Deterministic oscillatory search: a new meta-heuristic optimization algorithm

N ARCHANA^{1,*}, R VIDHYAPRIYA¹, ANTONY BENEDICT² and KARTHIK CHANDRAN²

¹PSG College of Technology, Coimbatore 641004, India

²Alpha Research and Development, Coimbatore 641004, India
e-mail: archana.nathan31@gmail.com

MS received 2 June 2016; accepted 14 November 2016

Abstract. The paper proposes a new optimization algorithm that is extremely robust in solving mathematical and engineering problems. The algorithm combines the deterministic nature of classical methods of optimization and global converging characteristics of meta-heuristic algorithms. Common traits of nature-inspired algorithms like randomness and tuning parameters (other than population size) are eliminated. The proposed algorithm is tested with mathematical benchmark functions and compared to other popular optimization algorithms. The results show that the proposed algorithm is superior in terms of robustness and problem solving capabilities to other algorithms. The paradigm is also applied to an engineering problem to prove its practicality. It is applied to find the optimal location of multi-type FACTS devices in a power system and tested in the IEEE 39 bus system and UPSEB 75 bus system. Results show better performance over other standard algorithms in terms of voltage stability, real power loss and sizing and cost of FACTS devices.

Keywords. Artificial intelligence; global optimization; oscillatory search; meta-heuristic optimization; power system problem.

1. Introduction

Engineering optimization is a challenging area that is aimed at increasing efficiency of design and operations in engineering by obtaining an optimal solution to the problem. In most of the cases, optimal solution is a vector that gives maximum or minimum value for an objective function. The vector may be constrained or unconstrained depending on the workspace. Several methods have been proposed for optimization and can be classified into two categories: (1) classical or gradient-based methods and (2) intelligent or meta-heuristic methods. Classical methods such as linear programming obtain optimal solution to the problem by rigorous methods supplemented by gradient information. Classical methods are good for solving problems with only one extreme point in the search space. They follow the same path and they provide mostly same results for a problem for a number of trials. When the search space becomes noisy, the solution obtained with classical techniques is usually sub-optimal. To overcome this, meta-heuristic algorithms were developed, which are capable of finding the global optimal solution even in noisy environments. Various meta-heuristic algorithms have been proposed in the literature, which include genetic algorithm (GA) [1], differential evolution (DE) [2], particle swarm

optimization (PSO) [3], artificial bee colony optimization (ABC) [4], bacteria foraging optimization (BFO) [5], firefly algorithm (FA) [6], symbiotic organism search (SOS) [7], krill herd migration (KH) [8], cuckoo search (CS) [9] and harmony search (HS) [10]. They are usually superior in solving engineering problems of multiple dimensions and complexity. They have the following common traits.

1.1 Nature inspired

ACO is inspired by food search of an ant colony, ABC is based on bees finding flowers with adequate nectar, BFO is based on E-Coli bacterial movement in human intestine and FA is based on the mating sequence of fireflies. Almost all algorithms study how organisms interact with their environment for their survival and apply the concepts in finding the optimal solution to a problem.

1.2 Random numbers

Random numbers have been widely used in all optimization algorithms. This makes them stochastic, which is unpredictable to a certain extent. The addition of random numbers to the algorithm makes the algorithm generate new values never considered before and explore new places.

*For correspondence

This makes them superior to gradient-based methods, but makes the result inconsistent most of the time. Almost all stochastic algorithms show deviations in results if run for a number of trials. Most algorithms overcome this by having a large number of particles and/or iterations. A few algorithms show no deviation of results for certain optimization problems, but show higher deviation of results for some other problems considered. Also, as a computer is a deterministic machine, there is no possibility of generating true random numbers unless there is an external source of noise.

1.3 Tuning parameters

All algorithms have some parameters that are set by the user, which alter their behaviour to a great extent. PSO has the parameters $C1$ and $C2$, which determine how much importance the particles give to the global best and local best, respectively. Most algorithms may have more parameters to be tuned (on an average four), while SOS has only one (size of ecosystem), which is an exception. The presence of tunable parameters in the algorithm adds the necessity to run the optimization algorithm with different settings several times to obtain the best result for the problem, which is again an optimization performed by the user and adds to the inconsistency of the results.

Deterministic oscillatory search (DOS) differs from the afore-mentioned optimization algorithms by having none of the above characteristics. It is not nature inspired, it does not have any random numbers and it does not have any tunable parameters other than the number of particles for a given number of function evaluations.

2. Deterministic oscillatory search

2.1 Inspiration

Unlike other optimization algorithms DOS does not look at nature for inspiration. It is known that gradient methods are capable of converging to a local optimum. The problem arises only when the search space is noisy and is filled with several peaks and valleys. Under such conditions, gradient-based methods become unstable and mostly fail to obtain the global optimum. Meta-heuristic algorithms

overcome this by resorting to random numbers and *swarming*. Most algorithms use the concept of sharing information between the particles, which aids the overall community in becoming fitter. Almost all optimization algorithms use this technique in one way or the other. Velocity-based algorithms such as PSO, FA, SOS and GSA use it more directly compared with evolution-based algorithms such as GA, DE, etc. Therefore, DOS is inspired to be a gradient-based algorithm with swarming nature. Also, the second inspiration is to create an algorithm that is very robust, which shows zero deviation when run repeatedly with the same settings. Due to the lack of random numbers in the algorithm, DOS gives the *same result* for a given problem for any number of trials, making it extremely robust. Even the convergence characteristics do not change. DOS is inspired to retain the robustness of classical methods with improved convergence characteristics of meta-heuristic algorithms.

2.2 Concept

2.2a Initialization of particles in the search space: An algorithm without random numbers is difficult to implement mainly because of the initialization of the particles in the search space. Most meta-heuristic algorithms initialize the particles randomly in search space. In the case of DOS the particles are distributed evenly in the search space. But distributing particles evenly in the search space poses a challenge as it is difficult to define a specific algorithm for even distribution, which can be applied for multiple dimensions. It may mean that all particles are distributed with equal distance from each other or all particles are at the centre of search space volumes equally divided. The particles may be considered as repulsive substances like electrons and placed at the centre of the search space. In equilibrium the particles will be approximately evenly distributed. But this method of distribution adds complexity to the algorithm and makes it stochastic in nature. Also other geometry-based approaches become very difficult to design, as the problem dimension may exceed well beyond three and it becomes difficult to comprehend the multi-dimensional shapes. In the proposed algorithm a combination-based distribution is used, but the concept can be researched further and improved.

Step 1: Initialize all the particle's position to the lower limit boundary of the search space

Step 2: Initialize particle counter (i=1) and combination counter (k=1)

Step 3: Iterative process:

```

Initialize particle positions as combinations of upper and lower boundaries
while (i<= total number of particles)
    comb =find all possible combinations (pCk), where comb is the array to store the combinations, p is the
    number of dimensions of the search space
    for c=1: total number of combinations
        Change parameter positions of the particle Xi as high values as per the combinations, so
        that all positions are unique.
        i++
    end for
    k ++
    if (k > dimensions) /Exit the loop if no more unique boundary combinations are possible/
        break
    end if
end while
Step 4: If (more particles are left, use factor based approach to initialize particle positions between two pre
initialized particles)
    Factorize the number of particles
    f=factor(n) /n = total number of particles/
    /f is an array to store the factor values/
    Iterative process:
    while (i<=n)
        for c=1:total number of factors (length of f)
            for j=1:dimensions (p)
                for l= 1:f(c)
                    Xi(j)=Min(j)+k*(Max(j)-Min(j))/(f(c)+1)
                    / repeat with Min Max swapped where Min and Max are the arrays to store upper and
                    lower boundary limits/
                    i++
                end
            end
            if i>n /exit loop if all possible combinations are exhausted and repeat step 4 with Min and Max
            swapped/
                break
            end if
        end for
    end while

```

Figure 1 shows the initialization of the particles in a search space of $(-512 < x < 512; -512 < y < 512)$. The algorithm initializes the particles along the edges of the search space. The particles are always influenced by attractive forces towards the global best and hence they fly towards the centre. This type of initial distribution is sufficient to solve most of the mathematical and engineering problems. The objective function value of each particle is calculated and the position of the particle with the best value is considered as x_{best} .

Figure 1. Initial population distribution (asterisk is the observed global best position).

2.2b. *Movement of particles in search space:* In DOS, the particles are considered to be in motion in each iteration under the influence of velocity. The velocity of the particles is initialized using the following equation.

$$V_i^0 = \left[\frac{x_{best} - x_i^0}{2} \right] \quad (1)$$

where V_i^0 and x_i^0 are the initial velocity and position of a particle 'i'. The particle's movement is based on

1. gradient-based movement
2. swarming movement

In gradient-based movement, the particles are allowed to move with constant velocity if their movement along the direction of the velocity produces better results in successive trials. If the results are worse than the previous result, the velocity of the particle is reversed with half the magnitude. This aids the particle to achieve the local optimum solution. This type of movement of the particle is similar to an oscillation or zigzag movement.

A concept of fitness slope is incorporated in the algorithm, with three scenarios:

Figure 2. Movement of the particle towards local optimal solution.

1. positive (current movement is producing better results),
2. negative (current movement is producing worse results) and
3. unknown (velocity updated and slope is unknown).

Figure 2 shows the zigzag movement of the particle towards the local best, aided by the fitness slope information for a function $y = f(x)$.

- (i) Particle moves from x_1 to x_2 . This movement results in a negative fitness slope.
- (ii) The velocity of the particle is multiplied by -0.5 , the particle moves from x_2 to x_3 and it experiences a positive fitness slope.
- (iii) The particle continues its movement and reaches x_4 . The particle again experiences a negative fitness slope and its velocity is again multiplied by -0.5 .
- (iv) The particle now reaches x_5 , and since the fitness slope is now positive, the particle continues its movement along the same direction and reaches x_6 .
- (v) The particle velocity is again reversed and ultimately it reaches the local optimal location x_{opt} .

In this way, all the particles reach their respective local optimal solution in an oscillatory motion. However, gradient method has inherent disadvantages when the search space is filled with several peaks and valleys. This drawback is overcome in the proposed technique by incorporating swarming behaviour. Swarming is applied only when the particle fails to produce better results by gradient movement. Swarming is implemented by making the particle move towards the global best position by updating the velocity using the following equation.

$$V_i^{t+1} = V_i^t + \left[\frac{x_{best} - x_i^t}{2} \right] \quad (2)$$

If the movement becomes favourable then the particle continues to find another local optimal solution, and this process continues till the maximum number of evaluations is reached.

3. Pseudocode

```

Step 1: The following parameters of the algorithm are initialized:
    (i) number of particles and
    (ii) maximum number of evaluations.
Step 2: The particles are distributed using the combinatorial deterministic method in the search space.
Step 3: The fitness values of all the particles are found and the best particle is found.
Step 4: The particle velocities are initialized using the equation and their fitness slope is categorized as unknown.
Step 5: Iterative process
    while (number of eval < max eval)
        for (i = 1:number of particles)
            1. find new position by adding velocity to the current position
            2. find the fitness of the particle with respect to new position
            3. increment eval counter, update global best
            4. if (slope == unknown)
                update slope of the particle to be either positive or negative based on the new fitness
            else if (slope == positive)
                if (new fitness worse than old fitness)
                    (i) update velocity  $\rightarrow (-\text{velocity}/2)$ 
                    (ii) update slope to be negative
            else if (slope == negative)
                if (new fitness worse than the old fitness)
                    (i) update velocity  $\rightarrow \text{velocity} + (\text{global best position} - \text{current position})/2$ 
                    (ii) update slope to be unknown
            end if
        end for
    end while
Step 6: The global best particle position and fitness are displayed.

```

4. Simulation and results

4.1 Mathematical function

The algorithm is tested with several benchmark mathematical test functions and a power system problem. The test function results are compared to the results obtained using PSO [11], FA [12], BFO [13], ABC [11], GA [11] and SOS [7]. The limit for objective function evaluations for all the algorithms is set at 1000. The algorithm-specific parameters are given in table 1.

Here N is population size, w is inertia, N_c is number of chemotactic steps, N_{ed} is number of elimination/dispersion steps, P_{ed} is probability of elimination/dispersion, C_x is crossover length and P_m is probability of mutation. The

performance of the algorithms was evaluated in 16 benchmark functions as given in table 2. The functions are solved using standard algorithms and the proposed DOS method for 10 trials and the average optimal solution is displayed in table 3.

The optimal results obtained for various functions employing various algorithms, including DOS, are tabulated in table 3. The values less than $1E-12$ are considered as 0. The bolded numbers indicate that the algorithm has converged to the most optimal solution possible for the problem. From table 3, it can be inferred that DOS does marginally better when compared with other latest optimization algorithms. The extreme robustness of the algorithm resulted in the same solution for a given problem for all the trials. The proposed algorithm obtained global

Table 1. Simulation parameters.

PSO	FA	BFO	ABC	GA	SOS	DOS
$N = 40$	$N = 40$	$S = 24$	$N = 40$	$N = 40$	$N = 40$	$N = 40$
$C1 = 1.5$	$\text{Alpha} = 0.3$	$N_c = 3$	$nF = 20$	$C_x = 0.3$		
$C2 = 2.5$	$\text{Gamma} = 1$	$N_s = 2$	$\text{lim} = 10$	$P_m = 0.12$		
$W = 0.3$	$\text{Delta} = 0.97$	$N_{ed} = 12$				
Damping ratio = 0.95	$\text{Betamin} = 0.2$	$P_{ed} = 0.2$				

Table 2. Mathematical benchmark functions.

Sl. no.	Function	Equation
1	Ackley	$f(x, y) = -20\exp(-0.2\sqrt{(0.5(x^2 + y^2))}) - \exp(0.5(\cos(2\pi x) + \cos(2\pi y))) + e + 20\sqrt{(0.5(x^2 + y^2))})$
2	Sphere	$f(x) = \sum_{i=1}^n x_i^2$
3	Rosenbrock	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2) + (x_i - 1)]$
4	Beale	$f(x, y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2$
5	Booth	$f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$
6	McCormick	$f(x, y) = \sin(x + y) + (x - y)^2 - 1.5x + 2.5y + 1$
7	Step	$f(x) = \sum_{i=1}^n (x_i + 0.5)^2$
8	Six-Hump Camel Back	$f(x, y) = 4x^2 - 2.1x^4 + \frac{1}{3}x^6 + xy - 4y^2 + 4y^4$
9	Rastrigin	$f(x) = \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i)) + 10$
10	Griewank	$f(x) = \frac{1}{4000} \left(\sum_{i=1}^n (x_i - 100)^2 \right) - \left(\prod_{i=1}^n \cos\left(\frac{x_i - 100}{\sqrt{i}}\right) \right) + 1$
11	Matyas Function	$f(x, y) = 0.26(x^2 + y^2) - 0.48xy$
12	Cross-In-tray	$f(x, y) = -0.0001(\sqrt{ y + \frac{x}{2} + 47 }) - x\sin(\sqrt{ x - (y + 47) })$
13	Schaffer N2	$f(x, y) = 0.5 + \frac{(\sin^2(x^2 - y^2)) - 0.5}{(1 + 0.001(x^2 + y^2)^2)}$
14	Schaffer N4	$f(x, y) = 0.5 + \frac{\cos^2(\sin(x^2 - y^2)) - 0.5}{(1 + 0.001(x^2 + y^2)^2)}$
15	Michalewicz	$f(x) = \sum_{i=1}^n \sin(x_i)(\sin(ix_i^2/\pi))^{20}$
16	Quartic	$f(x) = \sum_{i=1}^n ix_i^4 + \text{rand}$

Table 3. Mean optimization solution.

Function	Limits	PSO	FA	BFO	ABC	GA	SOS	DOS
Ackley	[-5 0, -5 0]	0	0	0	0	3.914978	0	0
Sphere	[-100 100, -100 100, -100 100]	0	0.00050	0	0	9	0	0.000006
Rosenbrock	[-5 5, -5 5]	0	0	0	0.088	0	0.000372	0
Beale	[-4.5 4.5, -4.5 4.5]	0	0.001226	0.000148	0	2.953125	0	0
Booth	[-10 10, -10 10]	0	0.000006	0.001226	0	0	0.000003	0
McCormick	[-1.5 4, -3 4]	-1.9133	-1.9133	-1.91321	-1.91056	-1.90929	-1.9133	-1.9133
Step	[-5.12 5.12, -5.12 5.12]	0	0	0.000026	0	0.25	0	0
Six-Hump Camel Back	[-5 5, -5 5]	-1.03162	-1.03162	-1.03098	-1.03162	-0.75051	-1.03162	-1.03162
Rastrigin	[-5.12 5.12, -5.12 5.12, -5.12 5.12]	0.74622	0.75	0.75	0	0.75	0.75	0.746219
Griewank	[-600 600, -600 600]	0	0.000002	0	0	0	0.005939	0.000002
Matyas function	[-10 10, -10 10]	0	0	0	0	0.039063	0	0
Cross-in-tray	[-10 10, -10 10]	-2.0626	-2.06261	-1.79059	-2.06260	-2.04370	-2.06260	-2.06261
Schaffer N2	[-100 100, -100 100]	0	0.009935	0	0	0	0.000076	0
Schaffer N4	[-100 100, -100 100]	0.292833	0.292604	0.301022	0.29266	0.392474	0.293004	0.292579
Michalewicz	[0 pi, 0 pi]	-1.77244	-1.80129	-1.2063	-1.80129	-0.82332	-1.8013	-1.8013
Quartic	[0 1.28, 0 1.28]	0	0	0	0.03	0	0	0
Total		12	7	7	10	5	10	14

optimal solutions for 14 functions, while PSO obtained solutions for 12 functions and SOS for 10 functions. Further, it is observed that DOS is the only algorithm that solves Rastrigin (function 9) and Schaffer N4 (function 14) within the allotted number of function evaluations.

4.2 Engineering design problem

Power system operation and control have taken a giant leap after the introduction of Flexible AC Transmission (FACTS) devices to control the quality of power. FACTS are power electronic devices that are connected in series or shunt or both to control one or many of the power quality problems. Static var compensators (SVC), thyristor-controlled series compensator (TCSC), static compensator (STATCOM) and unified power factor conditioner (UPFC) are the most commonly used. The advent of these devices has also led to the restructuring of the power system. Installing a FACTS device in a power system requires planning, as it involves lot of monetary investment. In addition to the selection and design of the devices, an optimal placement of FACTS in a large power system plays a major role as the placement of FACTS devices in different buses changes the performance of the system. To place a single FACTS device in the system, bio-inspired techniques like GA [14], PSO [15], simulated annealing [16], firefly [17] and ABC [18] have been used. Hybrid algorithms and analytical methods like PSO-GA [19], bacterial foraging—Nelder Mead [20] and ϵ -constraint approach with non-linear programming [21] are also employed to improve the power quality.

In the problem considered, optimal location of multi-type FACTS devices with multiple objectives is found with better voltage stability using Voltage Collapse Proximity Indicator (VCPI) and reduced real power losses. Two FACTS devices—STATCOM and UPFC—are modelled using power injection model, as it is robust and feasible when compared with current injection model. The system is investigated on the IEEE 39 bus system [22] and the Indian UPSEB-75 bus system [23]. The size of these devices is then calculated for maximum power transfer capability.

The VCPI index was proposed by [24], demonstrating the accuracy and reliability of the index. It is based on the maximum power transferred through a line (1).

$$VCPI(power) = \frac{P_j}{P_{j(max)}} = \frac{Q_j}{Q_{j(max)}} \quad (3)$$

where P_j is the power transferred to the receiving end and $P_{j(max)}$ is the maximum power that can be transferred, as calculated from the following equations.

$$P_{j(max)} = \frac{V_i^2}{Z_i} \frac{\cos \theta}{4 \cos^2 \left(\frac{\theta - \theta}{2} \right)} \quad (4)$$

$$Q_{j(max)} = \frac{V_j^2}{Z_j} \frac{\sin \theta}{4 \sin^2 \left(\frac{\theta - \theta}{2} \right)}. \quad (5)$$

The value of VCPI varies from 0 (for no load) to 1 (for voltage collapse), and hence the bus with the maximum VCPI is the critical one. It is faster to calculate VCPI when compared with L -index, in which the complex bus admittance matrix has to be inverted. Also, in contrast with L -index, VCPI includes power variations in addition to

voltage variation and hence power constraints can be included. In this optimal location problem, the location of the FACTS devices is chosen in such a way that the VCPI value is minimum, thus improving the voltage stability limit. The costs associated with the installation of statcom [25] and upfc [26] are given, respectively, in (4) and (5):

$$c_{\text{statcom}}(\$) = \text{Statcom rating in MVar} \times 50 \quad (6)$$

$$c_{\text{upfc}}(\$) = 188.2 - 0.2691 \times \text{upfc rating} + 0.0003 \times \text{upfc rating}^2. \quad (7)$$

The fitness function of the proposed multi-objective optimization problem includes VCPI index and real power loss:

$$F = f_1 \times \text{VCPI} + f_2 \times \text{real power loss} + pVD \quad (8)$$

where f_1 and f_2 are constants and p is the penalty factor, which is added based on how much the voltage deviates from the constraint set (11). The following equality and inequality constraints are considered.

Real and reactive power balance equations:

$$P_{gi} - P_{di} - \sum_{j=1}^N V_j [G_{ij} \cos(\theta_{ij}) + B_{ij} \sin(\theta_{ij})] = 0 \quad (9)$$

$$Q_{gi} - Q_{di} - \sum_{j=1}^N V_j [G_{ij} \sin(\theta_{ij}) - B_{ij} \cos(\theta_{ij})] = 0. \quad (10)$$

Considering the installation of FACTS devices:

$$P_{gi} + P_{\text{facts}_i} = P_{di} + \sum_{j=1}^n V_i V_j Y_{ij} [\cos \alpha_{ij} + \theta_i + \theta_j] \quad (11)$$

$$Q_{gi} + Q_{\text{facts}_i} = Q_{di} + \sum_{j=1}^n V_i V_j Y_{ij} [\sin(\alpha_{ij} + \theta_i - \theta_j)]. \quad (12)$$

Inequality constraints:

$$\text{voltage limits } V_{t\min} \leq V_t \leq V_{t\max} \quad (13)$$

Real power generation limits

$$P_{gt\min} \leq P_{gt} \leq P_{gt\max} \quad (14)$$

Reactive power limits

$$Q_{gt\min} \leq Q_{gt} \leq Q_{gt\max}. \quad (15)$$

The proposed DOS algorithm is first tested on the IEEE 39 bus system, with 10 generators and 46 lines. An optimal value of population size is chosen by simulating the system with different population sizes from 15 to 100. As given in table 4, a population size of 80 is fixed for obtaining the best results. Furthermore, three different loading conditions

Table 4. Variation of VCPI with number of particles.

Sl. no.	Number of particles	VCPI value
1	15	0.701103
2	30	0.683203
3	50	0.647534
4	80	0.646913
5	100	0.690722

have been used to prove the robustness of the proposed technique. The test results with the VCPI index, weak line, real power loss and size and cost of FACTS devices are given in table 5.

Under normal loading conditions, VCPI index is reduced by 14.58% when compared with PSO and by 3.15% when compared with ABC. Similarly, the real power losses are reduced by proper sizing of STATCOM and UPFC by 18.43 and 11.77% in comparison with PSO and ABC. With optimal sizing, costs of the devices have also been optimized. Similar results can be observed for 110 and 120% loading conditions. It can be seen that with normal loading, though ABC and DOS choose the same location for UPFC, the sizing is different and hence results in reduced VCPI and real power losses. Optimal metrics obtained are thus the result of proper choice of location and sizing of UPFC and STATCOM devices in the power system. To test the reliability of the proposed algorithm, it is also tested on an Indian bus system. The Uttar Pradesh State Electricity Board 75 bus system consists of 15 generating units and 98 transmission lines. The results are presented in table 6.

The performance of DOS under all the loading conditions outperforms the results of PSO and ABC. VCPI index is reduced by 8.9 and 6.26% when compared with PSO and ABC and real power losses by 4 and 1.5%, respectively. The sizing of the devices has also been optimized to result in better performance overall. The convergence characteristics of the three algorithms, for a particular case, are presented in figure 3. It is observed that PSO converges faster than ABC and DOS. But DOS converges faster than ABC and also provides better fitness function solution when compared with both PSO and ABC.

VCPI values and voltage in per unit for individual buses obtained using PSO, ABC and DOS are shown, respectively, in figures 4 and 5. It can be seen that the critical bus, line 24, has the maximum VCPI value.

DOS algorithm is compared to other popular algorithms in solving mathematical and engineering problems. The study confirms the better performance of DOS when compared with other algorithms for both the cases. The advantages of using DOS when compared with other algorithms are as follows.

Table 5. Results of IEEE 39 test system.

Algorithm/ loading (%)	Parameters									
	UPFC					STATCOM				
	VCPI	Weak line	Location	Size voltage, angle	Cost (US\$/ kVAr)	Location	Size MVar	Cost (US\$/ kVAr)	Real power loss (MW)	<i>t</i> (s)
PSO/100	0.69405	33	25–26	0.14506, 0.518758	157.0	7	1.997	99.855	29	888.21
ABC/100	0.61212	33	16–17	0.07731, 0.75904	150.7	39	1.23	61.7	28	768.67
DOS/100	0.59282	33	16–17	0.06875, 0.53968	138.9	2	0.75	37.5	26	741.23
PSO/110	0.76700	33	6–11	0.13664, 2.9004	179.6	4	2.428	121.41	34	897.21
ABC/110	0.62670	33	14–15	0.09119, 0.85095	155.4	3	1.45	72.99	33	881.23
DOS/110	0.61097	33	16–17	0.06041, 0.24531	141.1	8	0.859	42.968	31	823.05
PSO/120	0.78878	33	2–25	0.12358, 0	187.5	13	3.07	153.35	38	905.5
ABC/120	0.7210	33	25–26	0.08437, 0.33773	164.0	19	1.875	93.750	38	832.04
DOS/120	0.71320	33	17–18	0.0750, 0.58875	149.2	29	1.53	76.495	36	813.31

Table 6. Results of UPSEB 75 bus system.

Algorithm/ loading (%)	Parameters									
	UPFC					STATCOM				
	VCPI	Weak line	Location	Size voltage, angle	Cost (US\$/ kVAr)	Location	Size MVar	Cost (US\$/ kVAr)	Real power loss (MW)	<i>t</i> (s)
PSO/100	0.676995	24	19–20	0.08699, 3.14	182.548	29	4.087	204.35	189	1929
ABC/100	0.6801	24	26–27	0.34257, 0	168.03	56	3.567	156.25	187	970.7
DOS/100	0.646632	24	27–51	0.40012, 0	158.033	40	2.34375	117.187	183	956.8
PSO/110	0.825786	24	45–44	0.152417, 0.061172	186.352	28	4.82590	241.295	213	2148
ABC/110	0.80222	24	45–44	0.165625, 0	185.865	45	4.80949	240.474	208	1071
DOS/110	0.751958	24	26–29	0.1878, 2.491	182.757	16	2.7237	136.18	203	988.8
PSO/120	0.9446	24	34–54	0.19140, 1.28972	187.893	16	5.2376	261.58	228	1845.3
ABC/120	0.93783	24	36–37	0.30217, 0.37358	186.753	37	4.998	248.734	219	966.03
DOS/120	0.926871	24	45–44	0.096875, 0	185.528	16	3.5577	177.889	208	1048.9

1. Predictable behaviour:

DOS provides a single solution for a problem irrespective of the number of trials for a predefined number of particles and function evaluations.

2. Easier tuning of parameters:

In DOS, the number of particles is the only tuning parameter. Given the predictable nature of the algorithm, the tuning can be done by just varying the number of particles for each trial.

Figure 3. Convergence characteristics of different algorithms.

Figure 4. VCPI indexes of 75 bus system.

Figure 5. Voltage values of 75 bus system using different algorithms.

3. DOS shows similar characteristics to that of other meta-heuristic algorithms like PSO, SOS, GA and DE in terms of interactions of particles and movement.
4. DOS utilizes gradient information for making decisions, which is not present in most of the popular optimization algorithms.

5. Conclusion

A new algorithm called DOS inspired by robust nature of classical methods and problem solving characteristics of meta-heuristic algorithms is developed. The algorithm does not have any random numbers, which makes it extremely robust, not showing any deviations in result for any number of trials for a given problem. The algorithm was tested with benchmark mathematical functions and a power system problem and is found to be superior to recent popular algorithms. The algorithm does not have any algorithm-specific tuning parameters, which increases the stability of the algorithm and ease of application in problem solving. There is scope in initializing the particles in the search space in a deterministic way and further research can improve the results. The highly predictive nature of the algorithm opens new areas of research and can be applied to concepts like encryption.

References

- [1] Holland J 1975 *Adaptation in natural and artificial systems*. New York: University of Michigan Press
- [2] Storn R and Price K 1997 Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* 11(4): 341–359
- [3] Kennedy J and Eberhart R 1995 Particle swarm optimization. In: *Proceedings of the IEEE International Conference on Neural Networks 4*, pp. 1942–1948
- [4] Karaboga D and Basturk B 2007 A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J. Glob. Optim.* 39: 459–471
- [5] Passino K M 2002 Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Contr. Syst.* 22: 52–67
- [6] Yang X S 2010 Firefly algorithm, Levy flights and global optimization. *Research and Development in Intelligent Systems XXVI*, pp. 209–218
- [7] Cheng M Y and Prayogo D 2014 Symbiotic organisms search: a new metaheuristic optimization algorithm. *Comput. Struct.* 139: 98–112
- [8] Wang G G, Gandomi A H and Alavi A H 2014 An effective krill herd algorithm with migration operator in biogeography-based optimization. *Appl. Math. Model.* 38: 2454–2462
- [9] Gandomi A H, Yang X S and Alavi A H 2013 Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Eng Comput.* 29: 17–35

- [10] Geem Z W, Kim J H and Loganathan G V 2001 A new heuristic optimization algorithm: harmony search. *Simulation* 76: 60–68
- [11] Karaboga D and Akay B 2009 A comparative study of Artificial Bee Colony algorithm. *Appl. Math. Comput.* 214: 108–132
- [12] Solano-Aragon C and Castillo O 2014 Optimization of benchmark mathematical functions using the firefly algorithm. *Recent Advances on hybrid approaches for designing intelligent systems*, Switzerland: Springer International Publishing, vol. 547, pp. 177–189
- [13] Tang W J, Wu Q H and Saunders J R 2007 A bacterial swarming algorithm for global optimization. In: *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, pp. 1207–1212
- [14] Najafi S R, Abedi M and Hosseinian S H 2006 A novel approach to optimal allocation of SVC using genetic algorithms and continuation power flow. In: *Proceedings of the 2006 IEEE International Power and Energy Conference*, pp. 202–206
- [15] Saravanan M, Slochanal S M R, Venkatesh P and Abraham J P S 2007 Application of particle swarm optimization technique for optimal location of FACTS devices considering cost of installation and system loadability. *Electr. Power Syst. Res.* 77: 276–283
- [16] Gerbex S, Cherkaoui R and Germond A J 2003 Optimal location of FACTS devices to enhance power system security. In: *2003 IEEE Bologna Power Tech Conference Proceedings*, pp. 3–7
- [17] Rao B V and Kumar G V N 2014 Sensitivity analysis based optimal location and tuning of static VAR compensator using firefly algorithm. *Indian J. Sci. Technol.* 7: 1201–1210
- [18] Balarama Krishna Rao Y V, Srinivasa Rao R and Reddy V V K 2014 ABC algorithm based comparative analysis of optimal SVC and TCSC placement to maximize loadability. *Int. J. Electr. Electron. Eng. Res.* 4: 53–64
- [19] Kheirizad I, Mohammadi A and Varahram M H 2008 A novel algorithm for optimal location of FACTS devices in power system planning. *J. Electr. Eng. Technol.* 3: 177–183
- [20] Hooshmand R A, Morshed M J and Parastegari M 2015 Congestion management by determining optimal location of series FACTS devices using hybrid bacterial foraging and Nelder–Mead algorithm. *Appl. Soft Comput.* 28: 57–68
- [21] Lashkar Ara A, Kazemi A and Nabavi Niaki S A 2012 Multiobjective optimal location of FACTS shunt–series controllers for power system operation planning. *IEEE Trans. Power Deliv.* 27: 481–490
- [22] University of Washington n.d. *Power system test case archive*. <http://www.ee.washington.edu/research/pstca/>
- [23] Chaturvedi D K 2008 *Soft computing: techniques and its applications in electrical engineering*. Berlin, Heidelberg: Springer
- [24] Moghavvemi M and Faruque M O 1999 Estimation of voltage collapse from local measurement of line power flow and bus voltages. In: *Proceedings of the International Conference on Electric Power Engineering*, pp. 77–78
- [25] Sode-Yome A, Mithulananthan N and Lee K Y 2005 Static voltage stability margin enhancement using STATCOM, TCSC and SSSC. In: *Proceedings of the IEEE/PES Transmission and Distribution Conference and Exhibition: Asia and Pacific*, pp. 1–6
- [26] Cai L J, Erlich I and Stamtzis G 2004 Optimal choice and allocation of FACTS devices in deregulated electricity market using genetic algorithms. In: *Proceedings of the IEEE PES Power Systems Conference and Exposition*, pp. 201–207