



# Flying Squirrel Optimizer (FSO): A Novel SI-Based Optimization Algorithm for Engineering Problems

Golamreza Azizyan<sup>1\*</sup>, Farid Miarnaeimi<sup>1</sup>, Mohsen Rashki<sup>1</sup> and Naser Shabakhty<sup>2</sup>

<sup>1</sup> Department of Civil Engineering, Faculty of Engineering (Shahid Nikbakht), University of Sistan and Baluchestan, Zahedan, Iran

<sup>2</sup> School of Civil Engineering, Iran University of Science and Technology, Tehran, Iran

**Received:** 04 October 2018

**Accepted:** 09 March 2019

## Abstract

This paper provides a novel meta-heuristic optimization algorithm. The behaviors of flying squirrels in the nature are the main inspiration of this research. These behaviors include flying from tree to tree and walking on the ground or on a tree branch to find food. They also contact each other with chirp or squeak. This algorithm is named flying squirrel optimizer (FSO). Two main theories of motion were used for the simulation of flying and walking of the flying squirrels and they are Lévy flight and normal random walk. FSO is also benchmarked on twelve mathematical functions and the answers are compared with MFO, PSO, GSA, BA, FPA, SMS, and FA. The results show that FSO can provide good results when compared with these well-known meta-heuristics approaches. Five classical engineering problems and a real issue in the field of dam engineering were employed to challenge the FSO abilities in solving engineering design problems. The results also show that the proposed FSO algorithm can be used on a wide range of problems with unknown search spaces.

## Keywords:

Flying squirrels  
meta-heuristic  
optimization  
engineering problems  
flying squirrel optimizer (FSO)

\* Correspondence E\_mail: [g.azizyan@eng.usb.ac.ir](mailto:g.azizyan@eng.usb.ac.ir)

**INTRODUCTION**

The description of optimization is to detect a feasible solution with the most affordable performance considering the constraints by maximizing the desired variables and minimizing the undesired ones. Optimization methods are made to supply the ‘best’ values of system design and operating policy variables, which will result in the highest levels of efficiency (Yang and Press, 2010).

Meta-heuristics are usually an efficient approach to use trial and error to create acceptable answers to a complex problem in a reasonably practical time. The complexness in a problem makes it impossible to find every possible solution or combination; the target is to discover good and feasible solutions in an adequate timescale (Mirjalili, Mirjalili and Hatamlou, 2016).

The principal aspects of any meta-heuristic algorithm are intensification and diversification. Intensification means focusing within the search

in a local region exploiting the content that a current good-enough answer is discovered in this region (this can be in combination with selecting the most beneficial solutions), while diversification means generating different solutions, which explore the variables domain on the global scale, Diversification using randomization improves the diversity of the solutions and keeps the solutions from being trapped at local optima. The great combination of these two major components will often ensure that the global best response is accessible (Kaveh and Ghazaan, 2017).

A general systematization of the optimization algorithms is: Evolutionary Algorithms (EAs), physics-based algorithm, and Swarm Intelligence (SI) algorithms. Some of the important algorithms in this category are listed on a timeline for a deeper review as shown in Fig. 1. EAs are shown in the red rectangles. The blue rectangles are physic-based and the SI-based algorithms are green rectangles.

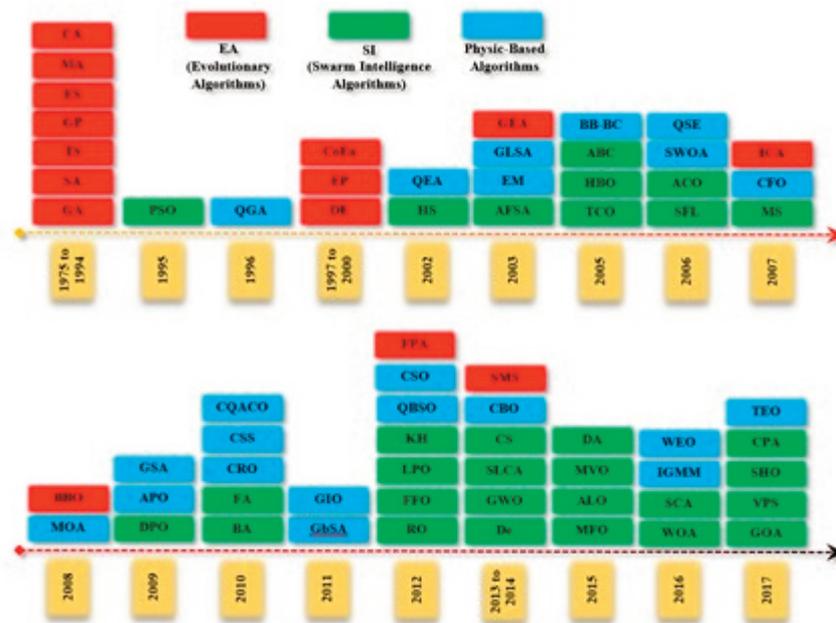


Fig. 1. History timeline of meta-heuristics

Evolutionary algorithms were the primary meta-heuristic algorithms, in 1975 to 1994. Physic-based meta-heuristics started to be highly considered from 1995 to 2011, although the PSO and a number of other SI-based algorithms were conducted in these years. Optimization re-

searchers have made considerable progress on Swarm Intelligence-based algorithms since 2012.

However, the only significant point is the fact that meta-heuristic algorithms have found many applications in numerous aspects of engineering, applied mathematics, medicine, economics, and

also other sciences.

Aside from this, recently various corrections are proposed within the basic versions of existing nature-inspired algorithms for solving complex optimization problems. For example, slow convergence rate of ABC algorithm is modified, developed and improved in the variant IABC (improved ABC) (Nourani et al., 2008; Chong and Zak, 2013) and tested on several reliability-based design optimization (RBDO) problems. Furthermore, the performance of basic Differential Evolution (DE) is improved for large scale optimization problems by embedding an effective switching mechanism for two main control parameters of DE (Liu, Cai and Wang, 2010). The matter of low convergence efficiency of basic cuckoo search algorithm is resolved by integrating chaos mechanism as well as the resulting improved cuckoo search (ICS) is successfully put on to optimization problem of visible light communications (VLC) in smart homes (Yang and Deb, 2009; Gandomi, X. S. Yang and Alavi, 2013).

All nature-inspired algorithms have some general specification like: (i) they mimic some natural phenomenon (ii) they don't need gradient information (iii) they use random variables (iv) and contain various parameters which need to be determined adequately to solve a problem (Chong and Zak, 2013). Each algorithm proposes specific advantages, in term of robustness, performance in the existence of uncertainty and unknown search spaces (Chong and Zak, 2013).

Regardless of the presence of many outstanding optimization algorithms in literature, scientific community continues to developing new optimization methods for solving new and more

advanced optimization problems underneath ideology of continuous improvement to get better design. In addition, according to the “no free lunch” (NFL) theorem, there isn't any single nature-inspired optimization method, which can optimally solve all optimization problems (Chong and Zak, 2013). Consequently, an optimization algorithm is qualified for solving a particular group of problems but ineffective on other class of problems (Chong and Zak, 2013). The NFL theorem, certainly, keeps this domain of research open and allows the researchists to develop the current algorithms or suggest new algorithms for better optimization process. Hereupon, this paper introduces a novel SI-based meta-heuristic algorithm, named Flying Squirrels Optimizer (FSO) and the concepts are inspired with the flying and walking of flying squirrels to find chestnut (food). The proposed algorithm is performed to unconstrained test functions and constrained engineering design issues with discrete and continues variables. Additionally, it can be used for the dam engineering design, to express its application to solve the real-world engineering problems.

The remaining paper is sectioned as follows: Section 2 describes the inspiration behind this work and presents the natural behaviors of flying squirrels. Mathematical formulation of the behaviors of flying squirrels is described in section 3. The experimental setup, results, discussion, and analysis are given in Section 4. This section also evaluates the potency of the proposed FSO algorithm in solving six constrained engineering design problems. Finally, all the conclusions of the paper are presented in section 5.



Fig. 2. Two motion phase of flying squirrels to find food (a) Walking, and (b) Flying

## FLYING SQUIRRELS

Flying squirrels optimizer is a new meta-heuristic swarm-based optimization algorithm and it has been performed to solve mathematic and engineering problems. This section introduces the flying squirrels apparent characteristics and their natural behavior to find food and contact each other.

Flying squirrels (*Glaucomys sabrinus*) range from the tree line in Alaska and Canada southward in the western world to Northern California and Colorado, in the middle of the continent to Central Michigan and Wisconsin, and through the east to northern North Carolina and Tennessee. Flying squirrels are clumsy on the ground, but they are able to glide gracefully from tree to tree. Their motion on the ground is modeled as a random walk with normal distribution steps and the flying phase is simulated based on Lévy flight theory. Flying squirrels emit a soft low chirp, thus, they squeak when distressed. In addition, they prefer scent and touch to communicate with one another. The communication foundation of search agents in FSO is also formed on the basis of the mentioned future of flying squirrels (Holloway and Malcolm, 2007; Weigl, 2007; Wilson, 2010).

There are two advantage of moving search agents in FSO, as stated earlier: random walk and Lévy flight. Mathematical modeling of these motions is explained in the following.

### MATHEMATICAL IMPLEMENTATION

This section contains two subheadings. The theoretical bases of the two main motions, including random walk and Lévy flights, are described in these subsections. The computational complexity of SFO is also described in these subsections.

#### Random walk

A random walk can be a random process which features is taking a number of consecutive random step. Random walks have many applications in physics, economics, statistics, computer sciences, environmental science, and engineering. Consider a scenario where a drunkard walks on a street, at each step, he can randomly go forward or backward, this forms a random walk in one di-

mensional. If this drunkard walks on a football pitch, he can walk in any direction randomly; this becomes a 2D random walk. Mathematically speaking, let  $S_N$  denotes the sum of each consecutive random step  $X_i$ , then  $S_N$  forms a random walk in N dimensions (Rao, 2009):

$$S_N = \sum_{i=1}^N X_i = X_1 + \dots + X_N \quad (1)$$

where  $X_i$  is a random step drawn from a random distribution. This relationship can also be written as a recursive formula:

$$S_N = \sum_{i=1}^{N-1} X_i + X_N = S_{N-1} + X_N \quad (2)$$

which means the next state  $S_N$  will only depend on the current existing state  $S_{N-1}$  and the motion or transition  $X_N$  from the existing state to the next state. Here, the step size or length in a random walk can be fixed or varied. In addition, there is no reason why each step length should be fixed. In fact, the step size can also vary according to a known distribution. If the step length obeys the Gaussian distribution, the random walk becomes the Brownian motion. In this theory, as the number of steps N increases, the central limit theorem implies that the random walk should approach a Gaussian (Normal) distribution. The probability density of the normal distribution is shown in Eq. (3) (Rao, 2009).

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3)$$

where  $\mu$  is the mean or expectation of the distribution (and also its median and mode) that shows the position of flying squirrels in the current iteration.  $\sigma$  is the standard deviation and  $\sigma^2$  is the variance. Fig. 3 shows two examples of random walk in 2 and 3 dimensions.

A semi sigmoid function can also be applied in the standard deviation of walking steps. It is named Sigma Reduction Factor (SRF), which helps to increase the search accuracy in random walk phase. Sigmoid function is a mathematical function having a characteristic S-shaped curve or sigmoid curve. It is defined by the formula in Eq. 4 (Mirjalili and Lewis, 2013; Kaveh, 2014; Mirjalili, Mirjalili and Yang, 2014).

$$f(x) = \frac{1}{1 + e^{-x}}$$

(4) Often, sigmoid function refers to the special case of the logistic function as shown in Fig. 4.

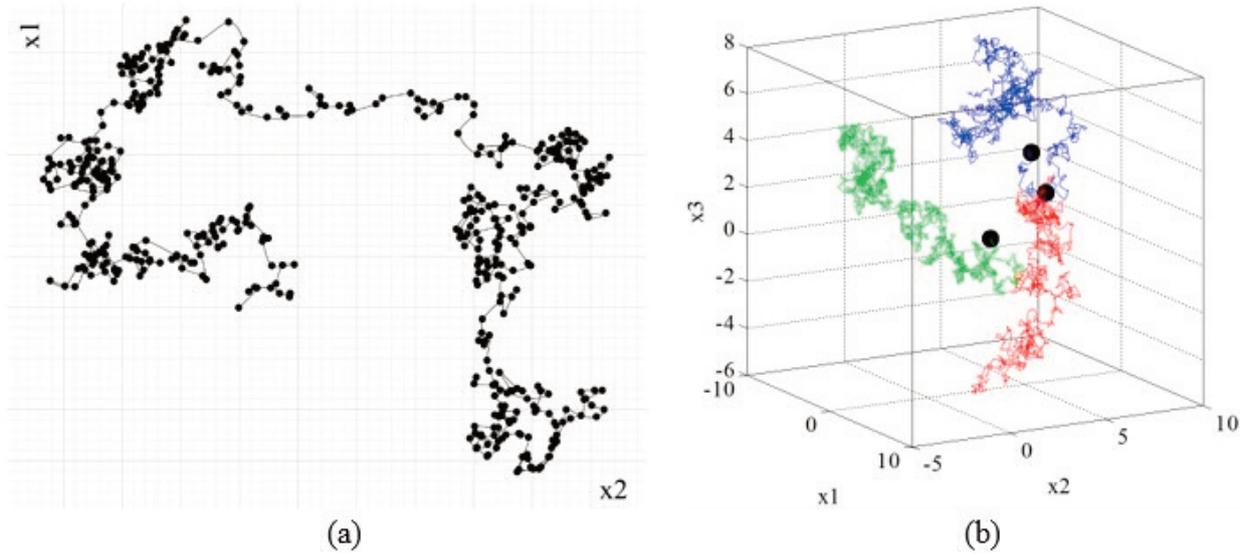


Fig. 3. Random Walk: a) a Flying Squirrel motion in 2D and 500 steps b) three Flying Squirrels (black and bold circles) in 3D and 1000 steps

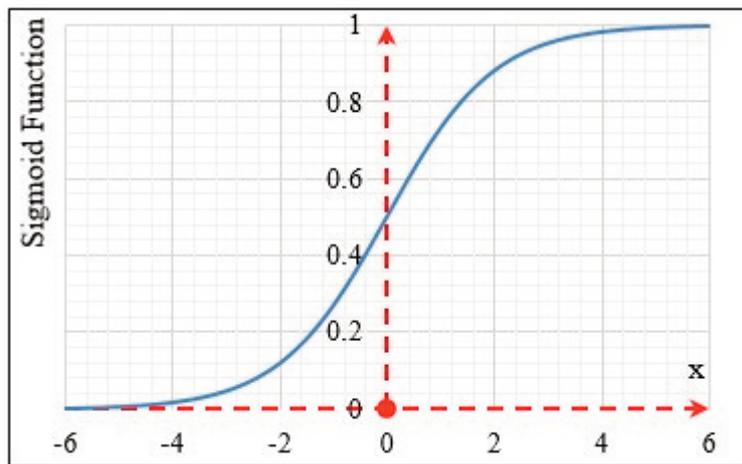


Fig. 4. S-shaped sigmoid function

SRF is a semi sigmoid function as stated earlier. It is considered as a factor of standard deviations of movement steps. It causes a decrease in the standard deviation of flying squirrels motion per each iteration. The inverse of sigmoid function was used to achieve this goal. Inverted sigmoid function is written as follows.

$$SRF_{it} = \sqrt{\left(-\ln\left(1 - \frac{1}{\sqrt{it}}\right)\right)^3} \tag{5}$$

SRF per iteration and the effects of SRF on the steps size are as shown in Fig. 5.

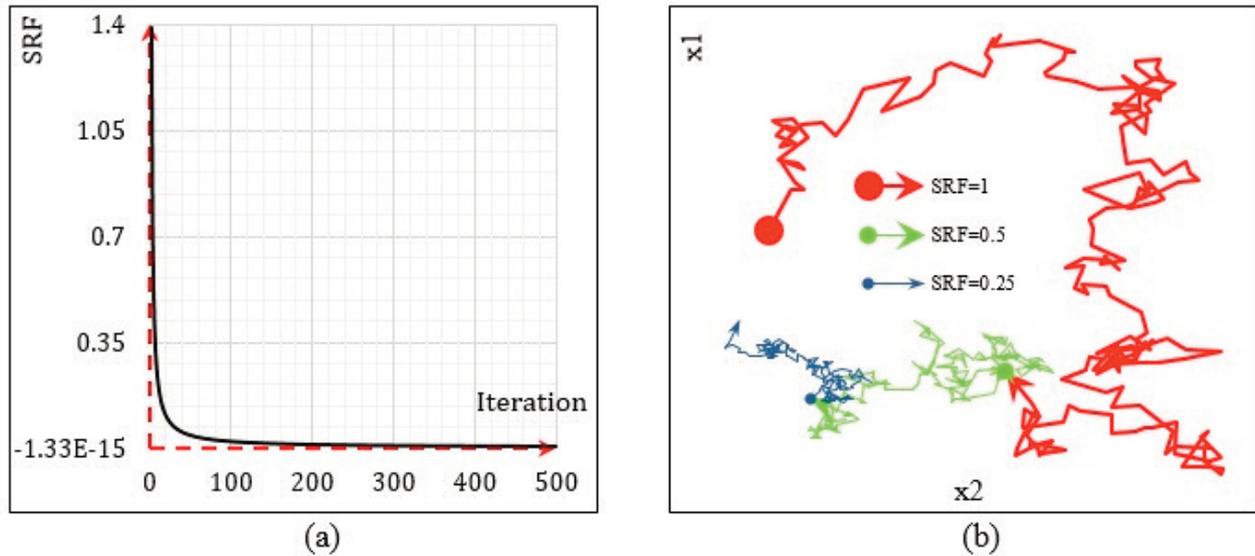


Fig. 5. (a) SFR per iteration and (b) the effects of SRF on the Brownian motion in 2D with 200 steps

**Lévy flight**

Lévy flights were observed among foraging patterns of albatrosses and fruit flies, and spider monkeys. Even when humans like the Ju/'hoansi hunter-gatherers can trace paths of Lévy-flight patterns. In addition, Lévy flights have many uses. Many physical phenomena such as the diffusion of fluorescent molecules, cooling behavior, and noise could show Lévy-flight characteristics under the right conditions. So, this can be applied on the flying phase of flying squirrels motion.

Mathematically speaking, a simple version of Lévy distribution can be defined as (Yang and Deb, 2009; Yang and Press, 2010):

$$L(s, \gamma, \mu) = \begin{cases} \sqrt{\frac{\gamma}{2\pi}} \exp\left[-\frac{\gamma}{2(s-\mu)}\right] \frac{1}{(s-\mu)^{\frac{3}{2}}} & 0 < \mu < s < \infty \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where  $\mu > 0$  is a minimum step and  $\gamma$  is a scale parameter. Clearly, as  $s \rightarrow \infty$ , we have:

$$L(s, \gamma, \mu) \approx \sqrt{\frac{\gamma}{2\pi}} \tag{7}$$

This is a special case of generalized Lévy distribution. In general, Lévy distribution should be defined in terms of Fourier transform as follows:

$$F(k) = \exp[-\alpha|k|^\beta], \quad 0 < \beta \leq 2 \tag{8}$$

where  $\alpha$  is a scale parameter. The inverse of this integral is not easy, since it does not have analytical form, it is not expected for a few special cases. In the case where  $\beta = 2$ , we have:

$$F(k) = \exp[-\alpha k^2] \tag{9}$$

whose inverse Fourier transform corresponds to a Gaussian distribution. Another special case is  $\beta = 1$ , and we have:

$$(k) = \exp[-\alpha|k|] \tag{10}$$

which correspond to a Cauchy distribution:

$$p(x, \gamma, \mu) = \frac{1}{\pi} \frac{\gamma}{\gamma^2 + (x-\mu)^2} \tag{11}$$

where  $\mu$  is the location parameter, while  $\gamma$  controls the scale of this distribution. For the general case, the inverse integral:

$$L(s) = \frac{1}{\pi} \int_0^\infty \cos(ks) \exp[-\alpha|k|^\beta] dk \tag{12}$$

can be estimated only when  $s$  is large. We have:

$$L(s) \rightarrow \frac{\alpha\beta\Gamma(\beta)\sin(\pi\beta/2)}{\pi|s|^{1+\beta}}, \quad s \rightarrow \infty \quad (13)$$

Here,  $\Gamma(z)$  is the Gamma function:

$$\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt \quad (14)$$

In the case when  $z=n$  is an integer, we have  $\Gamma(n)=(n-1)!$ .

Lévy flights are more efficient than Brownian random walks in exploring unknown large-scale search space. There are many reasons to explain

this efficiency and one of them is due to the fact that the variance of Lévy flights:

$$\sigma^2(t) \sim t^{3-\beta}, \quad 1 \leq \beta \leq 2 \quad (15)$$

increases much faster than the linear equation (that is,  $\sigma^2(t) \sim t$ ) of Brownian random walks. Fig. 6 shows the path of Lévy flights of 1000 steps starting from (0, 0) with  $\beta=1.5$ . It is good to point out that a power-law distribution is often linked to some scale-free characteristics and Lévy flights can thus show self-similarity and fractal behavior in the flight patterns.

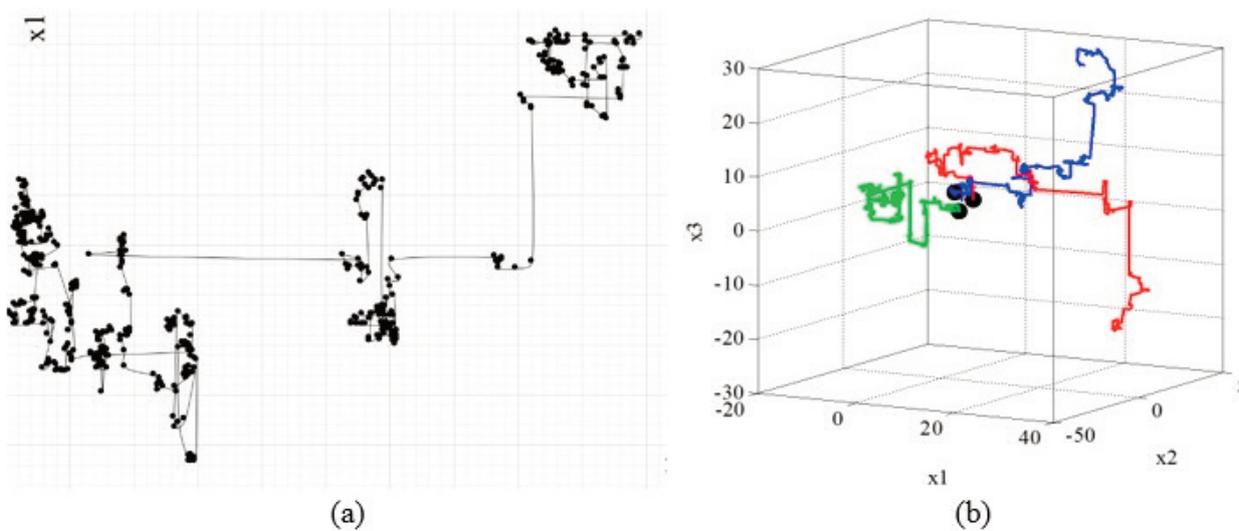


Fig. 6. Lévy flight (a) 2D space with 500 steps and (b) three Flying Squirrels (black and bold circles) in 3D and 1000 steps

From the implementation point of view, the generation of random numbers with Lévy flights involves two steps: the choice of a random direction and the generation of steps which obey the chosen Lévy distribution. The generation of a direction should be drawn from a uniform distribution, while the generation of steps is quite tricky. There are a few ways of achieving this, but one of the most efficient and yet straightforward ways is to use the so-called Mantegna algorithm for a symmetric Lévy stable distribution. Here, 'symmetric' means that the steps can be positive

and negative. A random variable  $U$  and its probability distribution can be called stable if a linear combination of its two identical copies (or  $U_1$  and  $U_2$ ) obeys the same distribution. That is,  $aU_2+bU_2$  has the same distribution as  $cU_2+d$ , where  $a,b>0$  and  $c,d \in \mathbb{R}$ . If  $d=0$ , it is called strictly stable. The step lengths can be calculated by:

$$s = \frac{u}{|v|^{1/\beta}} \quad (16)$$

where  $u$  and  $v$  are drawn from normal distribu-

tion. That is:

$$u \sim N(0, \sigma_u^2), v \sim N(0, \sigma_v^2) \tag{17}$$

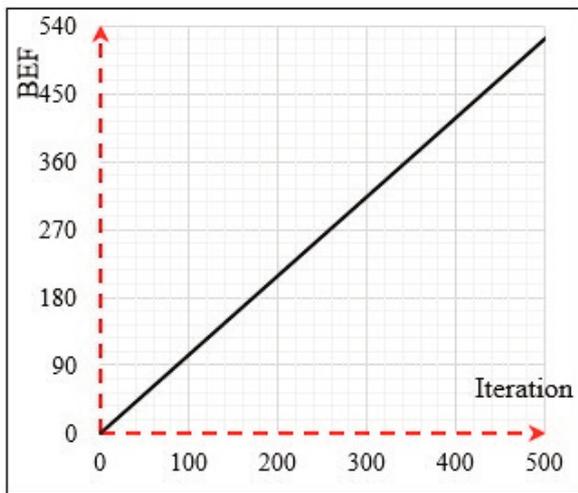
Where

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta) \sin(\pi\beta/2)}{\Gamma[(1+\beta)/2] 2^{\beta} 2^{(\beta-1)/2}} \right\}^{1/\beta}, \sigma_v = 1 \tag{18}$$

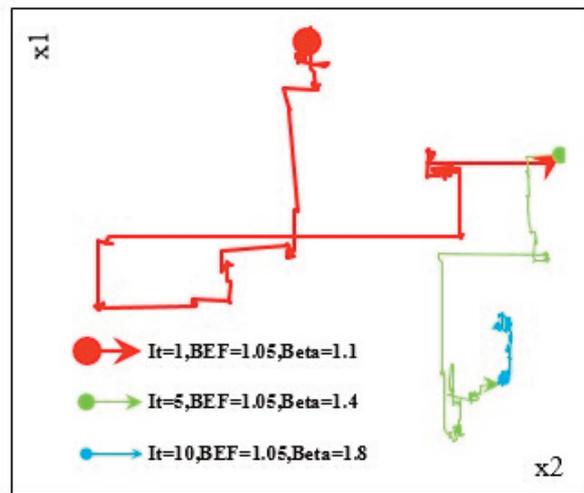
This distribution (for s) obeys the expected Lévy distribution for  $|s| \geq |s_0|$  where  $s_0$  is the

smallest step. In principle,  $|s_0| \gg 0$ , but in reality, it can be taken as a sensible value such as  $s_0=0.1$  to 1. Studies have shown that Lévy flights can maximize the efficiency of resource searches in uncertain environments.

A linear function is proposed to increase  $\beta$  in each iteration. It is called Beta Expansion Factor (BEF). This causes an increase in  $\beta$  of the Lévy flight of flying squirrels and an increase in steps length of the Lévy flight phase. It is recommended that a factor of 1.05 should be used for BEF to obtain sustainable results.



(a)



(b)

Fig. 7. (a) BEF per iteration and (b) the effects of BEF on the Lévy flight in 2D with 200 steps.

Broadly, all nature-inspired metaheuristics imitate two distinct features of nature i.e. adaptability and choice of the fittest, which gives them a similar appearance superficially. Most of the algorithms utilize the concept of motion pattern, which is constructed through randomly generated solutions of optimization problem under consideration (Yang, 2010b). Metaheuristic algorithms are generally differentiated on the basis of their solution updating strategy. The length and direction of each squirrel's step is recursively updated at each iteration through a suitable updating mechanism. This mechanism injects new attributes or patterns in the algorithm while maintaining diversity in solutions.

Consequently, the random walk searches more accurately and Lévy flight has more randomiza-

tion in its steps. Therefore, half of the search operators, which present better results, uses random walk to the optima. This makes it possible to increase the exploration trait of the FSO algorithm near the optimal location. Others are moved with Lévy flight in each iteration. It causes a more random movement in zones with low probability of optimal point. Therefore, a good balance is achieved between exploration and exploitation terms.

### CASE STUDY AND DISCUSSION

This section presents the results of FSO. Metaheuristic algorithms have a random nature to find optimum, so it is necessary to prove the absence of luck in finding the optimal solution. Obtained responses are carefully checked and compared

with the results of existing algorithms. Firstly, twelve well-known test functions are tested in a space with 100 dimensions and the results are compared with the MFO (Mirjalili, 2015a), PSO (Cagnina, Esquivel and Coello, 2008), GSA (Mirjalili and Lewis, 2014), BA (Yang, 2010a), FPA (Yang, 2012), SMS (Cuevas, Echavarría and Ramírez-Ortegón, 2014), and FA (Fausto et al., 2017) algorithms, to verify the results. FSO can also be applied to these semi-real problems: tension-compression spring design, welded beam design, pressure vessel design, multiple disk clutch brake design, and gear train design. More-

over, this section provides one case of real application of the proposed method in the field of concrete gravity dam engineering, including SHAFAROUH dam design optimization.

**Case study of unconstrained problems**

Unimodal test functions are ideal means for featuring the algorithm’s power throughout the exploration. Multimodal benchmark functions are also an effective criterion for the exploitation. Functions that are used for the test are shown in Tables 1 and 2.

Table 1: Unimodal benchmark functions.

Function	Dim	Range	f <sub>min</sub>
$f_1(x) = \sum_{i=1}^n x_i^2$	100	[-100,100]	0
$f_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	100	[-10,10]	0
$f_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	100	[-100,100]	0
$f_4(x) = \max( x_i )$	100	[-100,100]	0
$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	100	[-30,30]	0
$f_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	100	[-100,100]	0
$f_7(x) = \sum_{i=1}^n i x_i^4 + rand[0,1]$	100	[-1.28,1.28]	0

Table 2: Multimodal benchmark functions.

Function	Dim	Range	f <sub>min</sub>
$f_8(x) = -\frac{1 + \cos\left(12\sqrt{\sum_{i=1}^n x_i^2}\right)}{2 + 0.5\sum_{i=1}^n (x_i^2)}$	100	[-100,100]	-1
$f_9(x) = \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10)$	100	[-5.12,5.12]	0
$f_{10}(x) = -20 \exp\left(-0.2\sqrt{\frac{\sum x_i^2}{2}}\right) - \exp\left(\frac{\sum \cos(2\pi x_i)}{2}\right) + 20 + e$	100	[-32,32]	0
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	100	[-600,600]	0
$f_{12}(x) = \frac{\pi}{n} \{10\sin(\pi y_1) + \sum_{i=1}^{n-i} (y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1})] (y_n - 1)^2\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$	100	[-50,50]	0
$y_i = 1 + \frac{x_i + 1}{4}$ , $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$			

These functions are assessed in 100 dimensions and the obtained results are compared with the responses of MFO, PSO, GSA, BA, FPA, SMS, and FA algorithms, as earlier stated. The search agents and number of iteration loops for all of these algorithms are 50 and 1000, respectively, to conduct a fair comparison.

Tables 1 and 2 provide the test functions, where “Dim” is the states dimension of the function,

“Range” is the boundary of the function’s search space, and  $f_{min}$  is the optimum.

Generally speaking, the benchmark functions were employed to evaluate the ability of algorithms to minimization optimization. All the algorithms were run 30 times on each test functions, independently. The statistical results (Mean and Standard deviation, Std) are shown in Tables 3 and 5.

Table 3: Results of unimodal benchmark functions.

<i>f</i>	FSO		MFO		PSO		GSA	
	<i>Mean</i>	<i>Std</i>	<i>Mean</i>	<i>Std</i>	<i>ave</i>	<i>Std</i>	<i>ave</i>	<i>std</i>
<i>f1</i>	4.37523E-39	3.97999E-40	0.000117	0.00015	1.321152	1.153887	608.2328	464.6545
<i>f2</i>	3.45283E-20	2.88152E-20	0.000639	0.000877	7.715564	4.132128	22.75268	3.365135
<i>f3</i>	2.06989E-38	1.07899E-39	696.7309	188.5279	736.3931	361.7818	135760.8	48652.63
<i>f4</i>	1.68764E-20	8.51808E-22	70.68646	5.275051	12.97281	2.634432	78.78198	2.814108
<i>f5</i>	98.84350505	0.80333959	139.1487	120.2607	77360.83	51156.15	741.003	781.2393
<i>f6</i>	20.46836981	0.383942567	0.000113	9.87E-05	286.6518	107.0796	3080.964	898.6345
<i>f7</i>	7.71527E-05	9.82268E-05	0.091155	0.04642	1.037316	0.310315	0.112975	0.037607

<i>f</i>	BA		FPA		SMS		FA	
	<i>Mean</i>	<i>Std</i>	<i>Mean</i>	<i>Std</i>	<i>ave</i>	<i>std</i>	<i>ave</i>	<i>std</i>
<i>f1</i>	20792.44	5892.402	203.6389	78.39843	120	0	7480.746	894.8491
<i>f2</i>	89.78561	41.95771	11.1687	2.919591	0.020531	0.004718	39.32533	2.465865
<i>f3</i>	62481.35	29769.17	237.5681	136.6463	37820	0	17357.32	1740.111
<i>f4</i>	49.74324	10.14363	12.57284	2.29	69.17001	3.876667	33.95356	1.86966
<i>f5</i>	1995125	1252388	10974.95	12057.29	6382246	729967	3795009	759030.3
<i>f6</i>	17053.41	4917.567	175.3808	63.45257	41439.39	3295.23	7828.726	975.2106
<i>f7</i>	6.045055	3.045277	0.135944	0.061212	0.04952	0.024015	1.906313	0.460056

Table 4: Results of Wilcoxon rank sum test for uni-modal functions ( $P \geq 0.05$  have been bolded).

<i>f</i>	FSO	MFO	PSO	GSA	BA	FPA	SMS	FA
<i>f1</i>	N/A	1.83E-04	3.46E-02	3.46E-02	3.46E-02	3.46E-02	6.39E-05	3.22E-05
<i>f2</i>	N/A	1.83E-04	3.46E-02	3.46E-02	3.46E-02	1.83E-04	1.83E-04	<b>2.76E-01</b>
<i>f3</i>	N/A	5.83E-04	3.46E-02	3.46E-02	3.46E-02	3.46E-02	6.39E-05	4.85E-05
<i>f4</i>	N/A	1.83E-04	3.46E-02	3.46E-02	3.46E-02	3.46E-02	1.83E-04	7.28E-03
<i>f5</i>	N/A	1.83E-04	3.46E-02	<b>5.10E-01</b>	3.46E-02	1.83E-04	1.83E-04	<b>1.48E-01</b>
<i>f6</i>	3.46E-21	N/A	3.46E-02	3.46E-02	3.46E-02	1.83E-04	1.83E-04	8.90E-04
<i>f7</i>	N/A	1.83E-04	3.46E-02	4.31E-02	3.46E-02	1.83E-04	1.83E-04	4.15E-02

Table 5: Results of multimodal benchmark functions.

<i>f</i>	FSO		MFO		PSO		GSA	
	<i>Mean</i>	<i>Std</i>	<i>Mean</i>	<i>Std</i>	<i>ave</i>	<i>std</i>	<i>ave</i>	<i>std</i>
<i>f8</i>	-1	0.00	-3.22E-03	2.203E-03	-1.98E-03	3.420E-04	-1.62E-03	6.543E-04
<i>f9</i>	0.00	0.00	84.60009	16.16658	124.2973	14.25096	31.00014	13.66054
<i>f10</i>	8.8818E-16	0.00	1.260383	0.72956	9.167938	1.568982	3.740988	0.171265
<i>f11</i>	0.00	0.00	0.01908	0.021732	12.41865	4.165835	0.486826	0.049785
<i>f12</i>	2.085E-01	7.08E-03	0.894006	0.88127	13.87378	5.85373	0.46344	0.137598

<i>f</i>	BA		FPA		SMS		FA	
	<i>Mean</i>	<i>Std</i>	<i>Mean</i>	<i>Std</i>	<i>ave</i>	<i>std</i>	<i>ave</i>	<i>std</i>
<i>f8</i>	-2.225E-03	1.1104E-03	-2.20E-03	4.525E-04	-2.01E-03	7.08E-04	-7.56E-04	2.58E-04
<i>f9</i>	96.21527	19.58755	92.69172	14.22398	152.8442	18.55352	214.8951	17.21912
<i>f10</i>	15.94609	0.774952	6.844839	1.249984	19.13259	0.238525	14.56769	0.467512
<i>f11</i>	220.2812	54.70668	2.716079	0.727717	420.5251	25.25612	69.65755	12.11393
<i>f12</i>	28934354	2178683	4.105339	1.043492	8742814	1405679	368400.8	172132.9

Table 6: Results of Wilcoxon rank sum test for multi-modal functions ( $p \geq 0.05$  have been bolded).

<i>f</i>	FSO	MFO	PSO	GSA	BA	FPA	SMS	FA
<i>f8</i>	N/A	1.83E-04	3.46E-02	3.46E-02	N/A	<b>0.161972</b>	0.000183	<b>5.17E-01</b>
<i>f9</i>	N/A	1.83E-04	3.46E-02	<b>1.72E-01</b>	3.46E-02	0.000181	0.000181	3.74E-03
<i>f10</i>	N/A	1.83E-04	3.46E-02	3.46E-02	3.46E-02	0.000183	0.000183	7.69E-04
<i>f11</i>	N/A	1.83E-04	3.46E-02	3.46E-02	3.46E-02	0.000183	0.000183	9.32E-04
<i>f12</i>	N/A	<b>0.472676</b>	3.46E-02	3.46E-02	3.46E-02	0.000183	0.000182	<b>6.72E-01</b>

From Table 3, FSO algorithm gives appropriate results when compared with other algorithms. It is necessary to state that the unimodal functions are proper for benchmarking exploitation phase. Therefore, these results show the absolute best performance of FSO when exploiting the optimum. This is due to the good combination of random walk and Lévy flights of search agents.

On the other hand, multimodal benchmark functions have a lot of local optima; therefore, they are suitable for benchmarking the exploration capability of an algorithm. Also, this study shows that FSO is also able to provide proper results for the multimodal benchmark functions, according to the results shown in Table 5.

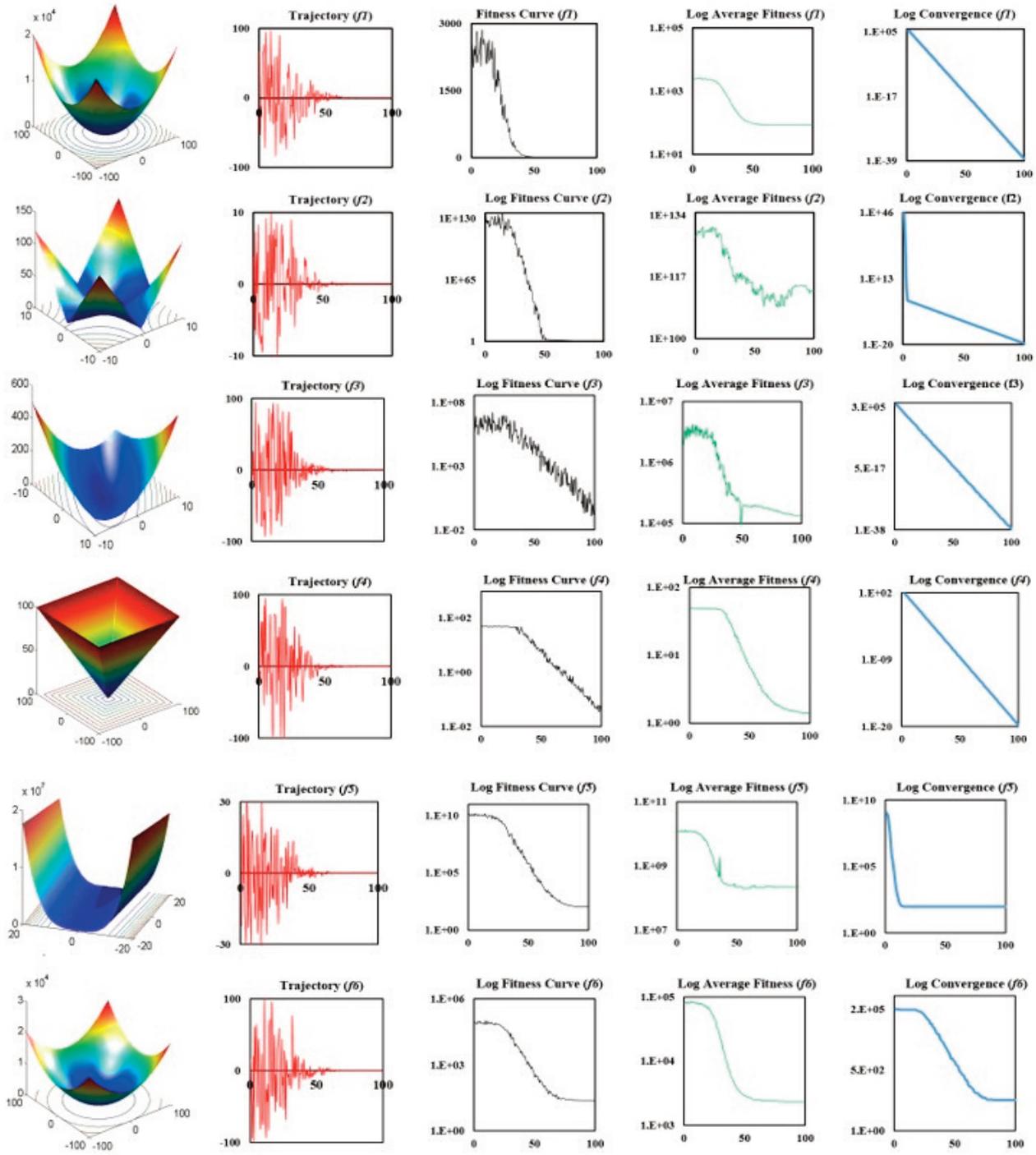
Wilcoxon test is also applied at 5% significance level and the p-values of the tests are shown in

Tables 4 and 6, to compare each of the run's results and decide on the significance of the results for each run. The best obtained results in each test function are chosen and compared with the results of other algorithms independently. Not Applicable (N/A) has been written for the best algorithm for each function, because the best algorithm cannot be compared with itself. The p-values are mostly N/A for FSO, which shows the superiority of this algorithm is statistically significant, except for the F6 function. So, the FSO algorithm has the potential to solve problems that cannot be solved efficiently by other algorithms, according to the no free lunch (NFL) theorem.

The convergence behavior of the FSO algorithm is as shown in Fig. 8. There should be sudden variations in the motion of particles over the

primary iterations, according to Van Den Bergh and Engelbrecht. This helps a meta-heuristic to explore the search space widely. These changes should be decreased to underscore exploitation at the end of optimization. The trajectory and the fitness (some cases are in the logarithmic scale) of the first flying squirrel are shown in the first and second columns of Fig. 8. It is obvious that

there are sudden variations in the initial steps of iterations which are reduced over the iterations. The third column of Fig. 8 shows the average fitness of all of the search agents. In addition, the fourth column of Fig. 8 shows the convergence of the FSO algorithm to the optimum. These results show that the FSO algorithm has significant merit in terms of exploration and exploitation.



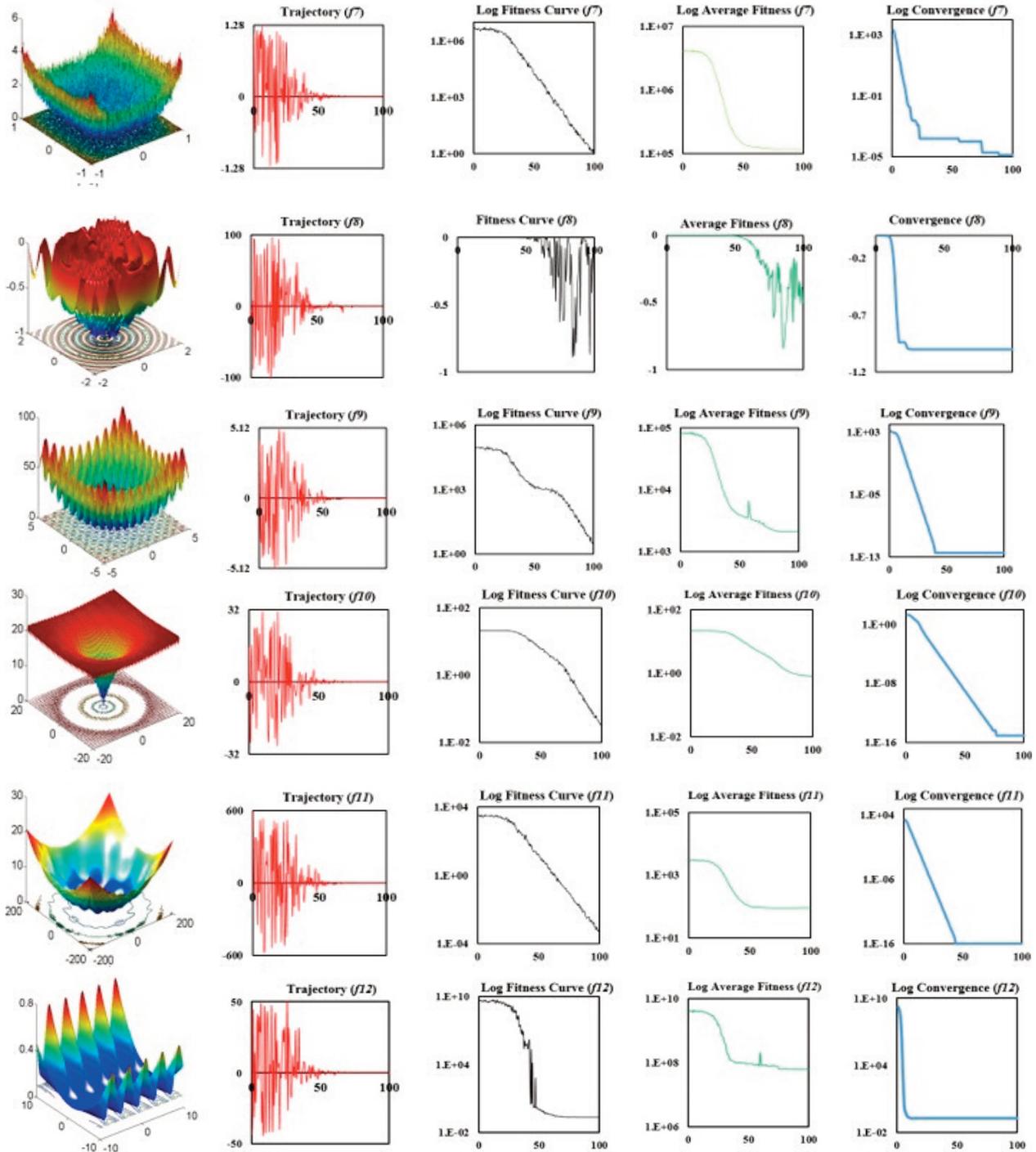


Fig. 8. Trajectory in first dimension, fitness and average fitness of all flying squirrels, and convergence rate. The performance of FSO, facing the benchmark functions, is verified by the presented results. Five classical engineering design problems and a real problem in dam engineering are used subsequently.

Case study of classical engineering problems

Here, the usability of FSO in dealing with five constrained classical engineering design problems and a concrete gravity dam design problem is investigated. A comprehensive comparison of

results with those available in the literature will be presented. The provided benchmarks problems include objective functions and constraints with various types and nature (quadratic, cubic, polynomial, and nonlinear) and several numbers

of design variables (continuous, discrete, and mixed). The mathematical formulations of the test problems are also provided. The obtained optimization results have been compared with other well-known optimization algorithms. Also, results were compared in terms of statistical results and NFEs. In this research, the computational cost which is considered as the best NFEs corresponding to the obtained best answer, is calculated using the product of the number of flying squirrels and the maximum number of iterations (that is,  $NFEs_{FSO} = n_{Squirrels} \times It_{max}$ ). The proposed algorithm was coded in MATLAB programming software and the simulations and numerical solutions were run on an Intel(R) Core(TM) i7-4500U CPU @ 1.80 GHz with 4 GB Random Access Memory (RAM).

The number of the flying squirrels is considered as 50 and the maximum number of iterations ( $It_{max}$ ) depends on the complexity of the optimization problems. The penalty function method is also adopted in FSO to consider constraints violation. This method is easy to implementation and has a simple principle, especially for continuous constrained problems. Typically, a constrained optimization problem is identified as follows:

$$\begin{aligned} & \text{minimize } f(x) \\ & \text{Subject to: } g_k(x) \leq 0, k=1, \dots, m \end{aligned} \quad (19)$$

where  $f(x)$  is the objective function and  $g_k(x)$  is the  $k^{th}$  inequality constraint. Integration of penalty functions into the objective function will transform the mentioned constrained problem to an unconstrained one. The penalized objective function  $f_p$  is then written as follows:

$$f_p(x) = f(x) + \lambda \sum_{k=1}^m \delta_k [g_k(x)]^2 \quad (20)$$

where  $\lambda > 0$  (e.g.  $\lambda = 10^5$ ) is a penalty factor and

$$\begin{cases} \delta_k = 1 & \text{if constraint } g_k \text{ is violated} \\ \delta_k = 0 & \text{if constraint } g_k \text{ is satisfied} \end{cases}$$

(Rao, 2009).

### Tension-compression spring design

The aim is to minimize the volume of a coil spring under a constant tension/compression load. The design variables are the wire diameter ( $x_1$ ), the winding diameter ( $x_2$ ) and the number of active coils of the spring ( $x_3$ ) specified in Fig. 9. The constraints of this problem include: shear stress, surge frequency, and minimum deflection.

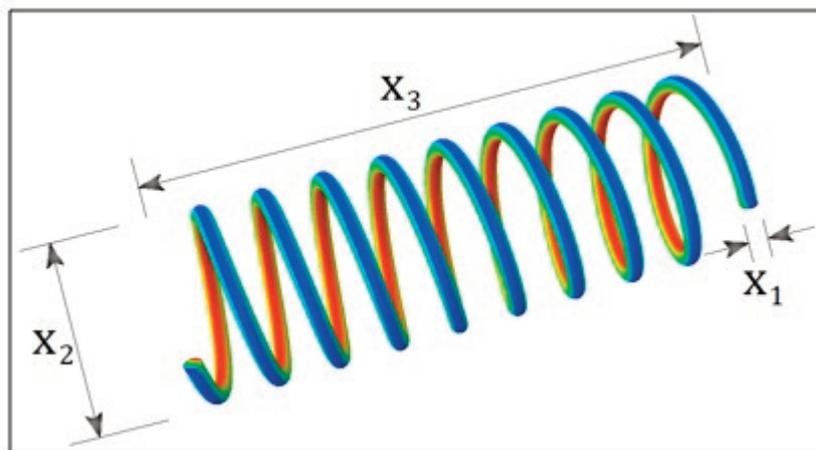


Fig. 9. Tension-compression spring design problem

Current issue was discussed repeatedly using precise mathematical methods and also meta-heuristic optimization algorithms. Wang and Li (Wang and Li, 2010) paid to solve this problem using PSO algorithm. The other algorithms such

as Evolutionary Strategy, GA, Search Harmony and Differential Evolution were employed to solve this problem. The objective function and the mechanical constraints are shown in Eq. (21).

$$\begin{aligned}
 \text{Minimize:} & & \text{Subject to:} & & \text{Variable range:} \\
 f(\vec{x}) = (x_3 + 2)x_2x_1^2 & & g_1(\vec{x}) = 1 - \frac{x_2^2x_3}{71785x_1^4} \leq 0 & & 0,05 \leq x_1 \leq 2,00 \\
 & & g_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0 & & 0,25 \leq x_2 \leq 1,30 \\
 & & g_3(\vec{x}) = 1 - \frac{140,45x_1}{x_2^2x_3} \leq 0 & & 2,00 \leq x_3 \leq 15,0 \\
 & & g_4(\vec{x}) = \frac{x_1 + x_2}{1,5} - 1 \leq 0 & & 
 \end{aligned} \tag{21}$$

The optimum locations founded by different algorithms and also the values of their constraints and costs are shown in Table 7.

the design variables corresponding to the best solutions which are all feasible.

Table 8 shows the best and statistical values for

Fig. 10 shows an impressive decrease in the cost of tension/compression spring design.

Table 7: Best results given by well-known optimization algorithm and FSO for the tension-compression spring design problem.

Parameter	DEDS	HEAA	NM-PSO	DELC	WCA	LCA	MBA	APSO	GWO	IGMM	FSO
x1	0.051689	0.051689	0.05162	0.051689	0.05168	0.051689	0.051656	0.052588	0.05169	0.051718	<b>0.051685996</b>
x2	0.356717	0.356729	0.355498	0.356717	0.356522	0.356718	0.35594	0.378343	0.356737	0.357415	<b>0.356644</b>
x3	11.288965	11.288293	11.333272	11.288965	11.30041	11.28896	11.344665	10.13886	11.28885	11.2482	<b>11.2933</b>
g1(x)	1.45E-09	3.96E-10	1.01E-03	-3.40E-09	-1.65E-13	NA	0	-1.55E-04	-1.00E+04	-7.87E-08	<b>-8.47E-7</b>
g2(x)	-1.19E-09	-3.59E-10	9.94E-04	2.44E-09	-7.9E-14	NA	0	-8.33E-04	-1.34E-01	-3.77E-08	<b>-1.45E-8</b>
g3(x)	-4.053785	-4.053808	-4.061859	-4.053785	-4.053399	NA	-4.052248	-4.08917	-4.0533835	-4.05516	<b>-4.053635</b>
g4(x)	-0.727728	-0.72772	-0.728588	-0.727728	-0.727864	NA	-0.728268	-1.06907	-0.7277153	-1.09087	<b>-0.7277800</b>
f(x)	0.012665	0.012665	0.01263	0.012665	0.012665	0.0126652	0.012665	0.0127	0.012666	0.01266525	<b>0.01266524</b>

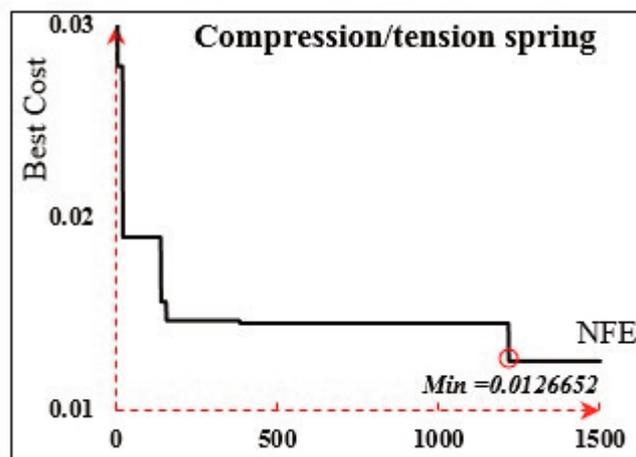


Fig. 10. Convergence curve of tension-compression spring design problem

Table 8: Statistical comparison of results for the tension-compression spring design problem.

Method	Worst	Mean	Best	Std	NFEs
GA1	0.012822	0.012769	0.012704	3.94E-05	900,000
GA2	0.012973	0.012742	0.012681	5.90E-05	80,000
CAEP	0.015116	0.013568	0.012721	8.42E -04	50,020
CPSO	0.012924	0.01273	0.012674	5.20E-04	240,000
HPSO	0.012719	0.012707	0.012665	1.58E -05	81,000
NM-PSO	0.012633	0.012631	0.01263	8.47E-07	80,000
G-QPSO	0.017759	0.013524	0.012665	0.001268	2000
QPSO	0.018127	0.013854	0.012669	0.001341	2000
PSO	0.071802	0.019555	0.012857	0.011662	2000
DE	0.01279	0.012703	0.01267	2.7E -05	204,800
DELC	0.012665	0.012665	0.012665	1.3E-07	20,000
DEDS	0.012738	0.012669	0.012665	1.3E-05	24,000
HEAA	0.012665	0.012665	0.012665	1.4E-09	24,000
PSO-DE	0.012665	0.012665	0.012665	1.2E-08	24,950
SC	0.016717	0.012922	0.012669	5.9E -04	25,167
( $\mu + \lambda$ )-ES	NA	0.013165	0.012689	3.9E-04	30,000
ABC	NA	0.012709	0.012665	1.28E-02	30,000
LCA	0.01266667	0.0126654	0.01266523	3.88E-07	15,000
WCA	0.012952	0.012746	0.012665	8.06E-05	11,750
MBA	0.0129	0.012713	0.012665	6.30E-05	7650
APSO	0.014937	0.013297	0.0127	6.85E-04	120,000
IGMM	0.0135125	0.0128657	0.0126653	2.56E-04	4000
<b>FSO</b>	<b>0.012675347</b>	<b>0.012680628</b>	<b>0.01266524</b>	<b>7.89E-04</b>	<b>1500</b>

This degradation shows the high ability of FSO in dealing with constrained engineering problems. The obtained results cannot be found in any of prior optimization algorithms.

### Welded beam design

The aim of this problem design is to minimize the construction cost of the beam and the con-

straints include: Shear stress ( $s$ ), Bending stress in the beam ( $h$ ), Buckling load on the bar ( $P_c$ ), End deflection of the beam ( $d$ ), and Side constraints. This problem has four variables: thickness of weld ( $x_1$ ), length of attached bar ( $x_2$ ), the height of the bar ( $x_3$ ), and thickness of the bar ( $x_4$ ).

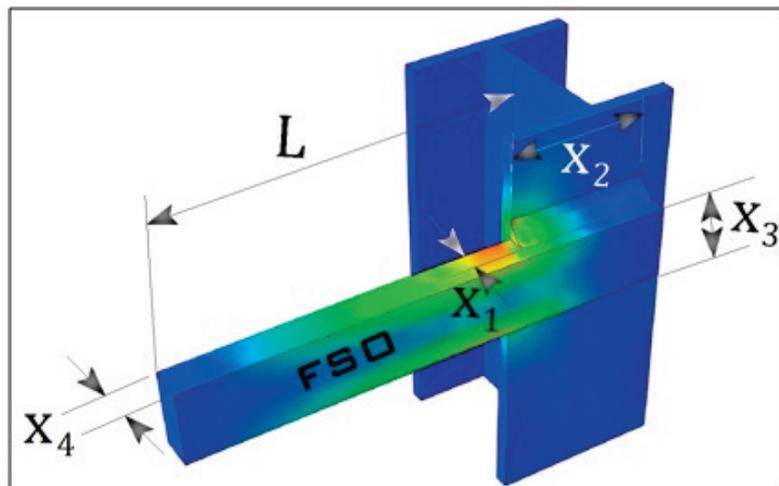


Fig. 11. Welded beam design problem

Coello and Deb (Deb, 1997; Coello and Montes, 2002) used the GA to solve this problem. Lee and Geem (Lee and Geem, 2005) applied HS algorithm to this problem. Ragsdell and Philips (Zahara and Kao, 2009) exerted Richard-

son random method, Simplex method, Davidon-Fletcher-Powell method, and Griffith and Stewart linear approach to be mathematical solving to this problem. The objective function and the mechanical constraints are shown in Eq. (22).

$$\begin{array}{lll}
 \text{Minimize:} & \text{Subject to:} & \text{in which:} \\
 f(\vec{x}) & g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max} \leq 0 & P = 6000 \text{ lb} \\
 = 1.10471x_1^2x_2 & g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \leq 0 & L = 14 \text{ in} \\
 + 0.04811x_3x_4(14.0 + x_2) & g_3(\vec{x}) = \delta(\vec{x}) - \delta_{max} \leq 0 & \tau_{max} = 13600 \text{ psi} \\
 & g_4(\vec{x}) = x_1 - x_4 \leq 0 & \sigma_{max} = 30000 \text{ psi} \\
 \text{Variable range:} & g_5(\vec{x}) = P - P_c(\vec{x}) \leq 0 & \delta_{max} = 0.25 \text{ in} \\
 0.10 \leq [x_1, x_4] \leq 2.00 & g_6(\vec{x}) = 0.125 - x_1 \leq 0 & E = 30 \times 10^6 \text{ psi} \\
 0.10 \leq [x_2, x_3] \leq 10.0 & g_7(\vec{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5 \leq 0 & G = 12 \times 10^6 \text{ psi}
 \end{array}$$

Where:

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}, \quad \tau' = \frac{P}{\sqrt{2}x_1x_2}, \quad \tau'' = \frac{MR}{J}, \quad M = P\left(L + \frac{x_2}{2}\right), \quad R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2}, \quad J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2\right]\right\}, \quad \sigma(\vec{x}) = \frac{6PL}{x_3^2x_4}, \quad \delta(\vec{x}) = \frac{6PL^3}{Ex_3^2x_4}, \quad P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^3}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \tag{22}$$

Optimum location and also the amount of constraints are shown in Table 9. A good balance between constrains can be observed, compared to the others. This makes it possible for FSO to find

a better response in a feasible search space. The best results obtained by different methods are shown in Table 10.

Table 9: Best results given by well-known optimization algorithm and FSO for the welded beam design problem.

Parameter	GA2	CPSO	CAEP	HGA	NM-PSO	WCA	MBA	APSO	GWO	IGMM	FSO
x1	0.205986	0.202369	0.2057	0.2057	0.20583	0.205728	0.205729	0.202701	0.205676	0.205729	<b>0.20572964</b>
x2	3.471328	3.544214	3.4705	3.4705	3.468338	3.470522	3.470493	3.574272	3.478377	3.470496	<b>3.25312005</b>
x3	9.020224	9.04821	9.0366	9.0366	9.036624	9.03662	9.036626	9.040209	9.03681	9.036625	<b>9.03662391</b>
x4	0.20648	0.205723	0.2057	0.2057	0.20573	0.205729	0.205729	0.2059215	0.205778	0.205730	<b>0.20572964</b>
g1(x)	-0.103049	-13.65554	1.988676	1.988676	-0.02525	-0.034128	-0.001614	-117.467062	-794.279	-0.00627	<b>-4.888E-05</b>
g2(x)	-0.231747	-78.81407	4.481548	4.481548	-0.053122	-3.49E-05	-0.016911	-51.712981	-8.2855887	-0.01478	<b>-2.881E-05</b>
g3(x)	-0.0005	-0.00335	0	0	0.0001	-0.00000119	-0.0000002	-0.003221	-0.2283169	-3.19E-07	<b>-0.2283104</b>
g4(x)	-3.430044	-3.424572	-3.433213	-3.433213	-3.433169	-3.43298	-3.432982	-3.421741	-1.02E-04	-3.43298	<b>0</b>
g5(x)	-0.080986	-0.077369	-0.0807	-0.0807	-0.08083	-0.080728	-0.080729	-0.077701	-4.313487	-0.08073	<b>-1.856E-05</b>
g6(x)	-0.235514	-0.235595	-0.235538	-0.235538	-0.23554	-0.23554	-0.23554	-0.235571	-0.0806	-0.23554	<b>-0.08072964</b>
g7(x)	-58.64688	-4.472858	2.603347	2.603347	-0.031555	-0.013503	-0.001464	-18.367012	-3.389578	-0.00504	<b>-3.4524255</b>
f(x)	1.728226	1.728024	1.724852	1.724852	1.724717	1.724856	1.724853	1.736193	1.72624	1.724853	<b>1.6952471</b>

Table 10: Statistical comparison of results for the welded beam design problem.

Method	Worst	Mean	Best	Std	NFEs
GA1	1.785835	1.771973	1.748309	0.0112	900000
GA2	1.993408	1.792654	1.728226	0.0747	80000
CAEP	3.179709	1.971809	1.724852	0.443	50020
CPSO	1.782143	1.748831	1.728024	0.0129	240000
HPSO	1.814295	1.74904	1.724852	0.0401	81000
PSO-DE	1.724852	1.724852	1.724852	6.7E-16	66600
NM-PSO	1.733393	1.726373	1.724717	0.0035	80000
SC	6.399678	3.002588	2.385434	0.96	33095
DE	1.824105	1.768158	1.733461	0.0221	204800
WCA	1.744697	1.726427	1.724856	0.00429	46450
LCA	1.7248523	1.7248523	1.7248523	7.11E-15	15000
MBA	1.724853	1.724853	1.724853	6.94E-19	47340
APSO	1.993999	1.877851	1.736193	0.076118	50000
IGMM	1.74769	1.732152	1.724853	0.00714	8000
<b>FSO</b>	<b>1.70139213</b>	<b>1.696729</b>	<b>1.6952471</b>	<b>7.24E-04</b>	<b>4500</b>

FSO offered better results using fewer NFEs as compared to all other considered algorithms. However, the best solution is obtained with a standard deviation value (SD) greater than that

given by LCA. It should be noted that the best result obtained by LCA violates some of the constraints and then cannot be compared to other best results. The FSO algorithm approaches the

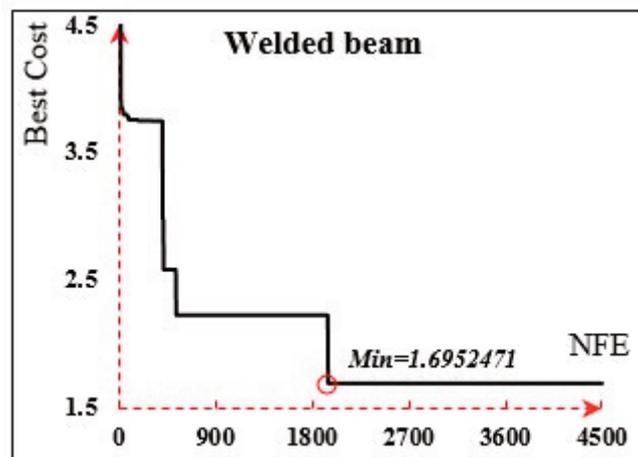


Fig. 12. Convergence curve of welded beam design problem

optimized solution at approximately the middle of iterations. The performances of the algorithms are also summarized in terms of NFEs and SD of responses in Table 10. Each of the algorithms is analyzed 20 times to obtain the statistical results in Table 10. FSO provides a sufficient amount of answers in welded beam design optimization problem and the results are so better than the others.

### Pressure vessel design

This problem corresponds to the weight minimization of a cylindrical pressure vessel with two spherical heads. There are four design variables (in inches): the thickness of the pressure vessel ( $x_1$ ), the thickness of the head ( $x_2$ ), the inner radius of the vessel ( $x_3$ ) and the length of the cylindrical component ( $x_4$ ). It becomes a nonlinearly

constrained mixed discrete-continuous optimization problem, since there are two discrete variables ( $x_1$  and  $x_2$ ) and two continuous variables

( $x_3$  and  $x_4$ ).

The objective function, constraints, and the domain of decision variables are given as Eq. (23).

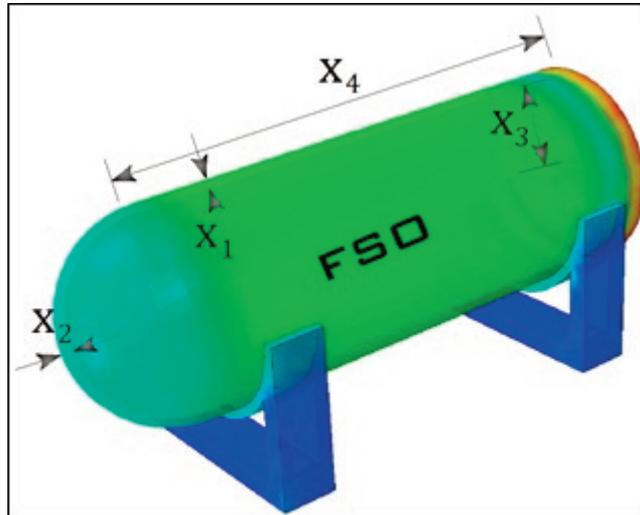


Fig. 13. Pressure vessel design problem

Minimize:

$$f(\vec{x}) = 0,6224x_1x_3x_4 + 1,7781x_2x_3^2 + 3,1661x_1^2x_4 + 19,84x_1^2x_3$$

Subject to:

$$g_1(\vec{x}) = -x_1 + 0,0193x_3 \leq 0$$

$$g_2(\vec{x}) = -x_2 + 0,00954x_3 \leq 0$$

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

$$g_4(\vec{x}) = x_4 - 240 \leq 0$$

Variable range:

$$[x_1, x_2] \in \{0,0,0625, \dots, 99\}$$

$$10,0 \leq [x_3, x_4] \leq 200,0$$

(23)

This problem was also assessed several times by researchers (Mirjalili, Mirjalili and Lewis, 2014; Mirjalili and Lewis, 2016; MiarNaeimi, Azizyan and Rashki, 2018). Meta-heuristic algorithms used to solve this problem include: PSO, GA, ES, DE, and ACO. Mathematical methods

also used to solve this problem include: Augmented Lagrangian-Multiplier and branch-and-bound method.

Results of the pressure vessel design by different algorithms are shown in Tables 11 and 12.

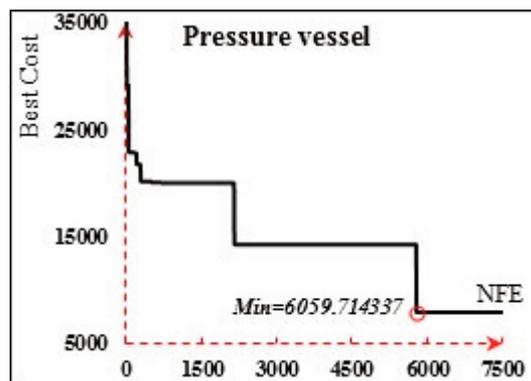


Fig. 14. Convergence curve of pressure vessel design problem

Table 11: Best results given by well-known optimization algorithm and FSO for the pressure vessel design problem.

Parameter	CDE	GA1	CPSO	HPSO	NM-PSO	G-QPSO	WCA	MBA	APSO	GWO	IGMM	FSO
x1	0.8125	0.8125	0.8125	0.8125	0.8036	0.8125	0.7781	0.7802	0.8125	0.8125	0.8125	<b>0.8125</b>
x2	0.4375	0.4375	0.4375	0.4375	0.3972	0.4375	0.3846	0.3856	0.4375	0.4345	0.4375	<b>0.4375</b>
x3	42.0984	42.0974	42.0913	42.0984	41.6392	42.0984	40.3196	40.4292	42.0984	42.089181	42.098445	<b>42.09844558</b>
x4	176.6376	176.654	176.7465	176.6366	182.412	176.6372	200	198.4964	176.6374	176.758731	176.63659	<b>176.6365968</b>
g1(x)	-6.67E-07	-2.01E-03	-1.37E-06	-8.80E-07	0.0000365	-8.79E-07	6.828e-05	+8.356e-05	-8.799E-07	-0.00312505	-3.1764E-10	<b>-3.059e-10</b>
g2(x)	-3.58E-02	-3.58E-02	-0.000359	-3.58E-02	0.0000379	-0.0358	4.898e-05	+9.456e-05	-0.0359	-0.03296921	-0.03756	<b>-0.03756476</b>
g3(x)	-3.705123	-24.7593	-118.7687	3.1226	-1.5914	-0.2179	1.33120	-86.3645	-1.3315386	-40.115408	-0.00012	<b>-4.237E-03</b>
g4(x)	-63.3623	-63.346	-63.2535	-63.3634	-57.5879	-63.3628	-40	-41.5035	-63.362	-63.241269	-63.3634	<b>-63.363403</b>
f(x)	6059.734	6059.9463	6061.0777	6059.7143	5930.3137	6059.7208	5885.3327	5889.3216	6059.72418	6051.5639	6059.7143	<b>6059.714337</b>

Table 12. Statistical comparison of results for the pressure vessel design problem.

Method	Worst	Mean	Best	Std	NFEs
GA1	6308.497	6293.8432	6288.7445	7.4133	900000
GA2	6469.322	6177.2533	6059.9463	130.9297	80000
CPSO	6363.8041	6147.1332	6061.0777	86.45	240000
HPSO	6288.677	6099.9323	6059.7143	86.2	81000
NM-PSO	5960.0557	5946.7901	5930.3137	9.161	80000
G-QPSO	7544.4925	6440.3786	6059.7208	448.4711	8000
QPSO	8017.2816	6440.3786	6059.7209	479.2671	8000
PSO	14076.324	8756.6803	6693.7212	1492.567	8000
CDE	6371.0455	6085.2303	6059.734	43.013	204800
WCA	6590.2129	6198.6172	5885.3327	213.049	27500
LCA	6090.6114	6070.5884	6059.8553	11.37534	24000
MBA	6392.5062	6200.64765	5889.3216	160.34	70650
APSO	7544.49272	6470.71568	6059.7242	326.9688	200000
IGMM	6061.2868	6060.1598	6059.7143	0.5421	8000
<b>FSO</b>	<b>6060.18427</b>	<b>6059.7166</b>	<b>6059.714337</b>	<b>0.2469</b>	<b>7500</b>

Clearly, the fabrication costs of pressure vessel are significantly reduced as compared to the values obtained from the other algorithms. The power and accuracy of the FSO algorithm are clearly shown to find the optimal responses. There is need to mention some constraints that cannot be satisfied in a number of algorithms that include NM-PSO, WCA, and MBA.

**Multiple disk clutch brake design**

This is to minimize the mass of a multiple disc clutch brake. The decision variables x1, x2, x3,

x4, and x5 are respectively internal radius, external radius, thickness of the disc, actuating force, and number of friction surfaces.

All design variables are discrete. Objective function, constraints, variables range, and additional parameters are listed in Eq. 24.

This issue was previously optimized using NSGA-II, TLBO, WCA, and APSO. The comparison to discover the best solution and the statistical optimization results, given by such algorithms are shown in Tables 13 and 14, respectively.

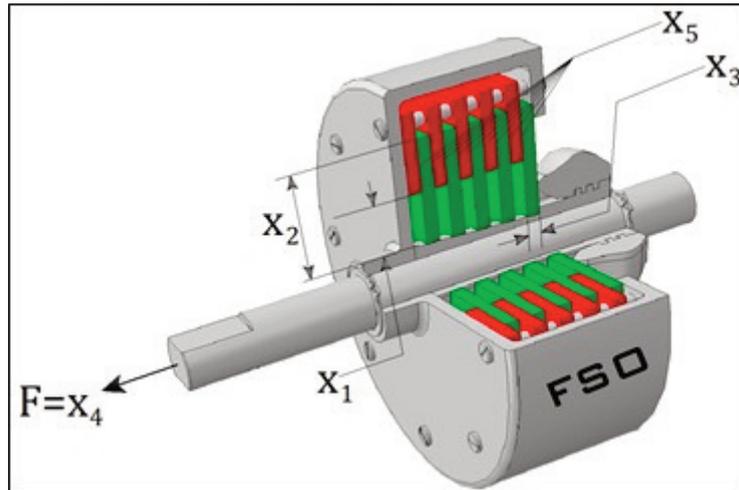


Fig. 15. Multiple disk clutch brake design problem

Minimize:

$$f(\vec{x}) = \pi(x_2^2 - x_1^2)x_3(x_5 + 1)\rho$$

Variable range:

$$x_1 \in \{60, 61, \dots, 79, 80\}$$

$$x_2 \in \{90, 91, \dots, 109, 110\}$$

$$x_3 \in \{1, 1.5, \dots, 3\}$$

$$x_4 \in \{600, 610, \dots, 990, 1000\}$$

$$x_5 \in \{2, 3, \dots, 8, 9\}$$

Where:

$$M_h = \frac{2\mu x_4 x_5 (x_2^3 - x_1^3)}{(x_2^2 - x_1^2)}, \quad p_{rz} = \frac{x_4}{\pi(x_2^2 - x_1^2)}$$

$$v_{sr} = \frac{2\pi n (x_2^3 - x_1^3)}{90(x_2^2 - x_1^2)}, \quad T = \frac{I_z \pi n}{30(M_h - M_f)}$$

Subject to:

$$g_1(\vec{x}) = x_2 - x_1 - \Delta r \geq 0$$

$$g_2(\vec{x}) = L_{max} - (x_5 + 1)(x_3 + \delta) \geq 0$$

$$g_3(\vec{x}) = p_{max} - p_{rz} \geq 0$$

$$g_4(\vec{x}) = p_{max} v_{sr, max} - p_{rz} v_{sr} \geq 0$$

$$g_5(\vec{x}) = v_{sr, max} - v_{sr} \geq 0$$

$$g_6(\vec{x}) = M_h - s M_s \geq 0$$

$$g_7(\vec{x}) = T \geq 0$$

$$g_8(\vec{x}) = T_{max} - T \geq 0$$

in which:

$$\Delta r = 20 \text{ (mm)}, \quad I_z = 55 \text{ (kg, mm}^2\text{)}, \quad p_{max} = 1 \text{ (MPa)}$$

$$T_{max} = 15 \text{ (sec)}, \quad \mu = 0,50$$

$$s = 1,50, \quad M_s = 40 \text{ (Nm)}, \quad M_f = 3 \text{ (Nm)}$$

$$n = 250 \text{ (rpm)}, \quad v_{sr, max} = 10 \left(\frac{m}{s}\right), \quad L_{max} =$$

$$30 \text{ (mm)}$$

(24)

Table 13: Best results given by well-known optimization algorithm and FSO for the multiple disk clutch brake design problem.

Parameter	NSGA-II	TLBO	WCA	APSO	FSO
x1	70	70	70	76	<b>70</b>
x2	90	90	90	96	<b>90</b>
x3	1.5	1	1	1	<b>1</b>
x4	1000	810	910	840	<b>870</b>
x5	3	3	3	3	<b>3</b>
g1(x)	0	0	0	0	<b>0</b>
g2(x)	22	24	24	24	<b>24</b>
g3(x)	0.9005	0.919427	0.909480	0.922273167327214	<b>0.913459499693782</b>
g4(x)	9.7906	9830.3710	9.809429	9.82421128537948	<b>9.81780598958333</b>
g5(x)	7.8947	7894.6965	7.894696	7.73837800183432	<b>7.89469658978184</b>
g6(x)	3.3527	0.702013	2.231421	1.3966105059236	<b>0.875577140380633</b>
g7(x)	60.625	37706.25	49.768749	48.8483720930233	<b>44.94374999999999</b>
g8(x)	11.6473	14.297986	12.768578	13.6033894940764	<b>14.1244228596193</b>
f(x)	0.4704	0.313657	0.313657	0.337181	<b>0.313656610534405</b>

Table 14: Statistical comparison of results for the multiple disk clutch brake design problem.

Method	Worst	Mean	Best	Std	NFEs
NSGA-II	0.5069	0.4829	0.4704	0.002354	>900
TLBO	0.392071	0.327166	0.313657	NA	>900
WCA	0.313656	0.313656	0.313657	1.69E-16	500
APSO	0.716313	0.506829	0.337181	0.09767	2000
<b>FSO</b>	<b>0.333705</b>	<b>0.3139423</b>	<b>0.313656610534405</b>	<b>4.25E-6</b>	<b>400</b>

As shown in Table 13, FSO, TLBO, and WCA other optimization algorithms with regard to methods converge at the same optimum answer. computation endeavor (NFEs) and stability (SD). FSO shows an advantage in comparison to

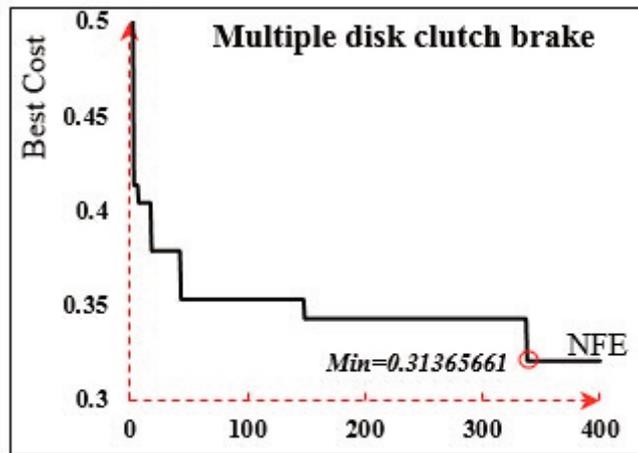


Fig. 16. Convergence curve of multiple disk clutch brake design problem

This is to minimize the cost of the gear ratio of the gear train as shown in Fig. 17. This problem has four decision variables:  $x_1$ ,  $x_2$ ,  $x_3$ , and  $x_4$  which are the number of teeth for the gears. The constraints are just limits on design variables (side constraints).

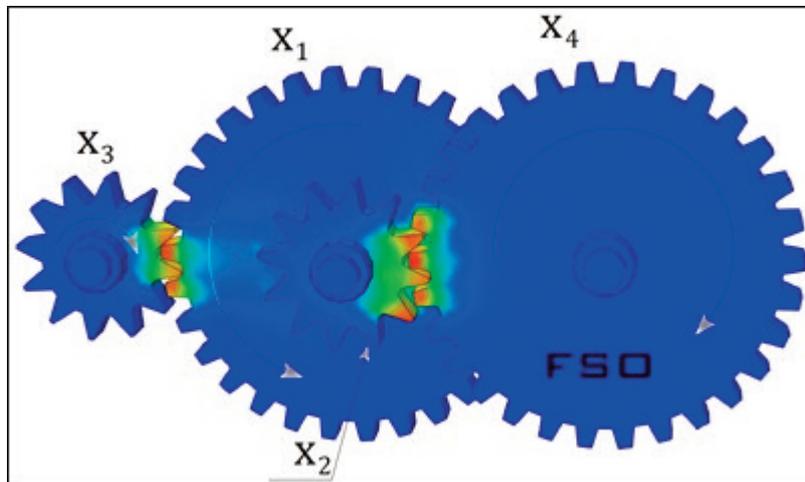


Fig. 17. Gear train design problem

Design variables to be optimized are in integer form since each gear requires an integer number of teeth. It is well known that constrained problems with discrete variables may increase the

complexness of the problem. The lower and upper bounds of integer design variables are 12 and 60, respectively. The mathematical formulation of the problem is reported in Eq. (25).

$$\text{Minimize: } f(\vec{x}) = \left( \left( \frac{1}{6,931} \right) - \left( \frac{x_2 x_3}{x_1 x_4} \right) \right)^2 \quad \text{Variable range: } x_i \in \{12, 13, \dots, 59, 60\}, i = 1, 2, 3, 4, \quad (25)$$

This problem was introduced by Sandgran (1990). The best achieved cost for the current issue is 2.7009e-12.

16, 19, and 49, respectively.

However, the decisive factor in this problem is to find the optimal solution with the minimal function evaluation.

Optimum locations of x1, x2, x3, and x4 are 43,

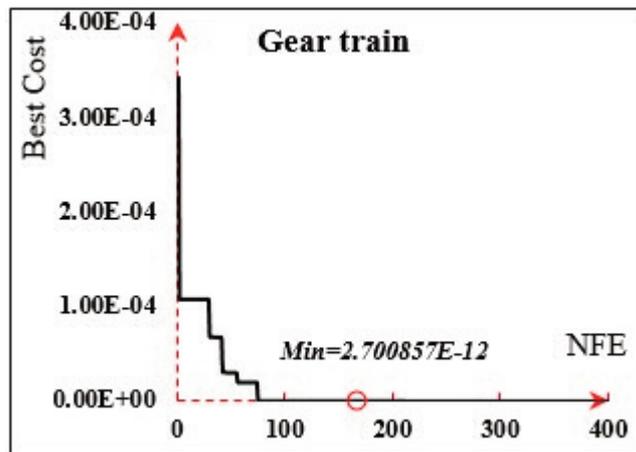


Fig. 18. Convergence curve of gear train design problem

Table 15: Best results given by well-known optimization algorithm and FSO for the gear train design problem.

Method	Worst	Mean	Best	Std	NFEs
CS	2.3576E-9	1.9841E-9	2.7009E-12	3.5546E-9	5000
MBA	2.06290E-8	2.471635E-09	2.700857E-12	3.94E-09	1120
APSO	7.0726E-06	4.781676E-07	2.700857E-12	1.44E-06	8000
IGMM	2.36E-09	6.25E-10	2.7009E-12	8.61E-10	480
<b>FSO</b>	<b>3.8248E-9</b>	<b>2.72645E-08</b>	<b>2.700857E-12</b>	<b>3.27E-9</b>	<b>400</b>

**Real application of FSO in gravity dam design engineering**

Gravity dams are made by concrete or rock masonry and are intended to hold back water by only using the weight of the concrete materials to face up to the applied loads. This problem aims to minimize the weight of a gravity dam. The

constraints are sliding and overturning safety factors, and also tensile and compressive stresses on the upstream and downstream faces of the dam. This paper presents an optimization of an important concrete gravity dams, including SHAFAROU D dam. SHAFAROU D gravity dam project is located in the North-Western of Iran

and it is used as a practical example. This problem includes nine decision variables (x1 to x9) that are shown in Fig. 19.

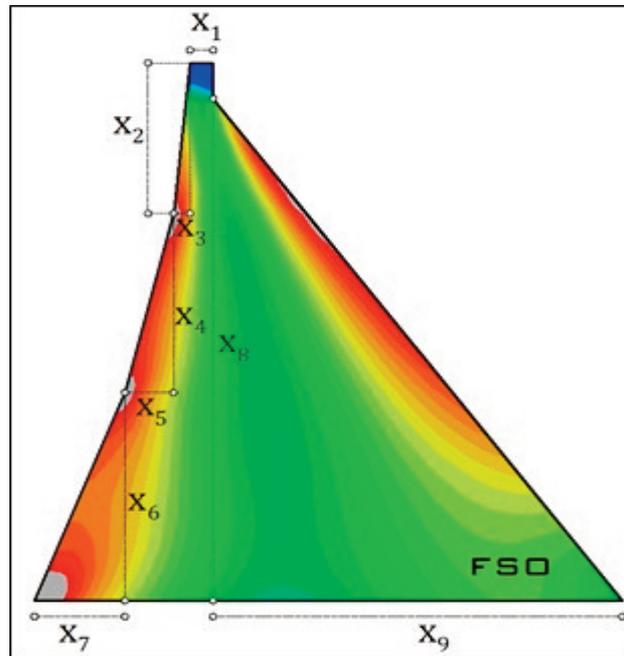


Fig. 19. Concrete gravity dam design problem

The objective function is a minimization of concrete volume of the dam. Unusual loading on the dam is considered, throughout these practices. The applied loads include dead load (weight of the dam), hydrostatic load in the full reservoir state, earthquake, sediment load and uplift. Researchers in the field of optimization of

concrete dams are Mahani et al. (Nourani et al., 2008) Cai et al. (Liu, Cai and Wang, 2010), Gandomi et al. (Mirjalili et al., 2017), and Guan (Shiqin, Jianjun and Guangxing, 2009).

Objective function, constraints, variables range, equilibrium equations, and constant parameters of the problem are shown in Eq. (26).

$$\text{Minimize } f(\vec{x}) = x_1 H + x_3(x_4 + x_6) + x_5 x_6 + \frac{1}{2}[x_2 x_3 + x_4 x_5 + x_6 x_7 + x_8 x_9]$$

Subject to:

$$g_1(\vec{x}) = 4 - SFS \leq 0,$$

$$g_2(\vec{x}) = 1,5 - SFO \leq 0,$$

$$g_3(\vec{x}) = -\sigma_U \leq 0,$$

$$g_4(\vec{x}) = -\sigma_D \leq 0,$$

$$g_5(\vec{x}) = \sigma_U - \sigma_{max} \leq 0,$$

$$g_6(\vec{x}) = \sigma_D - \sigma_{max} \leq 0,$$

Where:

$$SFS = \frac{f \cdot \sum F_V + \sigma(b)}{\sum F_H}, \quad SFO = \frac{\sum M_R}{\sum M_O},$$

$$\sigma_U = \frac{\sum F_V}{b} - \frac{6 \sum M_O}{b^2}, \quad \sigma_D = \frac{\sum F_V}{b} + \frac{6 \sum M_O}{b^2}, \quad \sum F_H = F h_H + P_{SH},$$

$$\sum F_V = P_{SV} + W + F h_V - F_{Uplift} - F_E,$$

$$b = x_1 + x_3 + x_5 + x_7 + x_9, \quad F h_H = \frac{1}{2} \gamma_w h^2, \quad F h_V = \gamma_w V_w,$$

$$F_{Uplift} = \frac{1}{2} \gamma_w h, \quad F_E = \alpha W, \quad P_{SH} = \frac{1}{2} \gamma_s h_s^2 \frac{1 - \sin \varphi}{1 + \sin \varphi},$$

$$P_{SV} = \frac{1}{2} \gamma_s h_s^2 \tan \theta, \quad \theta = \frac{x_6}{x_7},$$

Variable range:

$$0,10 \leq x_1 \leq 20,0,$$

$$1,00 \leq x_2 \leq 50,0,$$

$$1,00 \leq x_3 \leq 20,0,$$

$$5,00 \leq [x_4, x_5, x_6, x_7] \leq 50,0,$$

$$50,0 \leq [x_8, x_9] \leq 150,0.$$

in Which:

$$\gamma_w = 9,81 \frac{KN}{m^3}, \quad \gamma_c = 23 \frac{KN}{m^3}, \quad \gamma_s = 15 \frac{KN}{m^3}$$

$$\sigma_{max} = 25000 \frac{KN}{m^2}, \quad \sigma = 3500 \frac{KN}{m^2},$$

$$H = 150 \text{ m}, \quad h = 145 \text{ m}, \quad h_s = 7 \text{ m},$$

$$\varphi = 50^\circ, \quad \alpha = 0,05, \quad f' = 0,70.$$

The optimal values of decision variables ( $x_1$  to  $x_9$ ), summation of resistant and driving torques ( $\sum M_R$  and  $\sum M_O$ ) and forces ( $\sum F_H$  and  $\sum F_V$ ), horizontal hydrostatic loads ( $F_{H_H}$ ), horizontal sediment loads ( $P_{SH}$ ), vertical resistant load of water

and sediment on the slopes ( $F_{h_V}$  and  $P_{SV}$ ), upstream and downstream stresses ( $\sigma_U$  and  $\sigma_D$ ), safety factors ( $SFS$  and  $SFO$ ), constraints of problems ( $g_1(x^*)$  to  $g_6(x^*)$ ) and the amount of objective functions ( $f(x^*)$ ) are shown in Table 16.

Table 16: Best results given by well-known optimization algorithm and FSO for the concrete gravity dam design problem.

	SHAFAROU D Existing Dam	PSO	GA	FSO
$x_1$	4	6.0323	5.1965	<b>5.1678</b>
$x_2$	42	37.805	31.8972	<b>25.1978</b>
$x_3$	4.2	2.0582	2.3238	<b>4.9556</b>
$x_4$	50	73.2012	75.9014	<b>55.7811</b>
$x_5$	12.5	11.5573	11.0781	<b>16.0874</b>
$x_6$	58	39.02	42.2014	<b>21.0009</b>
$x_7$	23.2	30.6433	28.9231	<b>29.9036</b>
$x_8$	140	125.6454	132.025	<b>120.169</b>
$x_9$	105	98.3741	101.713	<b>85.382</b>
$\sum M_R$	23979329.67	20778650.8	21584896.2	<b>18520793</b>
$\sum M_o$	5056164.707	5055074.16	5055375.01	<b>5051705.95</b>
$\sum F_H$	103176.3093	103176.309	103176.309	<b>103176.309</b>
$\sum F_V$	262408.494	234946.624	242356.882	<b>215538.616</b>
$F_{h_H}$	103127.625	103127.625	103127.625	<b>103127.625</b>
$F_{h_V}$	32730.084	42455.6199	39249.451	<b>53236.0323</b>
$P_{SH}$	48.6843168	48.6843168	48.6843168	<b>48.6843168</b>
$P_{SV}$	918.75	468.382353	536.215499	<b>258.090638</b>
$\sigma_U$	406.6809	207.527923	262.034703	<b>9.3743472</b>
$\sigma_D$	3071.231545	2954.60698	2985.96602	<b>3037.18705</b>
$SFS$	6.899218925	6.63488199	6.70668075	<b>6.26223367</b>
$SFO$	4.742592669	4.11045419	4.26969238	<b>3.66624526</b>
$g_1(x^*)$	-2.8992	-2.6349	-2.7067	<b>-2.2622</b>
$g_2(x^*)$	-3.2426	-2.6105	-2.7697	<b>-2.1662</b>
$g_3(x^*)$	-406.6809	-207.5279	-262.0347	<b>-9.3743</b>
$g_4(x^*)$	-3.07E+03	-2.95E+03	-2.99E+03	<b>-3.04E+03</b>
$g_5(x^*)$	-2.46E+04	-2.48E+04	-2.47E+04	<b>-2.50E+04</b>
$g_6(x^*)$	-2.19E+04	-2.20E+04	-2.20E+04	<b>-2.20E+04</b>
$f(x^*)$	10502.1	8820.771	9303.5442	<b>7448.7743</b>

These quantities are shown for the existing dam at the first column, and the other columns show the comparison for the best solution of PSO, GA, and FSO in terms of the value of design variables and function value.

Table 16 shows that FSO has the best response for the SHAFAROU D gravity dam design problem, with respect to the function value (Cost). It

should be noted that FSO is able to find the answers with fewer NFEs and low SD as compared to PSO and GA.

PSO, GA and FSO represent 15.9, 11.4 and 28.2% reduction in the concrete volume of the SHAFAROU D dam, respectively. The FSO algorithm has a better performance than the other two algorithms. It shows the ability of FSO to

solve complex problems and search for the best answers in a close domain around the boundary of constraints.

As Shown in Figure 20, FSO is far better than PSO and slightly better GA, with the same NFEs.

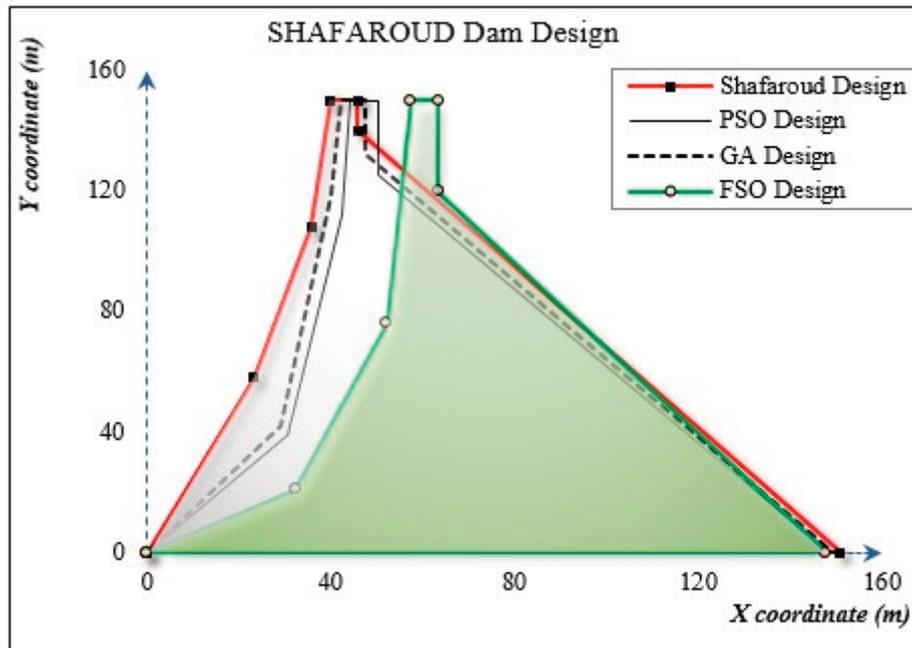


Fig. 20. Obtained results of concrete gravity dam problem

## CONCLUSIONS

The new proposed FSO algorithm is inspired by the flying squirrels behaviors in their natural life. These behaviors including jumping from branch to branch and walking on the ground or trees to find food and also contact each other in order to escape from enemies. The FSO algorithm was benchmarked on twelve well-known test functions in a 100 dimensions space and found to be very efficient in terms of exploration and exploitation. This algorithm is also employed for solving structural optimization problems. Five structural engineering problems of semi-real world have been used to evaluate the performance of FSO. The results show the ability of FSO to deal with these challenging problems with continuous, discrete, and mixed decision variables. Finally, FSO has been applied to a real problem in the field of concrete gravity dam engineering and it provides a considerable development of dam section design in comparison to the existing solutions. It shows the applicability of FSO in dealing with real-world engineering

problems. Flying squirrels are used as the search agents. They should be sorted based on their performance. Better half of the flying squirrels (that are close to the food) are moved by a random walk step (normal distribution) and the others fly (Lévy flight). They have a great desire to random search and finding the new possible global optimum locations. Using these motion patterns makes it possible to find the optima with a few NFEs. For future study, the development of FSO is in progress to solve the multi-objective optimization problems.

## REFERENCES

- Banichuk, N. V., & Neittaanmäki, P. (2009). *Structural optimization with uncertainties* (Vol. 162). Springer Science & Business Media.
- Cagnina, L. C., Esquivel, S. C., & Coello, C. A. C. (2008). Solving engineering optimization problems with the simple constrained particle swarm optimizer. *Informatica*, 32(3).
- Chong, E. K., & Zak, S. H. (2013). *An introduc-*

- tion to optimization (Vol. 76). John Wiley & Sons.
- Coello, C. A. C., & Montes, E. M. (2002). Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics*, 16(3), 193-203.
- Cuevas, E., Echavarría, A., & Ramírez-Ortegón, M. A. (2014). An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation. *Applied intelligence*, 40(2), 256-272.
- Deb, K. (1997). GeneAS: A robust optimal design technique for mechanical component design. In *Evolutionary algorithms in engineering applications*, 497-514, Springer, Berlin, Heidelberg.
- Dorigo, M., & Birattari, M. (2010). *Ant colony optimization*, 36-39, Springer US.
- Du, H., Wu, X., & Zhuang, J. (2006, September). Small-world optimization algorithm for function optimization. In *International Conference on Natural Computation* (pp. 264-273). Springer, Berlin, Heidelberg.
- Eberhart, R., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 39-43, Ieee.
- Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012). Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures*, 110, 151-166.
- Fausto, F., Cuevas, E., Valdivia, A., & González, A. (2017). A global optimization algorithm inspired in the behavior of selfish herds. *Biosystems*, 160, 39-55.
- Formato, R. A. (2007). Central force optimization: a new metaheuristic with applications in applied electromagnetics, *Prog Electromagn Res*, 77, 425-491.
- Gandomi, A. H., Yang, X. S., & Alavi, A. H. (2011). Mixed variable structural optimization using firefly algorithm. *Computers & Structures*, 89(23-24), 2325-2336.
- Gandomi, A. H., Yang, X. S., & Alavi, A. H. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with computers*, 29(1), 17-35.
- Gandomi, A. H., Yang, X. S., & Alavi, A. H. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with computers*, 29(1), 17-35.
- Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine learning*, 3(2), 95-99.
- Hatamlou, A. (2013). Black hole: A new heuristic optimization approach for data clustering. *Information sciences*, 222, 175-184.
- Holloway, G. L., & Malcolm, J. R. (2007). Northern and southern flying squirrel use of space within home ranges in central Ontario. *Forest Ecology and Management*, 242(2-3), 747-755.
- Kaveh, A. (2014). *Advances in metaheuristic algorithms for optimal design of structures* (pp. 11-43). Switzerland: Springer.
- Kaveh, A., & Bakhshpoori, T. (2016). Water evaporation optimization: a novel physically inspired optimization algorithm. *Computers & Structures*, 167, 69-85.
- Kaveh, A., & Farhoudi, N. (2013). A new optimization method: Dolphin echolocation. *Advances in Engineering Software*, 59, 53-70.
- Kaveh, A., & Ghazaan, M. I. (2017). A new meta-heuristic algorithm: vibrating particles system. *Scientia Iranica. Transaction A, Civil Engineering*, 24(2), 551.
- Kaveh, A., & Khayatazad, M. (2012). A new meta-heuristic method: ray optimization. *Computers & structures*, 112, 283-294.
- Kaveh, A., & Mahdavi, V. R. (2014). Colliding bodies optimization: a novel meta-heuristic method. *Computers & Structures*, 139, 18-27.
- Kaveh, A., & Talatahari, S. (2010). A novel heuristic optimization method: charged system search. *Acta Mechanica*, 213(3-4), 267-289.
- Kaveh, A., & Talatahari, S. (2010). An improved ant colony optimization for constrained engineering design problems. *Engineering Computations*, 27(1), 155-182.
- Kothari, R. P., & Kroese, D. P. (2009, December). Optimal generation expansion planning

- via the cross-entropy method. In *Proceedings of the 2009 Winter Simulation Conference (WSC)*, 1482-1491, IEEE.
- Koza, J. R., & Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection*. MIT press.
- Lam, A. Y., & Li, V. O. (2010). Chemical-reaction-inspired metaheuristic for optimization. *IEEE transactions on evolutionary computation*, 14(3), 381-399.
- Lamberti, L., & Pappalettere, C. (2011). Metaheuristic design optimization of skeletal structures: a review. *Computational technology reviews*, 4(1), 1-32.
- Lee, K. S., & Geem, Z. W. (2005). A new metaheuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Computer methods in applied mechanics and engineering*, 194(36-38), 3902-3933.
- Li, X. L. (2003). A new intelligent optimization-artificial fish swarm algorithm, *Doctor thesis, Zhejiang University of Zhejiang, China*.
- Liu, H., Cai, Z., & Wang, Y. (2010). Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Applied Soft Computing*, 10(2), 629-640.
- MiarNaeimi, F., Azizyan, G., & Rashki, M. (2018). Multi-level cross entropy optimizer (MCEO): an evolutionary optimization algorithm for engineering problems. *Engineering with Computers*, 34(4), 719-739.
- Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, 89, 228-249.
- Mirjalili, S. (2015). The ant lion optimizer. *Advances in Engineering Software*, 83, 80-98.
- Mirjalili, S. (2016a). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, 27(4), 1053-1073.
- Mirjalili, S. (2016b). SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*, 96, 120-133.
- Mirjalili, S., & Hashim, S. Z. M. (2012). BMOA: binary magnetic optimization algorithm. *International Journal of Machine Learning and Computing*, 2(3), 204.
- Mirjalili, S., & Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation*, 9, 1-14.
- Mirjalili, S., & Lewis, A. (2014). Adaptive gbest-guided gravitational search algorithm. *Neural Computing and Applications*, 25(7-8), 1569-1584.
- Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, 95, 51-67.
- Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, 114, 163-191.
- Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Computing and Applications*, 27(2), 495-513.
- Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, 69, 46-61.
- Mirjalili, S., Mirjalili, S. M., & Yang, X. S. (2014). Binary bat algorithm. *Neural Computing and Applications*, 25(3-4), 663-681.
- Moosavian, N., & Roodsari, B. K. (2014). Soccer league competition algorithm: A novel metaheuristic algorithm for optimal design of water distribution networks. *Swarm and Evolutionary Computation*, 17, 14-24.
- Mucherino, A., & Seref, O. (2007, November). Monkey search: a novel metaheuristic search for global optimization. In *AIP conference proceedings*, 953(1), 162-173, AIP.
- Nourani, V., Mohebbi, A., Bonab, S. B., & Nabi, M. (2008). *Honey bee mating optimization (HBMO) implementation in concrete gravity dam layout optimization*. na.
- Rao, S. S. (2009). *Engineering optimization: theory and practice*. John Wiley & Sons.
- Ray, T., & Liew, K. M. (2003). Society and civilization: An optimization algorithm based on the simulation of social behavior. *IEEE Transactions on Evolutionary Computation*, 7(4),

- 386-396.
- Rechenberg, I. (1994). Computational intelligence imitating life. chapter Evolution strategy.
- Roth, M. (2005). Termite: A swarm intelligent routing algorithm for mobile wireless ad-hoc networks.
- Sadollah, A., Bahreininejad, A., Eskandar, H., & Hamdi, M. (2013). Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Applied Soft Computing*, 13(5), 2592-2612.
- Santosa, B., & Safitri, A. L. (2015). Biogeography-based optimization (BBO) algorithm for single machine total weighted tardiness problem (SMTWTP). *Procedia Manufacturing*, 4, 552-557.
- Saremi, S., Mirjalili, S., & Lewis, A. (2014). Biogeography-based optimisation with chaos. *Neural Computing and Applications*, 25(5), 1077-1097.
- Shiqin, Y., Jianjun, J., & Guangxing, Y. (2009, May). A dolphin partner optimization. In *2009 WRI Global Congress on Intelligent Systems*, 1, 124-128, IEEE.
- Simon, D. (2008). Biogeography-based optimization, *IEEE transactions on evolutionary computation*. IEEE, 12(6), 702-713.
- Varaee, H., & Ghasemi, M. R. (2017). Engineering optimization based on ideal gas molecular movement algorithm. *Engineering with Computers*, 33(1), 71-93.
- Wang, L., & Li, L. P. (2010). An effective differential evolution with level comparison for constrained engineering design. *Structural and Multidisciplinary Optimization*, 41(6), 947-963.
- Webster, B., & Bernhard, P. J. (2003). *A local search optimization algorithm based on natural principles of gravitation*.
- Weigl, P. D. (2007). The northern flying squirrel (*Glaucomys sabrinus*): a conservation challenge. *Journal of Mammalogy*, 88(4), 897-907.
- Wilson, T. M. (2010). *Limiting factors for northern flying squirrels (Glaucomys sabrinus) in the Pacific Northwest: A spatio-temporal analysis*. Union Institute and University.
- Yang, X. S. (2010a). A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)*, 65-74, Springer, Berlin, Heidelberg.
- Yang, X. S. (2010b). *Engineering optimization: an introduction with metaheuristic applications*. John Wiley & Sons.
- Yang, X. S. (2010c). Firefly algorithm, stochastic test functions and design optimisation. *arXiv preprint arXiv:1003.1409*.
- Yang, X. S. (2012, September). Flower pollination algorithm for global optimization. In *International conference on unconventional computing and natural computation*, 240-249, Springer, Berlin, Heidelberg.
- Yang, X. S., & Deb, S. (2009, December). Cuckoo search via Lévy flights. In *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, 210-214, IEEE.
- Yang, X. S., & Press, L. (2010). *Nature-Inspired Metaheuristic Algorithms Second Edition*.
- Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary computation*, 3(2), 82-102.
- Zahara, E., & Kao, Y. T. (2009). Hybrid Nelder-Mead simplex search and particle swarm optimization for constrained engineering design problems. *Expert Systems with Applications*, 36(2), 3880-3886.
- Zhang, M., Luo, W., & Wang, X. (2008). Differential evolution with dynamic stochastic selection for constrained optimization. *Information Sciences*, 178(15), 3043-3074.