City Research Online

City, University of London Institutional Repository

This is the accepted version of the paper.

This version of the publication may differ from the final published version.

Flower Pollination Algorithm Parameters Tuning

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 metaheuristic optimization algorithm that is inspired by the pollination process of flowering species. FPA is characterised by simplicity in its formulation and high computational performance. Previous studies on FPA assume fixed parameter values based on empirical observations or experimental comparisons of limited scale and scope. In this study, a comprehensive effort is made to identify appropriate values of the FPA parameters that maximize its computational performance. To serve this goal, a simple non-iterative, single-stage sampling tuning method is employed, oriented towards practical applications of FPA. The tuning method is applied to the set of 28 functions specified in IEEE-CEC'13 for real-parameter single-objective optimization problems. It is found that the optimal FPA parameters depend significantly on the objective functions, the problem dimensions and affordable computational cost. Furthermore, it is found that the FPA parameters that minimize mean prediction errors do not always offer the most robust predictions. At the end of this study, recommendations are made for setting the optimal FPA parameters as a function of problem dimensions and affordable computational cost.

Keywords: Optimization; Metaheuristics; Evolutionary; Flower pollination algorithm; Parameters tuning

1 Introduction

In many complex multi-modal design problems in industry and engineering, tracking of global optimum solutions remains a highly challenging task. Often, conventional optimization methodologies do not perform adequately in this class of problems as they may be trapped in local optima. Then, the use of nature-inspired metaheuristic algorithms is recommended (Yang 2008). There is a high number of well-established metaheuristic optimisation algorithms in literature, including the Genetic Algorithm (GA) (Holland 1975), Firefly Algorithm (Yang

---

* 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, Tel. 0044 (0) 207040 8417

2010), Particle Swarm Optimization (PSO) (Kennedy 2011), Cuckoo Search (CS) (Gandomi et al. 2013) as well as very recent ones such as the Coronavirus Herd Immunity Optimizer (CHIO) (Al-Betar *et al*. 2021) and the Aquila Optimizer (Abualigah *et al*. 2021).

The Flower Pollination Algorithm (FPA) was developed by Xin-She Yang in 2012 and it is a population-based metaheuristic optimization algorithm inspired by the evolution process of flowering plants. FPA is characterised by simplicity and flexibility in its formulation as well as high efficiency in its computational performance (Alyasseri *et al*. 2018). Furthermore, many studies show that it can outperform other well-established metaheuristic optimization algorithms (e.g. Yang 2012; Bekdas et al. 2015; Mergos and Mantoglou 2020). A simple explanation of the efficiency of FPA is based on the fact that it mimics the reproduction process of flowering plants. The latter has been so successful that flower species dominate the landscape of earth (Walker 2009).

As a result, FPA has been adopted by many optimization studies and it has been applied successfully to numerous optimization problems in diverse scientific fields, including electrical and power systems (e.g. Abdelaziz et al. 2016a; Abdelaziz et al. 2016b; Singh and Salgotra 2016), structural design (e.g. Bekdas et al. 2015; Mergos and Mantoglou 2020, Nigdeli et al. 2016), computer gaming (e.g. Abdel-Raouf et al. 2014a), meteorology (e.g. Heng et al. 2016), image science (e.g. Zhou et al. 2016a) and many others (Alyasseri et al. 2018; Abdel-Basset et al. 2018b).

Following its original development, several studies proposed modified and hybridized versions of FPA to improve its performance for different optimization problems (Alyasseri et al. 2018). For example, Abdel-Raouf *et al*. (2014b) 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*. (2016b) developed an elite opposition-based FPA version that was tested with 18 benchmark functions yielding excellent results. Putra *et al*. (2016), 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 (2016) developed a new FPA variant based on the so-called generalized opposition-based learning (GOBL). Wang *et al*. (2016) merged the standard FPA with the concept of the bee-pollinator to solve the data clustering problem. Al-Betar *et al*. (2019) used the island model population technique to restrain premature convergence of FPA. Abdel-Basset *et al*. (2018a) developed a modified FPA version based on the crossover for solving the multidimensional knapsack problems. Zhou *et al*. (2017) 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 (2019) 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. Rodrigues *et al.* (2015) developed a binary version of FPA to address combinatorial and discrete optimization problems. Multi-objective versions of FPA have also been developed (e.g. Yang *et al.* 2014; Tamilselvan and Jayabarathi 2016; Gonidakis 2016) 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. Jensi and Jiji 2015; Abdel-Basset and Hezam 2016), population-based algorithms (e.g. Abdel-Raouf et al. 2014c; Dubey *et al.* 2015; Hezam et al. 2016; Nigdeli *et al.* 2017) or other components.

In the previous studies, researchers apply specific sets of parameters of the original FPA or their invented FPA variants that outperform some benchmarks optimization algorithms on benchmark problems. Nevertheless, the selection of these parameters sets is typically based on conventions, empirical choices and/or experimental comparisons of limited scale. The latter typically take place for very specific and limited in scope and number optimization problems that cannot drive to more general conclusions.

In this paper, a comprehensive and systematic experimental study is conducted to identify appropriate FPA parameters values that maximize its computational performance on a wide range of optimization problems. To serve this goal, FPA parameters tuning is conducted herein, which represents a separate optimization problem with FPA parameters set as the design variables and computational performance set as the design objective. It is clarified that the tuning approach followed herein aims at identifying fixed optimal parameter values. The adaptation and control of these parameters during the run of the algorithm to further maximise its performance is not examined in the present study. In the following, section 2 discusses the inspiration and formulation of the FPA algorithm, section 3 sets FPA parameters tuning as an appropriate optimization problem, section 4 describes the functions used to tune FPA parameters and section 5 presents the main findings of this study.

2 Flower Pollination Algorithm

FPA imitates the reproduction process of flowering plants. Similar to other biological systems, the ultimate goal of flower species is reproduction, which is achieved by pollination. Flower

pollination, which is typically related to the transfer of pollen, is either biotic or abiotic (Yang 2012, Glover 2007). In the former pollination type, pollen transfer takes place by animals and insects (e.g. bees, bats, butterflies, birds) which are called pollinators. Pollinators are capable of flying long distances. Therefore, biotic pollination could be considered as a global optimization mechanism (Yang 2012). Furthermore, the flight behaviour of pollinators has similar characteristics to Lévy flights (Yang 2008; Pavlyukevich 2007). In the abiotic type of pollination, pollen is transferred by the wind and/or water diffusion. A well-known example of abiotic pollination is the grass (Yang 2012; Glover 2007). Typically, abiotic pollination occurs at short distances. Hence, it could be considered as a local optimization mechanism (Yang 2012). An additional important feature of flower pollination is the so-called flower constancy. More particularly, some pollinators tend to select specific flower species and bypass others (Yang 2012). In this way, flowers manage to transfer more pollen to the same species. Furthermore, pollinators avoid the risks related to exploring other flower species and ensure guaranteed nectar intake.

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

1. Biotic pollination is treated as a global pollination process with pollinators performing Lévy flights.
2. Abiotic pollination is treated as a local pollination mechanism.
3. Flower constancy is considered by the assumption that the reproduction probability is proportional to the similarity of flowers involved.
4. The type of pollination mechanism (local or global) is governed by a switching probability p in [0, 1].

In the following, for simplicity, it is assumed that each plant develops one flower, which produces only one pollen gamete (Yang 2012). Following this assumption, there exists no need to differentiate between pollen gametes, flowers and plants.

In FPA, a solution vector x_i is represented by a flower i. The algorithm employs two different search procedures: the global and local pollination. Following the first and third rules of FPA, the global pollination procedure could be represented mathematically by the following equation:

$$x_i^{t+1} = x_i^t + \gamma \cdot L(\lambda) \cdot (g^* - x_i^t), \qquad (1)$$

4

where $\boldsymbol{x}_i^t$ represents flower i at iteration t, $\boldsymbol{g}^*$ is the best flower of all the population of flowers at iteration t, λ is a constant, γ is a constant scaling factor to control the step size, and $L(\lambda) > 0$ is the Lévy flight step size that represents the strength of the pollination and is drawn from a Lévy distribution, where $\Gamma(\lambda)$ is the standard gamma function and $s > 0$.

$$L \sim \frac{\lambda \Gamma(\lambda) \sin\left(\frac{\pi \lambda}{2}\right)}{\pi} \cdot \frac{1}{s^{1+\lambda}}, \quad (s > 0), \tag{2}$$
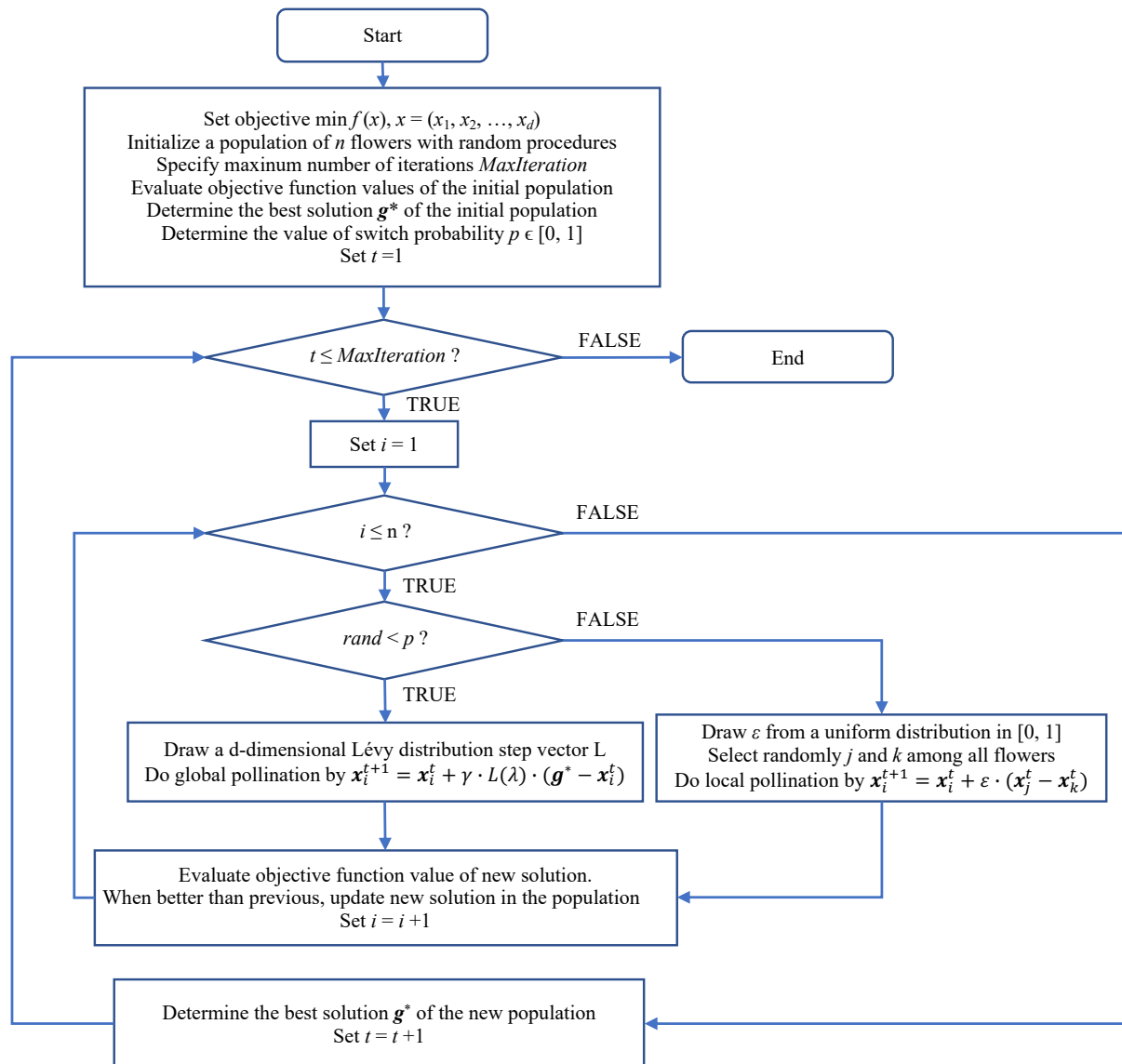
Fig. 1: Flowchart of the original FPA.

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].

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

Following the fourth rule, the type of flower pollination (local or global) is controlled by a switch probability p in [0, 1]. Summarizing the previous information, the flowchart of FPA is shown in Fig. 1, where d represents the number of problem dimensions and n is the size of flowers population.

3 Parameter tuning

Parameter tuning of an optimization algorithm aims at finding appropriate values of its parameters that maximize its computational performance. Therefore, parameter tuning can be considered as an optimization problem in the search space of the algorithm's parameters (Yang 2008, Eiben and Smit 2011). In the case of stochastic evolutionary algorithms, parameter tuning can represent a highly complex optimization problem of stochastic nature, interacting design variables, multiple local optima and very expensive objective function evaluations (Yang 2008, Eiben and Smit 2011).

If A represents an evolutionary optimization algorithm with a vector of parameters $\boldsymbol{p}$, Φ stands for a specified optimization problem and ξ represents of a measure of computational performance, then parameter tuning can be stated as:

$$maximize\ \xi = A(\Phi, \boldsymbol{p}). \tag{4}$$

In order to maximize the performance of an evolutionary algorithm, appropriate performance metrics are necessary. These metrics should refer to either the solution speed or the solution quality achieved by the algorithm. Solution speed may be quantified in terms of the computational effort (e.g. number of function evaluations N_r) required to achieve a minimum value δ (target value) of the objective function $f(\boldsymbol{x})$. In this case, the parameter tuning problem of Eq. (4) can be stated as:

$$minimize\ N_r =\ A(\Phi(f(\boldsymbol{x}),\delta),\boldsymbol{p}), where:\qquad\qquad(5)$$
$$\Phi(f(\boldsymbol{x}),\delta) = minimize\ f(\boldsymbol{x})\ to\ reach\ target\ value\ \delta$$

Alternatively, computational performance can be expressed in terms of solution quality which may be quantified as the best fitness value f_b achieved after terminating a pre-specified computational effort in terms, for example, of function evaluations N. In this case, parameter tuning may be written in the following form:

$$minimize\ f_b =\ A(\Phi(f(\boldsymbol{x}),N),\boldsymbol{p}), where:\qquad\qquad(6)$$
$$\Phi(f(\boldsymbol{x}),N) = minimize\ f(\boldsymbol{x})\ after\ N\ function\ evaluations$$

Due to the stochastic nature of evolutionary algorithms, multiple runs on the same optimization problem with the same set of parameters may yield different results. Therefore, statistical measures are required to quantify the computational performance such as the mean f_b and N_r values as well as their standard deviations after a number of independent solution runs.

A wide variety of tuning methods exist in literature (Eiben and Smit 2011). They can be classified as non-iterative or iterative depending on if they generate a fixed set of parameter vectors or if they start with a set of parameter vectors that modify iteratively during execution. Furthermore, they can be categorized as single-stage or multi-stage procedures depending on whether they conduct the same number of tests for all parameter vectors or they use a more sophisticated approach. In addition, they can be incorporated in four main categories: sampling methods, model-based methods, screening methods and meta-heuristic methods (Eiben and Smit 2011). Sampling methods reduce the number of parameter vectors with respect to a full-factorial design. They are characterised by simplicity but they can lead to high computational costs. Model-based methods generate metamodels of the objective function landscapes. They can reduce greatly the required computational effort, but their accuracy depends on the quality of the approximation of the real functions. Screening methods focus only on those parameter vectors that show more promising results. In this manner, they can reduce the computational cost, but they are also prone to get trapped in local optima. Metaheuristic methods use evolutionary algorithms to search for the parameters vector that is maximizing computational performance. These methods can be highly efficient in terms of both computational cost and quality of obtained solutions. Nevertheless, their computational performance depends on the values of their parameters, which could drive to an iterative procedure when the same

metaheuristic algorithm is used to tune its own parameters in a so-called self-tuning framework (Yang *et al* 2013).

Due to its simple formulation, FPA has only three parameters variables. These are the size of the population n, the switch probability p and the scaling factor to control the Lévy flight step sizes γ. An additional parameter that could be considered is constant λ of the Lévy distribution. However, since this value is more generally related to Lévy flights it was decided to assume in this study a fixed value of this parameter $\lambda = 3/2$, following the recommendations of Yang (2012). Based on the previous observations, the FPA parameters vector examined herein is the vector $\boldsymbol{p} = (n, p, \gamma)$.

The method employed in this study for tuning FPA parameters is a simple non-iterative, single-stage sampling method. This decision is made because FPA relies on a very limited number of independent parameters. Furthermore, in practice, these parameters are given limited rounded values. This effectively means that a sampling method of adequate density can represent sufficiently the range of parameters used in FPA practice. Moreover, this approach offers insight into how the performance of the FPA algorithm varies with its parameters revealing useful information regarding its robustness, distribution of solution quality and sensitivity. Nevertheless, it is clarified that the adopted approach may not be the most appropriate in cases where the degree of solution accuracy sought and/or the number of algorithm parameters are high since the associated computational cost can become prohibitive.

In the rest of this study, the following discrete values sets are examined for the different FPA parameters. An exhaustive search of all $5 \cdot 6 \cdot 5 = 150$ combinations of these parameter values is conducted to specify the optimal parameter vector $\boldsymbol{p}^*$ that maximizes FPA performance. Each of these combinations is termed as an FPA *instance* following the terminology recommended in Eiben and Smit (2011).

$$n = (20, 40, 60, 80, 100),$$
$$p = (0, 0.2, 0.4, 0.6, 0.8, 1.0)$$
$$\gamma = (10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1)$$

4 Optimization problems

As described in the previous section, the optimization problems related to parameter tuning can be described by Eqs (5) or (6). The first component of these problem settings is the objective function $f(\boldsymbol{x})$ that is bound to be minimized. In this study, the set of functions

specified in IEEE-CEC'13 (Liang et al. 2013) for real-parameter single-objective optimization problems is employed. This set is comprised of 28 benchmark functions f_i (i =1, 2, …, 28) shown in Table 1 (Liang et al. 2013) together with their global optimum values. All f_i functions represent minimization problems with a variable number of dimensions d. Three different dimensions are examined herein: d = 5, 10 and 20. All test functions are scalable and shifted to o = [o_1, o_2, …, 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 (Liang et al. 2013).

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 FPA instance (i.e. set of parameters), 20 independent runs are performed for each function f_i ($i = 1$ to 28) and number of dimensions d (i.e. $d = 5, 10, 20$), conducting in total $10000 \cdot d$ maximum function evaluations (MaxFES) for each run as recommended in Liang *et al.* (2013). 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 $N = (0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0) \cdot$ MaxFES to assess the speed of convergence of the different instances following the recommendations in Liang *et al.* (2013). It is recalled herein that an FPA instance is specified by a combination of the following parameters:

- n
- p
- γ

An instance run is assumed to converge to a global optimum of a function when the solution error becomes less than 10^{-8}. Since FPA instances do not necessarily converge to the global optima within $10000 \cdot d$ MaxFES, especially for the complex multi-modal functions of Table 1, the parameter tuning optimization problems addressed herein assume the form of Eq (6), where the number of function evaluations N takes all eleven values where function errors are recorded. In this manner, the variation of the optimal FPA parameters as a function of the provided computational effort will also be identified.

The total number of optimization problems examined in the following is equal to 28 different functions times 3 different dimensions and 11 different numbers of function evaluations (i.e. $28 \cdot 3 \cdot 11 = 924$). For each of these problems, the performance of all 150 FPA instances is examined and compared in terms of means or standard deviations of the error values calculated from the 20 independent runs. The parameters vector of the FPA instance with the smallest mean error or standard deviation of errors represents the optimal parameter tuning for the problem under investigation.

5 Calculation results

In this section, the main results obtained by the different FPA instances for the optimization problems of §4 are presented. Tables 2-7 present the best mean errors and corresponding standard deviations achieved by all FPA instances from 20 independent runs for all the optimization problems described above. It is important to clarify that these results do not

originate from a specific instance but from the optimal instances for each optimization problem. It is clear, in these tables, that the errors are more significant for the complex multi-modal functions. Furthermore, they always reduce with the number of function evaluations, but they tend to be higher as the number of dimensions increases. It is also worthy of mentioning that for most of the unimodal functions convergence to global optimum is achieved for all 20 independent runs from one or more FPA instances. For the cases where all runs of more than one instances converge to a global optimum, then the best FPA instance is defined herein as the one that achieved on average the earliest convergence. This effectively means that the optimization problem assumed in these cases takes the form of Eq. (5).

Table 2: Best mean solution errors of all FPA instances for the CEC'13 test functions with $d = 5$

Function	Number of function evaluations normalized to MaxFES = 10000·d										
	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
1	1.20E+02	9.96E-09	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	9.88E-09	1.00E-08	1.00E-08	1.00E-08
2	1.29E+06	4.32E-02	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	9.88E-09
3	1.56E+08	1.39E+01	1.25E-02	1.08E-07	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08
4	8.11E+03	6.39E-04	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08
5	5.97E+01	1.89E-07	9.86E-09	1.00E-08	9.78E-09	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08
6	5.41E+00	1.83E-01	2.51E-06	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08
7	3.17E+01	7.09E-01	1.54E-01	3.08E-02	1.11E-02	5.73E-03	7.67E-04	2.29E-04	4.79E-05	1.54E-05	1.44E-05
8	1.99E+01	1.89E+01	1.77E+01	1.66E+01	1.55E+01	1.38E+01	1.21E+01	1.01E+01	9.35E+00	8.93E+00	8.73E+00
9	3.53E+00	1.56E+00	1.09E+00	7.62E-01	5.33E-01	4.26E-01	3.35E-01	2.98E-01	2.64E-01	2.50E-01	2.41E-01
10	3.17E+01	1.98E-01	8.52E-02	5.33E-02	3.32E-02	2.68E-02	1.96E-02	1.52E-02	1.22E-02	9.88E-03	7.08E-03
11	2.47E+01	1.98E+00	7.89E-01	3.02E-01	2.04E-01	1.19E-01	7.60E-02	4.97E-02	6.67E-04	1.19E-06	1.30E-08
12	2.86E+01	4.52E+00	2.28E+00	1.12E+00	6.53E-01	4.56E-01	4.49E-01	4.48E-01	4.12E-01	3.10E-01	2.53E-01
13	2.89E+01	6.22E+00	3.26E+00	2.40E+00	1.99E+00	1.89E+00	1.50E+00	1.18E+00	1.05E+00	8.22E-01	6.23E-01
14	5.87E+02	1.66E+02	1.04E+02	8.60E+01	5.53E+01	4.41E+01	3.16E+01	2.80E+01	2.58E+01	2.31E+01	2.26E+01
15	7.11E+02	3.14E+02	2.08E+02	1.30E+02	9.39E+01	7.79E+01	6.27E+01	5.43E+01	4.38E+01	4.03E+01	3.72E+01
16	1.80E+00	4.60E-01	3.48E-01	2.78E-01	2.53E-01	2.34E-01	2.11E-01	2.02E-01	1.84E-01	1.75E-01	1.68E-01
17	3.39E+01	1.11E+01	7.86E+00	7.04E+00	6.42E+00	6.15E+00	5.86E+00	5.36E+00	4.83E+00	4.37E+00	4.20E+00
18	3.60E+01	1.30E+01	1.03E+01	8.36E+00	7.31E+00	6.61E+00	6.28E+00	5.80E+00	5.37E+00	5.00E+00	4.67E+00
19	4.24E+00	4.11E-01	2.43E-01	1.84E-01	1.61E-01	1.47E-01	1.30E-01	1.21E-01	1.13E-01	9.64E-02	9.23E-02
20	1.64E+00	9.49E-01	6.82E-01	5.47E-01	4.45E-01	3.73E-01	3.15E-01	2.80E-01	2.63E-01	2.49E-01	2.36E-01
21	4.19E+02	1.24E+02	9.70E+01	7.78E+01	6.45E+01	5.30E+01	4.39E+01	3.87E+01	3.65E+01	3.57E+01	3.53E+01
22	8.64E+02	3.77E+02	2.97E+02	2.58E+02	2.40E+02	2.25E+02	2.00E+02	1.92E+02	1.82E+02	1.70E+02	1.63E+02
23	9.46E+02	5.26E+02	4.20E+02	3.41E+02	2.99E+02	2.79E+02	2.51E+02	2.36E+02	2.32E+02	2.27E+02	2.24E+02
24	1.53E+02	1.12E+02	1.04E+02	1.00E+02	9.96E+01	9.84E+01	9.57E+01	9.05E+01	8.92E+01	8.51E+01	8.28E+01
25	1.48E+02	1.09E+02	1.04E+02	1.02E+02	1.02E+02	1.01E+02	1.00E+02	9.88E+01	9.73E+01	9.69E+01	9.67E+01
26	1.42E+02	9.88E+01	8.97E+01	8.29E+01	7.49E+01	7.18E+01	6.92E+01	6.60E+01	5.84E+01	5.74E+01	5.65E+01
27	4.00E+02	3.40E+02	3.14E+02	2.88E+02	2.85E+02	2.80E+02	2.78E+02	2.73E+02	2.61E+02	2.59E+02	2.58E+02
28	3.73E+02	1.79E+02	1.10E+02	9.29E+01	8.09E+01	7.06E+01	6.73E+01	6.39E+01	6.17E+01	5.97E+01	5.71E+01

11

Table 3: Best mean solution errors of all FPA instances for the CEC'13 test functions with $d = 10$

Function	Number of function evaluations normalized to MaxFES = 10000·d										
	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
1	1.23E+03	1.12E-05	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08
2	5.37E+06	2.15E+01	3.55E-05	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	9.76E-09	9.99E-09	1.00E-08
3	5.93E+09	1.08E+07	8.69E+04	5.04E+03	4.19E+01	1.12E+00	6.30E-01	3.39E-01	1.97E-01	9.74E-02	4.70E-02
4	1.11E+04	2.48E-01	1.62E-06	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08
5	1.71E+02	5.74E-04	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08
6	1.04E+02	3.71E+00	2.87E-01	2.39E-03	1.08E-06	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08
7	1.17E+02	2.65E+01	1.25E+01	8.29E+00	6.25E+00	5.30E+00	3.64E+00	2.59E+00	2.00E+00	1.68E+00	1.48E+00
8	2.07E+01	2.05E+01	2.04E+01	2.04E+01	2.04E+01	2.03E+01	2.03E+01	2.03E+01	2.03E+01	2.03E+01	2.03E+01
9	9.96E+00	6.18E+00	4.41E+00	3.68E+00	3.37E+00	3.21E+00	3.10E+00	3.05E+00	3.03E+00	3.02E+00	2.99E+00
10	1.49E+02	2.91E-01	5.61E-02	3.42E-02	2.94E-02	2.91E-02	2.90E-02	2.85E-02	2.38E-02	2.17E-02	1.95E-02
11	8.29E+01	1.62E+01	7.28E+00	4.95E+00	3.63E+00	3.46E+00	3.33E+00	2.79E+00	2.39E+00	2.14E+00	1.84E+00
12	9.02E+01	2.35E+01	1.18E+01	9.86E+00	9.25E+00	9.16E+00	9.03E+00	7.79E+00	6.86E+00	6.32E+00	6.03E+00
13	9.11E+01	3.10E+01	1.87E+01	1.60E+01	1.55E+01	1.44E+01	1.31E+01	1.22E+01	1.14E+01	1.07E+01	1.04E+01
14	1.81E+03	9.23E+02	6.14E+02	5.02E+02	3.91E+02	2.60E+02	2.30E+02	2.00E+02	1.98E+02	1.81E+02	1.62E+02
15	1.82E+03	1.13E+03	9.30E+02	7.98E+02	6.83E+02	6.22E+02	5.86E+02	5.57E+02	5.50E+02	5.36E+02	5.31E+02
16	1.60E+00	6.55E-01	5.18E-01	4.43E-01	4.15E-01	3.94E-01	3.83E-01	3.64E-01	3.60E-01	3.45E-01	3.37E-01
17	1.14E+02	4.29E+01	3.05E+01	2.42E+01	2.15E+01	2.04E+01	1.89E+01	1.77E+01	1.73E+01	1.66E+01	1.61E+01
18	1.23E+02	4.34E+01	3.27E+01	2.78E+01	2.44E+01	2.23E+01	2.05E+01	1.85E+01	1.78E+01	1.72E+01	1.69E+01
19	5.72E+01	2.17E+00	1.20E+00	8.60E-01	7.26E-01	6.59E-01	6.08E-01	5.87E-01	5.48E-01	5.28E-01	5.21E-01
20	4.29E+00	3.58E+00	3.35E+00	3.13E+00	3.05E+00	2.95E+00	2.86E+00	2.80E+00	2.76E+00	2.74E+00	2.71E+00
21	4.86E+02	2.08E+02	1.40E+02	1.11E+02	9.70E+01	9.15E+01	9.04E+01	9.01E+01	9.00E+01	9.00E+01	9.00E+01
22	2.12E+03	7.64E+02	4.61E+02	3.73E+02	3.65E+02	3.25E+02	3.11E+02	3.11E+02	2.96E+02	2.89E+02	2.85E+02
23	2.18E+03	1.46E+03	1.23E+03	1.01E+03	9.09E+02	8.27E+02	7.92E+02	7.78E+02	7.70E+02	7.53E+02	7.36E+02
24	2.23E+02	1.73E+02	1.56E+02	1.50E+02	1.44E+02	1.37E+02	1.34E+02	1.31E+02	1.28E+02	1.24E+02	1.22E+02
25	2.28E+02	2.04E+02	1.89E+02	1.84E+02	1.80E+02	1.76E+02	1.74E+02	1.72E+02	1.72E+02	1.70E+02	1.66E+02
26	2.00E+02	1.45E+02	1.28E+02	1.23E+02	1.22E+02	1.20E+02	1.16E+02	1.14E+02	1.12E+02	1.10E+02	1.09E+02
27	6.20E+02	4.15E+02	3.94E+02	3.86E+02	3.80E+02	3.71E+02	3.62E+02	3.52E+02	3.46E+02	3.40E+02	3.36E+02
28	8.70E+02	2.34E+02	1.67E+02	1.34E+02	1.16E+02	1.08E+02	1.03E+02	1.01E+02	1.00E+02	1.00E+02	1.00E+02

Table 4: Best mean solution errors of all FPA instances for the CEC'13 test functions with $d = 20$

Function	Number of function evaluations normalized to MaxFES = 10000·d										
	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
1	3.18E+03	7.36E-06	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08
2	2.49E+07	5.68E+04	2.30E+02	3.09E-01	8.26E-04	4.85E-07	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08
3	4.39E+10	8.77E+07	6.96E+06	1.83E+06	1.81E+05	1.65E+04	1.01E+03	1.65E+02	1.33E+02	1.23E+02	9.17E+01
4	2.52E+04	3.60E+02	2.13E+00	1.27E-02	6.93E-05	2.99E-07	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08
5	6.85E+02	4.77E-03	1.08E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	9.97E-09	1.00E-08
6	4.91E+02	6.71E-01	9.12E-02	9.10E-04	1.16E-06	1.00E-08	1.00E-08	1.00E-08	1.00E-08	1.00E-08	9.96E-09
7	2.19E+02	4.22E+01	3.16E+01	2.41E+01	2.02E+01	1.78E+01	1.59E+01	1.42E+01	1.31E+01	1.21E+01	1.13E+01
8	2.10E+01	2.08E+01	2.08E+01	2.08E+01	2.08E+01	2.08E+01	2.08E+01	2.07E+01	2.07E+01	2.07E+01	2.07E+01
9	2.41E+01	1.73E+01	1.46E+01	1.34E+01	1.31E+01	1.29E+01	1.26E+01	1.22E+01	1.19E+01	1.15E+01	1.12E+01
10	6.18E+02	1.05E+00	8.70E-02	8.36E-03	5.10E-03	4.09E-04	1.62E-05	4.23E-07	1.39E-08	1.00E-08	1.00E-08
11	2.03E+02	5.17E+01	2.54E+01	1.98E+01	1.94E+01	1.85E+01	1.76E+01	1.75E+01	1.71E+01	1.45E+01	1.22E+01
12	2.06E+02	6.23E+01	3.60E+01	3.44E+01	3.30E+01	2.75E+01	2.39E+01	2.19E+01	2.10E+01	2.07E+01	2.05E+01
13	1.99E+02	9.74E+01	6.95E+01	6.11E+01	5.81E+01	5.49E+01	5.12E+01	4.87E+01	4.78E+01	4.56E+01	4.25E+01

14	3.95E+03	1.99E+03	1.64E+03	1.42E+03	1.25E+03	1.17E+03	1.15E+03	1.11E+03	1.09E+03	1.04E+03	1.03E+03
15	4.18E+03	2.81E+03	2.52E+03	2.19E+03	1.99E+03	1.88E+03	1.79E+03	1.78E+03	1.73E+03	1.70E+03	1.70E+03
16	1.96E+00	9.20E-01	7.84E-01	7.14E-01	6.56E-01	6.22E-01	5.99E-01	5.86E-01	5.64E-01	5.56E-01	5.36E-01
17	2.95E+02	1.22E+02	8.60E+01	6.87E+01	5.98E+01	5.49E+01	5.13E+01	4.87E+01	4.56E+01	4.35E+01	4.27E+01
18	3.00E+02	1.32E+02	9.76E+01	7.86E+01	6.61E+01	6.00E+01	5.71E+01	5.56E+01	5.31E+01	5.18E+01	5.09E+01
19	1.09E+03	7.78E+00	4.03E+00	2.74E+00	2.23E+00	2.01E+00	1.98E+00	1.82E+00	1.69E+00	1.62E+00	1.56E+00
20	9.85E+00	8.53E+00	8.01E+00	7.61E+00	7.35E+00	7.11E+00	7.01E+00	6.95E+00	6.92E+00	6.89E+00	6.86E+00
21	1.07E+03	3.35E+02	3.03E+02	2.58E+02	2.06E+02	1.67E+02	1.51E+02	1.44E+02	1.41E+02	1.39E+02	1.38E+02
22	4.33E+03	2.50E+03	2.02E+03	1.73E+03	1.52E+03	1.48E+03	1.39E+03	1.36E+03	1.33E+03	1.30E+03	1.30E+03
23	4.89E+03	3.70E+03	3.09E+03	2.76E+03	2.52E+03	2.37E+03	2.29E+03	2.28E+03	2.27E+03	2.27E+03	2.27E+03
24	2.68E+02	2.47E+02	2.41E+02	2.38E+02	2.36E+02	2.36E+02	2.36E+02	2.35E+02	2.34E+02	2.33E+02	2.32E+02
25	2.69E+02	2.52E+02	2.45E+02	2.44E+02	2.42E+02	2.42E+02	2.41E+02	2.39E+02	2.38E+02	2.38E+02	2.37E+02
26	2.12E+02	2.00E+02	2.00E+02	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	9.65E+02	6.73E+02	5.02E+02	4.74E+02	4.71E+02	4.69E+02	4.44E+02	4.27E+02	4.18E+02	4.12E+02	4.09E+02
28	2.81E+03	9.37E+02	5.63E+02	5.39E+02	3.99E+02	3.55E+02	3.50E+02	3.49E+02	3.49E+02	3.48E+02	3.36E+02

Table 5: Best error standard deviations of all FPA instances for the CEC'13 test functions with $d = 5$

Function	Number of function evaluations normalized to MaxFES = 10000·d										
	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
1	9.42E+01	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24
2	1.03E+06	4.76E-02	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24
3	2.09E+08	8.26E+00	4.67E-02	4.34E-07	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24
4	4.59E+03	5.36E-04	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24
5	2.76E+01	2.75E-07	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24
6	3.28E+00	1.75E-01	5.48E-06	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24
7	9.53E+00	5.94E-01	1.15E-01	5.24E-02	3.69E-02	1.71E-02	1.44E-03	7.94E-04	1.49E-04	4.37E-05	4.35E-05
8	9.21E-02	4.39E-02	3.38E-02	3.60E-02	3.70E-02	2.98E-02	3.14E-02	2.86E-02	2.91E-02	2.90E-02	2.60E-02
9	3.60E-01	2.39E-01	1.86E-01	2.04E-01	2.15E-01	2.03E-01	2.04E-01	2.01E-01	1.61E-01	1.56E-01	1.42E-01
10	1.28E+01	6.03E-02	3.02E-02	2.10E-02	1.34E-02	1.11E-02	9.75E-03	7.18E-03	5.03E-03	6.10E-03	5.75E-03
11	4.98E+00	9.80E-01	5.71E-01	4.66E-01	4.06E-01	3.12E-01	2.46E-01	2.22E-01	2.88E-03	4.63E-06	1.18E-08
12	6.49E+00	1.66E+00	1.02E+00	6.77E-01	5.43E-01	6.22E-01	6.42E-01	4.79E-01	4.92E-01	4.42E-01	3.46E-01
13	5.73E+00	2.25E+00	1.61E+00	1.31E+00	1.36E+00	1.00E+00	8.41E-01	7.11E-01	7.48E-01	6.65E-01	6.16E-01
14	7.83E+01	5.90E+01	4.25E+01	3.88E+01	3.50E+01	2.91E+01	9.97E+00	8.74E+00	6.55E+00	7.68E+00	7.69E+00
15	1.02E+02	5.71E+01	4.87E+01	5.96E+01	4.80E+01	4.87E+01	4.16E+01	4.09E+01	3.73E+01	3.81E+01	3.80E+01
16	3.88E-01	1.14E-01	8.82E-02	5.55E-02	6.11E-02	5.42E-02	5.35E-02	5.04E-02	4.74E-02	4.37E-02	4.48E-02
17	6.09E+00	2.05E+00	1.69E+00	1.31E+00	1.18E+00	1.12E+00	1.04E+00	9.38E-01	9.18E-01	8.11E-01	7.95E-01
18	6.34E+00	2.16E+00	1.76E+00	1.34E+00	1.43E+00	1.21E+00	1.20E+00	1.09E+00	1.02E+00	7.82E-01	9.08E-01
19	2.26E+00	1.23E-01	7.09E-02	6.44E-02	4.98E-02	5.96E-02	5.65E-02	5.18E-02	5.06E-02	4.69E-02	4.49E-02
20	7.20E-02	7.96E-02	8.44E-02	9.17E-02	8.90E-02	1.09E-01	1.04E-01	1.05E-01	1.10E-01	9.85E-02	8.24E-02
21	4.07E+01	1.83E+01	1.60E+01	1.31E+01	1.33E+01	1.20E+01	2.11E+01	2.30E+01	2.10E+01	2.12E+01	2.18E+01
22	7.88E+01	5.73E+01	5.57E+01	5.00E+01	4.72E+01	3.56E+01	3.68E+01	3.43E+01	4.11E+01	3.83E+01	3.62E+01
23	9.70E+01	5.73E+01	5.92E+01	5.72E+01	5.98E+01	5.01E+01	4.49E+01	4.77E+01	4.69E+01	4.80E+01	4.34E+01
24	9.62E+00	3.48E+00	2.45E+00	2.66E+00	2.66E+00	2.36E+00	2.20E+00	3.74E+00	3.59E+00	3.54E+00	3.44E+00
25	9.88E+00	2.86E+00	1.79E+00	1.20E+00	1.41E+00	1.42E+00	1.31E+00	1.06E+00	5.16E-01	1.30E-01	3.82E-02
26	9.68E+00	2.88E+00	1.94E+00	1.18E+00	7.88E-01	8.50E-01	7.68E-01	7.70E-01	7.30E-01	7.29E-01	7.29E-01
27	1.00E+01	7.22E+00	7.64E+00	7.91E+00	7.90E+00	7.31E+00	7.41E+00	7.56E+00	6.79E+00	5.29E+00	4.91E+00
28	2.58E+01	2.52E+01	2.08E+01	6.33E+00	8.85E+00	1.60E+01	1.14E+01	1.62E+01	1.33E+01	1.82E+01	1.80E+01

Table 6: Best error standard deviations of all FPA instances for the CEC'13 test functions with $d = 10$

Function	Number of function evaluations normalized to MaxFES = 10000·d										
	0.01	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
1	5.23E+02	1.08E-05	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24
2	2.83E+06	7.48E+01	1.09E-04	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24
3	2.55E+09	1.16E+07	1.41E+05	1.32E+04	1.29E+02	2.43E+00	1.42E+00	8.07E-01	4.86E-01	2.55E-01	1.31E-01
4	3.71E+03	6.92E-01	3.46E-06	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24
5	7.02E+01	4.28E-04	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24
6	3.41E+01	1.46E+00	3.64E-01	8.91E-03	4.46E-06	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24
7	2.63E+01	6.66E+00	5.94E+00	4.49E+00	4.01E+00	3.20E+00	2.21E+00	1.61E+00	1.39E+00	1.34E+00	1.32E+00
8	7.18E-02	5.90E-02	4.84E-02	4.54E-02	4.61E-02	4.40E-02	3.94E-02	4.31E-02	4.26E-02	3.80E-02	3.54E-02
9	4.65E-01	4.13E-01	3.35E-01	3.61E-01	3.81E-01	4.24E-01	3.92E-01	4.33E-01	4.23E-01	4.45E-01	4.41E-01
10	7.89E+01	6.16E-02	2.24E-02	1.78E-02	1.50E-02	1.16E-02	1.15E-02	1.09E-02	9.97E-03	9.85E-03	8.79E-03
11	9.02E+00	4.68E+00	2.81E+00	1.92E+00	1.76E+00	1.48E+00	1.45E+00	1.45E+00	1.41E+00	1.31E+00	1.22E+00
12	1.27E+01	5.12E+00	4.08E+00	3.10E+00	3.09E+00	2.86E+00	2.46E+00	2.26E+00	1.89E+00	1.78E+00	1.76E+00
13	1.26E+01	4.81E+00	4.06E+00	3.39E+00	3.57E+00	3.64E+00	3.84E+00	3.99E+00	3.43E+00	3.77E+00	3.40E+00
14	1.17E+02	8.58E+01	6.95E+01	7.39E+01	7.03E+01	6.13E+01	6.99E+01	7.29E+01	6.97E+01	7.14E+01	7.46E+01
15	1.24E+02	8.31E+01	6.94E+01	8.30E+01	7.89E+01	8.18E+01	7.66E+01	7.99E+01	8.14E+01	7.37E+01	7.59E+01
16	3.24E-01	1.10E-01	8.83E-02	7.00E-02	6.20E-02	6.18E-02	5.92E-02	5.87E-02	5.96E-02	5.23E-02	5.54E-02
17	1.74E+01	4.91E+00	5.02E+00	3.70E+00	3.56E+00	3.24E+00	2.75E+00	2.55E+00	2.64E+00	2.09E+00	2.25E+00
18	1.67E+01	4.58E+00	3.06E+00	3.04E+00	3.66E+00	3.01E+00	2.96E+00	2.78E+00	2.49E+00	2.31E+00	2.31E+00
19	7.61E+01	5.04E-01	2.67E-01	2.15E-01	1.66E-01	1.49E-01	1.28E-01	1.44E-01	1.38E-01	1.47E-01	1.43E-01
20	3.41E-05	5.09E-03	5.09E-03	5.09E-03	5.09E-03	5.09E-03	5.09E-03	5.09E-03	5.09E-03	5.09E-03	5.09E-03
21	3.22E+01	2.04E+01	2.60E+01	1.22E+01	1.93E+01	1.87E+01	1.84E+01	2.09E+01	1.94E+01	1.98E+01	1.68E+01
22	1.22E+02	9.07E+01	8.34E+01	8.00E+01	8.79E+01	9.09E+01	6.63E+01	6.74E+01	7.35E+01	7.21E+01	7.16E+01
23	1.15E+02	9.47E+01	9.90E+01	1.05E+02	9.25E+01	8.79E+01	8.47E+01	7.66E+01	8.22E+01	8.32E+01	8.18E+01
24	1.40E+00	2.91E+00	3.29E+00	3.16E+00	3.17E+00	3.23E+00	3.28E+00	3.29E+00	3.30E+00	3.25E+00	3.24E+00
25	1.70E+00	2.40E+00	2.77E+00	2.42E+00	2.76E+00	2.87E+00	2.92E+00	2.70E+00	2.73E+00	2.61E+00	2.88E+00
26	3.11E+00	1.57E-01	4.13E+00	2.60E+00	1.80E+00	1.33E+00	2.23E+00	2.17E+00	2.24E+00	2.34E+00	2.31E+00
27	1.66E+01	1.24E+01	4.05E+00	1.68E+00	2.39E+00	2.56E+00	2.83E+00	1.67E+00	1.09E+00	7.90E-01	6.24E-01
28	7.10E+01	2.07E+01	7.97E-03	2.46E-06	4.69E-10	2.15E-13	2.02E-13	2.02E-13	2.02E-13	2.02E-13	2.02E-13

Table 7: Best error standard deviations of all FPA instances for the CEC'13 test functions with $d = 20$

Function	Number of function evaluations normalized to MaxFES = 10000·d										
	0.01	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
1	8.31E+02	4.47E-06	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24
2	1.00E+07	4.49E+04	2.85E+02	3.82E-01	2.64E-03	1.55E-06	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24
3	2.35E+10	6.35E+07	1.07E+07	2.22E+06	3.05E+05	2.91E+04	1.87E+03	4.51E+02	4.45E+02	4.30E+02	4.03E+02
4	5.31E+03	2.27E+02	1.40E+00	8.11E-03	3.08E-05	1.38E-07	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24
5	1.83E+02	2.81E-03	2.08E-09	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24
6	1.52E+02	8.28E-01	1.57E-01	1.54E-03	3.95E-06	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24	3.39E-24
7	5.90E+01	8.80E+00	6.44E+00	4.93E+00	4.09E+00	3.76E+00	4.07E+00	3.84E+00	3.74E+00	3.65E+00	3.35E+00
8	4.84E-02	3.36E-02	3.42E-02	3.37E-02	3.25E-02	3.08E-02	3.26E-02	3.38E-02	3.23E-02	3.09E-02	3.09E-02
9	6.97E-01	5.43E-01	5.46E-01	5.40E-01	5.50E-01	4.53E-01	5.39E-01	5.93E-01	5.76E-01	6.37E-01	6.38E-01
10	1.99E+02	2.88E-01	1.27E-02	9.44E-03	5.82E-03	5.47E-04	2.58E-05	7.69E-07	8.92E-09	3.39E-24	3.39E-24
11	1.82E+01	7.93E+00	7.30E+00	6.83E+00	5.85E+00	5.38E+00	3.93E+00	4.28E+00	3.79E+00	4.14E+00	3.59E+00
12	1.56E+01	8.64E+00	8.26E+00	6.68E+00	6.89E+00	6.51E+00	6.31E+00	5.68E+00	4.92E+00	4.72E+00	4.51E+00
13	1.69E+01	1.03E+01	8.30E+00	7.80E+00	7.60E+00	6.24E+00	6.83E+00	5.52E+00	7.48E+00	6.73E+00	8.00E+00

14	1.54E+02	1.24E+02	9.80E+01	9.17E+01	9.97E+01	1.05E+02	1.06E+02	1.07E+02	9.81E+01	1.08E+02	1.07E+02
15	1.49E+02	1.25E+02	1.05E+02	1.26E+02	1.18E+02	1.04E+02	1.13E+02	1.22E+02	1.15E+02	1.04E+02	9.47E+01
16	2.53E-01	1.24E-01	8.90E-02	8.31E-02	7.94E-02	8.38E-02	6.19E-02	5.78E-02	5.75E-02	7.31E-02	5.79E-02
17	3.03E+01	9.84E+00	7.77E+00	5.76E+00	5.18E+00	6.07E+00	5.36E+00	5.83E+00	5.17E+00	4.86E+00	4.85E+00
18	3.70E+01	9.54E+00	7.80E+00	6.80E+00	6.84E+00	6.63E+00	6.37E+00	5.83E+00	5.49E+00	5.37E+00	4.55E+00
19	1.17E+03	9.29E-01	8.02E-01	6.61E-01	5.66E-01	4.28E-01	3.53E-01	4.28E-01	4.41E-01	4.12E-01	3.67E-01
20	1.27E-13	1.45E-13	1.45E-13	3.04E-13	3.04E-13	3.04E-13	3.04E-13	3.04E-13	3.04E-13	3.04E-13	3.04E-13
21	6.66E+01	6.60E+00	8.46E+00	8.48E+00	6.75E+00	6.45E+00	1.91E+01	1.68E+01	1.45E+01	1.50E+01	1.41E+01
22	1.75E+02	1.53E+02	1.19E+02	1.52E+02	1.41E+02	1.47E+02	1.29E+02	1.14E+02	1.28E+02	1.27E+02	1.26E+02
23	1.67E+02	1.27E+02	1.31E+02	1.46E+02	1.35E+02	1.40E+02	1.34E+02	1.25E+02	1.35E+02	1.40E+02	1.36E+02
24	2.82E+00	2.18E+00	1.58E+00	2.15E+00	2.01E+00	2.13E+00	2.15E+00	2.27E+00	2.24E+00	2.25E+00	2.28E+00
25	2.97E+00	2.31E+00	1.98E+00	2.20E+00	2.14E+00	2.32E+00	2.46E+00	2.26E+00	2.25E+00	2.08E+00	2.40E+00
26	4.63E+00	2.27E-02	1.47E-03	3.43E-04	1.91E-04	1.31E-04	1.53E-04	1.74E-04	7.91E-05	1.10E-04	2.01E-04
27	2.16E+01	1.72E+01	1.96E+01	2.08E+01	2.34E+01	2.71E+01	2.98E+01	2.90E+01	2.44E+01	1.98E+01	1.85E+01
28	2.16E+02	9.04E+01	7.79E+01	8.06E+01	7.73E+01	7.58E+01	7.56E+01	9.57E+01	1.11E+02	1.05E+02	1.04E+02

Figure 2 presents, in the form of boxplots, the population sizes of the optimal FPA instances for all 28 CEC'13 functions as a function of the number of evaluations and the number of problem dimensions. In this figure, the optimal instances are determined on the basis of minimum mean errors. The box plots show the minimum, maximum and median (red line) n values. Inside the boxes, the 25th to 75th percentiles are contained. It can be seen that the optimal population sizes can take any value between 20 and 100, which are the boundaries assumed in this study, but the majority of optimal n values lies between 20 and 60 flowers. The optimal population sizes seem to be increasing as the function evaluations and the problem dimensions are increasing. The median optimal n value is 20 for almost all steps of function evaluations for $d = 5$ and the early evaluation steps for $d = 10$ and 20. However, the same median population size increases to 40 for the later evaluation steps of $d = 10$ and 20. These observations drive to the conclusion that larger population sizes can be more effective for high-dimensional problems and where higher computational effort is afforded.

Furthermore, Fig. 3 shows the population sizes of the optimal FPA instances for the same optimization problems as a function of the number of function evaluations and the number of problem dimensions when these instances are determined based on the minimum standard deviations of the errors of the 20 independent runs. These instances are termed here as 'robust' since they manage to minimize the uncertainty in the FPA outcomes. It can be seen that again the robust n values tend to increase with the number problem dimensions and function evaluations. The median values range between 40 and 60 for $d = 5$ and 10 but they can increase up to 80 for $d = 20$. Comparing these results with Fig. 2, it can be concluded that the optimal FPA instances based on minimum expected errors do not always match with the most robust instances and that larger population sizes tend to produce more robust FPA outcomes.
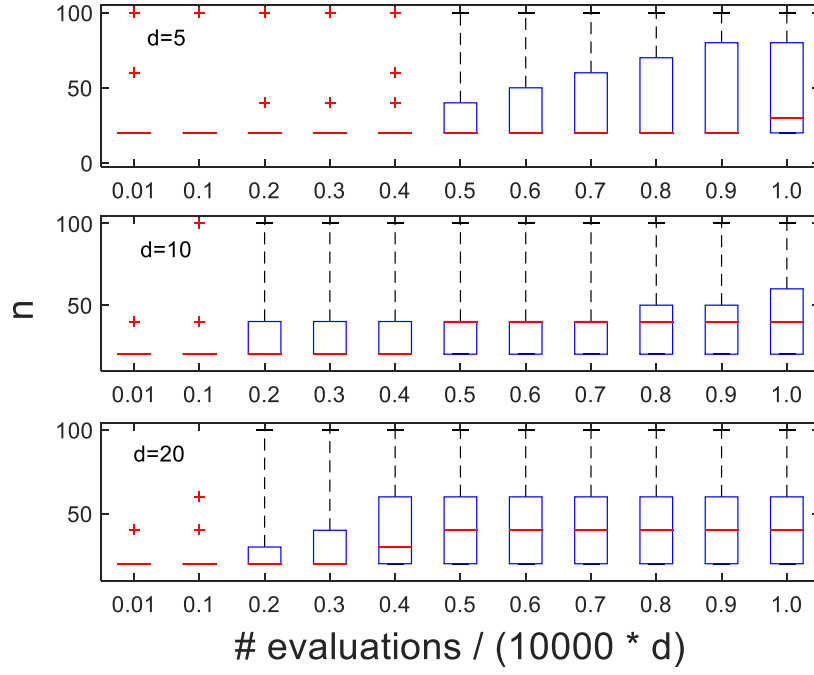
Fig. 2: Variation of optimal n values, based on minimum mean errors, with the number of function evaluations for different problem dimensions
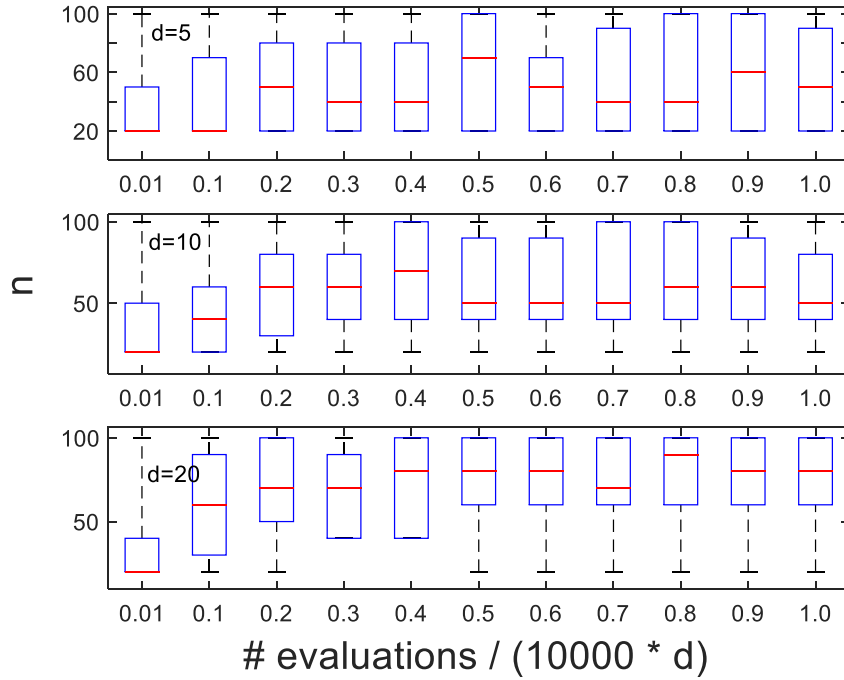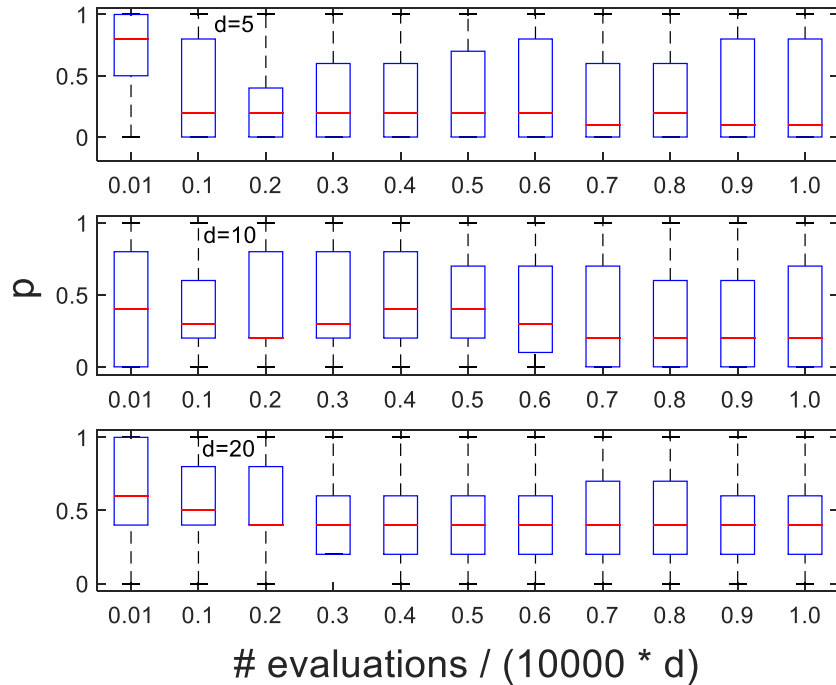

Fig. 3: Variation of optimal n values, based on minimum error standard deviations, with the number of function evaluations for different problem dimensions

Figure 4 presents the switch probabilities of the optimal FPA instances, based on minimum average errors, for all the objective functions under consideration as a function of the number of function evaluations and the number of problem dimensions. It is found that the optimal p

16

values tend to increase with the number of dimensions with a typical median of 0.2 for $d = 5$ and 10 and a median of 0.4 for $d = 20$. This observation shows that global (biotic) pollination can be more advantageous to high-dimensional problems. It is also interesting to note that the optimal p values tend to slightly decrease with the number of function evaluations especially for the low-dimensional problems. This decrease of the optimal switch probabilities in the later evaluation steps can be explained by the fact that it allows for more extensive exploitation of the search space via local pollination.

Moreover, Fig. 5 illustrates the most robust switch probabilities as a function of the computational effort and number of dimensions. It is observed that the most robust p values vary in large ranges with a typical median of 0.4 for $d = 5$ and 10 and of 0.6 for $d = 20$. The robust p values seem to be independent of the number of function evaluations. On the other hand, they tend to increase with the problem dimensions. When compared with the optimal switch probabilities based on minimum expected errors, the robust switch probabilities seem to obtain larger values. This observation could be explained by the fact that larger switch probabilities allow for more extensive exploration of the search space by biotic pollination and therefore it is less likely that FPA is trapped in local optima.

Fig. 4: Variation of optimal p values, based on minimum mean errors, with the number of function evaluations for different problem dimensions
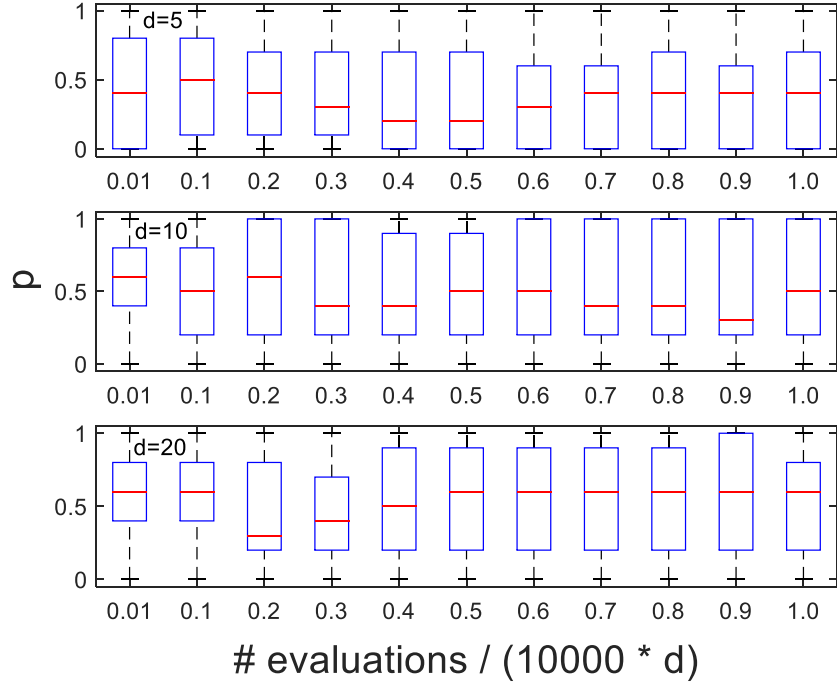
Fig. 5: Variation of optimal *p* values, based on minimum error standard deviations, with the number of function evaluations for different problem dimensions

Figure 6 illustrates the scale factors γ of the Lévy flight step sizes of the optimal FPA instances, based on minimum expected errors, of the different CEC'13 functions with respect to the number of objective function evaluations and problem dimensions. It is observed that the majority of optimal γ values range between 1 and 0.1, with a typical median of 0.1 for d = 5 and 10 and of 1.0 for the last evaluation steps of d = 20. The latter observation may be due to the fact that high-dimensional problems require larger Lévy flight steps for the search space to be sufficiently explored. The optimal γ values do not seem to be significantly influenced by the amount of function evaluations.

In addition, Fig. 7 shows the most robust γ values as a function of the computational effort and number of dimensions. It is observed that most of the robust γ values vary in broader ranges than previously (i.e. between 1 and 0.0001) with a typical median of 0.1 in the very early evaluation steps and 0.01 in the later steps. These results do not seem to be significantly affected by the number of problem dimensions. When compared with the optimal γ values based on minimum expected errors, the robust scale factors seem to get smaller values. This observation could be explained by the fact that smaller step sizes allow for better exploitation of the search space.
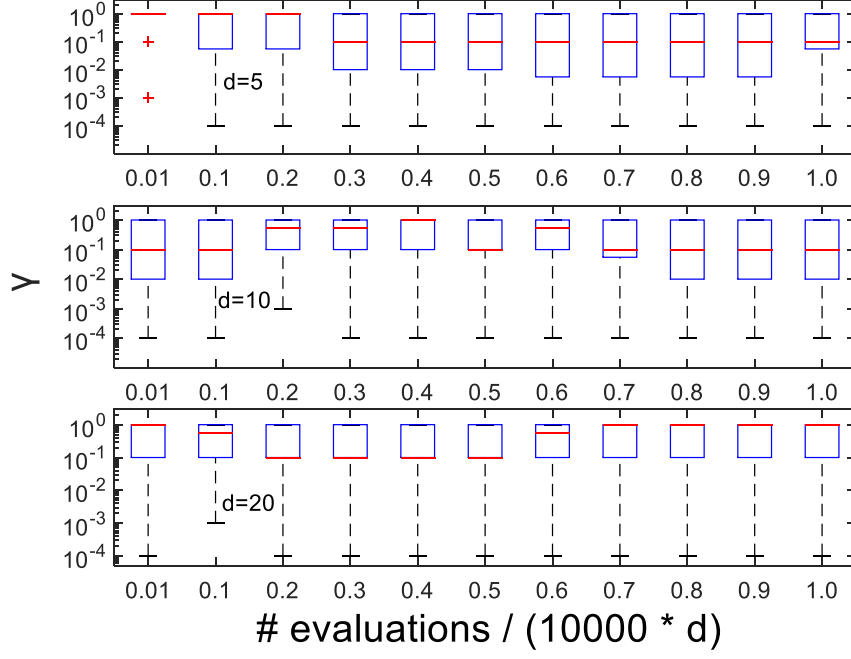
Fig. 6: Variation of optimal γ values, based on minimum mean errors, with the number of function evaluations for different problem dimensions
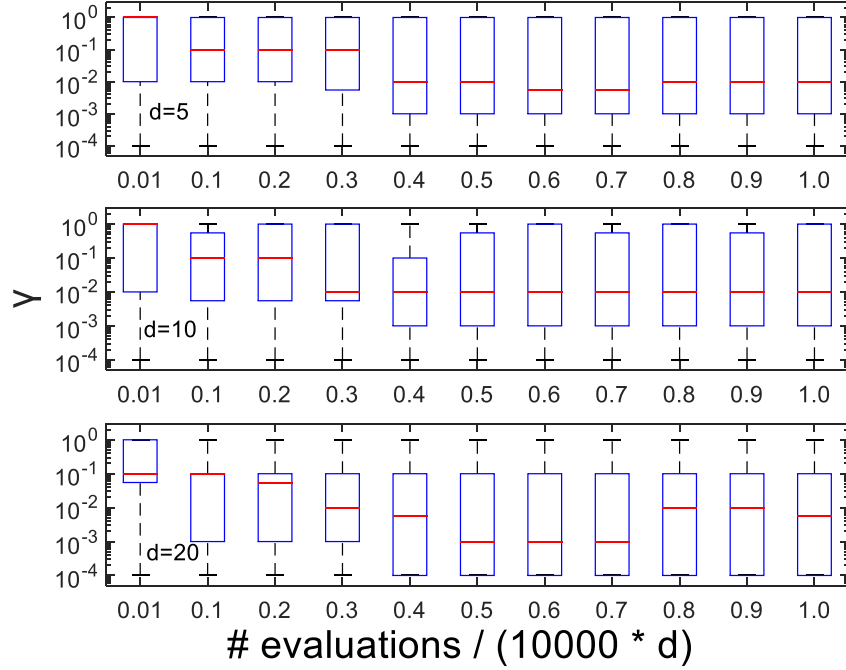

Fig. 7: Variation of optimal γ values, based on minimum error standard deviations, with the number of function evaluations for different problem dimensions

From the previous discussions, it can be concluded that the optimal instances for minimum average errors and error standard deviations of the 20 independent runs do not necessarily coincide. Hence, it is interesting to examine the Pareto fronts of average errors versus error standard deviations obtained from the various FPA instances. Fig. 8 presents such Pareto fronts obtained for the objective function f_7 with $d = 20$ after $4000 \cdot d$ and $10000 \cdot d$ function

evaluations. It can be seen in this figure that the best mean error drops considerably from 20.2 for 4000·d function evaluations to 11.4 for 10000·d function evaluations. On the other hand, the minimum error standard deviation drops slightly from 4.1 to 3.4 for the same numbers of function evaluations. The points that correspond to minimum standard deviations have mean errors of approximately 35.6 for 4000·d and 17.5 for 10000·d function evaluations. This means that, for the specific optimization problems, the mean errors of the robust solutions are 1.8 and 1.5 times higher than the minimum mean errors for 4000·d and 10000·d function evaluations, respectively.

To obtain a broader image of the trade-offs between the robust optimal solutions and the solutions for minimum mean errors, Fig. 9 presents the ratios of the mean errors of the robust optimal solutions to the minimum mean errors obtained for all optimization problems as a function of the number of function evaluations and the number of problem dimensions. It is interesting to note that the vast majority of these ratios range between 1 and 2 with typical median values ranging between 1.1 and 1.4. This shows that the robust solutions and the solutions for minimum mean errors are typically in close proximity. The ratios seem to increase slightly with the number of problem dimensions but there is no clear trend between these ratios and the number of function evaluations.
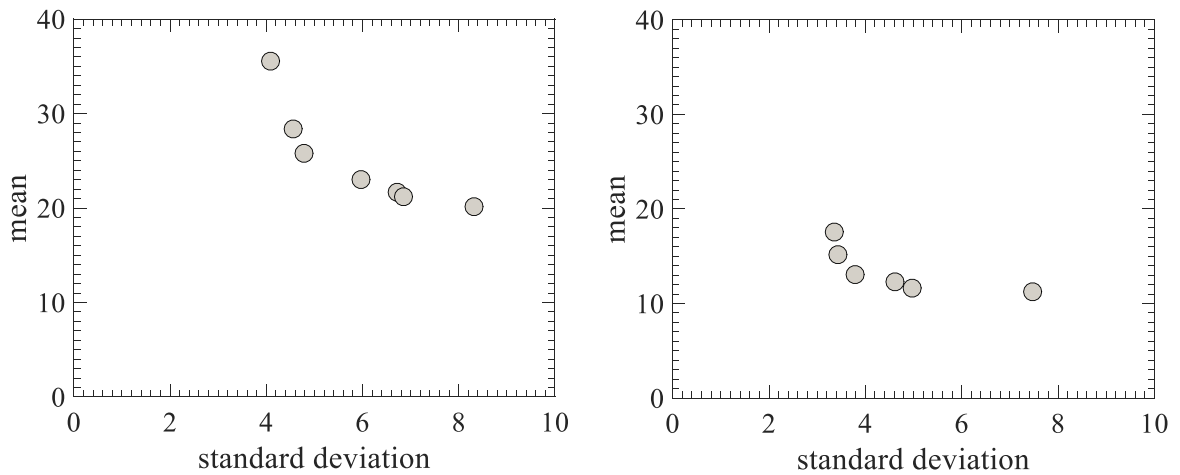
Fig. 8: Mean error vs standard error deviation Pareto fronts of function f_7 and $d = 20$ after a) 4000·d; b) 10000·d function evaluations
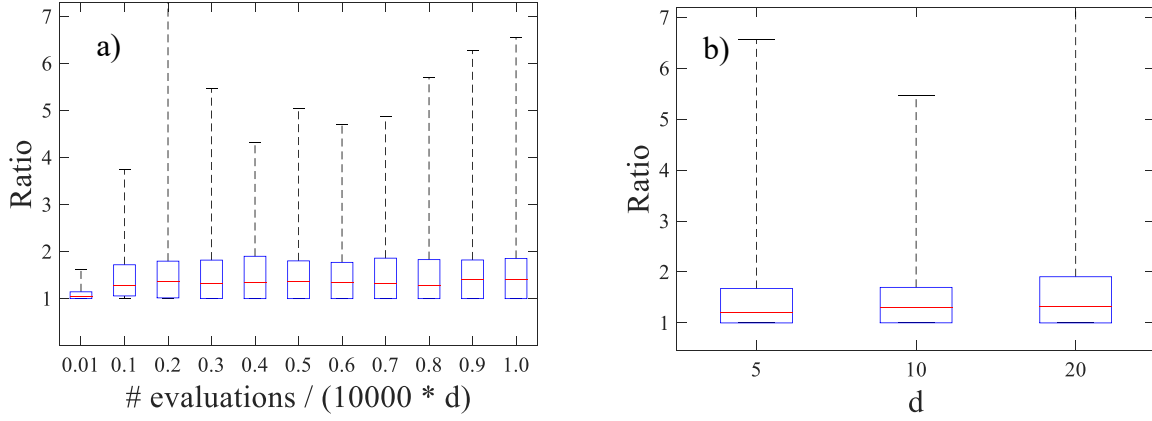
Fig. 9: Ratios of the mean errors of the robust optimal solutions to the minimum mean errors obtained for all optimization problems as a function of a) function evaluations and b) the problem dimensions

In the previous, the ranges of optimal parameter values are presented for the various optimization problems consisting of different objective functions, number of function evaluations and number of problem dimensions. Comparisons are also presented between the most robust FPA instances and the FPA instances yielding the minimum average errors after a number of independent runs. However, for practical applications, it is also important that specific parameter values are recommended herein that are also independent of the examined objective functions. The latter is useful because in practice FPA is applied to objective functions other than the ones specified in Table 1.

To recommend these optimal parameter values, for each objective function, number of function evaluations and problem dimensions, the FPA instances are ranked according to their average errors or error standard deviations from the 20 independent runs. Then, the average rankings from all 28 objective functions are obtained for each combination of function evaluations and problem dimensions. The FPA instance with the minimum ranking is considered as the optimal FPA instance for the specific combination of function evaluations and problem dimensions.

Figure 10 presents the derived optimal parameters as a function of problem dimensions and function evaluations and Tables 8-10 present the same values for more clarity. Fig. 10a shows that a population of 20 individuals is optimal for small numbers of function evaluations. For the later stages of function evaluations, the results depend on the number of problem dimensions. The larger the number of dimensions the earlier it is required to use a population of 40 individuals. More specifically, for $d = 20$, $n = 40$ becomes optimal after $5000 \cdot d$ function evaluations, for $d = 10$ after $7000 \cdot d$ and for $d = 5$ after $10000 \cdot d$ function evaluations. Regarding the switch probabilities, Fig. 10b presents that $p = 0.2$ is always the optimal value for $d = 5$ and $p = 0.4$ for $d = 20$. For $d = 10$, the results are less clear as it seems that the optimal p value

21

fluctuates between 0.2 and 0.6 depending on the number of function evaluations. With respect to the Lévy flight steps scale factor γ, Fig. 10c shows that $\gamma = 0.1$ is almost always optimal for $d = 5$ and 10. However, the optimal step size increases to $\gamma = 1$ for most function evaluations when $d = 20$. In almost all cases, optimal γ ranges between 0.1 and 1.

It is worth noting here that Figs 10a-c can be used synchronously to fully determine optimal FPA parameter settings. For example, it can be deduced that for an optimization problem with $d = 20$ dimensions where $10000 \cdot d$ function evaluations are afforded, the optimal parameter setting consists of a population size n of 40 individuals, switch probability p of 0.4 and step size γ of 1.
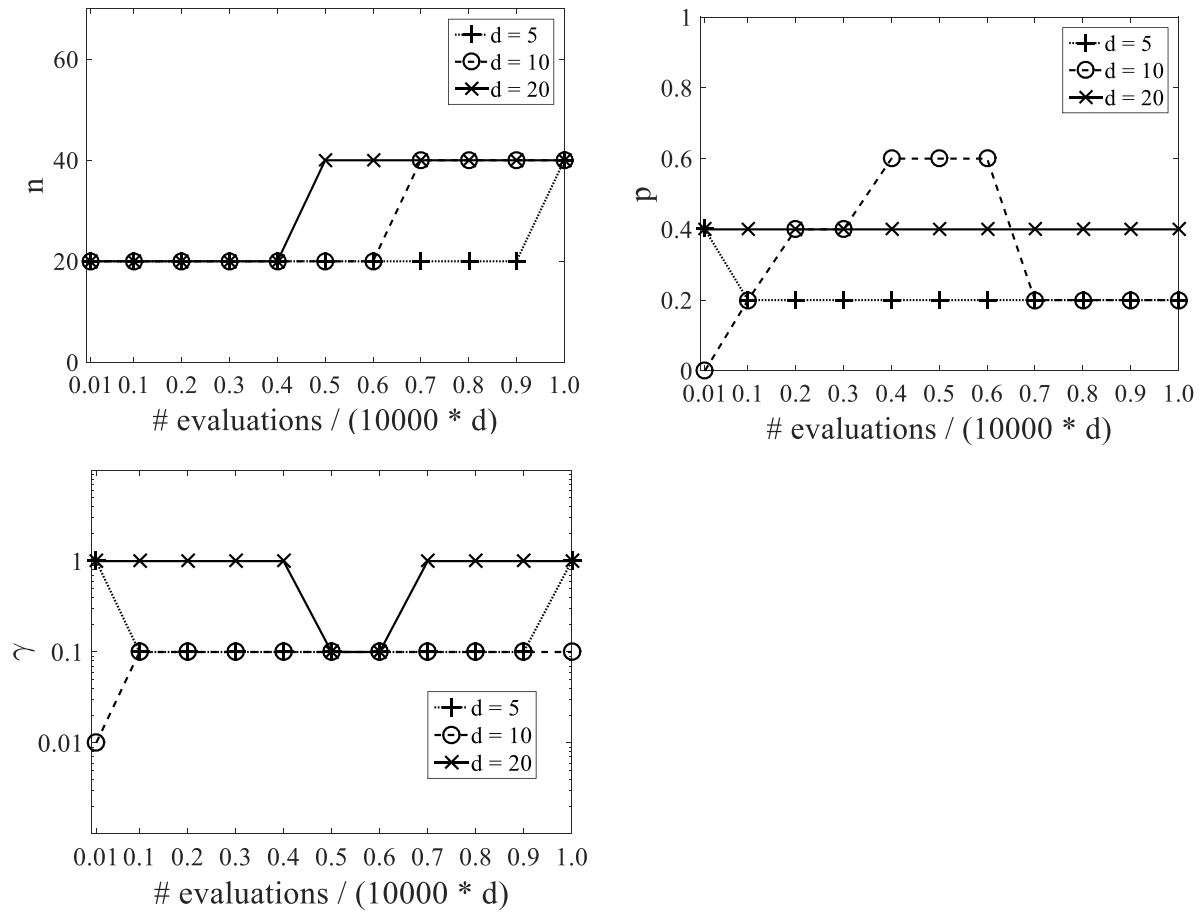
Fig. 10: Optimal parameter values based on minimum mean errors as a function of problem dimensions and function evaluations for different objective functions: a) n; b) p and c) γ

Table 8: Optimal n values based on minimum mean errors for different objective functions

	Number of function evaluations normalized to MaxFES = $10000 \cdot d$										
Dimensions	**0.01**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
$d = 5$	20	20	20	20	20	20	20	20	20	20	40
$d = 10$	20	20	20	20	20	20	20	40	40	40	40
$d = 20$	20	20	20	20	20	40	40	40	40	40	40

Table 9: Optimal *p* values based on minimum mean errors for different objective functions

Dimensions	Number of function evaluations normalized to MaxFES = 10000·*d*										
	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
d = 5	0.4	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
d = 10	0	0.2	0.4	0.4	0.6	0.6	0.6	0.2	0.2	0.2	0.2
d = 20	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4

Table 10: Optimal *γ* values based on minimum mean errors for different objective functions

Dimensions	Number of function evaluations normalized to MaxFES = 10000·*d*										
	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
d = 5	1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	1
d = 10	0.01	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
d = 20	1	1	1	1	1	0.1	0.1	1	1	1	1

5 Conclusions

The Flower Pollination Algorithm (FPA) is a recently developed population-based metaheuristic optimization algorithm imitating the evolution mechanisms of flowering plants. FPA is characterised by simplicity in its formulation as well as high computational performance and it has been found to outperform other well-known optimization algorithms. Previous studies related to FPA assume pre-fixed parameter values based on rather empirical choices and/or experimental comparisons of limited scale and scope.

In this study, a systematic effort has been made to specify appropriate FPA parameters values that maximize its computational performance on a wide range of optimization problems. To serve this goal, a simple non-iterative, single-stage sampling method is employed for parameters tuning oriented towards practical applications of FPA. The tuning method is applied to the set of 28 functions specified in IEEE-CEC'13 for real-parameter single-objective optimization problems for three different numbers of problem dimensions and eleven numbers of function evaluations.

It is found that the FPA parameters that minimize the average errors depend significantly on the objective function, the problem dimensions and function evaluations. More particularly, the optimal population sizes seem to be increasing as the function evaluations and the problem dimensions are increasing. The respective optimal switch probabilities and Lévy flight step scale factors seem to increase with the number of problem dimensions and not to be affected significantly by the number of function evaluations.

In addition to the FPA parameters minimizing average prediction errors, the parameters yielding the most robust FPA predictions for a number of independent runs are also examined herein. It can be seen that the robust parameters can differ significantly from the minimum mean error parameters. Nevertheless, it is found that the mean errors of the robust parameter sets do not vary highly from the minimum mean errors.

At the end of this study, an effort is made to recommend FPA parameter values that are independent of the examined objective functions. These recommendations are based on the average rankings of the different parameter sets for the different objective functions. It is found that the optimal population sizes vary between 20 and 40 individuals depending on the problem dimensions and number of function evaluations. Similarly, the optimal switch probabilities typically vary between 0.2 and 0.4 and the optimal Lévy flight steps scale factors between 0.1 and 1.0 again depending on the problem dimensions and afforded function evaluations. Further studies will focus on the application of the proposed tuning method to test and tune other nature-inspired algorithms for both the CEC test suites and other real-world design benchmarks. Different performance metrics will be investigated and compared, together with other tuning methods, in future studies.

6 Declarations

References

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

[2] 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 Computing 22: 4221-4239

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

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

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

[6] Abdel-Raouf O, Abdel-Baset M *et al*. (2014c) A new hybrid flower pollination algorithm for solving constrained global optimization problems. Int J Appl Oper Res-An Open Access J 4:1–13

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

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

[9] Abualigah L, Yousri D, Abd Elaziz M, Ewees AA, Al-qaness M, Gandomi A (2021) Aquila Optimizer: A novel metaheuristic optimization algorithm. Comput Ind Eng 157:107250

[10] 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 75:5280-5323

[11] Al-Betar MA, Alyasseri ZAA, Awadallah MA, Doush IA (2021) Coronavirus Herd Immunity Optimizer (CHIO). Neural Comput App 33:5011-5042

[12] 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, 91-118

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

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

[15] 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

[16] Eiben AE, Smit SK (2011) Parameter tuning for configuring and analysing evolutionary algorithms. Swarm and Evolutionary Computation 1:19-31.

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

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

[19] Glover BJ (2007) Understanding flowers and flowering: An integrated approach. Oxford University Press, UK

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

[21] 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

[22] 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

[23] 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

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

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

[26] 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

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

[28] 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

[29] 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

[30] Pavlyukevich I (2007) Lévy flights, non-local search and simulated annealing. J Comput Phys 226:1830-1844

[31] 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

[32] 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

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

[34] 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

[35] Walker M (2009) How flowers conquered the world BBC Earth News. http://news.bbc.co.uk/earth/hi/earth_news/newsid_8143000/8143095.stm

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

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

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

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

[40] Yang XS, Deb S, Loomes M, Karamanoglu M (2013) A framework for self-tuning optimization algorithm. Neural Comput Appl 23:2051-2057

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

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

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

[44] 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