CrossMark

METHODOLOGIES AND APPLICATION

A novel evolutionary technique based on electrolocation principle of elephant nose fish and shark: fish electrolocation optimization

Vivekananda Haldar[1] · Niladri Chakraborty[1]

Abstract Elephant nose fish searches its food such as larvae by active electrolocation. It discharges electric pulse through its electric organ in tail and detects the object by analyzing the geometrical property of projected electrical image on it. The capacitance value found out from that electric image helps the fish to reach near the food source. Shark also uses passive electrolocation for the same purpose. It can target its prey by sensing the electrical wave generated due to the muscle twitching of small living beings in water. Both the above physiological phenomena, concerning the active and passive electrolocation of fish, has been mathematically developed as nature-inspired meta-heuristic technique named fish electrolocation optimization (FEO). A comparative study based on benchmark functions has been done amongst real coded genetic algorithm, accelerated particle swarm optimization, particle swarm optimization, harmony search and the proposed algorithm. Furthermore, comparative study has been done with simulated annealing and differential evolution on eggcrate function. The proposed technique has also been implemented on real-world optimization problem related to cost-based reliability enhancement in radial distribution system. It can be said by comparing percentage of success, mean number of function evaluation and standard deviation that FEO algorithm works better than other mentioned meta-heuristic techniques.

✉ Vivekananda Haldar
    vivekananda.haldar@gmail.com

    Niladri Chakraborty
    chakraborty_niladri2004@yahoo.com

[1] Power Engineering Department, Jadavpur University,
    Salt Lake Campus, Block-LB, Plot-8, Sector-III, Salt Lake,
    Kolkata 700098, India

Abbreviations

$diff$	Difference between maximum and minimum limit of solution variable
$longrange$	A set of discrete values in long range
$shortrange$	A set of discrete values in short range
$p1_l, p2_l, p3_l$	Constant terms for $longrange$ formulation
$p1_s, p2_s, p3_s$	Constant terms for $shortrange$ formulation
$vshortrange$	A set of discrete values in very short range
i, k, j	Index terms
$slope, vs$	Electrical image slope, short distance interval value
$xnew, x^{\min}$ and $x^{\max}$	Calculated solution value after evolution, minimum and maximum limit of solution variable 'x'
$elec^{pulse}$	Value of electric pulse for generation of new electrical wave
$capu, capl$	Capacitor upper limit, capacitor lower limit
$capint, caphover$	Initial capacitor value, capacitor value when the conceptual electro-fish is hovering and searching
$rand, floor, fix, randperm, randn$ and $length$	Standard MATLAB® 7.0 library functions

🌱 Springer

$rand^i, rand^i_j$	Random value for ith individual amongst population, random value for ith individual and jth variable
$prob^{div}, prob^{sel}$	Probability of divergence, probability of selection
$prob^{rng}$	Probability of range
x^t_{best}, x^t_{worst}	Best found variable value at tth iteration, worst found variable value at tth iteration
$ch1$ and $ch2$	Minimum and maximum value of objective function for the first iteration
$g1, g2$	Constant terms for distance calculation
cap^{run}	Running capacitor value
$toggle$	Toggle switch or changeover switch
$\sigma(x_i)$	Symbol for standard deviation function
$slope^{const}$	A constant value for $elec^{pulse}$ generation
$S, m1$ and $m2$	Array of random index terms concerning the length of $longrange$, $shortrange$ and $vshortrange$
$n, h1$ and $h2$	Selected random values from $s, m1$ and $m2$
$c2$ and $c3$	Values concerning $shortrange$ and $vshortrange$

1 Introduction

Many natural procedures were mimicked as soft computing-based optimization tool. In the early 1970s Holland (1975) adopted the idea of Charles Darwin's theory of natural selection to formulate genetic algorithm. Later Goldberg (1989) studied that elaborately for making it fit for optimization and machine learning. Kirkpatrick et al. (1983) developed simulated annealing considering the metal's annealing process. That process behind the algorithm includes the metal's cooling and freezing phenomena into crystalline state with larger crystal size and minimum energy. Moscato (1989) had established mimetic algorithm derived from genetic algorithm and martial art. Deb (1991) applied genetic algorithm for engineering optimization problem. He also developed an efficient constraint handling method for genetic algorithm (Deb 2000). Reynolds (1994) proposed cultural algorithm considering belief space and population space as the two main ingredients. Kennedy and Eberhat (1995) established particle swarm optimization (PSO) on the basis of swarm behavior of birds. Later that algorithm had been improved. That improved PSO was applied for optimal power flow (Vo and Schegner 2013). Storn (1996) proposed differential evolution (DE) as a useful tool for function optimization. Later Storn and

Price (1997) applied that technique for global optimization over continuous spaces. Foraging behavior of ants especially built on chemical messenger pheromone was studied (Dorigo 1992). Dorigo and Caro (1999) presented that meta-heuristic technique as ant colony optimization.

On the other hand, meta-heuristic techniques had been generated from swarm intelligence and bee colony behavior. Honey bee mating optimization was developed by Afshar et al. (2007) for doing optimal reservoir operation. Artificial bee colony algorithm was proposed by Karaboga and Basturk (2007). Bee colony optimization was also developed in this regard (Teodorovic' and DelÍOrco 2005). Nakrani and Tovey (2004) proposed honey bee algorithm. Later Yang (2005) proposed virtual bee algorithm for implementation in engineering optimization. Pham et al. (2005) studied the concept of honey bee's behavior and developed bee algorithm. Later bee algorithm was also studied and applied to find optimal solution (Yuce et al. 2015). Geem et al. (2001) proposed harmony search algorithm inspired by harmony of music. Later that technique was applied for continuous engineering optimization problem (Lee and Geem 2005). Biogeography-based optimization technique was developed by Simon (2006). Firefly algorithm was formulated on the basis of flashing characteristics of fireflies (Yang 2009). Bat's echolocation (Muhaureq et al. 2010; Yang 2010a, b) had also been considered to develop meta-heuristic technique conceptualizing prey catching phenomena of it. Later firefly algorithm and bat algorithm had been implemented for optimal placement and sizing of static VAR compensator for enhancement of voltage stability (Rao and Kumar 2015). On the other hand, cuckoo's breeding procedure had also got the attention of researcher to build a meta-heuristic technique (Yang and Deb 2009). Foraging behavior of few animals was also mathematically studied as optimization method. Passino (2002) developed bacterial foraging algorithm conceptualizing survival mechanism of *Escherecia coli* in changing environment. Monkey search algorithm was formulated imitating the behavior of monkey climbing trees in its search for food (Seref and Akcali 2002). Bioluminescence (Krishnanand and Ghose 2005; Olivera et al. 2011) had also been selected for developing higher level searching tool. Not only the animal's behavior was imitated, but also the plant's growth (Tong et al. 2004) and its photosynthesis process (Yang 2010a, b) was also considered for this kind of technique development. Cai et al. (2008) developed plant growth optimization on the basis of leaf growth, branching, phototropism and spatial occupancy. Most of the above algorithms were modified and also hybridized to have better soft computing tool (Hedar and Fukushima 2006; Das et al. 2008; Haldar and Chakraborty 2011a, b).

However, prey searching technique had also become an encouraging research context for developing meta-heuristic technique. Group hunting of animals such as lions, wolves,

and dolphins was studied as a soft computing technique named hunting search algorithm (Oftadeh et al. 2010). On the other hand, collective animal (Cuevas et al. 2012) and animal searching behavior (He et al. 2009) were also chosen to formulate optimization algorithm. Fish's electrolocation is an interesting prey searching procedure. This is identified in this research work for developing an optimization method.

Elephant nose fish *Gnathonemus petersii* has electric discharge organ (Baffet et al. 2008; Emde 1998, 1999, 2004a, b). Generated electric wave from that organ strikes surrounding objects inside water. The difference in sensing, the electrical field with and without objects, creates an electrical image on the surface of that fish's body. It then analyzes that image minutely. Depending on image parameters it calculates electrical capacitance value of that object. The capacitance value of food particles is already known to fish. If it feels that the targeted object is within the range of known capacitance then proceeds towards it calculating the distance from the targeted object. Otherwise it generates electric wave of new amplitude. On finding the desired object, it stops its electric organ discharge.

On the other hand, like active electrolocation, passive electrolocation is used by shark fish *Scyliorhinus canicula* for food catching purpose (Kalmijn 1971; Passive electrolocation in fish). Shark does not generate electric wave like elephant nose fish. It can sense feeble electric pulse generated from the muscle contraction of other living beings in the water. If the shark finds out that kind of electric pulse, then it quickly targets the direction from which the electrical wave is coming. This food searching technique of the above said electric fishes can be useful as an optimization technique.

However, active electrolocation of elephant nose fish was studied on robotic sensor technology (Startchev et al. 2011). Algorithm inspired from the electrolocation behavior of electric fish has been implemented in autonomous robot. Artificial sensor array was developed by mimicking that biological process (Maciver and Nelson 2001). That could provide electro-sensory capabilities to a submarine robotic explorer. Robotic sensing system was designed to locate objects underwater (Solberg et al. 2008, 2013). This was done through active movement of an electric field emitter and sensor apparatus. Echolocation and electrolocation were both biologically studied as sensory acquisition in active sensing system (Nelson and Maciver 2006). Recently Ammari et al. (2013) proposed a complex conductivity model problem for the quantitative analysis of that electrolocation phenomenon. However, concept of electrlocation as an optimization method has been hardly found in the literature. Passive electrolocation of shark has also not been conceptualized as searching tool hitherto. Here a soft computing-based meta-heuristic method called fish electrolocation optimization (FEO) is developed by mixing both active and passive electrolocation procedure of two fish, viz. elephant nose fish

and shark, respectively. This has been implemented on some test bed functions. However, the proposed algorithm's theoretical background has been briefly described in the next section.

2 Theoretical background of fish electrolocation optimization (FEO)

Fish electrolocation optimization has been thought on the context of electro-sensory perception of electric fish. Nocturnal animals generally rely on other senses instead of vision for object localization and prey catching. Weakly electric fish strongly depends on its electro-sensory organs which are more or less situated throughout its whole body (Emde and Schwarz 2002). Electrolocation is an adopted navigational approach for finding food item. Elephant nose fish *G. petersii* adopts this navigational procedure in dark and fulfills basic need. Its electric organ discharges in water. It generates electrical wave of a certain voltage amplitude and waveform from its tail electric organ. This is shown in Fig. 1.

Figure 1 depicts electric field lines as circular lines fencing the electric fish. After the electrical discharge in water, electric image is projected on the skin of that fish. The black circular spots shown in Fig. 1, on the fish's body, are denoted as the projected electric image of food and other non-food particles inside water. The electric image is formed on the basis of change in electrical voltage amplitude and waveform distortion with and without the object in the surrounding water. The fish searches its food depending on that image slope, image width; distance from the object, resistance and capacitance value (active electrolocation). Food objects, which are water plant or larvae, have specific impedance in respect of electrical characteristics. Additionally, food objects have a certain range of electrical capacitance value. The fish's electro-sensory organs are sensitive within that certain interval of electrical capacitance value. Beyond that limit whether it is in the lower or upper boundary area concerning electrical capacitance value, the fish is insensitive. That means it cannot sense any more. Depending on this judged electrical capacitance value and distance from the targeted object, it approaches towards the prey. Interestingly, the resistance value depends on the voltage or electrical wave amplitude, but capacitance value depends on the wave form distortion. On the other hand, image slope is considered as transition from rim to center area of image and its necessary to obtain the distance from the prey. The ratio of image slope and voltage peak amplitude has been considered as distance from the targeted object. Depending on both targeted object distance from fish's body and quantity and quality of that object, the fish develops electric pulse and proceeds towards it. Finally, it fetches the food object without using any vision but using active electrolocation intelligently (Hopkins 2005).
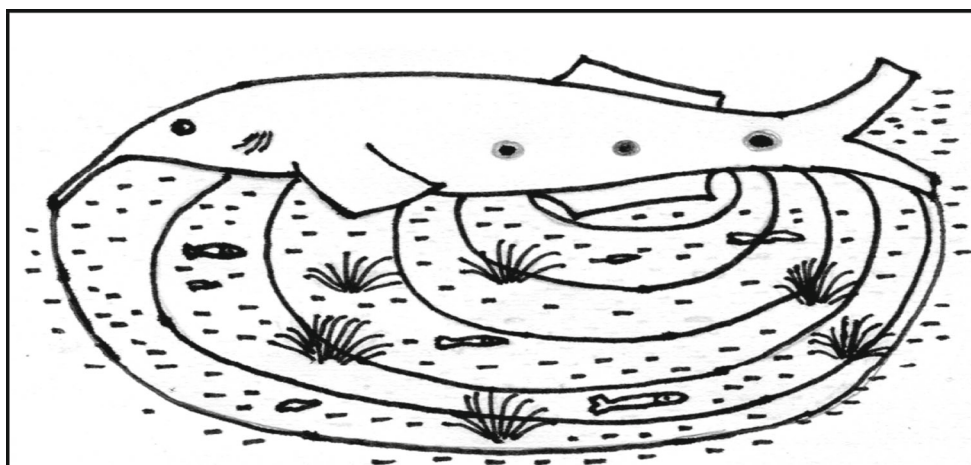
Fig. 1 Active electrolocation of elephant nose fish (*G. petersii*) (electric fish identifies the food objects or prey by judging the electric image projected on their body surface. The *big circular lines* in the water sig-

nify the electric field developed due the tail organ discharge. The *black circular spot* on the fish's body is denoted as the projected electric image)

Unlike elephant nose fish, shark utilizes its sense of passive electrolocation for detecting small fishes inside water (Shieh et al. 1996; Electric fish)). A glimpse of passive electrolocation by shark is shown in Fig. 2. Dotted circular lines surrounding small fishes in Fig. 2 show the low-voltage signals. The shark has such strong electro-sensory organ that it can sense muscular twitch of a fish dug in sand. It also uses passive electrolocation for navigation purpose analyzing ocean currents and earth's magnetic field.

In this work, a conceptual electro-fish has been considered to develop FEO. The fish has both the sense of active and passive electrolocation like the previously discussed elephant nose fish and shark, respectively. Not only it can create electrical wave and judge the electric image on the basis of electrical characteristics, but also can sense meagre electric pulse from other fishes. How it detects and localizes an object that has been mathematically developed as an algorithm in the next section.

3 Mathematical development of FEO as optimization tool

Fish electrolocation optimization has been mathematically developed through logical expression related to the conceptual electro-fish. The fish searches its food item by sensing feeble electric wave and also analyzing projected electric image. It simultaneously emanates electric pulse and senses feeble electrical wave. Feeble electrical wave is sensed in narrow range, whereas projected electric image gives the vision of surrounding environment. Sometimes it acts like elephant nose fish and sometimes it acts like shark. It toggles between active and passive electrolocation. This role reversal is linked

with the capacitance and distance value owing to toggle switch judgment. Resistance value, discussed in the previous section as a judgmental issue of elephant nose fish, has not been taken into consideration. This is done to restrict the list of parameters of this developed technique. However, FEO algorithm works through range discrimination/development, electric pulse calculation from slope analysis, distance calculation from the prey object; capacitance detection from waveform distortion value, toggle switch judgment through capacitance evaluation and finally electric organ discharge with tuned voltage amplitude. How these have been mathematically developed is described below sequentially.

3.1 Range development/discrimination

The developed electro-fish uses electrolocation dividing their search domain in a few zones. These zones have certain solution values with a definite difference between two consecutive values inside two limiting margins. Three kinds of range have been formulated based on difference between variable's maximum and minimum value, i.e., $diff$ and different multiplication factors. The $diff$ value is shown in (1). The $longrange$ is defined as the constant long search domain in (2). Constant terms $p1_l$ $p2_l$ and $p3_l$ are multiplied with the $diff$ value for defining the lower margin, gap value and higher margin, respectively. The domain consists of solution points between higher range, i.e., $diff \times p1_l$ and lower range, i.e., $diff \times p3_l$ with interval of $diff \times p2_l$ between any two consecutive values. Similarly, $shortrange$ is also restricted between $diff \times p1_s$ and $diff \times p3_s$ with a gap of $diff \times p2_s$ between two consecutive values in (3). The last solution domain which is used by the conceptual electro-fish is the $vshortrange$ mimicking the feeble electric pulse sen-
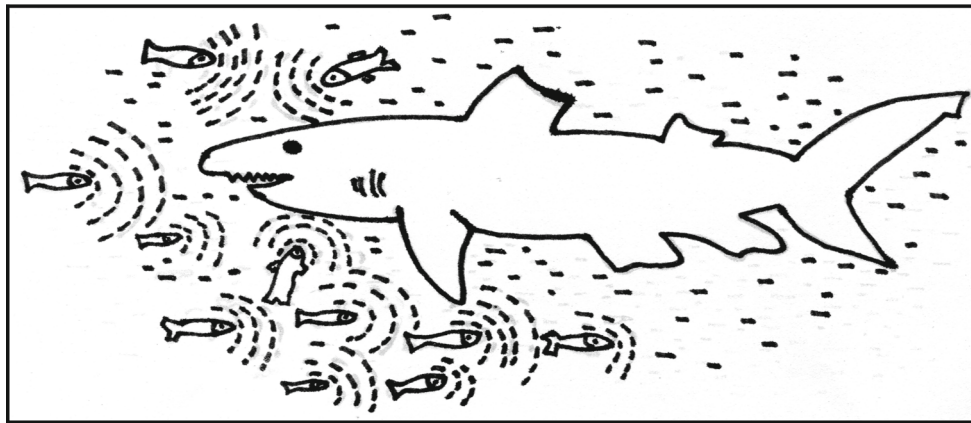
Fig. 2 Passive electrolocation of shark (*S. canicula*) (shark can judge meagre electric pulse coming out due to the muscular organism of small fishes. The *dotted circular lines* fencing the small fishes define that weak electric wave. Shark in the *picture* fetches its prey by sensing those electric signals)

sation of shark. This range is constricted within $\frac{x^{\min}}{diff}$ and $\frac{x^{\max}}{diff}$ with an interval gap of vs between two consecutive values in (4).

$$diff = x^{\max} - x^{\min} \tag{1}$$

$$longrange = \{diff \times p1_l : diff \times p2_l : diff \times p3_l\} \tag{2}$$

$$shortrange = \{diff \times p1_s : diff \times p2_s : diff \times p3_s\} \tag{3}$$

$$vshortrange = \left\{ \frac{x^{\min}}{diff} : vs : \frac{x^{\max}}{diff} \right\} \tag{4}$$

3.2 Electric pulse calculation from *longrange* and *slope* analysis

The electric pulse, i.e., $elec^{pulse}$ is calculated depending on projected image slope and developed longrange. The slope of the projected image is considered as the absolute difference between the best and worst found individual value amongst population at iteration t in (5). The value of $elec^{pulse}$ is important for formation of electric organ discharge by the conceptual electro-fish discussed later. However, the $elec^{pulse}$ is mathematically developed as a product of two terms in (6). The first term is chosen by taking a random value from longrange. The other term is considered as the ratio of slope and addition of two terms.

$$slope = \left| x_{best}^t - x_{worst}^t \right|$$
$$S = randperm \{length(longrange)\}$$
$$n \in S \tag{5}$$

$$elec^{pulse} = longrange(n) \times \frac{slope}{[(slope^{const} \times rand) + x_{best}^t]} \tag{6}$$

However, the distance from the targeted object by the conceptual electro-fish need to be found just like calculation of $elec^{pulse}$. This has been detected below.

3.3 Distance calculation from the prey object

The obtained minimum and maximum objective function value for all the individuals amongst population is considered as $ch1$ and $ch2$ at first iteration in (7). The nearness value towards targeted prey or food object is conditionally evaluated in (8).

$$ch1 = \min \{f(x_i)\} \quad i = 1, \ldots \ldots, n$$
$$ch2 = \max \{f(x_i)\} \quad i = 1, \ldots \ldots, n \tag{7}$$

$$Distance = \begin{cases} \frac{g1}{1+g2|ch1-ch2|} & \text{if } x_{best}^t = ch1 \\ \frac{g1}{1+g2|x_{best}^t - ch1|} & \text{otherwise} \end{cases} \tag{8}$$

After calculation of distance value, the conceptual electro-fish tries to find out the capacitance value of the targeted object.

3.4 Capacitance detection from waveform distortion value

Running capacitor value, i.e., cap^{run} is found by taking the standard deviation value amongst all the objective function values generated from every individual of population. The waveform distortion value is considered as the standard deviation value shown in (9). Capacitor upper limit, for judging whether the object is a food particle or not, is selected as *capint* and *caphover* for first and next iterations, respectively. This is shown in (10).

$$cap^{run} = \sigma \{f(x_1), f(x_2), \ldots, f(x_n)\} \tag{9}$$

$$capu = \begin{cases} capint & if \ iter = 1 \\ caphover & otherwise \end{cases} \qquad (10)$$

Conceptual electro-fish reaches the decision formation part when it gets the capacitance value.

3.5 Toggle switch judgment through capacitance evaluation

Toggle switch acts like changeover switch for the conceptual electro-fish's action. It is made '1' when running capacitor value is within the upper and lower limit and Distance value is less than setdist. It is made '0' if the above condition is not satisfied. This is shown in (11). The setdist value works like a variable throughout the iterations guiding the distance, i.e., nearness from the targeted object. This is manipulated as represented in (12).

$$toggle = \begin{cases} 1 & if \ capl < cap^{run} < capu \ and \ Distance < setdist \\ 0 & otherwise \end{cases}$$
$$(11)$$

$$setdist = Distance \ if \ capl < cap^{run} < capu$$
$$and \ Distance < setdist \qquad (12)$$

After the decision formation, conceptual electro-fish gears up its electric organ for emanation of next electrical wave.

3.6 Electric organ discharge with tuned voltage amplitude

The conceptual electro-fish generates new electric wave as xnew considering toggle switch operation in (13). If toggle switch shows '1' then xnew is generated incorporating $elec^{pulse}$ obtained earlier by doing slope analysis.

$$xnew = x_{best}^t + elec^{pulse} \times randn \ if \ toggle = 1 \quad (13)$$

When the **toggle** switch shows '0' then electrical wave is generated analyzing three kinds of probabilities. The three probabilities are **$prob^{sel}$**, **$prob^{div}$** and **$prob^{rng}$**, respectively. Probability of selection, i.e., **$prob^{sel}$** selects either randomization/diversification operation or localization operation. Randomization operation has taken place when the random variable **rand** is greater than **$prob^{sel}$**. Apart from

the above condition the conceptual electro-fish does localization. This is shown in (14).

$$if \ rand > prob^{sel}$$
$$do : randomization/diversification$$
$$else$$
$$do : localization$$
$$end \qquad (14)$$

3.6.1 Divergence operation

Divergence operation helps to create new electrical wave governed by the **$prob^{div}$**. The new electrical wave is generated randomly by the conceptual electro-fish in the whole search region. This is nothing but the randomization concept described in (15).

$$xnew = \begin{cases} x^{\min} + diff \times rand_i & if \ rand > prob^{div} \\ x^{\min} + diff \times rand_i^j & otherwise \end{cases}$$
$$(15)$$

3.6.2 Localization operation

Under the localization operation the imagined electro-fish emanates electric wave around the best found individual, i.e., x_{best}^t at iteration t. It does that with the influence of $prob^{rng}$. The two terms $c2$ and $c3$ are taken randomly from the set of shortrange and $vshortrange$, respectively. How those terms have been chosen is shown in (16) and (17), respectively. When the random variable $rand$ is greater than $prob^{rng}$, the new electrical wave xnew is formulated in vicinity of $c2$ and $c3$ shown in (18). It is also shown in (18) that if the random variable $rand$ is not greater than $prob^{rng}$, new electrical wave is generated around the x_{best}^t taking $elec^{pulse}$ and $c3$ into consideration.

$$m1 = randperm(length(shortrange))$$
$$h1 \in m1$$
$$c2 = shortrange(h1) \qquad (16)$$

$$m2 = randperm(length(vshortrange))$$
$$h2 \in m2$$
$$c3 = shortrange(h2) \qquad (17)$$

$$xnew = \begin{bmatrix} \begin{cases} x_{best}^t - c2 + 2 \times c2 \times rand & for \ j = 2, \ldots, pop - 1 \\ x_{best}^t + c3 & for \ j = 1 \end{cases} & if \ rand > prob^{rng} \\ \begin{cases} x_{best}^t + elec^{pulse} \times randn & for \ j = 2, \ldots, pop - 1 \\ x_{best}^t + c3 & for \ j = 1 \end{cases} & otherwise \end{bmatrix} \qquad (18)$$

How this newly developed FEO works as a meta-heuristic algorithm is written in the next section.

3.7 Algorithmic steps of fish electrolocation optimization (FEO)

Fish electrolocation optimization works through a few steps. These have been described below.

Step 1. Initialize $longrange$, $shortrange$ and $vshortrange$

Step 2. Set $capint$, $caphover$, $capl$ and $setdist$

Step 3. Set maximum iteration number

Step 4. Select three probabilities $prob^{div}$, $prob^{rng}$ and $prob^{sel}$

Step 5. Generate electrical wave randomly in the total search domain

Step 6. Analyze the objective function values tocalculate $slope$, cap^{run} and x_{best}^t and x_{worst}^t.

Step 7. Determine the $elec^{pulse}$ value and $Distance$ or nearness value from the targeted object.

Step 8. Do $toggle$ switch judgement and evaluate $setdist$ value

Step 9. If $toggle$ switch shows '1' then do $longrange$ operation otherwise consider $shortrange$ and $vshortrange$ for doing diversification and localization operation based on three mentioned probabilities to generate electrical wave.

Step 10. Repeat steps 6 to 9 until the convergence criterion is met or maximum iteration number is reached otherwise stop iteration process.

4 Fish electrolocation optimization (FEO) implementation and evaluation

Fish electrolocation optimization could not have been an optimization algorithm without proper testing. In this section, this testing has been shown with different kinds of benchmark functions. The developed meta-heuristic technique is simulated in the technical software MATLAB® 7.0. The code is written in Intel Core i3 2.4 GHz personal computer. While doing simulation it has been observed that FEO has two types of parameters. One is generalized parameters which remain

Table 1 Generalized parameter of FEO

No. of electro-fish (pop)	8
vs	10^{-2}
$slope^{const}$	1.8
$g1$	10^4
$g2$	10^2
$setdist$	10^6
$tolerance$	10^3
$capl$	0
$caphover$	10^5
$capint$	10^8

more or less same irrespective of nature of the problem. The other one is objective function-dependent parameters. Thirteen different kinds of objective functions have been chosen considering the dimension of the problem for examining the effectiveness of FEO. The best values are achieved by fixing the generalized parameters and problem-dependent parameters to certain values. These are shown in Tables 1 and 2, respectively, for the chosen benchmark functions.

The performance of the developed FEO has been compared with other well-known algorithms, viz. Real coded genetic algorithm (RCGA), accelerated particle swarm optimization (ACPSO), particle swarm optimization (PSO) and harmony search (HS). The study has been done for 100 time single independent run for every soft computing technique. These selected techniques apart from FEO have been implemented with population 50 and maximum number of iteration as 5000. The proposed algorithm FEO has also been implemented with the same maximum iteration number. Population, i.e., the number of electro-fish, has been considered here as eight. It would be a reasonable question as to why the population has not been fixed at 50 like others. The answer is that the performance of FEO is good at a certain range of population, but deteriorates with increment of population from that range. This has been demonstrated through extensive study done on *Schwefel* function in later section. However, *Camel Back* (CB), *Martin & Gaddy* (MG), two-dimensional ($d = 2$) *Rosenbrock* (Rk), *Branin* (Br), *Goldstein & Price* (GP), four-dimensional ($d = 4$) *Shekel* (Shk), six-dimensional ($d = 6$) *Sphere* (Sp), *B*2, *Shubert* (Su), *Rastrigin* (Ra), two- and five-dimensional ($d = 2, 5$) *Schwefel* (Sch) and two-dimensional ($d = 2$) *Michaelwicz* (Mi) functions have been considered for implementation purpose (Hedar 2005). The range and expression of these chosen functions are shown in Table 3.

Real coded genetic algorithm (RCGA) has been applied in two variant forms. Two variations have been considered on the basis of mutation phenomena, whereas crossover operation is taken as in literature (Wright 1991). One variant, i.e.,

Table 2 Objective function or problem-dependent parameters of FEO

Test functions	*longrange* multiplying factor			*shortrange* multiplying factor			$prob^{sel}$	$prob^{div}$	$prob^{rng}$
	$p1_l$	$p2_l$	$p3_l$	$p1_s$	$p2_s$	$p3_s$			
Camel back (CB)	10^{-2}	5×10^{-3}	5×10^{-2}	10^{-3}	10^{-4}	10^{-2}	0.47	0.85	0.10
Martin & Gaddy (MG)	10^{-1}	5×10^{-2}	5×10^{-1}	10^{-5}	10^{-6}	10^{-4}	0.50	0.01	0.01
Rosenbrock (Rk)	10^{-1}	5×10^{-2}	5×10^{-1}	10^{-5}	10^{-6}	10^{-4}	0.50	0.01	0.01
B2	10^{-2}	5×10^{-3}	5×10^{-2}	10^{-5}	10^{-6}	10^{-4}	0.50	0.05	0.10
Branin (Br)	10^{-2}	5×10^{-3}	5×10^{-2}	10^{-4}	10^{-5}	10^{-3}	0.79	0.50	0.10
Goldstein & Price (GP)	10^{-2}	5×10^{-3}	5×10^{-2}	10^{-4}	10^{-5}	10^{-3}	0.65	0.85	0.10
Shekel (Shk)	10^{-2}	5×10^{-3}	5×10^{-2}	10^{-4}	10^{-5}	10^{-3}	0.38	0.50	0.10
Sphere (Sp)	10^{-2}	5×10^{-3}	5×10^{-2}	10^{-5}	10^{-6}	10^{-4}	0.50	0.05	0.10
Rastrigin (Ra)	10^{-2}	5×10^{-3}	5×10^{-2}	10^{-4}	10^{-5}	10^{-3}	0.33	0.50	0.10
Schwefel (Sch)	10^{-2}	5×10^{-3}	5×10^{-2}	10^{-4}	10^{-5}	10^{-3}	0.75	0.32	0.15
Michalewicz (Mi)	10^{-1}	5×10^{-2}	5×10^{-1}	10^{-3}	10^{-4}	10^{-3}	0.79	0.50	0.21

Table 3 The selected test bed functions with chosen range for doing minimization

CB	$x_1, x_2 \in (-3, 3)$	$f(x_1, x_2) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$
MG	$x_d \in (-20, 20)$	$f(x_1, x_2) = (x_1 - x_2)^2 + \left[\frac{x_1+x_2-10}{3}\right]^2$
Rk	$x_d \in (-5, 5)$	$f(x) = \sum_{i=1}^{d}\left[100(x_{i+1} - x_i)^2 + (1 - x_i)^2\right]$
Br	$x_1, x_2 \in (-15, 15)$	$f(x_1, x_2) = a(x_2 - bx_1^2 + cx_1 - d)^2 + e(1 - f)\cos(x_1) + e$
		$a = 1, b = \frac{5.1}{4\pi^2}, c = \frac{5}{\pi}, d = 6, e = 10, f = \frac{1}{8\pi}$
GP	$x_1, x_2 \in (-2, 2)$	$f(x_1, x_2) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right] \times$
		$\left[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right]$
Shk	$x_j \in (0, 10)$	$f(x) = -\sum_{i=1}^{d}\left\{\sum_{j=1}^{d}\left[(x_j - a_{ij})^2\right] + c_j\right\}^{-1}$ $a_{ij} = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \end{bmatrix}$ $c_j = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \end{bmatrix}$
Sp	$x_i \in (-5.12, 5.12)$	$f(x) = \sum_{i=1}^{d} x_i^2$
B2	$x_i \in (-100, 100)$	$f(x_1, x_2) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$
Su	$x_1, x_2 \in (-5.12, 5.12)$	$f(x_1, x_2) = -\sum_{i=1}^{d} i\cos((i+1)x_1 + 1)\sum_{i=1}^{d} i\cos((i+1)x_2 + 1)$
Ra	$x_i \in (-5.12, 5.12)$	$f(x) = 10d + \sum_{i=1}^{d}\left[x_i^2 - 10\cos(2\pi x_i)\right]$
Sch	$x_i \in (-500, 500)$	$f(x) = \sum_{i=1}^{d}\left[-x_i\sin(\sqrt{|x_i|})\right]$
Mi	$m = 10, x_i \in (0, \pi)$	$f(x) = -\sum_{i=1}^{d}\sin(x_i)\left[\sin(\frac{ix_i^2}{\pi})\right]^{2m}$

RCGA[1] is considered by doing mutation according to (19). Another one, i.e., RCGA[2] is considered as (20).

$if\ rand > P^{mut}$

$if\ fix(2 \times rand) > 0$

$k = floor(pop \times rand) + 1$

$x_i^k = x_i^{\min} + diff \times rand$

end

end (19)

$if\ rand > P^{mut}$

$k = floor(pop \times rand) + 1$

$if\ rand > 0.5$

$x_i^k = x_i^k + \left|x_i^{\max} - x_i^k\right| \times rand$

$else$

$x_i^k = x_i^k - \left|x_i^k - x_i^{\min}\right| \times rand$

end

end (20)

The probability of mutation and crossover, i.e., P^{mut} and P^{cross} for RCGA[1] and RCGA[2] are taken to be considered as 0.01 and 0.98 and 0.05 and 0.88, respectively. APSO and PSO have been implemented according to (21) and (22) where x_i^{new}, x_i^{best} and g^{best} define the newly calculated variable value, current and global best value, respectively (Wright 1991). The parameters $alpha$, γ and $beta$ values are chosen as 0.7, 8 and 2, respectively, for ACPSO. On the other hand, parameters $alpha$ and $beta$ for PSO have been selected as 1.8 and 1.7, respectively.

$$x_i^{new} = x_i + alpha^{\gamma} \times (rand - 0.5) + beta \times rand \\ \times (g^{best} - x_i) \tag{21}$$

$$x_i^{new} = x_i + alpha \times rand \times (x_i^{best} - x_i) + beta \times rand \\ \times (g^{best} - x_i) \tag{22}$$

Harmony search (HS) has also been applied with harmony memory accept rate, pitch adjusting rate, harmony size and pitch range as 0.85, 0.7, 20 and [100, . . . , 100], respectively (Yang 2010a, b). All the above-mentioned parameters for different kinds of algorithms have been chosen after several trials. This is done for doing comparative study with the proposed FEO. The parameters of FEO have also been selected by doing several trials shown earlier in Tables 1 and 2. The chosen evolutionary algorithms are implemented on above-mentioned objective functions with convergence criterion shown in (23). That has been implemented for 100 times single independent run said earlier. Mean number of function evaluation (NF), standard deviation (dev) and percentage of success (% S) are considered as the evaluation criteria. The defined maximum iteration number is considered as 5000.

$$\left| f - f^* \right| < \varepsilon_1 f + \varepsilon_2 \tag{23}$$

The constants ε_1 and ε_2 in (23) are selected as 10^{-4}. The term f and f^* define the optimal value achieved and global optimal value, respectively, as shown in (23). The comparative study amongst the algorithms is shown in Tables 4, 5, 6 and 7, respectively.

It has been observed from Tables 4, 5, 6 and 7 that proposed FEO outperforms other algorithms in 9 amongst the chosen 13 minimization problems in respect of mean number of function evaluation and percentage of success. It is shown in Table 4 that proposed FEO outperforms RCGA[1] and RCGA[2] in large difference but remains slightly ahead to ACPSO, PSO and HS for *Camel Back* function. Fish electrolocation optimization takes 184 mean number of function evaluation with standard deviation of 100 for *Martin & Gaddy* function. RCGA[1], RCGA[2], ACPSO, PSO and HS take 65,095, 8456, 420, 420 and 1909 mean number of function evaluation with standard deviation of 19,964, 3562, 58, 85 and 970,

respectively. It has also been shown that for *Rosenbrock* function FEO outperforms other chosen algorithms with mean number of function evaluation and standard deviation as 337 and 227, respectively. It is shown in Table 5 that in case of *Branin* function ACPSO and PSO perform better than FEO. RCGA[1], RCGA[2] and HS takes more number of function evaluation than FEO to reach the desired convergence for *Branin* function. In the case of *Goldstein & Price* function, PSO outperforms all the other evolutionary algorithms including FEO. It has also been observed from Table 5 that FEO reaches desired convergence for *Shekel* function with mean number of function evaluation and standard deviation as 838 and 396, respectively. It fulfills the convergence criteria with 100 % success. RCGA[1], RCGA[2], ACPSO, PSO and HS achieve that desired convergence with 80, 18, 29, 36 and 45 % success, respectively. The other chosen soft computing techniques viz. RCGA[1], RCGA[2], ACPSO, PSO and HS take mean number of function evaluation and standard deviation as 71,269 and 25,519; 237,808 and 112,495; 15,753 and 11,964; 7373 and 22,056 and 10,380 and 4068, respectively. On the other hand, it is shown in Table 6 that FEO outperforms other chosen algorithms in respect of mean number of function evaluation and percentage of success for *Sphere*, *B*2 and *Shubert* function, respectively. It is shown in Table 7 that FEO takes less number of function evaluations in *Rastrigin* and five-dimensional *Schwefel* function. On the other hand, FEO takes more number of function evaluations for two-dimensional *Michalewicz* and *Schwefel* function. Interestingly it has been observed that RCGA[1] does not achieve the said convergence for two- and five-dimensional *Schwefel* function. RCGA[2] remains unsuccessful for doing minimization in five-dimensional *Schwefel* and two-dimensional *Michalewicz* function. Harmony search (HS) fails to achieve set convergence criteria similar like RCGA[1]. It has been observed that proposed FEO has achieved 100 % success in both two- and five-dimensional *Schwefel* function. APSO and PSO have achieved 49 and 85 and 3 and 13 % success for two- and five-dimensional *Schwefel* function, respectively. It can be stated that FEO outperforms in 10 amongst 13 minimization problems if two-dimensional *Schwefel* function is considered. This new win is on the basis of percentage of success as an addition to nine win in comparison to other algorithms stated earlier. From this comparative study it has been typically observed that chosen soft computing techniques apart from FEO severely stumble to get desired convergence for minimization of *Schwefel* function. On the other hand, only generalized parameters of FEO have been fixed for doing all the comparative study. Problem-dependent parameters have been suitably changed for getting the above shown results in Tables 4, 5, 6 and 7, respectively. A special study has been done on *Schwefel* function for that chosen solution range described in Table 3. This is done to make the fuzziness of generalized and problem-dependent

Table 4 Comparative study amongst soft computing techniques

Soft computing techniques	Camel Back (CB)		Martin & Gaddy (MG)		Rosenbrock (Rk) ($d = 2$)	
	Mean NF $\pm$ dev	% S	Mean NF $\pm$ dev	% S	Mean NF $\pm$ dev	% S
RCGA[1]	18,094 $\pm$ 6468	100	65,095 $\pm$ 19,964	100	362,102 $\pm$ 84,895	14
RCGA[2]	4946 $\pm$ 1985	100	8456 $\pm$ 3562	100	135,895 $\pm$ 97,465	100
ACPSO	309 $\pm$ 60	90	420 $\pm$ 58	100	2043 $\pm$ 1555	100
PSO	290 $\pm$ 60	100	420 $\pm$ 85	100	3131$\pm$1738	100
HS	708 $\pm$ 262	100	1909 $\pm$ 970	100	23,669 $\pm$ 18,245	100
Proposed FEO	265 $\pm$114	100	184 $\pm$ 100	100	337 $\pm$ 227	100

Table 5 Comparative study amongst soft computing techniques contd

Soft computing techniques	Branin (Br)		Goldstein & Price (GP)		Shekel (Shk) ($d = 4$)	
	Mean NF $\pm$ dev	% S	Mean NF $\pm$ dev	% S	Mean NF $\pm$ dev	% S
RCGA[1]	15726 $\pm$ 8575	100	56,230 $\pm$ 49,422	65	71,269 $\pm$ 25,519	80
RCGA[2]	7776 $\pm$ 5881	100	15,847 $\pm$ 8439	100	237,808 $\pm$ 112,495	18
ACPSO	464 $\pm$ 94	94	1212 $\pm$ 858	98	15753 $\pm$ 11,964	29
PSO	500 $\pm$ 784	100	378 $\pm$ 63	99	7373 $\pm$ 22,056	36
HS	1648 $\pm$ 638	100	1870 $\pm$ 1337	86	10380 $\pm$ 4068	45
Proposed FEO	713 $\pm$ 277	100	592 $\pm$ 290	100	838 $\pm$ 396	100

Table 6 Comparative study amongst soft computing techniques contd

Soft computing techniques	Sphere (Sp) ($d = 6$)		B2		Shubert (Su)	
	Mean NF $\pm$ dev	% S	Mean NF $\pm$ dev	% S	Mean NF $\pm$ dev	% S
RCGA[1]	37,144 $\pm$ 8377	100	26,549 $\pm$ 13,880	100	166,829 $\pm$ 128,473	86
RCGA[2]	238,686 $\pm$ 131695	30	209,478 $\pm$ 149,314	68	199,556 $\pm$ 136,587	57
ACPSO	8772 $\pm$ 8378	100	779 $\pm$ 255	99	550 $\pm$ 159	95
PSO	1876 $\pm$ 2375	75	559 $\pm$ 69	100	631 $\pm$ 180	100
HS	17699 $\pm$ 6550	100	6867 $\pm$ 3636	100	1167 $\pm$ 443	100
Proposed FEO	363 $\pm$ 257	100	374 $\pm$ 144	100	476 $\pm$ 243	100

parameter clear. This has been briefly discussed in the next segment.

5 More implementation of FEO on *Schwefel* function

Fish electrolocation optimization has been extensively applied on *Schwefel* function. The optimization tool FEO has been implemented on this said function for various dimensions. It has been applied by varying number of electro-fish or population for 100 time single independent run, maximum iteration number as 5000 and convergence criteria shown in (22). This is shown in Tables 8 and 9, respectively, starting with lower

dimension 10 and ending with higher dimension 200. It has been noticed that FEO achieves 100 % success for all the test conditions. That is why it is not shown in Tables 8 and 9, respectively.

It has been observed from Tables 8 and 9 that computation time increases with the increase in dimension number. Computation time varies with population, but not like the just said variation with dimension number. It can be observed from Table 8 that for dimension 60, the computation time in second for 5, 10, 20, 40, 60, 80, 100, 150, 200 and 300 number of electro-fish are 11.971, 10.491, 7.867, 7.411, 6.913, 8.676, 8.938, 9.838, 10.329 and 11.929, respectively. This can be comprehended that when the number of electro-fish is 60 the run time is least. On the contrary, when it comes

Table 7 Comparative study amongst soft computing techniques contd

Soft computing techniques	Rastrigin (Ra) Mean NF ± dev	% S	Schwefel (Sch) (d = 2) Mean NF ± dev	% S	Schwefel (Sch) (d = 5) Mean NF ± dev	% S	Michalewicz (Mi) (d = 2) Mean NF ± dev	% S
RCGA[1]	23,594 ± 58,328	75	Unsuccessful		Unsuccessful		22,683 ± 9493	100
RCGA[2]	39,784 ± 34,271	100	289,048 ± 118,393	7	Unsuccessful		Unsuccessful	
ACPSO	2278 ± 1804	35	314 ± 61	49	683 ± 115	3	382 ± 101	85
PSO	1068±846	98	1129 ± 4273	85	2238 ± 964	13	306 ± 75	100
HS	1750 ± 644	51	Unsuccessful		Unsuccessful		613 ± 256	100
Proposed FEO	624 ± 380	100	592±290	100	602 ± 384	100	451 ± 334	100

Table 8 FEO implementation on *Schwefel* function for different dimension

No. of electro-fish (pop)	d = 10 Mean NF ± dev	Run time in s	d = 20 Mean NF ± dev	Run time in s	d = 40 Mean NF ± dev	Run time in s	d = 60 Mean NF ± dev	Run time in s
5	583 ± 385	3.28	660 ± 510	5.077	738 ± 528	8.91	722 ± 543	11.971
10	793 ± 427	2.85	922 ± 612	4.320	1071 ± 739	7.60	1081 ± 651	10.491
20	1194 ± 696	3.038	1177 ± 727	3.749	1349 ± 675	6.223	1280 ± 861	7.867
40	1215 ± 748	2.692	1741 ± 840	4.205	1788 ± 1006	6.002	1695 ± 1151	7.411
60	1567 ± 1051	3.217	1897 ± 1192	4.146	2188 ± 1318	6.4	1839 ± 1208	6.913
80	1640 ± 955	3.430	2314 ± 1390	4.771	2470 ± 1509	6.658	2518 ± 1531	8.676
100	1680 ± 874	3.372	2663 ± 1573	5.223	2802 ± 1747	7.295	2722 ± 1909	8.938
150	2070 ± 1000	4.076	3172 ± 1700	5.929	3111 ± 1914	7.6	3180 ± 2175	9.838
200	2614 ± 1295	5.154	3428 ± 2120	6.265	3818 ± 2332	9.005	2175 ± 2383	10.329
300	3060 ± 1474	5.771	4275 ± 2543	7.986	4749 ± 3122	11.192	4074 ± 2500	11.929

Table 9 FEO implementation on *Schwefel* function for different dimension contd

No. of electro-fish (pop)	d = 80		d = 100		d = 150		d = 200	
	Mean NF ± dev	Runtimeins	Mean NF ± dev	Run time in s	Mean NF ± dev	Run time in s	Mean NF ± dev	Run time in s
5	874 ± 765	18.362	938 ± 1102	24.03	1038 ± 1558	37.662	1046 ± 1345	48.977
10	919 ± 846	11.166	1045 ± 953	15.289	1160 ± 1332	23.723	1206 ± 1318	31.732
20	1292 ± 1025	9.806	1437 ± 1172	13.022	1598 ± 1488	20.068	1660 ± 1659	26.627
40	1532 ± 1103	8.222	1779 ± 1373	11.124	1888 ± 1630	16.240	2085 ± 2062	22.706
60	1972 ± 1428	8.965	2107 ± 1609	11.242	2106 ± 1599	15.406	2051 ± 1911	18.761
80	2152 ± 1460	9.016	2193 ± 1583	10.836	2180 ± 1563	14.347	2725 ± 2237	22.625
100	2810 ± 1897	11.2	2719 ± 2058	12.501	2673 ± 1914	16.661	2823 ± 2421	21.985
150	2688 ± 1754	10.105	2892 ± 2411	12.458	2737 ± 2216	15.888	3106 ± 2427	22.519
200	3790 ± 2730	13.366	3790 ± 2563	15.788	3124 ± 2448	17.504	3756 ± 2731	26.294
300	3852 ± 2542	13.939	3882 ± 2917	16.098	3927 ± 2506	22.380	4059 ± 2578	29.709

the case of mean number of function evaluation then FEO reaches 100 % success by taking mean number of function evaluation and standard deviation as 722 and 543; 1081 and 651; 1280 and 861; 1695 and 1151; 1839 and 1208; 2518 and 1531; 2722 and 1909; 3180 and 2175; 2175 and 2383 and 4074 and 2500 for 5, 10, 20, 40, 60, 80, 100, 150, 200 and 300 number of conceptual electro-fish, respectively. It can be comprehended that FEO takes lesser number of function evaluations as the population or number of fish decreases and vice versa. There is only one defaulter at population 150. This trend of taking less number of function evaluations at lesser number of population remains more or less constant for the entire tested dimension shown in Tables 8 and 9, respectively. There is one defaulter at population number 150 for 60- and 80-dimensional *Schwefel* function. This outcome of mismatch or default value is due to stochastic nature of FEO. One thing is common in entire tests shown in Tables 8 and 9, i.e., FEO achieves 100 % success to get the desired convergence. The convergence criterion has been made more stringent by considering number of objective function evaluation as a limit. This is done to have variation in percentage of success. This study is shown in Tables 10 and 11, respectively.

It can be observed from Tables 10 and 11 that maximum number of function evaluation has been varied from 4000 to 500 as an extra convergence criteria to that of (22). It has been observed from Table 10 that FEO reaches 100 % success for all its chosen population when the maximum number of function evaluation is 4000. The set limit of function evaluation is 3000. Fish electrolocation optimization gets 90, 94, 94, 83, 82, 70, 65, 62, 56, 54 and 42 percentage of success for chosen population number of 5, 10, 20, 40, 60, 80, 100, 150, 180, 200 and 300, respectively. This trend of decreasing percentage of success with the increase in population number continues up to 500 maximum number of function evaluation. Percentage of success (%S) increases with the increase of population number. There are some defaulters also in that said pattern for maximum limit of 2000, 1000 and 500, respectively. Maximum number of function evaluation is set at 1000. The percentage of success increases from 21 to 27, from 11 to 19 and from 9 to 12 with the increase of population number from 40 to 60, from 80 to 100 and from 180 to 200, respectively. It is also shown in Table 11 that FEO becomes unsuccessful to achieve the convergence coupled with maximum number of function evaluation as 500 for population number of 180, 200 and 300, respectively. It has also been noticed that computation time is highest at population number 5. That gradually decreases up to population number 60, 60, 100, 60 and 60, respectively. Then it increases up to 300 for maximum number of function evaluation of 4000, 3000, 2000, 1000 and 500, respectively. However, from the total above analysis it can be concluded that percentage of success is higher with the lower number of electro-fish. Mean number of function

Table 10 Study of implementation of FEO on *Schwefel* function ($d = 200$) by varying maximum number of function evaluation as an extra convergence criterion

No. of electro-fish (pop)	Max function evaluation = 4000			Max function evaluation = 3000			Max function evaluation = 2000		
	Mean NF ± dev	% S	Run time in s	Mean NF ± dev	% S	Run time in s	Mean NF ± dev	% S	Run time (s)
5	759 ± 629	100	46.841	584 ± 462	90	49.049	495 ± 337	90	48.293
10	1070 ± 902	100	37.629	799 ± 591	94	28.694	762 ± 379	87	32.869
20	1237 ± 757	100	24.986	1140 ± 575	94	24.074	922 ± 410	79	23.716
40	1465 ± 894	100	19.690	1312 ± 622	83	22.785	1095 ± 449	65	21.118
60	1620 ± 891	100	17.489	1554 ± 782	82	18.772	1072 ± 548	53	20.294
80	1820 ± 1056	100	19.479	1632 ± 731	70	20.507	1109 ± 509	53	19.124
100	1983 ± 938	100	19.684	1675 ± 827	65	21.200	1156 ± 523	50	17.241
150	2194 ± 942	100	26.914	1722 ± 696	62	21.453	1267 ± 475	38	20.756
180	2076 ± 1047	100	21.097	1738 ± 743	56	21.404	1415 ± 450	29	25.837
200	2168 ± 979	100	23.408	1814 ± 730	54	23.507	1322 ± 476	36	27.069
300	2348 ± 818	100	32.790	1921 ± 581	42	30.323	1322 ± 354	22	32.360

evaluation is lower with the lower number of electro-fish. Run time is lower with the moderate number of electro-fish. This analysis more or less justifies the cause of fixing population number at lower value, i.e., 8 for doing comparative study. Though study of variation in population number and also set convergence condition gives clarification of inner strength of FEO but can't accommodate the entire parameters of this technique. There are problem-dependent parameters too apart from generalized parameters. That is number of electro-fish or population. These parameters include different range discrimination constants and probabilities. The study of variation of *shortrange* and *longrange* constants are described in the next section.

5.1 Variation in *shortrange* constants

shortrange constants $p1_s$, $p2_s$ and $p3_s$ are varied by fixing the *longrange* constants at 10^{-2}, 5×10^{-3} and 5×10^{-2}, respectively. The mean number of function evaluation (NF) and standard deviation (dev) and computation time are calculated for population number 5, 20, 60, 100, 150, 200 and 300, respectively, in Tables 12 and 13. This is done for 200-dimensional *Schwefel* function with selected Table 3. This study is done without disturbing the other parameters of FEO shown in Table solution domain as depicted in 1 and Table 2, respectively.

It has been observed from Table 12 that the run time in seconds, for a particular population number 5, gradually increases from 23.876 to 43.848. That has been observed with the decrease of shortrange constants value from 10^{-2}, 10^{-3} and 10^{-1}–10^{-3}, 10^{-4} and 10^{-2}. Then it decreases to 40.584 when *shortrange* constants values are 10^{-4}, 10^{-5} and 10^{-3}, respectively. It can be noticed from Table 13 that run time in seconds again decreases to 25.553 from the just said run time in Table 12. It remains more or less constant within 24.738 and 25.553 with the decrease in *shortrange* constants value. Not only the runtime is restricted to a small span, but also the mean number of function evaluation is observed to be restricted between 529 and 541. Fish electrolocation optimization achieves better results when the *shortrange* constant values exceed a certain boundary in comparison to sticking with a certain value. That can be observed from Tables 12 and 13, respectively. However, this study is done considering the *longrange* constants at a fixed value. The variation of the constant values regarding *longrange* is considered for study purpose in the very next segment.

5.2 Variation in *longrange* constants

The *longrange* constants, i.e., $p1_l$, $p2_l$ and $p3_l$ has been varied while the *shortrange* constants are kept fixed at 10^{-4}, 10^{-5} and 10^{-3}, respectively. It is done for implementation of FEO on 200-dimensional *Schwefel* function. The computed

Table 11 Study of implementation of FEO on *Schwefel* function ($d = 200$) by varying maximum number of function evaluation as an extra convergence criterion contd

No. of electro-fish (pop)	Max function evaluation = 1000			Max function evaluation = 500		
	Mean NF ± dev	% S	Run time in s	Mean NF ± dev	% S	Run time (s)
5	480 ± 229	82	55.224	347 ± 116	37	39.179
10	581 ± 243	61	29.254	332 ± 108	27	34.894
20	582 ± 254	43	25.028	360 ± 118	23	25.524
40	577 ± 233	21	24.896	375 ± 124	8	20.097
60	580 ± 210	27	17.553	340 ± 116	9	18.542
80	603 ± 287	11	22.090	390 ± 90	8	22.014
100	726 ± 199	19	20.050	425 ± 95	4	19.401
150	613 ± 156	11	22.377	450 ± 0	1	22.717
180	740 ± 60	9	23.515		Unsuccessful	
200	800 ± 147	12	21.580		Unsuccessful	
300	900 ± 0	4	29.113		Unsuccessful	

Table 12 Study of application of FEO on *Schwefel* function ($d = 200$) by varying the *shortrange* constants ($p1_s$, $p2_s$, $p3_s$) fixing *longrange* constants ($p1_l$, $p2_l$, $p3_l$) at (10^{-2}, 5×10^{-3}, 5×10^{-2})

No. of electro-fish (pop)	(10^{-2}, 10^{-3}, 10^{-1})		(10^{-3}, 10^{-4}, 10^{-2})		(10^{-4}, 10^{-5}, 10^{-3})	
	Mean NF±dev	Run time in s	Mean NF±dev	Run time in s	Mean NF±dev	Run time (s)
5	500 ± 272	23.876	928 ± 1288	43.848	861 ± 1199	40.584
20	1003 ± 563	16.437	1468 ± 1705	23.904	1279 ± 1138	20.622
60	1809 ±1084	16.840	1803 ± 1664	16.545	2242 ± 2015	20.716
100	2169 ± 1477	17.192	2251 ± 1630	18.362	2529 ± 1651	19.914
150	2943 ± 1735	21.477	2800 ± 1785	20.477	2865 ± 1884	20.821
200	3494 ± 1720	24.914	3124 ± 2142	22.362	3152 ± 2205	22.514
300	4269 ± 2548	31.190	4095 ± 3207	30.200	4233 ± 2808	30.878

Table 13 Study of application of FEO on *Schwefel* function ($d = 200$) by varying the *shortrange* constants ($p1_s$, $p2_s$, $p3_s$) fixing *longrange* constants ($p1_l$, $p2_l$, $p3_l$) at (10^{-2}, 5×10^{-3}, 5×10^{-2}) contd

No. of electro-fish (pop)	(10^{-5}, 10^{-6}, 10^{-4})		(10^{-21}, 10^{-22}, 10^{-20})		(10^{-99}, 10^{-100}, 10^{-98})	
	Mean NF ± dev	Run time in s	Mean NF ± dev	Run time in s	Mean NF ± dev	Run time (s)
5	541 ± 284	25.553	523 ± 293	24.738	529 ± 298	25.003
20	1160 ± 671	18.718	1094 ± 618	17.688	1125 ± 648	18.457
60	1816 ± 1150	16.925	1924 ± 1100	17.740	1815 ± 985	16.769
100	2473 ± 1610	19.304	2388 ± 1604	18.854	2282 ± 1328	18.284
150	3222 ± 2143	23.326	2809 ± 1799	20.897	2724 ± 1695	20.596
200	3512 ± 2189	24.780	3444 ± 1988	24.521	3680 ± 2416	26.950
300	4083 ± 2753	30.155	3627 ± 2480	28.149	3915 ± 2480	29.897

results are shown in Tables 14 and 15, respectively. The longrange. constants are varied for population number 5, 20, 60, 100, 150, 200 and 300, respectively.

It has been observed from Tables 14 and 15 that mean number of function evaluation (NF) and computation time changes with the decrement of *longrange* constant values. That said change is more in comparison to previously dis-

cussed changes observed in Tables 12 and 13, respectively. It can be observed from Tables 14 and 15 that for population number 100, run time in seconds and mean number of function evaluation are 18.914 and 2345; 23.493 and 2685; 88.108 and 11,022 and 422.243 and 49,812 for *longrange* constant values of 10^{-1}, 5×10^{-2} and 5×10^{-1}; 10^{-3}, 5×10^{-4} and 5×10^{-3}; 10^{-4}, 5×10^{-5} and 5×10^{-4} and

Table 14 Study of application of FEO on *Schwefel* function ($d = 200$) by varying the *longrange* constants ($p1_l$, $p2_l$, $p3_l$) fixing *shortrange* constants ($p1_s$, $p2_s$, $p3_s$) at (10^{-4}, 10^{-5}, 10^{-3})

No. of electro-fish (pop)	(10^{-1}, 5×10^{-2}, 5×10^{-1})			(10^{-3}, 5×10^{-4}, 5×10^{-3})		
	Mean NF ± dev	Run time in s	% S	Mean NF ± dev	Run time in s	% S
5	983 ± 1331	46.264	100	1116 ± 1073	53.140	100
20	1400 ± 1565	22.878	100	2580 ± 2210	42.958	100
60	2313 ± 2174	21.635	100	3040 ± 3368	29.998	100
100	2345 ± 1487	18.914	100	2685 ± 2106	23.493	100
150	3144 ± 2089	23.792	100	2959 ± 2478	24.693	100
200	3898 ± 2570	27.733	100	3086 ± 1979	24.957	100
300	4575 ± 4062	33.502	100	4080 ± 3202	33.565	100

Table 15 Study of application of FEO on *Schwefel* function ($d = 200$) by varying the *longrange* constants ($p1_l$, $p2_l$, $p3_l$) fixing *shortrange* constants ($p1_s$, $p2_s$, $p3_s$) at (10^{-4}, 10^{-5}, 10^{-3}) contd

No. of electro-fish (pop)	(10^{-4}, 5×10^{-5}, 5×10^{-4})			(10^{-5}, 5×10^{-6}, 5×10^{-5})		
	Mean NF ± dev	Run time in s	% S	Mean NF ± dev	Run time in s	% S
5	3593 ± 2433	170.263	100	4971 ± 3334	233.883	100
20	13,647 ± 14,245	220.940	100	35,526 ± 26333	908.186	68
60	12,687 ± 16,970	118.084	100	55,392 ± 53658	641.240	94
100	11,022 ± 9898	88.108	100	49,812 ± 48047	422.243	99
150	8227 ± 7845	61.881	100	54,709 ± 60463	381.211	100
200	11,260 ± 11,493	79.895	100	51,466 ± 63006	340.964	100
300	9513 ± 11,195	67.787	100	41,844 ± 56317	225.976	100

10^{-5}, 5×10^{-6} and 5×10^{-5}, respectively. It is also shown earlier in Table 9 that the run time in seconds and mean number of function evaluation are 21.985 and 2823 for longrange constant value of 10^{-2}, 5×10^{-3} and 5×10^{-2} at population number 100. The other selected population number more or less obeys this kind of drastic change. That is shown in Tables 14 and 15, respectively, with the changes of *longrange* constant. This said pattern of function evaluation and computation time describes that the change in *longrange* constant value in this way can be very sensitive to computation. Even the percentage of success value drops down to 68, 94 and 99 for population number 20, 60 and 100, respectively. That has been observed for the longrange constant are 10^{-5}, 5×10^{-6} and 5×10^{-5}, respectively. It can be stated from the study done on *Schwefel* function that *longrange* constants should be strictly restricted within a definite zone. That is to avoid bad results. On the other hand, *shortrange* constants analyzed in the previous section can be fine-tuned to have better results.

However, FEO has not been compared with well established technique differential evolution and simulated annealing. This has been done on eggcrate function in the next section.

6 Comparative study of fish electrolocation optimization with differential evolution and simulated annealing on eggcrate function

Fish electrolocation optimization has been compared with differential evolution (DE) and simulated annealing (SA) in this section. This extra comparative study has been done to show the effectiveness of FEO with other well established meta-heuristic techniques such as DE and SA. These stated algorithms have been implemented on two-dimensional eggcrate function shown in expression (24).

$$f(x, y) = x^2 + y^2 + 25((\sin x)^2 + (\sin y)^2) \qquad (24)$$

The range of eggcrate function has been considered here as $[-5, 5]$. The global minimum value of this function is at $(0, 0)$ for each variable and the global minimum value is 0. The convergence criterion for this comparative study has been considered same as earlier denoted expression (23). The number of population individual has been chosen as 30 for all the considered soft computing techniques. In this case, the number of population individuals has not been considered as eight as described earlier. The population number has been

considered here as stated thirty for the proposed technique to have similarity with other algorithms' implementation. The maximum iteration number of this comparative study has been chosen as the stopping criteria if the desired convergence has not been achieved. This has been chosen as 5000 for every soft computing technique. The study has been done for hundred time single independent run by changing the random number seed every time. Differential evolution has been implemented for two schemes on eggcrate function. The scheme I and scheme II regarding differential evolution varies on the principle of generation of mutation matrix. The genesis of mutation vector $\overrightarrow{V_m}$ for scheme I and scheme II have been shown in expressions (25) and (26), respectively.

$$\overrightarrow{V_m} = \overrightarrow{X_{r1,G}} + F_m.(\overrightarrow{X_{r2,G}} - \overrightarrow{X_{r3,G}}) \tag{25}$$

$$\overrightarrow{V_m} = \overrightarrow{X_{i,G}} + \lambda.(\overrightarrow{X_{best,G}} - \overrightarrow{X_{i,G}}) + F_m.(\overrightarrow{X_{r2,G}} - \overrightarrow{X_{r3,G}}) \tag{26}$$

The notations $\overrightarrow{X_{r1,G}}$, $\overrightarrow{X_{r2,G}}$ and $\overrightarrow{X_{r3,G}}$ define the different population individual values inside population at Gth iteration. The terms $\overrightarrow{X_{i,G}}$ and $\overrightarrow{X_{best,G}}$ in (26) denote the ith individual and current best individual at the Gth iteration, respectively. The F_m value has been considered as 0.8 for both the schemes after several trial runs. The value of λ is chosen as 0.3 for the scheme II related to differential evolution in (26). On the other hand, the initial temperature for the simulation of simulated annealing (SA) has been considered as 1. The minimum temperature for SA has been chosen as 10^{-10}. The Boltzmann constant for SA has been selected as 1. The cooling factor value for SA has been chosen as 0.9. The energy norm value for SA has been selected as 10^{-5}. The above said parameters for implementation of simulated annealing have been chosen after several trial runs. In the case of FEO, the generalized parameter values except the number of electro-fish have been chosen for optimization of eggcrate function. This is already shown in Table 1 in the earlier section. The number of electro-fish, i.e., the population size has been considered as 30 as said earlier. The three probabilities values have been considered same as Martin and Gaddy function optimization in Table 2. The *long range* multiplying factors have been chosen same as for Michaelwicz function optimization in Table 2. The *short range* multiplying factors have been selected same as for Sphere function optimization in Table 2. The problem-dependent and generalized parameters of FEO have been chosen after several trial runs.

The comparative study is shown in Table 16. It has been observed from Table 16 simulated annealing takes mean number of function evaluation and standard deviation as 5854 and 3289, respectively. The percentage of success of simulated annealing has been found as 84 in Table 16 for 100

time single independent runs. The total running time taken by simulated annealing is 167.08 s. It has been noticed from Table 16 that mean number of function evaluation and standard deviation taken by differential evolution (Scheme I) are 3138 and 378 for achieving the desired convergence. The percentage of success is 100 in this case in Table 16. The total running time taken by this differential evolution (scheme I) is 83.71 s for 100 time single independent run in Table 16. Finally, mean number of function evaluation and standard deviation have been observed as 2813 and 480, respectively, for differential evolution (scheme II) in Table 16. Again the percentage of success in this case is 100. The total running time taken by this algorithm is 75.08 s for 100 time single independent run in Table 16. Lastly, it has been noticed from Table 16 that FEO takes mean number of function evaluation and standard deviation as 1076 and 591 for eggcrate function optimization with the desired convergence. The percentage of success for 100 time single independent runs is 100. Significantly the total running time for FEO is 11.87 s. It can be said from the above description that FEO works quite better than differential evolution and simulated annealing for this kind of continuous function optimization.

The newly developed soft computing technique FEO has been successfully implemented on eggcrate function to find the global minimum point. The simulation study regarding FEO has been done for continuous variable related optimization problem hitherto. The proposed technique has not been tested for real-world discrete combinatorial optimization problem. This study has been done in the next section for reliability worth enhancement by optimal capacitor and distributed generator placement.

7 Implementation of FEO on real-world optimization problem

Fish electrolocation optimization has been applied to real-world optimization problem. The real-world problem is cost-based reliability enhancement in radial distribution system. Cost-based reliability has been improved by suitably placing capacitors and distributed generators at the sensitive buses. Capacitor allocation and distributed generation penetration have been considered to minimize the reliability worth as well as real power loss. The objective function has been described in (27) as:

$$Minimize\ S. \tag{27}$$

The term S is reliability worth. That is depicted in (28) as *Relworth*.

$$S = Relworth \tag{28}$$

The reliability worth value has been illustrated in (29) as:

Table 16 Comparative study with differential evolution and simulated annealing for 100 independent runs

Parameter/methods	Mean number of function evaluation	Standard deviation	Percentage of success	Total running time (s)
Simulated annealing	5854	3289	84	167.08
Differential evolution (scheme I)	3138	378	100	83.71
Differential evolution (scheme II)	2813	480	100	75.08
Fish electrolocation optimization	1076	591	100	11.87

$$Relworth = \sum_{j=1}^{nbr} La(j).C(j).\lambda(j). \qquad (29)$$

In that expression, $La(j)$ is the connected load at branch j and $C(j)$ is the interruption cost at branch j. Furthermore, $\lambda(j)$ in expression (29) is the failure rate of branch j. The value of α_i is depicted in (30) as:

$$\alpha_i = \frac{Inew^i}{Iold^i}. \qquad (30)$$

It is a ratio between absolute or modal value of new current to old current. The new current value is generated after DG penetration and capacitor placement. The old current value is the current amplitude of original configuration, i.e., without capacitor placement and DG penetration. The expression of new failure rate is illustrated in (31) as:

$$\lambda^{new} = \alpha_i(\lambda^{uncomp} - \lambda^{comp}) + \lambda^{comp}. \qquad (31)$$

In that Eq. (31), λ^{uncomp} is the failure rate of AC cable without any DG penetration and capacitor allocation. The another term in Eq. (31) is λ^{comp}. This term is considered as the minimum failure rate owing to capacitor placement and DG penetration. The Eq. (31) is valid when the value of α_i is greater than 0.5. Otherwise the minimum failure rate is considered as the new failure rate. The expression of cost-based reliability index called CBRI is illustrated in (32) as:

$$CBRI = \frac{(Relworth^{ini} - Relworth^{it}) \times loss^{ini}}{loss^{it} \times cost^{it}}. \qquad (32)$$

It can be observed from the RHS of expression (32) that CBRI is a ratio of two terms. The numerator of that ratio is the multiplication of two terms. One term is the difference of two reliability worth. The first reliability worth is without any capacitor placement and DG penetration. The second reliability worth is after capacitor placement and DG penetration. The multiplication factor with that said difference of two reliability worth is the initial real power loss without any capacitor placement and DG penetration. The denominator term of the discussed ratio related to CBRI is the multiplication of two terms. One term is the real power loss at certain

iteration. The second term is the cost term connected to the real power loss in (33) as:

$$cost^{it} = K_P \cdot loss^{it}. \qquad (33)$$

The discussed ratio CBRI is actually one type of difference of two reliability worth connected with real power lost and concerning cost. As discussed earlier the objective is to minimize reliability worth. So the reliability worth after capacitor allocation and DG penetration is necessarily lower than the reliability worth of original configuration, i.e., without any capacitor and DG placement. That is why the ratio value is positive if the objective of the optimization problem is fulfilled. And more to it, the CBRI value will be proportionately higher with the decrement of reliability worth value after capacitor and DG placement.

Fish electrolocation optimization has been implemented for cost-based reliability enhancement. It has been intelligently done by suitable choosing DG and capacitor values at particular bus positions. The DG penetration has been considered as the 10 % of the total active power load. The DG values have been chosen as 100 and 200 kW, respectively, for implementation of the soft computing techniques. The dollar conversion constant term K_P has been chosen as US\$ 168/kW. The simulation work is done on technical software MATLAB. A standard 34 bus radial distribution system has been considered for implementation of this said soft computing technique (Mekhamer et al. 2003). The scheme chosen for capacitor allocation and DG penetration in this radial distribution system is applying disconnects between buses and fuse gear protection at lateral junction point. The failure rate for maximum impedance and minimum impedance of line data has been considered as 0.5 and 0.1, respectively. The other failure rate has been enumerated between the two mentioned limits linearly with the rest impedance value. The value of failure rate after full compensation related to capacitor allocation and DG penetration has been chosen as 85 % of the uncompensated failure rate as denoted earlier. The interruption cost value has been chosen as US\$ 15.4752, US\$ 7.6317 and US\$ 1.86 for 4, 2 and 0.5 h of outage time, respectively. A comparative study with real coded genetic algorithm (RCGA[2]) and PSO is shown in Table 17. It can be observed from the Table 17 that without placing any capacitor and DG into the radial distribution system, the real power

Table 17 Comparative study of results after DG and capacitor placement

Parameters	Without any DG and capacitor placement	RCGA[2]	PSO	FEO
Loss (kW)	221.5	152.12	150	145.59
Reliability worth (US$)	163,478.43	159,690	160,150	156,100
CBRI	0	0.2162	0.1949	0.4591
DG value (kW)	–	200, 200	200	100, 100, 200
DG placement	–	26, 29	18	15, 31, 32
Capacitor value (kVAR)	–	900, 150, 300, 150	300, 900, 450, 600, 150, 450	1200, 1200, 150, 150, 600
Capacitor placement	–	5, 15, 21, 22	1, 18, 21, 22, 29, 33	5, 6, 14, 21, 23

loss has been found out at 221.5 kW. The reliability worth for that mentioned situation is found out at US$ 163,478.43 in Table 17. It has been observed from Table 17 that the cost-based reliability index named CBRI has been found at zero for that said condition. It is shown in Table 17 that utilizing RCGA[2], the real power loss has been reduced from 221.5 to 152.12 kW. It has been observed from Table 17 that with the implementation of PSO, the real power loss has been decreased from 221.5 to 150 kW. On the other hand, the real power loss has been observed to be reduced from 221.5 to 145.59 kW with the implementation of the proposed algorithm FEO in Table 17. This said reduction of real power loss has been occurred due to the reactive power compensation by capacitor placement and reduction of line loading by DG penetration. It has been observed from Table 17 that reliability worth has been reduced from US$ 163,478.43 to US$ 159,690 with the application of RCGA[2]. Reliability worth has been observed to be reduced from US$ 163,478.43 to US$ 160,150 with the implementation of PSO. On the other hand, with the implementation of proposed algorithm FEO reliability worth has been reduced from US$ 163,478.43 to US$ 156,100. The reliability worth value in (29) is linked with the failure rate of the cable. After doing capacitor allocation and DG penetration the failure rate decreases. That is why the reliability worth reduces after suitable choosing capacitor and DG's location and value, respectively, by those mentioned techniques. It has been noticed from Table 17 that the reliability index CBRI has been increased from zero to 0.2162 with the implementation of RCGA[2]. The reliability index CBRI has been observed to be increased from zero to 0.1949 with the utilization of PSO in Table 17. On the other hand, it is shown in Table 17 that the reliability index CBRI has been increased from zero to 0.4591 with the implementation of the said algorithm FEO. The proposed algorithm FEO has got higher and better CBRI value amongst all the chosen soft computing techniques. It has been observed from Table 17 that RCGA has chosen DG values as 200 and 200 kW for bus positions 26 and 29, respectively. It is shown in Table 17 that PSO has selected 200 kW DG value for the bus number 18. On the other hand, three DG values are chosen by proposed algorithm FEO for bus locations 15, 31 and 32, respectively. The selected DG values for those said locations are 100, 100 and 200 kW in Table 17. As said earlier reactive power compensation by capacitor allocation has been done by all the mentioned techniques. It is shown in Table 17 that capacitor value of 900, 150, 300 and 150 kVAR have been chosen by RCGA[2] for bus position 5, 15, 21 and 22, respectively. It has been noticed from Table 17 that six capacitor values have been selected by PSO for bus position 1, 18, 21, 22, 29 and 33, respectively. The six capacitor values for those mentioned locations are 300, 900, 450, 600, 150 and 450 kVAR in Table 17. On the other hand, it has been observed from Table 17 that capacitor values of 1200, 1200, 150, 150 and 600 kVAR have been chosen by FEO for 5, 6, 14, 21 and 23, respectively. It can be concluded from the discussion that proposed algorithm FEO has beaten other chosen soft computing techniques viz. RCGA[2] and PSO. The said algorithm FEO has reduced the real power loss at a higher degree in comparison to RCGA[2] and PSO. It has also been discussed that the reliability worth related to customer interruption cost has been decreased by FEO at a higher degree in comparison to RCGA[2] and PSO. Furthermore, the cost-based reliability index CBRI has been increased at a higher degree by FEO in comparison to RCGA[2] and PSO as discussed in this section. Finally, it can be said that the proposed algorithm FEO works better in finding the suitable DG and capacitor value for certain locations. It does the above engineering work to enhance the reliability economically and also technically.

However, the FEO has been successfully applied on real-world optimization problem. The critical issues regarding the proposed algorithm should be discussed. This has been stated in the next section.

8 Analysis of critical issues of fish electrolocation optimization

There are many critical issues of an algorithm. They are robustness, flexibility, benefits, feasibility, divergence and convergence capability, stopping criteria, stability and reli-

Table 18 Experiment on eggcrate function applying FEO for 10 times run

Pop. size	Probabilities			Objective function value				Computation time (s)
	P^{rng}	P^{div}	P^{sel}	Max.	Min.	Mean	Std. dev.	
5	0.01	0.01	0.5	9.7985×10^{-5}	6.0048×10^{-6}	5.1630×10^{-5}	3.5532×10^{-5}	1.7
5	0.01	0.02	0.6	9.9598×10^{-5}	6.6110×10^{-7}	6.4478×10^{-5}	3.9409×10^{-5}	0.68
5	0.02	0.02	0.6	9.5121×10^{-5}	6.0826×10^{-6}	7.0631×10^{-5}	2.9134×10^{-5}	0.89
15	0.01	0.01	0.5	9.9797×10^{-5}	4.0349×10^{-6}	5.9242×10^{-5}	2.7204×10^{-5}	1.14
15	0.01	0.02	0.6	8.5023×10^{-5}	1.7428×10^{-6}	4.9021×10^{-5}	2.6253×10^{-5}	0.96
15	0.02	0.02	0.6	7.4022×10^{-5}	5.3016×10^{-6}	4.6947×10^{-5}	2.2949×10^{-5}	0.79
30	0.01	0.01	0.5	8.2354×10^{-5}	1.4460×10^{-5}	4.9109×10^{-5}	2.4547×10^{-5}	1.38
30	0.01	0.02	0.6	9.7812×10^{-5}	3.4143×10^{-5}	7.0763×10^{-5}	2.0640×10^{-5}	1.61
30	0.02	0.02	0.6	9.1455×10^{-5}	1.2644×10^{-5}	5.7065×10^{-5}	2.4121×10^{-5}	1.49

ability, efficiency and effectiveness, productivity, computational complexity and advantages and disadvantages of the proposed method, etc. The important issues amongst the above-mentioned criteria have been discussed here for FEO.

- *Robustness* In computer science, robustness of an algorithm is defined as the capability to pursue computation despite abnormalities in input, enumeration, etc. Though random number generation is the governing factor for creation of population individual in FEO, there are three kinds of probabilities and three kinds of ranges in the proposed technique. If there are any abnormalities in input, the three kinds of ranges, i.e., *longrange*, *shortrange* and *vshortrange* make it feasible to have some nearness to the global optimal point. Furthermore, electrical capacitance value by the toggle switch judgment helps it to continue enumeration in the converging path. An experiment has been done to show the robustness of FEO. This is shown in Table 18. It has been observed from Table 18 that with the tuning of probabilities the minimum and maximum objective function value for the minimization problem change very little. Whereas it has been observed that with less population number FEO performs best. This simulation study tells about the robustness of FEO.
- *Flexibility* FEO is flexible in nature. It can be applied to one-dimensional problem as well as multidimensional objective function. The discussion in the earlier sections is the evidence of that. It has been applied to continuous optimization problem. It can also be applied to discrete optimization problem if the three kinds of ranges have been chosen judiciously. It can be said from the description stated in the previous sections that proposed FEO is more flexible than simulated annealing and differential evolution and it performs well for combinatorial optimization problem. This has been shown in Sect. 7.
- *Benefits* Benefits of FEO are primarily the advantages of utilizing the proposed method. The simulation studies done in the entire research work tell that FEO algorithm is robust, flexible, efficient and effective in finding not only theoretically correct optimal solution but also it can find real-world feasible optimal solution. This also states about the stability of the proposed method.
- *Feasibility* FEO has been tested for continuous optimization problem as well as combinatorial optimization problem. The simulation results show that the solutions are feasible especially for real-world optimization problem studied in Sect. 7.
- *Divergence and convergence capability* Divergence and convergence capability of an evolutionary algorithm are considered as the randomization and localization capability towards the global optimal point. The randomization of the FEO is expressed through Eq. (15). The localization phenomenon of the proposed technique has been illustrated by the expressions (16), (17) and (18), respectively. The proposed algorithm divergence and convergence criteria are governed by the $prob^{div}$ and $prob^{rng}$, respectively. It has been observed in earlier sections that FEO has better divergence and convergence capability than PSO and real coded genetic algorithm. In Sect. 6, it has been found that the randomization and localization quality of FEO is better than simulated annealing and differential evolution.
- *Stopping criteria* stopping criteria of the FEO have been considered as the desired convergence criterion as expressed in Eq. (23) and maximum iteration number.
- *Stability and reliability* Stability of an algorithm is defined as how changes to the training data influence the result of the algorithm. It can be said that from earlier discussion that proposed FEO is more stable than genetic algorithm and PSO. The proposed method as discussed earlier is also able to find out the real-world feasible solution in Sect. 7. On the other hand, reliable product in computer science means that it would be totally free of technical errors. So, it depends on the carefulness and

Table 19 Experiment on system of linear equations

System of linear equations	FEO	Gaussian elimination method	Actual solutions
$x_1 + 2x_2 + 3x_3 = 14$	$x_1 = 1$	$x_1 = 0$	$x_1 = 1$
$x_1 + x_2 + x_3 = 6$	$x_2 = 2$	$x_2 = 4$	$x_2 = 2$
$3x_1 + 2x_2 + x_3 = 10$	$x_3 = 3$	$x_3 = 2$	$x_3 = 3$
	$x_1 = 0$		
	$x_2 = 4$		
	$x_3 = 2$		
	$x_1 = 2$		
	$x_2 = 0$		
	$x_3 = 4$		

diligence of the person coding FEO in a certain computer language. Furthermore, it can be said from the discussion written in the earlier section that proposed FEO is more stable and reliable than simulated annealing and differential evolution.

- *Efficiency and effectiveness* Previous simulation studies tell about the efficiency of FEO. The various parameters of FEO have been set on the basis of several run and best efficiency. The effectiveness of FEO has been tested with Gaussian elimination method on system of linear equations (Ikotun et al. 2011). It has been observed from Table 19 that FEO has obtained all the three sets of solutions for the system of linear equations, whereas the Gaussian elimination method has found out only one set of results. This simulation study shows the effectiveness of FEO.
- *Productivity* Productivity has been considered here as the ratio of number of source code line and computation time in i3, 2.4 GHz machine. The source code line in MATLAB is 206 for implementing FEO on the system of linear equations in Table 19. The computation time is 0.62 s. The productivity for this case is 332.25.
- *Computational complexity* Computational complexity is a subject of computer science. It is of two types. One is time complexity and the other one is space complexity. Time complexity of an algorithm is defined as the amount of time taken by that algorithm to run. It is quantified as a function of the length of string representing the input. Generally, time complexity is calculated by counting the number of elementary operations performed by the algorithm. Significantly the elementary operation takes a fixed amount of time. Time complexity is a property of deterministic algorithm. But FEO is a stochastic or non-deterministic algorithm. Time complexity of this kind of algorithm would be number of generation multiplied by the population size. But the number of generation varies with random number generation as the genesis of population individual is governed by it. Furthermore, the number of generation of this kind of evolutionary algorithm is problem specific.

Space complexity of an algorithm is defined by the total space taken by that algorithm with respect to input size. The space complexity of the proposed FEO is $O(m.NL)$. Here m is the number of rows in one column vector or population individual. It can be said that m is the dimensionality of the objective function. The other term NL is the population size or number of population individuals.

- *Advantages and disadvantages* There are a few advantages and disadvantages of FEO. The advantages are as follows:

  1. The convergence time or computation time is quicker than the other well established algorithms.
  2. The algorithm is flexible and robust.
  3. It has good divergence and convergence capability.

The disadvantages are as follows:

1. There are three kinds of ranges in the algorithm. One has to carefully design the three ranges for continuous as well as discrete or combinatorial optimization problem.
2. There are a few parameters in the algorithm. Those are more or less problem specific.

9 Conclusion

Optimization algorithm based on the behavior of electrolocation technique of elephant nose fish and shark has been successfully developed and tested. It has been observed that FEO algorithm works better on test bed function than established evolutionary algorithm such as genetic algorithm, particle swarm optimization, differential evolution, simulated annealing, etc. Comparative study shows that the method has got potential to be applied like other well established techniques. The proposed method has also outperformed chosen soft computing techniques for real-world optimization problem related to reliability worth enhancement in radial

distribution system. Finally, hybridization of this proposed method with other well established techniques can also be done to reap the benefit of the participating techniques.

Compliance with ethical standards

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

References

Active electrolocation. http://www.nbb.cornell.edu/neurobio/hopkins/Publication.htm. Accessed 3 Apr 2012

Afshar A, Haddad OB, Marino MA, Adams BJ (2007) Honey bee mating optimization algorithm for optimal reservoir operation. J Frankl Inst 344:452–462

Ammari H, Boulier T, Garnier J (2013) Modeling active electrolocation in weakly electric fish. SIAM J Imaging Sci 6(1):285–321

Baffet G, Boyer F, Gossiaux PB (2008) Biomimetic localization using the electrolocation sense of the electric fish. In: Proceedings of the IEEE international conference on robotics and biomimetics, Bangkok, pp 659–664

Cai W, Yang WW, Chen X (2008) A global optimization algorithm based on plant growth theory: plant growth optimization. In: International conference on intelligent computation technology and automation, pp 1194–1199. doi:10.1109/1CICTA.2008.416

Cuevas E, González M, Zaldivar D, Pérez-Cisneros M, García G (2012) An algorithm inspired by collective animal behaviour. Discret Dyn Nat Soc 1–24

Das S, Abraham A, Konar A (2008) Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives. Stud Comput Intell 116:1–38

Deb K (1991) Optimal design of a welded beam via genetic algorithms. AIAA J 29:2013–2015

Deb K (2000) An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng 186:311–338

Dorigo M (1992) Optimization, learning and natural algorithms. Ph.D. thesis, Politencnico di Milano, Italy

Dorigo M, Caro GD (1999) Ant colony optimization: a new metaheuristic. In: Proceedings of the IEEE congress on evolutionary computation, pp 1470–1477

Electric fish. http://www.bio.davidson.edu/people/midorcas/animalphysiology/websites/2003/wilson/index.htm. Accessed 3 Apr 2012

Emde GVD (1998) Electric fish measures distance in the dark. Nature 395:890–894

Emde GVD (1999) Active electrolocation of objects in weekly electric fish. Exp Biol 202:1205–1215

Emde GVD (2004a) Remote sensing with electricity: active electrolocation in fish and technical devices. In: Presented at the 1st international industrial conference, Hannover Messe, Bionik

Emde GVD (2004b) Distance and shape: perception of the 3-dimensional world by weekly electric fish. Physiol Paris 98:67–80

Emde GVD, Schwarz S (2002) Imaging of objects through active electrolocation *Gnathonemus petersii*. Physiol Paris 96:431–444

Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76:60–68

Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison Wesley, Boston

Haldar V, Chakraborty N (2011a) Switched capacitor bank installation in practical transmission system using modified cultural algorithm. In: Proceedings of the IET 2nd international conference sustainable energy and artificial intelligence, Chennai, pp 444–449

Haldar V, Chakraborty N (2011b) Root shoot coordination optimization: conceptualizing ascent of sap and translocation of solute in plant. In: Presented at the international conference on soft computing and engineering application, Kolkata

He S, Wu QH, Saunders JR (2009) Group search optimizer: an optimization algorithm inspired by animal searching behaviour. IEEE Trans Evol Comput 13:973–990

Hedar AR (2005) Test Functions for unconstrained global optimization. http://www.optima.amp.ikyoto-u.ac.jp/member/student/hedar/Hedar-files/TestGo-files/Page364.htm. Accessed 13 May 2012

Hedar AR, Fukushima M (2006) Derivative free filter simulated annealing method for constrained continuous global optimization. J Glob Optim 35:521–649

Holland JH (1975) Adaptation in natural and artificial systems. The University of Michigan Press, Ann Arbor

Hopkins CD (2005) Passive electrolocation and the sensory guidance of oriented behavior, pp 264–289. http://www.nbb.cornell.edu/neurobio/Hopkins/Reprints/Hopkins_passive.pdf. Accessed 3 Apr 2012

Ikotun AM, Lawal ON, Adelokun AP (2011) The effectiveness of genetic algorithm in solving simultaneous equations. Intl J Comput Appl 14(2):1–4

Kalmijn AJ (1971) The electric sense of sharks and rays. J Exp Biol 55(2):371–383

Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39:459–471

Kennedy J, Eberhat R (1995) Particle swarm optimization. In: Proceedings of the 4th IEEE international conference on neural networks, pp 1942–1948

Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220:671–680

Krishnanand KN, Ghose D (2005) Detection of multiple source locations using a glowworm metaphor with applications to collective robotics. In: Proceedings of the IEEE swarm intelligence symposium, Pasadena, pp 84–91

Lee KS, Geem ZW (2005) A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng 194:3902–3933

Maciver MA, Nelson ME (2001) Towards a biorobotic electrosensory system. Auton Robot 11:263–266

Mekhamer SF, Soliman SA, Moustafa MA, El-Hawary ME (2003) Application of fuzzy logic for reactive power compensation of radial distribution feeders. IEEE Trans Power Syst 18:206–213

Moscato P (1989) On evolution search, optimization, genetic algorithms and martial arts: towards memetic algorithms. Report 826, Caltech Concurrent Computation Program, California

Muhaureq SA, Saad M, El-Saddik A (2010) Design and implementation of echolocation optimization algorithm and its application in wireless networks. Master's thesis, Sharjah University, Sharjah

Nakrani S, Tovey C (2004) On honey bees and dynamic server allocation in internet hosting centers. Adapt Behav 12:223–240

Nelson ME, Maciver MA (2006) Sensory acquisition in active sensing systems. J Comp Physiol A 192:573–586

Oftadeh R, Mahjoob MJ, Shariatpanahi M (2010) A novel meta-heuristic optimization algorithm inspired by a group hunting of animals: hunting search. Comput Math Appl 60:2087–2098

Olivera DRD, Parnelli RS, Lopes HS (2011) Bioluminescent swarm optimization algorithm. Numer Anal Sci Comput 69–84. http://tainguyenso.vnu.edu.vn/jspui/handle/123456789/17260. Accessed 24 Mar 2012

Passive electrolocation in fish. http://en.wikipedia.org/wiki/passive_electrolocation_in_fish. Accessed 3 Apr 2012

Passino KM (2002) Bio mimicry of bacterial foraging. IEEE Control Syst Mag 52–67

Pham DT, Ghanbarzadeh A, Koc E, Otri S, Rahim S, Zaidi M (2005) The bees algorithm. Technical Note, Manufacturing Engineering Centre, Cardiff University

Rao BV, Kumar GVN (2015) A comparative study of BAT and Firefly algorithm for optimal placement and sizing of static var compensator for enhancement of voltage stability. Int J Energy Optim Eng 4:68–84

Reynolds RG (1994) An introduction to cultural algorithm. In: Proceedings of the 3rd annual conference on evolutionary programming, pp 131–139

Seref O, Akcali E (2002) Monkey search: a new metaheuristic approach. In: Proceedings of the INFORMS annual meeting, San Jose

Simon D (2006) Biogeography based optimization. IEEE Trans Evol Comput 12:702–713

Shieh KT, Wilson W, Winslow M, Mcbride DW Jr, Hopkins CD (1996) Short-range orientation in electric fish: an experimental study of passive electrolocation. Exp Biol 199:2383–2393

Solberg JR, Lynch KM, Maciver MA (2008) Active electrolocation for underwater target localization. Int J Robot Res 27:529–548

Solberg JR, Lynch KM, Maciver MA (2013) Robotic electrolocation: active underwater target localization with electric fields. http://nxr.northwestern.edu/sites/default/files/publications/Solb07a.pdf. Accessed 10 July 2013

Startchev K, Fua P, Porez M, Crepsi A, Ijspeert A (2011) Algorithms inspired from active electrolocation behaviour of weak electric fish, developed for autonomous eel-like swimming robot. http://www.emn.fr/z-dre/bionic-robots-workshop/uploads/Abstracts%20BRW%202011/15.pdf. Accessed 10 July 2013

Storn R (1996) On the usage of differential evolution for function optimization. In: Biennial conference of the north American fuzzy information processing society, pp 519–523

Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359

Teodorovic' D, DellOrco M (2005) Bee colony optimization—a cooperative learning approach to complex transportation problem. In: Proceedings of the 10th EWGT meeting, Poznan

Tong L, Wei-Ling S, Chung-feng W (2004) A global optimization bionics algorithm for solving integer programming—plant growth simulation algorithm. In: Proceedings of the international conference management science and engineering, Harbin, pp 531–535

Vo DN, Schegner P (2013) An improved particle swarm optimization for optimal power flow. In: Vasant PM (ed) Meta-heuristics optimization algorithms in engineering, business, economics, and finance, vol 1, pp 1–40

Wright AH (1991) Genetic algorithm for real parameter optimization, pp 1–12. http://citeseerx.ist.psu.edu. Accessed 13 Mar 2012

Yang XS (2005) Engineering optimization via nature-inspired virtual bee algorithms. In: IWINAC'05. Lecture notes in computer science, vol 3562, pp 317–323

Yang XS (2009) Firefly algorithms for multimodal optimization. Lect Notes Comput Sci 5792:169–178

Yang XS (2010a) A new meta-heuristic bat-inspired algorithm. Stud Comput Intell 284:65–74

Yang XS (2010b) Nature inspired meta-heuristic technique. Luniver Press, Bristol

Yang XS, Deb S (2009) Cuckoo search via lévy flights. In: Proceedings of the IEEE world congress nature and biologically inspired computing, pp 210–214

Yuce B, Mastrocinque E, Packianather MS, Lambiase A, Pham DT (2015) The bees algorithm and its application. In: Vasant PM (ed) Handbook of research on artificial intelligence techniques and algorithms, vol 4, pp 122–151