Advanced Engineering Technology and Application
An International Journal

http://dx.doi.org/10.12785/aeta/040203

An Effective Hybrid Flower Pollination and Genetic Algorithm for Constrained Optimization Problems

Mohamed Abdel-Baset [*,1] and *Ibrahim M. Hezam* [2].

[1] Department of Operations Research, faculty of Computers and Informatics, Zagazig University, El-Zera Square, Zagazig, Sharqiyah, Egypt.
[2] Department of computer, Faculty of Education, Ibb University, Ibb city, Yemen.

Received: 29 Mar. 2015, Revised: 20 Apr. 2015, Accepted: 24 Apr. 2015.
Published online: 1 May 2015.

Abstract: Flower pollination algorithm (FPA) is a new nature-inspired algorithm, based on the characteristics of flowering plants .In this paper, a new hybrid optimization method called hybrid flower pollination algorithm with genetic (FPA-GA) is proposed. The method combines the standard flower pollination algorithm (FPA) with the genetic (GA) algorithm to improve the searching accuracy. The FPA-GA algorithm is used to solve constrained optimization problems. To verify the performance of FPA-GA, seven benchmark optimization problems chosen from literature are employed. Experimental results indicate that the proposed method performs better than, or at least comparable to state-of-the-art methods from literature when considering the quality of the solutions obtained. Experimental results further demonstrate the proposed method is very effective.

Keywords: Flower pollination algorithm; hybrid optimization; global optimization; genetic algorithm; Constrained Optimization.

1 Introduction

Optimization is a field of applied mathematics that deals with finding the extremal values of a function in a domain of definition, subject to various constraints on the variable values [1].Global optimization refers to finding the extreme value of a given nonconvex function in a certain feasible region and such problems are classified in two classes; unconstrained and constrained problems. Solving global optimization problems has made great gain from the interest in the interface between computer science and operations research [1-5].

There are two categories of optimization techniques: exact and heuristic. Exact strategies guarantee the optimal solution will be found, and work well for many problems. However for complex problems or ones with a very large number of parameters, exact strategies may require very high computational costs [3]. A large amount of real-world problems fall in this category of complex problems, and in order to solve them in a reasonable amount of time a different approach is needed [3,6]. For these problems, Meta-heuristic algorithms are considered as efficient tools to obtain optimal solutions [6-29]. Two important characteristics of meta-heuristics are intensification and diversification. Intensification, also called exploitation, intends to use the information from the current best solutions. This process searches around the neighborhood of the current best solutions and selects the best candidates. Diversification, also called exploration, guarantees that the algorithm can explore the search space more efficiently, often by randomization. This is the essential step that guarantees that the system can jump out of any local optima and can generate new solutions as diversely as possible [6-7].

These methods have received remarkable attentions as they are known to be derivative free, robust and often involve a small number of parameter tunings [6-29]. However, applying such single methods is sometimes too restrictive, especially for high dimensional and nonlinear problems. This is because these methods usually require a substantially huge amount of computational times and are frequently trapped in one of the local optima. Recently, different methods combining meta-heuristics with local search methods is a practical remedy to overcome the drawbacks of slow convergence and random constructions of meta-heuristics [30-38]. In these hybrid methods, local search strategies are inlaid inside meta-heuristics in order to guide them especially in the vicinity of local minima, and overcome their slow convergence especially in the final stage of the search.

Recently, Yang [39] developed a new Flower pollination algorithm (FP) that draws its inspiration from the flow pollination process of flowering plants. In this paper, a new

*Corresponding author e-mail: analyst_mohamed@yahoo.com

hybrid optimization method is introduced. The proposed method, hybrid flower pollination algorithm with genetic algorithm for solving constrained global optimization problems. The experimental results showed that the accuracy and speed performance of the FPA-GA method had outperformed the other existing methods.

2. The Flower pollination Algorithm

The rest of this paper is organized as follows: In Section 2 we review the basic FPA. The genetic algorithm is presented in Section 3 respectively. The proposed algorithm is given in Section 4. The results are given in Section 5. In the last section we conclude this paper and point out some future research directions.

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 [39].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:

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

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 [39-43]. The basic steps of FPA can be summarized as the pseudo-code shown in Figure 1.

Algorithm 1: Flower pollination algorithm

Define Objective function $f(x)$, $x = (x_1, x_2, ..., x_d)$
Initialize a population of n flowers/pollen gametes with random solutions
*Find the best solution **B** in the initial population*
Define a switch probability $p \in [0, 1]$
Define a stopping criterion (either a fixed number of generations/iterations or accuracy)
while (t <MaxGeneration)
  for i = 1 : n (all n flowers in the population)
   if rand <p,
    Draw a (d-dimensional) step vector L which obeys a L´evy 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 Genetic Algorithm

The basic steps of GA can be summarized as the pseudo-code shown in Figure 2.

Algorithm 2: Genetic algorithm

Define Objective function f (x), x = (x1, x2, ..., xd)
Encode the solution into chromosomes (binary strings)
Define fitness F (F $\alpha f(x)$ *) for maximization)*
Generate the initial population
Initial probabilities of crossover (p_c*) and mutation (*p_m*)*
while *(t < MaxGeneration) or (stop criterion);*
Generate new solution by crossover and mutation
if p_c *>rand, Crossover; end if*
if p_m *>rand, Mutate; end if*
Accept the new solutions if their fitness increase
Select the current best for new generation (elitism)
end while
Decode the results and visualization
End

Fig. 2 Pseudo code of genetic algorithm

4 Proposed Hybrid Flower Pollination Algorithm and Genetic Algorithm for Constrained Optimization Problems

This section describes the proposed hybrid flower pollination and genetic algorithm. The initial motivation of developing hybrid FPA-GA approach is to combine the advantages of both flower pollination algorithm and genetic algorithm. To find an optimal solution to an optimization problem is often a very challenging task, depending on the choice and the correct use of optimization technique. In general, an ideal global optimization technique should have the following characteristics: method should be easy and simple to implement, good balance between exploration and exploitation, true global optimum should be found in each run, convergence should be fast, algorithm should have minimum control parameters to tune and algorithm should require minimum computational power to run effectively. In the first step, FPA explores the search place in order to either isolate the most promising region of the search space. In the second step, to improve global search and get rid of trapping into several local optima, it is introduced GA to explore search space (starting with the solution obtained by FPA and find new better solutions. The steps of the proposed algorithm for solving constrained global optimization problems can be summarized as the pseudo-code shown in Figure 3.

4.1 Handling Constraints

One of the well-known techniques of handling constraints is using penalty function, which transforms constrained problem into unconstrained ones, consisting of a sum of the objective and the constraints weighted by penalties. By using penalty function methods, the objectives are inclined to guide the search toward the feasible solutions. Hence, in this paper the corresponding objective function used in is defined and described as:

$$\min \quad F(x) = f(x) + \lambda \sum_{n=1}^{K} \max(0, g_n) \qquad (4)$$

Where $f(x)$ is the objective function for assignment problem is, λ is the penalty coefficient and it is set to a value of 10^7 in this paper, K is the number of constraints and g_n the constraints of the problem.

Algorithm 3: Hybrid FPA-GA

Define Objective function f (x), x = (x1, x2, ..., 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]*
Define a stopping criterion (either a fixed number of generations/iterations or accuracy)
Define Genetic algorithm parameters pc, pm
Begin FPA
while (t <MaxGeneration)
for i = 1 : n (all n flowers in the population)
if rand <p,
Draw a (d-dimensional) step vector L which obeys a L´evy 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
Final the best solution found
End begin FPA
Begin GA
i=0
Initial population P(0) = Final best population of flowers/pollen gametes
Evaluate P(0) fitness
while *(t < MaxGeneration) or (stop criterion); do*
i=i+1
Select P(i) from P(i − 1)
Recombine P(i) with crossover probability p_c
Mutate P(i) with mutation probability p_m
Evaluate P(i) fitness
end *while*
Rank the chromosomes, find the current best and save
Postprocess results and visualization
end begin GA

Fig. 3 Pseudo code of the FPA-GA

5 Experimental 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 FPA-GA, 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 optimization problems to investigate the performances of the proposed algorithm. The best results obtained by FPA-GA for test problems (1–7) are presented in Table 1. In these problems, the initial parameters are set at *n*= 50 and the number of iterations is set to *t* = 1000.The results of FPA-GA algorithm are conducted from 40 independent runs for each problem. The comparison between the results determined by the proposed approach and the compared algorithms are reported in Table 1. The statistical results of the FPA-GA on the benchmark problems are summarized in Table 2. It includes the known optimal solution for each test problem and the obtained best, median, mean and worst values and the standard deviations (SD).

The results have demonstrated the superiority of the proposed approach to finding the global optimal solution. So far, these problems have been widely used as benchmarks for research with different methods by many researchers. Definitions of benchmark problems are described as follows:

5.1 Test problem 1

This problem, originally introduced by Bracken and McCormick [44], is a constrained minimization problem. Table 1 shows the best solution from the FPA-GA algorithm and also provides the results obtained using the GA (Homaifar et al. [20]), the evolutionary programming (Fogel [45]) and harmony search (Lee and Geem [46]).The problem can be formulated as:

$$\min\ f(x) = (x_1-2)^2 + (x_2-1)^2 \qquad (5)$$

$$\begin{aligned} &s.t. \\ &g_1(x) = (x_1-2)^2 + (x_2-1)^2, \\ &g_2(x) = \frac{-x_1^2}{4} - x_2^2 + 1 \geq 0, \\ &-10 \leq x_1 \leq 10, -10 \leq x_2 \leq 10 \end{aligned}$$

5.2 Test problem 2

This function is a minimization problem with two design variables and two inequality constraints. The FPA-GA best solutions were compared to the previous solutions reported by Deb [47] using GA and Lee and Geem [45] using harmony search in Table 1. The problem formulation is:

$$\min\ f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \qquad (6)$$

$$\begin{aligned} &s.t. \\ &g_1(x) = 4.84 - (x_1-0.05)^2 - (x_2-2.5)^2 \geq 0, \\ &g_2(x) = x_1^2 + (x_2-2.5)^2 - 4.84 \geq 0, \\ &0 \leq x_1 \leq 6, 0 \leq x_2 \leq 6 \end{aligned}$$

5.3 Test problem 3

The welded beam structure is a practical design problem that has been often used as a benchmark for testing different optimization methods [5, 47-49]. The structure consists of beam A and the weld required to hold the beam to member B. A welded beam is designed for minimum cost *f(x)* subject to constraints: g_1 shear stress τ, g_2 bending stress in the beam σ, g_7 buckling load on the bar $\zeta(x)$, g_6 end deflection of the beam δ and g_3; g_4; g_5 side constraints[2,5]. And there are four design variables. The FPA-GA best solutions were compared to the previous solutions reported by other method in Table 1. The problem can be stated as follows:

$$\min\ f(x) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2), \qquad (7)$$

$$\begin{aligned} &s.t. \\ &g_1(x) = \tau(x) - \tau_{\max} \leq 0, \\ &g_2(x) = \sigma(x) - \sigma_{\max} \leq 0, \\ &g_3(x) = x_1 - x_4 \leq 0, \\ &g_4(x) = 0.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \leq 0, \\ &g_5(x) = 0.125 - x_1 \leq 0, \\ &g_6(x) = \delta(x) - \delta_{\max} \leq 0, \\ &g_7(x) = \zeta - \zeta(x) \leq 0 \\ &where \end{aligned}$$

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

$$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = P(L + \frac{x_2}{2}),$$

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

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

$$\sigma(x) = \frac{6PL}{x_4 x_3^2}, \delta(x) = \frac{4PL^3}{E x_3^3 x_4},$$

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

$$\begin{aligned} &P = 6000 lb, L = 14 in., E = 30 \times 10^6\ psi, \\ &G = 12 \times 10^6\ psi, \tau_{\max} = 30600\ psi, \sigma_{\max} = 30000, \\ &\delta_{\max} = 0.25 in. \end{aligned}$$

5.4 Test problem 4

Himmelblau's Nonlinear Optimization Problem, This problem is originally proposed by Himmelblau [50] and solved using Generalize Reduced Gradient method (GRG).Table 1 lists the optimal values of the function problem obtained by the FPA-GA algorithm, and compares them with earlier results reported by other methods Has been solved by Deb [47], Lee and Geem [46].

$$\min\ f(x) = 5.357847x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141, \quad (8)$$

$$\begin{aligned} &s.t. \\ &g_1(x) = 85.334407 + 0.0056858x_2x_5 + 0.00026x_1x_2 - 0.0022053x_3x_5, \\ &g_2(x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2, \\ &g_3(x) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4, \\ &0 \le g_1(x) \le 92, 90 \le g_2(x) \le 110, 20 \le g_3(x) \le 25 \\ &78 \le x_1 \le 102, 33 \le x_2 \le 45, 27 \le x_j \le 45,\ j = 3,4,5. \end{aligned}$$

5.5 Test problem 5

Tension/Compression String, This problem, is described by Arora [9], Coello [51] and Belegundu [52], and it consists of minimizing the weight of a tension/compression spring subject to constraints on minimum deflection, shear stress, surge frequency, limits on outside diameter and on design variables. The design variables are the wire diameter $d=x_1$, the mean coil diameter $D= x_2$, and the number of active coils $N=x_3$ .Formally, the problem can be expressed as:

$$\min\ f(x) = (x_3 + 2)x_2x_1^2, \quad (9)$$

$$\begin{aligned} &s.t. \\ &g_1(x) = 1 - \frac{x_2^3x_3}{71785x_1^4} \le 0, \\ &g_2(x) = \frac{4x_2^2 - x_2x_1}{2566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \le 0, \\ &g_3(x) = 1 - \frac{140.45x_1}{x_2^2x_3} \le 0, \\ &g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \le 0, \\ &0.05 \le x_1 \le 2, 0.25 \le x_2 \le 1.3, 2 \le x_3 \le 15 \end{aligned}$$

Table 1 lists the best solution of Tension/Compression String problem obtained by the FPA-GA algorithm, and compares them with previous best solutions reported by Belegundu [52], Arora [9], Coello [51], Mahdavi et al. [53], Shi and Eberhart[41].

5.6 Test problem 6

The pressure vessel design was previously analyzed by Sandgren [54] who first proposed this problem. The objective is to minimize the total cost *f(x)* including the cost of the material, forming and welding. There are four design variables: x_1 (T_s, shell thickness), x_2 (T_h, spherical head thickness), x_3 (R, 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 $40\le R \le 80$ in. and $20\le L \le 60$ in., respectively. The mathematical formulation of the optimization problem can be stated as follows:

$$\min\ f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (10)$$

$$\begin{aligned} &s.t. \\ &g_1(x) = -x_1 + 0.193x_3 \le 0, \\ &g_2(x) = -x_2 + 0.00954x_3 \le 0, \\ &g_3(x) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0, \\ &g_4(x) = x_4 - 240 \le 0, \\ &g_5(x) = 1.1 - x_1 \le 0, \\ &g_6(x) = 0.6 - x_2 \le 0, \end{aligned}$$

The FPA-GA algorithm was applied to the pressure vessel optimization problem and the optimal results were compared to earlier solutions reported by Sandgren [54] and Wu and Chow [55], Geem [46] and Mahdavi et al. [53], as shown in Table 1.

5.7 Test problem 7

Heat Exchanger Design 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, and has been solved previously by Deb [47], Michalewicz [56], Joines et al. [57], Lee and Geem[46]. The results show in Table 1.The problem formulation is:

$$\min\ f(x) = x_1 + x_2 + x_3, \quad (11)$$

$$\begin{aligned} &s.t. \\ &g_1(x) = 0.0025(x_4 + x_6) - 1 \le 0, \\ &g_2(x) = 0.0025(x_5 + x_7 - x_4) - 1 \le 0, \\ &g_3(x) = -1 - 0.01(x_8 - x_5) \ge 0, \\ &g_4(x) = x_1x_6 - 833.33252x_4 - 100x_1 + 83333.333 \ge 0, \\ &g_5(x) = x_2x_7 - 1250x_5 - x_2x_4 + 1250x_4 \ge 0, \\ &g_6(x) = x_3x_8 - x_3x_5 + 2500x_5 - 1250000 \ge 0, \\ &100 \le x_1 \le 10000.1000 \le x_2, x_3 \le 10000, 10 \le x_j \le 1000, (j = 4,...,8) \end{aligned}$$

6 Conclusions

In the present study, FPA-GA algorithm has been employed to solve constrained optimization problems. FPA-GA has been validated using several benchmark mathematical and engineering design problems. To verify the performance of FPA-GA, seven benchmark optimization problems chosen from the literature are employed. The results show that the proposed FPA-GA algorithm clearly outperforms the basic FPA and GA. Compared with some evolution algorithms from literature, we find that our algorithm is superior to or at least highly competitive with these algorithms. In the last, experiments have been conducted on Lorenz system and

Chen system. Simulation results and comparisons demonstrate the proposed method is very effective. The proposed FPA-GA algorithm can be extended to solve other problem such as combinatorial optimization problems.

Table 1: the best solution of proposed algorithm and other algorithms for solving constrained optimization problems.

Test problem	Optimum solution	The proposed algorithm		Other algorithms		
		The best solution	CPU time (s)	Name	The best solution	CPU time (s)
P1	1.3935	1.3935	0.11	Homaifar et al. [20]	1.4339	Unavailable
				Fogel [45]	2.3772	Unavailable
				Lee and Geem[46]	1.3770	Unavailable
P2	13.59085	13.59085	0.8	Lee and Geem [46]	13.590845	Unavailable
				Deb [47]	13.58958	Unavailable
				Mahdavi et al. [53]	13. 590841	Unavailable
P3	-	1.724840	0.71	Fesanghary et al. [30]	1.7248	4. 138
				Shi and Eberhart [41]	1.72485084	Unavailable
				Lee and Geem [46]	2.38	Unavailable
				Mahdaviet al. [53]	1.7248	Unavailable
				Coello [58]	1.7483	Unavailable
P4	-	-31025.56541	0.51	Fasanghary et al. [30]	-31024.316	1.306
				Shi and Eberhart [41]	-31025.56142	Unavailable
P5	-	0.0126657981091	0.86	Arora [9]	0.0127302737	Unavailable
				Shi and Eberhart [41]	0.0126661409	Unavailable
				Coello [51]	0.012681	Unavailable
				Belegundu [52]	0.0128334378	Unavailable
P6	-	6059.719980	1.017	Mahdavi et al.[53]	7197.730	Unavailable
				Lee and Geem[46]	7198.433	Unavailable
				Wu and Chow[55]	7207.494	Unavailable
				Sandgren [54]	7980.894	Unavailable
P7	7049.3307	7049.3307	1.48	Lee and Geem [46]	7057.274414	Unavailable
				Deb [47]	7060.221	Unavailable
				Michalewicz [56]	7377.976	Unavailable
				Joines [57]	7068.6880	Unavailable

Table 2: Best Results for the benchmark problems by FPA-GA

Test problem	Global optimal	Results of proposed FPA-GA				
		best	median	mean	worst	SD
P1	1.3935	1.3935	1.3935	1.3935	1.3935	2.1E−04
P2	13.59085	13.59085	13.59085	13.59080	13.59085	1.1E−14
P3	-	1.724840	1.724840	1.7248401	1.724840	0.00E+00
P4	-	-31025.56541	-31025.56541	-31025.56541	-31025.56541	0.00E+00
P5	-	0.0126657	0.0126657	0.0126657	0.0126657	1.35E−16
P6	-	6059.719980	6059.719980	6059.719980	6059.719980	0.00E+00
P7	7049.3307	7049.3307	7049.3307	7049.3307	7049.2216	1.3E−02

Acknowledgements

The authors would like to thank the anonymous reviewers for their careful reading of this paper and for their helpful comments.

References

[1] A. Ravindran, G. V. Reklaitis, and K. M. Ragsdell, *Engineering optimization: Methods and applications*: Wiley. com, (2006).

[2] S. S. Rao and S. Rao, *Engineering optimization: theory and practice*: John Wiley & Sons, (2009).

[3] L. Liberti and N. Maculan, *Global Optimization: Volume 84, From Theory to Implementation* vol. **84**: Springer, (2006).

[4] C. A. Floudas, *Deterministic global optimization: theory, methods and applications* vol. 37: Springer, (2000).

[5] O. Abdel-Raouf, M. Abdel-Baset, and I. El-henawy. "A New Hybrid Flower Pollination Algorithm for Solving Constrained Global Optimization Problems." *International Journal of Applied* vol. **4**, pp. 1-13, (2014).

[6] X.-S. Yang, *Engineering optimization: an introduction with metaheuristic applications*: Wiley. com, (2010).

[7] R. S. Parpinelli and H. S. Lopes, "New inspirations in swarm intelligence: a survey," *International Journal of Bio-Inspired Computation,* vol. **3**, pp. 1-16, (2011).

[8] X.-S.Yang,*Nature-inspired metaheuristic algorithms*: Luniver Press, (2010).

[9] J. Arora, *Introduction to optimum design*: Academic Press, (2004).

[10] D. A. Pelta and N. Krasnogor, "Nature-inspired cooperative strategies for optimization," *International Journal of Intelligent Systems,* vol. **24**, pp. 723-725, (2009).

[11] X.-S. Yang, "Review of meta-heuristics and generalised evolutionary walk algorithm," *International Journal of Bio-Inspired Computation,* vol. **3**, pp. 77-84, (2011).

[12] A. H. Gandomi, X.-S. Yang, S. Talatahari, and S. Deb, "Coupled eagle strategy and differential evolution for unconstrained and constrained global optimization," *Computers & Mathematics with Applications,* vol. **63**, pp. 191-200, (2012).

[13] A. H. Gandomi and A. H. Alavi, "Krill herd: a new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation,* vol. **17**, pp. 4831-4845, (2012).

[14] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: charged system search," *Acta Mechanica,* vol. **213**, pp. 267-289, (2010).

[15] X.-S. Yang, "Harmony search as a metaheuristic algorithm," in *Music-inspired harmony search algorithm*, ed: Springer, pp. 1-14, (2009).

[16] X.-S. Yang and A. H. Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Engineering Computations,* vol. **29**, pp. 464-483, (2012).

[17] A. H. Gandomi, X.-S. Yang, A. H. Alavi, and S. Talatahari, "Bat algorithm for constrained optimization tasks," *Neural Computing and Applications,* pp. 1-17, (2013).

[18] 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).

[19] X.-S. Yang and S. Deb, "Engineering optimisation by cuckoo search," *International Journal of Mathematical Modelling and Numerical Optimisation,* vol. **1**, pp. 330-343, (2010).

[20] A. Homaifar, C. X. Qi, and S. H. Lai, "Constrained optimization via genetic algorithms," *Simulation,* vol. **62**, pp. 242-253, (1994).

[21] Z. Michalewicz, *Genetic algorithms+ data structures= evolution programs*: springer, (1996).

[22] K. Deep, "A self-organizing migrating genetic algorithm for constrained optimization," *Applied Mathematics and Computation,* vol. **198**, pp. 237-250, (2008).

[23] C.-l. Sun, J.-c. Zeng, and J.-s. Pan, "An improved vector particle swarm optimization for constrained optimization problems," *Information Sciences,* vol. **181**, pp. 1153-1163, (2011).

[24] I. G. Tsoulos, "Solving constrained optimization problems using a novel genetic algorithm," *Applied Mathematics and Computation,* vol. **208**, pp. 273-283, (2009).

[25] J. Brest, A. Zamuda, I. Fister, and M. S. Maucec, "Large scale global optimization using self-adaptive differential evolution algorithm," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pp. 1-8, (2010).

[26] J.-Y. Wu, "Solving constrained global optimization via artificial immune system," *International Journal on Artificial Intelligence Tools,* vol. **20**, pp. 1-27, (2011).

[27] L. A. Zadeh, "Fuzzy logic, neural networks, and soft computing," *Communications of the ACM,* vol. **37**, pp. 77-84, (1994).

[28] J. Peters, "Computational Intelligence: Principles, Techniques and Applications," *The Computer Journal,* vol. **50**, pp. 758-758, (2007).

[29] E.-G. Talbi, *Metaheuristics: from design to implementation* vol. **74**: John Wiley & Sons, (2009).

[30] M. Fesanghary, M. Mahdavi, M. Minary-Jolandan, and Y. Alizadeh, "Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems," *Computer methods in applied mechanics and engineering,* vol. **197**, pp. 3080-3091, (2008).

[31] P. Xu, "A hybrid global optimization method: the multi-dimensional case," *Journal of computational and applied mathematics,* vol. **155**, pp. 423-446, (2003).

[32] L. Guo, "A novel hybrid bat algorithm with harmony search for global numerical optimization," *Journal of Applied Mathematics,* vol. **2013**, (2013).

[33] M. Lozano, D. Molina, and F. Herrera, "Editorial scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems," *Soft Computing,* vol. **15**, pp. 2085-2087, (2011).

[34] D. Molina, M. Lozano, and F. Herrera, "MA-SW-Chains: Memetic algorithm based on local search chains for large

scale continuous global optimization," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pp. 1-82010.

[35] A. LaTorre, J. M. Peña, S. Muelas, and M. Zaforas, "Hybrid evolutionary algorithms for large scale continuous problems," in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pp. 1863-1864, (2009).

[36] R. Kuo and Y. Han, "A hybrid of genetic algorithm and particle swarm optimization for solving bi-level linear programming problem–A case study on supply chain model," *Applied Mathematical Modelling,* vol. **35**, pp. 3905-3917, (2011).

[37] P. Shelokar, P. Siarry, V. K. Jayaraman, and B. D. Kulkarni, "Particle swarm and ant colony algorithms hybridized for improved continuous optimization," *Applied mathematics and computation*, vol. **188**, pp. 129-142, (2007).

[38] P. Xu, "A hybrid global optimization method: the one-dimensional case," *Journal of computational and applied mathematics*, vol. **147**, pp. 301-314, (2002).

[39] X.-S. Yang, "Flower pollination algorithm for global optimization," in *Unconventional Computation and Natural Computation*, ed: Springer, pp. 240-249, (2012).

[40] O. Raouf, I. El-henawy, and M. Abdel-Baset. "A novel hybrid flower pollination algorithm with chaotic harmony search for solving Sudoku puzzles." *International Journal of Modern Education and* Computer *Science vol.* **3**, pp. 38-44, (2014).

[41] O. Abdel-Raouf, M. Abdel-Baset, and I. El-henawy. "An Improved Flower Pollination Algorithm with Chaos." (2014).

[42] M. Abdel-Baset and I. Hezam, "An Improved Flower Pollination Algorithm based on Simulated Annealing for Solving Engineering Optimization Problems," Asian Journal of Mathematics and Computer Research, vol. **3**, Issue: 3, (2015).

[43] M. Abdel-Baset and I. Hezam "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).

[44] J. Bracken and G. P. McCormick, "Selected applications of nonlinear programming," DTIC Document, (1968).

[45] D. B. Fogel, "A comparison of evolutionary programming and genetic algorithms on selected constrained optimization problems," *Simulation,* vol. **64**, pp. 397-404, (1995).

[46] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice," *Computer methods in applied mechanics and engineering,* vol. **194**, pp. 3902-3933, (2005).

[47] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer methods in applied mechanics and engineering,* vol. **186**, pp. 311-338, (2000).

[48] K. Ragsdell and D. Phillips, "Optimal design of a class of welded structures using geometric programming," *Journal of Engineering for Industry,* vol. **98**, p. 1021, (1976).

[49] J. N. Siddall, *Analytical decision-making in engineering design*: Prentice-Hall Englewood Cliffs, NJ, (1972).

[50] D. M. Himmelblau, B. Clark, and M. Eichberg, *Applied nonlinear programming* vol. 111: McGraw-Hill New York, (1972).

[51] C. A. Coello Coello and E. Mezura Montes, "Constraint-handling in genetic algorithms through the use of dominance-based tournament selection," *Advanced Engineering Informatics,* vol. **16**, pp. 193-203, (2002).

[52] A. D. Belegundu and J. S. Arora, "A study of mathematical programming methods for structural optimization. Part I: Theory," *International Journal for Numerical Methods in Engineering,* vol. **21**, pp. 1583-1599, (1985).

[53] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Applied mathematics and computation,* vol. **188**, pp. 1567-1579, (2007).

[54] E. Sandgren, "Nonlinear integer and discrete programming in mechanical design optimization," *Journal of Mechanical Design,* vol. **112**, p. 223, (1990).

[55] S.-J. Wu and P.-T. Chow, "Genetic algorithms for nonlinear mixed discrete-integer optimization problems via meta-genetic parameter optimization," *Engineering Optimization+ A35,* vol. **24**, pp. 137-159, (1995).

[56] Z. Michalewicz, "Genetic algorithms, numerical optimization, and constraints," in *Proceedings of the Sixth International Conference on Genetic Algorithms*, pp. 151-158, (1995).

[57] J. A. Joines and C. R. Houck, "On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's," in *Evolutionary Computation, IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pp. 579-584, (1994).

[58] C. A. Coello Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Computers in Industry,* vol. **41**, pp. 113-127, (2000).