Novel Metaheuristic Algorithms Applied to Optimization of Structures

LETÍCIA FLECK FADEL MIGUEL[1] and LEANDRO FLECK FADEL MIGUEL[2]
[1]Department of Mechanical Engineering and [2]Department of Civil Engineering
[1]Federal University of Rio Grande do Sul and [2]Federal University of Santa Catarina
[1]Av. Sarmento Leite 425, 2º andar, CEP 90050-170, Porto Alegre, RS, Brazil.
BRAZIL
letffm@ufrgs.br http://www.ufrgs.br

Abstract: - This paper aims at to evaluate the performance in engineering problems of two of the most recent metaheuristic algorithms developed in the last decade. The selected metaheuristic methods are Harmony Search (HS) and Firefly Algorithm (FA). Initially a brief description of both methods is presented and next they are applied in the field of classical structural optimization problems of truss structures. The effectiveness of Harmony Search and Firefly Algorithm is demonstrated through four benchmark structural optimization problems. The results show the ability of both metaheuristic algorithms to find optimal solutions for structural optimization problems in a relatively low computational cost.

Key-Words: - Harmony Search, Firefly Algorithm, Optimization, Engineering Design, Benchmark Problems, Truss Structures.

1 Introduction

Haftka and Gürdal [1] affirm that optimization is concerned with achieving the best outcome of a given operation while satisfying certain restrictions. So, optimization is present in many human activities, since the simple task of planning holidays up to complex engineering designs. Thus, researches of different parts of the world have been studding and developing optimization methods.

In Engineering, most design optimization problems are often nonlinear, involving different design variables under complex constraints. This nonlinearity may result in multimodal response landscape. Consequently, local search algorithms are not suitable, only global algorithms should be used to obtain optimal solutions (Arora, [2]; Deb, [3]; Yang, [4]). Nowadays, numerical simulations become an indispensable tool for solving such optimization problems with various efficient search algorithms. Modern metaheuristic algorithms have been developed with an aim to carry out global search.

Optimization of truss structures has been an active area of research in the field of search and optimal design. In the sizing optimization of trusses, cross sectional areas of members are considered as design variables and the coordinates of the nodes and connectivity among various members are considered to be fixed. The resulting optimization problem is a nonlinear programming problem.

Various methods have been developed to find optimal truss structures. Metaheuristic algorithms have been broadly used in these optimization problems, especially because they are not a gradient-based search, so they avoid most of the pitfalls of any gradient-based search algorithms. Thus, these algorithms have fewer mathematical requirements and they can be used to deal with complex objective functions. Many researches affirm that the metaheuristic algorithms are very efficient in the optimization area, and they attribute this efficiency to the fact that they imitate the best features in nature, especially the selection of the fittest in biological systems which have evolved by natural selection over millions of years.

Within this context, the present paper presents an analysis of two of the last proposed metaheuristic algorithms applied to optimization of truss structures. The two metaheuristic algorithms (Harmony Search and Firefly Algorithm) are applied to optimization of a number of truss structure design problems and the optimized trusses are compared with that reported in the literature.

2 Metaheuristic Algorithms

Many classical or conventional algorithms for structural optimization are deterministics and most of them used the gradient information, *i.e.*, they use the function values and their derivatives. They normally work extremely well for smooth unimodal problems; however, if there is some discontinuity in the objective function, they may not converge. Thus, in this kind of problem, a non-gradient algorithm is preferred. Non-gradient based or gradient-free algorithms do not use any derivative, but only the function values.

The stochastic or metaheuristic algorithms are a good alternative to deal with discontinuous objective function, since they are based on randomization and local search. Randomization provides a good way to move away from local search to the search on the global scale. Therefore, almost all metaheuristic algorithms intend to be suitable for global optimization.

According to Yang [4], two major components of any metaheuristic algorithms are: intensification and diversification, or exploitation and exploration. Diversification means to generate diverse solutions so as to explore the search space on the global scale, while intensification means to focus on the search in a local region by exploiting the information that a current good solution is found in this region. This is in combination with the selection of the best solutions. The selection of the best ensures that the solutions will converge to the optimality, while the diversification via randomization avoids the solutions being trapped at local optima and, at the same time, increases the diversity of the solutions. The good combination of these two major components will usually ensure that the global optimality is achievable.

Many different metaheuristic algorithms are in existence and new variants are continually being proposed. Yang [4] presents a very interesting history of metaheuristic algorithms, since the Second World War. In the last decade many metaheuristic algorithms were developed, for example, Harmony Search (Geem *et al.*, [5]), Honey Bee (Nakrani and Tovey, [6]), Bee Algorithm (Pham *et al.*, [7]), Artificial Bee Colony (Karaboga, [8]), Firefly Algorithm (Yang, [9]), Cuckoo Search (Yang and Deb, [10]), among others.

In this paper it will be studied two of these novel metaheuristic methods: Harmony Search (HS) and Firefly Algorithm (FA). Since HS and FA are metaheuristic algorithms, they are not a gradient-based search, so they avoid most of the pitfalls of any gradient-based search algorithms. Thus, these algorithms have fewer mathematical requirements and they can be used to deal with complex objective functions.

In the following, a brief review of the two optimization algorithms used in this paper will be presented.

2.1 Harmony Search (HS)

Harmony search (HS) is a recent metaheuristic optimization algorithm, which was developed by Geem *et al.* [5]. HS is a music-based metaheuristic optimization algorithm, which is inspired by the observation that the aim of music is to search for a perfect state of harmony. This harmony in music is analogous to find the optimality in an optimization process. The search process in optimization can be compared to a musician's improvisation process.

Geem *et al.* [5] observed that when musicians are improvising they used to (a) play famous piece of music (a series of pitches in harmony) from their memory; or (b) play something similar to a known piece (thus adjusting the pitch slightly); or (c) compose new or random notes. Thus, in Harmony Search Algorithm, these three options become (a) use of harmony memory; (b) pitch adjusting; and (c) randomization.

The use of harmony memory will ensure that the best harmonies will be carried over to the new harmony memory. It is reached through the use of a parameter called Harmony Memory Considering Rate (HMCR), $0 \leq HMCR \leq 1$. So, if this parameter is too low, only few best harmonies are selected and it may converge too slowly. On the other hand, if this parameter is extremely high (near 1), almost all the harmonies are used in the harmony memory, then other harmonies are not explored well, leading to potentially wrong solutions. Therefore, this parameter usually assumes values from 0.7 to 0.95 (Yang [9]).

The second component is the pitch adjustment determined by a pitch bandwidth range (bw) and a pitch adjusting rate (PAR). In HS pitch adjustment corresponds to generate a slightly different solution. The PAR is used to control the degree of the adjustment. Thus, a low PAR with a narrow bandwidth can slow down the convergence of HS because the limitation in the exploration of only a small subspace of the whole search space. On the other hand, a very high PAR with a wide bandwidth may cause the solution to scatter around some potential optima as in a random search. So, this parameter usually assumes values from 0.1 to 0.5 in most simulations (Yang, [9]).

The third component is the randomization, which is to increase the diversity of the solutions.

Although adjusting pitch has a similar role, but it is limited to certain local pitch adjustment and thus corresponds to a local search. The use of randomization can drive the system further to explore various diverse solutions so as to find the global optimality (Yang, [9]). The three components in HS can be summarized as the pseudo code shown in Figure 1 (Yang, [9]).

---

begin
  Objective function f(x), $x = (x_1, ..., x_p)^T$
  Generate initial harmonics (real number arrays)
  Define pitch adjusting rate and pitch limits
  Define harmony memory accepting rate
  while (*t < Max number of iterations*)
    Generate new harmonics by accepting bets harmonics
    Adjust pitch to get new harmonics (solutions)
    if (*rand > HMCR*), *choose an existing harmonic randomly*
    else if (*rand > PAR*), *adjust the pitch randomly within limits*
    else *generate new harmonics via randomization*
    end if
    Accept the new harmonics (solutions) if better
  end while
  Find the current best estimates
end

Figure 1: Pseudo code of Harmony Search (adapted from Yang, [9]).

Finally, according to Yang [9], HS could be more efficient than Genetic Algorithms (GA), for instance, because HS does not use binary encoding and decoding, but it does have multiple solution vectors. Therefore, HS is faster during each iteration. Besides, the implementation of HS algorithm is also easier. In addition, there is evidence to suggest that HS is less sensitive to the chosen parameters, which means that it is not necessary to fine-tune these parameters to get quality solutions.

Recently, Degertekin [11] and Degertekin *et al.* [12] realized a comparative study of HS with other optimization methods for optimum design of steel frame structures and concluded that the HS algorithm yielded lighter designs for the examples presented and in many cases it also required less computational effort for the presented examples.

Geem *et al.* [5] and Geem [13] present a detailed explanation about HS.

2.2 Firefly Algorithm (FA)

Firefly Algorithm (FA) is a very new metaheuristic optimization algorithm, which was developed by Yang [9] and it is inspired by the flashing behavior of fireflies.

According to Yang [4], there are three idealized rules in the FA optimization: (a) all fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex; (b) attractiveness is proportional to the their brightness, thus for any two flashing fireflies, the less brighter one will move towards the brighter one. The attractiveness is proportional to the brightness and they both decrease as their distance increases. If there is no brighter one than a particular firefly, it will move randomly; and (c) the brightness of a firefly is affected or determined by the landscape of the objective function. In other words, Yang [4] affirms that a firefly will be attracted to brighter or more attractive fireflies, and at the same time they will move randomly. This attractiveness is proportional to the brightness of the flashing light which will decrease with distance, therefore, the attractiveness will be evaluated in the eye of the beholders (other fireflies) and the decrease of light intensity is controlled by the light absorption coefficient γ which is in turn linked to a characteristic scale. Based on these three rules, the basic steps of the FA can be summarized as the pseudo code shown in Figure 2 (Yang, [4]).

---

begin
  Objective function f(x), $x = (x_1, ..., x_d)^T$
  Generate initial population of fireflies x_i $(i = 1, 2, ..., n)$
  Light intensity I_i *at* x_i *is determined by* $f(x_i)$
  Define light absorption coefficient γ
  while *(t < MaxGeneration)*
  for *i = 1 : n all n fireflies*
    for *j = 1 : d loop over all d dimensions*
    if $(I_i < I_j)$, *Move firefly i towards j;* **end if**
    Vary attractiveness with distance r via $\exp[-\gamma r]$
    Evaluate new solutions and update light intensity
    end for *j*
  end for *i*
  Rank the fireflies and find the current global best
  end while
  Post-process results and visualization
end

Figure 2: Pseudo code of Firefly Algorithm (adapted from Yang, [4]).

FA is a population-based algorithm, which may share many similarities with Particle Swarm Optimization (PSO). In fact, it has been proved by Yang [9] that when $\gamma \to \infty$, the FA will become an accelerated version of PSO, while $\gamma \to 0$, the FA reduces to a version of random search algorithms.

In the FA optimization, the diversification is represented by the random movement component, while the intensification is implicitly controlled by the attraction of different fireflies and the attractiveness strength β. Unlike other metaheuristics, the interaction between exploration and exploitation is intermingled in some way; this might be an important factor for its success in solving multiobjective and multimodal optimization problems (Yang, [14]).

Finally, according to Yang ([4], [9]), FA has the advantage that it can find the global optima as well as the local optima simultaneously and effectively. A further advantage of FA is that different fireflies will work almost independently, it is thus particular suitable for parallel implementation. It is even better than GA and PSO because fireflies aggregate more closely around each optimum. It can be expected that the interactions between different sub-regions are minimal in parallel implementation. Yang ([4], [9]) describes in detail the FA.

3 Benchmark Examples

Standard test problems are useful for the purpose of checking optimization algorithms. The four benchmark examples given in this section have been widely used for this purpose.

HS and FA were implemented in Matlab code, as well as, the subroutines of truss analysis developed by the authors.

HS and FA change the cross sectional areas (A_i), which are the design variables, looking for the minimum structural mass (M_{min}), subject to stresses (σ_i) and displacements (δ_j) constraints. Thus, the mathematical relationships that led to the numerical results are:

Minimize $$M_{\min} = \sum_{i=1}^{n} \rho_i \ell_i A_i \tag{1}$$

Subjected to
$$|\sigma_i| - \sigma_i^{\max} \le 0\,,\ i = 1, \ldots, n$$
$$|\delta_j| - \delta_j^{\max} \le 0\,,\ j = 1, \ldots, q$$
$$A_i^{\min} \le A_i \le A_i^{\max}\,,\ i = 1, \ldots, n$$

in which M_{min} is the minimum structural mass, n is the number of members in the current design, q is the number of nodes in the current design, ρ_i is the specific mass of the material of each bar, ℓ_i is the length of each bar, σ_i and $\sigma_i^{\max}$ are the stress and maximum allowed stress of the i^{th} bar, respectively, δ_j and $\delta_j^{\max}$ are the displacement and maximum allowed displacement at node j, respectively, and finally $A_i^{\min}$ and $A_i^{\max}$ are respectively the lower and upper bounds of the cross sectional area of the i^{th} bar.

3.1 10 Bar Plane Truss

The first example considered is the ten bar plane truss shown in Figure 3. The design variables are the cross sectional areas (treated as continuous design variables) of the ten elements.

The material properties, loading, allowable and optimum design are shown in Tables 1 to 4, respectively.

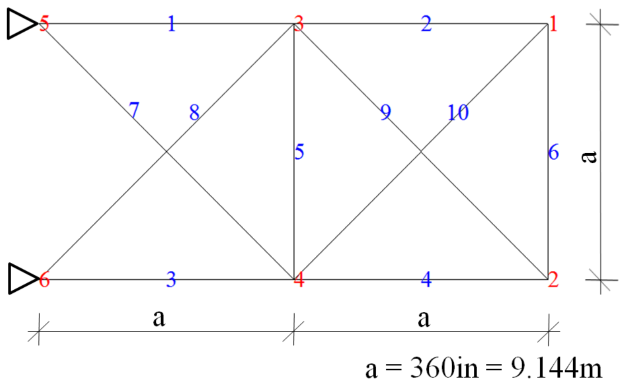

Figure 3: Ten bar truss.

Table 1: Material properties for the 10 bar truss.

Property	Value
Material	Aluminum
Young's modulus	10^7psi = 68.95GPa
Specific mass	0.1lb/in^3 = 2767.99kg/m^3

Table 2: Nodal load components for the 10 bar truss.

Node	x	y
2	0	-100kip = -444.82kN
4	0	-100kip = -444.82kN

Table 3: Allowable for the 10 bar truss.

Allowable	Value
Stress	±75ksi = ±517.11MPa for member 9 and ±25ksi = ±172.37MPa for all other members
Displacement	±2in = ±50.8mm in y direction for nodes 1 to 4
Range of the design variables	64.5mm^2 ≤ A_i ≤ 20000mm^2

Table 4: Optimum design for the 10 bar truss.

	(Areas in^2) Areas mm^2					
Member	Haftka and Gürdal [1]		Ghasemi *et al.* [15]		Present Paper - Harmony Search	Present Paper - Firefly Algorithm
1	(30.52)	19690.28	(25.73)	16599.97	19547.00	19869.00
2	(0.10)	64.52	(0.109)	70.32	505.93	497.82
3	(23.20)	14967.71	(24.85)	16032.23	15633.00	15258.00
4	(15.22)	9819.34	(16.35)	10548.37	10173.00	9938.30
5	(0.10)	64.52	(0.106)	68.39	65.88	64.50
6	(0.55)	354.84	(0.109)	70.32	384.34	495.34
7	(7.46)	4812.89	(8.70)	5612.89	5199.60	5406.70
8	(21.04)	13574.17	(21.41)	13812.88	13772.00	14030.00
9	(21.53)	13890.29	(22.30)	14387.07	13790.00	13585.00
10	(0.10)	64.52	(0.122)	78.71	665.69	685.91
Mass (lb) kg	(5060.85) **2295.56**		(5095.65) **2311.35**		**2368.62**	**2373.94**

As it can be observed in Table 4, the results obtained in the present paper, as much using HS as FA, are close to the values found in literature. The maximum difference was around only 3%, considering as reference the smallest value found in literature (2295.56kg).

It is important to point out that the results obtained by Ghasemi *et al.* [15] and presented in Table 4, slightly violated the displacement constraints in nodes 1 and 2.

In all simulations presented in this paper, as much using HS as FA, each optimum solution was obtained after 20,000 searches.

Other researchers also studied this structure and they arrived to the following optimum values: Venkayya [16] - 2306.5kg (5084.9lb), Gellatly and Berke [17] - 2318.8kg (5112.0lb), Schmit and Farshi [18] - 2308.3kg (5089.0lb), Schmit and Miura [19] - 2302.82kg (5076.85lb), Dobbs and Nelson [20] - 2304.2kg (5080.0lb), Li *et al.* [21] - 2295.59kg (5060.92lb), Kaveh and Talatahari [22] - 2293.62kg (5056.56lb), and Farshi and Alinia-Ziazi [23] - 2295.8kg (5061.4lb), among others. Recently, Togan *et al.* [24] obtained the minimum weight of this truss under uncertainties on the load, material and cross section areas with HS using reliability index and performance measure approaches, obtaining a minimum mass of 2700kg.

3.2 17 Bar Plane Truss

The second standard test problem is the seventeen bar plane truss shown in Figure 4. Again, the design variables are the cross sectional areas of the seventeen elements. The material properties, loading, allowable and optimum design are shown in Tables 5 to 8, respectively.

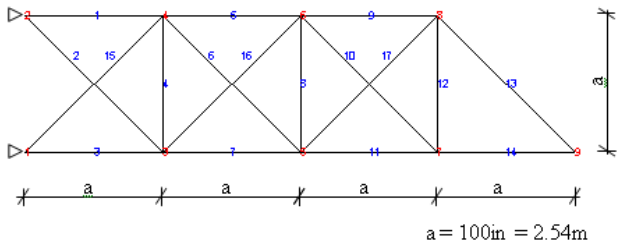

Figure 4: Seventeen bar truss.

Table 5: Material properties for the 17 bar truss.

Property	**Value**
Material	Steel
Young's modulus	$3{\times}10^7$psi = 206.84GPa
Specific mass	0.268lb/in^3 = 7418.21kg/m^3

Table 6: Nodal load components for the 17 bar truss.

Node	**x**	**y**
9	0	-100kip = -444.82kN

Table 7: Allowable for the 17 bar truss.

Allowable	**Value**
Stress	±50ksi = ±344.74MPa
Displacement	±2in = ±50.8mm in x and y direction for all nodes
Range of the design variables	64.5mm$^2 \leq A_i \leq$ 12900mm^2

Table 8: Optimum design for the 17 bar truss.

Members	Khot and Berke [25]		Adeli and Kumar [26]		Present Paper - Harmony Search	Present Paper - Firefly Algorithm
	(Areas in^2) Areas mm^2					
1	(15.93)	10277	(16.03)	10341	10444.00	10196.00
2	(0.10)	65	(0.11)	69	78.46	64.50
3	(12.06)	7780	(12.18)	7860	7890.60	7644.70
4	(0.10)	65	(0.11)	71	71.12	64.77
5	(8.06)	5199	(8.42)	5430	5157.80	5322.50
6	(5.56)	3587	(5.71)	3687	3399.50	3522.00
7	(11.93)	7697	(11.33)	7310	7588.80	7650.70
8	(0.10)	65	(0.11)	68	67.57	64.50
9	(7.94)	5125	(7.30)	4710	4983.80	5110.20
10	(0.10)	65	(0.11)	74	99.49	64.50
11	(4.05)	2616	(4.05)	2610	2775.20	2584.20
12	(0.10)	65	(0.10)	65	66.15	64.50
13	(5.65)	3645	(5.61)	3620	3694.60	3725.80
14	(4.00)	2580	(4.05)	2610	2495.40	2692.10
15	(5.56)	3585	(5.15)	3324	3591.90	3643.30
16	(0.10)	65	(0.11)	69	140.81	64.50
17	(5.58)	3599	(5.29)	3410	3650.30	3608.10
Mass (lb) kg	(2581.89) **1171.13**		(2594.42) **1176.81**		**1173.22**	**1171.47**

As it can be observed in Table 8, the results obtained in the present paper are very close to the values found in literature. The maximum difference was less than 0.18%, considering as reference the smallest value found in literature (1171.13kg). Again, in all simulations, each optimum solution was obtained after 20,000 searches.

3.3 25 Bar Space Truss

The third benchmark problem is the twenty five space truss shown in Figure 5. The truss is subjected to two distinct loading conditions and has eight independent design variables after linking, as indicated in Table 9. The material properties, loadings and allowable are shown in Tables 10 to 12, respectively, while Table 13 shows the optimum design.

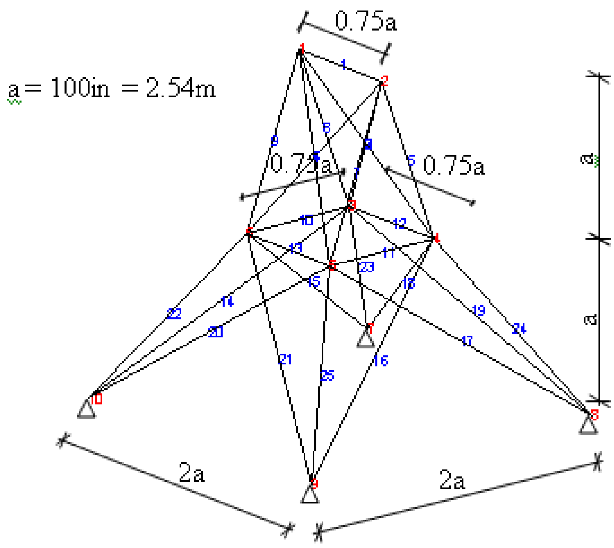

Figure 5: Twenty five bar truss.

Table 9: Member linking detail for the 25 bar truss.

Group Number	Members
1	1
2	2-5
3	6-9
4	10-11
5	12-13
6	14-17
7	18-21
8	22-25

Table 10: Material properties for the 25 bar truss.

Property	Value
Material	Aluminum
Young's modulus	10^7psi = 68.95GPa
Specific mass	0.1lb/in^3 = 2767.99kg/m^3

Table 11: Nodal load components for 25 bar truss.

Case	Node	x	y	z
1	1	1kip = 4.4482kN	10kip = 44.482kN	-5kip = -22.241kN
	2	0	10kip = 44.482kN	-5kip = -22.241kN
	3	0.5kip = 2.2241kN	0	0
	6	0.5kip = 2.2241kN	0	0
2	1	0	20kip = 88.964kN	-5kip = -22.241kN
	2	0	-20kip = -88.964kN	-5kip = -22.241kN

Table 12: Allowable for the 25 bar truss.

Allowable	Value
Tension stress for all members	40ksi = 275.79MPa
Compression stress for member 1	-35.092ksi = -241.95MPa
Compression stress for members 2-5	-11.590ksi = -79.91MPa
Compression stress for members 6-9	-17.305ksi = -119.31MPa
Compression stress for members 10-11	-35.092ksi = -241.95MPa
Compression stress for members 12-13	-35.092ksi = -241.95MPa
Compression stress for members 14-17	-6.759ksi = -46.60MPa
Compression stress for members 18-21	-6.959ksi = -47.98MPa
Compression stress for members 22-25	-11.082ksi = -76.41MPa
Displacement	±0.35in = ±8.89mm in x, y and z directions for nodes 1 and 2
Range of the design variables	6.45mm^2 $\leq A_i \leq$ 2000mm^2

Table 13: Optimum design for the 25 bar truss.

	(Areas in^2) Areas mm^2			
Members	Saka [27]	Farshi and Alinia-Ziazi [23]	Present Paper - Harmony Search	Present Paper - Firefly Algorithm
1	(0.010) 6.45	(0.010) 6.45	10.66	7.07
2-5	(2.058) 1327.74	(1.998) 1289.09	1233.10	1283.70
6-9	(2.988) 1927.74	(2.983) 1924.38	1926.90	1930.80
10-11	(0.010) 6.45	(0.010) 6.45	6.59	6.45
12-13	(0.010) 6.45	(0.010) 6.45	21.65	6.45
14-17	(0.696) 449.03	(0.684) 441.10	458.44	431.54
18-21	(1.670) 1077.42	(1.675) 1080.64	1132.30	1081.40
22-25	(2.592) 1672.25	(2.667) 1720.51	1687.10	1730.70
Mass (lb) kg	(545.23) **247.31**	(545.37) **247.38**	**247.84**	**247.31**

As it can be observed in Table 13, the results obtained in the present paper, as much using HS as FA, are practically identical to the values found in literature.

It is important to point out that the results obtained by Saka [27] and presented in Table 13, slightly violated the displacement constraints in nodes 1 and 2, for loading case 1 and they also slightly violated the compression stress constraints in bars 19 and 20 for loading case 2. In all simulations each optimum solution was obtained after 20,000 searches.

Other researchers also studied this structure and they arrived to the following optimum values: Venkayya [16] - 247.43kg (545.49lb), Templeman and Winterbottom [28] - 247.35kg (545.32lb), Schmit and Farshi [18] - 247.31kg (545.22lb), Schmit and Miura [19] - 247.28kg (545.17lb), and Adeli and Kamal [29] - 247.51kg (545.66lb), among others.

3.4 72 Bar Space Truss

The last benchmark example is the seventy two bar space truss shown in Figure 6. The truss is subjected to two distinct loading conditions and has sixteen independent design variables after linking, as indicated in Table 14. The material properties, loadings and allowable are shown in Tables 15 to 17, respectively, while Table 18 shows the optimum design.

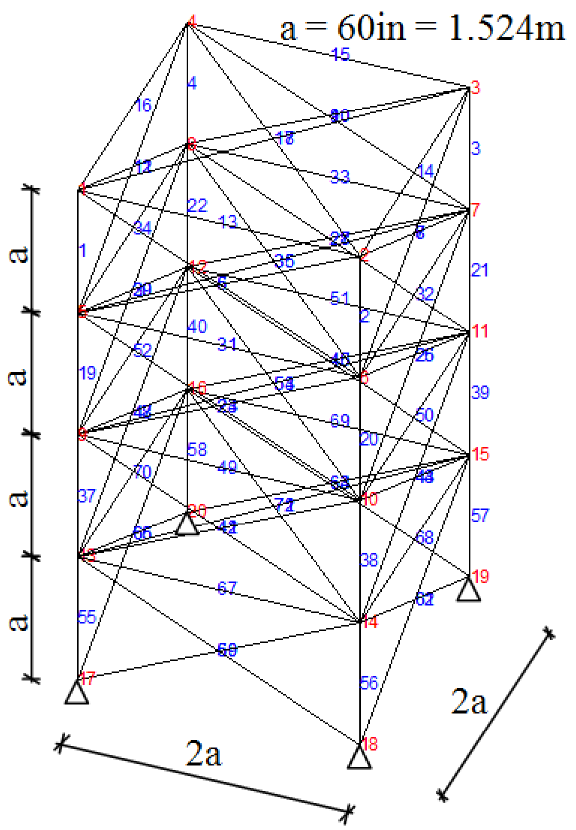

Figure 6: Seventy two bar truss.

Table 14: Member linking detail for the 72 bar truss.

Group Number	Members
1	1-4
2	5-12
3	13-16
4	17-18
5	19-22
6	23-30
7	31-34
8	35-36
9	37-40
10	41-48
11	49-52
12	53-54
13	55-58
14	59-66
15	67-70
16	71-72

Table 15: Material properties for the 72 bar truss.

Property	Value
Material	Aluminum
Young's modulus	10^7psi = 68.95GPa
Specific mass	0.1lb/in^3 = 2767.99kg/m^3

Table 16: Nodal load components for 72 bar truss.

Case	Node	x	y	z
1	1	5kip = 22.24kN	5kip = 22.24kN	-5kip = -22.24kN
2	1	0	0	-5kip = -22.24kN
	2	0	0	-5kip = -22.24kN
	3	0	0	-5kip = -22.24kN
	4	0	0	-5kip = -22.24kN

Table 17: Allowable for the 72 bar truss.

Allowable	Value
Stress	±25ksi = ±172.37MPa
Displacement	±0.25in = ±6.35mm in x, y and z directions for nodes 1 to 4
Range of the design variables	64.5mm$^2 \leq A_i \leq$ 1610mm^2

Table 18: Optimum design for the 72 bar truss.

Members	Haftka and Gürdal [1] (Areas in^2)	Haftka and Gürdal [1] Areas mm^2	Erbatur *et al.* [30] (Areas in^2)	Erbatur *et al.* [30] Areas mm^2	Present Paper - Harmony Search Areas mm^2	Present Paper - Firefly Algorithm Areas mm^2
1-4	(0.157)	101.35	(0.161)	103.87	103.84	101.02
5-12	(0.536)	345.55	(0.544)	350.97	334.02	342.74
13-16	(0.410)	264.45	(0.379)	244.52	266.19	265.24
17-18	(0.569)	367.10	(0.521)	336.13	395.29	323.46
19-22	(0.507)	326.90	(0.535)	345.16	327.35	415.19
23-30	(0.520)	335.48	(0.535)	345.16	319.72	328.15
31-34	(0.100)	64.52	(0.103)	66.45	65.30	64.52
35-36	(0.100)	64.52	(0.111)	71.61	72.76	121.75
37-40	(1.280)	825.80	(1.310)	845.16	774.66	826.82
41-48	(0.515)	332.13	(0.498)	321.29	316.02	323.15
49-52	(0.100)	64.52	(0.111)	70.97	64.77	64.59
53-54	(0.100)	64.52	(0.103)	66.45	65.15	65.93
55-58	(1.897)	1223.87	(1.910)	1232.26	1250.20	1294.50
59-66	(0.516)	332.77	(0.525)	338.71	385.77	328.46
67-70	(0.100)	64.52	(0.122)	78.71	64.63	64.50
71-72	(0.100)	64.52	(0.103)	66.45	64.89	64.50
Mass (lb) kg	(379.66)	**172.21**	(383.12)	**173.78**	**173.56**	**173.52**

As it can be observed in Table 18, the results obtained in the present paper, as much using HS as FA, are practically identical to the values found in literature. Other researchers also studied this structure and they arrived to the following optimum values: Venkayya [16] - 172.91kg (381.2lb), Gellatly and Berke [17] - 179.61kg (395.97lb), Schmit and Farshi [18] - 176.28kg (388.63lb), Schmit and Miura [19] - 172.20kg (379.64lb), Chao *et al.* [31] - 172.19kg (379.62lb), Perez and Behdinan [32] - 173.23kg (381.91lb), and Farshi and Alinia-Ziazi [23] - 172.21kg (379.65lb), among others. In all simulations each optimum solution was obtained after 20,000 searches.

4 Conclusions

This paper presented an analysis of two of the last proposed metaheuristic algorithms applied to engineering problems of optimization of truss structures. The two metaheuristic algorithms, HS and FA, were applied to optimization of a number of truss structure design problems and the optimized trusses are compared with that reported in the literature. The comparison of the results of these benchmark problems clearly illustrates the effectiveness and applicability of both metaheuristic algorithms. And it is important to point out that, in all presented problems, as much using HS as FA, none of the constraints were violated.

Many simulations were performed and the results showed that both methods are little sensitive to the chosen parameters, principally HS, which means that it is not necessary to fine-tune these parameters to get quality solutions, which is an advantage of both metaheuristic algorithms in relation to other optimization methods.

Ten independent runs of each problem were carried out for each algorithm, and these statistical results showed, as much using HS as FA, a little standard deviation from the mean value of the independent runs, showing that both methods are effective and reliable.

As much HS as FA could find the optimal solution in a relatively short computational time. For the same number of iterations HS found the optimal solution in a slightly shorter time than FA. However, for these tested examples, in most cases, FA found solutions slightly better than HS.

Acknowledgments:
The authors acknowledge the financial support of CNPq and CAPES.

References:

[1] Haftka, R. T., and Gürdal, Z., *Elements of structural optimization.* Kluwer Academic Publishers, 3rd Ed., 481p, 1992.

[2] Arora, J. S., *Introduction to optimum design.* McGraw-Hill, 625p, 1989.

[3] Deb, K., *Optimization for engineering design: algorithms and examples.* New Delhi: Prentice-Hall, 1995.

[4] Yang, X. S., *Nature-inspired metaheuristic algorithms.* Luniver Press, 2nd Ed., 148p, United Kingdom, 2010.

[5] Geem, Z. W., Kim, J. H., and Loganathan, G.V., A new heuristic optimization algorithm: Harmony Search. *Simulation*, 76(2), 60-68, 2001.

[6] Nakrani, S., and Tovey, C., On honey bees and dynamic server allocation in Internet hostubg centers, *Adaptive Behavior*, 12, 223-240, 2004.

[7] Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., and Zaidi, M., The bees algorithm. *Technical Note*, Manufacturing Engineering Center, Cardiff University, 2005.

[8] Karaboga, D. An idea based on honey bee swarm for numerical optimization. *Technical Report*, Erciyes University, 2005.

[9] Yang, X. S., *Nature-inspired metaheuristic algorithms.* Luniver Press, 1st Ed., 116p, United Kingdom, 2008.

[10] Yang, X. S., and Deb, S., Cuckoo search via Lévy flights. In: *Proc. of World Congress on Nature & Biologically Inspired Computing* (NaBic 2009), IEEE Publications, USA, 210-214, 2009.

[11] Degertekin, S. O., Harmony search algorithm for optimum design of steel frame structures: A comparative study with other optimization methods. *Structural Engineering and Mechanics*, 29, 4, 391-410, 2008.

[12] Degertekin, S. O., Hayalioglu, M. S., and Gorgun, H., Optimum design of geometrically non-linear steel frames with semi-rigid connections using a harmony search algorithm. *Steel and Composite Structures*, 9, 6, 535-555, 2009.

[13] Geem, Z. W., Global optimization using harmony search: theoretical foundations and applications. A. Abraham *et al.* (Eds.): *Foundations of Comput. Intel.* Vol. 3, SCI 203, pp. 57-73, Springer-Verlag Berlin Heidelberg, 2009.

[14] Yang, X. S., Harmony Search as a metaheuristic algorithm. in: *Music-Inspired Harmony Search Algorithm: Theory and Applications* (Editor Z. W. Geem), Studies in Computational Intelligence, Springer Berlin, vol. 191, pp. 1-14, 2009.

[15] Ghasemi, M. R., Hinton, E., and Wood, R. D., Optimization of trusses using genetic algorithms for discrete and continuous variables. *Engineering Computations*, 16(3), 272-301, 1999.

[16] Venkayya, V. B., Design of optimum structures. *Computers and Structures*, 1, 265-309, 1971.

[17] Gellatly, R. A., and Berke, L. Optimal structural design. *AFFDLTR-70-165*, Air Force Flight Dynamics Lab., Wright-Patterson AFB, OH, 1971.

[18] Schmit, L. A., and Farshi, B., Some approximation concepts for structural synthesis. *AIAA Journal*, 12, 692-699, 1974.

[19] Schmit, L. A., and Miura, H., Approximation concepts for efficient structural synthesis. *NASA-CR-2552*, NASA, Washington, DC, 1976.

[20] Dobbs, M. W., and Nelson, R. B. Application of optimality criteria to automated structural design. *AIAA Journal*, 14, 1436-1443, 1976.

[21] Li, L. J., Huang, Z. B., Liu, F., and Wu, Q. H., A heuristic particle swarm optimizer for optimization of pin connected structures. *Computers and Structures*, 85, 340-349, 2007.

[22] Kaveh, A. and Talatahari, S., Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. *Computers and Structures*, 87, 267-283, 2009.

[23] Farshi, B., and Alinia-Ziazi, A., Sizing optimization of truss structures by method of centers and force formulation. *International Journal of Solids and Structures*, 47, 2508-2524, 2010.

[24] Togan, V., Daloglu, A. T., and Karadeniz, H., Optimization of trusses under uncertainties with harmony search. *Structural Engineering and Mechanics*, 37, 5, 543-560, 2011.

[25] Khot, N. S., and Berke, L., Structural optimization using optimality criteria methods. *New directions in optimum structural design*, E. Atrek, R. H. Gallagher, K. M. Ragsdell, and O. C Zienkiewicz. John Wiley & Sons, New York, 1984.

[26] Adeli, H., and Kumar, S., Distributed genetic algorithm for structural optimization. *Journal of Aerospace Engineering*, ASCE, 8(3), 156-163, 1995.

[27] Saka, M. P., Optimum design of pin-jointed steel structures with practical applications.

Journal of Structural Engineering, ASCE, 116, 2599-2620, 1990.

[28] Templeman, A. B., and Winterbottom, S. K., Structural design by geometric programming. Second symposium on structural optimization, *AGARD Conference*, Preprint-123, Milan, 1973.

[29] Adeli, H., and Kamal, O., Efficient optimization of space trusses. *Computers and Structures*, 24, 501-511, 1986.

[30] Erbatur, F., Hasançebi, O., Tütüncü, I., and Kiliç, H., Optimal design of planar and space structures with genetic algorithms. *Computers and Structures*, 75, 209-224, 2000.

[31] Chao, N., Fenves, S., and Westerberg, A., Application of reduced quadratic programming technique to optimal structural design. In: Atrex, E., Gallagher, R. H., Radgsdell, K. M., Zienkiewicz, O. C. (Eds.), *New Directions in Optimum Structural Design.* Wiley, New York, 1984.

[32] Perez, R. E., and Behdinan, K., Particle swarm approach for structural design optimization. *Computers and Structures*, 85, 1579-1588, 2007.