Mergos, P.E., Yang, XS. Flower pollination algorithm with pollinator attraction. Evolutionary Intelligence (2022). https://doi.org/10.1007/s12065-022-00700-7

Flower Pollination Algorithm with Pollinator Attraction

Panagiotis E. Mergos [a,*], Xin-She Yang [b]

[a] *Department of Civil Engineering, City, University of London, London EC1V 0HB, UK*

[b] *Design Engineering and Mathematics, Middlesex University London, The Burroughs, London, UK*

Abstract.

The Flower Pollination Algorithm (FPA) is a highly efficient optimization algorithm that is inspired by the evolution process of flowering plants. In the present study, a modified version of FPA is proposed accounting for an additional feature of flower pollination in nature that is the so-called pollinator attraction. Pollinator attraction represents the natural tendency of flower species to evolve in order to attract pollinators by using their colour, shape and scent as well as nutritious rewards. To reflect this evolution mechanism, the proposed FPA variant with Pollinator Attraction (FPAPA) provides fitter flowers of the population with higher probabilities of achieving pollen transfer via biotic pollination than other flowers. FPAPA is tested against a set of 28 benchmark mathematical functions, defined in IEEE-CEC'13 for real-parameter single-objective optimization problems, as well as structural optimization problems. Numerical experiments show that the modified FPA represents a statistically significant improvement upon the original FPA and that it can outperform other state-of-the-art optimization algorithms offering better and more robust optimal solutions. Additional research is suggested to combine FPAPA with other modified and hybridized versions of FPA to further increase its performance in challenging optimization problems.

Keywords: Flower pollination algorithm; Pollinator attraction; Metaheuristics; Evolutionary; Optimization

1 Introduction

In plenty of challenging optimization problems in industry and engineering, tracking of global optimum solutions remains a highly complex task. Conventional optimization methods are often not performing satisfactorily in this category of problems, and thereby the application of metaheuristic algorithms inspired by nature is required [1]. In the literature, a significant

---

* Corresponding author. Panagiotis E. Mergos, Lecturer in Structural Engineering, Research Centre for Civil Engineering Structures, City, University of London, London, EC1V 0HB, UK.

E-mail address: panagiotis.mergos.1@city.ac.uk

number of efficient metaheuristic optimisation algorithms have been proposed, including the Genetic Algorithm (GA) [2], Firefly Algorithm [3], Particle Swarm Optimization (PSO) [4], Cuckoo Search (CS) [5], and several others.

Recently, the Flower Pollination Algorithm (FPA) was developed by Xin-She Yang [6], which is a population-based metaheuristic optimization algorithm inspired by the evolution process of flowering plants. FPA is characterised by formulation simplicity and flexibility as well as high computational performance [7]. Furthermore, many studies show that it can outperform other metaheuristic optimization algorithms (e.g. [6, 8-10]). As a result, FPA has been adopted by many optimization studies and it has been applied successfully to many optimization problems in diverse scientific areas including electrical and power systems (e.g. [11-13]), structural design (e.g. [8-10, 14-15]), computer gaming (e.g. [16]), meteorology (e.g. [17]), image science (e.g. [18]) and others [7, 19].

Following its original development, several studies proposed modified and hybridized versions of FPA to improve its performance for different optimization problems [7, 19]. For example, Abdel-Raouf *et al*. [20] developed an improved FPA variant by using chaotic maps instead of random numbers and they found significant increase in the computational performance. Zhou *et al*. [21] developed an elite opposition-based FPA version that was tested with 18 benchmark functions yielding excellent results. Putra *et al*. [22], developed a modified version of FPA with dynamic switching probability and the use of real-coded GA as mutation for local and global search to solve economic load dispatch optimization problems in power generation systems. Draa [23] developed a new FPA variant based on the so-called generalized opposition-based learning (GOBL). Wang *et al*. [24] merged the standard FPA with the concept of the bee-pollinator to solve the data clustering problem. Al-Betar *et al*. [25] used the island model population technique to restrain premature convergence of FPA. Abdel-Basset *et al*. [26] developed a modified FPA version based on the crossover for solving the multidimensional knapsack problems. Zhou *et al*. [27] developed the discrete greedy flower pollination algorithm that is using order-based crossover, pollen discarding behaviour and partial behaviours for solving the spherical traveling salesman problem. Fouad and Gao [28] developed a novel FPA variant for global optimization by generating a set of global orientations for all members of the population and constructing a set of best solution vectors relating to all generated global orientations. Khurseed et al. [29] used a modified FPA with double exponential based dynamic switch probability and a dynamic step size function for model parameter estimation of Photovoltaic cells and modules. Xiao et al. [30] developed a modified FPA to solve the problem of robust visual target tracking system. The proposed variant, namely GTFPA, is based on the gravitational search algorithm and features an improved local search by a mutation operation based on trigonometric functions. Ozsoydan and Baykasoglu [31] introduced a new FPA variant by embedding to the original FPA a chaotic switch with irregular motion and an intensifying step size function for a more detailed search. Furthermore, Ozsoydan and Baykasoglu [32] developed a new multi-population FPA variant for multimodal optimization problems and a modified natural selection based on symmetry. In addition to the above, Rodrigues *et al*. [33] developed a binary version of FPA to address combinatorial and discrete optimization problems. Multi-objective versions of FPA have also been developed (e.g. [34-36]) to solve optimization problems with more than one design objectives. In addition, hybridized FPA versions have been proposed in literature to achieve better balance between local and global search. In these versions, hybridization of FPA is achieved using local search algorithms (e.g. [37-38]), population-based algorithms (e.g. [39-42]) or other components.

An explanation of the efficiency of FPA is based on the fact that it is imitating the reproduction process of flowering plants. The latter has been so successful that flower species dominate the landscape of earth [43]. As with other biological systems, the ultimate objective of flowers is reproduction via pollination. Flower pollination, which is typically related to the transfer of pollen, can be either biotic or abiotic [6, 44]. In the former pollination type, pollen is transferred via animals and insects (e.g. bees, butterflies, birds and bats) that are called pollinators. Pollinators are able to fly long distances. Hence, biotic pollination can be considered as a global pollination mechanism [6]. In addition, the flight behaviour of pollinators has characteristics of Lévy flights [1, 45]. In the abiotic pollination type, pollen is transferred by water diffusion and/or the wind. A characteristic example of abiotic pollination is the grass [6, 44]. Typically, abiotic pollination takes place at short distances. Therefore, it can be considered as a local pollination mechanism [6]. Another significant feature of flower pollination is the so-called flower constancy. According to this feature, some pollinators prefer to select specific flower species and bypass others [6]. In this manner, flowers increase pollen transfer to the same species. Furthermore, pollinators ensure guaranteed nectar intake and avoid the risk of exploring other flower species.

All previous characteristics of flower pollination have been considered in the formulation of the original FPA [6]. However, an additional important characteristic of the flower pollination process in nature is the fact that flower species evolve to attract pollinators and ensure pollen transfer via biotic pollination [46-47]. To serve this goal, flowers entice pollinators by employing a variety of attractions. For example, they offer pollinators nutritious rewards such as pollen and nectar. Pollinators eat pollen to produce their eggs. Furthermore, nectar offers significant amounts of energy to pollinators. Honeybees, in particular, use nectar to produce honey. In addition to nutritious rewards, flowers have developed other methods to attract pollinators. Many flowers have developed shapes (e.g. bowl-shaped flowers) that facilitate unique access of certain types of pollinators [47]. Furthermore, some flowers have developed bright colours to attract pollinators with colour vision such as bees and birds [46-47]. In addition, flowers attract pollinators by scent. It is interesting to note that flowers relying on night pollinators (e.g. bats) focus mainly on scent to entice pollinators and most of them are colourless [46-47]. It is such the need of some plants to attract pollinators that they produce flowers resembling female pollinators in colour, shape and scent such as the case with orchids and bees [47]. It is clear from the above that flower species do evolve to attract pollinators; the more successful they are in this evolution process, the more likely it is that they will transfer their pollen via biotic pollination [47].

In our proposed approach, the well-observed and successful in nature evolution mechanism of pollinator attraction will be introduced to the mathematical formulation of the existing FPA algorithm by increasing the probability of fitter flowers to conduct biotic pollination as they are more attractive to pollinators. The resulting variant, namely FPAPA (Flower Pollination Algorithm with Pollinator Attraction), is then compared with the original FPA and other efficient optimization algorithms to establish its computational performance against benchmark mathematical functions and real-world optimization problems.

In the following, the original FPA is introduced in §2. In §3, the proposed modifications to the formulation of the original FPA are described to account for the pollinator attraction evolution mechanism. In §4, the proposed FPAPA is compared with the original FPA and other well-established optimization algorithms against mathematical and structural optimization problems to test its computational efficiency. In §5, the main conclusions of the present study are summarized.

2 Original FPA

The types of flower pollination process, the behaviour of pollinators and flower constancy have been idealized in the following basic rules of the original version of FPA:

1. Biotic pollination is assumed as a global pollination process with pollinators performing Lévy flights.

2. Abiotic pollination is assumed as a local pollination mechanism.
3. Flower constancy is considered by assuming the reproduction probability to be proportional of the similarity of flowers involved.
4. The mechanism of pollination mechanism (global or local) is controlled by a switching probability *p* in [0, 1].

In the following, for reasons of simplicity, it is assumed that each plant develops one flower, which produces only one pollen gamete [6]. Under this assumption, there exists no need to differentiate between plants, flowers and pollen gametes. In FPA, a flower *i* represents a candidate solution vector $\boldsymbol{x_i}$. The algorithm employs two separate search procedures or search mechanisms. The global and local pollination. Following the first and third rules of FPA, the global pollination procedure can be represented mathematically by the following equation:

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \gamma \cdot L(\lambda) \cdot (\boldsymbol{g}^* - \boldsymbol{x}_i^t), \quad (1)$$

where $\boldsymbol{x}_i^t$ stands for flower *i* at iteration *t*, $\boldsymbol{g}^*$ represents the best flower of the population again at iteration *t*, λ is a constant, γ is a scaling factor to set the step size, and $L(\lambda) > 0$ represents the size of the flight step reflecting pollination strength. More particularly, $L(\lambda)$ is taken from a Lévy distribution as follows:

$$L \sim \frac{\lambda \Gamma(\lambda) \sin\left(\frac{\pi\lambda}{2}\right)}{\pi} \cdot \frac{1}{s^{1+\lambda}}, \quad (s > 0), \quad (2)$$

where $\Gamma(\lambda)$ is the standard gamma function and s > 0. In the present study, based on a preliminary parametric analysis and recommendations in literature [6], it is assumed that $\lambda = 3/2$ and $\gamma = 0.01$ as these values yielded the best performance of the algorithm.
On the other hand, the local pollination rule (second rule) and flower constancy (third rule) are represented by the following equation, where $\boldsymbol{x}_j^t$ and $\boldsymbol{x}_k^t$ are different flowers of the same population and ε is drawn from a uniform distribution in [0, 1].

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \varepsilon \cdot \left(\boldsymbol{x}_j^t - \boldsymbol{x}_k^t\right). \quad (3)$$

```
Set objective min f (x), x = (x1, x2, …, xd)
Initialize a population of n flowers with random procedures
Determine the best solution g* of the initial population
Determine the value of switch probability p ϵ [0, 1]
while (t < MaxIteration)
        for i = 1 : n (for all flowers of the population)
            if rand < p
                Draw a d-dimensional Lévy distribution step vector L
                Do global pollination by x_i^{t+1} = x_i^t + γ · L(λ) · (g* − x_i^t)
            else
                Draw ε from a uniform distribution in [0, 1]
                Select randomly j and k among all flowers of the population
                Do local pollination by x_i^{t+1} = x_i^t + ε · (x_j^t − x_k^t)
            end if
            Evaluate objective function values of new solutions
            When better, update new solutions in the population
        end for
        Determine the best solution g* of the new population
end while
```

Fig. 1: Pseudo-code of the original FPA.

According to the fourth rule, the type of flower pollination (global or local) is controlled by a switch probability p in [0, 1]. In §4 of this research, a parametric study is conducted to establish the values of p that yield the best performance of FPA algorithm.

Summarizing the previous information, the pseudo code of FPA is presented in Fig. 1, where d is the number of problem dimensions and n the size of flowers population.

3 FPA with Pollinator Attraction (FPAPA)

From the discussion above, it is concluded that the original version of FPA provides all flowers with the same switch probability p of transferring pollen by biotic or abiotic pollination. However, as discussed in the introduction section, many flower species have developed evolution mechanisms to attract pollinators and achieve pollen transfer via biotic pollination. Hence, it is expected that the fitter flowers (i.e. the ones that have developed more efficient attraction mechanisms) will have higher probabilities of attracting pollinators and conducting biotic pollination than the other flowers.

The simplest, perhaps, way to model this observation in FPAPA is to assume that the switch probability p_i^t of flower i at iteration t is not the same for all flowers of the population but it depends on the rank of the flower i in the population in terms of the objective function value $f(\boldsymbol{x}_i)$. For simplicity, it is assumed herein that this probability varies linearly (Fig. 2) between two values p_1 and p_2 in [0, 1], where p_1 is the switch probability of the flower with the worst objective function value and p_2 is the respective probability of the flower with the best objective function value in the population. Hence, if $\boldsymbol{f}_{sh}$ is the sorted, in descending order for minimization problems, vector of objective function values of the population and ξ_i^t is the index (location) of flower i in $\boldsymbol{f}_{sh}$ at iteration t, then the probability p_i^t is given by:

$$p_i^t = \frac{1}{n-1}[(p_2 - p_1) \cdot \xi_i^t + np_1 - p_2] \tag{4}$$

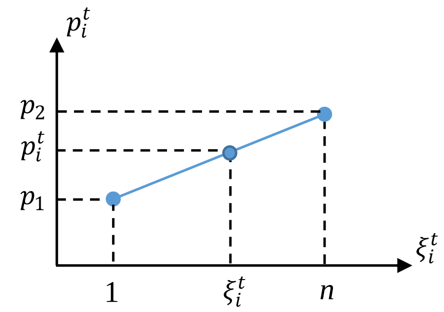

Fig. 2: Variation of switch probabilities p_i^t for different flowers (i = 1 to n) of the population based on their indices ξ_i^t in the sorted population in the descending order vector $\boldsymbol{f}_{sh}$ (i.e. index $\xi_i^t = 1$ is the flower with the maximum objective function value and index $\xi_i^t = n$ is the flower with the minimum respective value).

Set objective min $f(\boldsymbol{x})$, $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)$
Initialize a population of n flowers with random procedures
Form shorted, in descending order, objective values vector $\boldsymbol{f}_{sh}$ of the initial population
Determine the best solution $\boldsymbol{g}^*$ of the initial population
Determine the values of switch probabilities p_1 *and* p_2 in [0, 1]
while (t < *MaxIteration*)
    for i = 1 : n (for all flowers of the population)
        Find flower indices ξ_i^t in $\boldsymbol{f}_{sh}$
        Determine flowers switch probability by $p_i^t = \frac{1}{n-1}[(p_2 - p_1) \cdot \xi_i^t + np_1 - p_2]$
        if $rand < p_i^t$
            Draw a d-dimensional Lévy distribution step vector L
            Do global pollination by $\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \gamma \cdot L(\lambda) \cdot (\boldsymbol{g}^* - \boldsymbol{x}_i^t)$
        else
            Draw ε from a uniform distribution in [0, 1]
            Select randomly j and k among all flowers of the population
            Do local pollination by $\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \varepsilon \cdot (\boldsymbol{x}_j^t - \boldsymbol{x}_k^t)$
        end if
        Evaluate objective function values of new solutions
        When better, update new solutions in the population
    end for
    Form shorted, in descending order, objective values vector $\boldsymbol{f}_{sh}$ of the new population
    Determine the best solution $\boldsymbol{g}^*$ of the new population
end while

Fig. 3: Pseudo-code of FPAPA

In Eq. (4), to be consistent with the pollinator attraction evolution mechanism, $p_2 > p_1$ should hold (i.e. maximum probability for the fittest flower). However, the algorithm works for any p_1 and p_2 values in [0, 1]. Furthermore, for $p_1 = p_2$ the original FPA is derived. To accommodate the aforementioned pollinator attraction rule, the pseudo-code of the FPAPA version is shown in Fig. 3.

4 Numerical Simulations

4.1 Mathematical optimization problems

In this section, the proposed FPAPA algorithm is compared with the original FPA and other state of the art optimization algorithms in order to validate its numerical efficiency. To serve this goal, the set of functions specified in IEEE-CEC'13 [48] for real-parameter single-objective optimization problems is employed herein. This set is comprised of 28 benchmark functions f_i (i =1, 2, …, 28) shown in Table 1 [48] together with their global optimum values. All f_i functions represent minimization problems with variable number of dimensions d. All test functions are scalable and shifted to $\boldsymbol{o} = [o_1, o_2, \ldots, o_d]$, which is randomly distributed in $[-80, 80]^d$. Moreover, the search space for all functions is defined in $[-100, 100]^d$. In addition, some functions are rotated by using orthogonal (rotation) matrices that are generated from standard normally distributed entries by the Gram-Schmidt orthonormalization. The test functions can be classified in three main categories: unimodal, basic multimodal and compositions functions that are generated by combinations of the former functions [48].

Table 1: IEEE-CEC'13 benchmark functions

	Function No.	Function Name	Global optimum f_i^*
Unimodal	1	Sphere Function	-1400
	2	Rotated High Conditioned Elliptic Function	-1300
	3	Rotated Bent Cigar Function	-1200
	4	Rotated Discus Function	-1100
	5	Different Powers Function	-1000
Basic Multimodal	6	Rotated Rosenbrock's Function	-900
	7	Rotated Schaffers F7 Function	-800
	8	Rotated Ackley's Function	-700
	9	Rotated Weierstrass Function	-600
	10	Rotated Griewank's Function	-500
	11	Rastrigin's Function	-400
	12	Rotated Rastrigin's Function	-300
	13	Non-Continuous Rotated Rastrigin's Function	-200
	14	Schwefel's Function	-100
	15	Rotated Schwefel's Function	100
	16	Rotated Katsuura Function	200
	17	Lunacek Bi_Rastrigin Function	300

	18	Rotated Lunacek Bi_Rastrigin Function	400
	19	Expanded Griewank's plus Rosenbrock's Function	500
	20	Expanded Scaffer's F6 Function	600
Composite Multimodal	21	Composition Function 1	700
	22	Composition Function 2	800
	23	Composition Function 3	900
	24	Composition Function 4	1000
	25	Composition Function 5	1100
	26	Composition Function 6	1200
	27	Composition Function 7	1300
	28	Composition Function 8	1400

For each algorithm, 20 independent runs are conducted for each function with 10000·d maximum number of function evaluations MaxFES and by using a population size of 50 flowers. Furthermore, two different numbers of problem dimensions are examined: d = 10 and 30. Uniform random initialization within the search space is assumed. For each algorithm run, the error value (i.e. the best solution found by the algorithm minus the global optimum of the test function shown in Table 1) is recorded when the number of function evaluations becomes equal to (0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0)·MaxFES. In this manner, the speed of convergence of the different algorithmic solutions can also be assessed. The solutions are terminated when either MaxFES is reached or the error value is smaller than 10^{-8}. For each function, algorithm and number of function evaluations, the mean and standard deviation of the error values are calculated from the 20 independent runs. The algorithms are subsequently ranked according to these mean and standard deviation values.

To better illustrate this procedure, the example of function f_{11} (i.e. Rastrigin's Function) for d = 10 is presented in the following. Figs 4a & 4b present, in the form of box plots, the minimum, maximum and median (red line) errors obtained by the 20 independent runs by 5 different FPA options (i.e. FPA with p = 0.2, 0.4 and 0.6 and FPAPA with p_1 = 0.2 and p_2 = 0.6 or p_1 = 0.6 and p_2 = 0.2) after 0.2·MaxFES and 1.0·MaxFES function evaluations respectively. Inside the boxes, the 25th to 75th percentile solutions are contained. It is found that the proposed FPAPA with p_1 = 0.2 and p_2 = 0.6 demonstrates the best performance out of the 5 FPA variants in terms of both median and minimum errors and for both numbers of function evaluations. Furthermore, Fig. 4c presents the progression of the mean prediction errors for the same optimization task and algorithms as a function of the computational budget as measured by the fraction of the number of function evaluations with respect to MaxFES. It is observed that the proposed FPAPA with p_1 = 0.2 and p_2 = 0.6 offers better mean predictions for almost the full range of function evaluations. This is also illustrated in Fig. 4d that shows the ranks of the different FPA options for the case of f_{11} with d = 10 based on the mean prediction errors shown in Fig. 4c and in relation to the number of function evaluations.

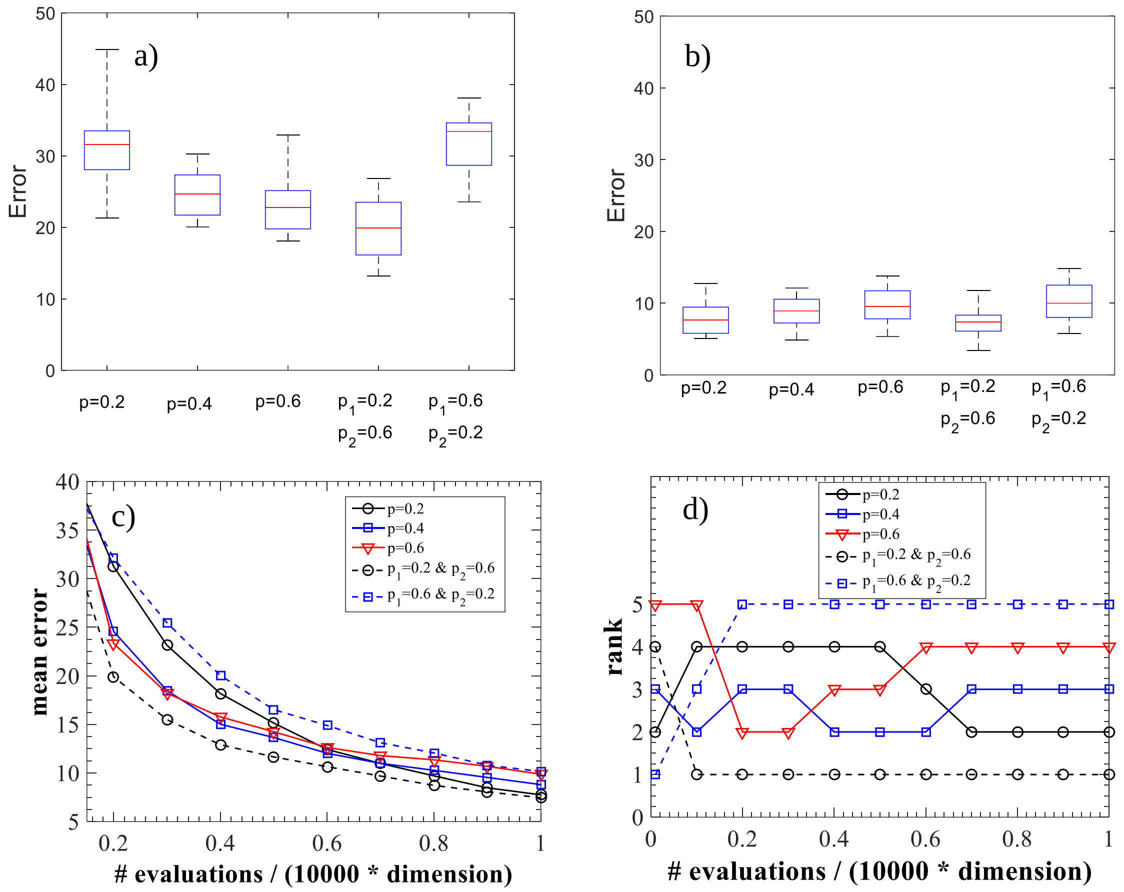

Fig. 4: a) Box plots of error predictions after 0.2·MaxFES evaluations; b) box plots of error predictions after 1.0·MaxFES evaluations; c) mean prediction errors; d) ranks of the original FPA and the proposed FPAPA algorithms for the f_{11} test function with d =10

In the following, the results of FPAPA are compared with the results of the original FPA for different switch probability values and across the whole range of CEC'13 functions. To serve this goal, parametric analyses are first conducted with the original FPA for switch probability values: p = 0, 0.2, 0.4, 0.6, 0.8 and 1. The mean aggregated rank of these p values across all 28 CEC'13 test functions is shown in Fig. 5 for both d = 10 and 30 problem dimensions in dependence of the number of function evaluations. It is clear that the original FPA performs better for probability values p between 0 and 0.4 for d = 10 and between 0.2 and 0.6 for d = 30. In both cases, the worst performance is obtained for p = 1 that sets the algorithm to conduct only global and no local pollination.

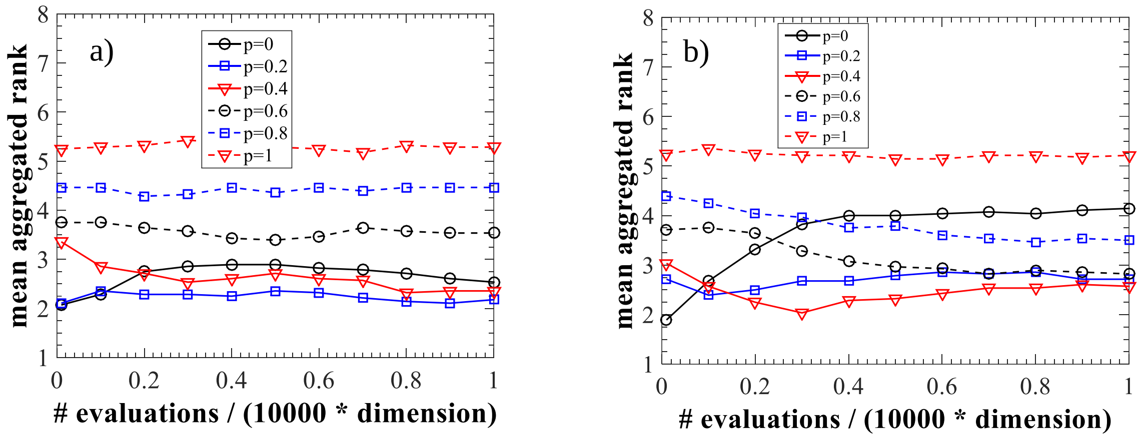

Fig. 5: Mean aggregated rank of original FPA p values across all 28 CEC'13 test functions for a) d = 10; b) d = 30 problem dimensions

Based on the previous results, numerical analyses with the proposed FPAPA are conducted assuming $p_1 = 0$ and $p_2 = 0.4$ for $d = 10$ and $p_1 = 0.2$ and $p_2 = 0.6$ for $d = 30$. It is recalled that this arrangement of switch probabilities (i.e. $p_2 > p_1$) supports the pollinator attraction evolution mechanism recommended in this study (i.e. higher probability of biotic pollination for better flowers in the population). For comparison purposes, it is also examined herein the use of FPAPA with $p_1 = 0.4$ and $p_2 = 0$ for $d = 10$ and $p_1 = 0.6$ and $p_2 = 0.2$ for $d = 30$. The latter probability arrangements with $p_1 > p_2$ are opposed to the pollinator attraction rule and therefore it is interesting to see how they affect the efficiency of the proposed algorithm.

Figure 6 presents the mean aggregated ranks of the mean error values of the original FPA and the proposed FPAPA algorithms within the same probability ranges and across all 28 test functions for d = 10 and 30 problem dimensions in dependence of the number of function evaluations. It is obvious that the proposed FPAPA supporting the pollinator attraction rule (i.e. $p_2 > p_1$) outperforms the original FPA algorithm for all switch probability values and for both numbers of problem dimensions. It is also important to note that the proposed FPAPA demonstrates better computational performance from the very early stages of function evaluations which means that it exhibits higher convergence rates. On the other hand, the FPAPA algorithm with switch probability values opposing the pollinator attraction rule (i.e. $p_1 > p_2$) demonstrates one of the worst performances out of the different FPA options. The latter represents another strong indication that the proposed pollinator attraction evolution mechanism can indeed affect positively the efficiency of the FPA algorithm.

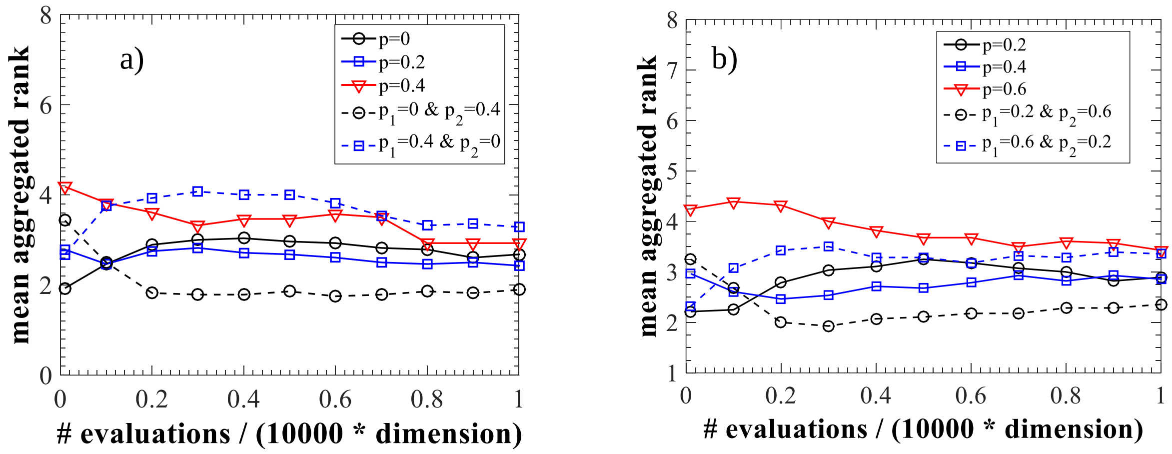

Fig. 6: Mean aggregated ranks of the mean error values of the original FPA and the proposed FPAPA algorithms across all 28 test functions for a) d =10; b) d = 30 problem dimensions

Furthermore, Fig. 7 shows the mean aggregated ranks of the standard deviations of the error values of the original FPA and the proposed FPAPA algorithms. It is evident that the proposed FPAPA not only exhibits better mean errors as shown in Fig. 6 but also outperforms or shows equivalent performance to the other FPA options in terms of computational robustness.

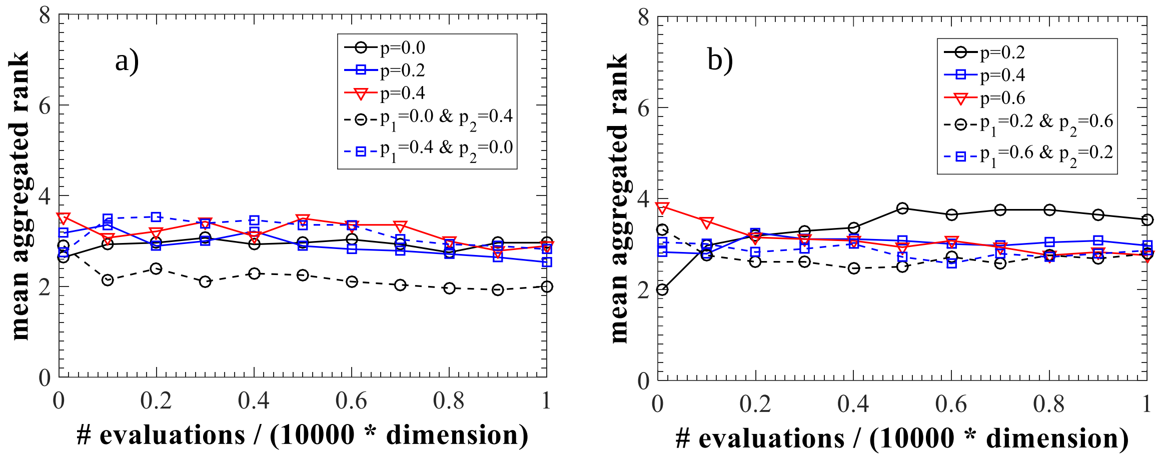

Fig. 7: Mean aggregated ranks of the standard deviations of the error values of the original FPA and the proposed FPAPA algorithms across all 28 test functions for a) d =10; b) d =30 problem dimensions

Moreover, Fig. 8 presents the mean aggregated ranks of the mean error values of the original FPA and the proposed FPAPA algorithms across the different categories of test functions (i.e. unimodal; multimodal and composite) for d = 10 and 30 problem dimensions. It can be concluded that the proposed algorithm FPAPA with $p_2 > p_1$ outperforms the original FPA algorithm for all categories of test functions and for both numbers of problem dimensions. The only exception is the case of unimodal functions for d = 10, where it appears that the standard FPA with p = 0 offers better numerical performance. This is explained by the fact that FPA with p = 0 conducts only local pollination and therefore may converge faster to the single optimum of these functions.

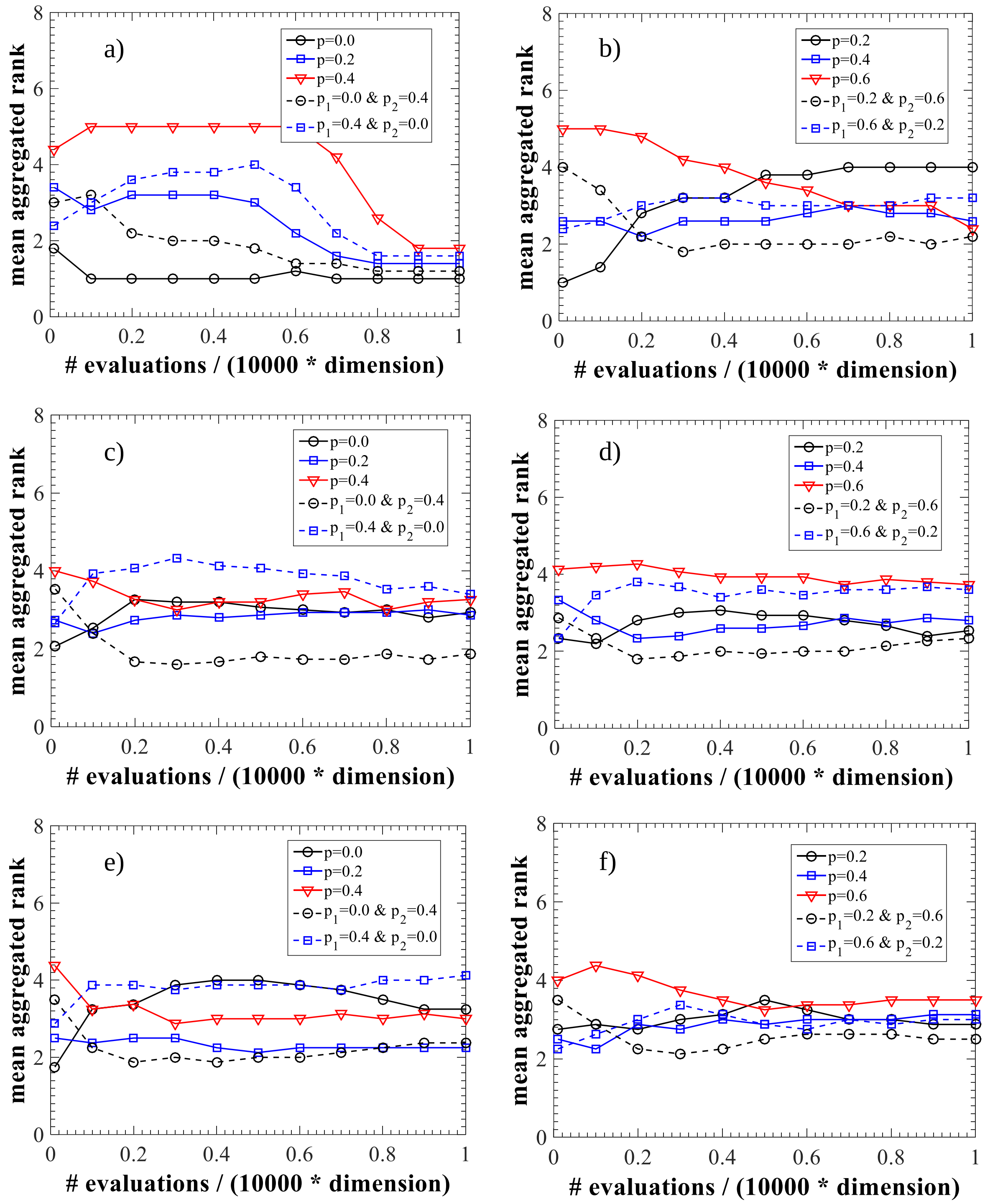

Fig. 8: Mean aggregated ranks of the mean error values of the original FPA and the proposed FPAPA algorithms across test functions: a) f_1-f_5 (unimodal) for d =10; b) f_1-f_5 (unimodal) for d = 30; c) f_6-f_{20} (basic multimodal) for d =10; d) f_6-f_{20} (basic multimodal) for d = 30; e) f_{21}-f_{28} (composite) for d =10; f) f_{21}-f_{28} (composite) for d = 30.

The mean error values of the original FPA and the proposed FPAPA algorithms of the 28 test functions for d = 10 after 10000·d function evaluations are shown in Table 2. The same results are presented in Table 3 for $d = 30$. As expected, the errors increase with the number of problem dimensions and appear to be more significant for the composite functions due to the higher degree of complexity involved.

Furthermore, Tables 4 and 5 present the Sign test pairwise comparisons of the proposed FPAPA with the original FPA options for MaxFES function evaluations. This test compares the performances of two algorithms by counting the number of wins of one algorithm with respect to the other [49]. For a given problem, an algorithm wins when the mean error observed is smaller than the other algorithm [49]. For n_p number of problems, if an algorithm wins $n_p/2 + 1.96\cdot\sqrt{n_p}/2$ times or above then the algorithm is considered significantly better than the other with level of significance $\alpha \leq 0.05$ [49]. The latter is a strong indication against the null hypothesis [28, 49]. For the IEEE-CEC'13 set of functions, $n_p = 28$ and therefore 19 wins are required for an algorithm to be significantly better than its rival. In Tables 4 and 5, the wins of the proposed FPAPA against the other FPA formulations are presented. It is noted that equivalences are split evenly between two algorithms in this table. It can be observed than in all cases the proposed FPAPA has 19 wins and above when compared with the other FPA options. Therefore, the proposed FPAPA can be considered as a significant improvement with respect to the other options.

In addition, Tables 6 and 7 show the Wilcoxon signed ranks test for the same algorithms [49]. This test calculates the differences of the performances of two algorithms for n_p problems and ranks these differences according to their absolute values. Next, the sum of ranks R^+ for the problems in which the first algorithm outperforms the second and the sum of ranks R^- of the opposite cases are calculated. If T is the minimum of R^+ and R^- and T is smaller or equal than the Wilcoxon's distribution for n_p degrees of freedom then the winning algorithm outperforms the other with the significance level associated [49]. In Tables 6 and 7, the R^+, R^- and T values for the comparisons between the proposed FPAPA and the original FPA options are presented for $d = 10$ and $d = 30$ respectively. In these tables, ties are split evenly among the sums. Again, it can be concluded that the proposed FPAPA offers significant improvement with respect to the other options with significance level $\alpha \leq 0.05$ since all T values are below the limit value, which is 116 for $\alpha = 0.05$ and $n_p = 28$.

Table 2: The mean error values of the original FPA and the proposed FPAPA algorithms of the 28 test functions for $d = 10$ after $10000\cdot d$ function evaluations

	FPA						FPAPA	
Function	$p = 0$	$p = 0.2$	$p = 0.4$	$p = 0.6$	$p = 0.8$	$p = 1.0$	$p_1 = 0.0$ $p_2 = 0.4$	$p_1 = 0.4$ $p_2 = 0.0$
1	1.00E-08	1.00E-08	1.00E-08	1.00E-08	2.15E-07	8.30E-02	1.00E-08	1.00E-08

2	1.00E-08	1.00E-08	1.00E-08	2.49E-08	2.75E-02	5.36E+05	1.00E-08	1.00E-08
3	1.15E+03	2.40E+03	1.75E+04	1.27E+05	7.90E+05	4.57E+07	2.11E+03	3.18E+03
4	1.00E-08	1.00E-08	1.00E-08	1.55E-08	1.10E-02	1.45E+04	1.00E-08	1.00E-08
5	1.00E-08	1.00E-08	1.00E-08	4.71E-08	6.50E-05	7.03E-01	1.00E-08	1.00E-08
6	1.00E-08	1.00E-08	1.00E-08	2.49E-07	1.75E-02	1.68E+00	1.00E-08	1.00E-08
7	3.15E+00	9.73E+00	1.44E+01	2.62E+01	4.17E+01	7.42E+01	7.81E+00	6.57E+00
8	2.05E+01	2.04E+01	2.03E+01	2.04E+01	2.04E+01	2.03E+01	2.04E+01	2.03E+01
9	4.28E+00	4.42E+00	4.96E+00	5.38E+00	5.81E+00	8.18E+00	4.71E+00	4.78E+00
10	3.44E-02	3.01E-02	2.92E-02	3.84E-02	8.32E-02	2.14E+00	2.72E-02	3.75E-02
11	7.76E+00	8.81E+00	9.87E+00	1.16E+01	1.56E+01	2.68E+01	7.46E+00	1.02E+01
12	6.49E+00	1.00E+01	1.13E+01	1.50E+01	2.52E+01	1.12E+02	8.21E+00	9.08E+00
13	1.25E+01	1.27E+01	1.54E+01	2.22E+01	2.86E+01	1.03E+02	1.16E+01	1.43E+01
14	9.16E+02	5.56E+02	4.87E+02	4.75E+02	4.64E+02	4.56E+02	4.84E+02	7.97E+02
15	8.60E+02	7.83E+02	7.73E+02	8.30E+02	8.69E+02	9.67E+02	7.33E+02	8.47E+02
16	9.73E-01	8.69E-01	8.96E-01	7.85E-01	8.43E-01	8.45E-01	8.88E-01	9.17E-01
17	3.05E+01	2.62E+01	2.43E+01	2.65E+01	3.17E+01	8.90E+01	2.12E+01	2.89E+01
18	3.07E+01	2.63E+01	2.75E+01	2.81E+01	3.83E+01	1.49E+02	2.44E+01	3.04E+01
19	1.13E+00	8.96E-01	8.60E-01	9.78E-01	1.04E+00	4.65E+00	7.37E-01	1.19E+00
20	3.03E+00	3.16E+00	3.21E+00	3.30E+00	3.50E+00	3.90E+00	3.11E+00	2.98E+00
21	1.45E+02	1.35E+02	1.45E+02	1.30E+02	1.35E+02	2.84E+02	1.35E+02	1.30E+02
22	9.06E+02	7.56E+02	6.67E+02	7.09E+02	7.18E+02	6.52E+02	6.69E+02	9.77E+02
23	9.36E+02	9.20E+02	9.51E+02	1.11E+03	1.21E+03	1.39E+03	9.20E+02	1.06E+03
24	1.33E+02	1.35E+02	1.51E+02	1.54E+02	1.70E+02	2.25E+02	1.32E+02	1.38E+02
25	1.96E+02	1.85E+02	1.93E+02	2.09E+02	2.07E+02	2.26E+02	1.94E+02	1.97E+02
26	1.18E+02	1.19E+02	1.20E+02	1.31E+02	1.37E+02	1.97E+02	1.10E+02	1.20E+02
27	3.91E+02	3.91E+02	3.97E+02	4.08E+02	4.07E+02	4.02E+02	4.05E+02	4.02E+02
28	2.10E+02	1.80E+02	1.40E+02	1.50E+02	1.23E+02	6.23E+02	1.70E+02	2.00E+02

Table 3: The mean error values of the original FPA and the proposed FPAPA algorithms of the 28 test functions for d = 30 after 10000·d function evaluations

	FPA						FPAPA	
Function	p = 0	p = 0.2	p = 0.4	p = 0.6	p = 0.8	p = 1.0	p_1 = 0.2 p_2 = 0.6	p_1 = 0.6 p_2 = 0.2
1	7.53E+02	1.50E-07	1.00E-08	1.00E-08	4.96E-08	4.92E-05	1.00E-08	1.03E-08
2	1.32E+05	1.97E+01	2.31E+00	1.70E-01	6.19E+01	4.81E+06	2.76E+01	2.78E+00
3	1.22E+09	9.68E+05	1.50E+06	7.98E+06	4.25E+07	4.45E+08	8.07E+05	2.99E+06
4	2.38E+03	1.32E+02	1.38E+02	9.99E+01	7.05E+02	7.30E+04	1.11E+02	3.22E+01
5	5.85E+01	2.36E-04	9.40E-06	2.30E-05	5.18E-05	1.21E-02	3.79E-06	6.51E-05
6	8.91E+01	9.01E+00	9.18E+00	1.18E+01	1.35E+01	2.30E+01	1.07E+01	1.24E+01
7	6.35E+01	8.33E+01	9.82E+01	1.07E+02	1.11E+02	1.40E+02	9.74E+01	9.75E+01
8	2.09E+01	2.10E+01	2.10E+01	2.09E+01	2.09E+01	2.09E+01	2.10E+01	2.09E+01
9	2.43E+01	2.66E+01	2.78E+01	2.95E+01	3.04E+01	3.40E+01	2.63E+01	2.87E+01
10	7.64E+01	1.07E-02	3.19E-03	5.42E-04	1.10E-03	4.11E-01	6.12E-03	2.22E-03
11	1.05E+02	6.63E+01	6.85E+01	8.02E+01	8.36E+01	9.20E+01	6.47E+01	8.19E+01
12	1.38E+02	1.18E+02	1.38E+02	1.60E+02	2.10E+02	6.59E+02	1.32E+02	1.51E+02
13	1.78E+02	1.98E+02	2.00E+02	2.27E+02	2.41E+02	6.31E+02	2.08E+02	1.98E+02
14	4.78E+03	3.31E+03	3.07E+03	2.97E+03	2.85E+03	2.81E+03	2.87E+03	3.41E+03

15	4.75E+03	4.23E+03	4.10E+03	4.21E+03	4.30E+03	4.55E+03	3.98E+03	4.29E+03
16	1.86E+00	1.98E+00	1.77E+00	2.18E+00	2.13E+00	2.44E+00	2.00E+00	2.10E+00
17	2.05E+02	1.98E+02	1.83E+02	1.93E+02	1.96E+02	4.22E+02	1.71E+02	1.82E+02
18	2.08E+02	1.90E+02	2.08E+02	2.26E+02	2.39E+02	6.95E+02	1.89E+02	1.99E+02
19	2.95E+01	1.36E+01	1.16E+01	1.14E+01	1.18E+01	2.49E+01	9.96E+00	1.29E+01
20	1.21E+01	1.21E+01	1.24E+01	1.27E+01	1.31E+01	1.46E+01	1.24E+01	1.27E+01
21	6.28E+02	2.79E+02	2.67E+02	2.60E+02	2.18E+02	2.26E+02	2.72E+02	2.57E+02
22	5.39E+03	3.89E+03	3.81E+03	3.64E+03	3.29E+03	3.36E+03	3.30E+03	3.98E+03
23	5.31E+03	4.66E+03	4.75E+03	5.23E+03	5.20E+03	5.80E+03	4.63E+03	5.05E+03
24	2.73E+02	2.78E+02	2.79E+02	2.83E+02	2.91E+02	3.15E+02	2.78E+02	2.79E+02
25	2.91E+02	2.97E+02	3.00E+02	3.04E+02	3.12E+02	3.41E+02	2.99E+02	2.97E+02
26	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02
27	9.96E+02	1.07E+03	9.93E+02	9.91E+02	9.17E+02	1.03E+03	1.07E+03	1.01E+03
28	1.30E+03	3.83E+02	3.66E+02	3.66E+02	3.01E+02	4.12E+03	3.00E+02	3.62E+02

Table 4: Sign test pairwise comparisons of the proposed FPAPA with the original FPA options for the 28 test functions and d = 10 after 10000·d function evaluations

Proposed FPAPA	FPA						FPAPA opposing pollinator attraction
p_1 = 0.0, p_2 = 0.4	p = 0	p = 0.2	p = 0.4	p = 0.6	p = 0.8	p = 1.0	p_1 = 0.4, p_2 = 0.0
Wins	19.5	19.5	20.5	22.5	24	23	20.5
Losses	8.5	8.5	7.5	5.5	4	5	7.5

Table 4: Sign test pairwise comparisons of the proposed FPAPA with the original FPA options for the 28 test functions and d = 30 after 10000·d function evaluations

Proposed FPAPA	FPA						FPAPA opposing pollinator attraction
p_1 = 0.0, p_2 = 0.4	p = 0	p = 0.2	p = 0.4	p = 0.6	p = 0.8	p = 1.0	p_1 = 0.4, p_2 = 0.0
Wins	19	19	19.5	21.5	22	24	19
Losses	9	9	8.5	6.5	6	4	9

Table 6: Wilcoxon signed ranks test for pairwise comparisons of the proposed FPAPA with the original FPA options for the 28 test functions and d = 10 after 10000·d function evaluations

Comparison	R^+	R^-	T
Proposed FPAPA versus FPA with p = 0.0	293	104	104
Proposed FPAPA versus FPA with p = 0.2	308	89	89
Proposed FPAPA versus FPA with p = 0.4	315	82	82
Proposed FPAPA versus FPA with p = 0.6	337	70	70
Proposed FPAPA versus FPA with p = 0.8	352	54	54
Proposed FPAPA versus FPA with p = 1.0	371	35	35
Proposed FPAPA versus FPAPA opposing pollinator attraction	329	68	68

Table 7: Wilcoxon signed ranks test for pairwise comparisons of the proposed FPAPA with the original FPA options for the 28 test functions and d = 10 after 10000·d function evaluations

Comparison	R^+	R^-	T
Proposed FPAPA versus FPA with p = 0.0	339	67	67
Proposed FPAPA versus FPA with p = 0.2	291	115	115
Proposed FPAPA versus FPA with p = 0.4	305	102	102
Proposed FPAPA versus FPA with p = 0.6	326	81	81
Proposed FPAPA versus FPA with p = 0.8	323	83	83
Proposed FPAPA versus FPA with p = 1.0	358	48	48
Proposed FPAPA versus FPAPA opposing pollinator attraction	292	114	114

In addition to comparing with the original FPA algorithm, the proposed FPAPA is compared with three state-of-the art optimization algorithms including the Standard Particle Swarm Optimization Algorithm (SPSO-2011) [50], the Global and Local real-coded Genetic Algorithm (GL-25) [51], and the Covariance Matrix Adaptation Evolution Strategies (CMA-ES) [52]. SPSO-2011 represents a major improvement over previous PSO versions with an adaptive random topology and rotational invariance being the main advancements. GL-25 was developed by Garzia-Martinez *et al*. in 2008 and it is based on parent-centric real-parameter crossover operators to create off-springs. CMA-ES, developed by Hansen and Ostermeier in 2001, is one of the most successful and cited variants of Evolution Strategies that puts forward two useful methods for self-adaptation of the mutation distribution. Furthermore, FPAPA is compared with another variant of FPA, namely the Novel Modified FPA (NMFPA), by Fouad an Gao [28] that is described in the introduction section of this study and has been found to yield superior computational performance when compared with other algorithms and FPA variants [28].

Tables 8 and 9 show the mean solution errors of 20 independent runs of the afore-described optimization algorithms for the 28 CEC'13 test functions after 10000·d function evaluations for d =10 and 30 respectively. These results are taken from Fouad and Gao [28] where the algorithm settings used are the ones proposed in their original papers. The results of the proposed FPAPA are also included in these tables as well as the rankings of the five optimization algorithms based on their mean solution errors.

It can be seen in these tables that the proposed FPAPA is second best for both numbers of dimensions with average rankings 2.11 and 2.79 for d = 10 and 30 respectively following the NMFPA that exhibits average rankings of 1.43 and 1.61 for the same dimensions. The SPSO-2011 is the third best with 3.32 average ranking for d = 10 and CMA-ES is the third best for d = 30 with average ranking 2.96.

The previous results drive to the conclusion that the proposed FPAPA offers a significant improvement to the computational performance of the original FPA with only minor modifications to its formulation. Furthermore, FPAPA is able to perform better or equal to well-established metaheuristic optimization algorithms in literature. There exist in literature other FPA variants that seem to demonstrate higher computational performance than FPAPA. Therefore, it is worth investigating in the future the combination of FPAPA with other FPA variants to achieve even higher numerical efficiency.

Table 8: Mean solution errors and rankings of the proposed FPAPA and the SPSO-2011, GL-25, CMA-ES & NMFPA algorithms for the CEC'13 test functions with d = 10 after 10000·*d* function evaluations

	Mean errors					Rankings				
Function	**FPAPA**	**SPSO**	**GL-25**	**CMA-ES**	**NMFPA**	**FPAPA**	**SPSO**	**GL-25**	**CMA-ES**	**NMFPA**
1	1.00E-08	1.08E+03	1.00E-08	1.00E-08	1.00E-08	1	5	1	1	1
2	1.00E-08	3.90E+04	4.24E+06	1.00E-08	4.27E+02	1	4	5	1	3
3	2.11E+03	7.46E+04	3.28E+08	1.55E+01	2.35E+00	3	4	5	2	1
4	1.00E-08	2.12E+03	1.62E+04	1.00E-08	7.13E+00	1	4	5	1	3
5	1.00E-08	7.46E+02	1.00E-08	1.00E-08	1.00E-08	1	5	1	1	1
6	1.00E-08	5.82E+02	2.50E+01	7.35E+00	5.58E-01	1	5	4	3	2
7	7.81E+00	3.46E+02	2.64E+01	4.26E+02	7.13E-07	2	4	3	5	1
8	2.04E+01	2.08E+01	2.05E+01	2.03E+01	2.01E+01	3	5	4	2	1
9	4.71E+00	1.24E+01	5.78E+00	1.24E+01	2.14E+00	2	4	3	4	1
10	2.72E-02	4.19E+02	8.83E+00	1.74E+02	7.57E-02	1	5	3	4	2
11	7.46E+00	1.77E+02	9.52E+00	1.33E+02	1.44E+00	2	5	3	4	1
12	8.21E+00	1.86E+02	2.38E+01	4.00E+02	7.06E+00	2	4	3	5	1
13	1.16E+01	1.74E+02	2.68E+01	3.43E+02	7.17E+00	2	4	3	5	1
14	4.84E+02	6.52E+02	6.32E+02	1.61E+03	2.55E+01	2	4	3	5	1
15	7.33E+02	5.94E+02	1.45E+03	1.74E+03	3.49E+02	3	2	4	5	1
16	8.88E-01	6.19E-01	1.79E+00	2.37E+01	5.28E-01	3	2	4	5	1
17	2.12E+01	1.72E+01	2.78E+01	1.07E+03	1.03E+01	3	2	4	5	1
18	2.44E+01	1.90E+01	4.39E+01	1.01E+03	1.65E+01	3	2	4	5	1
19	7.37E-01	8.11E-01	1.40E+00	1.12E+00	5.13E-01	2	3	5	4	1
20	3.11E+00	2.37E+00	3.34E+00	4.05E+00	1.44E+00	3	2	4	5	1
21	1.35E+02	4.00E+02	4.00E+02	4.00E+02	2.85E+02	1	3	3	3	2
22	6.69E+02	6.48E+02	8.07E+02	2.29E+03	1.69E+02	3	2	4	5	1
23	9.20E+02	4.05E+02	1.47E+03	2.17E+03	5.55E+02	3	1	4	5	2
24	1.32E+02	2.01E+02	2.12E+02	4.63E+02	1.81E+02	1	3	4	5	2
25	1.94E+02	2.00E+02	2.08E+02	2.53E+02	1.91E+02	2	3	4	5	1
26	1.10E+02	1.34E+02	1.49E+02	2.73E+02	1.48E+02	1	2	4	5	3
27	4.05E+02	3.12E+02	3.88E+02	3.62E+02	3.58E+02	5	1	4	3	2
28	1.70E+02	2.80E+02	3.63E+02	9.03E+02	1.60E+02	2	3	4	5	1
				Average Rankings:		**2.11**	**3.32**	**3.64**	**3.86**	**1.43**

Table 9: Mean solution errors and rankings of the proposed FPAPA and the SPSO-2011, GL-25, CMA-ES & NMFPA algorithms for the CEC'13 test functions with d = 30 after $10000 \cdot d$ function evaluations

	Mean errors					Rankings				
Function	**FPAPA**	**SPSO**	**GL-25**	**CMA-ES**	**NMFPA**	**FPAPA**	**SPSO**	**GL-25**	**CMA-ES**	**NMFPA**
1	1.00E-08	1.28E+03	1.96E-04	1.00E-08	1.00E-08	1	5	4	1	1
2	2.76E+01	2.45E+05	3.18E+07	1.00E-08	1.26E+04	2	4	5	1	3
3	8.07E+05	4.46E+07	5.96E+09	1.94E+03	1.70E+05	3	4	5	1	2
4	1.11E+02	6.23E+03	4.25E+04	1.00E-08	7.88E-01	3	4	5	1	2
5	3.79E-06	9.62E+02	1.17E-04	1.00E-08	1.00E-08	3	5	4	1	1
6	1.07E+01	7.76E+02	1.02E+02	1.32E+00	6.59E+00	3	5	4	1	2
7	9.74E+01	5.03E+02	8.51E+01	1.60E+01	8.04E-01	4	5	3	2	1
8	2.10E+01	2.12E+01	2.10E+01	2.09E+01	2.08E+01	3	5	4	2	1
9	2.63E+01	4.77E+01	3.11E+01	4.43E+01	1.55E+01	2	5	3	4	1
10	6.12E-03	4.76E+02	6.35E+01	1.78E-02	2.90E-01	1	5	4	2	3
11	6.47E+01	3.80E+02	7.35E+01	1.27E+02	1.55E+01	2	5	3	4	1
12	1.32E+02	2.87E+02	1.74E+02	6.66E+02	3.51E+01	2	4	3	5	1
13	2.08E+02	1.94E+02	1.97E+02	2.13E+03	7.21E+01	4	2	3	5	1
14	2.87E+03	4.80E+03	4.84E+03	5.11E+03	1.35E+03	2	3	4	5	1
15	3.98E+03	4.31E+03	7.63E+03	5.18E+03	3.27E+03	2	3	5	4	1
16	2.00E+00	1.41E+00	3.09E+00	1.01E-01	1.78E+00	4	2	5	1	3
17	1.71E+02	1.26E+02	1.67E+02	3.77E+03	5.15E+01	4	2	3	5	1
18	1.89E+02	1.07E+02	2.43E+02	4.19E+03	7.72E+01	3	2	4	5	1
19	9.96E+00	5.77E+00	5.60E+01	3.51E+00	1.86E+00	4	3	5	2	1
20	1.24E+01	1.07E+01	1.35E+01	1.26E+01	9.70E+00	3	2	5	4	1
21	2.72E+02	3.18E+02	3.71E+02	2.84E+02	3.28E+02	1	3	5	2	4
22	3.30E+03	3.85E+03	3.89E+03	7.04E+03	1.07E+03	2	3	4	5	1
23	4.63E+03	4.19E+03	7.43E+03	6.73E+03	3.39E+03	3	2	5	4	1
24	2.78E+02	2.28E+02	2.50E+02	9.35E+02	2.31E+02	4	1	3	5	2
25	2.99E+02	2.62E+02	2.96E+02	2.59E+02	2.63E+02	5	2	4	1	3
26	2.00E+02	2.31E+02	2.17E+02	4.53E+02	2.00E+02	2	4	3	5	1
27	1.07E+03	5.78E+02	9.64E+02	5.75E+02	6.32E+02	5	2	4	1	3
28	3.00E+02	3.00E+02	1.02E+03	9.82E+02	3.00E+02	1	1	5	4	1
				Average Rankings:		**2.79**	**3.32**	**4.07**	**2.96**	**1.61**

4.2 Structural optimization problems

In this section, the performance of FPAPA is tested against real-world structural optimization problems. Two separate structural optimization problems are examined in the following.

4.2.1 Volume optimization of a cantilever beam

A stepped cantilever beam is considered herein carrying a concentrated load P = 50000 N at its free end as shown in Fig. 9 [53]. The objective is to minimize the volume of the beam while keeping bending stresses below the permissible limit of 14000 N/cm^2 and the displacement at the free end below 2.7 cm. The ten design variables (d = 10) of the problem are the widths b_i and heights h_i (i = 1 to 5) of the rectangular cross sections of the five beam segments with length l = 500 cm. Furthermore, the aspect ratio between the cross sections heights and widths cannot exceed the value of 20. It is also considered that sections widths range between 1 cm ≤ b_i ≤ 5 cm and heights 30 cm ≤ h_i ≤ 65 cm (i = 1 to 5). The elastic modulus of the beam is E = 200 GPa.

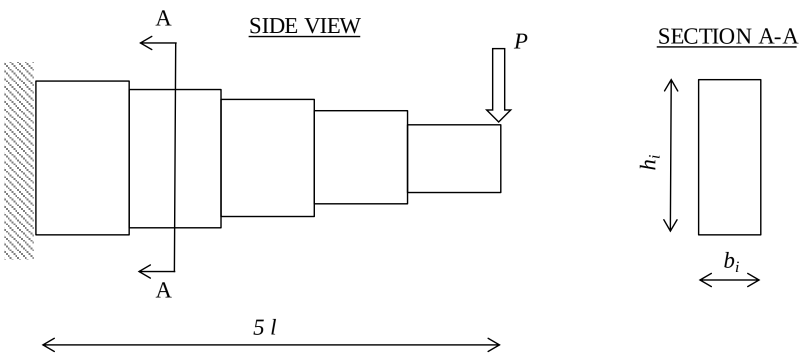

Fig. 9: Stepped cantilever beam.

For this optimization problem, the PSO and GA algorithms are applied in addition to the original FPA algorithm with switch probability values of p = 0 and 0.4 in accordance with the findings of §4.1. Furthermore, the proposed FPAPA is applied with one arrangement for the switch probabilities supporting the pollinator attraction rule and another one opposing it.
For each algorithm, 50 independent runs are conducted independently for three different values of function evaluations that are 100·d, 300·d and 500·d, respectively. Table 10 presents the mean values and standard deviations of the minimum beam volume values obtained by the different solutions. It can be seen that for all function evaluation values the proposed algorithm

solution provides the best mean beam volumes and standard deviations. In the last row of the table, the minimum volumes out of all 50 runs after 500·*d* evaluations are also presented. It is observed that the proposed solution yields again the best result with a value of 63111 cm^3, which approaches very closely to the optimum value of 63110 cm^3 reported in [53]. The PSO algorithm is characterised by significant variability and in most of the cases returns the worst mean beam volumes. Among the original FPA solutions, the $p = 0$ option yields the best results outperforming also the FPAPA algorithm opposing the pollinator attraction rule.

Table 10: Minimum beam volumes (cm^3)

	PSO	**GA**	**Original FPA** $p = 0$	**Original FPA** $p = 0.4$	**Proposed FPAPA** $p_1 = 0$ & $p_2 = 0.4$	**FPAPA opposing pollinator attraction** $p_1 = 0.4$ & $p_2 = 0$
	$MaxFES = 100 \cdot d$					
Mean	6.4336E+04	6.4733E+04	6.3801E+04	6.4409E+04	**6.3773E+04**	6.4346E+04
Standard Deviation	2.0302E+03	6.2398E+02	3.0836E+02	8.3313E+02	**2.6377E+02**	5.3064E+02
	$MaxFES = 300 \cdot d$					
Mean	6.3477E+04	6.3166E+04	6.3139E+04	6.3166E+04	**6.3136E+04**	6.3155E+04
Standard Deviation	1.0456E+03	2.0389E+01	2.3375E+01	2.5401E+01	**1.2989E+01**	1.5431E+01
	$MaxFES = 500 \cdot d$					
Mean	6.3403E+04	6.3125E+04	6.3120E+04	6.3132E+04	**6.3120E+04**	6.3125E+04
Standard Deviation	1.0130E+03	7.7142E+00	5.2836E+00	1.5689E+01	**4.9821E+00**	8.3286E+00
Minimum	6.3115E+04	6.3114E+04	6.3113E+04	6.3114E+04	**6.3111E+04**	6.3114E+04

4.2.2 Cost optimization of an earthquake-resistant reinforced concrete frame

In this section, the proposed FPAPA variant is applied to the seismic design of a three-storey two-bay (Fig. 10) reinforced concrete frame. The design constraints are set according to Eurocode 2 (EC2) [54] and Eurocode 8 (EC8) [55] structural design codes for Ductility Class Low (DCL). The frame is part of a building of ordinary importance that rests on soil class B. Uniform distributed loads of 22.5 kN/m act along beam members of all storeys and point loads of 67.5 kN and 135 kN are applied at the exterior and interior joints, respectively, for the quasi-permanent load combination. A design peak ground acceleration value of 0.40 g is assumed. Concrete class C25/30 and reinforcing steel grade B500C are used.

The objective of the design is to minimize the material cost which is the sum of the costs of concrete, reinforcing steel and formwork. The following unit prices are assumed for these materials: concrete 101 (€ / m^3), reinforcing steel 1.07 (€ / kg) and formwork 15.7 (€ / m^2). Eight (d = 8) independent design variables are used in this problem, assuming symmetric concrete frame configuration. These are the section heights of the central and exterior square columns, respectively, as well as the section heights and widths of the rectangular beams of the three storeys. All cross-sectional dimensions range between 0.3 m and 1 m. For construction simplicity, steel reinforcement is assumed to be uniform along column and beam members. More information on the assumptions and design methodology of this frame can be found in Mergos [56].

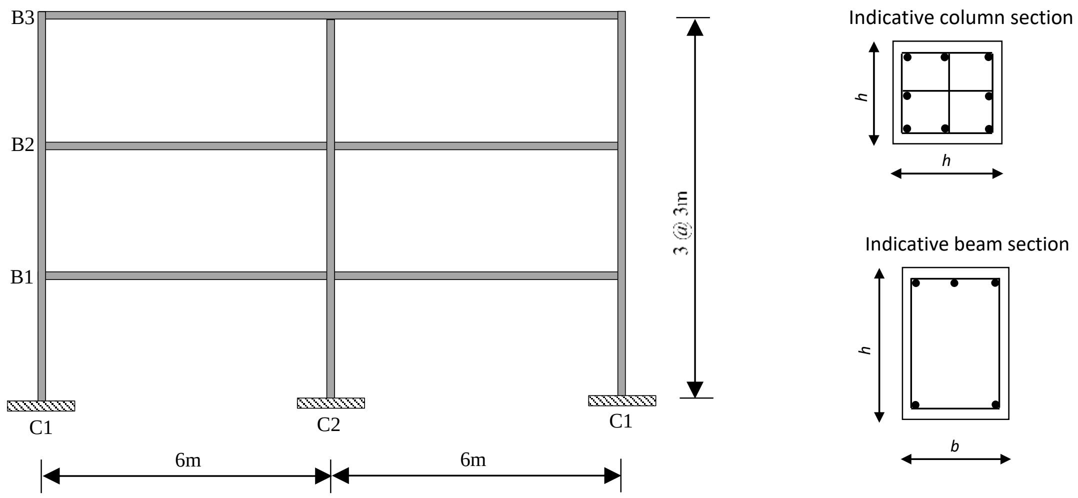

Fig. 10: Three-storey two-bay reinforced concrete frame [56]

Table 11: Material costs of the reinforced concrete frame.

Algorithm	**Minimum Costs (€)**
PSO	6792.91 ± 46.28 (60%)
GA	6765.45 ± 22.07 (100%)
Original FPA (*p* = 0)	6760.88 ± 26.21 (80%)
Original FPA (*p* = 0.4)	6805.49 ± 32.89 (80%)
Proposed FPAPA p_1 **= 0 &** p_2 **= 0.4**	6752.32 ± 16.29 (100%)
FPAPA opposing pollinator attraction p_1 **= 0.4 &** p_2 **= 0**	6785.91 ± 29.65 (90%)

Table 11 presents the statistical results of the minimum material costs obtained by 10 independent runs for each algorithm solution and 800 (= 100·d) function evaluations. The costs are provided in the form: mean ± standard deviation (success rate). Successful are the designs satisfying all design constraints of EC2 and EC8. It is evident that the proposed FPAPA outperforms the other solutions as it exhibits the minimum mean cost (€6752.32) and standard deviation (€16.29). Furthermore, it always drives to successful design solutions. The least satisfying performance is obtained by the PSO algorithm and the FPA algorithm for $p = 0.4$ considering both costs and success rates. The relatively poor performance of FPAPA opposing the pollinator attraction rule is also observed.

5 Conclusions

In this paper, a modified version of the Flower Pollination Algorithm is presented, namely Flower Pollination Algorithm with Pollinator Attraction (FPAPA), that accounts for the pollinator attraction evolution mechanism. The pollinator attraction mechanism reflects the observed natural tendency of flower species to evolve in order to attract pollinators by nutritious rewards and attractive shapes, colours and scents. Thereby, it is anticipated that the fitter flowers, that develop the most efficient mechanisms to entice pollinators, will be more likely to achieve pollen transfer by biotic pollination.

To model this expectation in FPAPA, the switch probability p that controls the pollination mechanism (biotic or abiotic) in FPA is not taken as constant for all flowers of the population, but it is varied appropriately in the population so that fitter flowers are provided with higher probabilities of conducting biotic pollination.

The proposed FPAPA has been validated against the set of 28 benchmark functions defined in IEEE-CEC'13 for real-parameter single-objective optimization problems as well as structural optimization problems. It is found that the proposed FPAPA, whilst maintaining almost the same level of simplicity as the original FPA code, outperforms significantly the original FPA. Furthermore, it offers superior performance when compared with other state-of-the-art metaheuristic algorithms.

Further research will focus on parameter tuning of the scheme of the variations of the flower probabilities to conduct biotic pollination based on their objective function as well as exploring additional schemes that may offer higher computational performance. Furthermore, the combination of FPAPA with other modified and hybridized versions of FPA will be examined to further improve its performance for different optimization problems in engineering and industries.

6 Declarations

Funding: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of Interest: The authors declare that they have no conflict of interest.

Availability of data and material: The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Code availability: The codes developed during the current study are available from the corresponding author on reasonable request.

References

[1] Yang XS (2008) Nature-inspired metaheuristic algorithms. Luniver Press, UK

[2] Holland JH (1975) Adaptation in natural and artificial systems. An introductory analysis with application to biology, control and artificial intelligence. University of Michigan Press, Ann Arbor, MI

[3] Yang XS (2010) Firefly algorithm, stochastic test functions and design optimization. Int J Bio-inspir Com 2:78-84

[4] Kennedy J (2011) Particle swarm optimization. Encyclopedia of Machine Learning, Springer, 760-766

[5] Gandomi AH, Yang XS, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Computation 29:17-35

[6] Yang XS (2012) Flower pollination algorithm for global optimization. Unconven Comput Nat Comput 7445:240-249

[7] Alyasseri ZAA, Khader AT, Al-Betar MA, Awadallah MA, Yang XS (2018) Variants of the flower pollination algorithm: A review. Studies in Computational Intelligence, Springer, 91-11

[8] Bekdas G, Nigdeli SM, Yang XS (2015) Sizing optimization of truss structures using flower pollination algorithm. Appl Soft Comput 37:322-331

[9] Mergos PE, Mantoglou F (2020) Optimum design of reinforced concrete retaining walls with the flower pollination algorithm. Struct Multidiscipl Optim 61:575-585

[10] Mergos PE (2021) Optimum design of 3D reinforced concrete building frames with the flower pollination algorithm. J Build Eng 44:102935

[11] Abdelaziz A, Ali E, Elazim SA (2016) Combined economic and emission dispatch solution using flower pollination algorithm. Int J Electr Power Energy Syst 80:264–274

[12] Abdelaziz A, Ali E, Elazim SA (2016) Implementation of flower pollination algorithm for solving economic load dispatch and combined economic emission dispatch problems in power systems, Energy 101:506–518

[13] Singh U, Salgotra R (2016) Synthesis of linear antenna array using flower pollination algorithm, Neural Comput Appl, 1–11

[14] Singh U, Salgotra R (2016) Synthesis of linear antenna array using flower pollination algorithm, Neural Comput Appl, 1–11.

[15] Nigdeli SM, Bekdaş G, Yang XS (2016) Application of the flower pollination algorithm in structural engineering, Metaheuristics and Optimization in Civil Engineering, Springer, 25–42

[16] Abdel-Raouf O, El-Henawy I, Abdel-Basset M (2014) A novel hybrid flower pollination algorithm with chaotic harmony search for solving sudoku puzzles. Int J Mod Educ Comput Sci 6:38

[17] Heng J, Wang C, Zhao X, Xiao L (2016) Research and application based on adaptive boosting strategy and modified CGFPA algorithm: a case study for wind speed forecasting. Sustainability 8:235

[18] Zhou Y, Zhang S, Luo Q, Wen C (2016) Using flower pollination algorithm and atomic potential function for shape matching. Neural Comput Appl 29:21–40

[19] Abdel-Basset M, Shawky LA (2018) Flower pollination algorithm: a comprehensive review. Artif Intell Rev 52:2533-2557

[20] Abdel-Raouf O, Abdel-Basset M, El-Henawy I (2014) An improved flower pollination algorithm with chaos. Int J Educ Managt Eng 4:1–8

[21] Zhou Y, Wang R, Luo Q (2016) Elite opposition-based flower pollination algorithm. Neurocomputing 188:294–310

[22] Putra PH, Saputra TA *et al* (2016) Modified flower pollination algorithm for non-smooth and multiple fuel options economic dispatch. 8th International Conference on Information Technology and Electrical Engineering (ICITEE), IEEE, 1-5

[23] Draa A (2016) On the performances of the flower pollination algorithm: qualitative and quantitative analyses. Appl Soft Comput 34:349-371

[24] Wang R, Zhou Y, Qiao S, Huang K (2016) Flower pollination algorithm with bee pollinator for cluster analysis. Inf Proc Lett 116:1–14

[25] Al-Betar MA, Awadallah MA, Doush IA, Hammouri AI, Mafarja M, Alyasseri ZAA (2019) Island flower pollination algorithm for global optimization. The Journal of Supercomputing 75:5280-532

[26] Abdel-Basset M, El-Shahat D, El-Henawy I, Sangaiah AK (2018a) A modified flower pollination algorithm for the multidimensional knapsack problem: human-centric decision making. Soft Comput 22:4221-4239

[27] Zhou Y, Wang R, Zhao C, Luo Q, Metwally MA (2017) Discrete greedy flower pollination algorithm for spherical traveling salesman problem. Neural Comput Appl 31:2155-2170

[28] Fouad A, Gao X-Z (2019) A novel modified flower pollination algorithm for global optimization. Neural Comput Appl 31:3875–3908

[29] Khursheed M, Alghamdi M, Khan M, Khan A, Khan I, Ahmed A, Kiani A (2021) PV Model parameter estimation using modified FPA with dynamic switch probability and step size function. IEEE Access 9: 42027:42044

[30] Xiao Y, Wu Y, Yang F (2021) Robust visual tracking based on modified flower pollination algorithm. IEEE Access 9:157458:157467

[31] Ozsoydan FB, Baykasoglu A (2021) Chaos and intensification enhanced flower pollination algorithm to solve mechanical design and unconstrained function optimization problems. Expert systems with applications 225:107125

[32] Ozsoydan FB, Baykasoglu A (2021) A species-based flower pollination algorithm with increased selection pressure in abiotic local pollination and enhance intensification. Knowl Based Syst 225:107125

[33] Rodrigues D, Yang XS, De Souza AN, Papa JP (2015) Binary flower pollination algorithm and its application to feature selection. Recent Advances in Swarm Intelligence and Evolutionary Computation, Springer, 85–100

[34] Yang XS, Karamanoglu M, He X (2014) Flower pollination algorithm: a novel approach for multi-objective optimization. Eng Optim 46:1222-1237

[35] Tamilselvan V, Jayabarathi T (2016) Multi-objective flower pollination algorithm for solving capacitor placement in radial distribution system using data structure load flow analysis. Arch Electr Eng 65:203–220

[36] Gonidakis D (2016) Application of flower pollination algorithm to multi-objective environmental/economic dispatch. Int J Manag Sci Eng Manag 11:213–221

[37] Jensi R, Jiji GW (2015) Hybrid data clustering approach using k-means and flower pollination algorithm. arXiv:1505.03236.

[38] Abdel-Basset M, Hezam I (2016) A hybrid flower pollination algorithm for engineering optimization problems. Int J Comput Appl 140:10-23

[39] Abdel-Raouf O, Abdel-Baset M *et al*. (2014) A new hybrid flower pollination algorithm for solving constrained global optimization problems. Int J Appl Oper Res-An Open Access J 4:1–13
[40] Dubey HM, Pandit M, Panigrahi B (2015) Hybrid flower pollination algorithm with time-varying fuzzy selection mechanism for wind integrated multi-objective dynamic economic dispatch. Renew Energy 83:188–202
[41] Hezam IM, Abdel-Baset M, Hassan B (2016) A hybrid flower pollination algorithm with tabu search for unconstrained optimization problems. Inf Sci Lett 5:29-34
[42] Nigdeli SM, Bekdaş G, Yang XS (2017) Optimum tuning of mass dampers by using a hybrid method using harmony search and flower pollination algorithm. International Conference on Harmony Search Algorithm, Springer, 222–231
[43] Walker M (2009) How flowers conquered the world BBC Earth News. http://news.bbc.co.uk/earth/hi/earth_news/newsid_8143000/8143095.stm
[44] Glover BJ (2007) Understanding flowers and flowering: An integrated approach. Oxford University Press, UK
[45] Pavlyukevich I (2007) Lévy flights, non-local search and simulated annealing. J Comput Phys 226:1830-1844
[46] Glover BJ (2011) Pollinator attraction: The importance of looking good and smelling nice. Curr Biol 21:R307-R309
[47] Wikipedia contributors, Flower, Wikipedia (2018) The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Flower&oldid=861525932
[48] Liang J, Qu B, Suganthan P, Hernandez-Daz AG (2013) Problem definitions and evaluation criteria for the CEC2013 special session on real-parameter optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Nanyang Technological University, Singapore, Technical Report 20121
[49] Derrac J, Garcia S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms.
[50] Clerc M (2012) Standard particle swarm optimisation. Technical Report HAL Id: hal-00764996, https://hal.archives-ouvertes.fr/hal-00764996
[51] Garcia-Martinez C, Lozano M, Herrera F, Molina D, Sanchez A (2008) Global and local real-coded genetic algorithms based on parent-centric crossover operators. Eur J Oper Res 185:1088–1113
[52] Hansen N, Ostermeier A (2001) Completely derandomized self-adaptation in evolution strategies. Evol Comput 9:159–195
[53] Thanedar PB, Vanderplaats GN (1995) Survey of discrete variable optimization for structural design, J. Struct. Eng. 12:301-306
[54] CEN (2000) Eurocode 2: Design of concrete structures. Part 1-1: General rules and rules for buildings. Brussels: European Standard EN 1992-1-1
[55] CEN (2004) Eurocode 8: Design of structures for earthquake resistance. Part 1: General rules, seismic actions and rules for buildings. Brussels: European Standard EN 1998-1
[56] Mergos PE (2018) Seismic design of reinforced concrete frames for minimum embodied CO_2 emissions. Energ. Buildings 162:177-186