RESEARCH

Topologically preoptimized ground structure (TPOGS) for the optimization of 3D RC buildings

Yohannes L. Alemu[1] · Bedilu Habte[2] · Tom Lahmer[1] · Girum Urgessa[3]

Abstract

As an optimization that starts from a randomly selected structure generally does not guarantee reasonable optimality, the use of a systemic approach, named the ground structure, is widely accepted in steel-made truss and frame structural design. However, in the case of reinforced concrete (RC) structural optimization, because of the orthogonal orientation of structural members, randomly chosen or architect-sketched framing is used. Such a one-time fixed layout trend, in addition to its lack of a systemic approach, does not necessarily guarantee optimality. In this study, an approach for generating a candidate ground structure to be used for cost or weight minimization of 3D RC building structures with included slabs is developed. A multiobjective function at the floor optimization stage and a single objective function at the frame optimization stage are considered. A particle swarm optimization (PSO) method is employed for selecting the optimal ground structure. This method enables generating a simple, yet potential, real-world representation of topologically preoptimized ground structure while both structural and main architectural requirements are considered. This is supported by a case study for different floor domain sizes.

Keywords Ground structure · TPOGS · Topology optimization · 3D reinforced concrete buildings · Structural optimization

Abbreviations

ACO	Ant colony optimization
PSO	Particle swarm optimization
RC	Reinforced concrete
TPOGS	Topologically pre-optimized ground structure
A_{B_i}	Area of functional block B_i
A_D	Area of floor domain
A_u	Unoccupied area of floor domain
B_i, B_j, B_p	Functional blocks i, j and pre-located block
$B_{ix}, B_{iy}, B_{jx}, B_{jy}, B_{px}, B_{py}$	X and Y components
C_i	Column i
d	Effective depth of a beam
F	Combined objective function
F_i	Objective function i
F_{inorm}	Normalized objective function i
l	Length of a beam
L_{B_i}	Center to center distance between floor domain and a functional block B_i
$L_{B_{ix}}, L_{B_{iy}}$	X and Y components of L_{B_i}
$L_{dl}, L_{dr}, L_{dt}, L_{db}$	Distances between C_i and edges of B_i
L_{ij}	Center to center distance between B_i and B_j
$L_{ijx}, L_{ijy}, L_{ipx}, L_{ipy}$	X and Y components
L_{tol}	Tolerable distance of inscribed C_i and nearest edge of floor domain
L_x, L_y	Lengths along X and Y axes
N_b	Number of beams
N_{bayX}, N_{bayY}	Number of bays along X and Y axes
N_c	Number of columns
O, O_i, O_j	Centers
R_{cb}	Number of columns to number of beams ratio

✉ Yohannes L. Alemu
lisanyoh@gmail.com

1 Institute of Structural Mechanics, Bauhaus University, Weimar, Germany

2 School of Civil and Environmental Engineering, Addis Ababa Institute of Technology, Addis Ababa, Ethiopia

3 Department of Civil, Environmental and Infrastructure Engineering, George Mason University, Fairfax, VA, USA

S_{max}, S_{max}	Maximum and minimum spacing of columns
S_{opp}	Spacing immediately opposite to the movement direction of C_i
S_{sam}	Spacing immediately same to the movement direction of C_i
w_i	Weight for F_i

Introduction and state of the art

Optimizing reinforced concrete (RC) structures is complex compared to that of steel-made structures because of the nonhomogeneous property of the concrete and the existence of reinforcing bars (Kaveh & Zakian, 2014).

While dealing with real-world optimization of building structures, the optimal placement of different functional regions (blocks) within a given domain may primarily be considered (Hsu & Hsu, 2005). The research in Chen and Chang (2006) discussed different floor optimization approaches for a predetermined domain. In the work of Michalek et al. (2002) the fundamental starting point was a unit rectangle; then, every polygon was defined in terms of a combination of this unit rectangle. The research in Merrell et al. (2010) addressed two floor layout techniques, the two-move approach: sliding walls and swapping rooms.

Wenming et al. (2018) considered both geometric and topological requirements. They used a polygonal boundary, and regions resulted from combining rectangles. Their objective function was to minimize unoccupied space in the initial domain.

All the above and most of the reinforced concrete optimization studies focus mostly on 2D structures (Carvalho et al., 2020), while a 3D layout optimization was reported in Guo and Li (2017). Sharafi et al. (2012) used ant colony optimization (ACO) for column layout optimization for 3D reinforced concrete frames. They developed an X–Y axis construction graph that indicates multiple possibilities for locating column nodes.

Nimtawat and Nanakorn (2010) presented beam-slab optimization of the rectilinear floor system in which the locations of columns are predefined. Connections of T-beams, beams connecting out of column location, are also considered. Then, optimal layouts are optimized using the genetic algorithm.

To date, the objective function has mainly been the minimization of the unoccupied area within a given domain. In this research, however, a second objective function is added to collect unoccupied areas towards a preferred location where additional functional blocks may be obtained and assigned.

A ground structure approach is commonly used as a starting point for truss and steel-framed structures (Changizi & Jalalpour, 2018) and credit is given to Dorn (1964) for using this approach first. A ground structure is a structure where optimization starts, and through iteration, its members and topology are changed to attain an optimal structure. As the node positions and numbers of connected members on the initial ground structure highly dominate the computational burden and optimality, different studies have been conducted in this regard (Han & Wen, 2018; Larsen et al., 2018; Zhang et al., 2016).

A conventional or fully connected ground structure results in a complex optimization process and consumes considerable computer capacity and, of course, time (Sokól, 2011). Thus, a fully connected ground structure may seem to be a continuous domain (region almost fully covered by many members) because of exceedingly many members (Pritchard et al., 2005). Overlapping members, the inclusion of slender members, and structural instability are challenges to overcome when thinking of such a ground structure (Ohsaki, 2016).

Another approach, which is selective under some optimality criteria, results in a less dense ground structure. This second approach is addressed in different ways in which unfit members might be gradually removed and fit members added, or levels of connectivity are generated to include the preferred number of members based on required solution accuracy (Ghoddosian et al., 2018; Li et al., 2021; Sanders et al., 2017).

Ranalli et al. (2018) used a ground structure to minimize the total installed costs of steel frame structures. They applied topology and size optimizations sequentially as outer and inner loops. Sokól (2011) applied a ground structure for topology optimization of large-scale trusses. His approach implemented shorter possible member to node connections in the beginning and then gradually included longer potential members. According to his method, while new members are possibly added, less important older members are removed.

The abovementioned works, however, are not further extended to RC structures because of the orthogonal member connectivity orientation. It is obvious that reinforced concrete structures have attracted interest in the research world, both as reinforced concrete (Alkam & Lahmer, 2019; Harirchian et al., 2020; Jaouadi et al., 2020; Kaveh, 2017) and composed of structural steel (Liu et al., 2016; Zhu et al., 2022) indicating that such optimization is of great importance. In this research, an approach for generating a ground structure for optimizing reinforced concrete buildings is proposed.

To minimize complexity, structures are usually optimized at a component level rather than at the system level. Such a representation is, however, far from the real-world problem, as component optimization cannot always guarantee system optimization. Component-level optimization refers to a separate optimization of a structural segment and reassembly of

such segments after optimization occurs, while system-level optimization refers to considering the whole structural system during optimization. In the literature, slab (Ahmadkhanlou & Adeli, 2005; Kaveh & Abadi, 2011), beam (Babiker et al., 2012; Lepš & Šejnoha, 2003) and frame (Kaveh & Sabzi, 2012; Kaveh et al., 2021) optimizations are treated separately depicting component-level optimization.

It is thus important to have a preoptimized 3D RC building including slab considering the real-world functional requirement. This approach is also useful when considering simultaneous optimization, which provides better optimal output (Miguel et al., 2013). In the case of two-step simultaneous optimization, topology optimization can be carried out first, and then size and shape optimizations are performed. This enables us to handle the most difficult topology optimization separately (Begg & Liu, 2000; Cheng & Jiang, 1992).

For topology optimization of a building structure, it seems feasible to handle floor, column, and beam layout optimization sequentially, although they are not explicitly independent of each other. This is done to satisfy functional requirements in building structures. Thus, columns and beams are located considering the alignments of functional floor blocks. This research attempts to answer the question, "How can a candidate, systemic ground structure be generated for 3D framed RC building structures to facilitate system and two-step simultaneous optimization during the implementation of a cost minimization objective function?"

Different metaheuristic algorithms are used and compared based on different bases for optimizing RC structures (Kaveh & Behnam, 2013; Kaveh et al., 2020) and PSO is also recommended for its simplicity and ease of application (Elbes et al., 2019; Perez & Behdinan, 2007).

Ground structure generation

In optimization, the structure selected for startup, named the ground structure, matters considerably; the better the initial approach is, the better the optimization result will be Hagishita and Ohsaki (2009). A method for generating a 3D ground structure for reinforced concrete buildings is devised below. It may be considered as both a floor layout optimization and a frame layout optimization. Many of such candidate ground structures are generated and used for separate runs of cost or weight minimization.

Instead of handling a large-scale fully connected ground structure with its entire computational burden (Zegard & Paulino, 2014) and, of course, nonpracticality in the case of RC buildings consisting mainly of orthogonal members, this approach provides a simplified yet better representation of the ground structure for optimizing 3D RC slab frames. It aims at selecting a feasible candidate ground structure

examined under some criteria before cost/weight minimization is applied. Such candidate ground structures are thus topologically preoptimized.

Floor optimization

Floor layout optimization aims at the optimal placement of a given set of functional blocks on pre- or postdefined floor areas. Given a rectangular floor domain with L_x and L_y side lengths along the X and Y axes, respectively, and a rectangular functional block B_i with sides $B_{i,x}$ and $B_{i,y}$, let L_{Bi} represent the length between the centres of the floor domain and the functional block, with $L_{Bi,x}, L_{Bi,y}$ being its X and Y components, respectively.

A functional block B_i represents a space for a specified room (such as a dining room or bedroom) or a specified household space (in which all required rooms for a specified household size are included and considered as one space).

Objective function

Referring to Fig. 1 above, floor optimization is defined by a multiobjective (Fan et al., 2015; Zavala et al., 2014) optimization function has been formulated that consists of two conflicting objective functions.

$$\max F_1 := \sum_{i=1}^{n} L_{B_i}. \tag{1}$$

$$\min F_2 := \frac{A_u}{A_D}. \tag{2}$$

where F_1 and F_2 are the two objective functions, L_{B_i} is the distance from centre O to the centre of functional block i and the domain, A_D is the area of the domain, and A_u is the unoccupied area within the domain. The first function defined in Eq. (1) locates as many functional blocks within the domain as possible from a chosen reference, in this case, the centre
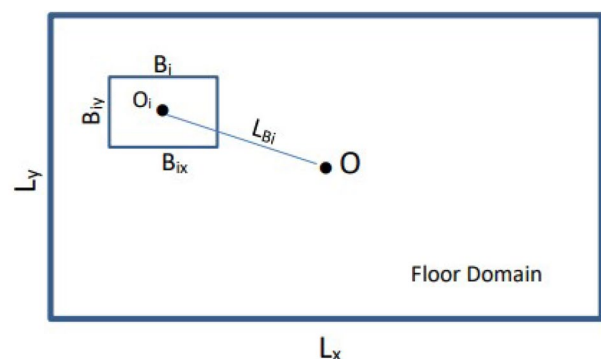

Fig. 1 Floor domain and functional blocks

of the domain. Thus, if free space is available, it will be around that chosen reference point. This approach attempts to collect unoccupied space fragments towards the reference point and enables further use of such space.

The second objective function, shown in Eq. (2), deals with the reduction in unoccupied areas. This is enabled by maximizing the area of a functional block through the optimization process without violating the bounds.

To solve the multi-objective problem in Eq. (2), we use a weighted sum approach where two objective functions are combined based on a weighting technique. One such technique used here is the weighted sum method, as provided in Eq. (3). According to this method, each objective function is multiplied by its respective assigned weight (von Butler, 2019).

$$\min F := (-w_1 F_{1,norm} + w_2 F_{2,norm}).$$
$$F_{i,norm} = \frac{F_i - F_{\min}}{F_{\max} - F_{\min}}, \quad i = 1, 2. \tag{3}$$

where F is the combined objective function, w_1 and w_2 refer to the weights and the negative sign in w_1 converts F_1 from maximization to minimization, $F_{i,norm}$ is the normalized ith objective function, $F_{\min}$ is a possible minimum value to the specified objective function, and $F_{\max}$ is the possible maximum value to the specified objective function.

Constraints and bounds

The first sets of constraints keep functional blocks to remain inside the given floor domain. A functional block will not be located out of the domain if either of the orthogonal distances from the domain centre to the far edges of the functional block does not exceed their corresponding half-lengths of the floor domain.

$$\frac{L_{B_{i,x}} + \frac{B_{i,x}}{2}}{\frac{L_x}{2}} - 1 \le 0.$$
$$\frac{L_{B_{i,y}} + \frac{B_{i,y}}{2}}{\frac{L_y}{2}} - 1 \le 0. \tag{4}$$

Equation (4) defines constraints of the normalized limits to contain all functional blocks within the floor domain

$$\sum_{i=1}^{n} \frac{A_{B_i}}{A_D} - 1 \le 0. \tag{5}$$

A constraint shall also limit the sum of the area of functional blocks against the area of the domain, Eq. (5). For area A_{B_i} of functional block ith in the number of functional blocks n and domain area A_D, the sum of the areas of all functional

blocks placed in the floor shall not exceed the gross area of that floor domain.

$$\frac{A_{B_{i,\min}}}{A_{B_i}} - 1 \le 0.$$
$$\frac{A_{B_i}}{A_{B_{i,\max}}} - 1 \le 0. \tag{6}$$

The values for the lower and upper bounds of a functional block are controlled based on Ernst & Neufert, (2000). Thus, the bounds for the sides of functional blocks constitute set of constraints, as is in Eq. (6).

$$\frac{A_u}{A_{B,i,\max}} - 1 \le 0. \tag{7}$$

Furthermore, the unoccupied area A_u shall be less than the maximum area $A_{B_{i,\max}}$ allowed for a single functional block, Eq. (7). This constraint, however, may be too strong to be satisfied because of many factors, such as the number of households and floor domain area. Thus, it may be more relaxed (such as comparing to areas of two or more functional blocks or the constraint might be avoided).

B_i and B_j in Fig. 2 are any two functional blocks within the given domain, $B_{i,x}$, $B_{i,y}$, $B_{j,x}$ and $B_{j,y}$ are their respective sides along X and Y axes, O_i and O_j are their respective centers, L_{ij} is the distance between the two centers and $L_{ij,x}$ and $L_{ij,y}$ are components of L_{ij} along X and Y axes respectively.

$$\frac{B_{i,x} + B_{j,x}}{2L_{ij,x}} - 1 \le 0.$$
$$\frac{B_{i,y} + B_{j,y}}{2L_{ij,y}} - 1 \le 0. \tag{8}$$

Functional blocks will not overlap with each other if the following requirement is satisfied along one orthogonal direction: the sum of the half-lengths of each functional block
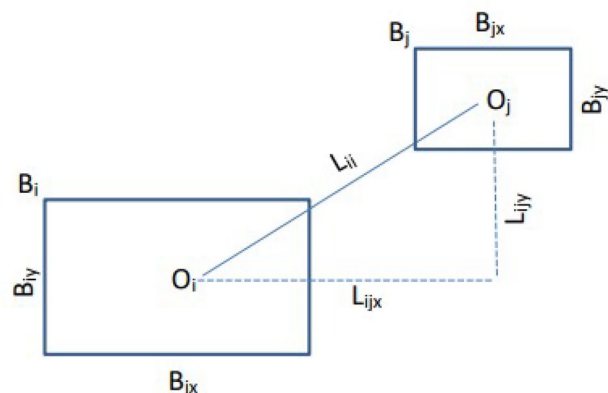

Fig. 2 Overlapping of functional blocks

along the same orthogonal direction equals or exceeds the corresponding centre-to-centre distance of the functional blocks. As shown in Fig. 2, functional blocks B_i and B_j do not overlap if the relations in Eq. (8) hold true.

Pre-located or restricted functional blocks

Locations of some functional blocks in a given building, such as toilets, staircases and corridors, might be fixed for better domain functionality. In such a case, such prelocations shall not overlap and be inscribed with the functional blocks.

Given B_p as a prelocated functional block within the given domain, $B_{i,x}$, $B_{i,y}$, $B_{p,x}$ and $B_{p,y}$ are their respective sides along the X and Y axes, $L_{ip,x}$ and $L_{ip,y}$ are the X and Y components of the centre-to-centre distance between B_i and B_p, and overlap between B_i and B_p shall be avoided as follows:

$$\frac{B_{i,x} + B_{p,x}}{2L_{ip,x}} - 1 \le 0.$$
$$\frac{B_{i,y} + B_{p,y}}{2L_{ip,y}} - 1 \le 0. \tag{9}$$

Therefore, for a prelocated functional block p, the corresponding constraint is provided by Eq. (9).

Implementation of the ordinary PSO-based floor layout optimization (Fan & Chiu, 2007) is thus governed by the combined objective function in Eq. (3), subjected to the constraints provided above from Eq. (4) to Eq. (9).

Thus, while generating a ground structure for a 3D RC building, the floor optimization stage enables the feasible allocation of randomly located functional blocks within a given domain considering the prelocated functional blocks. The sizes of the functional and the prelocated functional blocks can vary based on their upper and lower bounds, and they can also assume any orthogonal orientation for the better fulfilment of the multiobjective function defined above earlier.

Frame optimization

Objective function

A preoptimized ground structure, to be developed before implementing the grand cost optimization, should be built of fewer columns and more beams. These conflicting objectives pave the road towards cost minimization.

While it is generally obvious that a lower number of columns leads to a column lower cost, a greater number of beams, in contrast, contribute to the minimum cost through shorter-span beams, a smaller depth of slabs, and fewer internal forces on columns, which keeps the cost of each column as low as possible.

We now define:

$$\min R_{cb} := \frac{N_c}{N_b}, \tag{10}$$

where the objective function R_{cb} is the ratio of the number of columns to the number of beams and N_c and N_b denote the numbers of columns and the number of beams orthogonally connected to the columns, respectively. The objective function in Eq. (10) can be applied on the first orthogonal axes of the floor domain, as the remaining axes are only their offsets and do not affect the objective function.

The ratio R_{bc} can be computed using columns and beams along the first orthogonal axes only as the rest are offsets. Although the ratio computed using the whole numbers of columns and beams differs from the ratio computed using the first orthogonal axes only, the aim is to minimize the ratio, not to consider the ratio numerically for further computations, and this does not introduce any computational error.

Node location

In Fig. 3, C_i is an ith column within a functional block, and L_{dl}, L_{dr}, L_{dt} and L_{db} are distances between C_i and the left, right, top and bottom edges of B_i, respectively.

The frame layout starts by defining column nodes by considering the preoptimized floors, as shown in Fig. 3. Nodes are randomly generated and optimized for the objective function defined above considering all possible numbers of bays along the respective axes.

$$N_{bay,X} = \text{roundup}\left(\frac{L_x}{S_{\max}}, \frac{L_x}{S_{\min}}\right).$$
$$N_{bay,Y} = \text{roundup}\left(\frac{L_y}{S_{\max}}, \frac{L_y}{S_{\min}}\right) \tag{11}$$

where N_{bayX} and N_{bayY} are the number of bays along the X and Y axes and should be integers, and $S_{\max}$ and $S_{\min}$ are the maximum and minimum column spacing, respectively. N_{bayX} and N_{bayY} are integers lying between their possible minimum and maximum values.
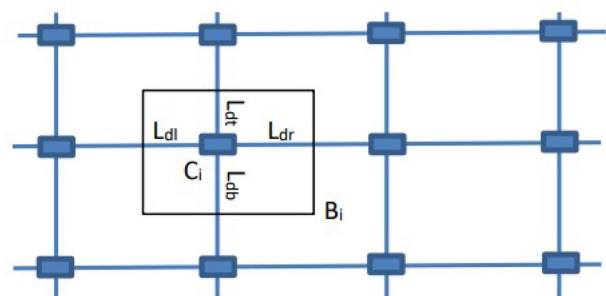

Fig. 3 Inscribed column treatment

Column escape

It is important to note that columns might be located within a functional block that could interfere with its functionality. Appropriate functions have been developed to treat such occurrences.

If the following condition holds true, moving column C_i along the side that satisfies the requirement is necessary. See Fig. 3 above.

$$L_{ox} > L_{tol} \quad \text{and} \quad L_{oy} > L_{tol}. \tag{12}$$

where $L_{ox} = \min(L_{dl}, L_{dr})$ and $L_{oy} = \min(L_{dt}, L_{db})$, L_{tol} is a tolerable distance that column C_i is inscribed, L_{ox} is the distance along X that C_i shall move, and L_{oy} is the distance along Y that C_i shall move. L_{tol} can be chosen based on different factors, one of which is based on the thickness of the inscribed column. While computing L_{ox} and L_{oy}, the respective minimum values are chosen such that columns can quickly escape from the functional blocks in which they are inscribed.

$$\frac{S_{min}}{S_{sam}} - 1 \leq 0 \quad \text{and} \quad \frac{S_{opp}}{S_{max}} - 1 \leq 0. \tag{13}$$

In Eq. (13), S_{sam} and S_{opp} refer to the immediate spacing of the considered column in the same direction of the move and in the opposite direction of the move, respectively.

If Eq. (12) is satisfied, the movement of the inscribed column C_i is constrained according to Eq. (13) such that the immediate spacing towards the direction in which column C_i moves is not less than S_{min} and the spacing opposite to the direction of the movement is not greater than S_{max}.

Beam connectivity

Five types of beam connectivity are formed in which the optimizer can use any of the types or their combinations. The length of a connecting beam can also be a criterion for selecting a connectivity type. Once the candidate ground structure is generated, the relevance of frame members based on their developed stress is checked through the grand cost optimization, and thus those with negligible relevance are removed (Fig. 4).

Beams usually have orthogonal connections to columns. These types of connecting beams are considered primary beams. In this study, however, diagonally emerging beams are also included. They only run from column to column diagonally and are identified as left-running and right-running diagonal beams. Other beam categories are those running orthogonally from the middle of the primary beams. However, such a fully connected beam may not be used during optimization, and the level of connectivity (type of
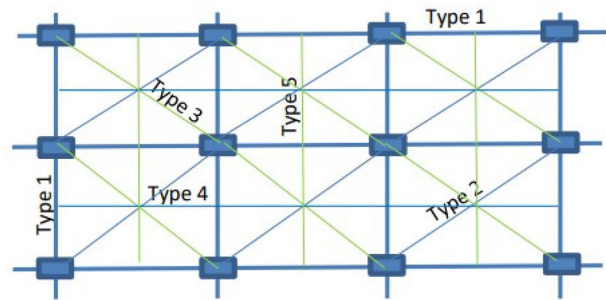

Fig. 4 Proposed types of beam layout

Table 1 Types of beam connectivity

Type of connectivity	Description
Type 1	Column to column orthogonally
Type 2	Column to column diagonally, from left to right and bottom to top or vice versa
Type 3	Column to column diagonally, from right to left and top to bottom or vice versa
Type 4	Connect mid points of Y-axis running type 1 beams
Type 5	Connect mid points of X-axis running type 1 beams

connectivity) is chosen by the designer. In this regard, only Type 1 beam connectivity or combination with other types listed in Table 1 above can be chosen. Alternatively, diagonal beams (Type 2, Type 3) may be included by setting a length constraint.

Implementation of the PSO-based frame layout optimization is thus governed by Eq. (10), subjected to the constraints provided above from Eq. (11) to Eq. (13).

The basic steps for generating a ground structure for 3D reinforced concrete building structures are depicted in Fig. 5. It is sequential; the floor layout is followed by the frame layout. The objective functions are those defined above.

MATLAB 2021a is used to automate the whole optimization process. First, scripts and functions that define an ordinary PSO subjected to a further refinement process are written. Second, scripts and functions are written for floor layout optimization. Third, the same is done for frame layout optimization. Finally, all the scripts and functions are integrated accordingly.

Case study (TPOGS)—residential apartment

A single-bedroom residential apartment with four, five, and 10 households on each floor is considered. One household consists of the required number of functional blocks. All that was discussed about functional blocks in the previous section is directly used for every household. In the four
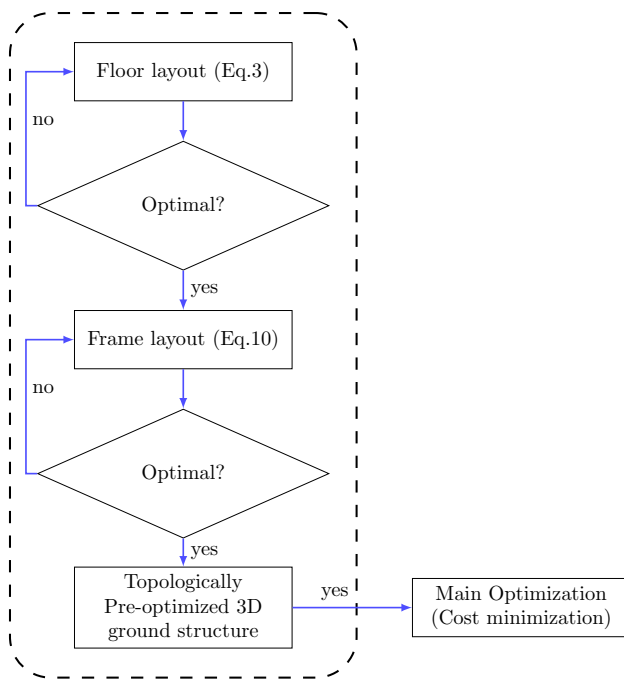
Fig. 5 TPOGS generating procedure

Table 2 Size range for each functional block in a household (Ernst & Neufert, 2000)

Functional blocks	Minimum		Maximum	
	Width/length	Area	Width/length	Area
Living and dining room	3.6	17	6.3	22.5
Bed room	2.8	13	9.3	26
Bath room with toilet	1.7	3.5	2.6	4.5
Kitchen	1.8	5.5	3.1	7.5
Storage	0.6	2	5	3
Stair	0.9		1.5	
Corridor	0.9		1.5	
Sum		41		63

household floor, the effectiveness of the defined objective function is examined considering the four corners of the domain.

In the five-household floor, we also attempt to examine how the number of households in excess of the number of corners of the given domain (which is four) is allocated and affect the optimization. Finally, in the 10-household floor, we attempt to represent many household cases. Each of the cases contains prelocated functional blocks and is bounded by a predefined domain size.

A single staircase is provided to floors with four and five households, while the floor with 10 households is provided with double staircases, each of which has a size of 3 by 5 m. There is also a corridor of size 5 by 7 m, the orientation of which is randomly generated to give freedom for optimal orientation. The area of each household varies between 41 and 63.5 m², and its side length is allowed to range from 4 to 13 m.

Using recommendations from Ernst & Neufert, (2000), the architectural data for functional blocks were collected. Each household in the above cases is composed of the functional blocks listed in Table 2.

This ground structure generation approach considers randomly selected household sizes within the stated bounds, considering the existence of prelocated functional block(s).

An ordinary PSO algorithm written on MATLAB 2021a running for 1000 iterations with a population size of 50 is used. Each case is subjected to 10 runs, and the optimal layout is identified. The size of each domain is determined

based on the maximum possible area of each household and the areas of prelocated blocks. While using the weight sum method to combine both objective functions under floor optimization, the value for the factor w is taken as 0.7.

Once the PSO is carried out, an automated refinement is performed in the implementation. It calculates unoccupied spaces and provides half of it to the pair of households in which the unoccupied space belongs without violating maximum area bounds. Columns within households (the columns still have the chance to be on the sides of functional blocks. This concept is more important in the case of optimizing functional blocks within a household than households themselves) with more than 50 cm indention (L_{tol} chosen to be 50 cm, i.e., assumed column thickness) are treated, and their placement is determined.

Frames are also subjected to such PSO for the stated number of iterations and population and run under the objective function in Eq. (9). The minimum spacing between columns is kept at 3 m considering the total width of the staircase, while the maximum spacing is limited to 7 m considering the l/d deflection requirement without any reducing factor according to Eurocode 2 (Sanders et al., 2017). The beam layout is generated using all five types of beam connectivity. Beams crossing staircases are automatically removed from the beam layout. The topology of the preoptimized structure is then determined, and it becomes a candidate for subsequent cost optimization.

Households are numbered based on their random generation. A fully inscribed functional block at the floor represents the staircase, while the region between the inscribed and circumscribing functional blocks represents a corridor. A bold dot on the floor represents the centre of the domain, which is chosen as a reference point for the objective function under floor optimization, as stated in "Floor optimization" section. Values for best costs indicated on the convergence graph do not represent anything, as they result from two combined and normalized objective functions, as

stated in "Floor optimization" section. They are only used to identify the optimal ground structure with the minimum numerical value.

As shown in Fig. 6, every household is fit to the domain corner to corner, which is only possible for a floor domain consisting of up to four households. This ensures that the objective function is well satisfied. The frame layout procedure developed in this study recommended relocating a column within household number 2, as shown by the heavy dot in Fig. 6. Then, the selected optimal column layout plan is shown. Finally, the 3D candidate ground structure is provided. There are 12 beams and seven columns on the

first orthogonal axes, and the value for R_{cb} is 0.583. This value is of course the least possible or optimal value.

The results of the ground structure generation for a floor having five households are shown in Fig. 7. While all four households are arranged towards each corner, the fifth one is also oriented to satisfy the objective function under the floor optimization subsection.

Figure 8 shows the resulting ground structure for a 10-household floor with two staircases. The convergence graph in Fig. 8 approached an optimum after relatively more iterations compared to the above two figures.

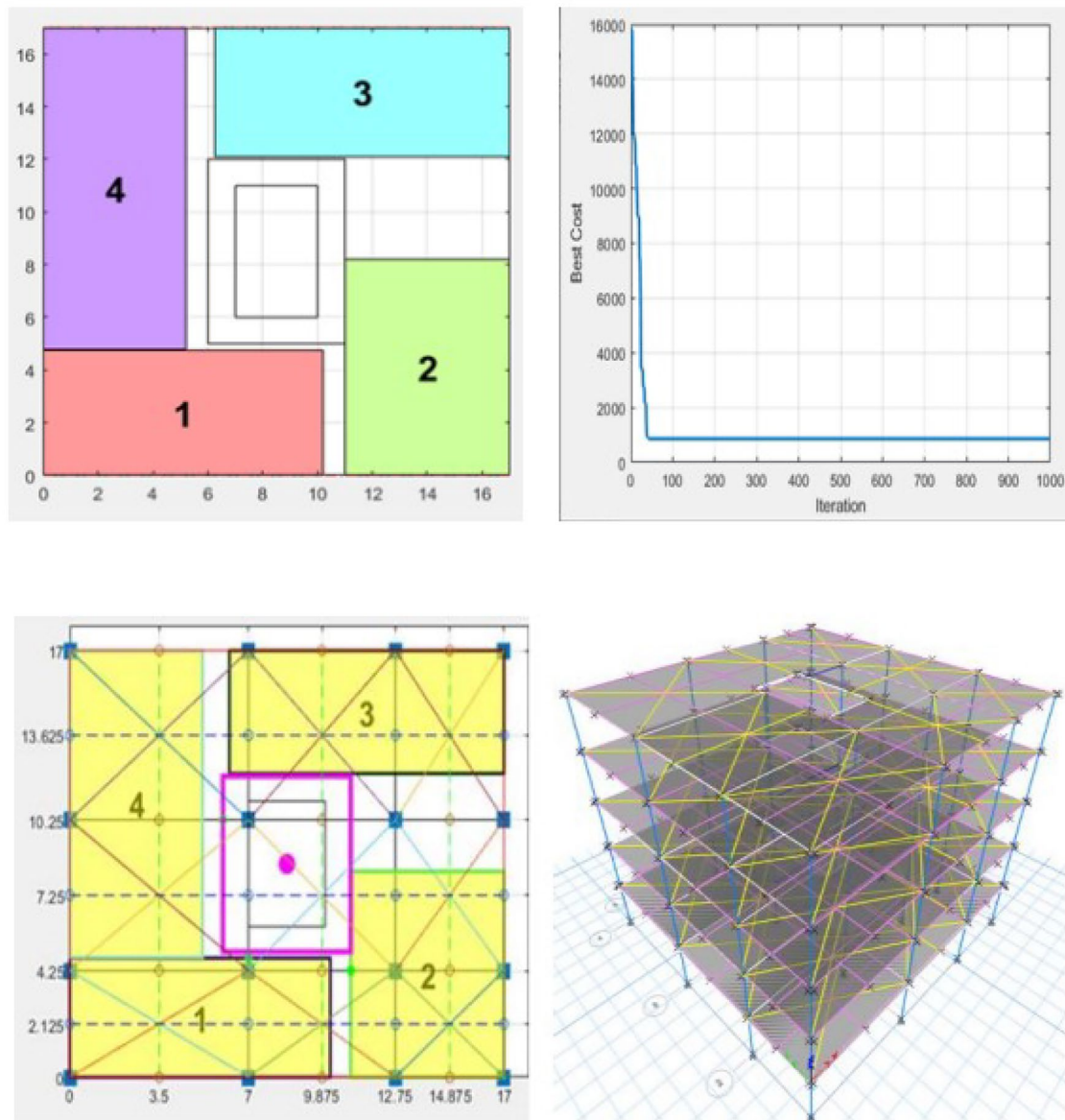The PSO convergence graphs depict that rapid convergence is completed at a higher number of iterations as the

Fig. 6 A four households floor with a domain size of 289 m^2: Optimally placed households; PSO convergence graph; floor layout including beams' and columns' initial layout; 3D ground structure as a candidate building model

number of households increases. In this case, for the four households before 100 iterations, for the five households before 200 iterations and for the 10-household cases after 400 iterations (Table 3).

This third table summarizes the domain area, summed household area, restricted functional block area and number of column-to-beam ratios R_{cb} used for the ground structure generation for each household case study. The
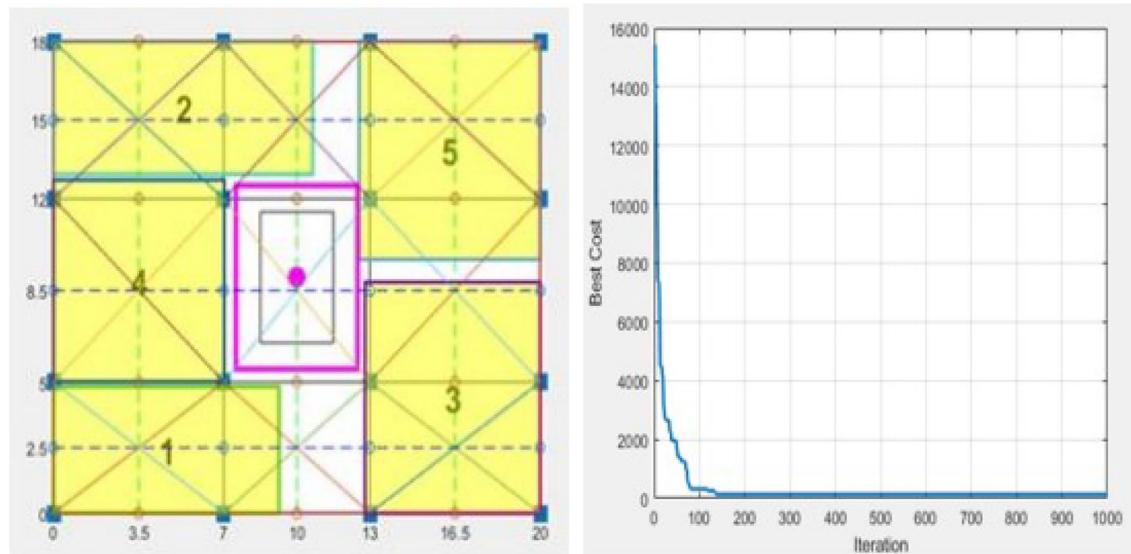

Fig. 7 A five households floor with a domain size of 360 m^2: Floor layout including beams and columns and their recommended locations; PSO convergence graph; pre optimized floor with adjusted structural elements
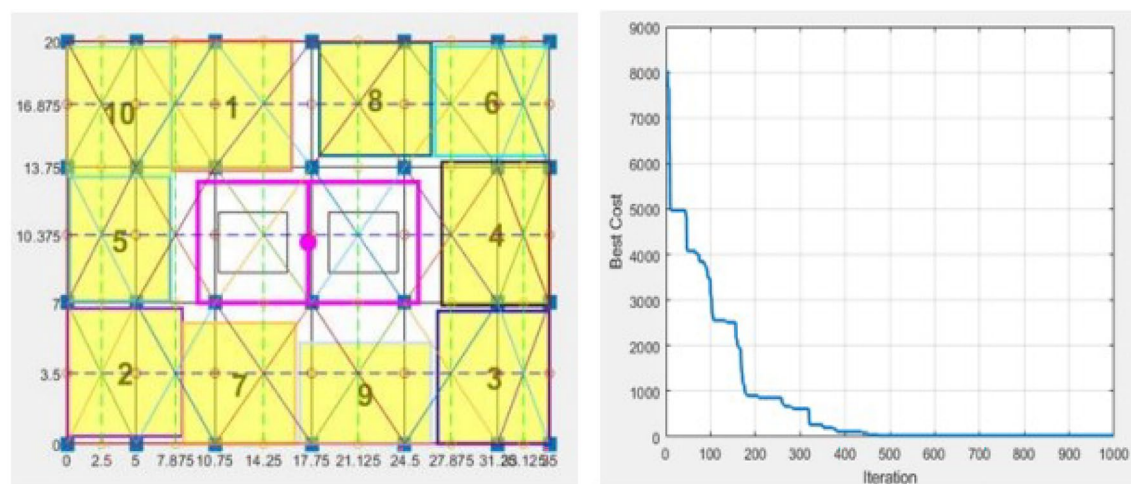
Fig. 8 A 10 household floor with a domain size of 700 m^2: floor layout including beams and columns with their recommended locations; PSO convergence graph

Table 3 Optimal values of ground structures for case studies considered	No. of households	Area of domain (m^2)	Area of functional blocks (m^2)	Area of restricted blocks (m^2)	R_{cb} of first orthogonal axes
	4	289	213.5	35	0.583
	5	360	277.9	35	0.583
	10	700	502.9	96	0.555

outputs are the optimal outputs after the optimization is carried out.

Conclusion

This article presents a two-stage method for developing a topologically preoptimized ground structure to be used for cost minimization of 3D RC buildings using particle swarm optimization. It thus avoids the use of randomly framed RC structures while considering their optimization to better represent reality and has a systemic approach towards 3D RC optimality. It also enables a kind of standard and fair comparison among RC optimization approaches because they follow the same trend, the ground structure. Three case studies were considered, each consisting of four, five and 10 households with 289, 360 and 700 square metres of floor domain, respectively.

The method developed is flexible such that restricted functional blocks can be placed anywhere they are required to be. Corridors and stairs can be of different numbers, sizes, locations and combinations. Functional blocks inside each household can also be of fixed size and can assume rotated positions for better optimality.

Through floor optimization, a selected number of functional blocks together with their detailed constraints are generated and placed, through frame layout optimization, an appropriate number of columns and beams that lead to minimum cost are generated, and the respective constraints are kept. Real-world representation and minimal cost-resulting criteria were developed and implemented to generate a simplified yet potential candidate ground structure for 3D slab-frame RC buildings.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

References

Ahmadkhanlou, F., & Adeli, H. (2005). Optimum cost design of reinforced concrete slabs using neural dynamics model. *Engineering Applications of Artificial Intelligence, 18*(1), 65–72.

Alkam, F., & Lahmer, T. (2019). Quantifying the uncertainty of identified parameters of prestressed concrete poles using the experimental measurements and different optimization methods. *Applied Sciences, 4*(4), 84–92.

Babiker, S., Adam, F., & Mohamed, A. E. (2012). Design optimization of reinforced concrete beams using artificial neural network. *International Journal of Engineering Inventions, 1*(8), 07–13.

Begg, W., & Liu, X. (2000). On simultaneous optimization of smart structures-part ii: Algorithms and examples. *Computer Methods in Applied Mechanics and Engineering, 184*(1), 25–37.

Carvalho, J. P. G., Lemonge, A. C. C., Hallak, P. H., & Vargas, D. E. C. (2020). Simultaneous sizing, shape, and layout optimization and automatic member grouping of dome structures. *Structures, 28*, 2188–2202. Elsevier.

Changizi, N., & Jalalpour, M. (2018). Topology optimization of steel frame structures with constraints on overall and individual member instabilities. *Finite Elements in Analysis and Design, 141*, 119–134.

Chen, T.-C., & Chang, Y.-W. (2006). Modern floor planning based on b/sup*/-tree and fast simulated annealing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 25*(4), 637–650.

Cheng, G., & Jiang, Z. (1992). Study on topology optimization with stress constraints. *Engineering Optimization, 20*(2), 129–148.

Dorn, W. (1964). Automatic design of optimal structures. *J. de Mecanique, 3*, 25–52.

Elbes, M., Alzubi, S., Kanan, T., Al-Fuqaha, A., & Hawashin, B. (2019). A survey on particle swarm optimization with emphasis on engineering and network applications. *Evolutionary Intelligence, 12*(2), 113–129.

Ernst, & Neufert, E. (2000). Architects' data. (3rd ed.) School of Architecture, Oxford Brookes University. Blackwell Science

Fan, S.-K.S., Chang, J.-M., & Chuang, Y.-C. (2015). A new multi-objective particle swarm optimizer using empirical movement and diversified search strategies. *Engineering Optimization, 47*(6), 750–770.

Fan, S.-K.S., & Chiu, Y.-Y. (2007). A decreasing inertia weight particle swarm optimizer. *Engineering Optimization, 39*(2), 203–228.

Ghoddosian, A., Vezvari, M. R., Azqandi, M. S., & Karimi, M. A. (2018). Topology optimisation of the discrete structures with the minimum growing ground structure method. *International Journal of Structural Engineering, 9*(1), 38–49.

Guo, Z., & Li, B. (2017). Evolutionary approach for spatial architecture layout design enhanced by an agent-based topology finding system. *Frontiers of Architectural Research, 6*(1), 53–62.

Hagishita, T., & Ohsaki, M. (2009). Topology optimization of trusses by growing ground structure method. *Structural and Multidisciplinary Optimization, 37*(4), 377–393.

Han, Y., & Lu, W. F. (2018). A novel design method for nonuniform lattice structures based on topology optimization. *Journal of Mechanical Design, 140*(9), 091403.

Harirchian, E., Lahmer, T., & Rasulzade, S. (2020). Earthquake hazard safety assessment of existing buildings using optimized multi-layer perceptron neural network. *Energies, 13*(8), 2060.

Hsu, M.-H., & Hsu, Y.-L. (2005). Generalization of two-and three-dimensional structural topology optimization. *Engineering Optimization, 37*(1), 83–102.

Jaouadi, Z., Abbas, T., Morgenthal, G., & Lahmer, T. (2020). Single and multi-objective shape optimization of streamlined bridge decks. *Structural and Multidisciplinary Optimization, 61*(4), 1495–1514.

Kaveh, A. (2017). Cost and co2 emission optimization of reinforced concrete frames using enhanced colliding bodies optimization algorithm. In *Applications of metaheuristic optimization algorithms in civil engineering* (pp. 319–350). Springer.

Kaveh, A., & Abadi, A. S. M. (2011). Cost optimization of reinforced concrete one-way ribbed slabs using harmony search algorithm. *Arabian Journal for Science and Engineering, 36*(7), 1179–1187.

Kaveh, A., & Behnam, A. F. (2013). Design optimization of reinforced concrete 3d structures considering frequency constraints via a charged system search. *Scientia Iranica, 20*(3), 387–396.

Kaveh, A., Izadifard, R. A., & Mottaghi, L. (2020). Optimal design of planar RC frames considering CO2 emissions using ECBO, EVPS and PSO metaheuristic algorithms. *Journal of Building Engineering, 28*, 101014.

Kaveh, A., Mottaghi, L., & Izadifard, R. A. (2021). An integrated method for sustainable performance-based optimal seismic design of rc frames with non-prismatic beams. *Scientia Iranica, 28*(5), 2596–2612.

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

Kaveh, A., & Zakian, P. (2014). Seismic design optimisation of RC moment frames and dual shear wall-frame structures via CSS algorithm. *Asian Journal of Civil Engineering, 15*, 435–465.

Larsen, S. D., Sigmund, O., & Groen, J. P. (2018). Optimal truss and frame design from projected homogenization-based topology optimization. *Structural and Multidisciplinary Optimization, 57*(4), 1461–1474.

Lepš, M., & Šejnoha, M. (2003). New approach to optimization of reinforced concrete beams. *Computers & Structures, 81*(18–19), 1957–1966.

Li, Z., Luo, Z., Zhang, L.-C., & Wang, C.-H. (2021). Topological design of pentamode lattice metamaterials using a ground structure method. *Materials & Design, 202*, 109523.

Liu, F., Yang, H., & Gardner, L. (2016). Post-fire behaviour of eccentrically loaded reinforced concrete columns confined by circular steel tubes. *Journal of Constructional Steel Research, 122*, 495–510.

Merrell, P., Schkufza, E., & Koltun, V. (2010). Computer-generated residential building layouts. In *ACM SIGGRAPH Asia 2010 papers* (pp. 1–12).

Michalek, J., Choudhary, R., & Papalambros, P. (2002). Architectural layout design optimization. *Engineering Optimization, 34*(5), 461–484.

Miguel, L. F. F., Lopez, R. H., & Miguel, L. F. F. (2013). Multimodal size, shape, and topology optimisation of truss structures using the firefly algorithm. *Advances in Engineering Software, 56*, 23–37.

Nimtawat, A., & Nanakorn, P. (2010). A genetic algorithm for beam-slab layout design of rectilinear floors. *Engineering Structures, 32*(11), 3488–3500.

Ohsaki, M. (2016). *Optimization of finite dimensional structures*. CRC Press.

Perez, R. E., & Behdinan, K. (2007). Particle swarm optimization in structural design. In *Swarm intelligence: Focus on ant and particle swarm optimization* (Vol. 532). InTech.

Pritchard, T. J., Gilbert, M. & Tyas, A. (2005). Plastic layout optimization of large-scale frameworks subject to multiple load cases, member self-weight and with joint length penalties. In *6th World congress of structural and multidisciplinary optimization*.

Ranalli, F., Flager, F., & Fischer, M. (2018). A ground structure method to minimize the total installed cost of steel frame structures. *International Journal of Civil and Environmental Engineering, 12*(2), 160–168.

Sanders, E. D., Ramos, A. S., Jr., & Paulino, G. H. (2017). A maximum filter for the ground structure method: An optimization tool to harness multiple structural designs. *Engineering Structures, 151*, 235–252.

Sharafi, P., Hadi, M. N. S., & Teh, L. H. (2012). Heuristic approach for optimum cost and layout design of 3d reinforced concrete frames. *Journal of Structural Engineering, 138*(7), 853–863.

Sokół, T. (2011). Topology optimization of large-scale trusses using ground structure approach with selective subsets of active bars. In *19th International conference on computer methods in mechanics*, Warsaw, Poland.

von Butler, N. (2019). *Scalarization methods for multi-objective structural optimization*. Weimar University Library, Germany

Wenming, W., Fan, L., Liu, L., & Wonka, P. (2018). Miqp-based layout design for building interiors. *Computer Graphics Forum, 37*, 511–521. Wiley Online Library.

Zavala, G. R., Nebro, A. J., Luna, F., & Coello Coello, C. A. (2014). A survey of multi-objective metaheuristics applied to structural optimization. *Structural and Multidisciplinary Optimization, 49*(4), 537–558.

Zegard, T., & Paulino, G. H. (2014). Grand-ground structure based topology optimization for arbitrary 2d domains using matlab. *Structural and Multidisciplinary Optimization, 50*(5), 861–882.

Zhang, X., Maheshwari, S., Ramos, A. S., Jr., & Paulino, G. H. (2016). Macroelement and macropatch approaches to structural topology optimization using the ground structure method. *Journal of Structural Engineering, 142*(11), 04016090.

Zhu, Y., Yang, H., Gardner, L., & Wan, J. (2022). Performance of reinforced concrete-filled steel tubular (rcfst) members subjected to transverse impact loading. *Journal of Constructional Steel Research, 188*, 107018.