

## OPTIMIZATION OF GEOMETRICALLY NONLINEAR LATTICE GIRDERS. PART I: CONSIDERING MEMBER STRENGTHS

Tugrul TALASLIOGLU

*Department of Civil Engineering, Osmaniye Korkut Ata University, 80000 Osmaniye, Turkey*

Received 11 Jun 2012; accepted 13 Nov 2012

**Abstract.** In this study, the entire weight, joint displacements and load-carrying capacity of tubular lattice girders are simultaneously optimized by a multi-objective optimization algorithm, named Non-dominated Sorting Genetic Algorithm II (NSGAI). Thus, the structural responses of tubular lattice girders are obtained by use of arc-length method as a geometrically nonlinear analysis approach and utilized to check their member strengths at each load step according to the provisions of the American Petroleum Institute specification (API RP2A-LRFD 1993). In order to improve the computing capacity of proposed optimization approach, while the optimization algorithm is hybridized with a radial basis neural network approach, an automatic lattice girder generator is included into the design stage. The improved optimization algorithm, called ImpNSGAI, is applied to both a benchmark lattice girder with 17 members and a lattice girder with varying span lengths and loading conditions. Consequently, it is demonstrated: 1) the optimal lattice girder configuration generated has a higher load-carrying capacity ensuring lower weight and joint displacement values; 2) the use of a multi-objective optimization approach increases the correctness degree in evaluation of optimality quality due to the possibility of performing a trade-off analysis for optimal designations; 3) the computing performance of ImpNSGAI is higher than NSGAI's.

**Keywords:** multi-objective optimization, lattice girder, geometrical nonlinearity, API RP2A-LRFD.

### Introduction

The aesthetic appearance, lower constructing cost and higher load-carrying capacity of steel tubular structures makes them the most popular one of the structural engineering applications (Wardenier *et al.* 1998; Wardenier 2001; Kurobane *et al.* 2005). Particularly, the lattice girders, which are used to construct the roof structures, special industrial applications (cranes) and bridges, are well-known due to spanning long distances without any supports (Davison, Owens 2005; Nelson, McCormac 2003). As in the design of the other steel structures, a well-known traditional approach to the design of lattice girders is to utilize a designer's personal experience. This trial-error approach is based on checking the structural responses of lattice girder obtained by a linear structural analysis method for its certain configuration in accordance with the provisions of a steel specification. However, whereas the determination of the best one among potential designations is required a longer computing time, the change in the shape of lattice girder have to be kept within small limits due to the computing procedure used by the mathematical model of linear structural analysis approach.

In order to overcome these bottlenecks mentioned above, a nonlinear structural analysis method should be

integrated with an optimization approach (Kamat *et al.* 1984; Hrinda, Nguyen 2008; Suleman, Sedaghati 2005). In this regard, the preliminary studies regarding to use of geometrically nonlinearity are based on utilization of strain energy densities of steel structure members or design sensitivity information for design optimization (Kamat, Raungasilasingha 1985; Wu, Arora 1988; Santos, Choi 1988). Although the recent optimization approaches have managed to integrate the nonlinear structural analysis method with the provisions of a design specification (Saka 2007; Çarbaş, Saka 2012; Kaveh, Talatahari 2011), the computational expense is higher since any designation that has a negative determinant of global rigidity matrix has to be discarded (Carbas, Saka 2011).

Since the mathematical model utilized to define the behavior of geometrically nonlinear lattice girder is based on computation of large joint displacements, the objective functions, entire weight, joint displacements and member forces are conflicted with each other. Therefore, the determination of a proper objective function for the design optimization of geometrically nonlinear lattice girder becomes an important task.

In this study, an optimization algorithm with multiple objectives, named NSGAI is employed to generate optimal designations. The member strengths of lattice

girders computed by use of arc-length method is checked according to design constraints based on the provisions of API RP2A-LRFD specification (see further details in References of American Petroleum Institute (API): API RP2A-LRFD (1993) and American Petroleum Institute (API): API RP2A-WSD (2000)).

In order to increase the computing efficiency of NSGAI, a radial basis neural network approach is implemented into computing procedure of NSGAI. Furthermore, the flexibility – of this improved NSGAI (ImpNSGAI) is enhanced by inclusion of an automatic lattice girder generator into the design stage of proposed optimal design approach. The computing procedure of proposed optimization approach is accordingly coded in MATLAB.

This study begins by describing firstly multi-objective optimization problem including a brief introduction to the recent multi-objective optimization approaches in Section 1. The objective functions and design constraints are introduced along with a verification example devised for the application of nonlinear structural analysis method. Then, the computing steps of proposed optimization approach are explained by presenting the toolbox names used for NSGAI's algorithm in Section 2. Moreover, further details about coding the design problem in MATLAB environment are summarized in Section 3. Whereas the computing performance of ImpNSGAI is evaluated in the section named "Discussion of results". The last section is reserved to present the concluding remarks.

## 1. Description of a multi-objective optimization problem and brief introduction of recent multi-objective optimization approaches

The basic elements of a general multi-objective optimization problem are:  $m$  objective functions,  $J$  constraints and  $N$  design (decision) variables. It is formulated as follows:

$$\min/\max F(x) = \{(f_1(x)), \dots, (f_m(x))\}, x \in DS; \quad (1)$$

$$DS = \{x_n^L \leq x_n \leq x_n^U, n = 1, 2, \dots, N\}; \quad (2)$$

$$SS = \{\forall x \in DS : F(x), \text{ only if, } g_j(x) \leq 0, j = 1, \dots, J\}. \quad (3)$$

A decision variable set defined in a design variable space (DS) is indicated by  $X$ , upper and lower bounds of which are  $x_n^U$  and  $x_n^L$ . It is also utilized to compute both objective functions  $f$  and constrains  $g_j(x)$  in a solution space (SS).

At each run of a multi-objective optimization algorithm, a random solutions set is obtained. Some of them are non-dominated solutions (none is better for all objectives) and referred as "Pareto solution" defined in a concept named as domination (Srinivas, Deb 1995). Thus, the Pareto solutions are used to form "Pareto front" which determines bounds of non-dominated solutions.

It is well known that a number of multi-objective optimization approaches has been utilized in various

areas of engineering fields. This study does not propose to present and survey the multi-objective optimization approaches. Recent popular studies based on mimicking the natural phonemes are briefly introduced. These are either generally nature (biologic and physique phenomena) inspired approaches or their hybridizing variants. Preliminary one of the biologic-based approaches are evolutionary algorithms. For example, non-dominated sorting genetic algorithms (NSGAI and NSGAI) by Srinivas and Deb (1995) and Deb *et al.* (2002), strength Pareto evolutionary algorithm (SPEA and SPEAI) by Zitzler and Thiele (1999), Zitzler *et al.* (2001), modified NSGAI by Ramesh *et al.* (2011), sub-population genetic algorithm II (SPGAI) by Chang and Chen (2009). While some of the biologic-based approaches mimics directly an artificial immune system (an evolutionary immune approach by Tan *et al.* (2008)), the immune system is also utilized to hybridize with a genetic algorithm approach (Park *et al.* 2009). The swarm intelligence techniques are also one of the recent popular optimization approaches. For example, artificial bee colony by Omkar *et al.* (2011) and artificial ant colony by Yagmahan (2011). They are not solely used (a pure particle swarm approach by Mahmoodabadi *et al.* (2011), a multi-swarm cooperative particle warm approach by Zhang *et al.* (2011)); but also, hybridized with the physique based approaches (charge system search by Kaveh and Laknejadi (2011), quantum based paradigm by Omkar *et al.* (2009)).

## 2. The elements of proposed optimal design approach with multiple objectives: objective functions, design constraints and nonlinear structural analysis approach

In this study, the design of lattice girders is optimized according to the design constraints based on provisions of API RP2A-LRFD specification (1993). Hence, whereas the entire weight of lattice girder  $f_1$  and joint displacements  $f_2$  are minimized, its member forces or stresses  $f_3$  are maximized (Eqns (4)–(6)). The corresponding objective functions are formulated as:

$$f_1 = \min\left(\sum_{k=1}^m (w * l)_k\right) \quad (k = 1, \dots, m); \quad (4)$$

$$f_2 = \min(d_{ij}) \quad (i = 1, \dots, 6 \text{ and } j = 1, \dots, n); \quad (5)$$

$$f_3 = \max(f_{ij}) \text{ or } \max(sc_k) \quad (k = 1, \dots, 4), \quad (6)$$

where the term  $W$  is computed using length of lattice girder member  $l$  and unit weight  $w$ . The corresponding unit weight values are selected from a steel profile list that contains 37 ready cross-sections with circular hollow. While  $d_{ij}$  is termed as a joint displacement corresponding to the related degree of freedom  $i$  and joint  $j$ , the terms  $n$  and  $m$  indicate total joint and member numbers of lattice girder. The member forces  $f_{ij}$  are represented by member axial forces, shear forces, bending and torsional moments computed for the member ends of lattice

girder. The corresponding stresses  $s_{ij}$  computed by use of  $f_{ij}$  are also utilized for evaluation of load-carrying capacity of tubular lattice girders. Furthermore, the design complexity is increased thereby combining the member stress  $s_{ij}$  to obtain their related stress combinations ( $sc$ ). Thus, total four extreme stress combinations are utilized in the design optimization: 1) (axial – bending stresses (Z)) ( $sc_1$ ); 2) (axial – bending stresses (Y)) ( $sc_2$ ); 3) (axial + bending stresses (Z)) ( $sc_3$ ); 4) (axial + bending stresses (Y)) ( $sc_4$ ).

In order to avoid a detailed description of design constraints, only names of the design constrains are presented using both same formulation number and same notation & index as in the provisions of API RP2A-LRFD specification (1993). Design constraints are represented by member-strength-related inequalities. In this study, these inequalities are transformed into the unities (see these inequalities in section D, named Cylindrical Member Design in API RP2A-LRFD specification (1993)). For example, the presentation of  $(f_t \leq \phi_t * F_y)$  for axial tension and  $(f_b \leq \phi_b * F_{bn})$  for bending is represented by  $Unity_{Axial} = \frac{f_t}{\phi_t * F_y}$  and  $Unity_{Bending} = \frac{f_b}{\phi_b * F_{bn}}$ . In this regard, member-strength-related unities:  $Unity_{Axial}^k$  (see D.2.1-1 and D.2.2-1),  $Unity_{Bending}^k$  (see D.2.3-1),  $Unity_{CombinedBending}^k$  (see D.2.3-1),  $Unity_{Shear}^k$  (see D.2.4-1),  $Unity_{AxialCompr\&BendingBuck}^k$  (see D.3.2-1),  $Unity_{Torsion}^k$  (see D.2.4-3),  $Unity_{AxialCompr\&BendingYield}^k$  (see D.3.1-1) are utilized for each member of lattice girder ( $k = 1, \dots, m$ ).

If one of these member-strength-related unities exceeds a fixed value “1” at any load step, then nonlinear structural analysis is terminated. The corresponding values of member forces to this load step is used to determine the load-carrying capacity of current lattice girder since the computing procedure of nonlinear structural analysis method is performed out by iteratively tracing an equilibrium path that contains the incremental values of load and displacements (see the details of computing procedure in the following Section 2.1).

**2.1. Computing procedure of nonlinear structural analysis approach, named arc-length method and its verification**

Ready software named ANSYS (2012) is employed to perform the computing procedures of arc-length method (see the governing equation in Eqn (15) and Chapter 15.3.6. arc-length method in ANSYS (2012) help for a further consideration):

$$[K_i^T] \{ \Delta u_i \} - \Delta \lambda \{ F^a \} = (\lambda_n + \lambda_i) \{ F^a \} - \{ F_i^{nr} \} = - \{ R_i \} \quad (7)$$

In order to validate the results obtained by ANSYS, a frame system which was used by Simo (1986) (Fig. 1a) is utilized. While BEAM3-element from the ANSYS library of elements is used to represent the member of frame system, a basic command list governed arc-length method are written in a file with an extension named “mac”. The important parameter values of these commands are taken as:  $F$  for convergence label, 0.0001 for convergence tolerance (see command CNVTOL), 400 for load step (see command NSUBST), 40 for maximum arc-length multiplier and 0.004 for minimum arc-length

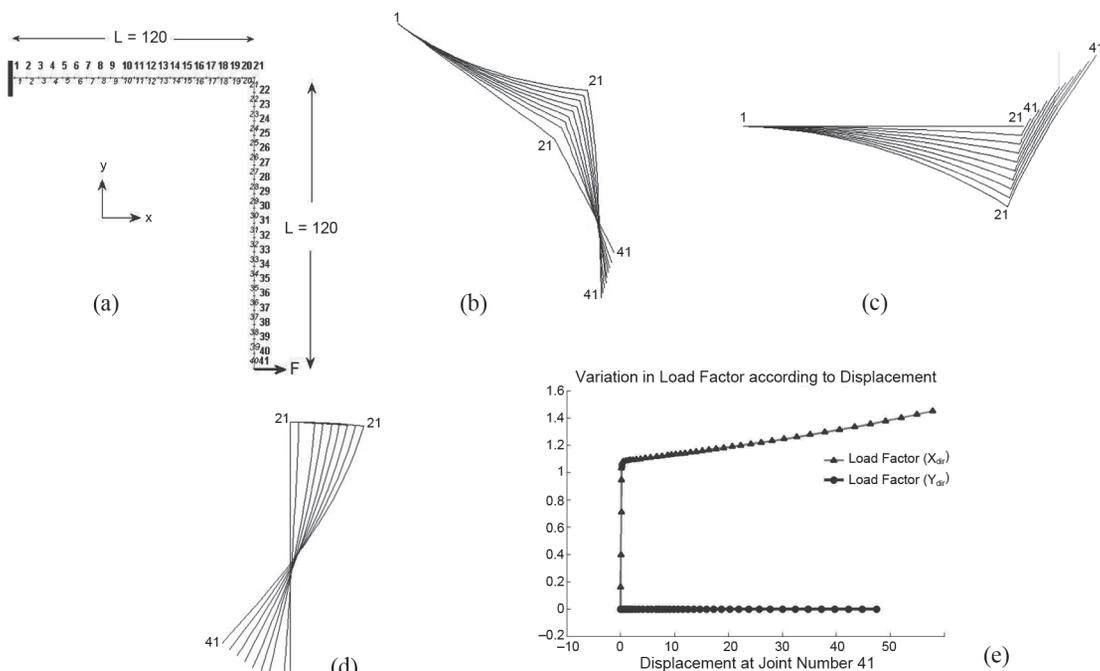


Fig. 1. A verification frame (a), its deformed shapes (c–d) and load factor-displacement graph (e)

multiplier (see command ARCLen). Considering the post-buckling graph (Fig. 1e) it is observed that the load factor, displacement value set (1.09 and –) corresponding to the limit point obtained Simo (1986) is a good agreement with the set (1.08819, 1.40875) from ANSYS (see deformations of this frame in Figs 1b–1d). It is noted that the member forces of this frame indicated the response to the external static load are utilized to compute the load factor. In this regard, the member force that is a main determinative factor for the load-carrying capacity of lattice girder is employed as an objective function in this study. Thus, the maximum values of member forces or stresses are utilized to assess the load-carrying capacity of lattice girders.

### 3. Optimum design of tubular lattice girders

In this study, NSGAI, computing procedure of which is coded in MATLAB, is improved by a simple but efficient implementation of a neural network. A pseudo code defined the computing steps of improved NSGAI (ImpNSGAI) are presented in Figure 2 in conjunction with its toolbox names, each of which contains related code scripts. Before the pseudo code is described, some details about the design of tubular lattice girder and implementation of neural network are given in sub-sections.

#### 3.1. Design of tubular lattice girder

In order to enhance the flexibility of NSGAI for the generation of optimal designations, an automatic lattice girder generator is included into design stage. Thus, it is possible to automatically generate the lattice girder with various framing configurations in order to increase its load-carrying capacity. A conceptual lattice girder model is presented in Figure 3.

In this study, this conceptual lattice girder model is generated using size, topology and shape related design variables. In this regard, the cross-sectional properties, which are assigned to girder members from a steel profile database with 37 ready cross-sections with circular hollow, are used to represent the size-related design variables ( $D_1$ ,  $D_2$ ,  $D_3$  and  $D_4$ , used to represent the top chord, bottom chord, vertical and brace members, respectively). Hence, the upper and lower limits of size-related design variables  $Par_{UDV}$  and  $Par_{LDV}$  are 37 and 1. The division number  $Par_{DN}$  used to divide the length of span into small ones is utilized to represent topology-related design variables. Their upper and lower limits are represented by the parameters  $Par_{DNU}$  and  $Par_{DNL}$  respectively. In order to obtain an appropriate geometrical configuration of the lattice girder, its two heights,  $H_1$  (for first longitudinal member) and  $H_2$  (for middle longitudinal member) are symmetrically adjusted. Hence, the shape-related design variables  $Par_{H1}$  and  $Par_{H2}$  are determined according to two intervals, upper and lower limits of which are represented by the parameters  $Par_{H1U}$ ,  $Par_{H1L}$ ,  $Par_{H2U}$  and  $Par_{H2L}$ , respectively.

The conceptual lattice girder model generated by use of size, shape and topology related design variables mentioned above is utilized in automatically preparation of an input file for ANSYS. Following the execution of the input file in batch mode, the response computation and constraint evaluations are automatically carried out thereby interacting with ANSYS (see Section 2.1 for some basic commands and the help of ANSYS for a further investigation). This full integrated programming unit in turn provides the values of objective functions (Eqns (4)–(6)) for optimization-related computations. It is noted that checking process is terminated once any of design constraints based on the provisions of API RP2A-LRFD (1993) specification is violated at any incremental stage of nonlinear structural analysis (see ROUTINE 1 in Fig. 2 for computing procedure of nonlinear structural analysis). Therefore, the proposed design optimization algorithm with multi-objective called ImpNSGAI does not requires any penalization process for inclusion of unfeasible designations into the solution space.

#### 3.2. Implementation of neural network

In the design process of a neural network, firstly an input and output data must be defined. Then, a neural network is created to be configured and trained. A building block for neural network may contain a number of neuron. In the training process, related network parameters are correspondingly adjusted considering the input and output data (see help of MATLAB). In this study, a radial basis neural network is utilized for its network architecture that is consisted of two layers, a hidden radial basis layer and output linear layer of neurons. Although a general radial basis neural network is defined depending on the previously assigned input and output data, in this study, the input data (see *ALLinput* in Fig. 2) is only pre-assigned. The output data (see *ALLoutput* in Fig. 2) is obtained in the end of each execution (see “Neural Network Implementation Number ( $Par_{NNIN}$ ) in Fig. 2). The matrix *ALLoutput* contains “Spread” and “Average Distance” values. Thus, the network architecture (see command “Newrb” in help of MATLAB) is easily created. Then, a new input data is obtained in the end of an adaptive and training based simulation process (see command “Sim” in help of MATLAB) depending on *ALLoutput*. The new input data is obtained thereby improving the spread and average distance values in associated with  $Par_{NNIN}$ .

#### 3.3. Description of computing steps of ImpNSGAI

The execution of ImpNSGAI governed by the parameter ( $Par_{NNIN}$ )” is carried out two levels: whereas optimization-related parameters (Generation Number ( $Par_{Gen}$ ), Population Size ( $Par_{PopS}$ ) Mutation Rate ( $Par_{MutR}$ ), Crossover Rate ( $Par_{CrosR}$ ), Crossover Fraction ( $Par_{CrosFrac}$ ), Migration Interval ( $Par_{MigIn}$ ), Migration Fraction ( $Par_{MigFrac}$ ), Selection Rate ( $Par_{SelR}$ ) are adjusted in first level thereby utilizing the radial basis neural network implementation,

```

WhichMethod=1 % WhichMethod=1 for NSGAI , WhichMethod=2 for ImpNSGAI
FitnessFunction = @ObjLatticeGirder;
Case=1
    if case==1
        Par_DNU=15, Par_DNL=5, Par_LoS=393.700, Par_HIU=35.433, Par_HIL=23.622, Par_H2U=23.622 and Par_H2L=3.937, Par_UDV=1 and Par_LDV=37
    elseif case==2
        Par_DNU=20, Par_DNL=5, Par_LoS=393.700, Par_HIU=47.244, Par_HIL=23.622, Par_H2U=23.622 and Par_H2L=3.937, Par_UDV=1 and Par_LDV=37
    elseif case==3
        Par_DNU=40, Par_DNL=10, Par_LoS=787.401, Par_HIU=59.055, Par_HIL=31.496, Par_H2U=31.496 and Par_H2L=7.874, Par_UDV=1 and Par_LDV=37
    end
ub = [Par_DNU Par_LS Par_HIU Par_H2U Par_UDV Par_UDV Par_UDV Par_UDV ]; % Lower bound
lb = [Par_DNL Par_LS Par_HIL Par_H2L Par_LDV Par_LDV Par_LDV Par_LDV ]; % Upper bound
Par_NNIN=10; Par_Gen=25, Par_Pop=50, Par_Mut=0.5, Par_CrosR=0.5, Par_CrosFrac=0.5, Par_Migh=2, Par_MigFrac=0.5, Par_Sel=2,
    for i=1: Par_NNIN
        options = gaoptimset('Generations', Par_Gen);
        options.PopulationSize= Par_Pop;
        options.MutationFcn= {@mutationuniform, Par_MutR };
        options.CrossoverFcn= {@crossoverheuristic, Par_CrosR };
        options.CrossoverFrac= Par_CrosFrac;
        options.MigrationInterval= Par_Migh;
        options.MigrationFraction= Par_MigFrac;
        options.SelectionFcn= {@selectiontournament, Par_SelR };

    gamultiobj
        % Validate constraints and options
        gacommon
        % Call appropriate multi objective optimization solver
        gamultiobjsolve
            gamultiobjMakeState % Create initial state: population, Evaluate fitness functions
            feval(options.CreationFcn) % (see ROUTINE1) Evaluate fitness function to get the number of objectives
            Fcnvectorizer % Evaluate fitness function to get the number of objectives
            rankAndDistance % Get the rank and distance measure of the population
end

ROUTINE1
1. Divide the applied loads ( Fa ) by into n load step
2. Assume { ui=0 } = 0
3. Compute tangent matrix [ KiT ] and restoring load { Finr } from configuration { ui }
4. Compute the incremental load factor ( Δλ ) depending on Δui and the proportional loading factor ( λ ) by an explicit spherical iteration method in a way of ensuring an orthogonality condition (see Eqn (7)).
5. Check the convergence degree that controls the termination condition of iteration process using the resulted vector called as residual vector ( Ri )
6. Compute the responses (Member forces, stresses and joint displacements) corresponding to the time step n and evaluate them according to provisions of API RP2A-LRFD specification, if convergence degree is satisfied.
7. Repeat steps 3 to 6 until any of design requirements API RP2A-LRFD specification is violated.
8. Save the entire weight of dome structure f1, joint displacements f2 and member forces f3 (see Eqns (4)–(6)).

ROUTINE 2
feval(options.SelectionFcn) % Selection
feval(options.MutationFcn) % Crossover
Fcnvectorizer % Evaluate fitness function
rankAndDistance % Get rank and distance
distanceAndSpread % Calc. dist., spread,

gadsplot; gaoutput % Give the plot/output Functions
while true % check to see if any stopping criteria have been met
    gamultiobjConverged
    stepgamultiobj
        ROUTINE1 and ROUTINE 2
    migrate % Migration
    gadsplot; gaoutput % Give the plot/output Functions
    if WhichMethod==2
        ALLoutput=[output.averagedistance, output.spread]
        ALLinput=[ALLinput [Par_Gen; Par_Pop; Par_MutR; Par_CrosR; Par_CrosFrac; Par_Migh; Par_MigFrac; Par_SelR]];
        ALLinput=NeuralNetworkImp(ALLinput, ALLoutput)
    end
end
end
    
```

Fig. 2. A pseudo code for NSGA II in conjunction with its toolbox names

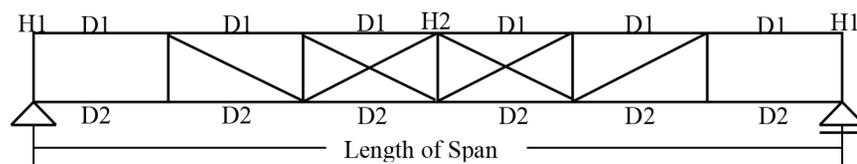


Fig. 3. A conceptual lattice girder model with various geometrical configurations

the application of these parameters to the optimization procedure of NSGAI is performed in the second level. At  $Par_{NNIN} = 1$  in the first level, the user-defined initial values are assigned to the optimization-related parameters, then these values are collected in a matrix “ALLinput” along with output values “average distance” and “spread” collected in a matrix “ALLoutput” that is resulted in execution of NSGAI algorithm. *ALLinput* obtained by the implementation of neural network by use of *ALLinput* and *ALLoutput* is re-assigned to the related parameters of NSGAI.

According to the pseudo code, firstly, fitness functions  $f_1, f_2$  and  $f_3$  (see ROUTINE1 defined by *FitnessFunction* in Fig. 2) are computed by using the first individual of the population ( $x^0$ ) and the upper-lower values of design variables ( $lb$  and  $ub$ ) (Eqns (8)–(10)). Although the maximum number of design variables (*number OfVariables*) are limited into 8 (Eqns (8)–(10)), fitness values are computed by use of some numbers located in each individuals depending on  $Par_{DN}$ . The fundamental parameter values and genetic operator names are defined by making use of a structure field name named as *option*. Then, the first toolbox named as *Gamultiobj* is executed using these parameters. In this toolbox, firstly constraints and parameters defined in *option* are checked against the violation of their pre-defined values. Then, the computational

procedure of NSGAI’s algorithm begins by an execution of the toolbox *gamultiobjsolve* which calls two toolboxes named *GamultiobjMakeState* and *stepgamultiobj*. In fact, an apart of *GamultiobjMakeState* named ROUTINE2 is used to constitute the toolbox named *stepgamultiobj*. The toolbox named *gamultiobjsolve* calls *GamultiobjMakeState* in order to create the first initial population using *options.CreationFcn*, compute fitness functions (*fcnvectorizer*) and rank them (*rankAndDistance*). Also, the execution of evolutionary genetic operators named selection, mutation and crossover is carried out in the toolbox named *GamultiobjMakeState* (indicated by ROUTINE2 in Fig. 2). NSGAI’s toolbox is equipped with rich features allowing the output to both plot and save. The desired output and plots are executed in two toolboxes named *gadsplot* and *gaoutput*. The main generation that is limited by the parameter *options.Generations* begins to run (see *gamultiobjConverged* for a further information about the other termination options); then, the toolbox named *stepgamultiobj* is employed to execute three evolutionary operators, selection, mutation and crossover which are defined by ROUTINE 1. After that, the migration process is activated; then, the desired output is both saved and plotted by *gadsplot* and *gaoutput*:

$$x^0 = \left[ \underbrace{fix(x_1^0)}_{Par_{DN}}, LS, \underbrace{x_2^0, x_3^0}_{Par_{H1} \text{ and } Par_{H2}}, \underbrace{fix(x_4^0), fix(x_5^0), fix(x_6^0), fix(x_7^0)}_{Par_{DV}} \right]; \quad (8)$$

$$ub = \left[ \underbrace{Par_{DNU}}_{Par_{DN}}, LS, \underbrace{Par_{H1U}, Par_{H2U}}_{Par_{H1} \text{ and } Par_{H2}}, \underbrace{Par_{UDV}, Par_{UDV}, Par_{UDV}, Par_{UDV}}_{Par_{DV}} \right]; \quad (9)$$

$$lb = \left[ \underbrace{Par_{DNL}}_{Par_{DN}}, LS, \underbrace{Par_{H1L}, Par_{H2L}}_{Par_{H1} \text{ and } Par_{H2}}, \underbrace{Par_{LDV}, Par_{LDV}, Par_{LDV}, Par_{LDV}}_{Par_{DV}} \right]. \quad (10)$$

#### 4. Discussion of results

In order to evaluate the computing performance of ImpNSGAI, a benchmark design example with 17 members and a lattice girder with various length and loading conditions are devised. The computational efficiency of ImpNSGAI is confirmed by comparing its optimal designs with those obtained by both NSGAI and the other approaches represented in literature.

In order to compare the computing performances of ImpNSGAI and NSGAI, two quality measuring metrics, named “spread” and “average distance” are utilized (see the further details about these metrics in Deb (2001)). Thus, after computing the average of these quality indicators obtained by independent 50 executions, consistency

of these results are checked through a statistical analysis in a certain level of confidence in order to determine about whether any difference in computing performance of ImpNSGAI and NSGAI exists. It is noted that lower values of spread and average distance indicates about both a higher approximated Pareto front form and higher homogeneous distribution among random solutions.

The computing procedures of statistical analysis are performed in MATLAB. The spread and average distance values obtained at the end of each execution are stored. Then, the average values of these spread and average distance values are checked about whether to exhibit a normal distribution thereby employing the lillie test. If those

values do not show a normal distribution, a kruskal-wallis test method (see function “kruskalwallis” in MATLAB Statistical Toolbox) is utilized to compare those average spread and average distance values. Furthermore, in order to accomplish a more explicit comparison among them, a comparison of pairs is made using a post hoc 5% hsd-test (also known as Tukey-Kramer test) (see function “multicompare” in MATLAB Statistical Toolbox). Basically, this function returns a matrix of pair wise comparison results with information about which pairs of distributions are significantly different.

**4.1. A benchmark design example: a tubular lattice girder with 17 members**

The design of this planar structure with 17 members, which has a elasticity module of 30000 ksi (206842.718 N/mm<sup>2</sup>) and yielding point 50 ksi (344.737 N/mm<sup>2</sup>) (Fig. 4), was optimized by Lee and Geem (2004), Khot and Berke (1984) and Li *et al.* (2007). It is noted that the optimal design of the design example with 17 members is obtained by checking their member strengths according to API RP2A-LRFD (1993) although optimal designs

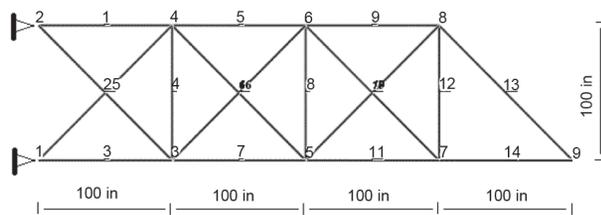


Fig. 4. Geometry of design example 1 with 17 members

presented in literature are obtained according to pre-determined stress and joint displacement values. The design variables of continuous type were assigned considering cross-sectional areas varied from a minimum value of 0.1 in<sup>2</sup>. While the joint displacement limitation is taken as 2.00 in., maximum stress values are limited to 50 ksi. The planar truss, members of which were represented by use of 17 independent design variables, was imposed by a single joint load of 100 kipf (444.822 kN) at joint 9.

The proposed design algorithms are performed to optimize the design of this lattice girder considering both joint displacement and member stress values as described in Literature. In the end of executions both ImpNSGAI and NSGAI, total 50 different spread and average distance values are obtained. Considering these quality measuring quantity value, the statistical test is performed to assess their computing performances. The corresponding statistical significance values to the quality measuring quantities are presented for both ImpNSGAI and NSGAI along with their maximum, minimum and average values (Table 1). Furthermore, the outputs obtained by MATLAB are visualized for a further examination of statistical test results (Figs 5a–5c).

According to the values of statistical significance (P < 0.05), there is a considerably difference among the computing performances of ImpNSGAI and NSGAI. It is clear that computing performance of ImpNSGAI is higher than NSGAI taking into account of lower average spread (0.0121) and average distance values (0.0767) (Table 1).

Table 1. A statistical assessment results of spread and average distances values for design example 1

		Average Distance			Statistical Significance	Spread			Statistical Significance
		Max.	Min.	Aver.		Max.	Min.	Aver.	
Truss with 17 mem.	NSGAI	0.1649	0.0058	0.0850	0.00131	0.0185	0.0014	0.0126	0.00303
	ImpNSGAI	0.1673	0.0002	0.0767		0.0189	0.0032	0.0121	

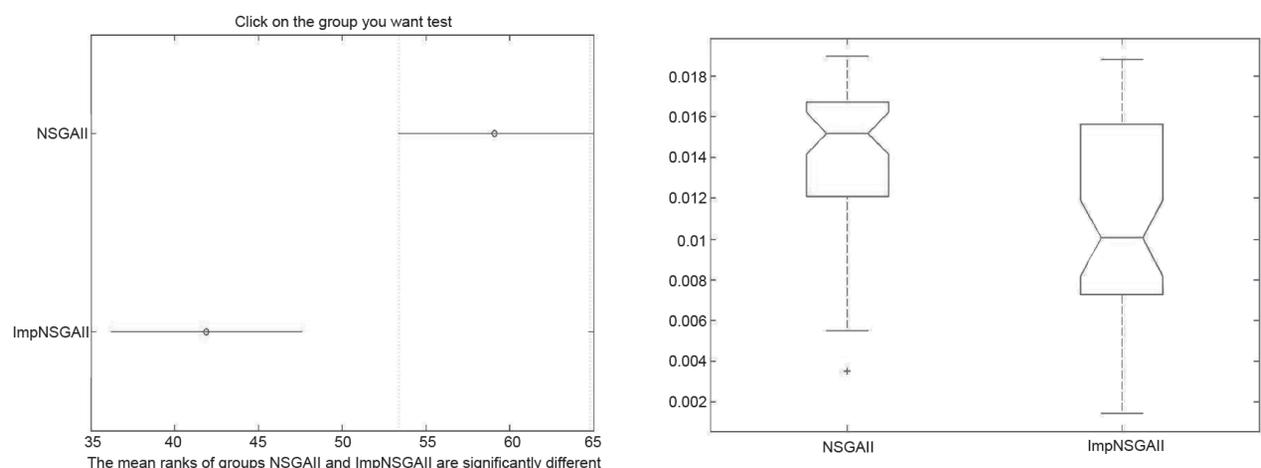


Fig. 5. MATLAB outputs (a–c) visualized for the statistical assessment results of spread (Table 1)

Table 2. Variation in optimization-related parameters of both ImpNSGAI and NSGAI corresponding to minimum spread and average distance values (Table 1)

Names of Genetic Parameters	Par <sub>NNIN</sub> (ImpNSGAI)									
	1	2	3	4	5	6	7	8	9	10
options.Generations	30	11	44	28	16	45	59	31	74	77
options.PopulationSize	38	46	8	38	9	12	16	4	10	28
options.MutationFcn {@mutationuniform.}	0.8960	0.8757	0.2225	0.6978	0.4471	0.7479	0.6727	0.3128	0.8306	0.3297
options.CrossoverFcn {@crossoverheuristic.}	0.7728	0.7510	0.6831	0.4964	0.3561	0.1266	0.3588	0.9005	0.9670	0.9128
options.CrossoverFraction	0.6670	0.4433	0.9855	0.7299	0.9047	0.8160	0.8192	0.3661	0.8856	0.3499
options.MigrationInterval	5	2	5	4	2	3	3	4	3	2
options.MigrationFraction	0.5662	0.1014	0.4209	0.2410	0.4374	0.6718	0.8514	0.5268	0.6158	0.5024
options.SelectionFcn {@selectiontournament.}	0.7800	0.4720	0.6222	0.1867	0.7613	0.8765	0.7294	0.8687	0.3587	0.7418
Par <sub>NNIN</sub> = 1 (NSGAI)										
options.Generations	100									
options.PopulationSize	50									
options.MutationFcn {@mutationuniform.}	0.5									
options.CrossoverFcn {@crossoverheuristic.}	0.5									
options.CrossoverFraction	0.5									
options.MigrationInterval	5									
options.MigrationFraction	0.5									
options.SelectionFcn {@selectiontournament.}	0.5									

The form of Pareto fronts of ImpNSGAI obtained by use optimization-related parameters corresponding to minimum spread (0.0032) and average distance (0.0002) values is also presented along with NSGAI's (Table 1 and Figs 6a–6d).

The search mechanism of ImpNSGAI is based on an adaptation of the evolutionary search by use of continuously changing evolutionary environment. This adaptation is carried out by adaptively adjustment of optimization-related parameters by a radial basis neural network. Thus, this self-adaptive nature of ImpNSGAI gives an opportunity to move the evolutionary search to promising regions of complex solution space. The variation in optimization-related parameters of ImpNSGAI corresponding to minimum spread (0.0032) and average distance (0.0002) values is presented in Table 2. A statistical data indicated the load step numbers is also tabulated in Table 3.

Taking into account Figure 6a, several extreme designations denoted by Des1, Des2, Des3, Des4 and Des5

Table 3. Statistical data indicated load step numbers corresponding to termination of nonlinear structural analysis (design example 1)

	ImpNSGAI			NSGAI		
	Max.	Min.	Aver.	Max.	Min.	Aver.
Truss with 17 Members	3	2	2	3	2	2

are listed to present the values of size-related design variables (Table 4) including their corresponding deformed configurations according to load steps performed by arc-length method (Figs 7a–7e). Some statistical values for load steps that define the load step numbers corresponding to termination of arc-length method due to the provisions of API RP2A-LRFD specification (1993) are also tabulated in Table 4. In order to verify the satisfied design constraint values, the variation in the values of each design constraints are schematized in Figures 8–11.

In order to investigate the load-carrying capacity of optimal designations, the total load value (100 kipf imposed on joint no 9) is represented by a flat surface in the Figure 6d. In this regard, considering the Figure 6d, it is obvious that the maximum member force values are higher than the total load value, 100 kipf. This observation points out that the configurations of lattice girder with 17 members have a higher load-carrying capacity. According to the values of their objective functions listed in Table 4, the proposed design approach achieves to obtain a lower weight value (1539.907 lb (698.503 kg)) compared to the other studies in literature (2580.81 lb (1170.635 kg)) by Lee and Geem (2004), 2581.89 lb (1171.148 kg) by Khot and Berke (1984), 2581.94 lb (1171.148 kg) by Li *et al.* (2007)). It is also presented that the designation with the highest load-carrying capacity indicated by maximum member force value (1240.472) is resulted with a highest weight value (9847.947 lb (4466.953 kg)). Furthermore,

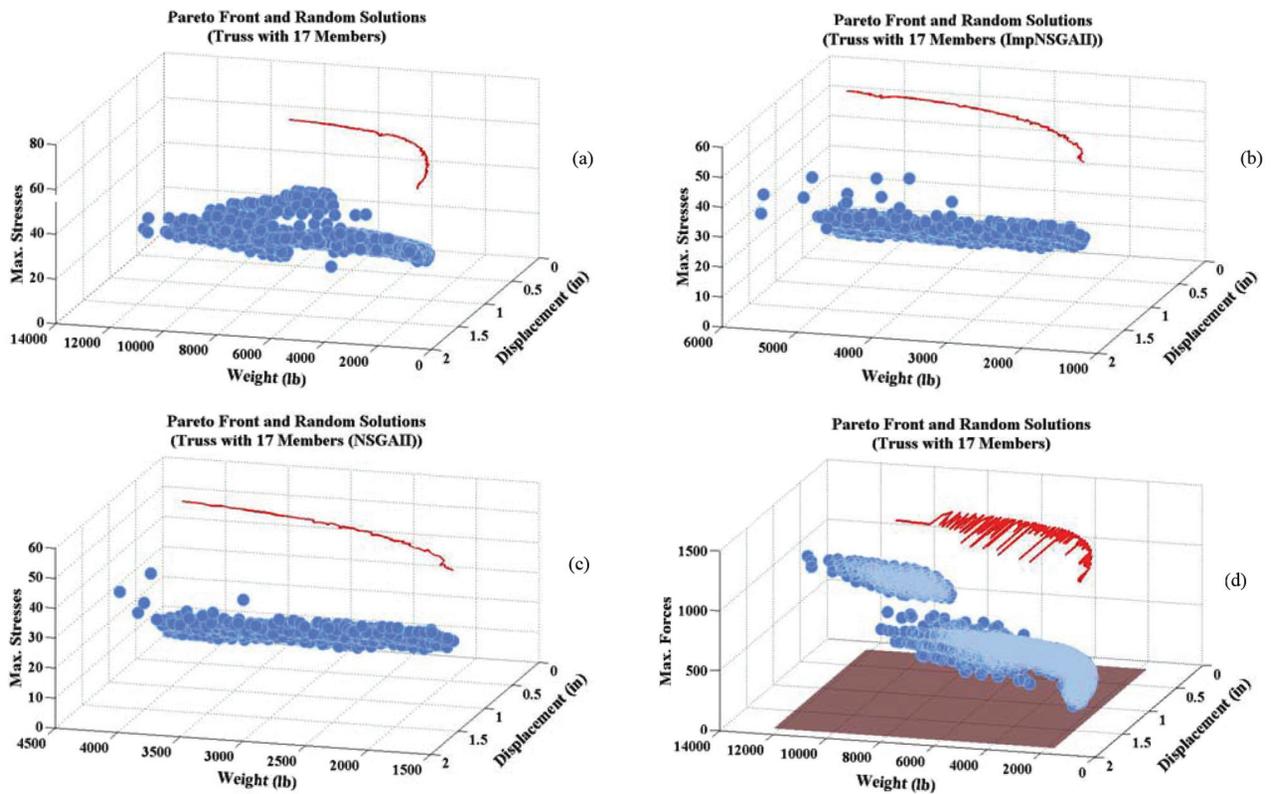


Fig. 6. True Pareto front (a), Pareto fronts and random solutions for ImpNSGAI (b) and NSGAI (c) obtained by use of stress combinations, and true Pareto front obtained by use of member forces corresponding to minimum spread and average values (Table1) (design example 1)

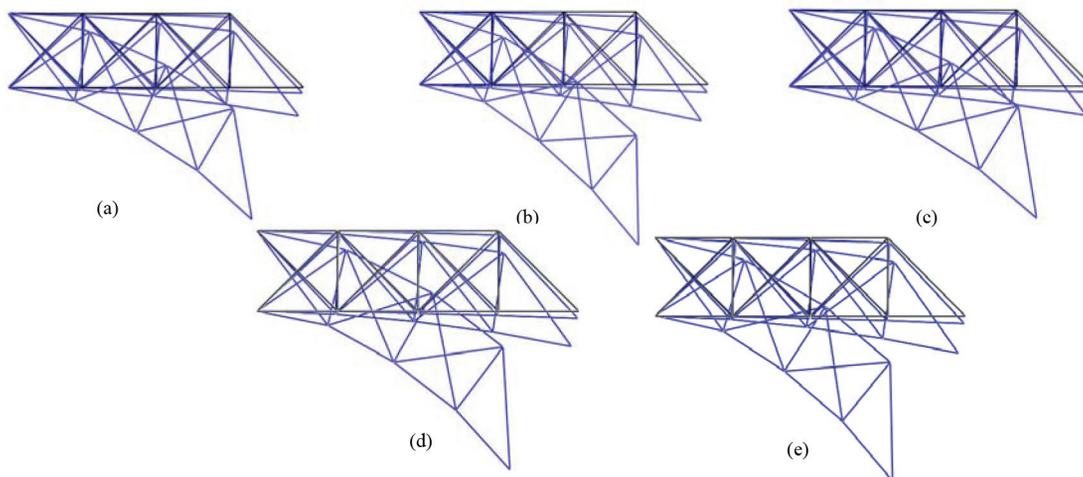


Fig. 7. Deformed shapes according to load steps obtained for Des1 (a), Des2 (b), Des3 (c), Des4 (d), Des5 (e) (Table 3)

although it is well known that a decrease in the weight of steel structure increases the joint displacement values, it is displayed that an increase in the weight values (from 1539.907 lb (698.490 kg) to 6386.020 lb (2896.649 kg) can cause to a decrease in the joint displacement values (from 1.4992 in (37.896 mm) to 1.999 in (50.774 mm)). Therefore, the use of a multi-objective optimization approach increases the correctness degree in the evaluation of optimal designation by preventing this dilemma.

It is mentioned that the joint displacements have to be constrained to an upper value due to correctly

executing the mathematical model of linear structural analysis method for the computation of structural responses. This assumption may in some degree acceptable in order to properly keep the framing geometry of lattice girder for laying the other structural elements. In this regard, it is also demonstrated that the use of an available design specification provides an increase in generation of promising optimal designations since constraining the user-defined design criteria (member stress, etc.) to its upper limits unnecessarily causes an exclusion of potential designations from the current feasible designation set.

Table 4. Some of extreme values obtained by considering all random designations (Fig. 6a)

	Size-related Design Variables																
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17
Des1 (MW)	PIPST5 <sup>a1</sup> (4.03 in <sup>2</sup> )	PIPST21/2 (1.59 in <sup>2</sup> )	PIPST5 (4.03 in <sup>2</sup> )	PIPST21/2 (1.59 in <sup>2</sup> )	PIPST31/2 (2.51 in <sup>2</sup> )	PIPST31/2 (2.51 in <sup>2</sup> )	PIPST5 (4.03 in <sup>2</sup> )	PIPST21/2 (1.59 in <sup>2</sup> )	PIPST31/2 (2.51 in <sup>2</sup> )	PIPST21/2 (1.59 in <sup>2</sup> )	PIPST5 (4.03 in <sup>2</sup> )	PIPST21/2 (1.59 in <sup>2</sup> )	PIPST31/2 (2.51 in <sup>2</sup> )	PIPST5 (4.03 in <sup>2</sup> )	PIPST31/2 (2.51 in <sup>2</sup> )	PIPST21/2 (1.59 in <sup>2</sup> )	PIPST31/2 (2.51 in <sup>2</sup> )
Des2 (MJD)	PIPEEST6 <sup>a3</sup> (14.7 in <sup>2</sup> )	PIPST6 (5.22 in <sup>2</sup> )	PIPEEST8 (20.0 in <sup>2</sup> )	PIPST6 (5.22 in <sup>2</sup> )	PIPST8 (7.85 in <sup>2</sup> )	PIPST10 (11.1 in <sup>2</sup> )	PIPEEST8 (20.0 in <sup>2</sup> )	PIPST6 (5.22 in <sup>2</sup> )	PIPST8 (7.85 in <sup>2</sup> )	PIPST6 (5.22 in <sup>2</sup> )	PIPEEST8 (20.0 in <sup>2</sup> )	PIPST6 (5.22 in <sup>2</sup> )	PIPST10 (11.1 in <sup>2</sup> )	PIPEEST8 (20.0 in <sup>2</sup> )	PIPST10 (11.1 in <sup>2</sup> )	PIPST6 (5.22 in <sup>2</sup> )	PIPST10 (11.1 in <sup>2</sup> )
Des3 (MEF)	PIPEST12 <sup>a2</sup> (17.9 in <sup>2</sup> )	PIPEST10 (15.0 in <sup>2</sup> )	PIPEST10 (15.0 in <sup>2</sup> )	PIPEST10 (15.0 in <sup>2</sup> )	PIPEST12 (17.9 in <sup>2</sup> )	PIPEST12 (17.9 in <sup>2</sup> )	PIPEST10 (15.0 in <sup>2</sup> )	PIPEST10 (15.0 in <sup>2</sup> )	PIPEST12 (17.9 in <sup>2</sup> )	PIPEST10 (15.0 in <sup>2</sup> )	PIPEST10 (15.0 in <sup>2</sup> )	PIPEST10 (15.0 in <sup>2</sup> )	PIPEST12 (17.9 in <sup>2</sup> )	PIPEST10 (15.0 in <sup>2</sup> )	PIPEST12 (17.9 in <sup>2</sup> )	PIPEST10 (15.0 in <sup>2</sup> )	PIPEST12 (17.9 in <sup>2</sup> )
Des4 (MS1)	PIPEEST8 (20.0 in <sup>2</sup> )	PIPST6 (5.22 in <sup>2</sup> )	PIPEEST8 (20.0 in <sup>2</sup> )	PIPST6 (5.22 in <sup>2</sup> )	PIPST8 (7.85 in <sup>2</sup> )	PIPST10 (11.1 in <sup>2</sup> )	PIPEEST8 (20.0 in <sup>2</sup> )	PIPST6 (5.22 in <sup>2</sup> )	PIPST8 (7.85 in <sup>2</sup> )	PIPST6 (5.22 in <sup>2</sup> )	PIPEEST8 (20.0 in <sup>2</sup> )	PIPST6 (5.22 in <sup>2</sup> )	PIPST10 (11.1 in <sup>2</sup> )	PIPEEST8 (20.0 in <sup>2</sup> )	PIPST10 (11.1 in <sup>2</sup> )	PIPST6 (5.22 in <sup>2</sup> )	PIPST10 (11.1 in <sup>2</sup> )
Des5 (MS2)	PIPEEST6 (14.7 in <sup>2</sup> )	PIPST10 (11.1 in <sup>2</sup> )	PIPEEST8 (20.0 in <sup>2</sup> )	PIPST10 (11.1 in <sup>2</sup> )	PIPST8 (7.85 in <sup>2</sup> )	PIPST10 (11.1 in <sup>2</sup> )	PIPEEST8 (20.0 in <sup>2</sup> )	PIPST10 (11.1 in <sup>2</sup> )	PIPST8 (7.85 in <sup>2</sup> )	PIPST10 (11.1 in <sup>2</sup> )	PIPEEST8 (20.0 in <sup>2</sup> )	PIPST10 (11.1 in <sup>2</sup> )	PIPST10 (11.1 in <sup>2</sup> )	PIPEEST8 (20.0 in <sup>2</sup> )	PIPST10 (11.1 in <sup>2</sup> )	PIPST10 (11.1 in <sup>2</sup> )	PIPST10 (11.1 in <sup>2</sup> )
Lee and Geem (2004) (in <sup>2</sup> )	15.821	0.108	11.996	0.100	8.150	5.507	11.829	0.100	7.934	0.100	4.093	0.100	5.660	4.061	5.656	0.100	5.582
Khot and Berke (1984) (in <sup>2</sup> )	15.930	0.100	12.070	0.100	8.067	5.562	11.993	0.100	7.945	0.100	4.055	0.100	5.657	4.000	5.558	0.100	5.579
Li <i>et al.</i> (2007) (in <sup>2</sup> )	15.896	0.103	12.092	0.100	8.063	5.591	11.915	0.100	7.965	0.100	4.076	0.100	5.670	3.998	5.548	0.103	5.537
	Entire Weight		Joint Displacement		Maximum Member Force and Stress Com.		Load Step Number										
Des1 (MW)	1539.907 lb (698.490 kg)		1.492 in (37.896 mm)		239.958 and 29.83 ksi (205.670 N/mm <sup>2</sup> )		2										
Des2 (MJD)	6386.020 lb (2896.649 kg)		1.999 in (50.774 mm)		1239.191 and 47.89 ksi (330.189N/mm <sup>2</sup> )		3										
Des3 (MEF)	9847.947 lb (4466.953 kg)		1.513 in (38.430 mm)		1240.472 and 36.80 ksi (253.727 N/mm <sup>2</sup> )		3										
Des4 (MS1)	8392.790 lb (3806.905 kg)		1.748 in (44.419 mm)		210.437 and 50.36 ksi (347.219 kN/mm <sup>2</sup> )		3										
Des5 (MS2)	7684.263 lb (3485.523 kg)		1.908 in (48.463 mm)		1239.290 and 49.96 ksi (344.462 N/mm <sup>2</sup> )		3										
Lee (2004)	2580.81 lb (1170.635 kg)		N/A		N/A		N/A										
Khot (1984)	2581.89 lb (1171.125 kg)		N/A		N/A		N/A										
Li (2007)	2581.94 lb (1171.148 kg)		N/A		N/A		N/A										

MW: Minimum Weight, MJD: Maximum Joint Displacement, MEF: Maximum Member Force

MS1: Maximum stress great than the predefined stress-related constraint (50 ksi) (if predefined displacement-related constraint (2 in) is only used)

MS2: Possible maximum stress less than the predefined stress-related constraint(50 ksi) (if both predefined displacement (2 in) and stress-related constraints (50 ksi) are simultaneously used)

a1: PIPST (Steel Profile with Standard Tubular Cross-sectional)

a2: PIPEST (Extra Strong Steel Profile with Standard Tubular Cross-sectional)

a3: PIPEEST (Double-Extra Strong Steel Profile with Standard Tubular Cross-sectional)

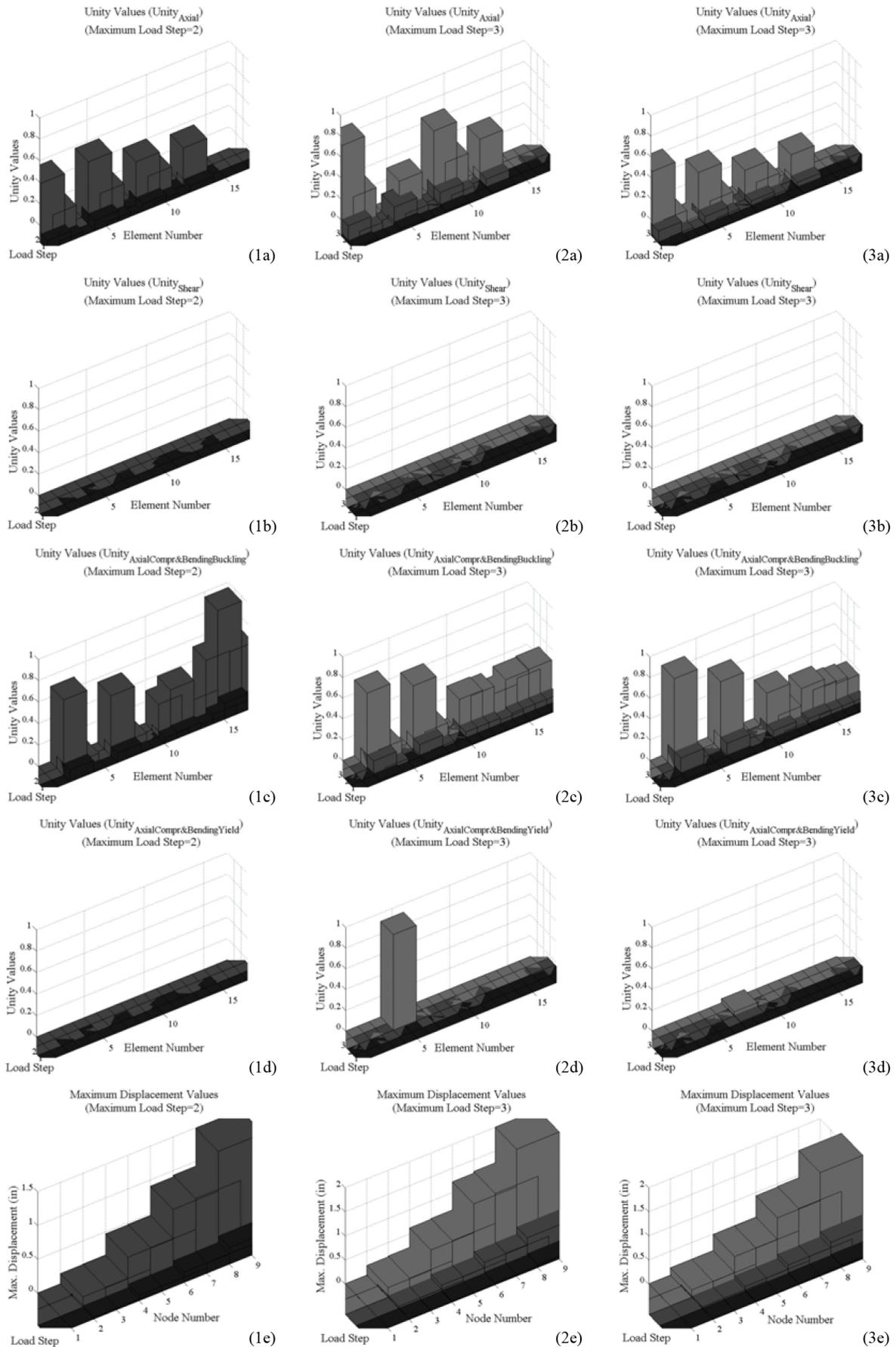


Fig. 8. Variation in unity values obtained for Des1 (1a–1e), Des2 (2a–2e) and Des3 (3a–3e) (Table 4)

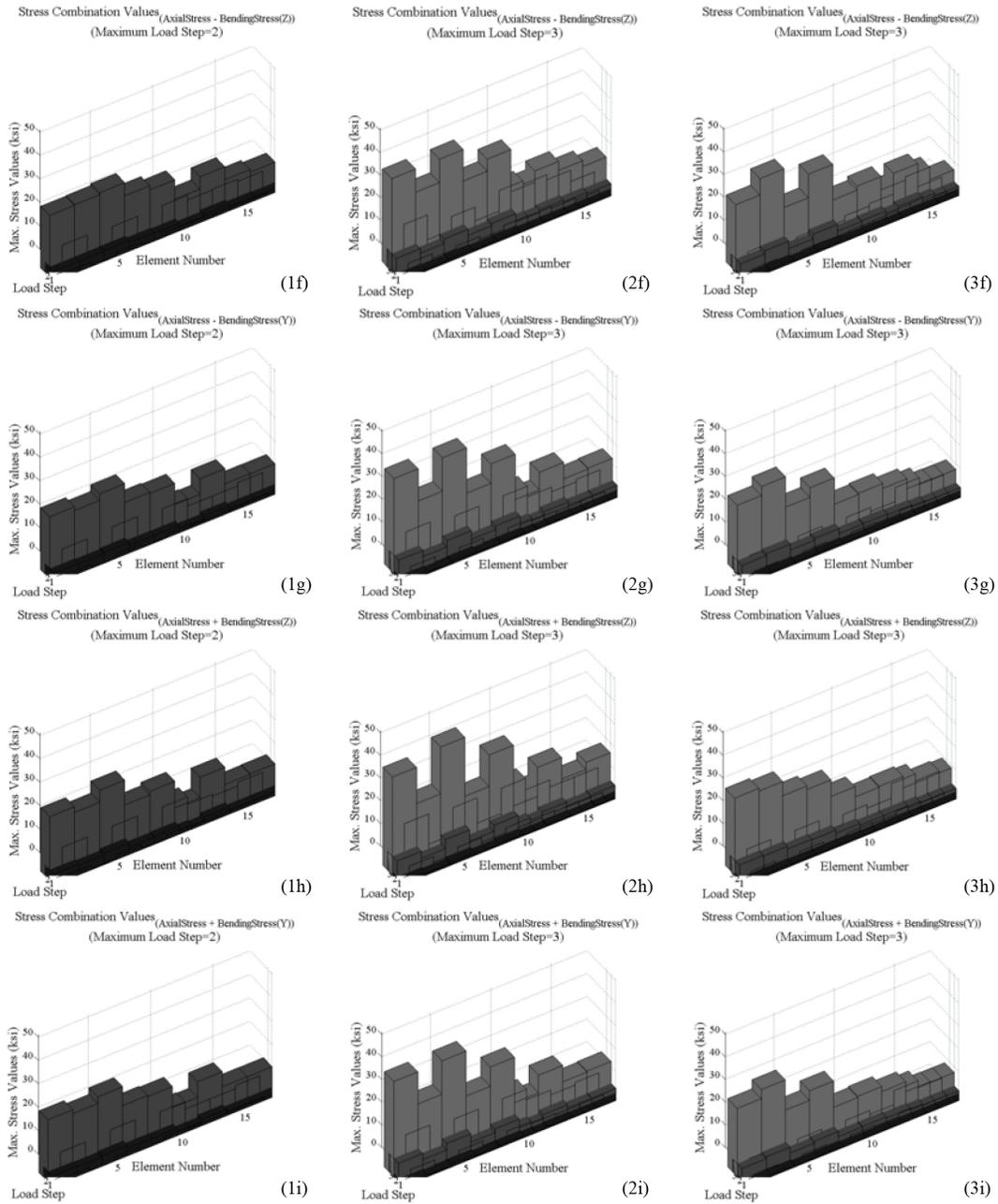


Fig. 9. Variation in maximum stress values obtained by use of stress combinations for Des1 (1f–1i), Des2 (2f–2i) and Des3 (3f–3i) (Table 4)

For example, a designation denoted by Des4 in Table 4 is obtained by constraining only the joint displacements to an upper value 2.00 in.; although its maximum stress value 50.36 ksi exceed its upper value 50 ksi, it achieves to satisfy the design constraints described according to the provisions of API RP2A-LRFD specification (1993). Furthermore, a designation denoted by Des5, maximum stress value of which is almost equal 50 ksi, is also included to illustrate the usage of an available specification to be more realistic approach (Table 4). It is noted that the load-carrying capacity 210.437 obtained by Des4 is lower than 1239.290 obtained by Des5 (see Table 4). Therefore, constraining the member stresses to the their upper values causes to a decrease in the number of

feasible designations since the designations satisfied the provisions of API RP2A-LRFD specification (1993) are automatically excluded from the feasible designation set.

Although it is highlighted that the usage of a design specification leads an increase in feasible design, the other preliminary reason is related with choosing the design variables from a ready steel profile set with various cross-sectional properties. The design variables of continuous type are generally utilized to represent the cross-sectional areas of lattice girder member and compute the structural response by use a linear structural analysis method ignoring the other important cross-sectional properties (torsional constant, elastic and plastic section modules, etc.). Thus, the entire weight of lattice

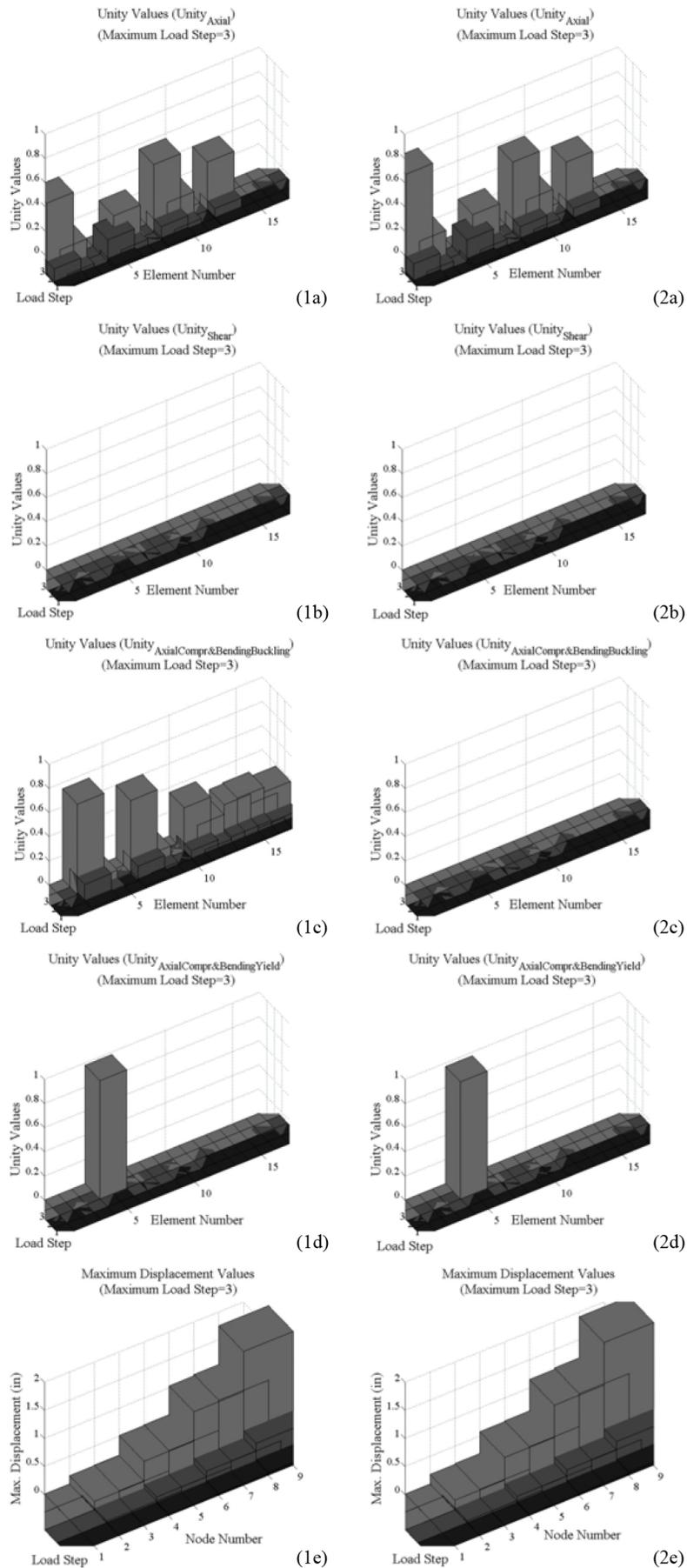


Fig. 10. Variation in unity values obtained for Des4 (1a–1e), Des5 (2a–2e) (Table 4)

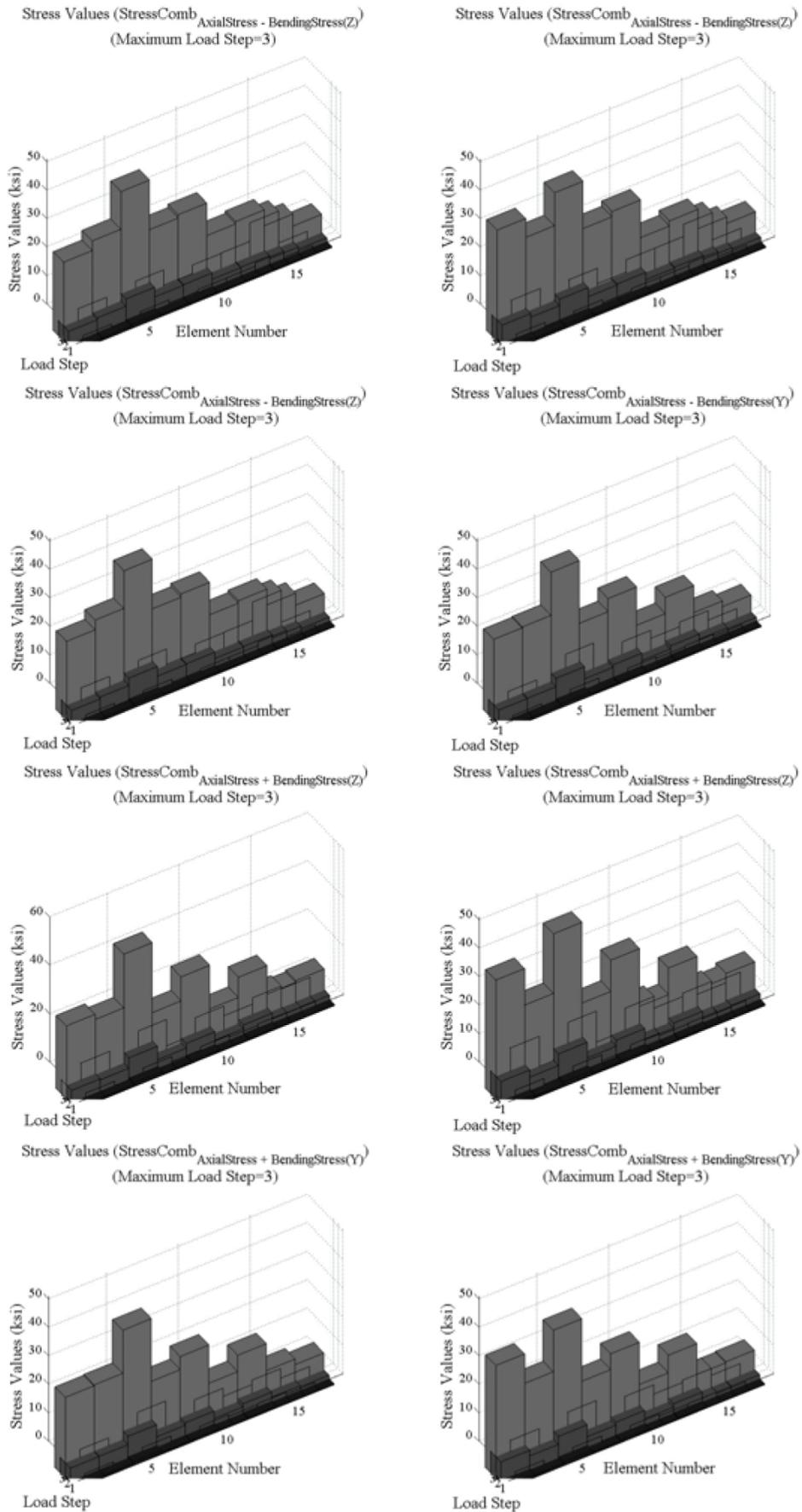


Fig. 11. Variation in maximum stress values obtained by use of stress combinations for Des4 (1f-1i), Des5 (2f-2i) (Table 4)

Table 5. Design input data of a lattice girder with three different span and loading conditions

	Cases		
	Lattice Girder 1	Lattice Girder 2	Lattice Girder 3
<i>Design Input Data</i>			
Member Material, Geometry and Loading Conditions Including Limit of Joint Displacement			
Total Load Value	70 kipf (311.375 kN)	140 kipf (622.751 kN)	150 kipf (667.233 kN)
Length of Span	393.70 in. (10 m)	393.70 in. (10 m)	787.40 in. (20 m)
Max. Joint Disp.	3.94 in (100 mm)		
Yielding Point Val.	36 ksi (248.211 N/mm <sup>2</sup> )		
Elasticity Module	29732 ksi (205 kN/mm <sup>2</sup> )		
Size-related design Variables			
Par <sub>ND</sub>	4	4	4
Par <sub>UDV</sub>	37	37	37
Par <sub>LDV</sub>	1	1	1
Topology-related design Variables			
Par <sub>UDN</sub>	15	20	40
Par <sub>LDN</sub>	5	5	10
Shape-related design Variables			
Par <sub>UH2</sub>	35.433 (0.9 m)	47.244 (1.2 m)	59.055 (1.5 m)
Par <sub>LH2</sub>	23.622 (0.6 m)	23.622 (0.6 m)	31.496 (0.8 m)
Par <sub>UH1</sub>	23.622 (0.6 m)	23.622 (0.6 m)	31.496 (0.8 m)
Par <sub>LH1</sub>	3.937 (0.1 m)	3.937 (0.1 m)	7.874 (0.2 m)

Table 6. A statistical assessment results of spread and average distances values for design example 2

		Average Distance			Statistical Significance	Spread			Statistical Significance
		Max.	Min.	Aver.		Max.	Min.	Aver.	
LattGir1	NSGAI	0.2990	0.1185	0.2065	0.0003	0.1998	0.0146	0.1053	0.0006
	ImpNSGAI	0.2883	0.1119	0.1758		0.1876	0.0047	0.1040	
LattGir2	NSGAI	0.3980	0.1263	0.2850	0.0041	0.2833	0.0364	0.1764	0.0052
	ImpNSGAI	0.3531	0.0688	0.2175		0.2572	0.0263	0.1523	
LattGir3	NSGAI	0.1961	0.0763	0.1429	0.0025	0.0904	0.0165	0.0560	0.0032
	ImpNSGAI	0.1807	0.0536	0.1290		0.0945	0.0017	0.0401	

girder is correspondingly increased. However, taking into account of these important cross-sectional properties provides an increase in the load-carrying capacity of lattice girder leading to a reduction in its entire weight. This observation is easily justified by considering the maximum cross-sectional areas of optimal designations corresponding to the minimum weight, 4.03 in<sup>2</sup> obtained by ImpNSGAI and 15.821 in<sup>2</sup>, 15.930 in<sup>2</sup> and 15.896 in<sup>2</sup> obtained by Lee and Geem (2004), Khot and Berke (1984) and Li and *et al.* (2007).

#### 4.2. Design example 2: a general tubular lattice girder with varying size, topology and shape

Considering the previous design example, it is shown that ImpNSGAI is an efficient optimization tool to explore the optimal designations by exploiting the evolutionary material gathered from a re-structured evolutionary

environment. In this design example, both ImpNSGAI and NSGAI are utilized to optimize a lattice girder with three different span lengths and total load values (Table 5). It is noted that the total load value imposed to the lattice girder is equally distributed to upper joints obtained by dividing the length of span (LS) by a division number (Par<sub>DN</sub>) in order to determine joint load values. Thus, total three cases with an increasing complexity are devised to investigate both the computing capacity of ImpNSGAI and the variation in load-carrying capacity of lattice girder.

Assessing the values of statistical significance values ( $P < 0.005$ ), the computing performance of ImpNSGAI and NSGAI are shown to be different with each other (Table 6). It is obvious that computing performance of ImpNSGAI is higher than NSGAI's considering the lower average values of spread (0.1758, 0.2175 and

Table 7. Variation in optimization-related parameters of both ImpNSGAI and NSGAI corresponding to minimum spread and average distance values (Table 6)

Names of Genetic Parameters	Par <sub>NNIN</sub> (ImpNSGAI)									
	1	2	3	4	5	6	7	8	9	10
options.Generations	19	8	11	2	6	4	5	23	24	11
options.PopulationSize	47	26	4	22	40	43	12	18	29	27
options.MutationFcn {@mutationuniform.}	0.1286	0.1640	0.2267	0.1056	0.4271	0.6158	0.4131	0.1101	0.2324	0.2934
options.CrossoverFcn {@crossoverheuristic.}	0.5631	0.5960	0.9042	0.9523	0.9015	0.3474	0.8937	0.7724	0.3496	0.7372
options.CrossoverFraction	0.4647	0.2603	0.1509	0.7672	0.1800	0.5081	0.9353	0.3858	0.8130	0.1871
options.MigrationInterval	2	3	4	4	4	2	2	2	3	3
options.MigrationFraction	0.1606	0.6823	0.5330	0.8600	0.5739	0.2375	0.4844	0.1307	0.7244	0.1817
options.SelectionFcn {@selectiontournament.}	0.7863	0.7407	0.9862	0.3958	0.5444	0.1221	0.3343	0.9326	0.6413	0.3648
Names of Genetic Parameters	Par <sub>NNIN</sub> =1 (NSGAI)									
	options.Generations	100								
options.PopulationSize	50									
options.MutationFcn {@mutationuniform.}	0.5									
options.CrossoverFcn {@crossoverheuristic.}	0.5									
options.CrossoverFraction	0.5									
options.MigrationInterval	5									
options.MigrationFraction	0.5									
options.SelectionFcn {@selectiontournament.}	0.5									

0.1290) and average distance values (0.1040, 0.1523 and 0.0401) for lattice girders (1–3) (Table 6).

The forms of total three true Pareto fronts are visualized in Figures 12a–14a including the Pareto fronts of ImpNSGAI (Figs 12b–14b) and NSGAI (Figs 12c–14c). It is noted that the Pareto fronts of ImpNSGAI and NSGAI are obtained by use of the optimization-related parameters corresponding to the lowest spread (0.1119, 0.0688 and 0.0536) and average distance values (0.0047, 0.0364 and 0.0017) for the lattice girder (1–3). Due to the self-adaptive search mechanism of ImpNSGAI, the

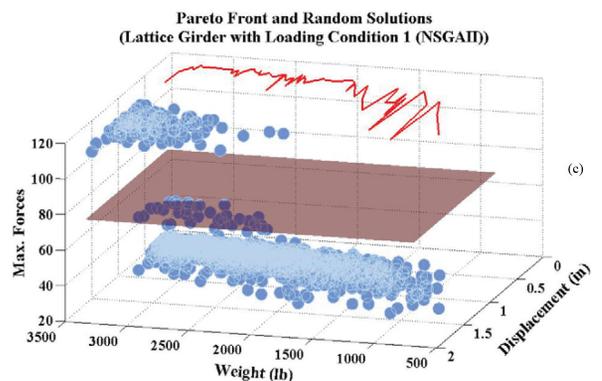
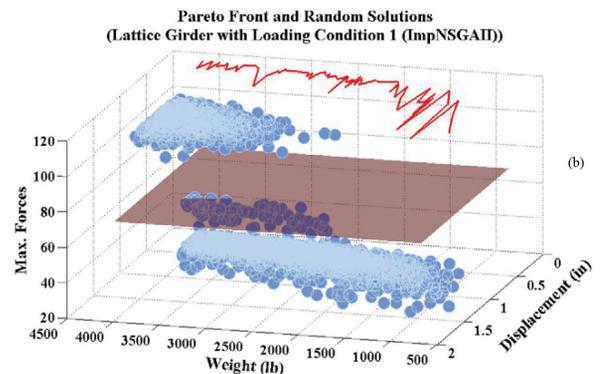
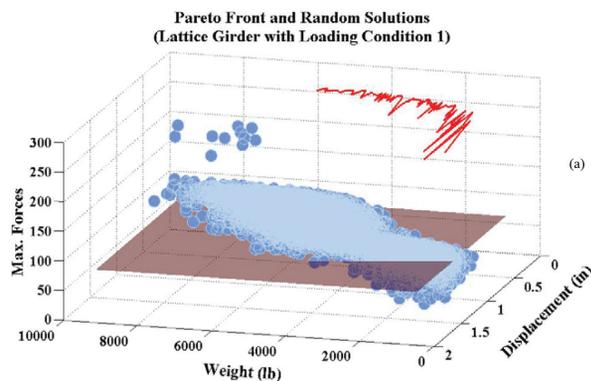


Fig. 12. True Pareto front (a), Pareto fronts and random solutions obtained by ImpNSGAI

Fig. 12. (b) and NSGAI (c) corresponding to minimum spread and average values (Table 6) (design example 2, lattice girder 1)

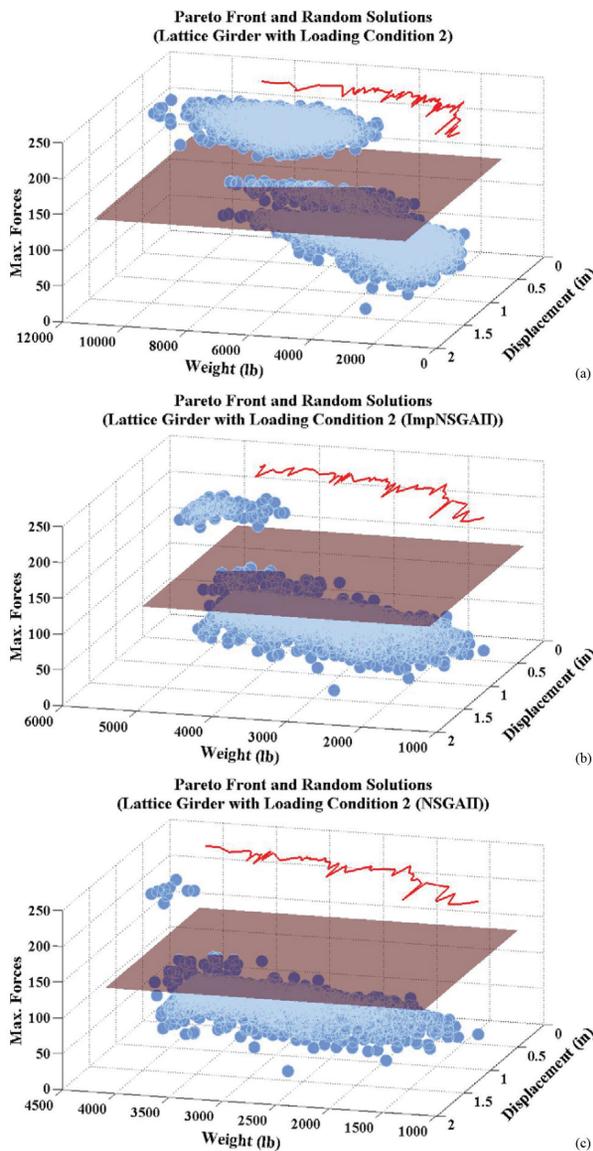


Fig. 13. True Pareto front (a), Pareto fronts and random solutions obtained by ImpNSGAI (b) and NSGAI (c) corresponding to minimum spread and average values (Table 6) (design example 2, lattice girder 2)

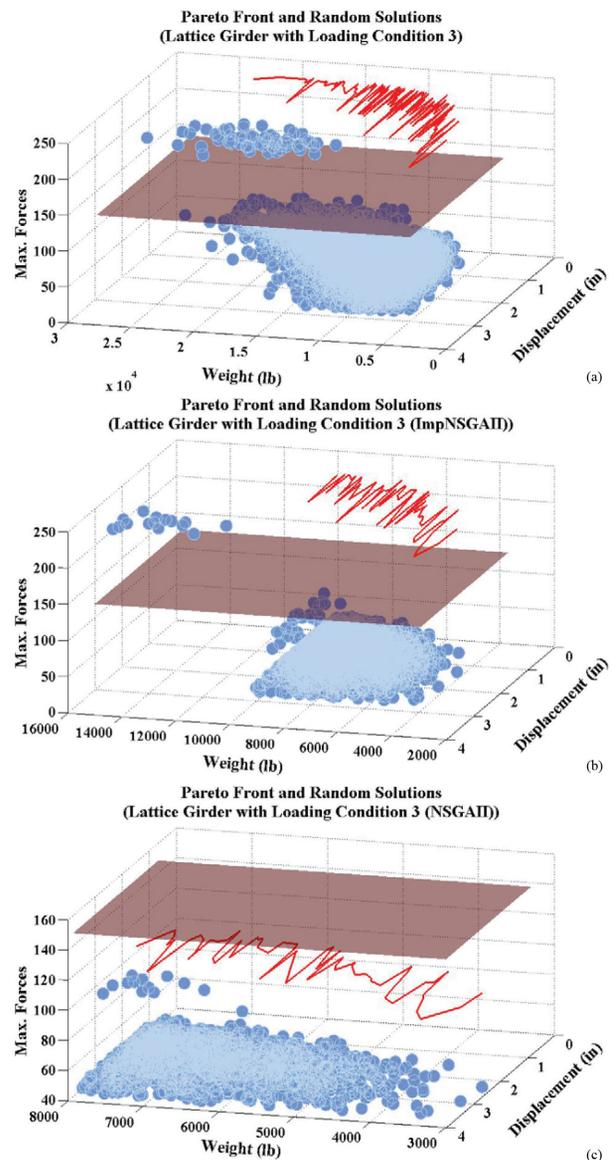


Fig. 14. True Pareto front (a), Pareto fronts and random solutions obtained by ImpNSGAI (b) and NSGAI (c) corresponding to minimum spread and average values (Table 6) (design example 2, lattice girder 3)

optimization-related parameters are adaptively adjusted according to parameter  $Par_{NNIN}$ . Therefore, the variation in the optimization-related parameters is listed in Table 7.

In order to investigate the load-carrying capacity of lattice girders (1–3), the total load values represented by flat surfaces are included into Figures 12–14 regarding to Pareto front forms. In this regard, the feasible designations, maximum member forces of which are higher than the total load values (70 kipf (311.375 kN), 140 kipf (622.751 kN) and 150 kipf (667.33 kN) for the lattice girders (1–3)), are easily seen. Considering Figures 12–14, the increasing complexity arisen from an elevation in both span lengths and loading conditions causes to a decrease in load-carrying capacity of lattice girders. Thus, the number of feasible designations is correspondingly decreased along with their load-carrying

capacities (Figs 12a–14a). Several extreme designations borrowed from Figures 12a–14a are tabulated to present the values of size, shape and topology-related design variables (Table 8). In order to make a further investigation for load-carrying capacities of lattice girders (1–3), the size, shape and topology-related design variables of feasible designations are listed in Table 9. Thus, it is observed that some designations defined as feasible designation in Table 9 are identical to ones tabulated in Table 8.

Taking into account of the designations presented in Table 9, it is achieved to keep the load-carrying capacity in a higher level in spite of an increase in span length and total load values. For example, whereas the maximum member force of lattice girder optimized using the design conditions of lattice girder 1 is 255.086, it is 232.224 for

Table 8. Some of extreme values obtained by considering all random designations (Figs 9a–11a)

	Lattice Girder 1			Lattice Girder 2			Lattice Girder 3		
	Des1 (MW)	Des2 (MJD)	Des3 (MEF)	Des1 (MW)	Des2 (MJD)	Des3 (MEF)	Des1 (MW)	Des2 (MJD)	Des3 (MEF)
Par <sub>DN</sub>	6	10	10	12	12	10	22	36	10
Par <sub>H2</sub>	7.187	4.047	6.117	12.903	5.366	23.208	28.858	8.915	10.393
Par <sub>H1</sub>	29.002	24.413	35.095	35.707	24.097	45.197	58.838	38.049	58.005
D1	PIPST <sup>a1</sup> 3	PIPST 10	PIPEST 12	PIPST 4	PIPST 5	PIPEST <sup>a2</sup> 8	PIPST 5	PIPST 8	PIPEST 12
D2	PIPST 3	PIPEST 6	PIPEST 10	PIPEST <sup>3</sup> 1/2	PIPEEST <sup>a3</sup> 2 1/2	PIPEEST 5	PIPEST 3 1/2	PIPEEST 3	PIPEST 12
D3	PIPEST 2	PIPEEST 5	PIPEST 12	PIPEST 2	PIPEST 10	PIPEEST 3	PIPEST 2	PIPST 12	PIPEST 12
D4	PIPEST 2 1/2	PIPEEST 5	PIPEST 8	PIPST 2 1/2	PIPEEST 3	PIPEEST 3	PIPEEST 2	PIPST 12	PIPST 10
Entire Weight	811.389 lb (368.039 kg)	3567.378 lb (1618.135 kg)	6062.082 lb (2749.714 kg)	1167.992 lb (529.792 kg)	2437.039 lb (1105.422kg)	3860.235 lb (1750.973 kg)	3041.241 lb (1379.483 kg)	11250.487 lb (5103.135kg)	12847.894 lb (5827.706 kg)
Joint Displacement	1.242 in (31.546 mm)	1.768 in (44.907 mm)	1.138 in (28.905 mm)	0.897 in (22.783 mm)	1.912 in (48.564 mm)	0.944 in (23.977 mm)	2.576 in (65.430 mm)	3.932 in (99.872 mm)	2.912 in (73.964 mm)
Maximum Member Force	21.002	108.392	255.086	41.951	41.923	217.007	44.979	44.925	232.224
Load Step Number	2	3	3	2	2	3	2	2	3
									

MW: Minimum Weight, MJD: Maximum Joint Displacement, MEF: Maximum Member Force

a1: PIPST (Steel Profile with Standard Tubular Cross-sectional)

a2: PIPEST (Extra Strong Steel Profile with Standard Tubular Cross-sectional)

a3: PIPEEST (Double-Extra Strong Steel Profile with Standard Tubular Cross-sectional)

Table 9. Some of extreme values corresponding to feasible designations with higher maximum member forces than total load value (Figs 9a–11a)

	Lattice Girder 1			Lattice Girder 2			Lattice Girder 3		
	Des1 (MW)	Des2 (MJD)	Des3 (MEF)	Des1 (MW)	Des2 (MJD)	Des3 (MEF)	Des1 (MW)	Des2 (MJD)	Des3 (MEF)
Par <sub>DN</sub>	6	10	10	8	10	10	12	26	10
Par <sub>H2</sub>	22.737	4.047	6.117	15.769	10.113	23.208	25.420	24.398	10.393
Par <sub>H1</sub>	32.489	24.413	35.095	40.122	27.435	45.197	58.205	44.795	58.005
D1	PIPEST 6	PIPST 10	PIPEST 12	PIPEST 8	PIPEST 10	PIPEST 8	PIPEST 10	PIPEEST 8	PIPEST 12
D2	PIPST 6	PIPEST 6	PIPEST 10	PIPST 12	PIPEST 8	PIPEEST 5	PIPEST 10	PIPEST 12	PIPEST 12
D3	PIPEST 4	PIPEEST 5	PIPEST 12	PIPST 6	PIPEST 10	PIPEEST 3	PIPST 8	PIPEEST 6	PIPEST 12
D4	PIPST 5	PIPEEST 5	PIPEST 8	PIPEST 5	PIPEST 12	PIPEEST 3	PIPEEST 4	PIPEEST 6	PIPST 10
Entire Weight	2137.142 lb (969.391 kg)	3567.378 lb (1618.135 kg)	6062.082 lb (2749.714 kg)	3804.047 lb (1725.486 kg)	6589.918 lb (2989.136 kg)	3860.235 lb (1750.973 kg)	11348.644 lb (5147.658 kg)	20222.117 lb (9172.597 kg)	12847.894 lb (5827.706 kg)
Joint Displacement	1.072 in (27.228 mm)	1.768 in (44.907 mm)	1.138 in (28.905 mm)	0.934 in (23.723 mm)	1.507 in (38.277mm)	0.944 in (23.977 mm)	2.798 in (71.069 mm)	3.569 in (90.652 mm)	2.912 in (73.964 mm)
Maximum Member Force	108.348	108.392	255.086	216.887	216.658	217.007	232.131	232.096	232.224
Load Step Number	3	3	3	3	3	3	3	3	3












Table 10. Statistical data indicated load step numbers corresponding to termination of nonlinear structural analysis (design example 2)

	ImpNSGAI			NSGAI		
	Max	Min	Mean	Max	Min	Mean
LattGir1	2	2	2	2	2	2
LattGir2	2	2	2	2	2	2
LattGir3	2	2	2	2	2	2

the lattice girder 3 that has a more severe design complexity. A statistical data indicated the load step numbers is also tabulated in Table 10.

**Concluding remarks**

The design of geometrically nonlinear tubular lattice girders is optimized by use a multi-objective optimization approach named NSGAI thereby checking their member strengths according to the design constraints

based on the provisions of API RP2A-LRFD specification. The incremental and iterative based arc-length method used for modeling the geometrical nonlinearity is utilized to compute the member forces of lattice girder. In order to increase the flexibility of NSGAI, an automatic lattice girder generator is included into the design procedure of NSGAI. Furthermore, the evolutionary search capacity of NSGAI is also increased by an implementation of radial-basis neural network to its optimization procedure. Consequently, the following results are drawn from the application of proposed design approach for the optimal design of lattice girders:

- The proposed optimization procedure named ImpNSGAI does not require any penalizing process utilized in case of any violation of the design constraints. Thus, it is possible to obtain a designation which is not feasible due to its load-carrying capacity being lower than the total load value imposed to the lattice girder.
- ImpNSGAI is capable of generating a lattice girder configuration with higher load-carrying capacity with respect to the total load value.
- The computing performance of ImpNSGAI is higher than both NSGAI and the other studies in Literature ensuring lower weight and joint displacement values and higher load-carrying capacity.
- The use of a multi-objective optimization approach increases the correctness degree in evaluating the optimality quality of designations obtained.
- Utilizing an available design specification along with the discrete design variables assigned from a ready steel profile set leads to an increase in the number of feasible designations compared to the usage of a user-defined design criteria (member stress and etc.) and design variables of continuous type.

Thus, the proposed optimal design approach developed by considering the geometrically nonlinear behavior of lattice girders is an efficient optimization tool in order to obtain an optimal design of lattice girders. In the second part of this study, joint strength-related design constraints will be also included into the current design constraints. Thus, it is possible to investigate the variation in the optimality quality of designations.

## References

- American Petroleum Institute (API). 1993. *API RP2A-LRFD, Recommended Practice for Planning, Designing and Constructing Fixed Offshore Platforms – Load and Resistance Factor Design*. 1<sup>st</sup> ed. Errata, USA. 49 p.
- American Petroleum Institute (API). 2000. *API RP2A-WSD, Recommended Practice for Planning, Designing and Constructing Fixed Offshore Platforms. Working Stress Design*. 22<sup>th</sup> ed. Errata, USA. 57 p.
- ANSYS [online], [cited 12 Oct 2012]. Available from Internet: <http://www.ansys.com>
- Çarbaş, S.; Saka, M. P. 2012. Optimum topology design of various geometrically nonlinear latticed domes using improved harmony search method, *Structural and Multidisciplinary Optimization* 45(3): 377–399. <http://dx.doi.org/10.1007/s00158-011-0675-2>
- Chang, P. C.; Chen, S. H. 2009. The development of a sub-population genetic algorithm II (SPGA II) for multi-objective combinatorial problems, *Applied Soft Computing* 9(1): 173–181. <http://dx.doi.org/10.1016/j.asoc.2008.04.002>
- Davison, B.; Owens, G. W. (Eds). 2005. *Steel designer's manual*. 6th ed. John Wiley & Sons. 1368 p. <http://dx.doi.org/10.1002/9780470775097>
- Deb, K. 2001. *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons. 518 p.
- Deb, K.; Agrawal, S.; Pratap, A.; Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation* 6(2): 182–197. <http://dx.doi.org/10.1109/4235.996017>
- Hrinda, G. A.; Nguyen, D. T. 2008. Optimization of stability-constrained geometrically nonlinear shallow trusses using an arc length sparse method with a strain energy density approach, *Finite Elements in Analysis and Design* 44(15): 933–950. <http://dx.doi.org/10.1016/j.finel.2008.07.004>
- Kamat, M. P.; Khot, N. S.; Venkayya, V. B. 1984. Optimization of shallow trusses against limit point instability, *AIAA Journal* 22(3): 403–408. <http://dx.doi.org/10.2514/3.48461>
- Kamat, M. P.; Raungasilasingha, P. 1985. Optimization of space trusses against instability using design sensitivity derivatives, *Engineering Optimization* 8(3): 177–188. <http://dx.doi.org/10.1080/03052158508902488>
- Kaveh, A.; Talatahari, S. 2011. Geometry and topology optimization of geodesic domes using charged system search, *Structural and Multidisciplinary Optimization* 43(2): 215–229. <http://dx.doi.org/10.1007/s00158-010-0566-y>
- Kaveh, A.; Laknejadi, K. 2011. A novel hybrid charge system search and particle swarm optimization method for multi-objective optimization, *Expert Systems with Applications* 38(12): 15475–15488. <http://dx.doi.org/10.1016/j.eswa.2011.06.012>
- Khot, N. S.; Berke, L. 1984. Structural optimization using optimality criteria methods, in Atrek, E.; Gallagher, R. H.; Ragsdell, K. M.; Zienkiewicz, O. C. (Eds.). *New directions in optimum structural design*. John Wiley & Sons Inc., 271–311.
- Kurobane, Y.; Packer, J. A.; Wardenier, J.; Yeomans, N. 2005. *CIDECT: Design Guide 9 for structural hollow section column connections*. Berlin: Verlag TÜV Rheinland. 159 p.
- Lee, K. S.; Geem, Z. W. 2004. A new structural optimization method based on the harmony search algorithm, *Computers & Structures* 82(9–10): 781–798. <http://dx.doi.org/10.1016/j.compstruc.2004.01.002>
- Li, L. J.; Huang, Z. B.; Liu, F.; Wu, Q. H. 2007. A heuristic particle swarm optimizer for optimization of pin connected structures, *Computers & Structures* 85(7–8): 340–349. <http://dx.doi.org/10.1016/j.compstruc.2006.11.020>
- Mahmoodabadi, M. J.; Bagheri, A.; Mostafaei, S. A.; Bisheban, M. 2011. Simulation of stability using Java application for Pareto design of controllers based on a new multi-objective particle swarm optimization, *Mathematical and Computer Modelling* 54(5–6): 1584–1607. <http://dx.doi.org/10.1016/j.mcm.2011.04.032>
- Nelson, J. K.; McCormac, J. C. 2003. *Structural analysis 3E WSE: using classical and matrix methods*. John Wiley & Sons, Inc. 544 p.
- Omkar, S. N.; Khandelwal, R.; Anath, T. V. S.; Narayana, G. N.; Gopalakrishnan, S. 2009. Quantum behaved Particle Swarm Optimization (QPSO) for multi-objective design optimization of composite structures, *Expert Systems with Applications* 36(8): 11312–11322. <http://dx.doi.org/10.1016/j.eswa.2009.03.006>
- Omkar, S. N.; Senthilnath, J.; Khandelwal, R.; Naik, G. N.; Gopalakrishnan, G. 2011. Artificial Bee Colony (ABC) for

- multi-objective design optimization of composite structures, *Applied Soft Computing* 11(1): 489–499. <http://dx.doi.org/10.1016/j.asoc.2009.12.008>
- Park, H.; Kwak, N. S.; Lee, J. 2009. A method of multiobjective optimization using a genetic algorithm and an artificial immune system, *Journal of Mechanical Engineering Science* 223(5): 1243–1252. <http://dx.doi.org/10.1243/09544062JMES1313>
- Ramesh, S.; Kannan, S.; Baskar, S. 2011. Application of modified NSGA II algorithm to multi-objective reactive power planning, *Trends in Computer Science, Engineering and Information Technology. Communications in Computing and Information Science* 204: 344–354. [http://dx.doi.org/10.1007/978-3-642-24043-0\\_35](http://dx.doi.org/10.1007/978-3-642-24043-0_35)
- Saka, M. P. 2007. Optimum topological design of geometrically nonlinear single layer latticed domes using coupled genetic algorithm, *Computers & Structures* 85(21–22): 1635–1646. <http://dx.doi.org/10.1016/j.compstruc.2007.02.023>
- Santos, J. L. T.; Choi, K. K. 1988. Sizing design sensitivity analysis of non-linear structural systems. Part II: numerical method, *International Journal of Numerical Methods in Engineering* 26(9): 2039–2055. <http://dx.doi.org/10.1002/nme.1620260913>
- Simo, J. C. 1986. A three-dimensional finite-strain rod model. Part II: computational aspects, *Computer Methods in Applied Mechanics and Engineering* 58(1): 79–116. [http://dx.doi.org/10.1016/0045-7825\(86\)90079-4](http://dx.doi.org/10.1016/0045-7825(86)90079-4)
- Srinivas, N.; Deb, K. 1995. Multiobjective optimization using nondominated sorting in genetic algorithms, *Evolutionary Computation* 2(3): 221–248. <http://dx.doi.org/10.1162/evco.1994.2.3.221>
- Suleman, A.; Sedaghati, R. 2005. Benchmark case studies in optimization of geometrically nonlinear structures, *Structural and Multidisciplinary Optimization* 30(4): 273–296. <http://dx.doi.org/10.1007/s00158-005-0524-2>
- Tan, K. C.; Goh, C. K.; Mamun, A. A.; Ei, E. Z. 2008. An evolutionary artificial immune system for multi-objective optimization, *European Journal of Operational Research* 187(2): 371–392. <http://dx.doi.org/10.1016/j.ejor.2007.02.047>
- Wardenier, J. 2001. *Hollow sections in structural applications*. Delft: Delft University of Technology. 199 p.
- Wardenier, J.; Dutta, D.; Yeomans, N.; Packer, J. A.; Bucak, Ö. 1998. *CIDECT: Design Guide 6 for structural hollow sections in mechanical applications*. Berlin: Verlag TÜV Rheinland. 157 p.
- Wu, C. C.; Arora, J. S. 1988. Design sensitivity analysis of non-linear buckling load, *Computational Mechanics* 3(2): 129–140. <http://dx.doi.org/10.1007/BF00317060>
- Yagmahan, B. 2011. Mixed-model assembly line balancing using a multi-objective ant colony optimization approach, *Expert Systems with Applications* 38(10): 12453–12461. <http://dx.doi.org/10.1016/j.eswa.2011.04.026>
- Zhang, Y.; Gong, D.-W.; Ding, Z.-H. 2011. Handling multi-objective optimization problems with a multi-swarm cooperative particle swarm optimizer, *Expert Systems with Applications* 38(11): 13933–13941.
- Zitzler, E.; Laumanns, M.; Thiele, L. 2001. *SPEA2: Improving strength Pareto evolutionary algorithm*. Zurich: Swiss Federal Institute Technology. 21 p.
- Zitzler, E.; Thiele, L. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, *IEEE Transaction on Evolutionary Computation* 3(4): 257–271. <http://dx.doi.org/10.1109/4235.797969>

**Tugrul TALASLIOGLU**, Dr, Assist. Prof. at Osmaniye Korkut Ata University (OKU) in Osmaniye, Turkey. He is a member of UCTEA (Union of Chambers of Turkish Engineers and Architects) and AISC (American Institute of Steel Construction). He is currently a Director of Vocational School of Kadirli. He is also Founder and Researcher of OKU-AESDC (Advanced Engineering Software Development Center) at the Vocational School of Kadirli. His research interests include the design optimization of steel structures and Engineering problems, improvement and development of optimization algorithms, and simulation of linear and non-linear behavior of steel structures.