INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2011; 4:507-520

AN OPTIMIZATION-BASED COMPARATIVE STUDY OF DOUBLE LAYER GRIDS WITH TWO DIFFERENT CONFIGURATIONS USING CUCKOO SEARCH ALGORITHM

A. Kaveh[*, †], T. Bakhshpoori and E. Afshari
Centre of Excellence for Fundamental Studies in Structural Engineering, School of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran

ABSTRACT

This paper is concerned with the economical comparison between two commonly used configurations for double layer grids and determining their optimum span-depth ratio. Two ranges of spans as small and big sizes with certain bays of equal length in two directions and various types of element grouping are considered for each type of square grids. In order to carry out a precise comparison between different systems, optimum design procedure based on the Cuckoo Search (CS) algorithm is developed. The CS is a meta-heuristic algorithm recently developed that is inspired by the behavior of some Cuckoo species in combination with the Lévy flight behavior of some birds and insects. The design algorithm obtains minimum weight grid through appropriate selection of tube sections available in AISC Load and Resistance Factor Design (LRFD). Strength constraints of AISC-LRFD specification and displacement constraints are imposed on grids. The comparison is aimed at finding the depth at which each of the different configurations shows its advantages. The results are graphically presented from which the optimum depth can easily be estimated for each type, while the influence of element grouping can also be realized at the same time.

Received: 10 October 2011; Accepted: 29 December 2011

KEY WORDS: Double layer grids, Cuckoo Search algorithm, Optimization, Span-depth ratio, Optimum depth, Element grouping

1. INTRODUCTION

[*]Corresponding author: A. Kaveh, Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran
[†]E-mail address: alikaveh@iust.ac.ir

Considering the industrialization and development of the modern structures, it seems to be a demand for efficient and adaptable long-span structures. Double-layer grids are one of the most efficient and light weight space grid structural systems. It is difficult to generalize the most economical span-depth ratio for the space grid structures, since it is influenced by the method of support, type of loading and to a large extent the configuration of the system being employed [1].

This study focuses on economical comparison of two commonly used double layer grid configurations, namely two-way on two-way grid and diagonal on diagonal grid and determining their optimum span-depth ratio. The span ranges of 15×15m and 40×40m with certain bays of equal length in two directions are considered as small and big sizes grids, respectively. Bottom layer is simply supported at the corner nodes, and as mid-edge at two parallel sides of the grid for the small and big span cases, respectively. The range of discrete depths from a certain interval with 0.5m increment is considered for each case to achieve the optimum depth. For determining the grouping effects various grouping patterns are applied in each case.

Optimum design procedure was developed based on the Cuckoo Search algorithm to carry out a precise comparison between different configurations. The CS is one of the recently developed meta-heuristic algorithms inspired by the behavior of some Cuckoo species in combination with the Lévy flight behavior of some birds and insects [2]. The CS has been used for optimum design of steel truss structures [3] and two dimensional steel frames [4]. The design algorithm is supposed to obtain minimum weight grid through suitable selection of tube sections available in AISC-LRFD [5]. Strength constraints of AISC-LRFD specification and displacement constraints are imposed on grids. Moreover, two other powerful advanced hybrid algorithms consisting of the HPSACO [6, 7] (based on PSO, ACO and HS algorithms) and the HBB-BC [8] (based on BB-BC and PSO methods) are applied to carry out a precise assessment, and demonstrate the effectiveness and robustness of the CS. Because of time-consuming optimization procedure, these methods are utilized only at the obtained optimum height and adjacent depths using the CS.

Based on the results, the optimum depth can be estimated for each type, and the influences of element grouping can be realized. The comparisons of the numerical results obtained by the CS with those of the two other optimization methods demonstrate the efficiency and robustness of the CS algorithm in achieving better designs.

2. OPTIMUM DESIGN OF DOUBLE LAYER GRIDS

The allowable cross sections are considered as 37 steel pipe sections shown in Table 1, where the abbreviations ST, EST, and DEST stand for standard weight , extra strong , and double-extra strong, respectively. These sections are taken from AISC -LRFD [5] which is also utilized as the code of design.

Table 1. The allowable steel pipe sections taken from AISC-LRFD

	Type	Nominal diameter (in)	Weight per ft (lb)	Area (in^2)	I (in^4)	Gyration radius (in)	J (in^4)
1	ST	1/2	0.85	0.25	0.017	0.261	0.082
2	EST	1/2	1.09	0.32	0.2	0.25	0.096
3	ST	3/4	1.13	0.333	0.037	0.334	0.142
4	EST	3/4	1.47	0.433	0.045	0.321	0.17
5	ST	1	1.68	0.494	0.087	0.421	0.266
6	EST	1	2.17	0.639	0.106	0.407	0.322
7	ST	1 1/4	2.27	0.669	0.195	0.54	0.47
8	ST	1 1/2	2.72	0.799	0.31	0.623	0.652
9	EST	1 1/4	3.00	0.881	0.242	0.524	0.582
10	EST	1 1/2	3.63	1.07	0.666	0.787	1.122
11	ST	2	2.65	1.07	0.391	0.605	0.824
12	EST	2	5.02	1.48	0.868	0.766	1.462
13	ST	2 1/2	5.79	1.7	1.53	0.947	2.12
14	ST	3	7.58	2.23	3.02	1.16	3.44
15	EST	2 1/2	7.66	2.25	1.92	0.924	2.68
16	DEST	2	9.03	2.66	1.31	0.703	2.2
17	ST	3 1/2	9.11	2.68	4.79	1.34	4.78
18	EST	3	10.25	3.02	3.89	1.14	4.46
19	ST	4	10.79	3.17	7.23	1.51	6.42
20	EST	3 1/2	12.50	3.68	6.28	1.31	6.28
21	DEST	2 1/2	13.69	4.03	2.87	0.844	4
22	EST	5	14.62	4.3	15.2	1.88	10.9
23	EST	4	14.98	4.41	9.61	1.48	8.54
24	DEST	3	18.58	5.47	5.99	1.05	6.84
25	ST	6	18.97	5.58	28.1	2.25	17
26	EST	5	20.78	6.11	20.7	1.84	14.86
27	DEST	4	27.54	8.1	15.3	1.37	13.58
28	ST	8	28.55	8.4	72.5	2.94	33.6
29	EST	6	28.57	8.4	40.5	2.19	24.4
30	DEST	5	38.59	11.3	33.6	1.72	24.2
31	ST	10	40.48	11.9	161	3.67	59.8
32	EST	8	43.39	12.8	106	2.88	49
33	ST	12	49.56	14.6	279	4.38	87.6
34	DEST	6	53.16	15.6	66.3	2.06	40
35	EST	10	54.74	16.1	212	3.63	78.8
36	EST	12	65.42	19.2	362	4.33	113.4
37	DEST	8	72.42	21.3	162	2.76	75.2

ST=standard weight, EST=extra strong, DEST=double-extra strong

The aim of optimizing the grid weight is to find a set of design variables that has the minimum weight satisfying certain constraints. This can be expressed as:

$$\text{Find} \qquad \{X\} = \left[x_1,\ x_2,\ \mathbf{K},\ x_{ng}\right], \quad x_i \in D = \{d_1, d_2, \mathbf{K}, d_{37}\}$$
$$\text{To minimize} \qquad W(\{X\}) = \sum_{i=1}^{ng} x_i \sum_{j=1}^{nm(i)} r_j . L_j \tag{1}$$

where *{X}* is the set of design variables; *ng* is the number of member groups in structure (number of design variables); *D* is the cross-sectional areas available for groups according to Table 1; *W({X})* presents weight of the grid; *nm(i)* is the number of members for the *ith* group; ρ_j and L_j denotes the material density and the length for the *jth* member of the *ith* group, respectively.

The constraint conditions for grid structures are briefly explained in the following:

Displacement constraint:

$$d^i \le d^{\max}, \quad i = 1, 2, \mathbf{K}, nn \tag{2}$$

Tension member constraint:

$$P_u \le P_r \quad : \qquad P_r = \min \begin{cases} F_y . A_g . f_t & f_t = 0.9 \\ F_u . A_e . f_t & f_t = 0.75 \end{cases} \tag{3}$$

Compression member constraint**:**

$$P_u \le P_r \qquad , P_r = f_c . F_{cr} . A_g \quad ; f_c = 0.85$$
$$F_{cr} = \begin{cases} (0.658^{\,F_y / F_e}) F_y & KL/r > 4.71\sqrt{E/F_y} \\ 0.877 F_e & KL/r > 4.71\sqrt{E/F_y} \end{cases} \quad , F_e = p^2 E / (KL/r)^2 \tag{4}$$

Slenderness ratio constraint:

$$\begin{aligned} l_c &= KL/r \le 200 && \textit{for compression members} \\ l_t &= KL/r \le 300 && \textit{for tension members} \end{aligned} \tag{5}$$

where δ_i and δ_i^{max} are the displacement and allowable displacement for the *ith* node; *nn* is the number of nodes; *nm* is the total number of members and *K* is the effective length factor taken equal to 1; P_u is the required strength (tension or compression); P_r is the nominal axial strength (tension or compression); A_g and A_e are the cross sectional and effective net area of a member, respectively.

In order to handle the constraints, a penalty approach is utilized. In this method, the aim of the optimization is redefined by introducing the cost function as:

$$f_{cost}\left(\{X\}\right)=(1+e_1.\,u)^{e_2}\times\,W\left(\{X\}\right),\quad u=\sum_{i=1}^{nn}u_i^d+\sum_{i=1}^{nm}\left(u_i^s+u_i^l\right) \tag{6}$$

where υ is the constraint violation function; υ_i^d, υ_i^σ, and υ_i^λ are constraint violation for displacement, stress and slenderness ratio, respectively. ε_1 and ε_2 are penalty function exponents which selected considering the exploration and the exploitation rate of the search space. Here, ε_1 is set to unity; ε_2 is selected in a way that it decreases the penalties and reduces the cross-sectional areas. Thus, in the first steps of the search process, ε_2 is set to 1 and ultimately increased to 3 [9].

3. OPTIMIZATION ALGORITHMS

Methods employed in structural optimization design problems can be divided into mathematical programming and meta-heuristic algorithms. Due to the difficulties encountered in mathematical programming (complex derivatives, sensitivity to initial values, and the large amount of enumeration memory required) [10] for complex problems, various kinds of meta-heuristic algorithms have been developed for optimum design of steel structures. The meta-heuristic algorithm selected for the solution of optimum discrete design of double layer grids is the Cuckoo Search algorithm. As stated before, two other powerful advanced hybrid algorithms, the HPSACO and the HBB-BC are also employed to gain a precise assessment and show the effectiveness of the CS. In the following subsections, the computational steps of the CS are briefly overviewed [4].

3.1. Cuckoo search algorithm

This algorithm is based on the obligate brood parasitic behavior of some Cuckoo species in combination with the Lévy flight behavior of some birds and fruit flies, which is recently developed by Yang [2]. These species lay their eggs in the nests of other host birds (almost other species) with amazing abilities such as selecting the recently spawned nests, and removing existing eggs that increase hatching probability of their eggs. On the other hand, some of the host birds are able to combat this parasites behavior of Cuckoos, and throw out the discovered alien eggs or build their new nests in new locations. This algorithm contains a population of nests or eggs. For simplicity, following representations is used; each egg in a nest represents a solution and a Cuckoo egg represents a new one. If the Cuckoo egg be very similar to the host's, then this Cuckoo's egg is less likely to be discovered, thus the fitness should be related to the difference in solutions. The aim is to employ the new and potentially better solutions (Cuckoos') to replace a not-so-good solution in the nests [11].

The Lévy flight is a random process in which a series of consecutive random steps perform with a power-law step-length distribution with a heavy tail. The generation of random numbers with Lévy flights includes two steps: choice of a random direction, and the generation of steps which obey the chosen Lévy distribution, while the generation of steps is quite tricky. There are a few ways to achieve this, but one of the most efficient and yet straightforward ways is to use the so-called Mantegna algorithm [2].

The original version of the CS [2] is sequential, and each iterations of the algorithm consists of two main steps, but another version of the CS which is supposed to be different and more efficient, is provided by Yang and Deb [11]. In this study the later version of the CS algorithm is utilized. The pseudo code of optimum design algorithm can be summarized as follows [4]:

3.1.1. Initialize the Cuckoo Search algorithm parameters

The CS parameters are set in the first step. These parameters are the number of nests (*n*), step size parameter (α), discovering probability (*pa*) and maximum number of grid analyses as the stopping criterion.

3.1.2. Generate initial nests or eggs of host birds

The initial locations of the nests are determined by the set of values assigned to each decision variable randomly as

$$nest_{i,j}^{(0)} = ROUND\left(x_{j,\min} + rand.(x_{j,\max} - x_{j,\min}) \right) \tag{7}$$

where $nest_{i,j}^{(0)}$ determines the initial value of the *j*th variable for the *i*th nest; $x_{j,min}$ and $x_{j,max}$ are the minimum and the maximum allowable values for the *j*th variable; rand is a random number in the interval [0, 1]. The rounding function is used because of the discrete nature of the problem.

3.1.3. Generate new Cuckoos by Lévy flights

In this step all of the nests except for the best so far, are replaced in order of quality by new Cuckoo eggs produced with Lévy flights from their positions as

$$nest_i^{(t+1)} = nest_i^{(t)} + a \; . \; S.(nest_i^{(t)} - nest_{best}^{(t)}).r \tag{8}$$

where $nest_i^t$ is the *i*th nest current position, α is the step size parameter; *S* is the Lévy flights vector as in Mantegna's algorithm ; *r* is a random number from a standard normal distribution and $nest_{best}$ is the position of the best nest so far.

3.1.4. Alien eggs discovery

The alien eggs discovery is preformed for all of eggs but in term of probability matrix for each component of each solution such as:

$$P_{ij} = \begin{cases} 1 & if \;\; rand < pa \\ 0 & if \;\; rand \geq pa \end{cases} \tag{9}$$

where *rand* is a random number in [0, 1] interval and P_{ij} is discovering probability for *j*th variable of *i*th nest. Existing eggs are replaced considering quality by newly generated ones from their current position by random walks with step size such as:

$$\begin{aligned} & S = rand\,.\left(nests\left(randperm(n),:\right) - nests\left(randperm(n),:\right)\right) \\ & nest^{t+1} = nest^{t} + S.*P \end{aligned} \tag{10}$$

where *randperm* is a random permutation function is used for different rows permutation applied on nests matrix and *P* is the probability matrix.

3.1.5. Termination criterion

The generating of new Cuckoos and discovering the alien eggs steps are performed alternatively until a termination criterion is satisfied. The maximum number of grid analyses is considered as algorithm's termination criterion.

4. STRUCTURAL MODELS

Two commonly used configurations for double layer grids considered in this study are two-way on two-way and diagonal on diagonal square grids. Two ranges of spans 15×15m and 40×40m with certain bays of equal length in two directions are considered as small and big size spans. Simply supported condition is employed for bottom layer at the corner nodes, and mid-edge at two parallel sides for the small and big span cases, respectively. The range of discrete depths from a certain interval with 0.5m increment is considered for each case to achieve the optimum depth.

Three element grouping patterns namely GP1, GP2 and GP3 are introduced for the purpose of practical fabrication and determining the grouping effects on the different systems. Considering different sections for elements at top layer, bottom layer and diagonal elements leads to the first grouping type which is only applied to the 15×15 span case with three design variables. In the second one, the elements at top layer, bottom layer, and diagonal elements are put into different groups in a diamond-like manner around central node. The GP3 grouping pattern is the same as the second one, but it is in a square form. The configuration, support locations and element grouping patterns of double layer grids are shown in Figure 4. Due to symmetry, only a quarter of the 15×15m span case is shown in this figure. The element grouping in the form of GP2 is depicted by black and light hatching.

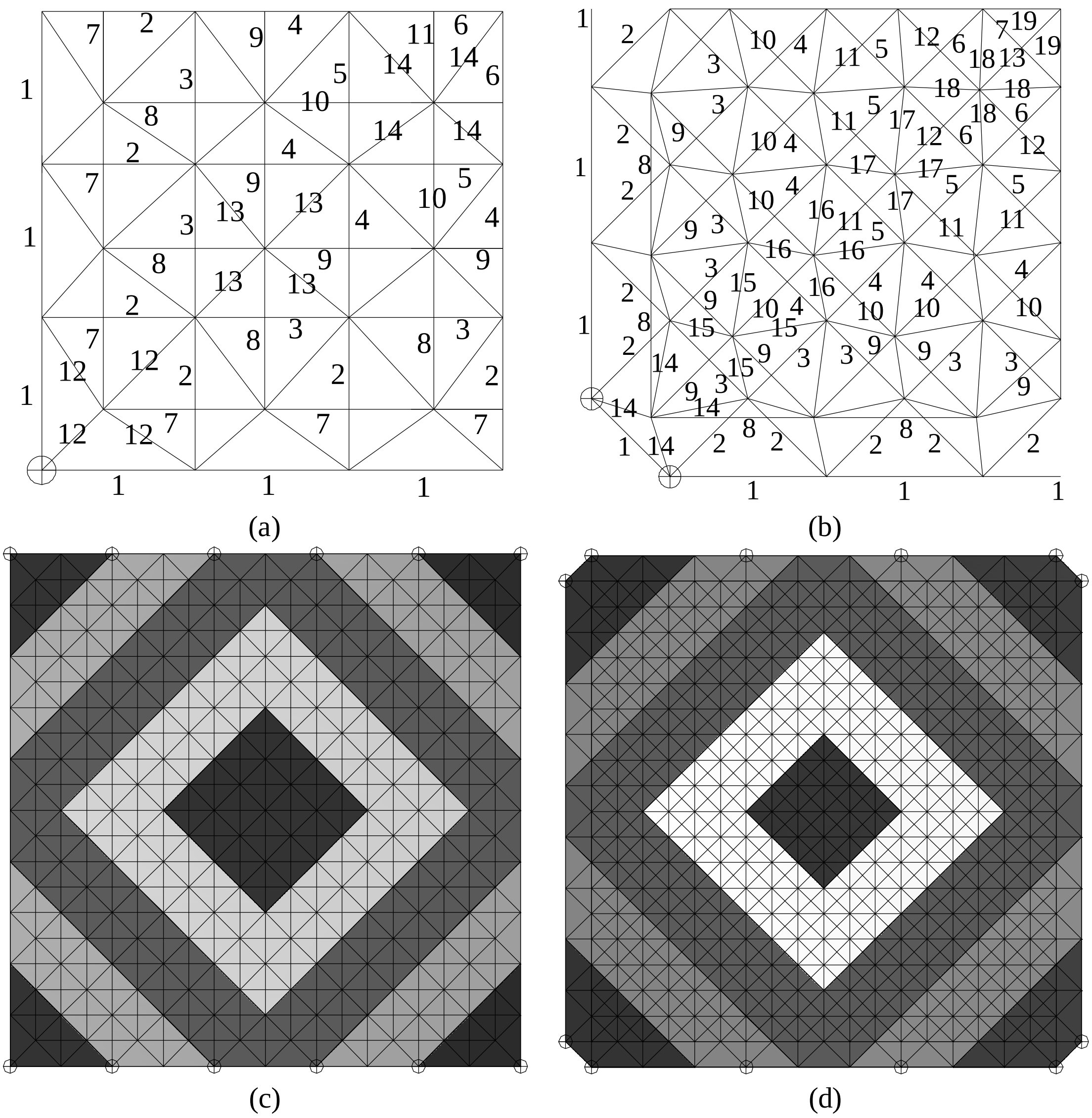


Figure 1. Topology, element grouping, support locations for different cases; (a) 15×15m two-way on two-way grid, (b) 15×15m diagonal on diagonal grid, (c) 40×40m two-way on two-way grid, (d) 40×40m diagonal on diagonal grid

5. THE NUMERICAL EXAMPLES

The double layer grids are assumed as pin-jointed, and top-layer joints are subjected to concentrated vertical loads transmitted from the uniformly distributed load of 200kg/m^2. Stress and slenderness constraints (Eqs. (3), (4) and (5)) according to AISC-LRFD provisions, and displacement limitations of span/600 were imposed on all nodes in vertical direction. The modulus of elasticity is taken as 205 kN/mm^2 and the yield stress of steel is taken as 248.2 MPa.

The parameters of the CS algorithm are considered as n=7, α=0.1 and pa=0.3 [4]. A population of 50 individuals is used for HPSACO and HBB-BC algorithms. In HPSACO algorithm, the value of constants C_1 and C_2 are set to 0.8 and passive congregation coefficient C_3 is taken as 0.6. The value of inertia weight $\omega(k)$ is altered linearly from maximum value to minimum value, which maximum and minimum of $\omega(k)$ is 0.9 and 0.4 in first iteration and last iteration, respectively [7]. The amount of step size η in ACO stage is recommended 0.01 [12]. *HMCR* is set to 0.95 and *PAR* is taken as 0.10. In HBB-BC algorithm coefficient of α_1=1, α_2=0.4 and α_3=0.8 are used [8]. The maximum numbers of grid analyses equaling to 4000, 6000 and 10,000 are considered as termination criteria for GP1, GP2 and GP3 grouping patterns in small span case and as 10,000 for both grouping patterns in big span case, respectively. The HBB-BC and the HPSACO algorithms are only used for obtained optimum depth by the CS algorithm and two adjacent depths. Considering the effect of the initial solution on the final results and the stochastic nature of the meta-heuristic algorithms, each case is independently solved for five times with random initial designs. Afterwards the best run is chosen for performance evaluation of each technique. The design algorithms are coded in MATLAB and structures are analyzed using the direct stiffness method [13].

Example 1: The 15×15 m double layer square grid

The 15×15 span case is studied as the small size of double layer grids. The first common type is the two-way on two-way grid which contains 85 nodes and 288 members, and the second one is the diagonal on diagonal grid with 145 nodes and 528 members. Each span contains 6 bays of equal length in both directions. Grouping patterns of GP1 and GP2 lead to 3 and 9 design variables for each type. The third grouping pattern yields 14 and 19 design variables for two-way on two-way and diagonal on diagonal grids, respectively. The range of discrete depths from [1, 4] interval with 0.5m increment is considered for each type to achieve the optimum depth.

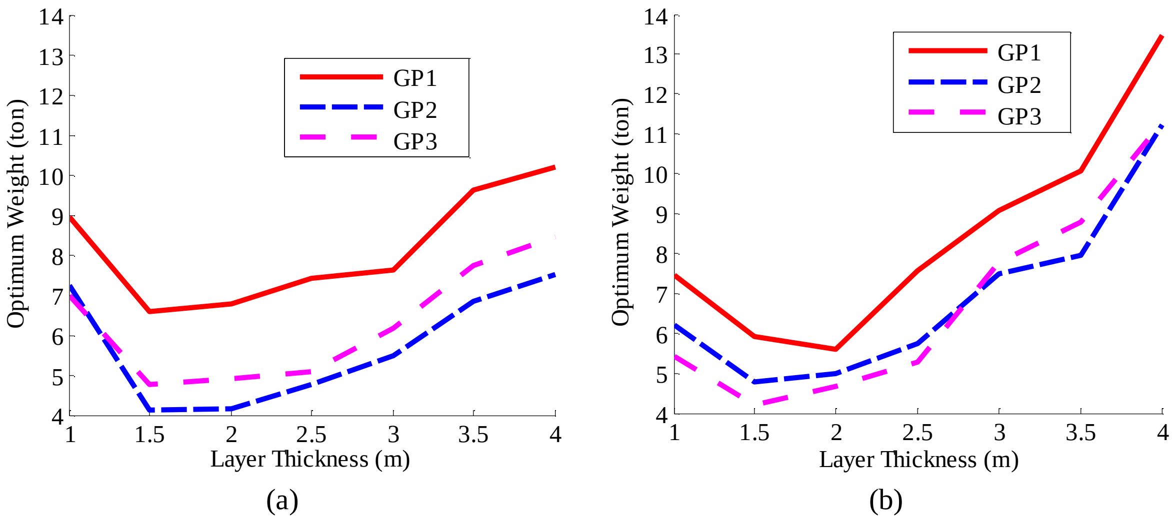


Figure 2. Obtained best results for the 15×15 span grid; (a) Two-way on two-way grid, (b) Diagonal on diagonal grid

Figure 2 shows the obtained optimum weight for various grouping patterns and depths of grids. As depicted, the optimum height of grid for both types equals to 1.5m and is independent of grouping pattern except for the case of GP1 in the second type which has the optimum depth of 2m. More importantly, the GP3 grouping type with more design variables results in heavier designs than that of GP2 grouping type for two-way on two-way grid. It is also worth mentioning that without considering number and complexity of joints, diagonal on diagonal grid for small spans is more suitable than two-way on two-way grid for GP1 grouping pattern containing three element groups often favored by engineers and architects because of its convenience and appealing features.

Table 2. Performance comparison for the 15×15 grid (kg)

Two-way on two-way grid									
	GP1			**GP2**			**GP3**		
	Height=1	Height=1.5	Height=2	Height=1	Height=1.5	Height=2	Height=1	Height=1.5	Height=2
CS	8931.492	6598.641	6768.999	7244.872	4127.063	4153.159	7006.406	4751.817	4920.276
HBB-BC	8931.492	6598.641	6768.999	7250.741	4371.717	4363.474	6999.783	5541.080	6550.501
HPSACO	8931.492	6598.641	6768.999	7235.120	4360.259	4533.831	7140.266	5056.159	5478.960
Diagonal on diagonal grid									
	GP1			**GP2**			**GP3**		
	Height=1	Height=1.5	Height=2	Height=1	Height=1.5	Height=2	Height=1	Height=1.5	Height=2
CS	7471.287	5927.232	6002.888	6319.767	4757.806	4978.462	5402.767	4180.124	4654.215
HBB-BC	7471.287	5927.232	5590.446	6209.962	5104.293	6099.758	5643.173	5749.242	5931.222
HPSACO	7471.287	5927.232	5590.446	6203.467	4873.785	5097.121	6157.951	5270.756	6373.701

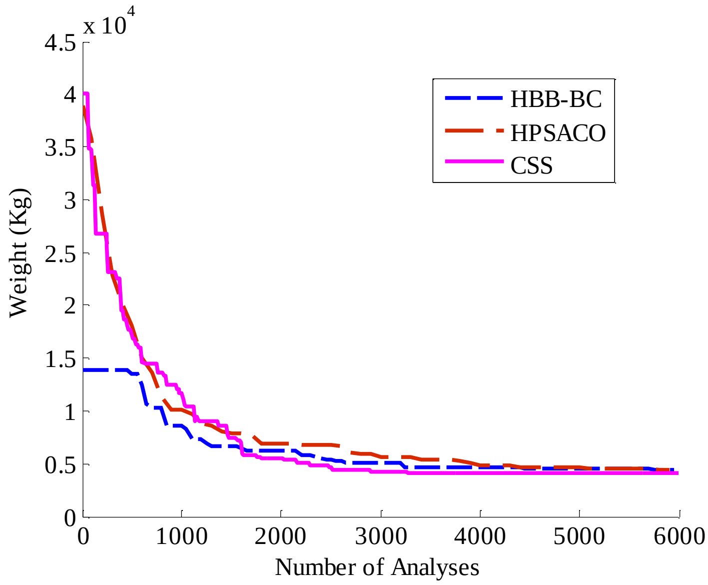


Figure 3. Convergence history obtained with meta-heuristic algorithms for the two-way on two-way 15×15 double layer grid (GP2 and depth= 1.5m)

Table 2 presents the performance of the CS and two other methods in which the best obtained weight is hatched for each case. It is apparent from the table that Cuckoo search has produced the lightest designs except for some cases with subtle differences. In the GP1 case in which the problem has only 3 design variables, all methods approximately yield the same design. For graphical comparison of algorithms, the convergence histories for the best result of 5 independent runs are shown in Figure 3 for the two-way on two-way, GP2 and depth = 1.5m. Table 3 shows the best solution vectors, the corresponding weights and the required number of analyses for three methods.

Table 3. Performance comparison for the 15×15 grid

Element group		**Optimal cross sectional area (in^2)**		
		CS	**HBB-BC**	**HPSACO**
1	A_1	1.48	1.48	1.48
2	A_2	1.7	1.7	3.17
3	A_3	2.23	2.23	2.23
4	A_4	0.669	0.669	0.799
5	A_5	2.23	2.23	2.25
6	A_6	0.799	0.799	0.799
7	A_7	0.639	0.669	0.639
8	A_8	1.48	2.66	1.48
9	A_9	0.669	0.669	0.669
Best weight (lb)		9098.62	9637.99	9612.73

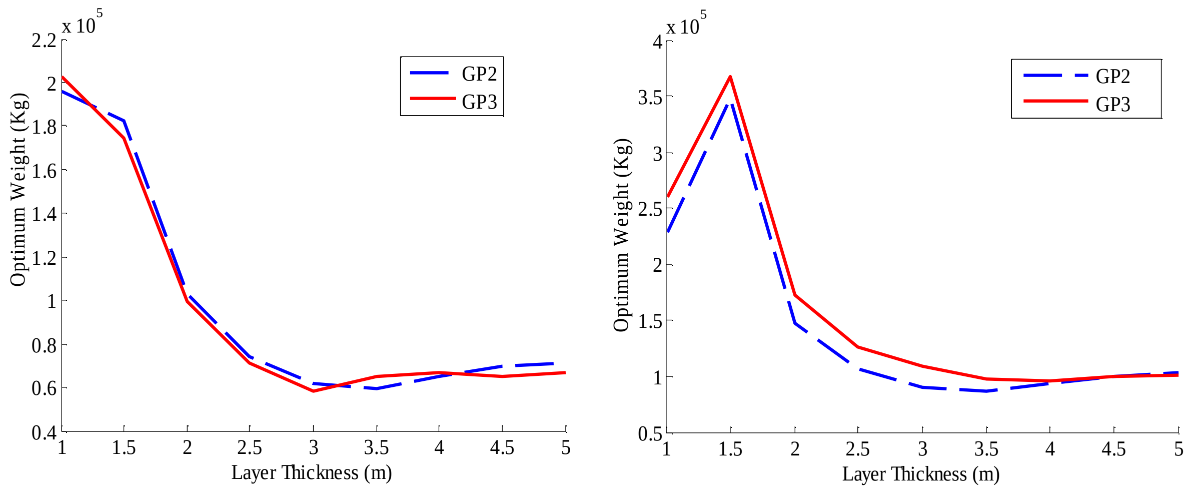
No. of analyses	4186	5950	6000

Example 2: The 40×40m double layer square grid

The 40×40m span case is considered as a big size of double layer grids. First common type is two-way on two-way grid which contains 221 nodes and 800 members. Second one is diagonal on diagonal grid with 401 nodes and 1520 members. Each span contains 10 bays of equal length in both directions. First grouping pattern is ignored in this case because of the size of structure. Grouping pattern of GP2 leads to 15 design variables for both types. Third grouping pattern leads to 24 and 31 design variables for two-way on two-way and diagonal on diagonal grids, respectively. The range of discrete depths from [1, 5] interval with 0.5m increment is considered for each type to achieve the optimum depth.

Figure 4 shows the obtained optimum weight for various grouping patterns and depths of grids. As depicted, the curves of different groups are approximately coincided with subtle differences. It should be noted that the GP2 is a more suitable way for grouping these two common types of big size double layer grids, because of the fewer number of groups. It is shown that optimum height of first type is equal to 3 and 3.5m and the second one equals 4 and 3.5m for GP2 and GP3, respectively. It can also be found that the slope of curves is smooth after the optimum height. In the other words, the optimum design is approximately independent of the depths bigger than 3m.

Table 4 in which the best obtained weight is hatched for each case presents the performance of algorithms. It can be realized that two-way on two-way type is more suitable for big span cases with the same number of span divisions (without considering the number and complexity of joints), in terms of the resulting low weight designs in a way that the obtained optimum designs are 38 and 33 percent lighter than those of diagonal on diagonal one for GP1 and GP2 cases, respectively. It is apparent from the table that Cuckoo Search has produced the lightest designs in all cases. The CS performs far better than the HBB-BC and the HPSACO algorithms in reaching low weight designs in the 40×40m span case which could be considered as approximately large-scale problems with more number of design variables and design constraints. To make the comparison of algorithms more tangible, the best obtained designs for different amount of depths are depicted as a bar-chart in Figure 5.

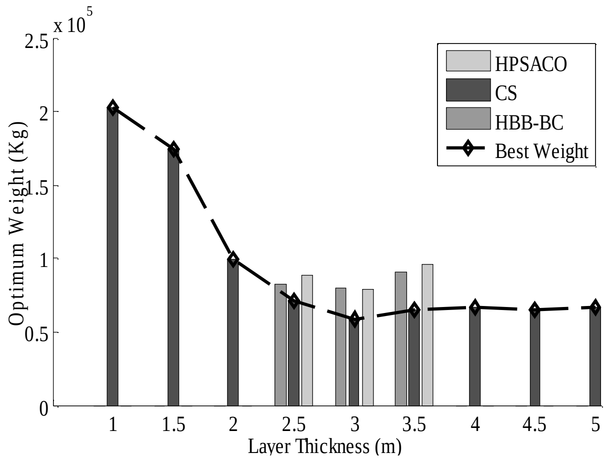


(a) (b)

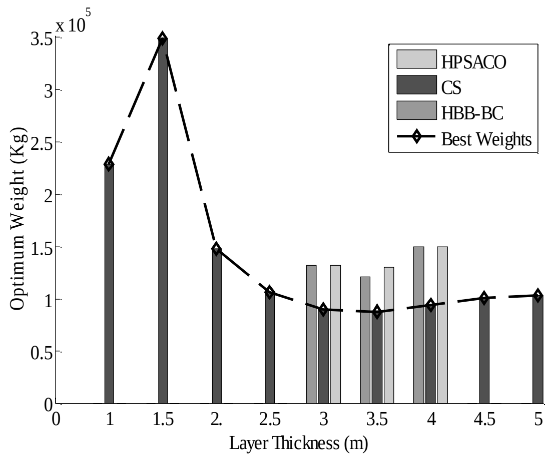
Figure 4. Obtained best results for the 40×40 span grid for grouping patterns; (a) Two-way on two-way grid, (b) Diagonal on diagonal grid

Table 4. Performance comparison for the 15×15 grid (kg)

	Two-way on two-way grid					
	GP1			GP2		
	Height=3	Height=3.5	Height=4	Height=2.5	Height=3	Height=3.5
CS	61564.751	59709.748	65272.400	71274.797	58474.360	64833.546
HBB-BC	80776.255	70748.113	92783.126	82342.363	79576.315	90213.388
HPSACO	82866.081	85118.951	102055.248	88623.380	79390.971	96137.848
	Diagonal on diagonal grid					
	GP1			**GP2**		
	Height=3.5	**Height=4**	**Height=4.5**	**Height=3**	**Height=3.5**	**Height=4**
CS	97965.173	95661.984	99775.377	89729.390	86883.682	93751.030
HBB-BC	113690.418	113987.966	135809.980	131973.303	120917.170	149910.633
HPSACO	133017.343	112800.347	124293.047	131363.973	129412.670	149261.498



(a)



(b)

Figure 5. Obtained best results for the 40×40m span grid by three algorithms; (a) Two-way on two-way grid, (b) Diagonal on diagonal grid

6. CONCLUDING REMARKS

The paper presents an economical comparison between tow-way on two-way and diagonal on diagonal square grids as common types of double layer grids. Two ranges of spans as small and big size of double layer square grids are considered with simply supported conditions at corners and semi full edge at two parallel sides for small and big cases, respectively. For precise comparison, an optimization study is carried out with the objective of minimizing self-weight based on CS, HBB-BC and HPSACO meta-heuristic algorithms. Grids were designed in accordance with AISC-LRFD specifications and displacement constraints. The results revealed in this study can be used for double layer grids in nearly similar conditions without considering the number and complexity of joints.

In small span cases both types conduct nearly the same-weight designs and it should be noted that diagonal on diagonal type has more connections and members. Using GP1 grouping pattern, which contains 3 design variables and has architectural and structural advantages for small cases (compared with other grouping patterns), leads to less consumption of materials in diagonal on diagonal system than that of two-way on two-way system. The optimum span-depth ratio for small span cases can be considered as 10 for both types.

In addition to fewer number of nodes in two-way on two-way grid, this type leads to lighter designs than diagonal on diagonal grid by approximately 35 percent for both grouping patterns in big span cases. In this case, a span-depth ratio between 8 and 13 can be considered acceptable, though the optimum value is approximately 13 for both groups.

The comparisons of the numerical results obtained by the CS with those by two others are carried out to demonstrate the robustness of the CS in achieving the best designs. Based on the study, it can be indicated that CS results in lighter designs especially in complicated cases.

REFERENCES

1. Chilton E. *Space Grid Structures*, Architectural Press, 2000.
2. Yang XS. *Nature-Inspired Metaheuristic Algorithms*. Luniver Press: UK, 2008.
3. Kaveh A, Bakhshpoori T. Optimum design of space trusses using Cuckoo Search algorithm with Lévy flights, *Iranian J Sci Technol* In press, 2012.
4. Kaveh A, Bakhshpoori T. Optimum design of steel frames using Cuckoo Search algorithm with Lévy flights, *Struct Des Tall Special Build* 2011. DOI: 10.1002/tal.754.
5. American Institute of Steel Construction (AISC). Manual of steel construction load resistance factor design. 2nd ed. Chicago: AISC 1994.
6. Kaveh A, Talatahari S. Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. *Comput Struct* 2009; **87**: 267-283.
7. Kaveh A, Talatahari S. A particle swarm ant colony optimization algorithm for truss structures with discrete variables. *J Construct Steel Res* 2009; **65**(89):1558-68.
8. Kaveh A, Talatahari S. Size optimization of space trusses using Big Bang-Big Crunch algorithm. *Comput Struct* 2009; **87**:1129-40.
9. Kaveh A, Talatahari S. Optimum design of skeletal structures using imperialist competitive algorithm. *Comput Struct* 2010; **88**:1220-29.
10. Lee KS, Geem ZW. A new structural optimization method based on the harmony search algorithm. *Comput Struct* 2004; **82**:781-98.
11. Yang XS, Deb S. Engineering Optimization by Cuckoo Search. *Int J Math Modelling Numer Optim* 2010; **1**:330-43.
12. Kaveh A, Talatahari S. A hybrid particle swarm and ant colony optimization for design of truss structures. *Asian J Civil Eng* 2008; **9**(4):329–48.
13. Kaveh A. *Optimal Structural Analysis*. 2nd ed. John Wiley, Chichister, UK, 2006.