GSS_B, have been developed (Kazemzadeh Azad and Hasançebi 2015). These methods were adopted to cope with the so-called curse of dimensionality (Bellman 1957) in challenging discrete sizing optimization problems with a high number of design variables (Kazemzadeh Azad and Aminbakhsh 2021, 2022).

With respect to the previous promising performance of the GSS when handling large-scale structural optimization problems, here, the aim is to further enhance its converge properties to deal with computationally expensive and very high-dimensional structural optimization instances of steel double-layer grids. To this end, a repair deceleration mechanism (RDM) is proposed, with its efficiency being assessed via challenging structural optimization examples of steel double-layer grids. First, a comprehensive preliminary evaluation of the proposed approach is accomplished via two high-dimensional test instances from the literature. Once a suitable parameter setting is obtained based on rigorous analyses of the preliminary test instances, the applicability of the proposed RDM is further demonstrated using two very high-dimensional examples of steel double-layer grids; a 21,212-member free-form double-layer grid, and a 25,514-member double-layer multi-dome, which have 21,212 and 25,514 design variables, respectively.

The organization of this manuscript is as follows. Section 2 describes the sizing optimization problem of steel double-layer grids. Section 3 outlines the main optimization steps required for the implementation of the GSS algorithm. The proposed repair deceleration mechanism is elaborated in Sects. 4, and 5 presents the numerical examples of steel double-layer grids with a considerably large number of design variables. Future research directions are outlined in Sect. 6. Finally, the concluding remarks are summarized in the last section.

2 Optimization problem formulation

The sizing optimization of steel double-layer grids aims to find an integer vector $\boldsymbol{X}$, which indicates the designation numbers or labels of standard steel sections assigned to the N_m number of grid members, to minimize the design weight of the structure. This process can be expressed as follows.

$$\text{Find}: \ \boldsymbol{X}^T = [x_1, x_2, \ldots, x_{N_m}] \tag{1}$$

$$\text{to minimize}: \ W(\boldsymbol{X}) = \sum_{i=1}^{N_m} \rho_i L_i A_i \tag{2}$$

where $W(\boldsymbol{X})$ is the design weight of the double-layer grid, and ρ_i, L_i, A_i are unit weight, length, and cross-sectional area of the *i-th* member, respectively. Meanwhile, the challenge is to achieve a lightweight design that suits a set of strength and displacement constraints as follows.

$$g_i(\boldsymbol{X}) \le 0 \quad i = 1, 2, \ldots, N_m \tag{3}$$

$$h_j(\boldsymbol{X}) \le 0 \quad j = 1, 2, \ldots, N_d \tag{4}$$

where $g_i(\mathbf{X})$ and $h_j(\mathbf{X})$ are the strength and displacement constraints, respectively; and N_d is the total number of active degrees of freedom. In this study, the design constraints are imposed with respect to AISC-LRFD (American Institute of Steel Construction (AISC) 1994) specification. Further details on the above-mentioned design optimization constraints can be found in Kazemzadeh Azad and Aminbakhsh (2021) and American Institute of Steel Construction (AISC) (1994), and will not be covered here.

3 An outline of the GSS algorithm

GSS algorithm is a computationally efficient sizing optimization method, which is capable of handling numerous discrete variables involved in the optimum design of large-scale structural systems. In addition to the two conventional variants of the GSS algorithm, namely GSS_A and GSS_B (Kazemzadeh Azad and Hasançebi 2015; Kazemzadeh Azad and Aminbakhsh 2021), an improved reformulation of the technique has recently been proposed in Kazemzadeh Azad and Aminbakhsh (2022) that can be easily used for handling both single- and multi-objective sizing optimization problems of large-scale steel double-layer grids. To distinguish, the single-objective variant of the latter is denoted by GSS_C in the present work. Considering the promising performance of the GSS_C already shown in Kazemzadeh Azad and Aminbakhsh (2022), the primary steps for implementation of the GSS algorithm are described below based on this variant.

Step 1. Initialization: The optimization process starts by randomly proposing a single candidate solution, which will be evaluated in the succeeding steps to determine the corresponding solution quality.

Step 2. Evaluation under real loading: Structural response computations are accomplished in this step for the newly produced candidate design under the exerted real loads so that possible constraint violations can be captured.

Step 3. Evaluation under virtual loading: For efficient handling of the displacement constraints over optimization iterations, it is necessary to obtain sensitivity index (SI) values (Kazemzadeh Azad and Hasançebi 2015; Charney 1993) for double-layer grid members. For this purpose, an additional structural analysis is typically needed at this step, so that the structural response under the virtual loads can be determined. However, as illustrated in Kazemzadeh Azad and Hasançebi (2015), it is possible to circumvent this difficulty by employing the IFM (Patnaik et al. 1991), so that the virtual internal forces of grid members can be computed, based entirely on the matrix structural analysis already done in the preceding step.

Step 4. Detection of critical members: Essentially, eliminating the constraint violations, as well as reducing the design weight, are the main concerns during the structural optimization process. In this context, two different groups of double-layer grid members need to be identified for the resizing operation. The first group comprises those members that will be upsized to alleviate the constraint violations. Meanwhile, the second group comprises those grid members that will be downsized for weight minimization. Assignment of the critical members to the aforementioned two groups is accomplished based on the following procedure.

Step 4.1. Increase-group for constraint satisfaction: A group named as increase-group (IG) is dedicated to holding those grid members that are to be increased in size for the sake of alleviating the constraint violations. IG is subdivided into two subgroups of IG_s and IG_d, which satisfy strength and displacement constraints, respectively. Accordingly, those grid members that are in violation of the strength constraints are promptly assigned to the IG_s subgroup. Identification of such grid members can easily be performed with regard to the computed load-to-capacity ratios (*LCR*s).

Identification of IG_d members, on the other hand, is achieved using the computed SI values. As per this approach, members featuring greater SI values constitute the most critical members for handling the displacement constraints. The size of IG_d subgroup is set with respect to R_d parameter (see step 6), an adaptive ratio that regulates the percentage of the design variables involved in resizing.

Step 4.2. Decrease-group for weight reduction: Overdesigned grid members, where the size reduction is anticipated to yield a relatively minor effect on the displacement response, are collected in the second member group, which is named as decrease-group (DG_w). Similar to IG_d, the size of DG_w is also specified based on an adaptive ratio parameter, i.e., R_w (see step 6).

Step 5. Stochastic member resizing: Bounded by a maximum incremental step size, the algorithm follows a stochastic pattern to randomly upsize the members of IG_s and IG_d subgroups. The core idea is to resize the IG members through a random move towards larger steel sections using the following relation:

$$I_i^{\text{new}} = I_i^{\text{pre}} + \text{Rand}^{\text{IG}} \tag{5}$$

where I_i^{pre} is the value of a design variable in the preceding iteration, I_i^{new} is the new value of the variable, denoting the sequence number of the new section chosen for an IG member in the current iteration, and Rand^{IG} is a random integer generated between 1 and the maximum incremental step size according to a uniform distribution.

In contrast, members of DG_w are downsized stochastically, considering a maximum decremental step size. In this regard, a random move towards smaller sections is accomplished for resizing DG_w members as follows:

$$I_i^{\text{new}} = I_i^{\text{pre}} - \text{Rand}^{\text{DG}_w} \tag{6}$$

where $\text{Rand}^{\text{DG}_w}$ is a random integer generated between 1 and the maximum decremental step size with respect to a uniform distribution. Recently, it has been proposed to set the above-mentioned maximum incremental/decremental step size randomly (Kazemzadeh Azad and Aminbakhsh 2022). Accordingly, the associated maximum step size for each structural member chosen for resizing is randomly produced as follows:

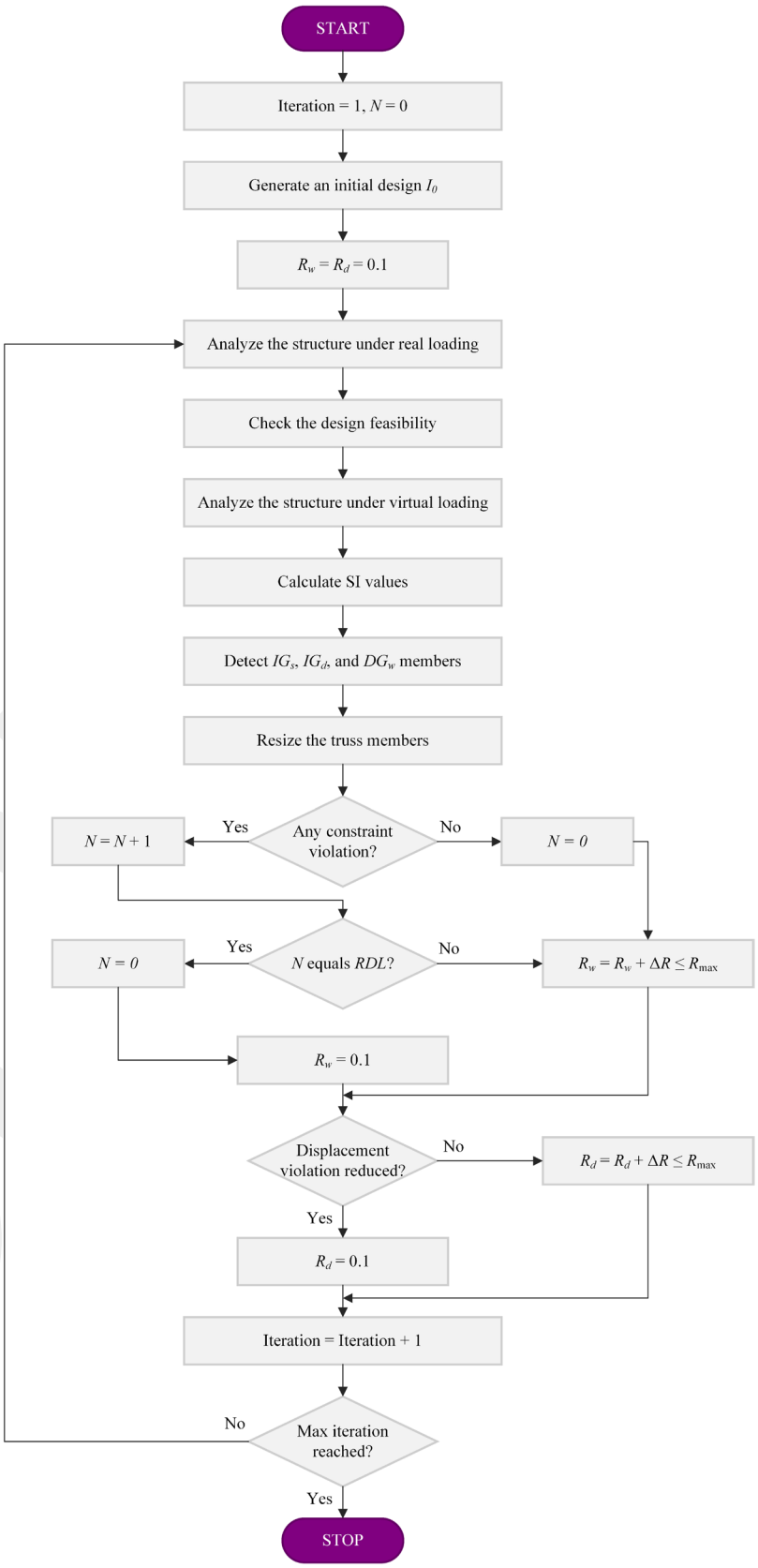
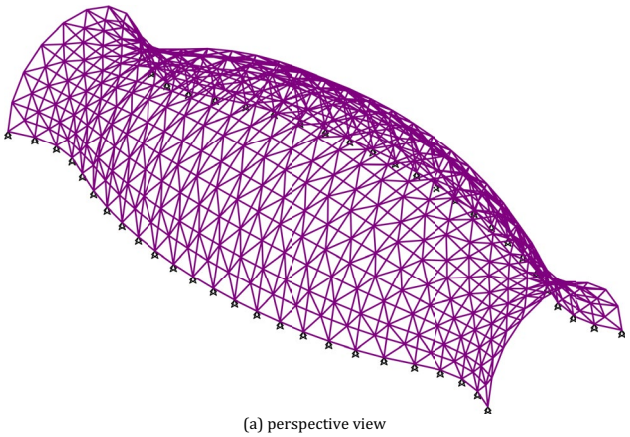
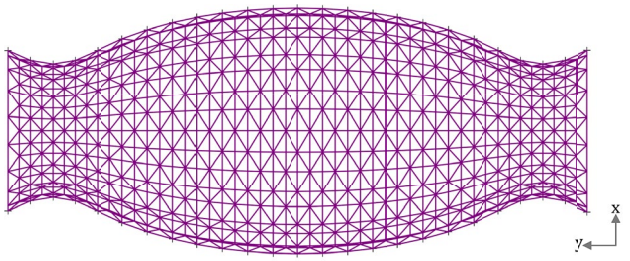


Fig. 1 Flowchart of the GSS algorithm with a repair deceleration mechanism



(a) perspective view



(b) plan view

Fig. 2 1728-member double-layer compound barrel vault: **a** perspective view, and **b** plan view

$$\text{Maximum step size} = \max\{1,\ \ |\alpha.\text{round}[N(\mu,\sigma)]|\} \tag{7}$$

where $N(\mu,\sigma)$ is a random number drawn from a normal distribution with a mean of μ and standard deviation of σ, subject to $\mu = 0$ and $\sigma = 1$, and α is a constant for tuning. Having set the rate of maximum incremental/decremental step size through Eq. (7), the resizing of IG members is carried out per Eq. (5). In a similar fashion, resizing of DG_w members is also accomplished using Eq. (6).

Step 6. Updating the rate of member resizing: This step administers the rate of grid members to be resized at every single iteration. Toward this end, an adaptive resizing procedure is implemented considering the feasibility of the generated candidate design in accordance with the following rules:

- The first rule stipulates that IG_s grid members—violating the strength constraints—are all upsized due to the significance of the strength criterion for generation of feasible designs.
- The second rule states that for weight minimization and handling displacement constraints, a specific percentage of grid members are appointed as IG_d and DG_w members in reference to the R_d and R_w ratios, respectively. In this work, the initial values for both the ratios are configured as the minimum value, 0.1.

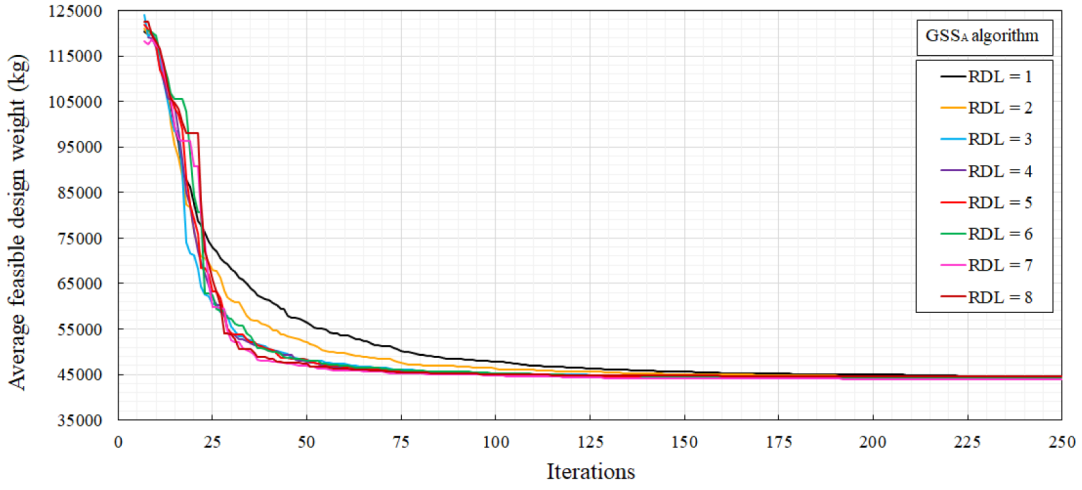
Typically, to facilitate an effective weight reduction process, R_w is increased by ΔR at every iteration, provided that the generated candidate solution is feasible. Consequently, this results in the contribution of more structural members to the weight reduction task. Yet, in the case any constraint violation is observed in the generated candidate design, the value of R_w is set back to its initial minimum value, so the repair of the produced infeasible candidate design can be accelerated. Besides, for IG_d members, the initial value of $R_d = 0.1$ can be retained during the optimization iterations. Nevertheless, if no improvement is observed when reducing the displacement constraint violations, the above-mentioned strategy employed for R_w could also be used for R_d, so the number of grid members that contribute to satisfying the displacement constraints is increased.

In the previous variants of the GSS algorithm, once an infeasible design was produced, a more conservative repair strategy was used by immediately limiting the number of DG_w members to the corresponding minimum value. However, in the present work, the effect of decelerating the repair of an infeasible candidate design on the convergence properties of the GSS is investigated. The proposed RDM is elaborated in the following section.

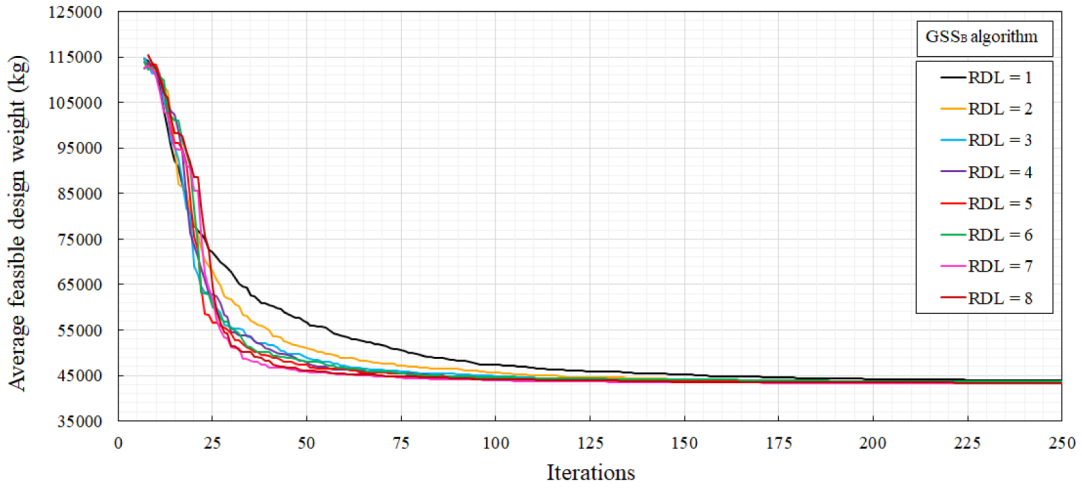
Step 7. Termination: The foregoing optimization process is iteratively performed, starting from the last generated candidate solution at the end of each iteration, until a termination criterion is met. In this study, the stopping criterion is defined by using a maximum number of optimization iterations.

4 The proposed repair deceleration mechanism

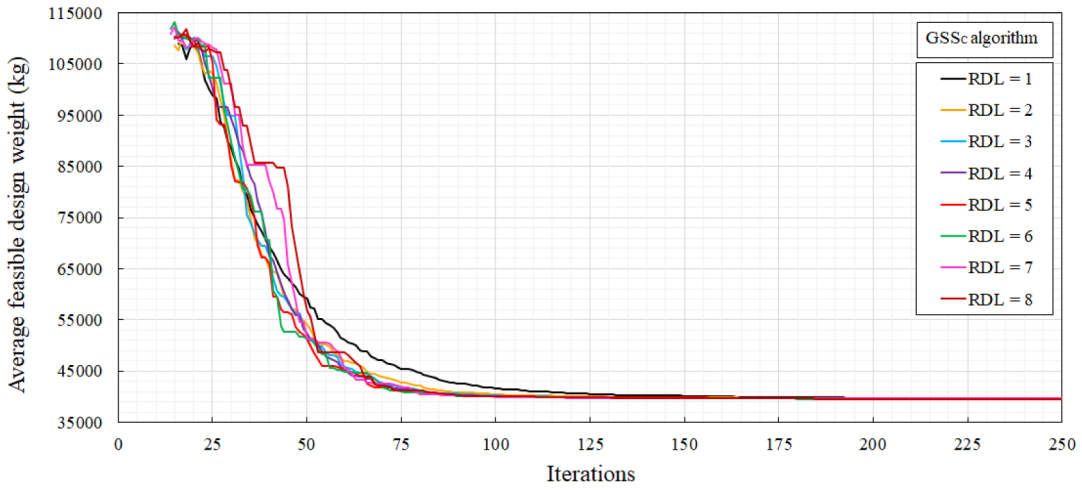
Typically, for weight minimization purposes in the GSS algorithm, a certain percentage of steel grid members are aggregated into the DG_w member group. This group comprises those overdesigned grid members, for which the size reduction is not expected to have a major effect on the displacement response. Accordingly, R_w parameter controls the percentage of structural members that should be involved in satisfying the weight minimization goal of the optimization process. Although an upper bound value of the R_w parameter, i.e., R_{max}, can be set to unity (Kazemzadeh Azad and Aminbakhsh 2022), it is important to notice that R_w



(a)



(b)



(c)

Fig. 3 Convergence histories for 1728-member double-layer compound barrel vault: **a** GSS_A, **b** GSS_B, and **c** GSS_C

Table 1 Optimization results for 1728-member compound barrel vault using GSS_A

Run no.	Best design weights (kg) at different repair deceleration levels							
	Standard	RDL = 2	RDL = 3	RDL = 4	RDL = 5	RDL = 6	RDL = 7	RDL = 8
1	44,326.7	44,811.4	44,873.5	44,330.8	44,469.6	43,920.6	43,455.9	43,596.9
2	44,130.5	43,975.1	44,469.1	45,108.6	43,122.3	43,549.4	44,338.1	44,342.6
3	44,155.2	45,294.9	43,663.1	44,870.2	44,092.7	43,301.6	43,834.5	43,905.5
4	44,304.1	44,056.9	43,701.9	44,527.1	43,598.6	43,333.8	44,220.2	44,762.7
5	44,378.7	44,001.5	44,395.7	43,858.8	44,513.4	44,882.9	44,168.8	44,151.7
6	45,260.6	44,590.7	43,721.2	44,016.9	43,581.0	44,947.0	43,843.6	44,810.0
7	43,937.0	44,029.9	44,305.9	43,730.0	43,212.9	45,010.6	44,371.1	45,448.6
8	43,431.2	43,975.0	44,707.0	43,400.1	45,160.2	44,669.2	43,417.1	44,237.4
9	44,803.3	44,165.8	44,818.9	43,761.7	45,577.6	44,032.5	43,568.1	45,099.5
10	44,505.7	44,831.1	43,776.0	43,876.3	44,303.6	44,772.9	44,410.3	44,683.0
11	45,274.6	44,865.7	44,591.2	45,481.8	44,776.6	43,729.2	44,309.9	44,495.8
12	44,514.5	43,994.4	43,769.5	43,893.7	43,969.2	44,793.9	44,392.1	44,973.5
13	44,927.2	44,402.4	44,349.0	43,592.1	44,740.4	43,995.8	43,589.0	44,512.2
14	44,090.2	45,002.0	44,188.0	44,782.3	44,208.2	43,962.0	43,724.2	44,141.5
15	44,523.8	44,418.1	44,361.7	45,250.6	44,120.4	43,834.8	43,810.5	44,224.9
Minimum (kg)	43,431.2	43,975.0	43,663.1	43,400.1	43,122.3	43,301.6	43,417.1	43,596.9
Maximum	45,274.6	45,294.9	44,873.5	45,481.8	45,577.6	45,010.6	44,410.3	45,448.6
Average	44,437.6	44,427.7	44,246.1	44,298.7	44,229.8	44,182.4	43,963.6	44,492.4
SD	490.0	443.6	423.9	660.6	684.8	604.6	366.7	483.3
COV (%)	1.1	1.0	1.0	1.5	1.5	1.4	0.8	1.1

Table 2 Optimization results for 1728-member compound barrel vault using GSS_B

Run no.	Best design weights (kg) at different repair deceleration levels							
	Standard	RDL = 2	RDL = 3	RDL = 4	RDL = 5	RDL = 6	RDL = 7	RDL = 8
1	43,268.6	42,983.5	43,339.7	43,283.6	43,771.2	44,003.5	43,059.4	43,926.5
2	43,852.7	43,555.7	42,855.0	44,887.6	43,952.0	43,442.2	43,237.7	43,556.6
3	44,803.5	43,934.2	43,461.4	43,250.3	42,348.4	43,490.5	42,719.6	43,980.6
4	42,921.3	44,158.2	43,414.1	43,468.1	43,459.0	43,159.1	42,449.1	43,309.5
5	43,498.2	44,023.2	42,780.8	42,520.8	44,027.2	44,581.9	43,852.5	42,573.3
6	43,407.7	43,590.7	43,199.3	44,030.4	44,000.3	42,662.1	43,436.8	43,233.9
7	43,916.1	42,382.6	43,143.9	43,837.1	45,080.0	44,415.7	44,456.7	42,885.0
8	44,052.7	43,488.1	43,354.9	42,325.5	42,565.3	44,663.2	44,487.5	43,046.6
9	45,252.6	43,942.2	43,220.3	42,482.2	42,762.7	44,455.3	42,868.4	43,049.2
10	44,154.2	42,511.6	43,570.1	43,762.6	43,345.5	44,197.0	43,312.2	43,293.6
11	44,928.6	43,033.8	43,824.1	43,897.4	42,838.9	42,721.9	42,643.8	43,257.5
12	42,623.0	43,134.0	43,449.2	43,487.1	43,712.1	44,710.6	42,936.6	43,237.6
13	43,593.7	42,642.3	44,042.9	42,471.2	43,354.0	43,019.6	43,685.2	43,606.7
14	43,787.6	43,830.8	42,925.8	42,644.7	42,857.6	43,335.1	43,148.6	43,094.4
15	43,879.5	43,448.8	43,721.3	42,205.0	44,065.2	42,885.2	42,411.2	43,724.5
Minimum (kg)	42,623.0	42,382.6	42,780.8	42,205.0	42,348.4	42,662.1	42,411.2	42,573.3
Maximum	45,252.6	44,158.2	44,042.9	44,887.6	45,080.0	44,710.6	44,487.5	43,980.6
Average	43,862.7	43,377.3	43,353.5	43,236.9	43,476.0	43,716.2	43,247.0	43,318.4
SD	720.7	571.2	353.4	777.4	722.0	748.4	647.1	384.6
COV (%)	1.6	1.3	0.8	1.8	1.7	1.7	1.5	0.9

adaptively changes over iterations with respect to the feasibility of the generated candidate design. In all the available implementations of the GSS algorithm, once an infeasible design is generated, the value of R_w is set back to its initial minimum value, so the number of DG_w members is restricted, and the weight reduction procedure is decelerated.

Table 3 Optimization results for 1728-member compound barrel vault using GSS_C

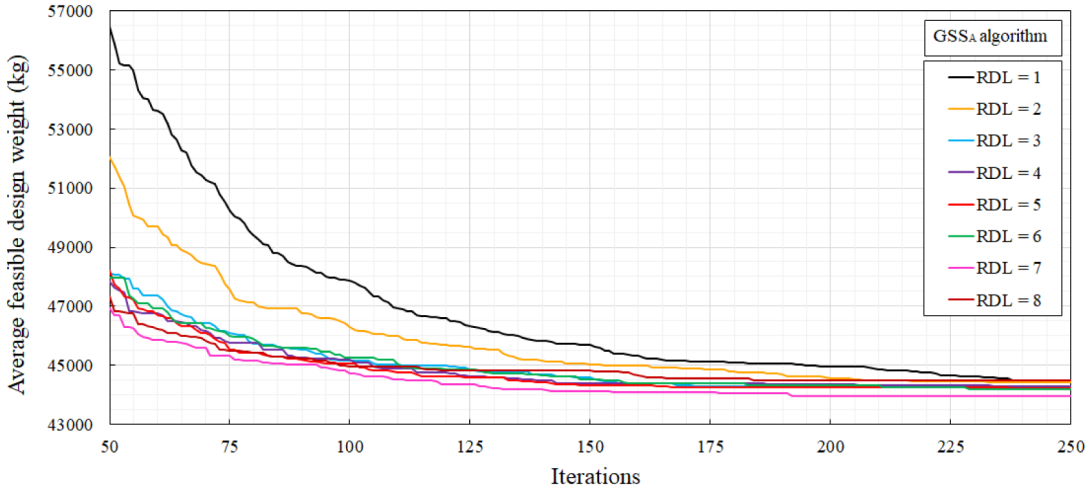
Run no.	Best design weights (kg) at different repair deceleration levels							
	Standard	RDL = 2	RDL = 3	RDL = 4	RDL = 5	RDL = 6	RDL = 7	RDL = 8
1	39,605.3	39,419.8	39,544.3	39,687.8	39,493.4	39,535.7	39,710.6	39,481.3
2	39,580.4	39,688.0	39,408.3	39,482.0	39,505.1	39,462.1	39,683.1	39,506.9
3	39,997.5	39,425.4	39,549.0	39,780.9	39,591.0	39,525.7	39,624.2	39,418.2
4	39,685.4	39,473.6	39,486.5	39,527.7	39,649.6	39,759.0	39,639.1	39,679.9
5	39,400.3	39,558.0	39,536.3	39,543.0	39,418.0	39,476.2	39,531.9	39,627.2
6	39,579.3	39,406.0	39,475.3	39,420.3	39,593.3	39,519.2	39,520.8	39,663.6
7	39,312.3	39,467.4	39,607.0	39,526.7	39,471.3	39,590.6	39,760.4	39,473.9
8	39,558.0	39,723.3	39,738.0	39,465.2	39,637.5	39,592.9	39,646.2	39,661.3
9	39,577.1	39,476.0	39,576.0	39,585.6	39,584.4	39,402.1	39,648.0	39,506.7
10	39,863.9	39,491.5	39,574.4	39,522.5	39,659.1	39,534.8	39,561.6	39,469.1
11	39,945.7	39,626.2	39,693.6	39,425.6	39,483.3	39,399.8	39,401.0	39,519.6
12	39,735.5	39,492.8	39,481.1	39,514.1	39,384.8	39,592.1	39,607.7	39,461.0
13	39,577.8	39,498.5	39,435.5	39,379.2	39,432.2	39,654.1	39,731.9	39,488.7
14	39,632.4	39,530.8	39,628.8	39,496.2	39,446.2	39,505.2	39,499.1	39,751.7
15	39,587.5	39,623.4	39,653.9	39,619.1	39,517.8	39,622.5	39,636.4	39,389.2
Minimum (kg)	39,312.3	39,406.0	39,408.3	39,379.2	39,384.8	39,399.8	39,401.0	39,389.2
Maximum	39,997.5	39,723.3	39,738.0	39,780.9	39,659.1	39,759.0	39,760.4	39,751.7
Average	39,642.6	39,526.7	39,559.2	39,531.7	39,524.5	39,544.8	39,613.5	39,539.9
SD	183.8	97.6	94.1	104.4	88.9	95.0	96.2	108.2
COV (%)	0.5	0.2	0.2	0.3	0.2	0.2	0.2	0.3

The rationale behind this approach is that reducing the cross-sectional areas of more structural members of an infeasible design will more likely result in a more infeasible solution. In this way, the previous versions of the algorithm aimed to immediately accelerate the repair of an infeasible candidate design, at the expense of decelerating the weight minimization process.

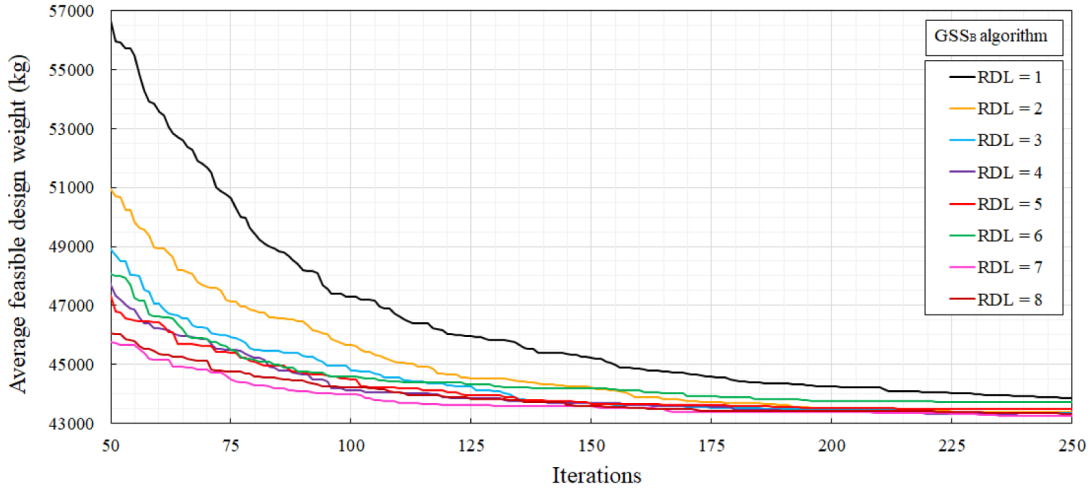
Despite a more conservative repair strategy employed so far for the GSS algorithm, in the present work, it has been recognized that a better balance between weight minimization and constraint handling features of the algorithm can be achieved through the RDM. To this end, during the optimization iterations, in case any constraint violation is observed in the generated candidate design, the number of DG_w members that contribute to weight minimization is kept constant for a specific number of iterations called repair deceleration level (RDL). Accordingly, although the repair mechanism of the algorithm via increase-group members, IG, will be active, the algorithm will meanwhile continue to reduce the weight of the structure at the same rate as before, i.e., using the same R_w ratio just before detecting the constraint violation. The proposed RDM, as the name implies, will decelerate the repair of the current infeasible design by employing more structural members for weight minimization over a specific number of iterations. Nonetheless, if no feasible solution is achieved after a certain number of iterations, the number of DG_w members is set back to its initial minimum value to help repair the infeasible design. It is worth mentioning that RDL = 1 implies the standard version of the algorithm where the number of DG_w members is set back to its initial minimum value in the succeeding iteration. Likewise, if RDL = 2, it means that, once an infeasible design is detected, the proposed RDM will keep reducing the structural weight over two successive iterations at the same rate just before detecting the infeasibility. To better illustrate the optimization procedure using the proposed approach, a general flowchart of the GSS algorithm with a repair deceleration mechanism is shown in Fig. 1. The effect of the proposed RDM on the convergence properties of the GSS is studied in the following section using challenging test instances of steel double-layer grids with numerous discrete design variables.

5 Numerical experiments

The present section evaluates the efficiency of the proposed repair deceleration mechanism when utilized together with the GSS algorithm. First, a more comprehensive analysis is carried out using two high-dimensional test examples from the literature, namely a 1728-member double-layer compound barrel vault (Kazemzadeh Azad and Aminbakhsh 2022), and a 6000-member double-layer scallop dome (Kazemzadeh Azad and Aminbakhsh 2021), at different repair deceleration levels. Once a suitable value for RDL is determined based on rigorous analyses of the



(a)



(b)

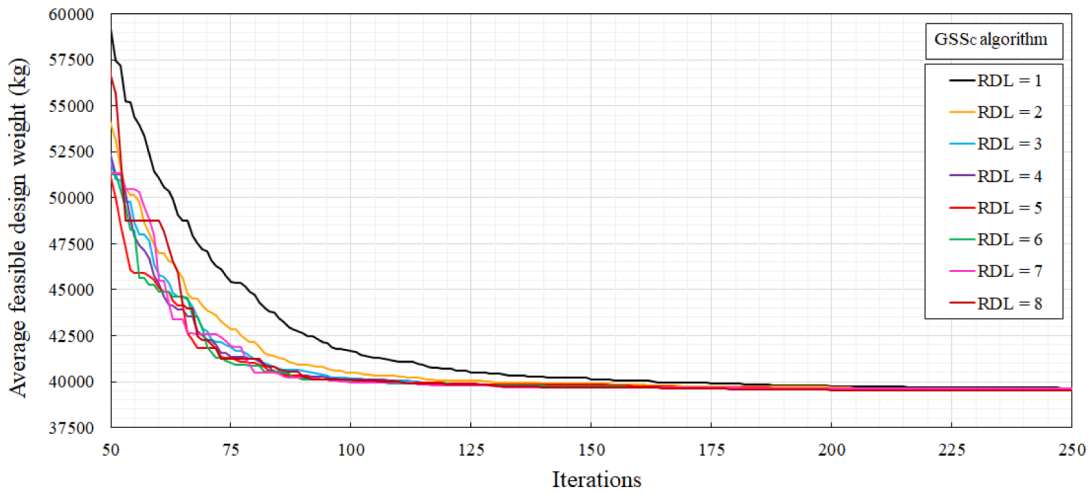


Fig. 4 Convergence histories for 1728-member double-layer compound barrel vault over iterations 50–250: **a** GSS_A, **b** GSS_B, and **c** GSS_C

Table 4 Comparison of optimization results for 1728-member compound barrel vault using GSS_A at different iterations

Iteration no	Average design weights (kg) of independent runs at different repair deceleration levels							
	Standard	RDL = 2	RDL = 3	RDL = 4	RDL = 5	RDL = 6	RDL = 7	RDL = 8
50	56,467.8	52,048.0	48,142.7	47,815.6	48,208.6	47,960.9	46,914.4	47,313.6
	Rank = 8	7	5	3	6	4	1	2
75	50,249.5	47,579.3	46,104.4	45,760.4	45,537.2	45,996.5	45,309.9	45,500.0
	8	7	6	4	3	5	1	2
100	47,850.2	46,279.5	45,152.3	45,182.6	45,050.1	45,262.9	44,736.2	44,972.5
	8	7	4	5	3	6	1	2
125	46,318.7	45,617.0	44,858.4	44,616.6	44,602.5	44,865.2	44,358.7	44,838.4
	8	7	5	3	2	6	1	4
150	45,687.9	45,023.9	44,573.0	44,401.9	44,326.6	44,512.9	44,114.2	44,814.1
	8	7	5	3	2	4	1	6
175	45,117.3	44,866.3	44,297.0	44,379.0	44,252.6	44,381.8	44,083.0	44,559.4
	8	7	3	4	2	5	1	6
200	44,955.3	44,570.7	44,287.1	44,339.5	44,247.7	44,312.7	43,963.6	44,501.7
	8	7	3	5	2	4	1	6
225	44,656.9	44,453.4	44,278.0	44,315.2	44,247.7	44,241.5	43,963.6	44,492.4
	8	6	4	5	3	2	1	7
250	44,437.6	44,427.7	44,246.1	44,298.7	44,229.8	44,182.4	43,963.6	44,492.4
	7	6	4	5	3	2	1	8

Table 5 Comparison of optimization results for 1728-member compound barrel vault using GSS_B at different iterations

Iteration no	Average design weights (kg) of independent runs at different repair deceleration levels							
	Standard	RDL = 2	RDL = 3	RDL = 4	RDL = 5	RDL = 6	RDL = 7	RDL = 8
50	56,645.0	50,940.1	48,910.3	47,702.7	47,316.0	48,050.0	45,770.5	46,064.7
	Rank = 8	7	6	4	3	5	1	2
75	50,645.5	47,141.7	45,915.2	45,498.5	45,375.6	45,492.5	44,497.6	44,770.1
	8	7	6	5	3	4	1	2
100	47,297.5	45,657.5	44,783.0	44,108.6	44,481.4	44,593.7	44,003.7	44,218.8
	8	7	6	2	4	5	1	3
125	45,956.5	44,521.7	44,252.9	43,856.5	43,960.3	44,318.6	43,622.3	43,832.3
	8	7	5	3	4	6	1	2
150	45,210.6	44,209.5	43,681.8	43,680.6	43,663.6	44,176.3	43,572.0	43,600.2
	8	7	5	4	3	6	1	2
175	44,590.7	43,717.0	43,530.1	43,564.8	43,604.0	43,917.9	43,388.0	43,417.2
	8	6	3	4	5	7	1	2
200	44,241.4	43,529.7	43,439.0	43,408.8	43,534.1	43,764.3	43,382.1	43,417.2
	8	5	4	2	6	7	1	3
225	44,013.5	43,396.4	43,378.4	43,330.4	43,482.7	43,716.2	43,329.7	43,379.2
	8	5	3	2	6	7	1	4
250	43,862.7	43,377.3	43,353.5	43,236.9	43,476.0	43,716.2	43,247.0	43,318.4
	8	5	4	1	6	7	2	3

above-mentioned preliminary test instances, the applicability of the proposed RDM is further investigated using two very high-dimensional examples of steel double-layer grids, specifically a 21,212-member free-form double-layer grid, and a 25,514-member double-layer multi-dome. The parameter setting for the two conventional variants of the GSS algorithm, namely GSS_A and GSS_B, is carried out in accordance with Kazemzadeh Azad and Aminbakhsh (2021). Meanwhile, parameters of GSS_C are set regarding the suggested values in Kazemzadeh Azad and Aminbakhsh (2022). For all the applied algorithms, a maximum of 250 iterations is adopted as a stopping criterion. For sizing optimization,

Table 6 Comparison of optimization results for 1728-member compound barrel vault using GSS_C at different iterations

Iteration no	Average design weights (kg) of independent runs at different repair deceleration levels							
	Standard	RDL = 2	RDL = 3	RDL = 4	RDL = 5	RDL = 6	RDL = 7	RDL = 8
50	59,161.8	54,008.6	51,897.2	52,281.6	51,128.4	51,289.1	51,340.9	56,655.5
	Rank = 8	6	4	5	1	2	3	7
75	45,408.2	42,823.5	41,874.5	41,352.1	41,279.8	41,012.7	41,937.4	41,225.7
	8	7	5	4	3	1	6	2
100	41,647.1	40,491.2	40,180.7	40,126.2	40,000.4	39,995.6	39,973.8	40,096.9
	8	7	6	5	3	2	1	4
125	40,493.6	40,028.8	39,820.9	39,861.3	39,892.9	39,778.4	39,771.3	39,856.2
	8	7	3	5	6	2	1	4
150	40,129.6	39,876.7	39,711.9	39,735.5	39,821.2	39,708.4	39,690.7	39,670.3
	8	7	4	5	6	3	2	1
175	39,914.9	39,742.0	39,649.8	39,648.6	39,682.0	39,647.9	39,654.0	39,629.9
	8	7	4	3	6	2	5	1
200	39,750.2	39,675.8	39,616.7	39,624.4	39,627.3	39,588.4	39,643.0	39,542.5
	8	7	3	4	5	2	6	1
225	39,702.7	39,593.7	39,577.1	39,560.5	39,573.6	39,581.2	39,631.0	39,542.2
	8	6	4	2	3	5	7	1
250	39,642.6	39,526.7	39,559.2	39,531.7	39,524.5	39,544.8	39,613.5	39,539.9
	8	2	6	3	1	5	7	4

cross-sectional areas of members are selected from a list of 37 standard pipe sections (American Institute of Steel Construction (AISC) 1994), and the steel used has the following material properties: modulus of elasticity (E) = 200 GPa, yield stress (F_y) = 248.2 MPa, and unit weight of the steel (ρ) = 7.85 ton/m^3.

The concepts of formex algebra and the programming language Formian (Nooshin and Disney 2000, 2001, 2002; Nooshin et al. 2017) can be readily used as efficient tools to generate complex configurations of grid structures. In the present work, Formian-2 software is applied to extract the necessary geometric data for the structural analysis stage. Moreover, to generate the geometry of the investigated steel double-layer grids, the parametric formex formulations as well as the general guidelines provided in Nooshin et al. (2017) and Moghimi (2006) are used with some amendments to the corresponding formulations and parameters.

5.1 Test example 1: 1728-member double-layer compound barrel vault

In order to explore the efficiency of the proposed RDM, the 1728-member compound barrel vault shown in Fig. 2 is chosen as the first test instance. Multi-objective design optimization of this double-layer grid has been recently carried out (Kazemzadeh Azad and Aminbakhsh 2022). Here, a single-objective version of the example is considered where, for serviceability requirements, the maximum displacement of all joints of the 1728-member double-layer grid in the x, y, and z directions is limited to 3 cm. Since no member grouping scheme is used, this high-dimensional test instance involves 1728 distinct sizing design variables. The 1728-member double-layer compound barrel vault is sized in a manner that concentrated loads are applied at all the unsupported joints of the steel double-layer grid under three independent loading cases:

(i) horizontal loads of 10 kN acting in the positive x-direction,
(ii) horizontal loads of 10 kN acting in the positive y-direction,
(iii) vertical loads of 10 kN acting in the negative z-direction.

Discrete sizing optimization of the 1728-member double-layer compound barrel vault is accomplished using three variants of the GSS algorithm, namely GSS_A, GSS_B, and GSS_C. The algorithms are integrated with the proposed RDM and different RDLs are studied. As discussed earlier, a repair deceleration level of RDL = 1 is equivalent to the standard variant of the algorithms where RDM is not activated. Figure 3 shows the convergence curves of optimization using the foregoing three variants of the GSS algorithm at different RDLs. The convergence curves are based on the average values obtained over fifteen independent runs of the algorithms for each RDL. Considering the applied three algorithm variants, and eight different RDLs where fifteen independent runs are accomplished at each RDL, a total of 360 independent optimization runs have been performed for

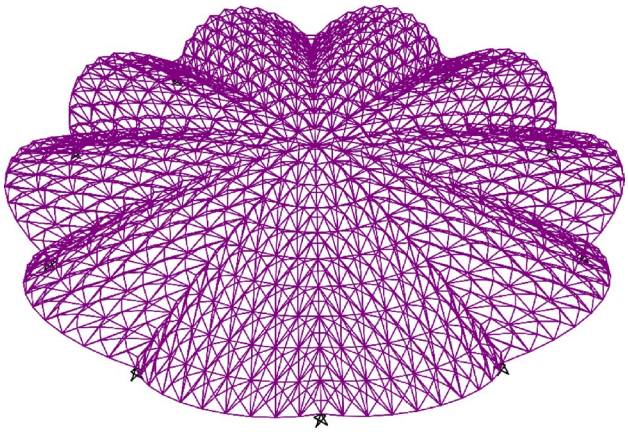
Table 7 Optimization results for 6000-member double-layer scallop dome using GSS_C

Run no	Best design weights (kg) at different repair deceleration levels							
	Standard	RDL = 2	RDL = 3	RDL = 4	RDL = 5	RDL = 6	RDL = 7	RDL = 8
1	114,394.5	111,893.3	111,897.9	112,008.4	110,889.6	111,253.7	111,383.4	111,468.9
2	112,709.6	111,556.6	111,847.1	111,750.8	111,449.8	111,609.1	111,769.4	111,738.6
3	114,312.7	111,416.8	111,987.1	111,348.6	111,755.8	111,275.2	111,539.7	112,368.9
4	113,903.1	111,765.9	112,138.6	112,262.8	111,524.6	111,525.3	111,936.7	111,670.8
5	113,748.0	113,379.2	111,586.0	111,602.0	111,936.0	111,785.8	112,390.2	112,454.3
6	115,000.2	111,924.5	112,125.8	112,153.1	111,977.3	112,053.3	111,164.8	112,142.7
7	114,765.9	112,057.9	112,020.9	111,331.8	112,299.9	111,802.4	111,074.3	112,318.5
8	113,872.9	113,097.0	111,572.3	111,564.9	111,565.7	111,502.0	112,174.6	112,268.3
9	114,113.4	111,438.5	111,078.4	111,912.4	111,117.4	111,587.1	111,843.8	111,560.6
10	114,293.2	111,795.2	111,628.4	111,744.0	111,916.4	110,903.3	112,212.8	112,128.5
11	113,641.2	111,909.6	111,229.4	111,727.7	111,578.3	111,631.7	111,234.2	111,428.4
12	114,023.3	111,360.6	111,897.7	111,755.1	110,961.4	112,086.7	112,198.2	111,902.2
13	115,206.4	111,951.8	112,080.6	111,445.1	111,811.6	111,045.0	112,593.2	111,053.7
14	114,112.3	112,119.8	111,857.6	111,052.3	111,409.7	111,643.9	112,090.8	111,810.9
15	113,575.8	112,002.0	111,528.1	111,738.2	111,140.9	111,482.3	111,754.6	110,956.9
Minimum (kg)	112,709.6	111,360.6	111,078.4	111,052.3	110,889.6	110,903.3	111,074.3	110,956.9
Maximum	115,206.4	113,379.2	112,138.6	112,262.8	112,299.9	112,086.7	112,593.2	112,454.3
Average	114,111.5	111,977.9	111,765.1	111,693.2	111,555.6	111,545.8	111,824.0	111,818.1
SD	614.4	567.0	320.4	321.0	406.0	330.0	465.8	467.8
COV (%)	0.5	0.5	0.3	0.3	0.4	0.3	0.4	0.4

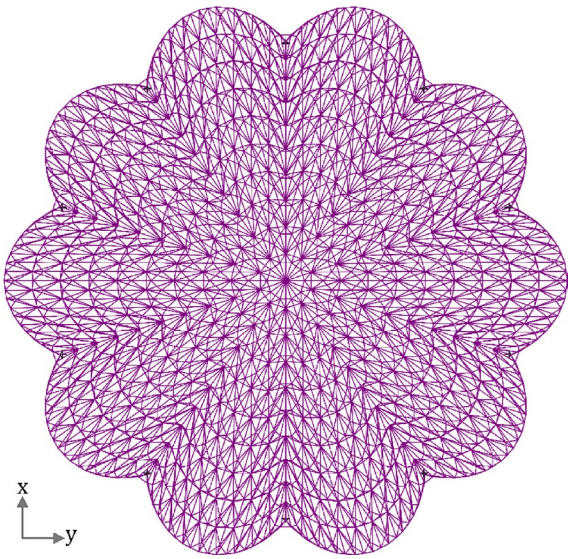
this test example to evaluate the efficiency of the proposed approach. Tables 1, 2, and 3 present the best feasible design weights obtained at different RDLs using GSS_A, GSS_B, and GSS_C, respectively. It is worth noting that besides the solution quality of the final design, the convergence property of the optimization algorithm is important as well. In particular, when dealing with computationally expensive optimization problems, finding a reasonably good solution at the early stages of optimization is highly favorable. For the sake of clarity, convergence histories for 1728-member double-layer compound barrel vault are plotted over iterations 50 to 250 in Fig. 4. As can be seen from this figure, all the three variants of the GSS algorithm achieve better convergence rates when integrated with RDM over iterations 50–250.

Due to the excessive computational effort required for optimizing large-scale steel double-layer grids with numerous discrete design variables, the number of iterations required to locate a near-optimum solution is an important factor. This will, in fact, represent the applicability of an optimization algorithm to large-scale structural optimization problems. Table 4 presents the average design weights of all runs using GSS_A at different RDLs, specifying the corresponding rank of each average solution compared to those obtained at other RDLs for the same iteration number. As shown in the table, the RDM-integrated GSS_A presents a better solution at RDL = 7 versus other RDLs. From Table 4, it can be seen that while the first rank belongs to the solutions obtained using RDL = 7, the standard variant of the algorithm without RDM (i.e., RDL = 1) tends to converge much slower than other counterparts at different RDLs. This clearly shows the efficiency of the proposed RDM to improve the convergence rate of the GSS_A algorithm.

Table 5 compares the results of optimization at different iterations using GSS_B, when integrated with RDM. Analogous to the performance of the RDM-integrated GSS_A, here also an enhancement in the convergence scheme of GSS_B can be observed at different RDLs versus the standard version. Particularly, except for the last iteration, by using RDL = 7 the algorithm obtains the first solution rank at the iterations used for monitoring, namely every 25 iterations from iteration 50–250. In the last monitoring point (iteration number 250), the second-best average weight is obtained by GSS_B, whereas the best average final solution belongs to the implementation of the algorithm using RDL = 4. It is important to highlight that although the rank of each implementation of the algorithm is reported over different iterations, the significance of these ranks at different iterations is not equivalent. In fact, for computationally expensive structural optimization problems, locating a reasonably good solution at the early stages of optimization (e.g., iteration number 50 or 100), with a tangible difference compared to other alternative techniques, could be much more favorable than locating a slightly better solution at iteration number 250. This can be seen in the case of GSS_B with RDL = 7, which



(a) perspective view



(b) plan view

Fig. 5 6000-member double-layer scallop dome: **a** perspective view, and **b** plan view

performs better in the earlier iterations, although achieving an average final design weight of 43,247.0 kg, which is only slightly inferior to that of RDL = 4 (43,236.9 kg). In this regard, the main focus of the present work is to enhance the performance of the GSS algorithm in terms of convergence rate so that reasonably good solutions can be achieved for very high-dimensional problems of steel double-layer grids using fewer iterations compared to the standard counterpart.

The optimization results of the 1728-member double-layer compound barrel vault using the RDM-integrated GSS_C are tabulated in Table 6. As can be observed from Table 6, as well as in Fig. 4c, the superiority of the results achieved by the RDM-integrated GSS_C is in line with the observations already noted for GSS_A and GSS_B. Moreover, considering the numerical results provided in Table 6 as well as the convergence curves depicted in Fig. 4c, RDL = 6 seems to be a suitable choice to improve the convergence scheme of GSS_C algorithm. Considering the computational costs of the investigated test examples, and since GSS_C is a more successful version of the GSS, compared to the other two conventional variants, in the following test examples of large-scale steel double-layer grids, only GSS_C is used for performance evaluation of the proposed approach.

5.2 Test example 2: 6000-member double-layer scallop dome

Discrete sizing optimization of the 6000-member double-layer scallop dome shown in Fig. 5 has already been studied in Kazemzadeh Azad and Aminbakhsh (2021) using GSS_A and GSS_B algorithms. Here, it is aimed to investigate the efficiency of the proposed RDM when combined with the GSS_C variant. The steel double-layer scallop dome consists of 6000 members and 1551 joints. The cross-sectional areas of all the members are considered as sizing variables, resulting in a total of 6000 discrete variables. For design purposes, concentrated loads are applied at all the unsupported joints of the dome in the following three cases:

(i) horizontal loads of 5 kN applied in the positive x-direction,
(ii) horizontal loads of 5 kN applied in the positive y-direction,
(iii) vertical loads of 7 kN applied in the negative z-direction.

Besides, to satisfy the serviceability requirements, the displacement of all nodes in the x, y, and z directions is limited to a maximum allowable value of 5 cm.

The optimum design of the 6000-member double-layer scallop dome is performed using the RDM-integrated GSS_C, with the best feasible design weights obtained at different RDLs being given in Table 7. It is worthwhile to note that in Kazemzadeh Azad and Aminbakhsh (2021), using a termination condition of 500 iterations for the same test example, the standard variants of GSS_A and GSS_B have already produced mean design weights of 132,446.3 and 129,181.0 kg, respectively. As shown in Table 7, the standard version of GSS_C achieves a considerably reduced mean weight of 114,111.5 kg, by using a maximum number of 250 iterations. As already stated, owing to the better performance of GSS_C, for this example as well as the next instances of steel double-layer grids, only this variant is used for performance evaluation of the proposed RDM. Considering the eight different RDLs associated, where fifteen independent runs are performed at each RDL, a total of 120 independent

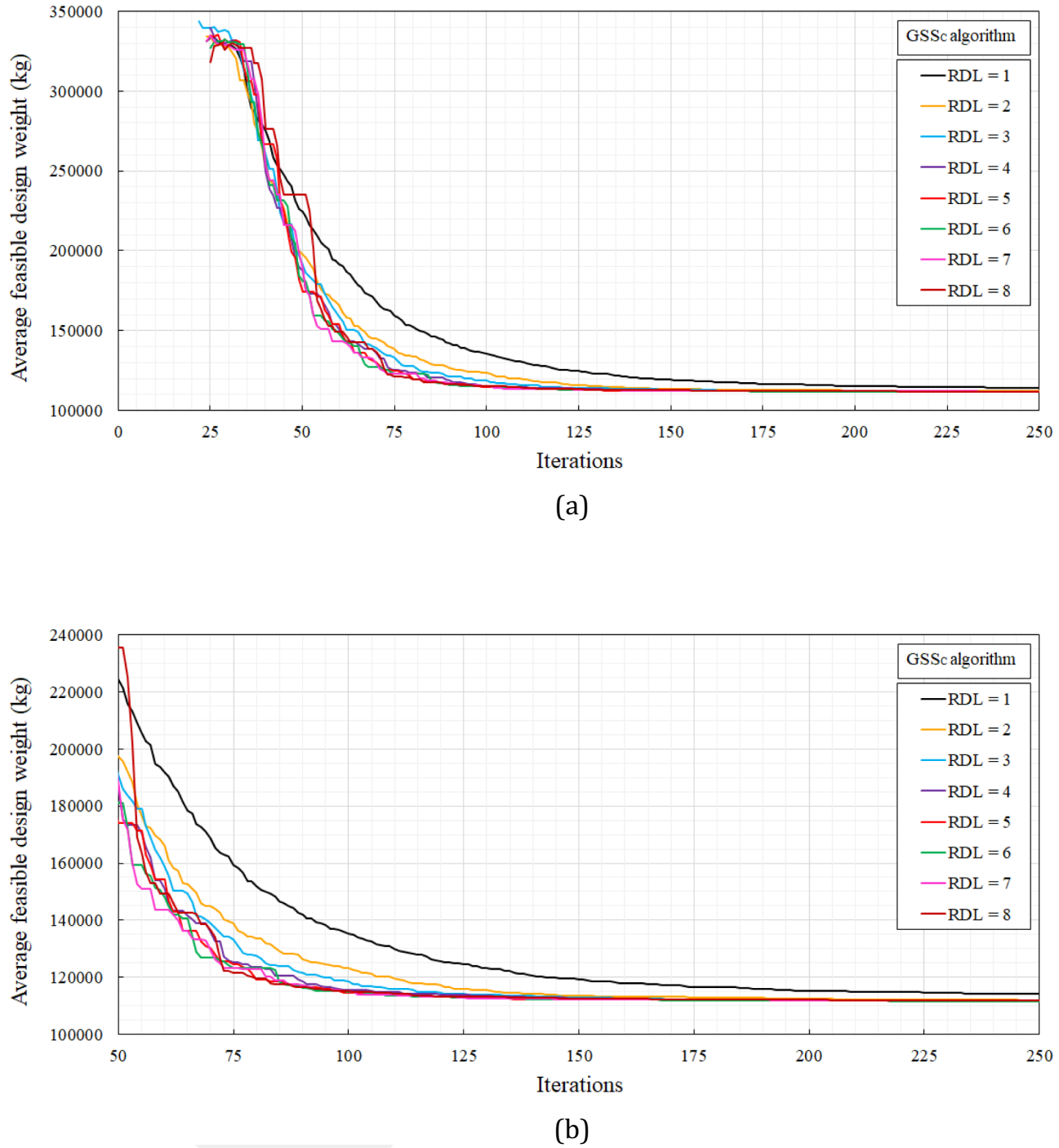


Fig. 6 Convergence histories for 6000-member double-layer scallop dome using GSS_C: **a** all the iterations, and **b** iterations 50–250

optimization runs have been accomplished for the investigated 6000-member double-layer scallop dome to study the usefulness of the proposed approach.

Figure 6 presents a comparison of different implementations of the RDM-integrated GSS_C algorithm at eight different RDLs. In general, it can be inferred from the figure that the proposed RDM is capable of improving the convergence rate of the algorithm. Rigorous observation of the convergence histories plotted in Fig. 6b, together with the design weight ranks provided in Table 8, indicates more abrupt changes in the early stages of optimization for RDL = 8. As can be seen from Table 8, an implementation of GSS_C using RDL = 8 yields an average design weight of 235,377.5 kg after 50 iterations. In fact,

Table 8 Comparison of optimization results for 6000-member double-layer scallop dome using GSS_C at different iterations

Iteration no	Average design weights (kg) of independent runs at different repair deceleration levels							
	Standard	RDL = 2	RDL = 3	RDL = 4	RDL = 5	RDL = 6	RDL = 7	RDL = 8
50	224,584.38	197,457.96	191,124.75	187,379.33	174,050.21	180,903.12	188,689.44	235,377.58
	Rank = 7							
75	159,177.08	138,802.57	133,369.26	125,390.05	124,606.54	123,130.32	123,187.23	121,477.31
100	135,281.08	123,334.67	118,671.06	115,545.15	114,698.61	115,007.62	115,238.44	115,109.93
125	124,733.58	115,919.37	114,193.86	113,726.85	112,840.62	113,031.13	112,736.81	113,261.44
150	119,266.08	113,530.37	112,999.96	112,585.84	112,261.91	112,273.82	112,353.53	112,676.65
175	116,633.88	112,937.37	112,212.15	112,079.83	111,991.12	111,954.11	112,088.14	112,342.06
200	115,346.88	112,528.17	111,966.64	111,869.12	111,901.53	111,792.11	111,992.25	112,133.16
225	114,724.68	112,288.57	111,792.04	111,776.23	111,639.51	111,717.02	111,981.56	111,864.05
250	114,111.58	111,977.97	111,765.14	111,693.23	111,555.62	111,545.81	111,824.06	111,818.15

this is the heaviest solution obtained in the corresponding monitoring point among the studied RDLs. However, at iteration number 75, RDL = 8 results in the best mean design weight of 121,477.3 kg among the other RDLs. Considering the convergence curves shown in Fig. 6, in contrast to RDL = 8, the use of RDL = 6 leads to a more robust and stable performance, seeming to be a more reasonable choice. It is important to mention that the use of RDL = 6 is suggested due to its efficiency, especially when employed in conjunction with GSS_C. This can be observed from the convergence curves provided in Figs. 4c and 6b for both the first and the second test examples, respectively. Hence, RDL = 6 is adopted to tackle the following very high-dimensional sizing optimization examples of steel double-layer grids.

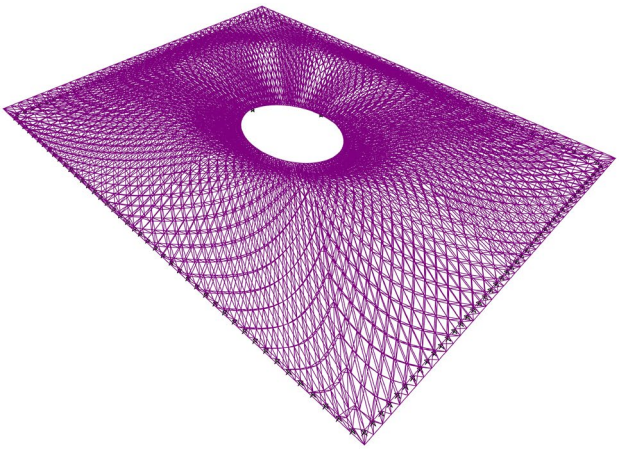
5.3 Test example 3: 21,212-member free-form double-layer grid

To further evaluate the efficiency of the proposed RDM in very high-dimensional sizing optimization examples, the 21,212-member free-form double-layer grid shown in Fig. 7 is considered as the third instance. The steel free-form double-layer grid is composed of 21,212 members and 4928 joints (which includes 180 supported joints). Similar to the previous test examples, to keep the dimension of the optimization problem as large as possible, no member grouping is carried out, resulting in a very high-dimensional test instance with 21,212 discrete sizing variables. The structural members are designed to carry the concentrated loads applied at all the unsupported joints of the structure in the following three cases:

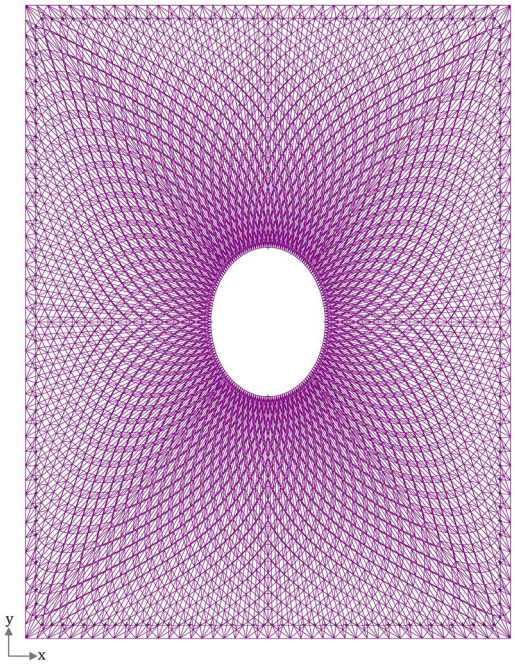
(i) horizontal loads of 4 kN applied in the positive *x*-direction,
(ii) horizontal loads of 4 kN applied in the positive *y*-direction,
(iii) vertical loads of 5 kN applied in the negative *z*-direction.

Moreover, the free-form double-layer grid shall be designed in such a way that the maximum displacement of all joints in the *x*, *y*, and *z* directions is limited to 5 cm under the aforementioned load cases.

Owing to the excessive computational effort required for the design optimization of the adopted steel double-layer grid, only two implementations of the optimization algorithm, namely the standard GSS_C and the RDM-integrated version with RDL = 6, are studied through performing 10 independent runs per implementation. Table 9 presents the statistical results of the best feasible design weights obtained over the foregoing independent runs using the standard and RDM-integrated variants of GSS_C. Figure 8 compares the convergence histories obtained using the RDM-integrated version of GSS_C and the standard counterpart for the investigated 21,212-member free-form double-layer grid. It can be seen from this figure that the integration of the RDM clearly improves the convergence rate of the GSS_C optimization algorithm. It should be noted that although a slightly worse convergence is achieved



(a) perspective view



(b) plan view

Fig. 7 21,212-member free-form double-layer grid: **a** perspective view, and **b** plan view

by the RDM-integrated algorithm compared to the standard version at initial iterations, the general convergence scheme of the former is much more favorable than that of the latter. Figure 9 shows the histograms of member *LCR*s for the first feasible design that is obtained after 32 iterations, an interim design located at iteration number 85, and the final best feasible design achieved at iteration number 232, in the best run of the algorithm using the proposed RDM with RDL = 6. The figure illustrates a gradual increase in member *LCR*s during the optimization iterations for the considered challenging structural optimization problem with 21,212 sizing design variables.

Table 9 Optimization results for test examples 3 and 4 using the standard and RDM-integrated variants of GSS_C

Run no.	21,212-member double-layer grid		25,514-member double-layer multi-dome	
	Standard	RDM-integrated	Standard	RDM-integrated
1	259,732.5	242,184.3	318,144.2	299,931.9
2	266,769.7	244,173.2	314,288.8	299,498.5
3	262,603.4	240,152.2	309,830.3	300,372.3
4	259,990.1	240,863.4	316,813.8	296,060.2
5	254,958.5	236,356.7	315,991.8	297,622.9
6	261,838.7	238,234.1	315,382.3	296,742.3
7	267,618.0	248,342.0	315,919.0	299,251.6
8	272,990.6	240,926.0	318,850.5	296,307.8
9	274,532.3	237,712.8	310,877.7	299,492.3
10	267,624.8	243,045.0	317,826.1	298,625.2
Minimum (kg)	254,958.5	236,356.7	309,830.3	296,060.2
Maximum	274,532.3	248,342.0	318,850.5	300,372.3
Average	264,865.9	241,199.0	315,392.4	298,390.5
SD	6153.8	3494.8	2993.5	1585.2
COV (%)	2.3	1.4	0.9	0.5

5.4 Test example 4: 25,514-member double-layer multi-dome

The 25,514-member double-layer multi-dome depicted in Fig. 10 is selected as the last structural optimization test example. The steel double-layer multi-dome is composed of 25,514 members and 6483 joints (which includes 186 supported joints). The cross-sectional areas of all the structural members are treated as design variables, resulting in a very high-dimensional test example (including 25,514 distinct discrete sizing variables). For design purposes, concentrated loads are applied at all the unsupported joints of the multi-dome in the following three cases:

(i) horizontal loads of 3 kN applied in the positive *x*-direction,
(ii) horizontal loads of 3 kN applied in the positive *y*-direction,
(iii) vertical loads of 6 kN applied in the negative *z*-direction.

In order to satisfy the serviceability criteria, the displacement of all nodes in the *x*, *y*, and *z* directions is limited to a maximum allowable value of 5 cm. Similar to the previous test example, only two implementations of the optimization algorithm, i.e., the standard GSS_C and the RDM-integrated version with RDL = 6, are considered here by performing 10 independent runs per implementation. Table 9 presents the

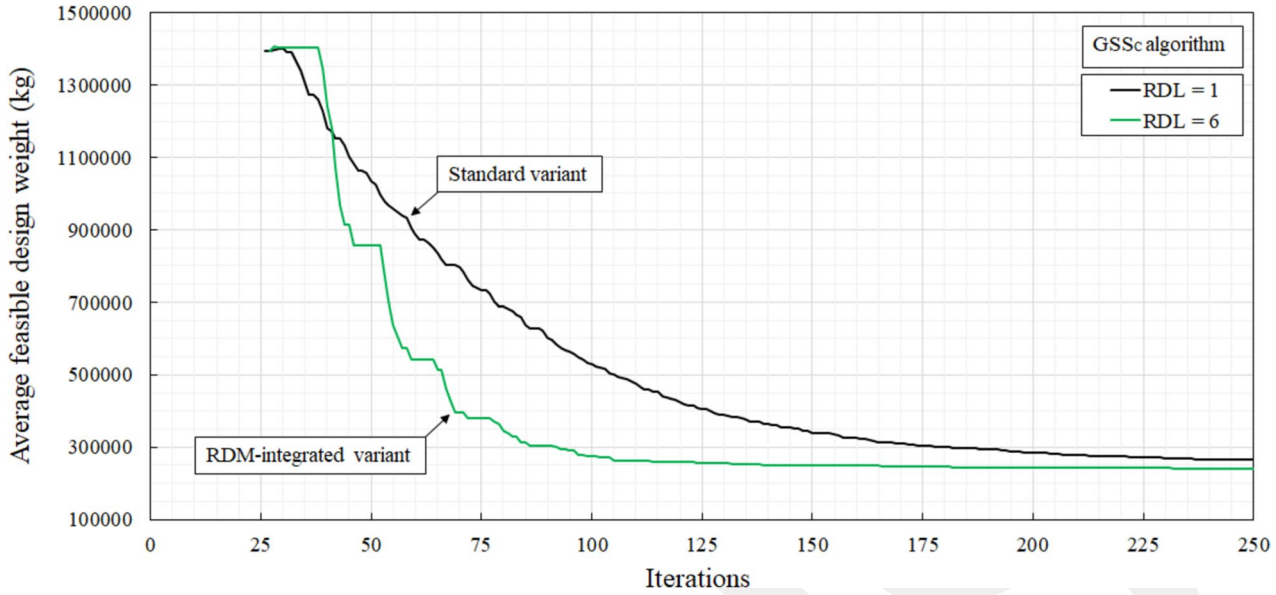


Fig. 8 Convergence histories for 21,212-member free-form double-layer grid using GSS_C

statistical results of the best feasible design weights obtained over the aforementioned runs using the standard and RDM-integrated variants of GSS_C. Figure 11 compares the convergence curve of the RDM-integrated version of GSS_C versus that of the standard counterpart for the 25,514-member double-layer multi-dome. It is apparent from the figure that employing RDM significantly enhances the convergence property of the GSS_C algorithm and facilitates locating reasonably good solutions at the early stages of optimization. Figure 12 presents the histograms of member *LCR*s for the first feasible design achieved after 23 iterations, an interim design found at iteration number 70, and the final best feasible design obtained at iteration number 226, in the best run of the proposed RDM-integrated variant. The figure shows a gradual increase in member *LCR*s in the course of optimization for the investigated very high-dimensional sizing optimization problem, which involves 25,514 sizing design variables. Considering the numerical experiments performed, it can be inferred that the proposed RDM-integrated GSS_C can be used as a computationally efficient tool for handling the sizing optimization problems of steel double-layer grids with a considerably large number of design variables.

Finally, it is worthwhile to note that, in contrast to the conventional gradient-based structural optimization methods (Erbatur and Al-Hussainy 1992; Tabak and Wright 1981; Saka 1990; Zhou and Rozvany 1992, 1993; Feury and Geradin 1978), generally the GSS-based algorithms possess the following advantageous attributes:

- No need for gradient information of objective functions or constraints
- No need for approximations on optimization constraints or design variables
- Efficient sensitivity analysis using the IFM (Patnaik et al. 1991) (one structural analysis is carried out per iteration.)
- Handling problems with excessive number of constraints and design variables
- Ease of algorithmic formulation and implementation
- Less dependency on the initial randomly generated design
- Stochastic search features to escape the local optima

These advantageous properties, make the GSS-based approaches promising alternatives to the conventional structural optimization techniques especially when dealing with challenging structural optimization instances.

6 Future research directions

Although the main focus of the present study is to enhance the convergence properties of the GSS algorithm in extremely high-dimensional structural optimization problems of steel double-layer grids, it would be fruitful to further investigate the efficiency of the proposed technique as well as other alternative design-driven approaches in structural optimization of steel frames with numerous discrete variables. In fact, substantial research is needed to develop versatile design-driven methodologies capable of handling high-dimensional structural optimization problems under dynamic loads and frequency constraints. Besides, regarding the recent developments in additive manufacturing technologies, the design and fabrication of lightweight structures

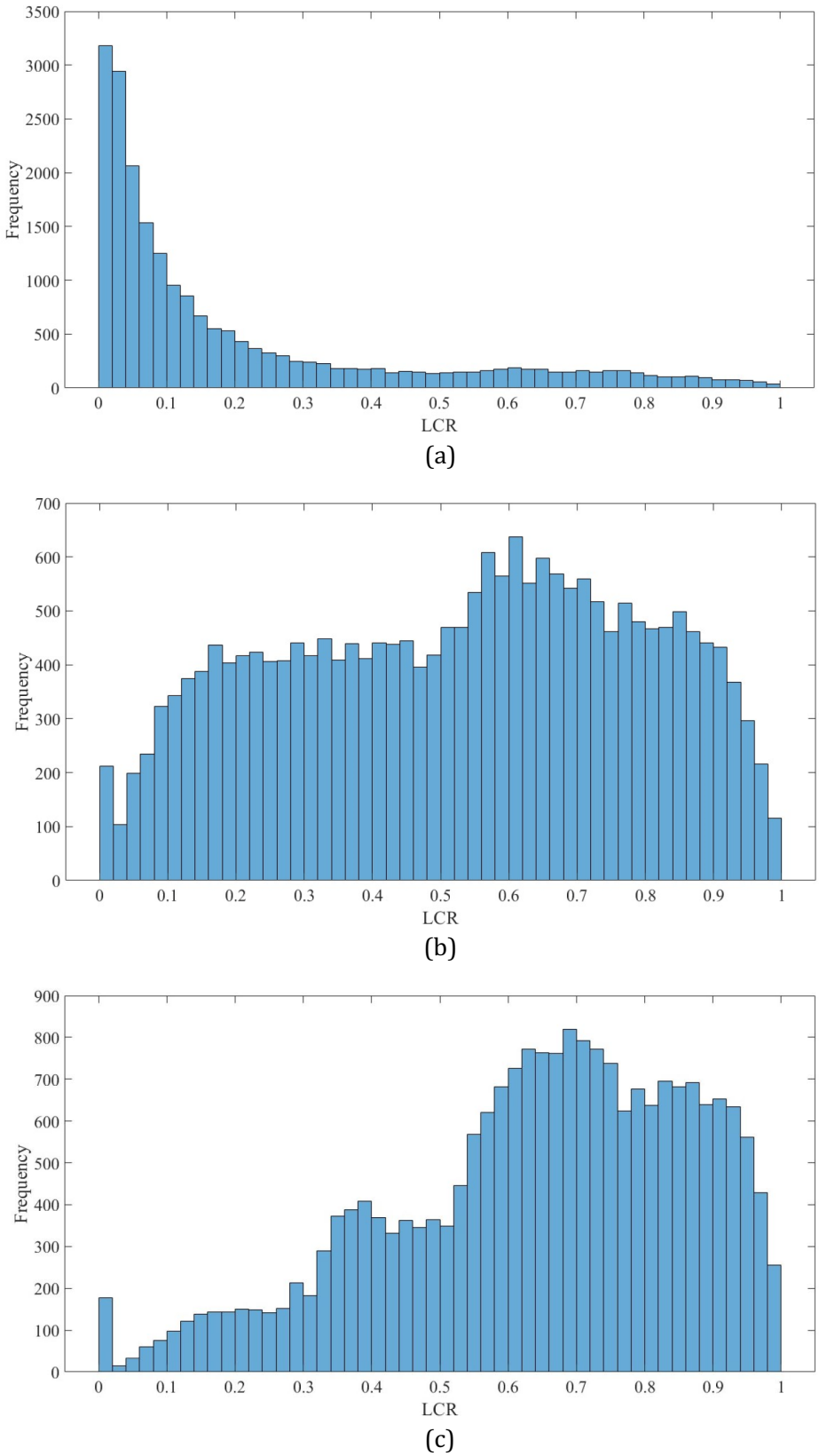


Fig. 9 Histograms of member *LCR*s during optimization using GSS_C with RDL = 6 for 21,212-member free-form double-layer grid: **a** first feasible design (iter = 32), **b** an interim design (iter = 85), and **c** best feasible design (iter = 232)

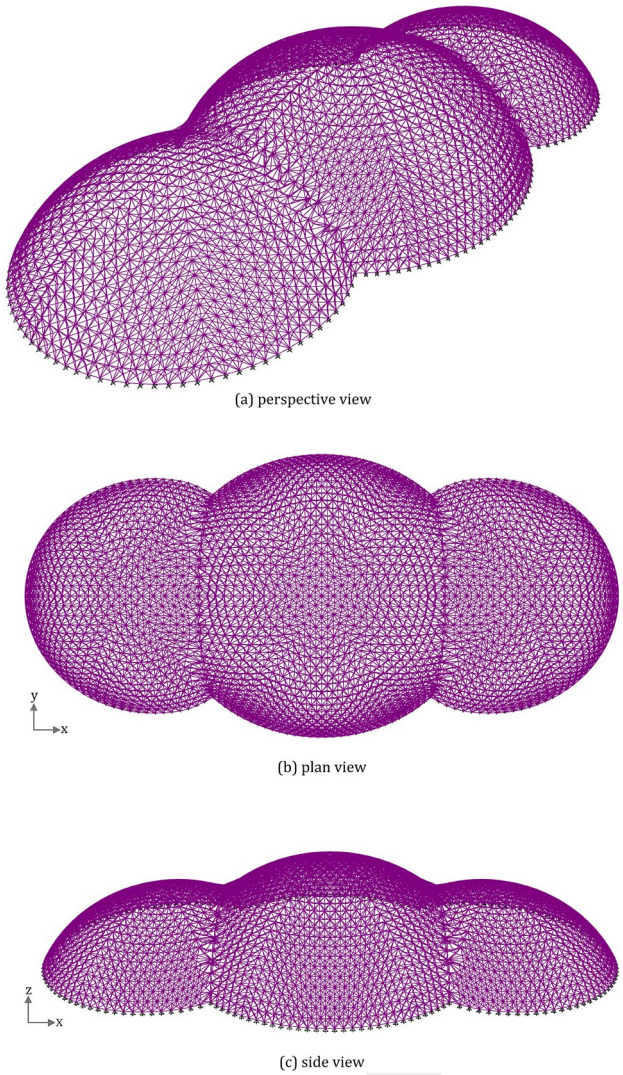


Fig. 10 25,514-member double-layer multi-dome: **a** perspective view, **b** plan view, and **c** side view

based on multiscale shape and topology optimization techniques (Fujioka et al. 2021; Ali and Shimoda 2022) could be another promising research direction to further reduce the material usage in large-scale structural systems.

Together with advancements in computation technologies, developing optimization algorithms for large-scale structural systems has become more popular in recent years (Kazemzadeh Azad et al. 2014b; Talatahari et al. 2022; Azizi et al. 2022; Talatahari et al. 2022; Kaveh et al. 2021). It should be noted that in structural optimization of steel skeletal structures, terms such as large-scale, full-scale, high-rise, and real size are often used either to reflect the size of the structure or the large number of structural members involved in the design process rather than the number of design variables. In other words, the design optimization problem of a large-scale steel skeletal structure may not necessarily represent a high-dimensional problem instance due to the reduced search space by variable linking or structural member grouping.

From a practical viewpoint, structural member grouping, as a common variable linking method, is usually inevitable to comply with fabrication or construction requirements. However, from an optimization perspective, structural member grouping can be seen as a procedure that transforms a high-dimensional search space into a low-dimensional search space and usually sacrifices solution quality for computational efficiency. Accordingly, another main research need is to study the trade-off between the computational efficiency gained by member grouping of large-scale structural systems versus the associated solution quality. On the one hand, this requires developing scalable algorithms which can handle structural optimization problems with extremely high-dimensional search spaces. On the other hand, to deal with the curse of dimensionality (Bellman 1957) in the foregoing search spaces, investigating the efficiency of dimensionality reduction or variable linking techniques (Omidvar et al. 2017) could be fruitful.

Finally, to better understand the properties of structural optimization problems, it is also essential to study the inherent complexities of structural optimization problems and extract their main features as well as the corresponding effects on the performance of algorithms. Accordingly, the use of landscape analysis methods for optimization (Malan 2021; Ochoa et al. 2021; Engelbrecht et al. 2022) could help better understand the properties of challenging structural optimization problems and could pave the way for selecting the most suitable algorithms for practical applications.

7 Concluding remarks

It can be envisaged that very high-dimensional optimization will form one of the main directions of future research on structural optimization. In this regard, owing to the previous promising performance of the design-driven GSS algorithm in handling large-scale structural optimization problems, this work attempts to further enhance the converge properties of the GSS to deal with computationally

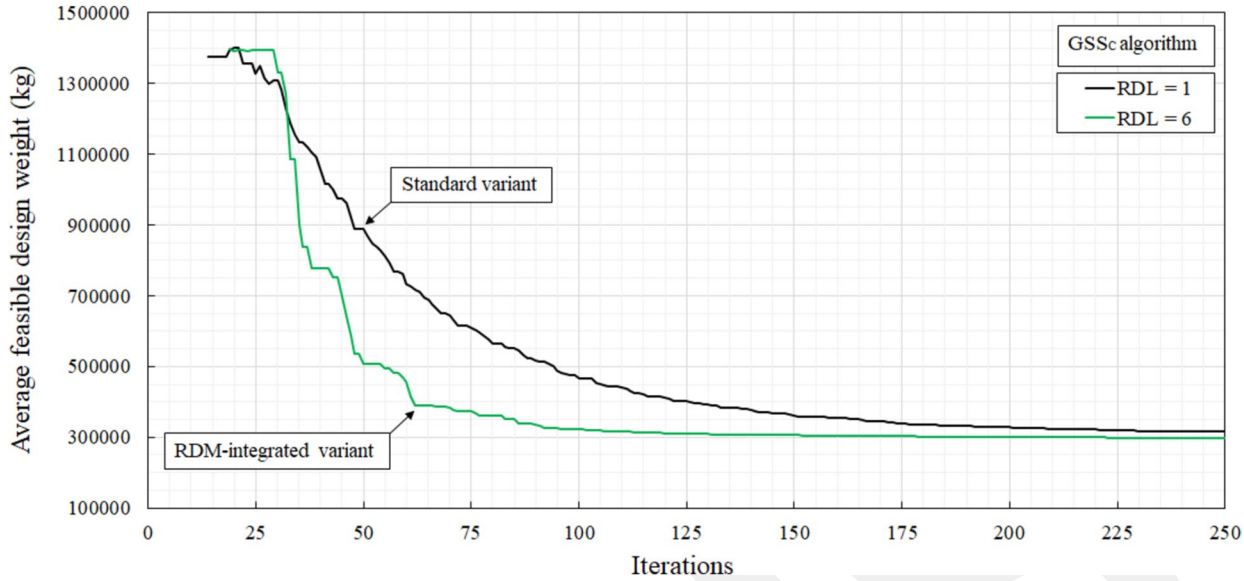
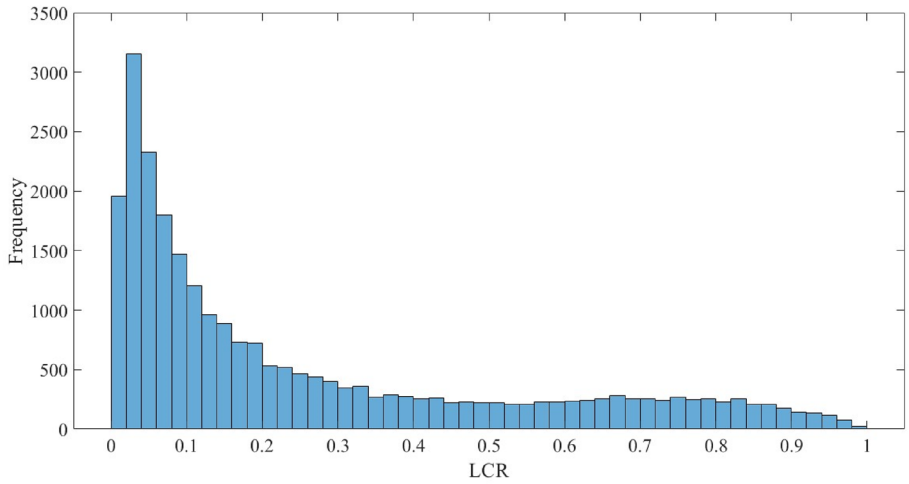
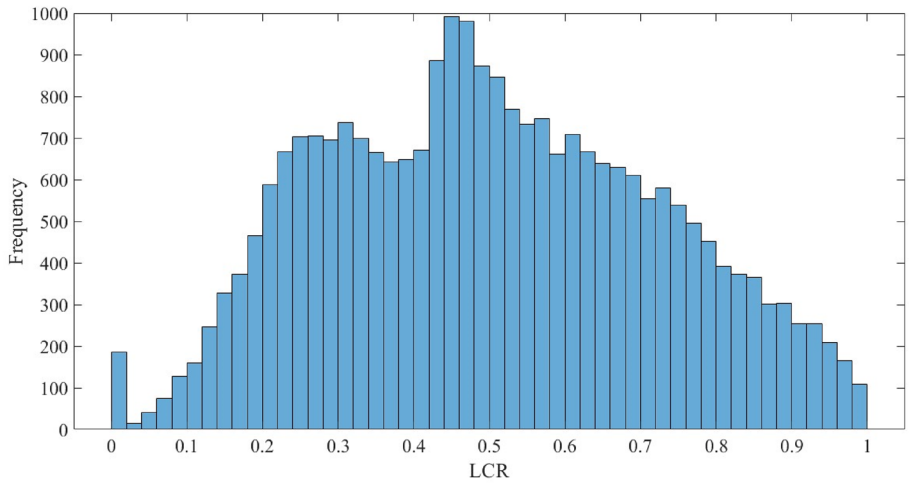


Fig. 11 Convergence histories for 25,514-member double-layer multi-dome using GSS_C

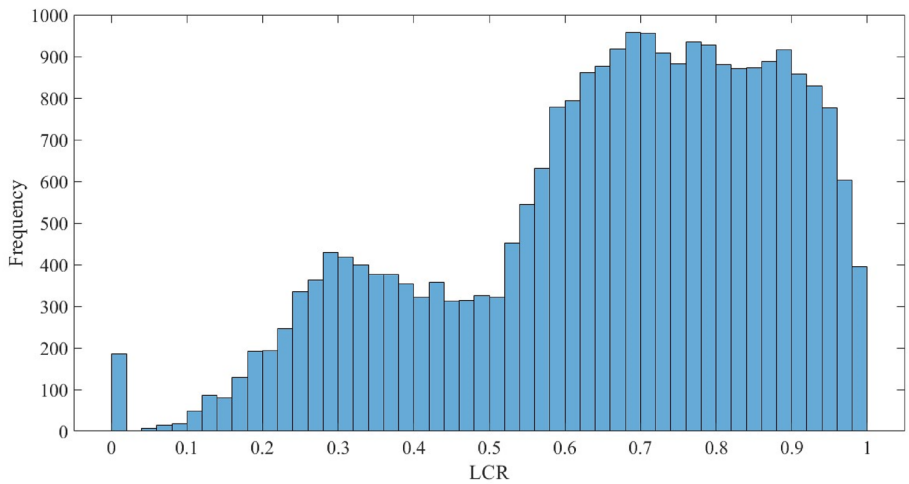
expensive, and very high-dimensional optimization problems of steel double-layer grids. For this purpose, a repair deceleration mechanism is proposed, and its efficiency is examined via challenging test examples of steel double-layer grids. First, a preliminary yet more comprehensive evaluation of the proposed methodology is carried out using two high-dimensional test instances from the literature. Once a suitable parameter setting is obtained based on rigorous analyses of the preliminary test examples, the applicability of the proposed RDM is further evaluated using two very high-dimensional instances of steel double-layer grids, namely a 21,212-member free-form double-layer grid, and a 25,514-member double-layer multi-dome. The obtained numerical results demonstrate that the proposed RDM can significantly enhance the convergence rate of the GSS algorithm and render it an efficient tool for tackling very high-dimensional sizing optimization problems of steel double-layer grids with up to 25,514 design variables. Further research is required to assess the usefulness of the proposed approach in sizing optimization of other types of large-scale structural systems. This can further be enriched by tackling high-dimensional structural optimization problems under dynamic loads and structural frequency constraints. Another possible extension could include efficient strategies to reduce the size of the search space in extremely high-dimensional structural optimization problems.



(a)



(b)



(c)

Fig. 12 Histograms of member *LCR*s during optimization using GSS_C with RDL = 6 for 25,514-member double-layer multi-dome: **a** first feasible design (iter = 23), **b** an interim design (iter = 70), and **c** best feasible design (iter = 226)

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00158-024-03898-5.

Acknowledgements Not applicable

Funding Open access funding provided by Óbuda University.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Replication of results Additional data for replication of the test examples are available as supplementary material.

References

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

Ali MA, Shimoda M (2022) Toward multiphysics multiscale concurrent topology optimization for lightweight structures with high heat conductivity and high stiffness using MATLAB. Struct Multidisc Optim 65:207

American Institute of Steel Construction (AISC) (1994) Manual of steel construction, load & resistance factor design, 2nd edn. Chicago

Azizi M, Talatahari S, Basiri M, Shishehgarkhaneh MB (2022) Optimal design of low- and high-rise building structures by Tribe-Harmony search algorithm. Decis Anal J 3:100067

Babaei M, Sheidaii MR (2014) Automated optimal design of double-layer latticed domes, using particle swarm optimization. Struct Multidisc Optim 50:221–235

Bellman RE (1957) Dynamic programming. Princeton University Press, Princeton

Charney FA (1993) Economy of steel framed buildings through identification of structural behavior. In: National steel construction conference, Orlando, FL, AISC, pp 1–33

Engelbrecht AP, Bosman P, Malan KM (2022) The influence of fitness landscape characteristics on particle swarm optimisers. Nat Comput 21:335–345

Erbatur F, Al-Hussainy MM (1992) Optimum design of frames. Comput Struct 45(5–6):887–891

Feury C, Geradin M (1978) Optimality criteria and mathematical programming in structural weight optimization. Comput Struct 8(1):7–17

Flager F, Soremekun G, Adya A, Shea K, Haymaker J, Fischer M (2014) Fully constrained design: a general and scalable method for discrete member sizing optimization of steel truss structures. Comput Struct 140:55–65

Fujioka M, Shimoda M, Ali MA (2021) Shape optimization of periodic microstructures for stiffness maximization of a macrostructure. Compos Struct 268:113873

Gholizadeh S, Torkzadeh P, Jabarzadeh S (2013) Optimum shape design of double-layer grids by quantum behaved particle swarm optimization and neural networks. Int J Optim Civ Eng 3(1):85–98

Hajela P, Berke L (1991) Neurobiological computational models in structural analysis and design. Comput Struct 41(4):657–667

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, Ilchi Ghazaan M (2016) Optimum design of large-scale truss towers using cascade optimization. Acta Mech 227:2645–2656

Kaveh A, Moradveisi M (2020) Simultaneous shape and size optimization of double-layer grids with nonlinear behavior. Periodica Polytechnica Civ Eng 64(4):1007–1025

Kaveh A, Servati H (2002) Neural networks for the approximate analysis and design of double layer grids. Int J Space Struct 17(1):77–89

Kaveh A, Khodadadi N, Azar BF, Talatahari S (2021) Optimal design of large-scale frames with an advanced charged system search algorithm using box-shaped sections. Eng Comput 37:2521–2541

Kazemzadeh Azad S, Hasançebi O (2015) Discrete sizing optimization of steel trusses under multiple displacement constraints and load cases using guided stochastic search technique. Struct Multidisc Optim 52:383–404

Kazemzadeh Azad S, Aminbakhsh S (2021) High-dimensional optimization of large-scale steel truss structures using guided stochastic search. Structures 33:1439–1456

Kazemzadeh Azad S, Aminbakhsh S (2022) ε-constraint guided stochastic search with successive seeding for multi-objective optimization of large-scale steel double-layer grids. J Build Eng 46:103767

Kazemzadeh Azad S, Hasançebi O, Saka MP (2014a) Guided stochastic search technique for discrete sizing optimization of steel trusses: a design-driven heuristic approach. Comput Struct 134:62–74

Kazemzadeh Azad S, Hasançebi O, Kazemzadeh Azad S (2014b) Computationally efficient optimum design of large scale steel frames. Int J Optim Civ Eng 4:233–259

Lagaros ND, Plevris V, Papadrakakis M (2005) Multi-objective design optimization using cascade evolutionary computations. Comput Methods Appl Mech Eng 194(30–33):3496–3515

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

Malan KM (2021) A survey of advances in landscape analysis for optimisation. Algorithms 14(2):40

Mashayekhi M, Salajegheh E, Salajegheh J, Fadaee MJ (2012) Reliability-based topology optimization of double layer grids using a two-stage optimization method. Struct Multidisc Optim 45:815–833

Moghimi M (2006) Formex configuration processing of compound and freeform structures. Ph.D. thesis, University of Surrey, UK

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

Nooshin H, Disney P (2000) Formex configuration processing I. Int J Space Struct 15(1):1–52

Nooshin H, Disney P (2001) Formex configuration processing II. Int J Space Struct 16(1):1–56

Nooshin H, Disney P (2002) Formex configuration processing III. Int J Space Struct 17(1):1–50

Nooshin H, Kamyab R, Samavati OA (2017) Exploring scallop forms. Int J Space Struct 32(2):84–111

Ochoa G, Malan KM, Blum C (2021) Search trajectory networks: a tool for analysing and visualising the behaviour of metaheuristics. Appl Soft Comput 109:107492

Omidvar MN, Yang M, Mei Y, Li X, Yao X (2017) DG2: A faster and more accurate differential grouping for large-scale black-box optimization. IEEE Trans Evol Comput 21(6):929–942

Papadrakakis M, Lagaros ND, Tsompanakis Y (1999) Optimization of large-scale 3-D Trusses using evolution strategies and neural networks. Int J Space Struct 14(3):211–223

Papadrakakis M, Lagaros ND, Fragakis Y (2003) Parallel computational strategies for structural optimization. Int J Numer Methods Eng 58:1347–1380

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

Patnaik SN, Gendy AS, Hopkins DA (1994) Design optimization of large structural systems with substructuring in a parallel computational environment. Comput Syst Eng 5(4–6):425–440

Patnaik SN, Coroneos RM, Hopkins DA (1997) A cascade optimization strategy for solution of difficult design problems. Int J Numer Methods Eng 40:2257–2266

Rajasekaran S (2001) Optimization of large scale three dimensional reticulated structures using cellular genetics and neural networks. Int J Space Struct 16(4):315–324

Ramaswamy GS (1997) Case study of barrel vault space frames for the platform shelter roofs of the thirumyilai station of the madras rapid transit system, the international course on space structures. Anna University, Chennai

Saka MP (1990) Optimum design of pin-jointed steel structures with practical applications. J Struct Eng ASCE 116(10):2599–2620

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

Spengemann F, Thierauf G (1991) Optimization of large-scale structures: optimality criteria and domain decomposition. Comput Struct 41(3):495–499

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

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 21:825–842

Talatahari S, Azizi M, Toloo M, Baghalzadeh Shishehgarkhaneh M (2022) Optimization of large-scale frame structures using fuzzy adaptive quantum inspired charged system search. Int J Steel Struct 22:686–707

Torkzadeh P, Jaffari T, Shojaee S (2015) Layout optimization of double-layer grids using modified genetic algorithm based on fuzzy inference system. Scientia Iranica Trans A Civ Eng 22(5):1723–1733

Zhou M, Rozvany GIN (1992) DCOC: an optimality criteria method for large systems part I: theory. Struct Optim 5:12–25

Zhou M, Rozvany GIN (1993) DCOC: an optimality criteria method for large systems part II: algorithm. Struct Optim 6:250–262