

A Hybrid Flower Pollination Algorithm for Engineering Optimization Problems

Mohamed Abdel-Baset*
Faculty of Computers & Informatics
Zagazig University, Egypt

Ibrahim Hezam
Department of computer, Faculty of Education, Ibb
University, Ibb city, Yemen

ABSTRACT

Flower pollination algorithm (FP) is a new nature-inspired algorithm, based on the characteristics of flowering plants. Combining with the features of flower pollination algorithm, an improved simulated annealing algorithm is proposed in this paper (FPSA). It can improve the speed of annealing. The initial state of simulated annealing and new solutions are generated by flower pollination. Therefore, it has the advantage of high quality and efficiency. The method combines the standard flower pollination algorithm (FP) with simulated annealing to enhance the search performance and speeds up the global convergence rate. Structural engineering optimization problems are presented to demonstrate the effectiveness and robustness of the proposed algorithm. The experimental results showed that the accuracy of finding the best solution and convergence speed performance of the proposed algorithm is competitive to those achieved by the existing algorithms.

Keywords

Flower pollination; simulated annealing algorithm; constrained optimization problems; engineering optimization problems; global optimization.

1. INTRODUCTION

Most real-world engineering optimization problems in structural engineering are very complex, highly nonlinear, involving many different design variables under complex constraints. These constraints can be written either as simple bounds such as the ranges of material properties or as nonlinear relationships including maximum stress, maximum deflection, minimum load capacity, and geometrical configuration[1]. Such nonlinearity often results in multimodal response landscape. However, they are very complex in nature and quite difficult to solve. If there is more than one local optimum in the problem, subsequently, the gradient search may become unstable when the objective function and constraints have multiple or sharp peaks. Furthermore, local search algorithms such as hill-climbing and Nelder-Mead downhill simplex methods are not suitable; only global algorithms should be used so as to obtain optimal solutions.

Global optimization is an important task in most scientific and engineering problems. In global optimization problem, it is difficult to obtain an optimal solution due to time complexity. Accordingly metaheuristics became important in such situations for their independence on a specific problem. Metaheuristics are optimization approaches which make use of the best solution improved iteratively to the next search. For example, finding optimal solutions for nonlinear, non-differentiable fractional objective functions is very difficult to deal with. The complexity of these problems makes it impossible to search for all possible optimal exact solutions. So is the search for any solution near the optimal solution.

The metaheuristic optimization is the best for finding near optimal. Recently growing popularity in the hybridization of different algorithmic concepts has been to obtain better performing systems that exploit and combine the advantages of the individual pure strategies, that is, hybrids are believed to benefit from synergy. In fact, choosing an adequate combination of multiple algorithmic concepts is often the key to achieving top performance in solving many hard optimization problems. [1] combined two nature inspired algorithms and introduced the CS/PSO algorithm. Cuckoo birds are aware of each other positions and make use of swarm intelligence in PSO in order to reach for better solutions. [2] combined the differential evolution (DE) and cuckoo search (CS) algorithm to solve the uninhabited combat air vehicle UCAV path planning problem. DE is applied to optimize the process of selecting cuckoo of the CS model during the process of cuckoo in nest updating. [3] proposed hybrid optimization algorithm of PSO and CS. By CS-PSO, the search area of PSO was extended, and the defect of PSO is easily fallen into point of local extremum that was improved. [4] proposed a hybrid algorithm which combines the merits of Ant Colony Optimization (ACO) and Cuckoo Search for Job scheduling. The major problem in the ACO is that, the ant will walk through the path where the chemical substances called pheromone is deposited. This acts as if it lures the artificial ants[4]. Cuckoo search can perform the local search more efficiently and there is only a single parameter apart from the population size. It minimizes the makes pan and the scheduling can be used in scientific computing and high power computing. [5] introduced a modify firefly algorithm and used this algorithm with cellular learning automata. [6] combines the standard Firefly Algorithm (FA) with the evolutionary operations of Differential Evolution (DE) method to improve the searching accuracy and information sharing among the fireflies. [7] proposed ant colony optimization (ACO) and firefly algorithm (FFA) algorithm for constrained optimization problems, The methodology of the proposed algorithm is introduced based on a parallel mechanism of ACO and FFA for updating the solutions of ACO-FFA. [8] introduced a new hybrid swarm intelligence algorithm that encompasses the feature of three major swarm algorithms. It combined the fast convergence of the Cuckoo Search (CS), the dynamic root change of the Firefly Algorithm (FA), and the continuous position update of the Particle Swarm Optimization (PSO). [9] presented the evolutionary hybrid genetic-firefly algorithm for the optimization of complex problems and to search global solution more precisely.

The purpose of this paper is to combine the search features of flower pollination algorithm and simulated annealing algorithm. The initial state of simulated annealing and new solutions are generated by flower pollination. To enhance the search performance the global convergence rate.

This paper is organized as follows: after introduction, the original Flower pollination algorithm is briefly introduced.

Section 3 introduces the simulated annealing algorithm. In section 4, the proposed algorithm is described, while the results are discussed in section 5. Finally, conclusions are presented in section 6.

2. THE ORIGINAL FLOWER POLLINATION ALGORITHM

Flower Pollination Algorithm (FPA) was founded by Yang in the year 2012. Inspired by the flow pollination process of flowering plants are the following rules:

Rule 1: Biotic and cross-pollination can be considered as a process of global pollination process, and pollen-carrying pollinators move in a way that obeys Le'vy flights.

Rule 2: For local pollination, a biotic and self-pollination are used.

Rule 3: Pollinators such as insects can develop flower constancy, which is equivalent to a reproduction probability that is proportional to the similarity of two flowers involved.

Rule 4: The interaction or switching of local pollination and global pollination can be controlled by a switch probability $p \in [0,1]$, with a slight bias toward local pollination .

In order to formulate updating formulas, we have to convert the aforementioned rules into updating equations. For example, in the global pollination step, flower pollen gametes are carried by pollinators such as insects, and pollen can travel over a long distance because insects can often fly and move in a much longer range[17].Therefore, Rule 1 and flower constancy can be represented mathematically as:

$$x_i^{t+1} = x_i^t + \gamma L(\lambda)(x_i^t - B) \quad (1)$$

Where x_i^t is the pollen i or solution vector x_i at iteration t , and B is the current best solution found among all solutions at the current generation/iteration. Here γ is a scaling factor to control the step size. In addition, $L(\lambda)$ is the parameter that corresponds to the strength of the pollination, which essentially is also the step size. Since insects may move over a long distance with various distance steps, we can use a Le'vy flight to imitate this characteristic efficiently. That is, we draw $L > 0$ from a Levy distribution:

$$L \sim \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda / 2)}{\pi} \frac{1}{S^{1+\lambda}}, (S \gg S_0 > 0) \quad (2)$$

Here, $\Gamma(\lambda)$ is the standard gamma function, and this distribution is valid for large steps $s > 0$.

Then, to model the local pollination, both Rule 2 and Rule 3 can be represented as:

$$x_i^{t+1} = x_i^t + U(x_j^t - x_k^t) \quad (3)$$

Where x_j^t and x_k^t are pollen from different flowers of the same plant species. This essentially imitates the flower constancy in a limited neighborhood. Mathematically, if x_j^t and x_k^t comes from the same species or selected from the same population, this equivalently becomes a local random walk if we draw U from a uniform distribution in $[0, 1]$.Though Flower pollination activities can occur at all scales, both local and global, adjacent flower patches or flowers in the not-so-far-away neighborhood are more likely to be

pollinated by local flower pollen than those faraway. In order to imitate this, we can effectively use the switch probability like in Rule 4 or the proximity probability p to switch between common global pollination to intensive local pollination. To begin with, we can use a naive value of $p = 0.5$ as an initially value.

A preliminary parametric showed that $p = 0.8$ might work better for most applications[10-11].

The basic steps of FPA can be summarized as the pseudo-code shown in Fig. 1.

Flower pollination algorithm

Define Objective function $f(x)$, $x = (x1, x2, \dots, xd)$
Initialize a population of n flowers/pollen gametes with random solutions
Find the best solution B in the initial population
Define a switch probability $p \in [0, 1]$
while ($t < \text{MaxGeneration}$)
 for $i = 1 : n$ (all n flowers in the population)
 if $\text{rand} < p$,
 Draw a (d -dimensional) step vector L which obeys a Lévy distribution
 Global pollination via $x_i^{t+1} = x_i^t + L(B - x_i^t)$
 else
 Draw U from a uniform distribution in $[0,1]$
 Do local pollination via
 $x_i^{t+1} = x_i^t + U(x_j^t - x_k^t)$
 end if
 Evaluate new solutions
 If new solutions are better, update them in the population
 end for
 Find the current best solution B
end while
Output the best solution found

Fig. 1 Pseudo code of the Flower pollination algorithm

3. SIMULATED ANNEALING ALGORITHM

Simulated annealing (SA) is one of the simplest and most popular heuristic algorithms [12]. SA is a global search algorithm, based on annealing process of metal processing. It has been proved that it can have global convergence, though the convergence rate can be very slow. It is based on Monte Carlo iterative solution strategy. Its main advantage is that it not only accepts better solution than the current state, but also it can jump out of local minimum. The basic steps of SA can be summarized as the pseudo-code shown in Fig.2

Simulated Annealing Algorithm

Initialize the temperature T_0 and the solutions x_0
Set the final temperature T_f and the maximum number of iterations N
Define cooling table $T \rightarrow \alpha T$; ($0 < \alpha < 1$)
While ($T > T_f$ and $t < N$)
 Generate new solutions randomly $x_{t+1} = x_t + \epsilon$
 Calculate $\Delta f = f_{t+1}(x_{t+1}) - f_t(x_t)$
 Accept the new solution when it is better

If (the new solution is not accepted)
Generate a random number r
If ($p = \exp[-\Delta f/T] > r$)
Accept
End if
Update the optimal solution x^* and the optimal value f^*
 $t=t+1$
End while

Fig. 2. Pseudo code of the Simulated Annealing algorithm

There are many studies to combine simulated annealing and other optimization algorithms to produce hybrid algorithms [13–15], such combination may have some advantages, which deserves further investigation. In fact, many nature-inspired algorithms have become very popular, due to their simplicity, flexibility and efficiency [16]. Therefore, this paper is the first attempt to combine simulated annealing with the standard flower pollination algorithm to further enhance its performance.

4. THE PROPOSED ALGORITHM (SAFA) FOR SOLVING ENGINEERING OPTIMIZATION PROBLEMS

In the proposed algorithm, the initial solution is randomly generated in the standard simulated annealing algorithm. Therefore, the size of the solution is uneven. This feature will affect the effectiveness of algorithm. The defect can be avoided by using flower pollination algorithm to create the initial solution. In order to enhance the searching efficiency, the new solution is generated by FP algorithm. The worst solution is replaced by the new solution. The advantage of the searching process lies in retaining the intermediate optimal solution and updating on time. Finally, annealing process is executed once again based on the final optimal solution.

The steps of the proposed algorithm for solving global optimization problems are as follows:

Step 1: The Initialization of annealing temperature is T . FPSA uses the FP algorithm to create initial solution library MS . The formula of generating initial solution is

$$x = LB + r * (UB - LB) \quad (4)$$

where UB is the maximum value of solution, LB is the minimum value of solution, r is uniformly distributed random number in $(0,1)$. The iterations of each T is L .

Step 2: Disturbing and receiving processes are repeatedly executed L times under temperature T to determine whether the sample stability criterion is reached. If the criterion is not satisfied step3 –step4 will be executed. Otherwise the step5 will be executed to lower the temperature by using exponential function.

Step 3: The new solution is generated by FP algorithm.

Step 4: Difference value ΔE between the objective function values E' of new solution and the smallest solution E is calculated. If $\Delta E < 0$ then the new solution is accepted, otherwise the new solution is accepted with probability

$\min\{1, \exp[-\Delta E / t]\}$. Then the MS is updated and the worst solution in the library is replaced.

Step 5: Annealing. $T_{k+1} = 0.96 * T_k$, $k = k+1$. If the convergence criterion is satisfied, then the annealing process is executed based on the final solution once again and output terminal result else go to step2.

4.1 Handling Constraints

The feasible-based mechanism proposed by Deb [17] is used to handle the constraints problem and select the best individuals from one generation according to the following three rules:

Rule 1: Between two feasible solutions, the one with the higher fitness value is preferred.

Rule 2: Any feasible solution is preferred to any infeasible solution.

Rule 3: If both solutions are infeasible, the one with the lowest sum of constraint violation is preferred. This sum is calculated as:

$$\sum_{i=1}^n \max(0, g(\bar{x})) + \sum_{j=1}^p \max(0, |h_j(\bar{x}) - \varepsilon|) \quad (5)$$

5. NUMERICAL RESULTS

Most real-world engineering optimization problems are nonlinear with complex constraints. In some cases, the optimal solutions of interest do not even exist. In order to evaluate the performance of FPSA, it is tested against the following well-known benchmark design problems. In this section, we will carry out numerical simulation based on some well-known constrained engineering optimization problems to investigate the performances of the proposed algorithm. In addition, we will also compare the performance of the proposed algorithm with CS, FA, and PSO algorithms. The algorithms have been implemented by MATLAB R2011 on core (TM) i3, 2.27 GHz processor. Where the simulation parameter settings results of CS, FA, and PSO algorithms are shown in table 1.

Table 1. Parameters of CS, FA, and PSO

CS	Number of nests $n=50$, discovery rate of alien eggs/solutions $pa=0.25$;
FA	Population size : 50, α (randomness): 0.25, minimum value of β : 0.20, γ (absorption): 1.0
PSO	Population size of 50, the inertia weight W , set to change from 0.9 (w_{max}) to 0.4 (w_{min}) over the iterations. Set weighting coefficients, $c1 = 0.12$ and $c2 = 1.2$.

5.1 Illustrative Example

This simple example is a global optimization problem named Booth's function to show efficiency the proposed algorithm. Booth's function have an exact minimum of 0 at (1,3).

$$f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$$

When applying the FPSA to solve the above problem, the optimal basic feasible solution at (1.0000014, 2.9999998), corresponding to $f(x,y) = 2.00000E-013$ was obtained.

5.1.1 The Welded Beam design

The welded beam design problem often used as a standard test problem for constrained design optimization [18–21]. The objective is to minimize the overall fabrication cost $f(x)$ of the beam. Subject to constraints on shear stress (τ), bending stress (σ) in the beam, buckling load on the bar (P_b), end deflection

of the beam (δ), and side constraints. There are four design variables as shown in Fig. 3: the width $h(x_1)$ and length $l(x_2)$ of the welded area, the depth $t(x_3)$, and thickness $b(x_4)$ of the main beam. The mathematical formulation of the optimization problem can be stated as follows:

Minimize

$$f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$$

subject to

$$\begin{cases} g_1(x) = \tau(x) - 13.600 \leq 0, \\ g_2(x) = \sigma(x) - 30.000 \leq 0, \\ g_3(x) = x_1 - x_4 \leq 0, \\ g_4(x) = 6.000 - P_c(x) \leq 0, \\ g_5(x) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 0.5 \leq 0, \\ g_6(x) = \delta(x) - 0.25 \leq 0, \end{cases}$$

$$\tau(x) = \sqrt{(\tau')^2 + (\tau'')^2 + \frac{2\tau'\tau''x_2}{2R}}$$

$$\tau' = P / (\sqrt{2}x_1x_2)$$

$$\tau'' = \frac{MP}{J}, \quad M = P \left(L + \frac{x_2}{2} \right)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2} \right)^2}$$

$$J = 2 \left\{ \sqrt{2}x_1x_2 \left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2} \right)^2 \right] \right\}$$

$$\sigma(x) = \frac{6PL}{x_4x_3^2}, \quad \delta(x) = \frac{4PL^3}{Ex_4x_3^3}$$

$$P_c(x) = \frac{4.013E \sqrt{\frac{x_2^2x_4^6}{36}}}{L^2} \left(1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right)$$

$$P = 6000lb, \quad L = 14in, \quad E = 30 \times 10^6 \text{ psi},$$

$$G = 12 \times 10^6 \text{ psi}, \quad x_1 \in [0.125, 10], \quad x_2, x_3, x_4 \in [0.1, 10]$$

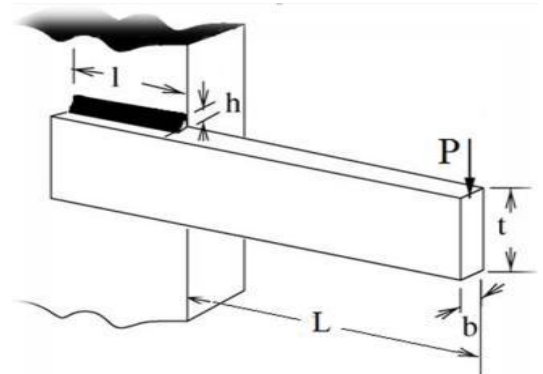


Fig. 3. Welded beam

The comparisons of the statistical results obtained by FPSA, CS, FA, and PSO algorithms are presented as shown in Table 2. Accuracy performance is measured in terms of the best, mean, and standard deviation values of the solutions obtained by 20 independent runs. The best solution was obtained using FPSA with an objective function value of 1.72485205903234. Fig. 4: illustrates the function values with respect to the number of iterations for the welded beam design problem.

Table 2. Comparison of statistical results of welded beam design

	FPSA	CS	FA	PSO
Best	1.72485205903234	1.72485229196730	1.724853054825791	1.72485307029703
Mean	1.724853099019661	1.724853498495976	72485401019653	1.72485391951394
Std Dev	6.094401179370e-07	678275871223e-07	5.247579855645e-07	5.5689534654981e-07

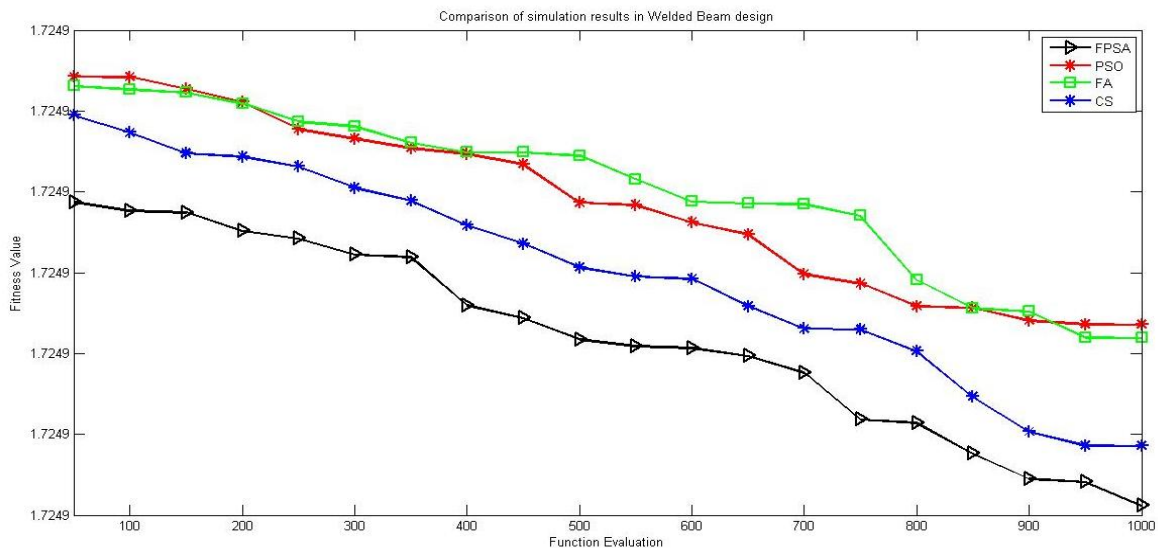


Fig. 4. Illustrates the comparison of simulation results in welded beam

5.1.2 The tension/compression spring design

The tension/compression spring design problem [18–23] shown in Fig. 5. A standard spring design problem has three design variables: the wire diameter $d(=x_1)$, the mean coil diameter $D(=x_2)$, and the length (or number of active coils) $P(=x_3)$. The objective is to minimize the weight ($f(x)$) of a tension/compression spring subject to constraints on minimum deflection, shear stress, surge frequency, limits on outside diameter and on design variables. Formally, the problem can be expressed as:

Minimize

$$V(x) = (2 + x_1)x_2x_3^2$$

subject to

$$g_1(x) = 1 - \frac{x_1x_2^3}{71785x_3^4} \leq 0,$$

$$g_2(x) = \frac{4x_2^2 - x_3x_2}{12566(x_2x_3^3 - x_3^4)} + \frac{1}{5108x_3^2} - 1 \leq 0,$$

$$g_3(x) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0,$$

$$g_4(x) = \frac{x_2 + x_3}{1.5} - 1 \leq 0,$$

$$2 \leq x_1 \leq 15, \quad 0.25 \leq x_2 \leq 1.3, \quad 0.05 \leq x_3 \leq 2.$$



Fig. 5. Schematic view of tension/compression spring problem.

The comparisons of the statistical results obtained by FPSA, CS, FA, and PSO algorithms are presented as shown in Table 3. Accuracy performance is measured in terms of the best, mean, and standard deviation values of the solutions obtained by 20 independent runs. The best solution was obtained using FPSA with an objective function value of 0.0126650555494454. Fig. 6: illustrates the function values with respect to the number of iterations for the tension/compression spring design problem.

Table 3. Comparison of statistical results of tension/compression spring design

	FPSA	CS	FA	PSO
Best	0.0126650555494454	0.0126740326999630	0.012681243353847	0.0126679529561246
Mean	0.0126824784876601	0.0127418557385097	0.012765894187375	0.0127883417176863
StdDev.	1.171356394879e-05	4.720102943650e-05	6.115620208651e-05	6.1304311322847e-05

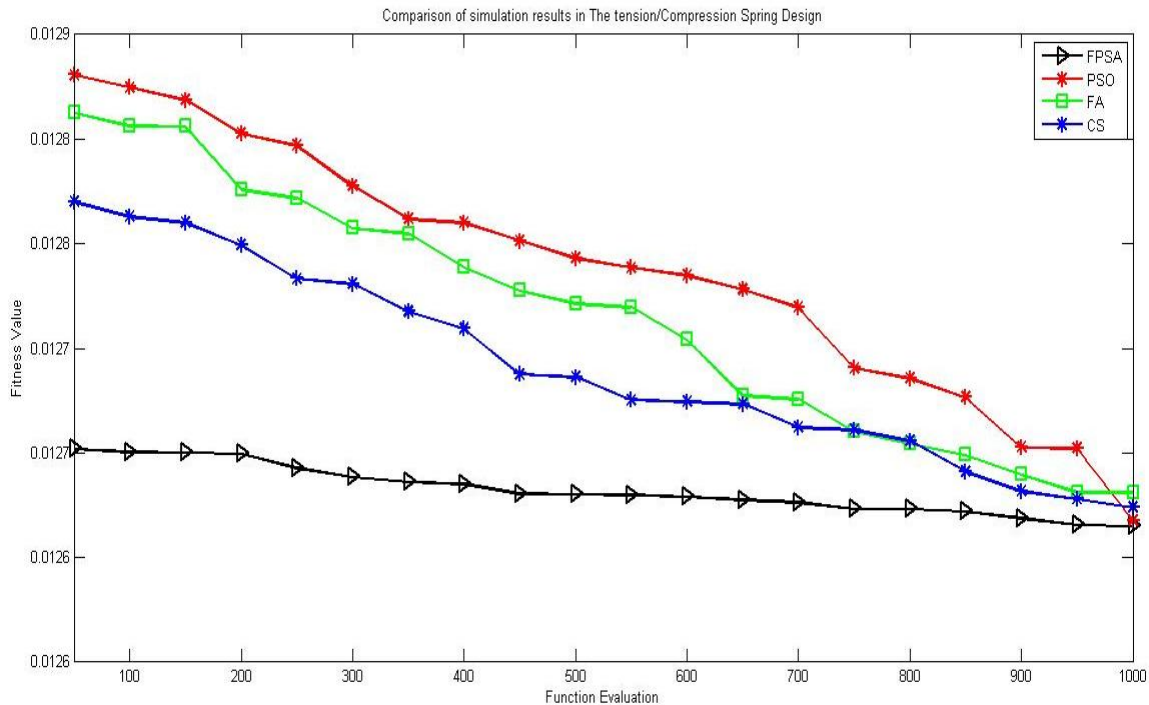


Fig. 6. Illustrates the comparison of simulation results in tension/compression spring design

5.1.3 The speed reducer design

In speed reducer design problem [19,24,21] shown in Fig. 7, the weight of speed reducer is to be minimized subject to constraints on bending stress of the gear teeth, surface stress,

transverse deflections of the shafts, and stresses in the shafts. The variables x_1 to x_7 represent the face width (b), module of teeth (m), number of teeth in the pinion (z), length of the first shaft between bearings (l_1), length of the second shaft between bearings (l_2), and the diameter of first (d_1) and

second shafts (d2), respectively. This is an example of a mixed integer-programming problem. The third variable x3 (number of teeth) is of integer values while all other variables are continuous. There are 11 constraints in this problem resulting in high complexity of the problem.

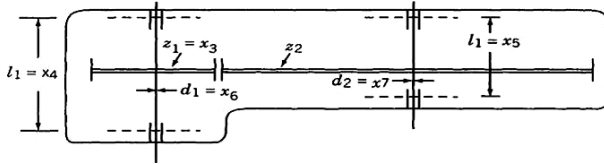


Fig. 7. Speed reducer

Table 4. Comparison of statistical results of speed reducer design problem

	FPSA	CS	FA	PSO
Best	2887.42702290486	2880.46482722267	2877.92580198275	2905.37998331683
Mean	2945.23818793807	2941.90941560529	2946.97407374565	2945.57565992893
StdDev.	41.6750980729589	44.6057435229911	41.1218438217774	31.3007507913235

The comparisons of the statistical results obtained by FPSA, CS, FA, and PSO algorithms are presented as displayed in Table 4. Accuracy performance is measured in terms of the best, mean, and standard deviation values of the solutions obtained by 20 independent runs. The nearly equal solutions were obtained using all algorithms with slight preference for CS with an objective function value of 2880.46482722267. Fig 8: illustrates the function values with respect to the number of iterations for the speed reducer design problem.

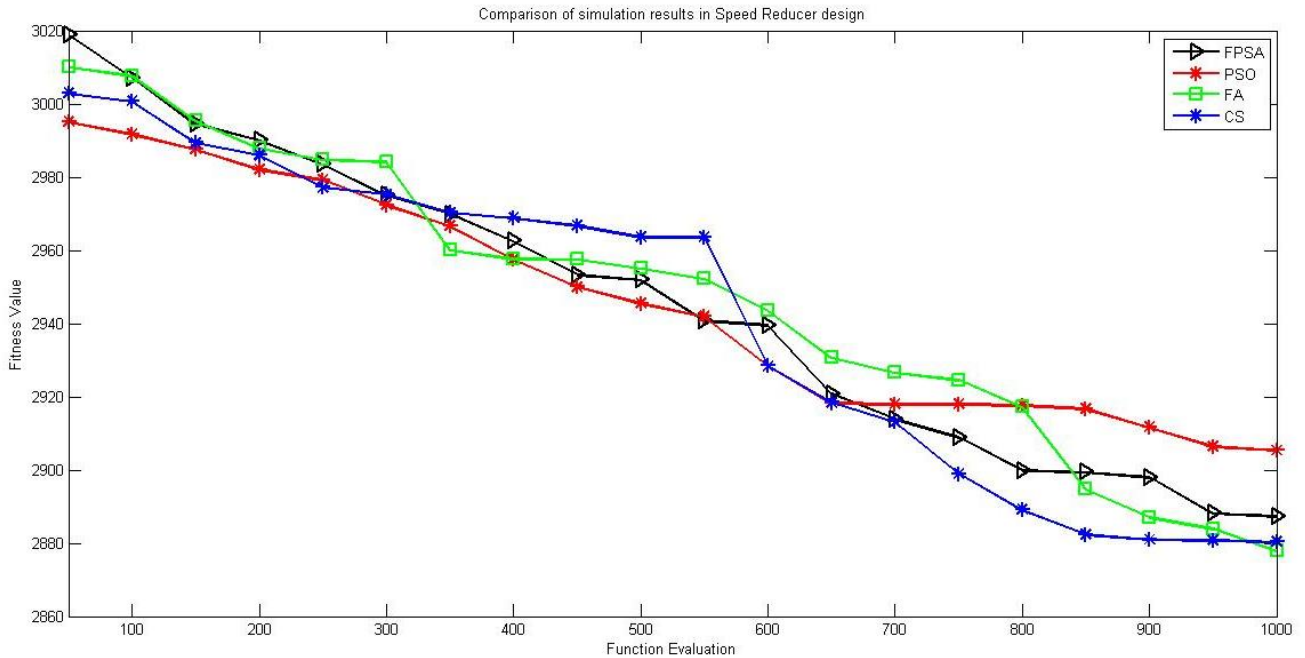


Fig. 8. illustrates the comparison of simulation results in speed reducer design problem

The mathematical formulation of the optimization problem can be written as:

Minimize

$$f(x) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 - x_7^2) + 7.4777(x_6^3 + x_7^3) + 7.854(x_4x_6^2 + x_5x_7^2)$$

subject to

$$g_1(x) = \frac{27}{x_1 x_2^2 x_3} \leq 1,$$

$$g_2(x) = \frac{397.5}{x_1 x_2^2 x_3^2} \leq 1,$$

$$g_3(x) = \frac{1.93x_4^3}{x_2 x_3 x_6^4} \leq 1,$$

$$g_4(x) = \frac{1.93x_5^3}{x_2 x_3 x_7^4} \leq 1,$$

$$g_5(x) = \frac{\sqrt{\left(\frac{745x_4}{x_2 x_3}\right)^2 + 16.9 \times 10^6}}{110x_6^3} \leq 1,$$

$$g_6(x) = \frac{\sqrt{\left(\frac{745x_5}{x_2 x_3}\right)^2 + 157.5 \times 10^6}}{85x_7^3} \leq 1,$$

$$g_7(x) = x_2 x_3 \leq 40,$$

$$g_8(x) = \frac{x_1}{x_2} \geq 5,$$

$$g_9(x) = \frac{x_1}{x_2} \leq 12,$$

$$g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} \leq 1,$$

$$g_{11}(x) = \frac{1.1x_7 + 1.9}{x_5} \leq 1,$$

$$2.6 \leq x_1 \leq 3.6, \quad 0.7 \leq x_2 \leq 0.8, \quad 17 \leq x_3 \leq 28, \quad 7.3 \leq x_4 \leq 8.3$$

$$7.3 \leq x_5 \leq 8.3, \quad 2.9 \leq x_6 \leq 3.9, \quad 0.5 \leq x_7 \leq 5.5.$$

5.1.4 The pressure vessel design

The pressure vessel design problem [18–23] is a well-known benchmark for validating optimization algorithms. Fig. 9 illustrates the problem. The main objective is to minimize the overall cost $f(x)$ including the cost of the material, forming and welding. It has four design variables: x_1 (T_s , shell thickness),

x_2 (T_h , thickness of spherical head), x_3 (Radius of cylindrical shell) and x_4 (L , shell length). $T_s(=x_1)$ and $T_h(=x_2)$ are integer multipliers of 0.0625 in. in accordance with the available thickness of rolled steel plates, and $R(=x_3)$ and $L(=x_4)$ have continuous values of $40 < R < 80$ in. and $20 < L < 60$ in., respectively.

This optimization problem can be written as:

Minimize

$$f(x) = 0.6224x_1 x_3 x_4 + 1.7781x_2 x_3^2 + 3.1661x_1^2 x_4 + 19.84x_1^2 x_3$$

subject to

$$g_1(x) = -x_1 + 0.193x_3 \leq 0,$$

$$g_2(x) = -x_2 + 0.00954x_3 \leq 0,$$

$$g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3} \pi x_3^3 + 1296000 \leq 0,$$

$$g_4(x) = x_4 - 240 \leq 0,$$

$$0 \leq x_i \leq 100, \quad i = 1, 2$$

$$10 \leq x_i \leq 200 \quad i = 3, 4.$$

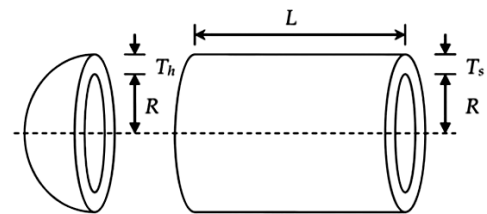


Fig. 9. Schematic view of pressure vessel problem

The comparisons of the statistical results obtained by FPSA, CS, FA, and PSO algorithms are presented as displayed in Table 5. Accuracy performance is measured in terms of the best, mean, and standard deviation values of the solutions obtained by 20 independent runs. The nearly equal solutions were obtained using all algorithms with slight preference for FA with an objective function value of 6062.71755606089. Fig 10: illustrates the function values with respect to the number of iterations for the pressure vessel problem.

Table 5. Comparison of statistical results of pressure vessel problem

	FPSA	CS	FA	PSO
Best	6062.9461682320962	6072.30661137600	6062.71755606089	6066.51211504558
Mean	6852.295591652	6251.87042467342	6276.49610209687	6273.41300752752
StdDev.	115.622843271676	132.067431385067	121.202340495361	130.085266132072

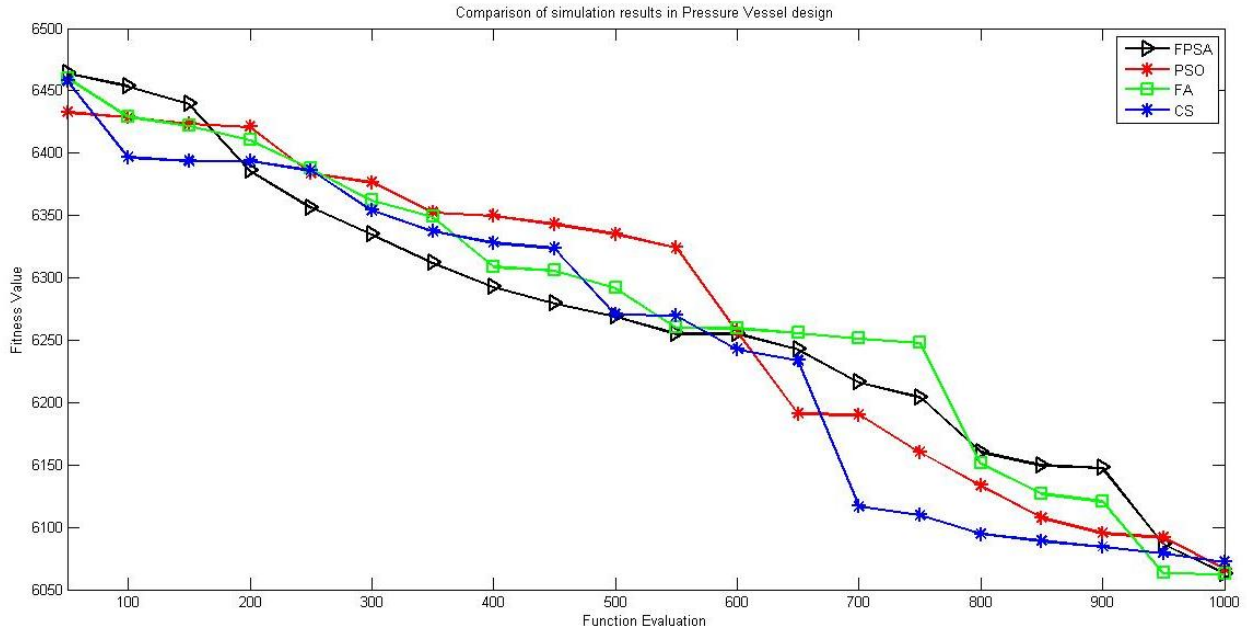


Fig. 10. Illustrates the comparison of simulation results in pressure vessel problem

5.1.5 The cantilever beam design

In cantilever beam design problem [23] shown in Fig. 11, the mean objective is to minimize the overall weight of a cantilever beam with square cross sections. The beam is rigidly supported at node 1, and there is a given vertical force acting at node 6. The design variables are the heights (or widths) of the different beam elements, and the thickness is held fixed (here $t = 2/3$). The mathematical formulation of the optimization problem can be stated as follows:

Minimize

$$f(x) = 0.0624 \sum_{i=1}^5 x_i$$

$$g(x) = \sum_{i=1}^5 \frac{a_i}{x_i^3} - 1 \leq 0,$$

$$a = (a_1, a_2, a_3, a_4, a_5) = (61, 37, 19, 7, 1)$$

$$0.01 \leq x_i \leq 100, \quad i = 1, \dots, 5,$$

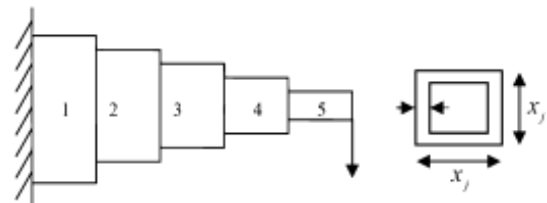


Fig. 11. The Cantilever Beam

The comparisons of the statistical results obtained by FPSA, CS, FA, and PSO algorithms are presented as displayed in Table 6. Accuracy performance is measured in terms of the best, mean, and standard deviation values of the solutions obtained by 20 independent runs. The best solution was obtained using FPSA with an objective function value of 1.33999122280701. Fig. 12: illustrates the function values with respect to the number of iterations for the cantilever beam design problem.

Table 6. Comparison of statistical results of cantilever beam design

	FPSA	CS	FA	PSO
Best	1.33999122280701	1.33999509315973	1.34078835627094	1.34013960563364
Mean	1.34450023221826	1.34654331839642	1.34430747388793	1.34399097305102
StdDev.	0.002739386596007	0.002519033574168	0.002408741945849	0.002566539714930

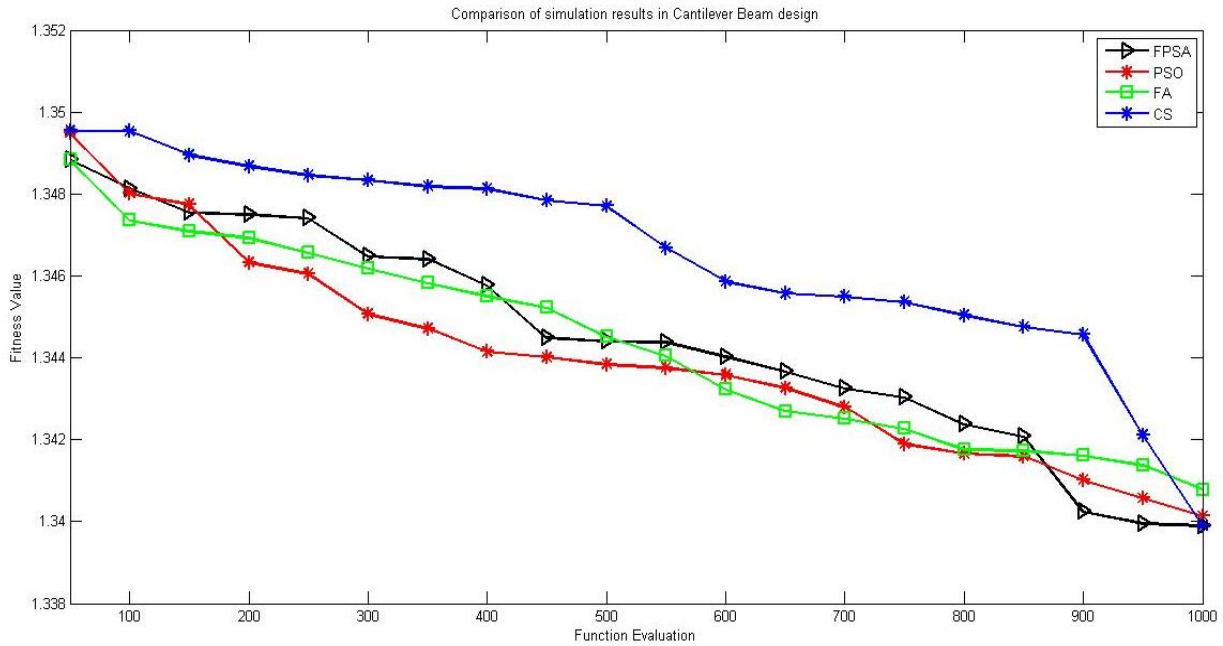


Fig. 12. illustrates the comparison of simulation results in cantilever beam design

5.1.6 The three-bar truss design

The three-bar truss design problem [19–21] shown in Fig. 13, which often used as a standard test problem for constrained design optimization, this problem was first presented by Nowcki (1974). The volume of a statically loaded three-bar truss is to be minimized subject to stress (σ) constraints on each of the truss members. The objective is to evaluate the optimal cross sectional areas. This optimization problem can be written as:

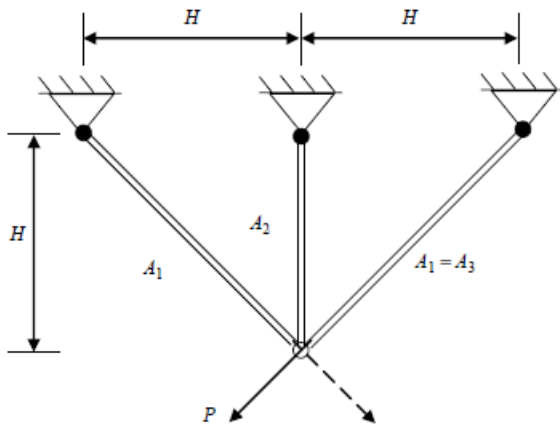


Fig. 13. Four bar truss

Minimize :

$$f(x) = (2\sqrt{2x_1} + x_2) \times l$$

subject to

$$g_1(x) = \frac{\sqrt{2x_1} + x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \leq 0,$$

$$g_2(x) = \frac{x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \leq 0,$$

$$g_3(x) = \frac{1}{x_1 + \sqrt{2}x_2} P - \sigma \leq 0,$$

where $0 \leq x_1, x_2 \leq 1; l = 100\text{cm}$,

$P = 2\text{KN}/\text{cm}^2$ and $\sigma = 2\text{KN}/\text{cm}^2$.

The comparisons of the statistical results obtained by FPSA, CS, FA, and PSO algorithms are presented as displayed in Table 7. Accuracy performance is measured in terms of the best, mean, and standard deviation values of the solutions obtained by 20 independent runs. The nearly equal solutions were obtained using all algorithms with slight preference for FA with an objective function value of 263.89586390931. Fig 14: illustrates the function values with respect to the number of iterations for the three-bar truss design problem.

Table 7. Comparison of statistical results of the three-bar truss design

	FPSA	CS	FA	PSO
Best	263.89595502186	263.895885687007	263.895863909310	263.896017785072
Mean	263.896566975811	263.896539611530	263.896389983192	263.897514777445
StdDev.	0.000389659036066	0.000406673310220	0.000369856927395	0.001062109487269

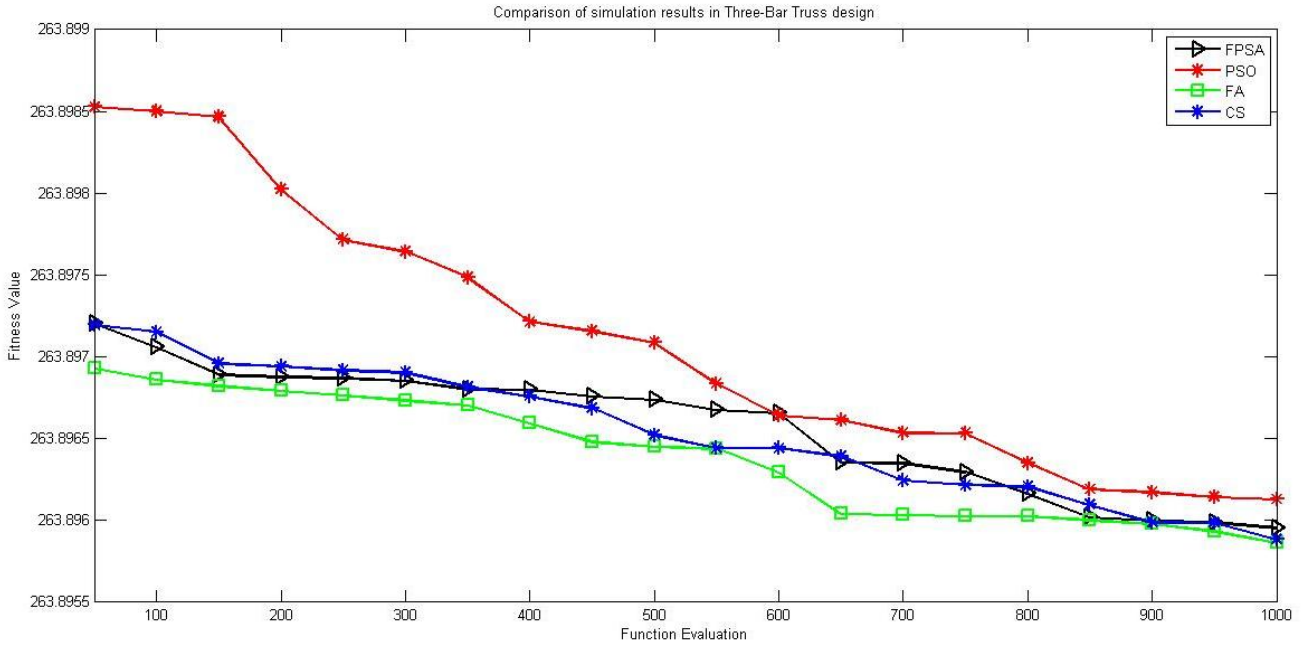


Fig. 14. illustrates the comparison of simulation results in the three-bar truss design

5.1.7 Heat exchanger design

Heat Exchanger Design [23] is a benchmark minimization problem that is regarded as difficult test case due to all the constraints are binding. This constrained function has eight variables and six inequality constraints. The mathematical formulation of the optimization problem can be stated as follows:

Minimize :

$$f(x) = x_1 + x_2 + x_3,$$

$$g_1(x) = 0.0025(x_4 + x_6) - 1 \leq 0,$$

$$g_2(x) = 0.0025(x_5 + x_7 - x_4) - 1 \leq 0,$$

$$g_3(x) = 1 - 0.01(x_8 - x_5) \geq 0,$$

$$g_4(x) = x_1x_6 - 833.33252x_4 - 100x_1 + 83333.333 \geq 0,$$

$$g_5(x) = x_2x_7 - 1250x_5 - x_2x_4 + 1250x_4 \geq 0,$$

$$g_6(x) = x_3x_8 + 2500x_5 - x_3x_5 - 125000 \geq 0,$$

$$100 \leq x_1 \leq 10000, 1000 \leq x_2, x_3 \leq 10000, 10 \leq x_i \leq 1000 (i = 4, \dots, 8).$$

The comparisons of the statistical results obtained by FPSA, CS, FA, and PSO algorithms are presented as displayed in Table 8. Accuracy performance is measured in terms of the best, mean, and standard deviation values of the solutions obtained by 20 independent runs. The best solution was obtained using FPSA with an objective function value of 7,049.57814832128. Fig. 15: illustrates the function values with respect to the number of iterations for the heat exchanger design problem.

Table 8. Comparison of statistical results of heat exchanger design

	FPSA	CS	FA	PSO
Best	7,049.57814832128	7,049.86990534642	7,050.26039418716	7,049.73843584822
Mean	7,060.53191098780	7,056.30534764689	7,057.99511681114	7,059.28972947729
StdDev.	5.74817059120558	4.37524610926571	6.04952221016639	6.52636679314155

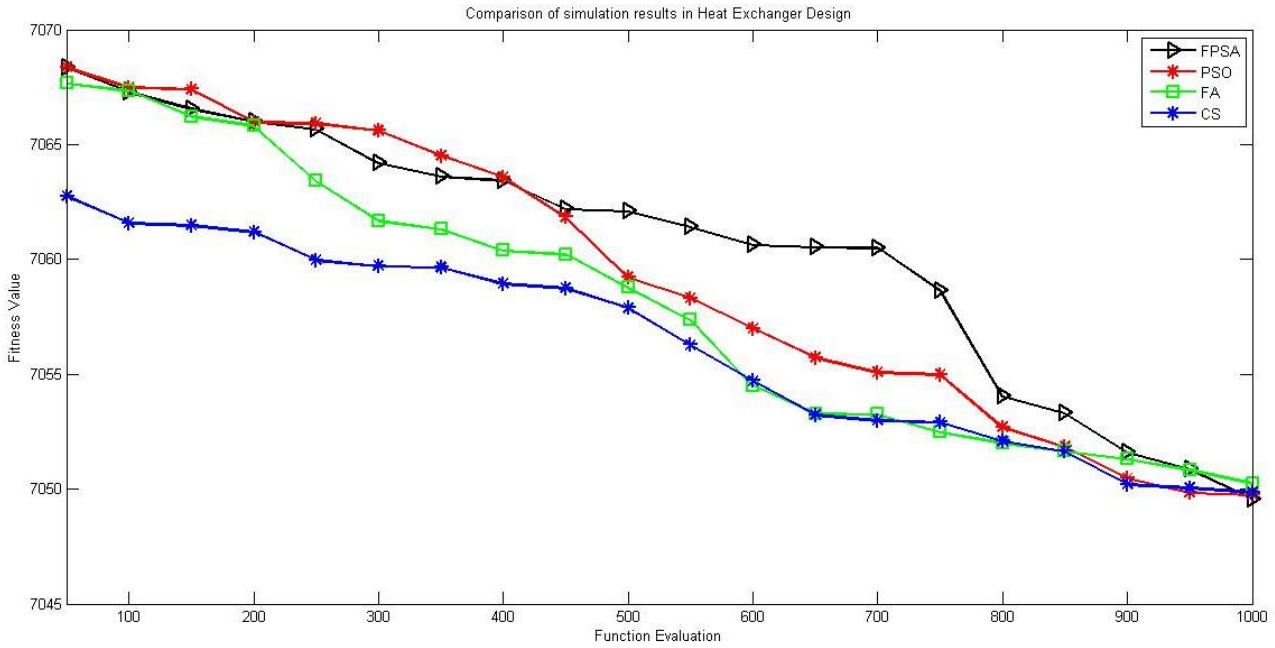


Fig. 15. Illustrates the comparison of simulation results in heat exchanger design

5.1.8 Corrugated bulkhead design

Corrugated bulkhead design [25,26,23] are often used in chemical tankers and product tankers in order to help facilities cargo tank washing effectively. This problem is as an example of minimum-weight design of the corrugated bulkheads for a tanker. Four design variables of the problem are width (b), depth (h), length (l), and plate thickness (t) for minimum-weight design of the corrugated bulkheads for a tanker, the mathematical formula for the optimization problem as follows [27]:

$$\text{Minimize : } f(b, h, l, t) = \frac{5.885t(b+l)}{b + \sqrt{l^2 - h^2}}$$

$$g_1 = th \left(0.4b + \frac{1}{6} \right) - 8.94 \left(b + \sqrt{l^2 - h^2} \right) \geq 0$$

$$g_2 = th^2 \left(0.2b + \frac{1}{12} \right) - 2.2 \left(8.94 + \left(b + \sqrt{l^2 - h^2} \right) \right)^4 \geq 0$$

$$g_3 = t - 0.0156b - 0.15 \geq 0$$

$$g_4 = t - 0.0156l - 0.15 \geq 0$$

$$g_5 = t - 1.05 \geq 0$$

$$g_6 = l - h \geq 0, \quad 0 \leq b, h, l \leq 100 \text{ and } 0 \leq t \leq 5$$

Table 9. Comparison results of the corrugated bulkhead design example

	b(cm)	h(cm)	l(cm)	t(cm)	best
FPSA	57.69231	37.26590	57.69231	1.05	7.008391
CS	57.69231	37.36410	57.69231	1.05	7.01413
FA	57.69231	38.00090	57.69231	1.05	7.05219
PSO	57.69231	37.56410	57.69231	1.05	7.02594

5.1.9 Design of a gear train

The below Fig. 16 shows the gear train problem [27,25,28]. A gear ratio between the driver and driven shafts must be achieved when designing a compound gear train. The gear ratio for gear train is defined as the ratio of the angular velocity of the output shaft to that of the input shaft. It is

The comparison result of minimum-weight and the statistical values of the best solution obtained by FPSA, CS, FA and PSO are given in Table 9. Accuracy performance is measured in terms of the best, mean, and standard deviation values of the solutions obtained by 20 independent runs. The best minimum-weight in this study is 7.008391 with thickness 1.05 cm. while using Gandomi et al [27] obtained 5.894331, with thickness 0.7306255. In Gandomi et al also the constraint g5 is unverified and must be at least 1.05. Results are obtained from FPSA algorithm is better than the results obtained using Gandomi et al [27]. It gave us a greater thickness, with a slight increase in minimum-weight and verify all the constraints of given problem. While Gandomi et al got minimum-weight but did not verify all the constraints. Thickness is very small leading of poor quality. If we exclude thickness constraint we could get a better result than that of (5.894331). Since it is between (0-5) when t approaches to zero, b also approaches to zero.

desirable to produce a gear ratio as close as possible to 1/6.931. For each gear, the number of teeth must be between 12 and 60. The design variables Ta, Tb, Td, and Tf are the numbers of teeth of the gears a, b, d and f, respectively, which must be integers.

$$\bar{x} = (T_d, T_b, T_a, T_f)^T = (x_1, x_2, x_3, x_4)^T$$

The optimization problem is expressed as:

$$\min z = \left(\frac{1}{6.931} - \frac{\lfloor T_d \rfloor \lfloor T_b \rfloor}{\lfloor T_a \rfloor \lfloor T_f \rfloor} \right)^2 = \left(\frac{1}{6.931} - \frac{\lfloor x_1 \rfloor \lfloor x_2 \rfloor}{\lfloor x_3 \rfloor \lfloor x_4 \rfloor} \right)^2$$

subject to $12 \leq x_i \leq 60 \quad i = 1, 2, 3, 4$

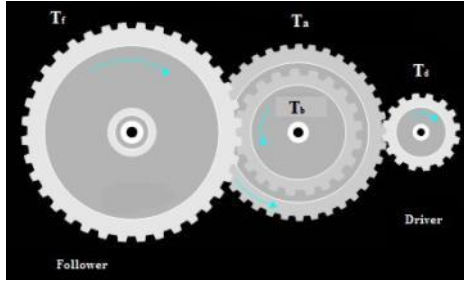


Fig. 16. A gear train

Table 10. Comparison results of the FPSA, CS, FA and PSO

	FPSA	CS	FA	PSO
Best	2.7E-012	2.70009E-012	2.7E-012	2.700857E-012
Mean	2.7E-012	1.04E-010	5.1314E-012	1.1371E-011
StdDev.	0.00E+00	2.7E-010	7.6868E-012	1.01307E-011

5.1.10 Proportional-integral- derivative (PID) controller

Proportional-integral-derivative (PID) controllers [28] are widely used to build automation equipment in industries. Shown in Fig. 17 below. They are easy to design, implement, and are applied well in most industrial control systems process control, motor drives, magnetic, etc.



Fig. 17. Generic Closed Loop System

Assume that the system is modeled by an nth-order process with time delay L:

$$G_p(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0} e^{-Ls} \quad (6)$$

Here, we assume $n > m$ and the system (6) is stable. The PID controller has the following transfer function:

$$G_C(s) = K_p + T_i s^{-\lambda} + T_d s^{\delta}$$

The optimization problem is summarized as follows:

$$\min z = z(K_p, T_i, T_d, \lambda, \delta)$$

subject to $L \leq K_p, T_i, T_d, \lambda, \delta \leq U$.

Where the z, L, U is given by the designer. Note that the constraints introduced to guarantee the stability of the closed-

The constraint ensures that the error between obtained gear ratio and the desired gear ratio is not more than the 50% of the desired gear ratio.

The comparison resulted obtained by the FPSA, CA, FA and PSO algorithms are given in Table 10. The comparison in terms of the best, error, mean, standard deviation values, these values were obtained out of 20 independent runs. The result indicates a better achievement for FPSA, with an objective function value of 2.7 E-012.

Correct implementation of the PID depends on the specification of three parameters: proportional gain (Kp), integral time (Ti) and derivative time (Td). These three parameters are often tuned manually by trial and error, which has a major problem in the time needed to accomplish the task.

Also, the values of three design parameters (Kp, Ti, Td,) are directly determined by solving the above optimization problem.

5.2 Simulation Example

Consider the following transfer function presented in [29]

$$G_p(s) = \frac{1}{0.8s^{2.2} + 0.5s^{0.9} + 1}$$

The initial parameters are chosen randomly in the following range: Kp: [1, 1000], Ti: [1, 500], Td: [1, 500], λ : [0,2], δ : [0,2]. We want to design a controller such that the closed loop system has a maximum peak overshoot $M_p = 10\%$ and trise = 0.3 seconds. This translates to $\zeta = 0.65$ (damping ratio), $\omega_0 = 2.2$ s⁻¹ (undamped natural frequency). we then find out the positions of the dominant poles of the closed loop system,

$$p_{1,2} = -\xi\omega_0 \pm j\omega_0\sqrt{1-\xi^2}$$

The dominant poles for the closed loop controlled system should lie at $(-1.43+ j1.67)$ and $(-1.43- j1.67)$.

For $p1=(-1.43+ j1.67)$, the characteristic equation is:

$$1 + \frac{K_p + T_i (-1.43 + j1.67)^{-\lambda} + T_d (-1.43 + j1.67)^{\delta}}{0.8(-1.43 + j1.67)^{2.2} + 0.5(-1.43 + j1.67)^{0.9} + 1} = 0$$

Table 11 illustrates the calculated optimized parameters of PID controller using FPSA, CS, PSO and FA .It could be observed that the optimized parameters calculated using

FPSA algorithm generates the best control step response as illustrate in Fig. 18.

Table 11. Results for the PID

Technique	The order PID controller			Step response
	Kp	Ti	Td	
FPSA	415.4	402.2	100.3	1.0020
CS	512.75	361.6	148.48	1.0030
FA	442.68	320	120.2	1.0208
PSO	515.58	375.23	145.48	1.0339

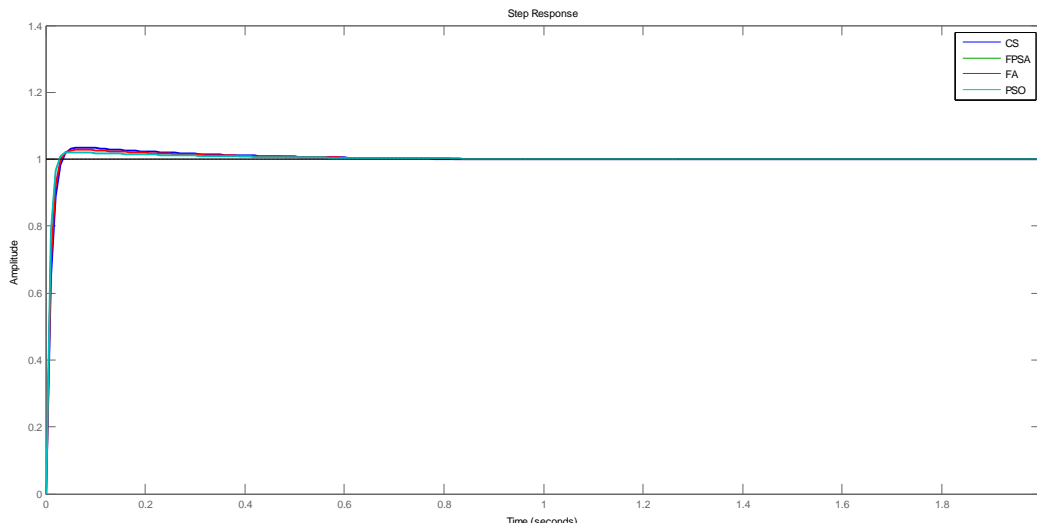


Fig. 18. Comparison Close Loop step response of the system with FPSA, CS,FA and PSO for PID controller

6. CONCLUSIONS

In the present study, FPSA algorithm has been validated using several benchmark mathematical and engineering design problems. Several simulation examples have been completed to verify the weight of the planned algorithm .The comparison among the results determined by the proposed algorithm and the compared algorithms are reported in Tables (2-11). Considering the comparison and statistical results obtained, it can be concluded that the FPSA is more efficient than the other optimization engines for engineering optimization problems. Moreover, the superiority of the FPSA algorithm to finding the near global optimum solution. The results indicate that FPSA is accurate, reliable and efficient at finding near global optimal solution, also the proposed algorithm was a strong competitor compared with other known algorithms so obtained the best results in more often. Therefore, the solutions obtained by our approach represent great contribution for finding the optimum solutions of these problems.

7. REFERENCES

[1] A. Ghodrati and S. Lotfi, "A hybrid CS/PSO algorithm for global optimization," in *Intelligent Information and Database Systems*, Springer, 2012, pp. 89–98.

[2] G. Wang, L. Guo, H. Duan, L. Liu, H. Wang, and B. Wang, "A hybrid meta-heuristic DE/CS algorithm for UCAV path planning," *Journal of Information and Computational Science*, vol. 5, pp. 4811–4818, 2012.

[3] F. Wang, X. He, L. Luo, and Y. Wang, "Hybrid optimization algorithm of PSO and Cuckoo Search," in *Artificial Intelligence, Management Science and*

Electronic Commerce (AIMSEC), 2011 2nd International Conference on, 2011, pp. 1172–1175.

[4] R. Raju, R. Babukarthik, and P. Dhavachelvan, "Hybrid Ant Colony Optimization and Cuckoo Search Algorithm for Job Scheduling," in *Advances in Computing and Information Technology*, Springer, 2013, pp. 491–501.

[5] T. Hassanzadeh and M. R. Meybodi, "A new hybrid algorithm based on Firefly Algorithm and cellular learning automata," in *20th Iranian Conference on Electrical Engineering (ICEE)*, 2012, pp. 628–633.

[6] A. Abdullah, S. Deris, M. S. Mohamad, and S. Z. M. Hashim, "A new hybrid firefly algorithm for complex and nonlinear problem," in *Distributed Computing and Artificial Intelligence*, Springer, 2012, pp. 673–680.

[7] A. A. El-Sawy, E. M. Zaki, and R. Rizk-Allah, "A Novel Hybrid Ant Colony Optimization and Firefly Algorithm for Solving Constrained Engineering Design Problems," *Journal of Natural Sciences and Mathematics*, vol. 6, 2012.

[8] I. M. Hezam and M. Abdel-Baset, "An Improved Flower Pollination Algorithm for Ratios optimization Problems". *Applied Mathematics & Information Sciences Letters An International Journal*, vol. 3, No. 2, pp. 83–91, 2015.

[9] S. Farook and P. S. Raju, "Evolutionary Hybrid Genetic-Firefly Algorithm for Global Optimization."

[10] A.-R. Osama, A.-B. Mohamed and E.-h. Ibrahim, "A Novel Hybrid Flower Pollination Algorithm with Chaotic Harmony Search for Solving Sudoku Puzzles," *International Journal of Engineering Trends and*

- Technology (IJETT), vol. 7, no. 3, pp. 126-132, January 2014.
- [11] O. Abdel-Raouf, M. Abdel-Baset, and I. El-henawy, "An Improved Flower Pollination Algorithm with Chaos," *I.J. Education and Management Engineering*, vol. 2, pp. 1–8, 2014.
- [12] V. M. Gellat CD Kirkpatrick S, "Optimization by simulated annealing," *Science*, vol. 220(4598):6, pp. 71–80, 1983.
- [13] A. Askarzadeh, "A discrete chaotic harmony search-based simulated annealing algorithm for optimum design of PV/wind hybrid system," *Solar Energy*, vol. 97, pp. 93–101, 2013.
- [14] S. C. Leung, D. Zhang, C. Zhou, and T. Wu, "A hybrid simulated annealing metaheuristic algorithm for the two-dimensional knapsack packing problem," *Computers & Operations Research*, vol. 39, pp. 64–73, 2012.
- [15] S. M. Mousavi and R. Tavakkoli-Moghaddam, "A hybrid simulated annealing algorithm for location and routing scheduling problems with cross-docking in the supply chain," *Journal of Manufacturing Systems*, vol. 32, pp. 335–347, 2013.
- [16] X.-S. Yang, *Nature-inspired metaheuristic algorithms*. Luniver Press, 2011.
- [17] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer methods in applied mechanics and engineering*, vol. 186, pp. 311–338, 2000.
- [18] L. C. Cagnina, S. C. Esquivel, and C. A. C. Coello, "Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer," *Informatica (Slovenia)*, vol. 32, pp. 319–326, 2008.
- [19] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, "Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems," *Computers & Structures*, vol. 110, pp. 151–166, 2012.
- [20] M. Jaberipour and E. Khorram, "Two improved harmony search algorithms for solving engineering optimization problems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, pp. 3316–3331, 2010.
- [21] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, "Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems," *Applied Soft Computing*, vol. 13, pp. 2592–2612, 2013.
- [22] X.-S. Yang, *Engineering optimization: an introduction with metaheuristic applications*. Wiley, 2010.
- [23] X.-S. Yang and A. Hossein Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Engineering Computations*, vol. 29, pp. 464–483, 2012.
- [24] M.-H. Lin, J.-F. Tsai, N.-Z. Hu, and S.-C. Chang, "Design Optimization of a Speed Reducer Using Deterministic Techniques," *Mathematical Problems in Engineering*, vol. 2013, 2013.
- [25] O. Abdel-Raouf, I. El-henawy and M. Abdel-Baset "chaotic Harmony Search Algorithm with Different Chaotic Maps for Solving Assignment Problems " *International Journal of Computational Engineering & Management*, Vol. 17, pp. 10-15 ,2014.
- [26] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Nature & Biologically Inspired Computing*, 2009. NaBIC 2009. World Congress on, 2009, pp. 210–214.
- [27] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, pp. 17–35, 2013.
- [28] O. A. Raouf, M. A. Baset, I. M. Elhenawy, "A New Hybrid Flower Pollination Algorithm for Solving Constrained Global Optimization Problems", *International Journal of Applied Operational Research* Vol. 4, No. 2, pp. 1-13, Spring 2014.
- [29] D. Maiti, S. Biswas, and A. Konar, "Design of a fractional order PID controller using particle swarm optimization technique," *arXiv preprint arXiv:0810.3776*, 2008.