Engineering Applications of Artificial Intelligence 87 (2020) 103330

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Barnacles Mating Optimizer: A new bio-inspired algorithm for solving engineering optimization problems☆

Mohd Herwan Sulaiman [a,*], Zuriani Mustaffa [b], Mohd Mawardi Saari [a], Hamdan Daniyal [a]

[a] Faculty of Electrical & Electronics Engineering Technology, Universiti Malaysia Pahang (UMP), 26600 Pekan Pahang, Malaysia
[b] Faculty of Computing, Universiti Malaysia Pahang (UMP), 26300 Gambang Pahang, Malaysia

ARTICLE INFO

Keywords:
Barnacles Optimization Algorithm
Benchmarked functions
Loss minimization
Meta-heuristic technique
Optimal reactive power dispatch

ABSTRACT

This paper presents a novel bio-inspired optimization algorithm namely the Barnacles Mating Optimizer (BMO) algorithm to solve optimization problems. The proposed algorithm mimics the mating behaviour of barnacles in nature for solving optimization problems. The BMO is first benchmarked on a set of 23 mathematical functions to test the characteristics of BMO in finding the optimal solutions. It is then applied to optimal reactive power dispatch (ORPD) problem to verify the reliability and efficiency of BMO. Extensive comparative studies with other algorithms are conducted and from the simulation results, it is observed that BMO generally provides better results and exhibits huge potential of BMO in solving real optimization problems.

1. Introduction

Optimization is the process of finding the best combination of variables or parameters that fulfil the constraints to achieve the objective function whether for minimization or maximization purposes. The objective function is normally formulated based on applications or problems to be solved and it can be in terms of cost, efficiency, profits etc. To solve optimization problem, there are various techniques or algorithms have been used from hard-computing until meta-heuristic approaches. The hard-computing approaches rely on gradient-based information of the involved functions to find the optimal solution. Even though such techniques are still being used by different users to solve optimization problems until today, they normally suffer from the local optima entrapment as well as ineffective for unknown problems with computationally expensive derivation (Saremi et al., 2017).

On the other hand, meta-heuristic approaches are becoming more popular especially in engineering optimization problems due to their ability to escape from the local optima with relying on simple concepts that mimic from nature and can be utilized in a wide range of problems from various disciplines. Nature-inspired meta-heuristics are fairly simple and mostly inspired by uncomplicated concepts. They can be classified into four groups: evolutionary based, swarm based, physics-based and human behaviour-based algorithms (Mirjalili and Lewis, 2016).

Evolutionary based algorithms mimic the evolutionary process in nature where the global optima are obtained by producing new offspring that inherit the properties from the parents. The set of candidate solution is improved iteratively until satisfying the terminating condition. Thus, throughout iterations or generations, the probability of achieving better results near the global optima will be increased even though it is not guaranteed to obtain a very accurate approximation of the global optima. Among the popular algorithms under this category are Genetic Algorithm (GA) (Holland, 1992; Goldberg and Holland, 1988), Differential Evolution (DE) (Storn and Price, 1997), Evolutionary Programming (Yao et al., 1999), Genetic Programming (Koza, 1990, 1999) and Evolutionary Strategies (Amoretti, 2014).

Swarm based algorithms mimic the social behaviour of groups of animals. The most popular algorithm is Particle Swarm Optimization (PSO) which is originally developed by Eberhart and Kennedy (1995) and Kennedy and Eberhart (1995, 1997). It is inspired by the movement behaviour of swarm of birds and school of fish. The movements of the particles are guided by their own best-known position in search space as well as the entire swarm's best-known position. Other swarm based algorithms that have been proposed in literature are Ant Colony Optimization (ACO) (Colorni et al., 1999; Dorigo et al., 2006), Artificial Bee Colony (ABC) (Karaboga and Akay, 2009; Karaboga and Basturk, 2007, 2008), Firefly Algorithm (Yang, 2010), Ant–Lion Optimizer (ALO) (Mirjalili, 2015a), Grey Wolf Optimizer (GWO) (Mirjalili et al., 2014), Dragonflies Algorithm (DA) (Mirjalili, 2016a), Grasshoppers Optimization Algorithm (GOA) (Saremi et al., 2017), Satin Bowerbird Optimizer (SBO) (Samareh Moosavi and Khatibi Bardsiri, 2017) and many more. It can be said that the most common behaviour of

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.engappai.2019.103330.

* Corresponding author.
E-mail addresses: herwan@ump.edu.my, mherwan@ieee.org (M.H. Sulaiman).

https://doi.org/10.1016/j.engappai.2019.103330
Received 7 June 2018; Received in revised form 30 September 2019; Accepted 28 October 2019
Available online 7 November 2019

swarm-based algorithms is mimicking the food foraging and socializing behaviours.

The third category is called physics-based algorithms which mimic the physical rules in their algorithm's development. Every algorithm is inspired by different nature and physic law such as Gravitational Search Algorithm (GSA) (Rashedi et al., 2009) which mimics the concept of Newtons' gravity law, Big-Bang Big-Crunch (BBBC) (Erol and Eksin, 2006) that mimics the evolution of the universe, Black Hole (BH) (Hatamlou, 2013) algorithm mimics the black hole phenomenon, Lightning Search Algorithm (LSA) (Shareef et al., 2015) that inspired by the natural phenomenon of lightning and the mechanism of step leader propagation, Multi-Verse Optimization (MVO) (Mirjalili et al., 2016) that mimics the concepts of cosmology that consists of white hole, black hole and wormhole, Yin–Yang-pair Optimization (YYPO) (Punnathanam and Kotecha, 2016) which is inspired by the Chinese philosophy of duality of opposite forces in nature and many more.

Finally, the fourth category is called human-based algorithms which mimic the human behaviour in life such as learning, competition, socializing etc. Among the algorithms that fall under this category are Harmony Search Algorithm (HSA) (Lee and Geem, 2005) that is inspired by the improvisation process of jazz musicians, Teaching Learning Based Optimization (TLBO) (Ouyang et al., 2015; Rao et al., 2011; Zou et al., 2015) that mimics teaching and learning in classroom, Volleyball Premier League Algorithm (VPA) (Moghdani and Salimifard, 2018) which is motivated by the volleyball competition and interaction among volleyball teams during a season, Soccer League Competition (SLC) algorithm (Moosavian, 2015; Moosavian and Kasaee Roodsari, 2014) based on the professional soccer leagues competition and Imperialist Competition Algorithm (ICA) (Ardalan et al., 2015) which is based on a new socio-politically motivated global search strategy. The algorithm inspired from technical traders in the stock market called Fibonacci Indicator Algorithm (FIA) has been proposed in Etminaniesfahani et al. (2018).

From all algorithms that have been proposed in literature, it can be said that there are two main issues in proposing new algorithms, which are (1) no free lunch theorem (NFL) and (2) determination of exploitation and exploration processes. The NFL theorem (Wolpert and Macready, 1997) has logically proved that there are no meta-heuristic algorithms best suited for solving all optimization problems. It means that one algorithm might be excellent on a particular set of problems, but when it tested on a different set of problems, contradict performance may be obtained. NFL makes this field of study very active which indirectly encourage researchers to propose new algorithms continuously. The second issue is how the proposed algorithm balance in determining the process of exploitation and exploration in searching towards the global optima. Every algorithm has unique mechanism to require the search agents to tune the level of exploration to exploitation and vice versa. Too much exploitation may lead the algorithm trapped into local optima while too much exploration may bring the algorithm to nowhere. Thus, it is a difficult task to find a proper balance between exploitation and exploration due to stochastic nature of optimization process (Mirjalili and Lewis, 2016).

This paper proposes a new bio-inspired meta-heuristic algorithm namely Barnacle Mating Optimizer (BMO) which mimics the mating behaviour of barnacles. BMO can be categorized into group of evolutionary algorithms. To the knowledge of the authors, there is no previous study on this subject in the optimization literature. The effectiveness of the BMO is demonstrated and evaluated by using 23 mathematical test functions as well as real application of power system problem and the results demonstrate the BMO is very competitive compared to the state-of-the-art optimization algorithms. It is worth to mention that the initial works related the development of BMO have been published in Sulaiman et al. (2018a,b) and the application of BMO into economic dispatch problems can be found in Sulaiman et al. (2019). In this paper, the extension works of BMO in presenting the properties of BMO as well as the superiority of BMO into solving the Optimal Reactive Power Dispatch (ORPD) problem will be discussed.

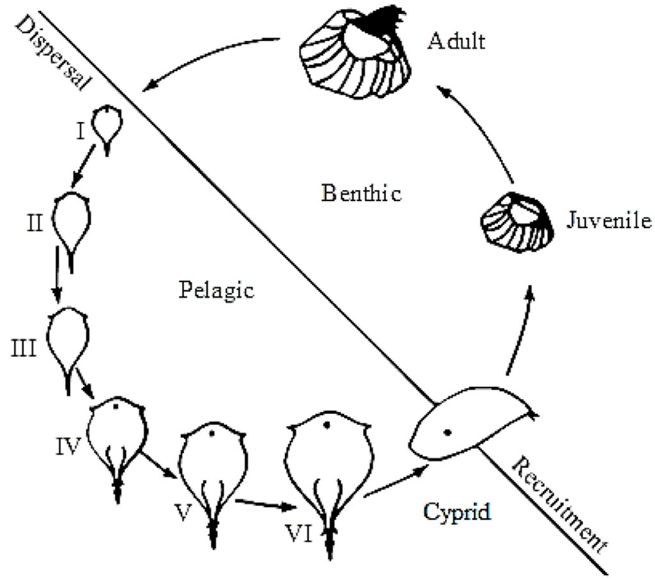

Fig. 1. Life cycle of barnacles (Khandeparker and Anil, 2007).

The rest of the paper is organized as follows: Section 2 discusses the BMO development. It is followed by the simulation studies in Section 3. Finally, Section 4 states the conclusion of this paper.

2. Barnacles mating optimizer algorithm

2.1. Inspiration

Barnacles are micro-organisms existed since Jurassic times. Barnacles can swim at births and when they reach adult stage, they attach themselves to objects in the water and grow a shell. Most of the barnacles are hermaphroditic, which means that they have both male and female reproductions. There are more than 1400 species of barnacles found in the world's waterways and the most common are called acorn barnacles.

The most interesting fact about barnacles is they are famous for their long penises, including some of the longest in animals relative to their body (Barazandeh et al., 2013), which is seven to eight times the length of their bodies in order to cope with the changing tides and sedentary lifestyle. A barnacle's mating group consists of all the neighbours within reach of its penis and all its potential competitors for mates. Variation in penis reach may have an important role in determining mating group size and local mate competition (Hoch, 2008). Fig. 1 shows the life cycle of barnacles. The larval development of barnacles includes six naupliar instars and a non-feeding pre-settling cyprid instar. The cyprid after settling on a surface, metamorphoses into an adult (Khandeparker and Anil, 2007).

Another unique characteristic of barnacles is their capability for tenacious underwater adhesion by secreting proteinaceous substances. However, in this paper, the mating process of barnacles will be used as an inspiration in developing a new novel optimization algorithm.

2.2. Hardy–Weinberg principle

The Hardy–Weinberg principle (Crow, 1999; Guo and Thompson, 1992) will be used in the off-spring generation of BMO. In the simplest case with two alleles of D and M that represent the *Dad* and *Mum* with frequencies $f(\mathrm{D}) = p$ and $f(\mathrm{M}) = q$, respectively, the expected genotype frequencies under normal mating can be expressed as $f(DD) = p^2$ for the DD homozygotes, $F(MM) = q^2$ for the MM homozygotes and $f(DM) = 2pq$ for heterozygotes. The different way to form the

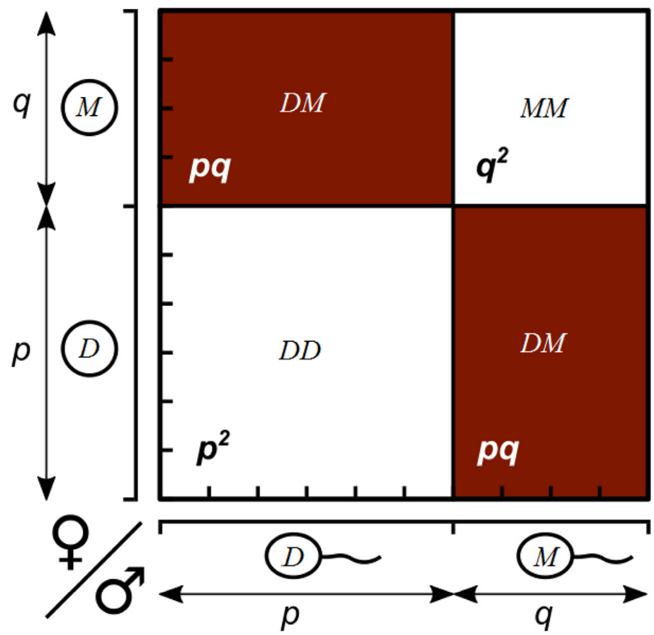

Fig. 2. Punnett square.

genotypes for the next off-spring generation can be shown in a *Punnett square*, as exhibited in Fig. 2.

It can be seen from the figure that p =0.6, q =0.4 which gives the area of rectangle represents genotype frequencies as follows: *DD* : *DM* : *MM* = 0.36 ; 0.48 : 0.16. The sum of the entries is $p^2 + 2pq + q^2 = 1$. It can be seen also that $p + q$ =1. Thus, for simplification, the generation of new off-spring is based on the p and q of the barnacles' parents which will be explained in the next sub section.

2.3. BMO

2.3.1. Initialization

In the proposed BMO, it is assumed that the candidate of solution is barnacles where the vector of population can be expressed as follows:

$$X = \begin{bmatrix} x_1^1 & \cdots & x_1^N \\ \vdots & \ddots & \vdots \\ x_n^1 & \cdots & x_n^N \end{bmatrix} \tag{1}$$

where N is the number of control variables and n is the number of population or number of barnacles. The control variables in Eq. (1) is subject to the upper and lower bounds of the problem to be solved as follow:

$$ub = [ub_1, \ldots, ub_i] \tag{2}$$

$$lb = [lb_1, \ldots, lb_i] \tag{3}$$

where *ub* and *lb* indicate the upper and lower bounds of *i*th variable. The evaluation of the vector X is done initially, and the sorting process is performed to locate the best solution so far at the top of the vector X.

2.3.2. Selection process

The proposed BMO is using the different approach for the selection to be mated compared to other evolutionary algorithms such as GA, DE etc. since the selection of two barnacles is based on the length of their penises, *pl*. The selection process mimics the behaviour of barnacles which are based on the following assumptions:

(i) The selection process is done randomly but it will be restricted to the penis length of the barnacle, *pl*.

(ii) Each barnacle may contribute its sperm as well as to receive sperm from other barnacle and each barnacle only can be fertilized by one barnacle only at one time even though in real life, the female can probably be fertilized by more than one male (Barazandeh et al., 2013).

(iii) If at the certain point, the selection process selects the same barnacle which means that self-mating or self-fertilization supposed to be happened. From Yusa et al. (2012), self-mating is very rare even though barnacle has both male and female reproductions (most of barnacles not self-mating), thus in this paper, self-mating is not will be considered where at this point, no new off-spring will be generated.

(iv) If the selection at the certain iteration is more than the *pl* that has been set, the sperm cast process is happened.

It can be noted that from above assumptions, the processes of exploitation (points no. 1 and 2) and exploration (points no 4) are enforced in this algorithm. The selection of ten barnacles' mating process can be visualized in Fig. 3.

It can be seen from Fig. 3 that the best solution so far is located at the top of the candidates of solution X. Let say that the maximum penis length of barnacles is seven times longer than its size (pl =7), thus at a certain iteration, barnacle #1 is only able to mate with one of the barnacles #2–#7 only. If barnacle #1 selects barnacle #8, it is over the limit, thus the normal mating process does not occur. So, the offspring generation is proceeded by sperm cast process (exploration) which will be explained later. Of course, this is just how the algorithm is operated in terms of virtual distance and not specifically related to real distance of the barnacles. The following simple selection is used which are expressed in mathematical forms:

$$barnacle\_d = randperm(n) \tag{4}$$

$$barnacle\_m = randperm(n) \tag{5}$$

where the *barnacle_d* and *barnacle_m* are the parents to be mated and n is the number of population.

Eqs. (4) and (5) show that the selection is made randomly and fulfil the assumption number 1 in the previous sub section.

2.3.3. Reproduction

The reproduction process proposed in BMO is slightly different compared to other evolutionary algorithms. Since there are no specific equations or formulas to derive the reproduction of barnacles, the BMO is mainly emphasizing on the inheritance characteristics or genotype frequencies of barnacles' parents in producing the offspring based on Hardy–Weinberg principle. To show the simplicity of the proposed BMO, the following expressions are proposed to produce new variables of offspring from barnacles' parents:

$$x_i^{N\_new} = p x_{barnacle\_d}^N + q x_{barnacle\_m}^N \tag{6}$$

where p is the normally distributed pseudo random numbers between [0, 1], $q = (1 - p)$, $x_{barnacle\_d}^N$ and $x_{barnacle\_m}^N$ are the variables of *Dad* and *Mum* of barnacles respectively which has been selected in Eqs. (4) and (5). It can be said that p and q represent the percentage of characteristic of *Dad* and *Mum* that embedded in the generation of new offspring. Thus, the offspring inherits the behaviours of *Dad* and*Mum* based on probability of random number between 0 to 1. For easy illustration, let us say p is 0.6 (randomly generated), it means that 60% of the *Dad's* characters and 40% of *Mum's* characters are embedded in the new offspring generation.

It is worth to highlight that the value of *pl* plays an important role in determining the exploitation and exploration processes. If the selection of barnacles to be mated is within the range of the penis length of *Dad's barnacle*, the exploitation process is occurred (Eq. (6)). As been mentioned in selection process, the sperm cast is treated as the exploration process in BMO. Sperm cast is happened when the selection of barnacles to be mated exceeds the value of *pl* that has been set

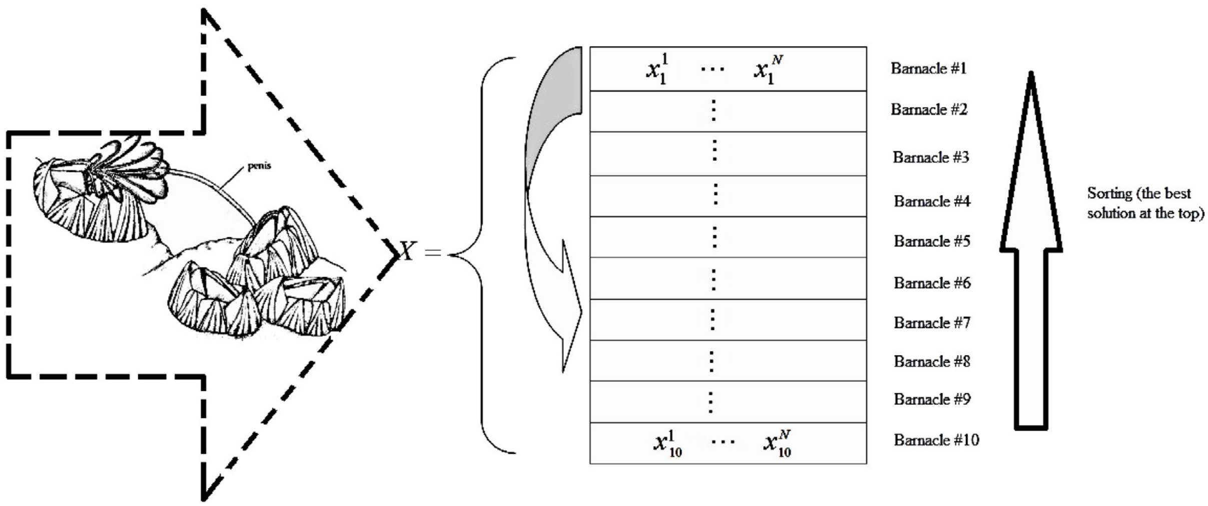

Fig. 3. Selection of mating process of BMO (figure of barnacles adopted from Brusca and Brusca, 2002).

initially (refer to Section 2.3.2). The sperm cast process is expressed as follows:

$$x_i^{n\_new} = rand() \times x_{barnacle\_m}^{n} \tag{7}$$

where $rand()$ is the random number between [0, 1].

It can be noted that Eq. (7) shows the simple approach of the barnacle's off spring to be evolved. The new off spring is generated from the *Mum's* barnacle for the exploration process. This is due to the new offspring is generated by *Mum's* barnacle since it receives the sperm from the water that has been released by the other barnacles elsewhere.

The pseudo code of BMO is shown in Fig. 4. The BMO starts the optimization process by creating a set of random solutions. New off spring of barnacles are generated based on Eqs. (6)–(7). The best solution so far is updated in each iteration where it is located at the top of the vector X. In order to control the matrix expansion from the population size, each new off spring of barnacles is evaluated and merged with the parents. From here, the sorting process is done to select the half of the top solution that fit the population size. The poor results are assumed to be dead and eliminated.

```
Initialize the population of barnacles Xi
Calculate the fitness of each barnacle
Sorting to locate the best result at the top of the population
(T=the best solution)
while (I < Maximum iterations)
    Set the value of pl
    Selection using equations (4) and (5):
    if selection of Dad and Mum = pl
        for each variable
            Off spring generation using equation (6):
        end for
    else if selection of Dad and Mum >pl
        for each variable
            Off spring generation using equation (7):
        end for
    end if
    Bring the current barnacle back if it goes outside the
    boundaries
    Calculate the fitness of each barnacles
    Sorting and update T if there is a better solution
    I=I+1
end while
Return T
```

Fig. 4. Pseudo code of the BMO.

3. Simulation studies

In this section, several simulation studies are carried out to demonstrate the effectiveness of the proposed BMO. Firstly, the performance of BMO has been conducted on the 23 benchmark functions (Saremi et al., 2017; Mirjalili and Lewis, 2016; Mirjalili, 2015a; Mirjalili et al., 2014; Mirjalili, 2016a; Mirjalili and Lewis, 2016; Mirjalili, 2015b). These test functions can be classified into three categories viz. unimodal (F1–F7), multimodal (F8–F13) and composite test functions (F14–F23). The unimodal functions are tested for benchmarking the exploitation of algorithms due to these functions only have one global optima. The multimodal and composite functions on the other hand, have many local optima which make them suitable for benchmarking the performance of algorithms in avoiding local optima as well as exploration evaluation (Saremi et al., 2017).

For solving all test functions, 30 search agents and 500 iterations were employed. Each test function was run 30 times to generate the statistical results. For a fair comparison, all compared algorithms were simulated using the same framework with a same computer of similar performances and characteristics. Then, different performance indicators were used to quantitatively compared the algorithms which is average and standard deviation of the best solutions. Qualitative results including convergence curves, search history and average fitness of population have been illustrated and analysed in the next subsection. For comparison, seven recent algorithms were used namely Salp Swarm Algorithm (SSA) (Mirjalili et al., 2017), Grasshopper Optimization Algorithm (GOA) (Saremi et al., 2017), Whale Optimization Algorithm (WOA) (Mirjalili and Lewis, 2016), Sine Cosine Algorithm (Mirjalili, 2016b), Dragonfly Algorithm (DA) (Mirjalili, 2016a), Moth-Flame Optimizer (MFO) (Mirjalili, 2015b) and Ant–Lion Optimizer (ALO) (Mirjalili, 2015a). Comparison with GA and PSO also have been done in this paper.

3.1. Effect of pl in BMO

The most important issue in optimization algorithm is how the algorithm works in deciding the exploitation and exploration processes. Too much exploitation may lead to stuck in local optima while too much exploration may end up with not achieving the global optima. Thus, in BMO, the tuning of pl plays an important role in determining the exploitation and exploration processes. To show the effect of tuning the pl, a sphere function (F1) with dimension of d =2 is utilized. 10

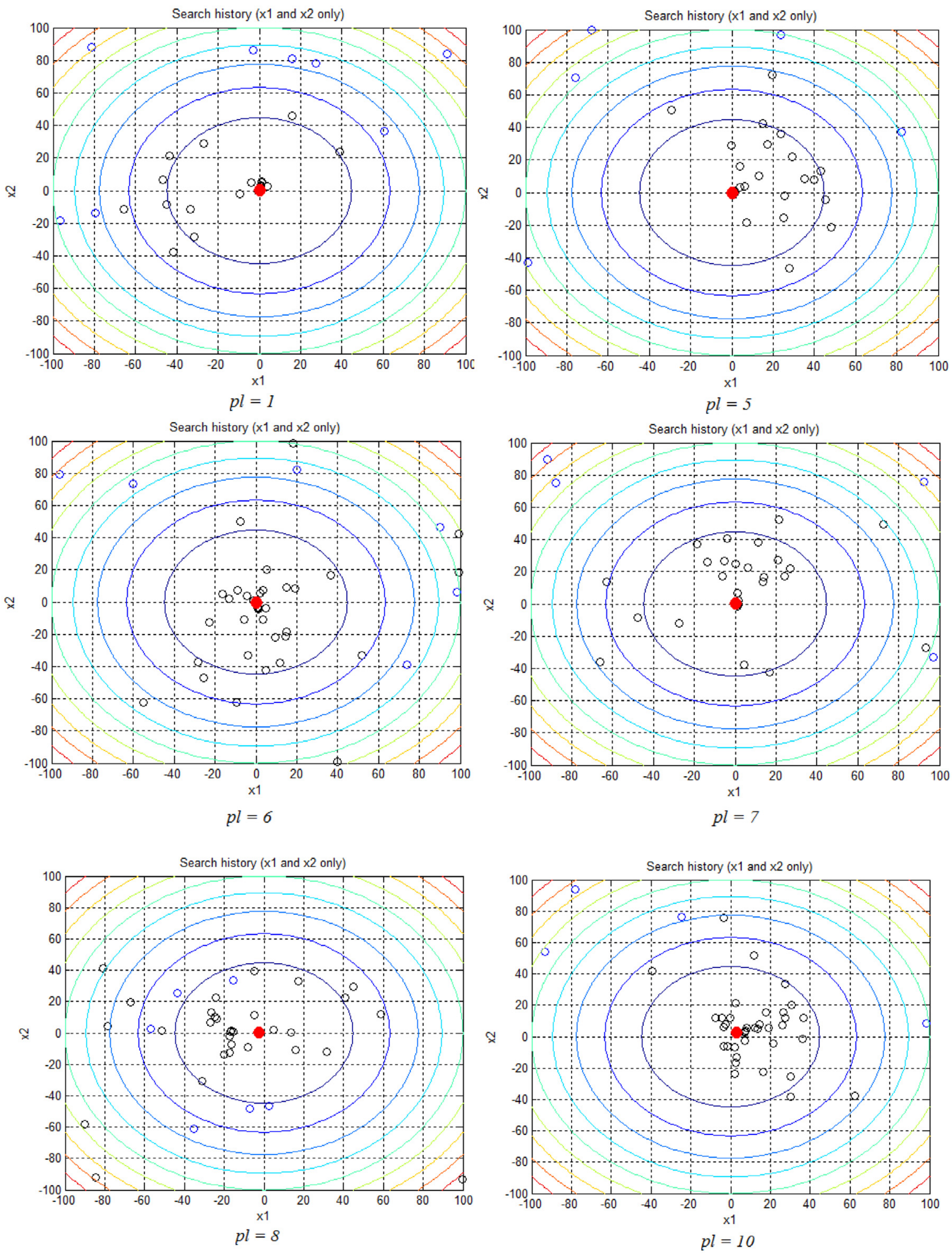

Fig. 5. Effects of tuning the *pl*. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

barnacles are set for the population, and the *pl* is set for 1, 5, 6, 7, 8 and 10 as shown in Fig. 5. The *pl* =1 indicates that the barnacle can only mate with the nearest neighbour having a distance of 1 from the barnacle. For example, after performing the evaluation of initial population, and sorting has been done, the selection process is generated from Eqs. (4) and (5). The generation of the new offspring using Eq. (6) is only occurred for adjacent barnacles such barnacle #2 can only be mated with barnacles #1 and #3 only. If this selection did not happen, the new off-spring is generated using Eq. (7). If the *pl* =10, this means all barnacles can be mated and Eq. (7) is never been applied.

It is worth to highlight that the blue markers of 'o' show the initial locations of barnacles while the closed red markers show the best location of barnacle. For this simulation, only 10 iterations are set. It can be seen that for *pl* = 1, too much exploration process occurred and on the other hand, too much exploitation takes place when setting *pl* =10. Meanwhile, when *pl* = 5, 6 and 7, it can be seen that the exploitation

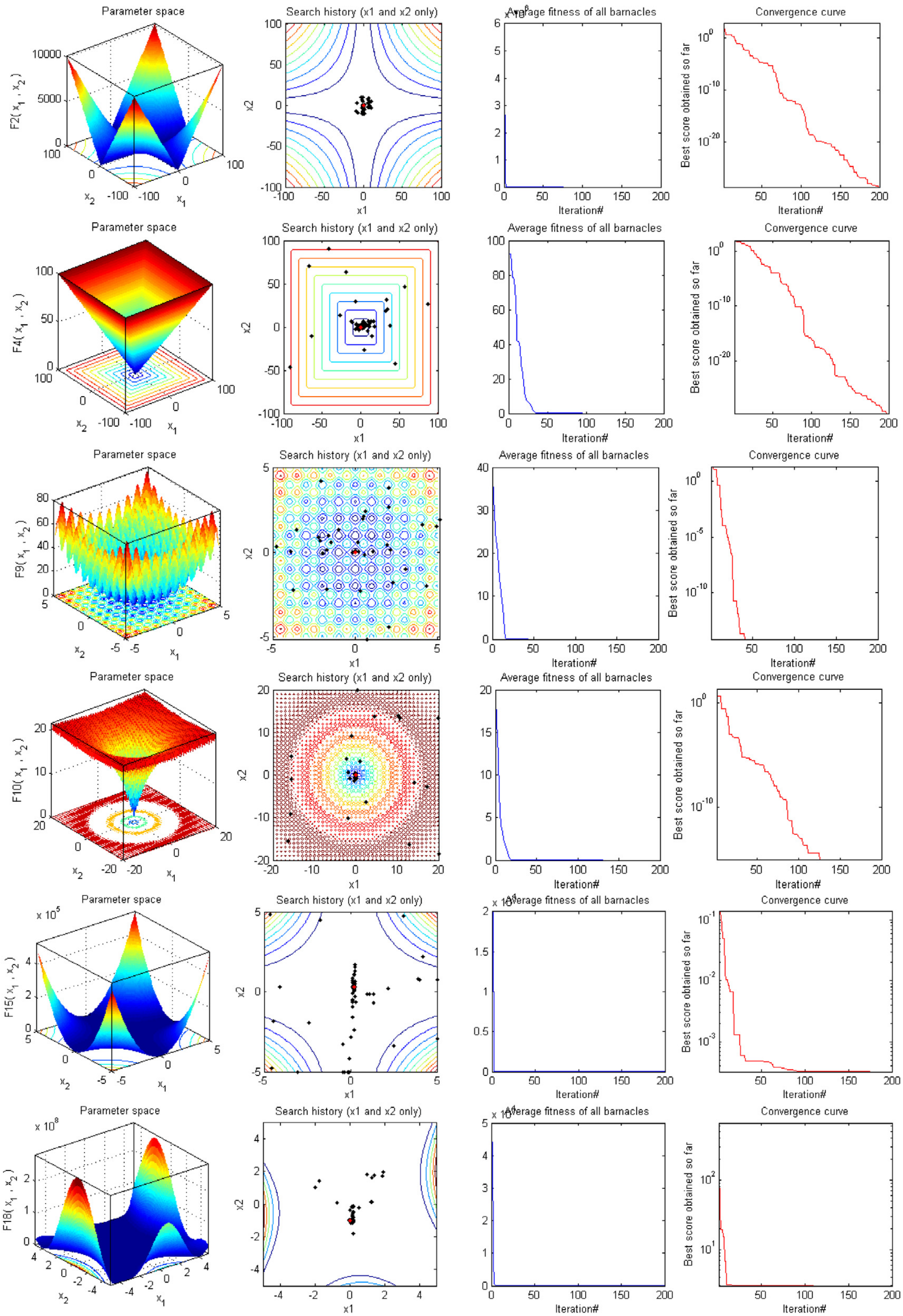

Fig. 6. Performance of BMO in unimodal, multi-modal and composite test functions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

and exploration processes are balance in order to obtain the global optima. While for *pl* =8, it can be seen that the searching process is weighing more on exploitation process where the best solution is missed from the optimal value (red closed marker). Thus, it is suggested that the selection of *pl* can be set ranging from 50 to 70% from the total population size in order to balance the exploitation and exploration processes. Nevertheless, it is still depending on the different problems to be applied in tuning the value of *pl*.

Table 1
Results of unimodal benchmark functions.

Function		ALO	DA	GOA	MFO	SSA	SCA	WOA	GA	PSO	BMO
F1	Best	2.10E−04	2.30E+02	2.05E−01	2.57E−01	2.26E−08	3.61E−03	2.58E−87	2.04E−01	3.63E−14	**4.72E−149**
	Worst	8.50E−03	5.61E+03	4.91E+02	1.00E+04	1.53E−06	1.90E+02	3.04E−72	1.64E+00	1.09E−05	**1.43E−124**
	Mean	1.33E−03	1.76E+03	1.18E+02	1.67E+03	2.81E−07	1.66E+01	1.60E−73	7.05E−01	7.85E−07	**5.37E−126**
	Std	1.65E−03	1.14E+03	1.49E+02	3.79E+03	3.94E−07	3.71E+01	6.20E−73	3.65E−01	2.68E−06	**2.61E−125**
F2	Best	2.48E+00	7.00E+00	2.52E+00	3.02E−01	7.39E−02	1.24E−04	4.80E−58	7.50E−02	2.59E−03	**5.18E−79**
	Worst	1.18E+02	3.47E+01	2.25E+03	9.00E+01	5.77E+00	1.19E−01	4.81E−50	3.13E−01	4.16E−01	**5.53E−67**
	Mean	3.95E+01	1.58E+01	1.56E+02	3.83E+01	2.34E+00	1.95E−02	2.37E−51	1.81E−01	7.03E−02	**2.17E−68**
	Std	4.33E+01	7.46E+00	3.99E+02	2.75E+01	1.74E+00	2.91E−02	8.82E−51	6.57E−02	9.65E−02	**1.01E−67**
F3	Best	2.09E+03	5.86E+02	1.79E+02	4.99E+03	4.74E+02	1.84E+03	1.35E+04	3.48E+03	1.31E+01	**5.45E−140**
	Worst	9.78E+03	3.05E+04	1.01E+04	6.51E+04	4.91E+03	1.75E+04	7.34E+04	1.88E+04	8.22E+02	**3.83E−110**
	Mean	4.58E+03	1.23E+04	1.76E+03	2.40E+04	1.63E+03	8.56E+03	4.91E+04	1.19E+04	1.14E+02	**2.96E−111**
	Std	1.77E+03	7.50E+03	1.95E+03	1.20E+04	1.01E+03	4.82E+03	1.45E+04	3.68E+03	1.52E+02	**9.60E−111**
F4	Best	9.24E+00	1.82E+01	2.67E+00	5.24E+01	4.98E+00	9.00E+00	4.10E−01	8.23E+00	6.72E−01	**3.65E−79**
	Worst	3.25E+01	5.88E+01	1.07E+01	8.14E+01	2.09E+01	5.63E+01	8.78E+01	3.12E+01	4.56E+00	**5.33E−59**
	Mean	1.93E+01	3.14E+01	5.59E+00	6.87E+01	1.30E+01	3.25E+01	5.42E+01	1.93E+01	2.36E+00	**2.32E−60**
	Std	5.08E+00	8.91E+00	2.14E+00	8.24E+00	3.98E+00	1.21E+01	2.50E+01	5.90E+00	1.09E+00	**9.85E−60**
F5	Best	2.51E+01	5.61E+03	5.52E+01	1.34E+02	2.75E+01	1.22E+02	2.74E+01	8.92E+01	**1.13E+01**	2.68E+01
	Worst	1.59E+03	1.26E+06	6.79E+04	9.43E+04	1.07E+03	3.27E+05	2.87E+01	3.28E+03	1.73E+02	**2.80E+01**
	Mean	2.99E+02	4.19E+05	3.31E+03	1.94E+04	2.07E+02	3.64E+04	2.81E+01	5.22E+02	5.63E+01	**2.72E+01**
	Std	4.10E+02	3.18E+05	1.23E+04	3.67E+04	2.86E+02	6.75E+04	**4.25E−01**	8.82E+02	4.38E+01	3.12E−01
F6	Best	8.97E−05	6.76E+02	4.70E−01	3.91E−01	2.62E−08	4.59E+00	8.61E−02	9.48E−02	**1.59E−12**	9.14E−03
	Worst	1.17E−02	4.90E+03	6.02E+02	2.02E+04	**4.80E−07**	1.64E+02	8.93E−01	2.11E+00	1.01E−04	5.81E−01
	Mean	1.73E−03	2.60E+03	1.27E+02	3.36E+03	**1.23E−07**	1.74E+01	4.24E−01	8.55E−01	3.94E−06	8.29E−02
	Std	2.60E−03	1.12E+03	1.49E+02	5.50E+03	**1.05E−07**	2.85E+01	2.15E−01	4.87E−01	1.85E−05	1.28E−01
F7	Best	1.05E−01	1.51E−01	2.04E−01	1.02E−01	4.17E−02	9.03E−03	**5.37E−05**	4.00E−02	1.04E−02	1.59E−04
	Worst	5.37E−01	1.70E+00	9.83E−01	1.94E+01	3.84E−01	7.17E−01	1.85E−02	1.74E−01	5.85E−02	**2.80E−03**
	Mean	2.63E−01	5.61E−01	5.03E−01	2.85E+00	1.73E−01	1.17E−01	3.82E−03	8.80E−02	2.76E−02	**9.00E−04**
	Std	9.45E−02	3.27E−01	1.83E−01	5.52E+00	6.70E−02	1.43E−01	4.85E−03	3.29E−02	1.23E−02	**6.61E−04**

3.2. Qualitative results and discussion

The first simulation was performed on the 2D of the first 2 variables for the test functions using the proposed BMO. The main objective of this simulation was for behaviour observation of BMO. Selected test functions from 3 categories have been depicted in Fig. 6 to show the performance of the proposed BMO. As shown in Fig. 6, unimodal test function (F2 and F4) only has one global optima solution, multi-modal (F9 and F10) and composite (F15 and F18) test functions have many local optima which make them suitable for testing the developed algorithms in terms of local optima avoidance and exploration evaluations. The parameter setting for these results are: population of barnacles = 10, maximum iteration = 200 and $pl = 7$.

As exhibited in Fig. 6, generation of the new off-spring of barnacles tends to explore the promising regions of search space around the global optima eventually. This pattern can be seen in all test functions. The results show that the BMO balances the exploitation and exploration to generate a new off-spring towards the global optima solution. It is worth to highlight that only the top solutions are selected for the next iteration to maintain the population size from growing bigger. Thus, it is look like the barnacle is moving towards the global optima but actually the barnacles are reproduced and those with good fitness are maintained in the search history as clearly shown such in function F2. The red dot indicates the global optimum is achieved.

3.3. Quantitative results and discussion

To show the effectiveness of the proposed BMO, all test functions F1 to F23 have been carried out and compared with the recent algorithms in the literature that has been described previously. The simulation results are tabulated in Tables 1–3 for the unimodal, multi-modal and composite test functions respectively.

By referring to Table 1, BMO has gained the best performance compared to other algorithms. The best results are marked in bold for better visualization. The results shown in this table depicted that BMO substantially better in most of the unimodal tests functions. Unimodal test functions have only one global optimum, so the results clearly show that the BMO has better exploitation ability. The results in Table 2 are consistent with those tabulated in Table 1, where BMO is outperformed other algorithms significantly. The multi-modal test functions have a significant number of local solutions. From the simulations that have been conducted, the results depicted in Table 2 quantitatively show the effectiveness of the proposed BMO in avoiding local solution during optimization.

The results of the identified algorithms on composite test functions are presented in Table 3. It can be seen that BMO provides competitive results compared to the others. Composite test functions are the combination of many real search space and different test functions which offer challenging test beds (Mirjalili et al., 2017). This make the optimizer of these test functions require a proper balance between exploitation and exploration which is closed to the real practical optimization problems. From the highlighted results tabulated in Table 4, it shows that BMO is properly balance in exploitation and exploration processes especially for test functions F16 to F20.

3.4. Application of BMO in real application: ORPD problem

To further show the effectiveness of the proposed BMO, the application of BMO into the optimal reactive power dispatch (ORPD) problem has been conducted and compared with the recent algorithms in the literature.

The basis of formulating ORPD problem can be first described as follows:

Minimize $f(x,u)$

$$\text{s.t}\quad \begin{matrix} g(x,u)=0 \\ h(x,u)\le 0 \end{matrix} \tag{8}$$

where the function of $f(x,u)$ is the objective function, $g(x,u) = 0$ is the equality constraint, $h(x,u) \le 0$ is the inequality constraint, x is the vector of dependent variables and u is the vector of control variables. In this paper, the objective function to be minimized is the total transmission loss, F which is expressed as follow (Khazali and Kalantar, 2011):

$$F = P_{Loss}(x,u) = \sum_{L=1}^{Nl} P_{Loss} \tag{9}$$

Table 2
Results of multi-modal benchmark functions.

Function		ALO	DA	GOA	MFO	SSA	SCA	WOA	GA	PSO	BMO
F8	Best	−1.63E+03	−1.55E+03	−1.62E+03	−1.62E+03	−1.66E+03	−1.14E+03	**−1.91E+03**	−1.76E+03	−1.56E+03	−1.74E+03
	Worst	−1.63E+03	−9.47E+02	−1.08E+03	−1.08E+03	−1.31E+03	−7.62E+02	−1.63E+03	−1.32E+03	−1.04E+03	−1.64E+03
	Mean	−1.63E+03	−1.17E+03	−1.38E+03	−1.38E+03	−1.50E+03	−8.98E+02	**−1.86E+03**	−1.53E+03	−1.28E+03	−1.68E+03
	Std	**6.94E−13**	1.32E+02	1.35E+02	1.35E+02	8.59E+01	8.89E+01	9.89E+01	1.11E+02	1.26E+02	2.67E+01
F9	Best	8.36E+01	1.16E+02	7.78E+01	8.50E+01	7.66E+01	1.46E−01	**0.00E+00**	5.85E−01	4.18E+01	**0.00E+00**
	Worst	2.55E+02	3.74E+02	3.33E+02	3.95E+02	1.63E+02	1.42E+02	**0.00E+00**	8.91E+00	1.33E+02	**0.00E+00**
	Mean	1.65E+02	2.23E+02	1.80E+02	1.98E+02	1.18E+02	6.49E+01	**0.00E+00**	3.68E+00	7.53E+01	**0.00E+00**
	Std	4.05E+01	5.67E+01	6.80E+01	8.18E+01	2.22E+01	4.14E+01	**0.00E+00**	1.86E+00	2.43E+01	**0.00E+00**
F10	Best	8.88E−16	2.00E+01	2.00E+01	2.00E+01	2.22E+00	2.02E+01	8.88E−16	2.00E+01	3.79E+00	**8.88E−16**
	Worst	1.87E+01	2.06E+01	2.04E+01	2.00E+01	2.00E+01	2.05E+01	7.99E−15	2.00E+01	2.03E+01	**8.88E−16**
	Mean	4.42E+00	2.02E+01	2.02E+01	2.00E+01	1.50E+01	2.03E+01	4.44E−15	2.00E+01	1.94E+01	**8.88E−16**
	Std	5.10E+00	1.68E−01	8.88E−02	0.00E+00	7.82E+00	7.70E−02	2.29E−15	7.76E−04	2.96E+00	**4.01E−31**
F11	Best	7.39E−03	1.13E+00	1.76E−02	2.70E−02	1.66E−04	4.05E−04	0.00E+00	1.83E−02	9.29E−13	**0.00E+00**
	Worst	8.69E−02	2.04E+00	1.12E+00	3.50E+00	4.75E−02	1.02E+00	1.40E−01	1.38E−01	4.43E−02	**0.00E+00**
	Mean	3.55E−02	1.48E+00	6.68E−01	1.05E+00	1.21E−02	5.28E−01	4.66E−03	4.94E−02	1.39E−02	**0.00E+00**
	Std	2.28E−02	2.40E−01	3.41E−01	1.31E+00	1.18E−02	3.08E−01	2.55E−02	2.32E−02	1.43E−02	**0.00E+00**
F12	Best	4.58E+00	2.74E+00	1.20E+00	1.93E−01	6.44E−01	7.80E−01	4.72E−03	1.67E−05	**4.84E−17**	5.72E−03
	Worst	3.06E+01	7.68E+01	5.86E+00	1.28E+01	6.35E+00	4.23E+01	4.43E−02	1.04E−01	4.15E−01	**2.13E−01**
	Mean	9.19E+00	1.91E+01	3.10E+00	4.90E+00	4.03E+00	6.78E+00	1.39E−02	3.61E−03	**2.42E−02**	4.17E−02
	Std	5.00E+00	1.65E+01	1.31E+00	2.81E+00	1.34E+00	8.54E+00	8.54E−03	**1.89E−02**	8.02E−02	4.02E−02
F13	Best	1.05E+00	3.32E+05	4.83E−01	1.38E+01	3.36E−03	3.15E+00	2.69E−01	8.52E−02	**1.10E−02**	4.64E−02
	Worst	7.78E+01	1.47E+08	6.24E+02	1.11E+05	5.19E+01	1.77E+07	1.25E+00	**8.96E−01**	3.61E+00	2.97E+00
	Mean	2.92E+01	2.66E+07	4.53E+01	7.23E+03	2.14E+01	1.07E+06	7.56E−01	**2.58E−01**	9.06E−01	5.49E−01
	Std	2.45E+01	3.29E+07	1.13E+02	2.33E+04	1.33E+01	3.37E+06	2.50E−01	**1.94E−01**	1.11E+00	6.74E−01

Table 3
Results of composite benchmark functions.

Function		ALO	DA	GOA	MFO	SSA	SCA	WOA	GA	PSO	BMO
F14	Best	9.98E−01	9.98E−01	**9.98E−01**	9.98E−01	9.98E−01	3.61E−03	9.98E−01	**9.98E−01**	9.98E−01	9.98E−01
	Worst	1.08E+01	3.97E+00	**9.98E−01**	8.84E+00	3.97E+00	1.90E+02	1.08E+01	**9.98E−01**	1.64E+01	7.87E+00
	Mean	2.48E+00	1.26E+00	**9.98E−01**	2.05E+00	1.43E+00	1.66E+01	3.06E+00	**9.98E−01**	5.22E+00	2.45E+00
	Std	2.41E+00	7.33E−01	**4.52E−16**	1.81E+00	9.26E−01	3.71E+01	3.58E+00	**4.52E−16**	3.74E+00	1.95E+00
F15	Best	6.54E−04	6.98E−04	7.41E−04	3.11E−04	4.24E−04	1.24E−04	3.09E−04	3.94E−04	**3.07E−04**	3.08E−04
	Worst	2.04E−02	2.26E−02	9.55E−02	8.33E−03	2.04E−02	1.19E−01	**2.81E−03**	2.07E−02	2.04E−02	2.04E−02
	Mean	2.39E−03	2.86E−03	1.25E−02	1.23E−03	1.57E−03	1.95E−02	**8.21E−04**	5.76E−03	2.45E−03	1.97E−03
	Std	4.93E−03	5.41E−03	2.08E−02	1.39E−03	3.57E−03	2.91E−02	**6.88E−04**	7.25E−03	6.08E−03	5.00E−03
F16	Best	**−1.03E+00**	−1.03E+00	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	1.84E+03	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**
	Worst	**−1.03E+00**	−1.03E+00	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	1.75E+04	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**
	Mean	**−1.03E+00**	−1.03E+00	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	8.56E+03	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**
	Std	**6.78E−16**	1.83E−05	**6.78E−16**	**6.78E−16**	**6.78E−16**	4.82E+03	**6.78E−16**	**6.78E−16**	**6.78E−16**	**6.78E−16**
F17	Best	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	9.00E+00	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**
	Worst	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	5.63E+01	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**
	Mean	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	3.25E+01	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**
	Std	**1.69E−16**	**1.69E−16**	**1.69E−16**	**1.69E−16**	**1.69E−16**	1.21E+01	1.22E−05	3.65E−06	**1.69E−16**	**1.69E−16**
F18	Best	**3.00E+00**	**3.00E+00**	3.00E+00	**3.00E+00**	**3.00E+00**	3.00E+00	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**
	Worst	**3.00E+00**	**3.00E+00**	8.40E+01	**3.00E+00**	**3.00E+00**	3.00E+00	**3.00E+00**	**3.00E+01**	**3.00E+00**	**3.00E+00**
	Mean	**3.00E+00**	**3.00E+00**	8.40E+00	**3.00E+00**	**3.00E+00**	3.00E+00	**3.00E+00**	**3.90E+00**	**3.00E+00**	**3.00E+00**
	Std	**0.00E+00**	**0.00E+00**	2.06E+01	**0.00E+00**	**0.00E+00**	5.35E−05	1.72E−04	4.93E+00	**0.00E+00**	**0.00E+00**
F19	Best	**−3.00E−01**	**−3.00E−01**	**−3.00E−01**	**−3.00E−01**	**−3.00E−01**	**−3.00E−01**	**−3.00E−01**	**−3.00E−01**	**−3.00E−01**	**−3.00E−01**
	Worst	**−3.00E−01**	**−3.00E−01**	**−7.44E−09**	**−3.00E−01**	**−3.00E−01**	**−3.00E−01**	**−3.00E−01**	**−3.00E−01**	**−3.00E−01**	**−3.00E−01**
	Mean	**−3.00E−01**	**−3.00E−01**	**−1.39E−01**	**−3.00E−01**	**−3.00E−01**	**−3.00E−01**	**−3.00E−01**	**−3.00E−01**	**−3.00E−01**	**−3.00E−01**
	Std	**1.13E−16**	**1.13E−16**	**1.27E−01**	**1.13E−16**	**1.13E−16**	**1.13E−16**	**1.13E−16**	**1.13E−16**	**1.13E−16**	**1.13E−16**
F20	Best	**−3.32E+00**	**−3.32E+00**	**−3.32E+00**	**−3.32E+00**	**−3.32E+00**	−3.00E+00	**−3.32E+00**	**−3.32E+00**	**−3.32E+00**	**−3.32E+00**
	Worst	−3.12E+00	−1.66E−01	−1.09E−04	−1.84E+00	−8.67E−01	−7.78E−03	−5.61E−02	−3.20E+00	−3.20E+00	−3.09E+00
	Mean	−3.28E+00	−2.71E+00	−1.84E+00	−3.15E+00	−2.86E+00	−6.43E−01	−2.61E+00	−3.28E+00	−3.27E+00	−3.26E+00
	Std	6.76E−02	8.96E−01	1.20E+00	2.79E−01	7.18E−01	8.27E−01	1.03E+00	5.83E−02	6.03E−02	8.17E−02
F21	Best	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	**−5.95E+00**	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01
	Worst	−2.63E+00	−2.63E+00	−2.63E+00	−2.63E+00	−2.63E+00	−3.51E−01	−8.82E−01	−2.63E+00	−2.63E+00	−2.63E+00
	Mean	−6.80E+00	−6.61E+00	−4.39E+00	−6.89E+00	−7.81E+00	−1.72E+00	−8.13E+00	−5.00E+00	−5.98E+00	−5.84E+00
	Std	3.11E+00	2.61E+00	2.79E+00	3.23E+00	3.22E+00	1.63E+00	2.78E+00	3.46E+00	3.55E+00	2.30E+00
F22	Best	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	**−6.32E+00**	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01
	Worst	−1.84E+00	−2.77E+00	−1.84E+00	−2.75E+00	−2.75E+00	**−5.22E−01**	−2.76E+00	−2.75E+00	−1.84E+00	−2.75E+00
	Mean	−7.14E+00	−7.37E+00	−4.88E+00	−7.24E+00	−9.23E+00	**−3.61E+00**	−6.99E+00	−6.63E+00	−6.74E+00	−7.10E+00
	Std	3.22E+00	2.93E+00	3.19E+00	3.50E+00	2.69E+00	**1.82E+00**	3.07E+00	3.66E+00	3.78E+00	3.01E+00
F23	Best	−1.05E+01	−1.05E+01	−1.05E+01	−1.05E+01	−1.05E+01	**−5.29E+00**	−1.05E+01	−1.05E+01	−1.05E+01	−1.05E+01
	Worst	−2.42E+00	−2.43E+00	−1.68E+00	−2.43E+00	−2.42E+00	**−9.42E−01**	−2.80E+00	−2.42E+00	−2.43E+00	−2.43E+00
	Mean	−6.00E+00	−7.77E+00	−5.93E+00	−8.41E+00	−8.79E+00	**−3.55E+00**	−8.45E+00	−6.06E+00	−7.49E+00	−7.06E+00
	Std	3.15E+00	3.07E+00	3.89E+00	3.32E+00	3.22E+00	**1.51E+00**	2.77E+00	3.75E+00	3.61E+00	3.41E+00

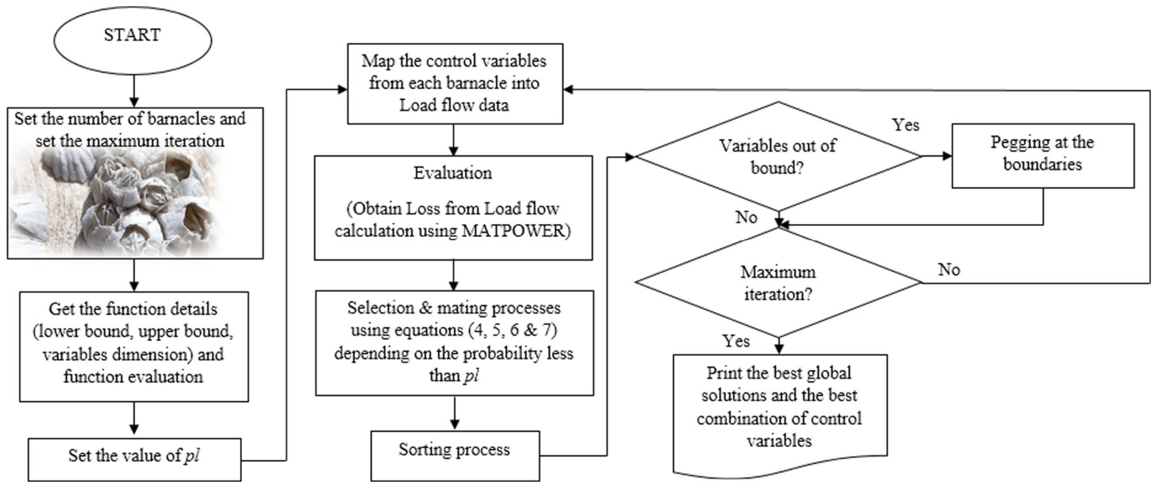

Fig. 7. Flow of proposed BMO for solving ORPD.

The equality constraint equations suggested in Khazali and Kalantar (2011) are still valid to give the power balanced of load flow, as follows:

$$P_{Gi} - P_{Di} = V_i \sum_{j \in N_i} V_j \left(G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij} \right) \tag{10}$$

$$Q_{Gi} - Q_{Di} = V_i \sum_{j \in N_i} V_j \left(B_{ij} \cos\theta_{ij} - G_{ij} \sin\theta_{ij} \right) \tag{11}$$

On the other hand, the inequality constraints can be represented in terms of operating constraints, as follow:

- Generator constraints: Real and reactive power generation as well as generation bus voltages are restricted by their upper and lower limits, as follow:

$$P_{Gi}^{\min} \le P_{Gi} \le P_{Gi}^{\max} \qquad i = 1, \ldots, N_G \tag{12}$$

$$Q_{Gi}^{\min} \le Q_{Gi} \le Q_{Gi}^{\max} \qquad i = 1, \ldots, N_G \tag{13}$$

$$V_{Gi}^{\min} \le V_{Gi} \le V_{Gi}^{\max} \qquad i = 1, \ldots, N_G \tag{14}$$

  where N_G is the number of generators.
- Transformer tap setting are restricted by their lower and upper limits, as follows:

$$T_i^{\min} \le T_i \le T_i^{\max} \qquad i = 1, \ldots, N_T \tag{15}$$

  where N_T is the of transformers.
- Reactive compensators (Shunt VARs) are restricted by their limits as follows:

$$Q_{Ci}^{\min} \le Q_{Ci} \le Q_{Ci}^{\max} \qquad i = 1, \ldots, N_C \tag{16}$$

  where N_C is the number of the shunt compensators.

It is vital to highlight that a different approach has been taken in order to obtain the objective function in this paper. To ensure the accurate result of total transmission loss and no violation of the constraints can be achieved smoothly, load flow programme by MATPOWER software package (Zimmerman et al., 2011) is used to assist the analysis.

In order to find the objective function (evaluation process), each position is mapped into the load flow data and the load flow programme is executed to obtain the loss. The implementation of BMO for solving the ORPD problem is depicted in Fig. 7.

All simulations for solving ORPD problem using BMO and other compared algorithms are implemented using MATLAB on a Windows 7 Professional Intel® Core ™ i3-2330M CPU @ 2.20 GHz 6 GB RAM. The algorithm of BMO has been implemented on the IEEE-30 bus system and the results are compared with various state-of-the-art techniques in literature. The case study of the IEEE 30-bus system is based on the Khazali and Kalantar (2011) which consists of 13 control variables that needed to be optimized.

Table 4
Limit setting for control variables for IEEE 30-bus system.

Variables	Lower limit	Upper limit
Voltages	0.9 p.u	1.1 p.u
Tap setting	0.95 p.u	1.05 p.u
Reactive compensation devices	−12 MVar	36 MVar

The IEEE 30-bus system consists of six generators, 41 lines, four transformers that are located at lines 6–9, 4–12, 9–12 and 27–28 as shown in Fig. 8. For this case study, three reactive compensation devices are placed at buses 3, 10 and 24. Setting for the minimum and maximum boundaries for the transformer's tap setting, reactive compensation devices and generators voltages are tabulated in Table 4. For this case study, demands are set as follows:

$$P_{Load} = 2.832 \text{ p.u} \quad Q_{Load} = 1.262 \text{ p.u.}$$

For a fair comparison, all identified algorithms are simulated using the similar approach to assess the total transmission losses. Table 5 shows the results of the ORPD problem solution using BMO with other algorithms. From this table, it can be seen that the optimize results of the control variables obtained by BMO produces the lowest power loss among all algorithms. The second best is MFO (Ng Shin Mei et al., 2017) which produces a relatively very small difference with the proposed BMO. The worst result for this test system is obtained using SCA where about 11.6% difference from the result that obtained by BMO. The base case without applying any optimization approach was 5.663 MW compared to the application of BMO gave 23% loss reduction for this IEEE-30 bus system.

Fig. 9 shows the convergence curve and searching history of BMO in optimizing the control variables to obtain the minimum loss. It is worth to highlight that the results presented in this figure is only the best results of 30 barnacles of population among 30 free running simulations. From this figure, it can be noted that 150 iterations were adequate to obtain a good combination of control variables of ORPD. Fig. 10 shows the performance of BMO compared to the other algorithms in terms of the convergence curve. For this case study, the value of *pl* is set to 21 where it is 70% from the total population. This value is obtained by experimentally. The effect of *pl* to obtain optimal results of ORPD is shown in Fig. 11. It can be seen that value of pl =21 gave the best results compared to pl = 7 and 15. The detail of the best and worst results of the convergence curve for BMO is exhibited in Fig. 12.

To further illustrate the effectiveness of the proposed BMO in solving ORPD problem, the performance of BMO for 30 free running of

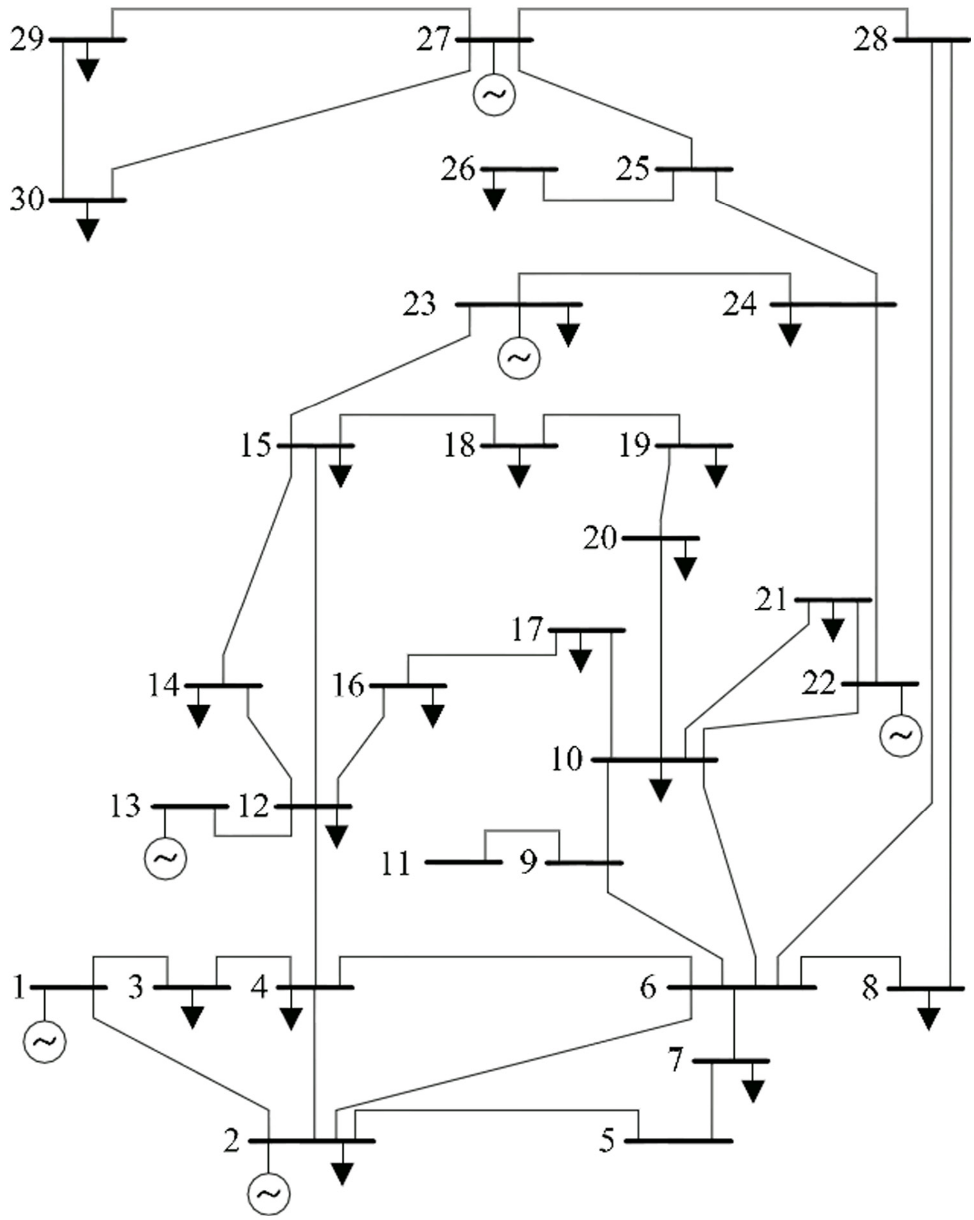

Fig. 8. IEEE 30-bus system (30-Bus System (IEEE Test Case), 2015).

Table 5
Results of control variables after optimization by BMO and other algorithms for IEEE-30 bus system with 13 control variables.

Control device	ALO	DA	GOA	MFO	SSA	SCA	WOA	BMO
V_1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
V_2	1.095366	1.1	1.093792	1.0946	1.094116	1.1	1.1	1.0943
V_5	1.077282	1.094105	1.073721	1.0756	1.073362	1.1	1.082122	1.0749
V_8	1.079279	1.1	1.075878	1.0772	1.076706	1.1	1.081394	1.0766
V_{11}	1.094156	1.094268	1.021233	1.0868	1.1	0.950055	1.028407	1.1
V_{13}	1.099956	1.1	1.1	1.1	1.095382	1.066935	1.070859	1.1
T_1	1.016513	1.05	0.973391	1.0411	0.978213	1.044791	1.022904	1.032
T_2	1.018778	1.05	1.009651	0.95007	0.979477	1.05	1.030829	0.95
T_3	1.024616	1.05	0.960947	0.95541	1.049999	1.05	1.002609	0.95456
T_4	0.982881	1.05	0.962777	0.95754	0.993189	0.95	1.027666	0.95698
Q_1	1.080679	−12	10.67912	7.1032	−4.86056	1.417035	7.341233	7.3492
Q_2	15.09492	36	36	30.796	−11.6389	33.83627	35.48301	26.368
Q_3	9.263296	7.229794	10.08084	9.8981	3.435665	13.23732	12.14184	9.8463
Loss	4.63289	4.939243	4.619063	4.5864	4.782662	5.119641	4.722495	**4.5862**

simulations is depicted in Fig. 13. The best, average and worst results are 4.5862 MW, 4.6366 MW and 4.9046 MW respectively. It can be said that BMO offers stable and consistent results for loss minimization of ORPD problem which is superior compared to other identified algorithms.

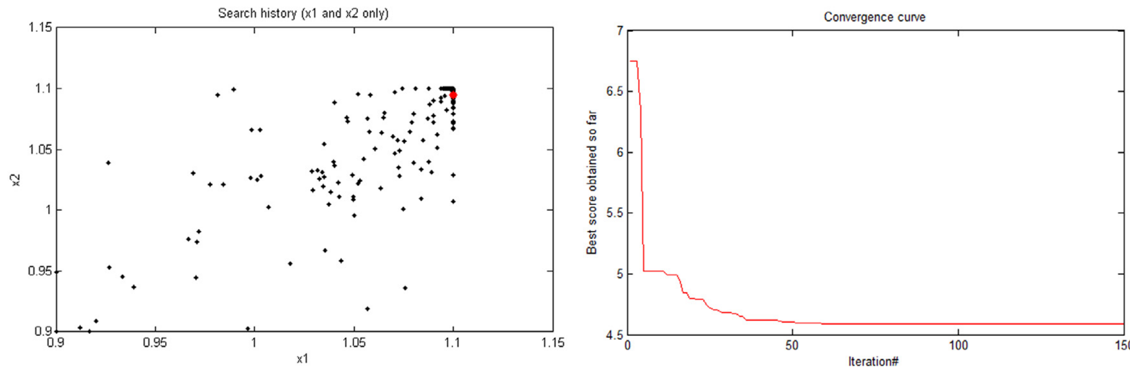

Fig. 9. Performance of BMO to solve ORPD for IEEE-30 bus system.

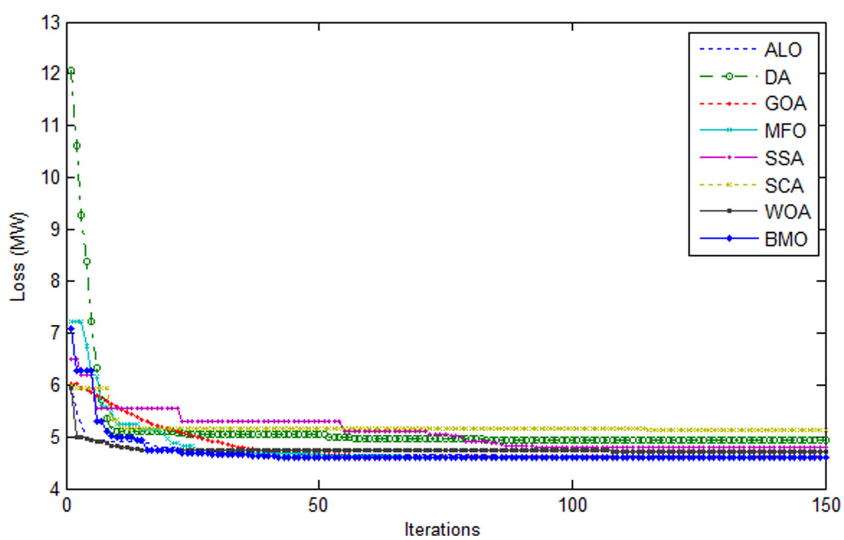

Fig. 10. Convergence curve for all algorithms applied in solving ORPD problem.

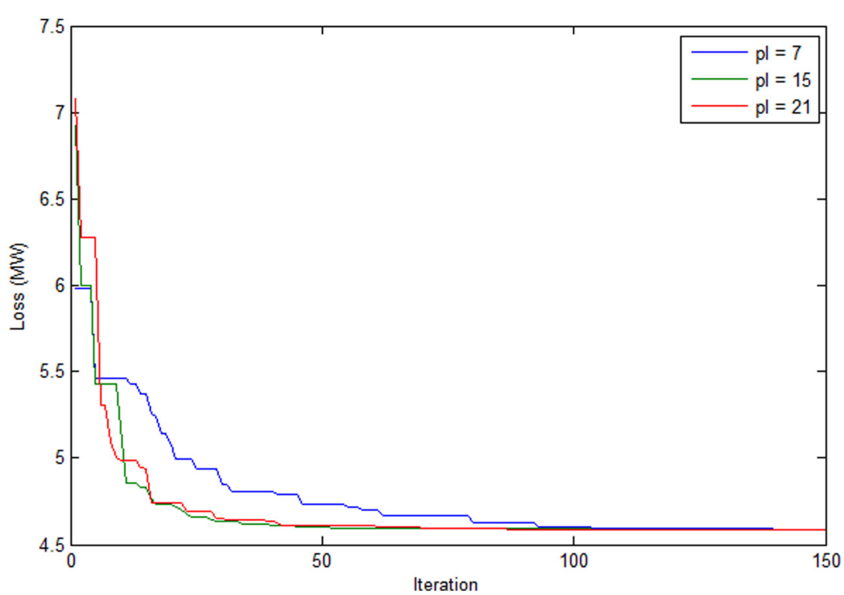

Fig. 11. Effect of *pl* in solving ORPD problem for IEEE 30 bus system.

4. Conclusion

This paper proposed a novel bio-inspired optimization algorithm inspired by barnacles' life. The proposed algorithm mimicked the mating behaviour of barnacles. 23 test functions were employed to benchmark the performance of proposed algorithm in terms of exploitation, exploration and convergence properties. The results showed that BMO was able to provide very competitive results compared to recent algorithms

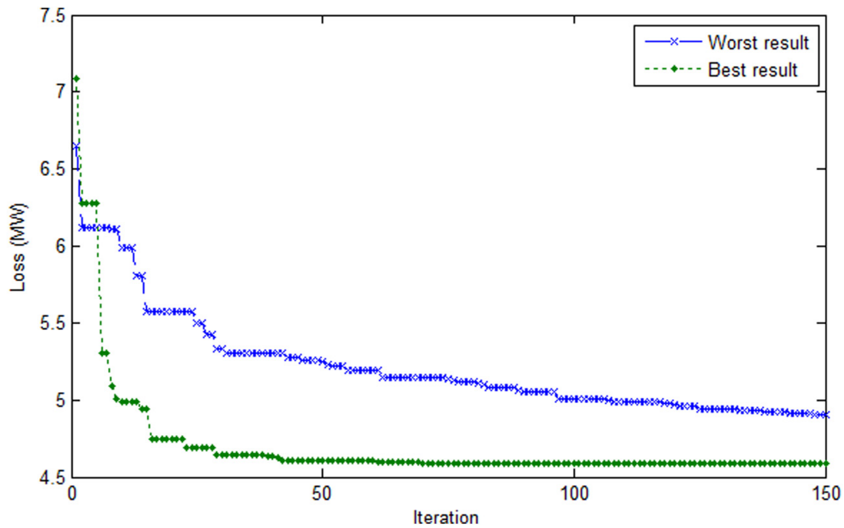

Fig. 12. Performance for the best and worst results for BMO in solving ORPD problem.

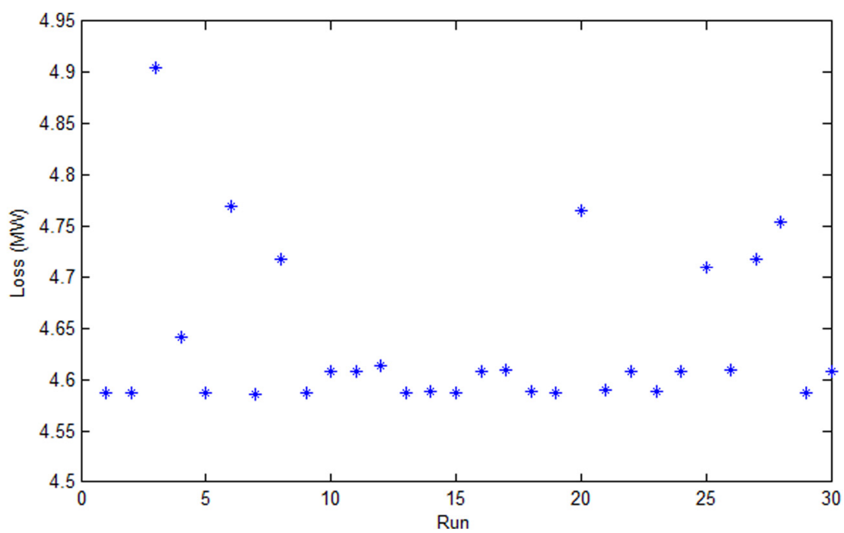

Fig. 13. Performance for 30 barnacles for 30 free running of simulations of IEEE-30 bus system.

namely ALO, DA, GOA, MFO, SCA, SSA and WOA as well as with GA and PSO in obtaining the global optima for unimodal functions, exploration ability for multi-modal functions as well as ability to avoid local optima in composite functions. Moreover, the results of real power system engineering problem viz. ORPD also showed that BMO is superior compared to those selected algorithms. The development of binary and multiobjectives version of BMO can be proposed in the near future. The MATLAB code for BMO is available at http://ee.ump.edu.my/herwan/index.php/research/barnacles-mating-optimizer-bmo.

Acknowledgement

This study is supported by the Ministry of Higher Education Malaysia and Universiti Malaysia Pahang under Fundamental Research Grant Scheme Grant FRGS/1/2017/ICT02/UMP/02/3 & #RDU170105.

References

30-Bus System (IEEE Test Case), 2015. [WWW Document]. URL https://www.al-roomi.org/power-flow/30-bus-system. (Accessed 26 September 2019).

Amoretti, M., 2014. Evolutionary strategies for ultra-large-scale autonomic systems. Inf. Sci. (Ny) 274, 1–16. http://dx.doi.org/10.1016/j.ins.2014.03.003.

Ardalan, Z., Karimi, S., Poursabzi, O., Naderi, B., 2015. A novel imperialist competitive algorithm for generalized traveling salesman problems. Appl. Soft Comput. 26, 546–555. http://dx.doi.org/10.1016/j.asoc.2014.08.033.

Barazandeh, M., Davis, C.S., Neufeld, C.J., Coltman, D.W., Palmer, A.R., 2013. Something Darwin didn't know about barnacles: spermcast mating in a common stalked species. Proc. R. Soc. B Biol. Sci. 280, 201229. http://dx.doi.org/10.1098/rspb.2012.2919.

Brusca, G., Brusca, R., 2002. Reproduction & amp; Development [WWW Document]. URL http://www.uas.alaska.edu/arts_sciences/naturalsciences/biology/tamone/catalog/arthropoda/balanus_glandula/reproduction_and_development.html.

Colorni, A., Dorigo, M., Maniezzo, V., 1999. Distributed optimization by ant colonies. In: European Conference on Artificial Life, pp. 134–142.

Crow, J.F., 1999. Hardy, Weinberg and language impediments. Genetics 152, 821, LP–825.

Dorigo, M., Birattari, M., Stutzle, T., 2006. Ant colony optimization. IEEE Comput. Intell. Mag. 1, 28–39. http://dx.doi.org/10.1109/MCI.2006.329691.

Eberhart, R., Kennedy, J., 1995. A new optimizer using particle swarm theory. In: Micro Machine and Human Science, 1995. MHS '95. Proceedings of the Sixth International Symposium on. pp. 39–43. http://dx.doi.org/10.1109/MHS.1995.494215.

Erol, O.K., Eksin, I., 2006. A new optimization method: Big Bang–Big Crunch. Adv. Eng. Softw. 37, 106–111. http://dx.doi.org/10.1016/j.advengsoft.2005.04.005.

Etminaniesfahani, A., Ghanbarzadeh, A., Marashi, Z., 2018. Fibonacci indicator algorithm: A novel tool for complex optimization problems. Eng. Appl. Artif. Intell. 74, 1–9. http://dx.doi.org/10.1016/j.engappai.2018.04.012.

Goldberg, D.E., Holland, J.H., 1988. Genetic algorithms and machine learning. Mach. Learn. 3, 95–99. http://dx.doi.org/10.1023/A:1022602019183.

Guo, S.W., Thompson, E.A., 1992. Performing the exact test of Hardy–Weinberg proportion for multiple alleles. Biometrics 48, 361–372. http://dx.doi.org/10.2307/2532296.

Hatamlou, A., 2013. Black hole: A new heuristic optimization approach for data clustering. Inf. Sci. (Ny) 222, 175–184. http://dx.doi.org/10.1016/j.ins.2012.08.023.

Hoch, J.M., 2008. Variation in penis morphology and mating ability in the acorn barnacle, Semibalanus balanoides. J. Exp. Marine Biol. Ecol. 359, 126–130. http://dx.doi.org/10.1016/j.jembe.2008.03.002.

Holland, J.H., 1992. Genetic algorithms. Sci. Am. 267, 66–73.

Karaboga, D., Akay, B., 2009. A comparative study of Artificial Bee Colony algorithm. Appl. Math. Comput. 214, 108–132. http://dx.doi.org/10.1016/j.amc.2009.03.090.

Karaboga, D., Basturk, B., 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Global Optim. 39, 459–471. http://dx.doi.org/10.1007/s10898-007-9149-x.

Karaboga, D., Basturk, B., 2008. On the performance of artificial bee colony (ABC) algorithm. Appl. Soft Comput. 8, 687–697. http://dx.doi.org/10.1016/j.asoc.2007.05.007.

Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. In: Neural Networks, 1995. Proceedings. IEEE International Conference on, Vol. 4, pp. 1942–1948. http://dx.doi.org/10.1109/ICNN.1995.488968.

Kennedy, J., Eberhart, R.C., 1997. A discrete binary version of the particle swarm algorithm. In: 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Vol. 5, pp. 4104–4108. http://dx.doi.org/10.1109/ICSMC.1997.637339.

Khandeparker, L., Anil, A.C., 2007. Underwater adhesion: The barnacle way. Int. J. Adhesion Adhes. 27, 165–172. http://dx.doi.org/10.1016/j.ijadhadh.2006.03.004.

Khazali, A.H., Kalantar, M., 2011. Optimal reactive power dispatch based on harmony search algorithm. Int. J. Electr. Power Energy Syst. 33, 684–692. http://dx.doi.org/10.1016/j.ijepes.2010.11.018.

Koza, J.R., 1990. Genetically breeding populations of computer programs to solve problems in artificial intelligence. In: [1990] Proceedings of the 2nd International IEEE Conference on Tools for Artificial Intelligence. pp. 819–827. http://dx.doi.org/10.1109/TAI.1990.130444.

Koza, J.R., 1999. Darwinian invention and problem solving by means of genetic programming. In: Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on, Vol. 3. pp. 604–609. http://dx.doi.org/10.1109/ICSMC.1999.823281.

Lee, K.S., Geem, Z.W., 2005. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput. Methods Appl. Mech. Engrg. 194, 3902–3933. http://dx.doi.org/10.1016/j.cma.2004.09.007.

Mirjalili, S., 2015a. The ant lion optimizer. Adv. Eng. Softw. 83, 80–98. http://dx.doi.org/10.1016/j.advengsoft.2015.01.010.

Mirjalili, S., 2015b. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowl.-Based Syst. 89, 228–249. http://dx.doi.org/10.1016/j.knosys.2015.07.006.

Mirjalili, S., 2016a. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput. Appl. 27, 1053–1073. http://dx.doi.org/10.1007/s00521-015-1920-1.

Mirjalili, S., 2016b. SCA: A sine cosine algorithm for solving optimization problems. Knowl.-Based Syst. 96, 120–133. http://dx.doi.org/10.1016/j.knosys.2015.12.022.

Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H., Mirjalili, S.M., 2017. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. Adv. Eng. Softw. 114, 163–191. http://dx.doi.org/10.1016/j.advengsoft.2017.07.002.

Mirjalili, S., Lewis, A., 2016. The whale optimization algorithm. Adv. Eng. Softw. 95, 51–67. http://dx.doi.org/10.1016/j.advengsoft.2016.01.008.

Mirjalili, S., Mirjalili, S.M., Hatamlou, A., 2016. Multi-Verse Optimizer: a nature-inspired algorithm for global optimization. Neural Comput. Appl. 27, 495–513. http://dx.doi.org/10.1007/s00521-015-1870-7.

Mirjalili, S., Mirjalili, S.M., Lewis, A., 2014. Grey wolf optimizer. Adv. Eng. Softw. 69, 46–61. http://dx.doi.org/10.1016/j.advengsoft.2013.12.007.

Moghdani, R., Salimifard, K., 2018. Volleyball premier league algorithm. Appl. Soft Comput. 64, 161–185. http://dx.doi.org/10.1016/j.asoc.2017.11.043.

Moosavian, N., 2015. Soccer league competition algorithm for solving knapsack problems. Swarm Evol. Comput. 20, 14–22. http://dx.doi.org/10.1016/j.swevo.2014.10.002.

Moosavian, N., Kasaee Roodsari, B., 2014. Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks. Swarm Evol. Comput. 17, 14–24. http://dx.doi.org/10.1016/j.swevo.2014.02.002.

Ng Shin Mei, R., Sulaiman, M.H., Mustaffa, Z., Daniyal, H., 2017. Optimal reactive power dispatch solution by loss minimization using moth-flame optimization technique. Appl. Soft Comput. J. 59, http://dx.doi.org/10.1016/j.asoc.2017.05.057.

Ouyang, H., Gao, L., Kong, X., Zou, D., Li, S., 2015. Teaching-learning based optimization with global crossover for global optimization problems. Appl. Math. Comput. 265, 533–556. http://dx.doi.org/10.1016/j.amc.2015.05.012.

Punnathanam, V., Kotecha, P., 2016. Yin-Yang-pair optimization: A novel lightweight optimization algorithm. Eng. Appl. Artif. Intell. 54, 62–79. http://dx.doi.org/10.1016/j.engappai.2016.04.004.

Rao, R.V, Savsani, V.J., Vakharia, D.P., 2011. Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. Comput. Des. 43, 303–315. http://dx.doi.org/10.1016/j.cad.2010.12.015.

Rashedi, E., Nezamabadi-pour, H., Saryazdi, S., 2009. GSA: A gravitational search algorithm. Inf. Sci. (Ny) 179, 2232–2248. http://dx.doi.org/10.1016/j.ins.2009.03.004.

Samareh Moosavi, S.H., Khatibi Bardsiri, V., 2017. Satin bowerbird optimizer: A new optimization algorithm to optimize ANFIS for software development effort estimation. Eng. Appl. Artif. Intell. 60, 1–15. http://dx.doi.org/10.1016/j.engappai.2017.01.006.

Saremi, S., Mirjalili, S., Lewis, A., 2017. Grasshopper optimisation algorithm: Theory and application. Adv. Eng. Softw. 105, 30–47. http://dx.doi.org/10.1016/j.advengsoft.2017.01.004.

Shareef, H., Ibrahim, A.A., Mutlag, A.H., 2015. Lightning search algorithm. Appl. Soft Comput. 36, 315–333. http://dx.doi.org/10.1016/j.asoc.2015.07.028.

Storn, R., Price, K., 1997. Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. 11, 341–359. http://dx.doi.org/10.1023/A:1008202821328.

Sulaiman, M.H., Mustaffa, Z., Aliman, O., 2019. An application of barnacles mating optimizer for solving economic dispatch problems. In: 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT). pp. 835–839. http://dx.doi.org/10.1109/JEEIT.2019.8717377.

Sulaiman, M.H., Mustaffa, Z., Saari, M.M., Daniyal, H., Daud, M.R., Razali, S., Mohamed, A.I., 2018a. Barnacles mating optimizer: A bio-inspired algorithm for solving optimization problems. In: 2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD). pp. 265–270. http://dx.doi.org/10.1109/SNPD.2018.8441097.

Sulaiman, M.H., Mustaffa, Z., Saari, M.M., Daniyal, H., Musirin, I., Daud, M.R., 2018b. Barnacles mating optimizer: An evolutionary algorithm for solving optimization. In: 2018 IEEE International Conference on Automatic Control and Intelligent Systems. I2CACIS, pp. 99–104. http://dx.doi.org/10.1109/I2CACIS.2018.8603703.

Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. IEEE Trans. Evol. Comput. 1, 67–82. http://dx.doi.org/10.1109/4235.585893.

Yang, X.-S., 2010. Firefly algorithm, Lévy Flights and global optimization. In: Bramer, M., Ellis, R., Petridis, M. (Eds.), Research and Development in Intelligent Systems XXVI. pp. 209–218.

Yao, X., Liu, Y., Lin, G., 1999. Evolutionary programming made faster. IEEE Trans. Evol. Comput. 3, 82–102. http://dx.doi.org/10.1109/4235.771163.

Yusa, Y., Yoshikawa, M., Kitaura, J., Kawane, M., Ozaki, Y., Yamato, S., Høeg, J.T., 2012. Adaptive evolution of sexual systems in pedunculate barnacles. Proc. R. Soc. B 279, 959–966. http://dx.doi.org/10.1098/rspb.2011.1554.

Zimmerman, R.D., Murillo-Sanchez, C.E., Thomas, R.J., 2011. MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. IEEE Trans. Power Syst. 26, 12–19. http://dx.doi.org/10.1109/TPWRS.2010.2051168.

Zou, F., Wang, L., Hei, X., Chen, D., 2015. Teaching–learning-based optimization with learning experience of other learners and its application. Appl. Soft Comput. 37, 725–736. http://dx.doi.org/10.1016/j.asoc.2015.08.047.