Heba F. Eid
Erik Cuevas

ENHANCED BONOBO OPTIMIZER FOR OPTIMIZING DYNAMIC PHOTOVOLTAIC MODELS

Abstract *Bonobo optimizer (BO) is a novel metaheuristic algorithm motivated by the social behaviour of the bonobos. This paper presents a quantum behaved bonobo optimization algorithm (QBOA) employing an innovative metaheuristic based on the reproductive strategies and social behavior of bonobos. Whereby, the quantum mechanics are embedded into the bonobo optimizer to direct the search agents through the search space. Accordingly, under this quantum-behaved movement, the proposed QBOA's exploitation capability is promoted. The performance of the proposed QBOA is exhibited on CEC2005 and CEC2019 benchmarks. Moreover, the QBOA algorithm was adapted to optimize the dynamic photovoltaic models parameters. QBOA exhibits the efficiency and adequacy to solve various optimization problems based on experimental and comparison findings, as well as its ability to implement competitive and promising results optimizing dynamic photovoltaic models.*

Keywords bonobo optimization algorithm, quantum behaved mechanism, photovoltaic model, dynamic photovoltaic model, parameters estimation

Citation Computer Science 25(3) 2024: 1–25

1. Introduction

Meta-heuristics algorithms have been effectively employed to solve a variety of real-world optimization problems, due to their reliable, robust and computationally efficient in avoiding local minima [11]. Various well-known meta-heuristics algorithms that have been proposed in the literature are Particle swarms [10], Coyote Optimization Algorithm (COA) [24], Dragonfly Algorithm (DA) [21], Particle Swarm Optimization (PSO) [10], Gravitational Search Algorithm (GSA) [27], Moth-Flame Optimization (MFO) [20], Genetic algorithm (GA) [17]. The bonobo optimizer (BO) is a novel meta-heuristic algorithm that is modeled on bonobo social behaviour [6]. The BO's search capabilities was made more resilient and efficient by using a fission-fusion approach for selection, four well-developed methods for producing offspring, and the application of two distinctive phases.

Despite the fact that population based approaches can grant promising solutions for optimization problems, as the dimension of the search space grows, they experience a series of challenges. A major key challenges is that, population based approaches frequently become stuck in local optimum when dealing with multi-modal complicated problems [3]. Correspondingly,to achieve high performance on complicated optimization problems, the exploration and exploitation stages should be well balanced [12].

The photovoltaic (PV) solar model presents one of the most exciting themes that has led to an increase in researchers' interest [13, 28]. However, one of the key challenges for researchers is to insure that the PV model captures the maximum amount of available power [4]. Different PV solar models have been presented including the static and dynamic PV models. Nevertheless, in the static PV model the representation of the load connection, switching and variation is not taken into account. Therefore, dynamic PV model has been proposed to overcome the drawbacks of the static PV models by representing the load connection in the PV model [8, 16]. The accuracy of dynamic PV models is essentially impacted by the precision of their parameters values which are obtained under various operating conditions. Accordingly, an accurate identification of the PV parameters is vital to gain the maximum power of the dynamic PV model.

Several conventional methods are applied for the PV parameters identification [14,29]. Meanwhile, for dynamic PV models, only the non-linear least square approaches and least square have been utilized for parameters identification [1,8]. Nevertheless, conventional methods based on classical numerical/analytical tools might be unable to accurately fit with the PV model, in consequence of the multi-modal and non-linear nature of the problem, which leads to negatively impact the maximum available power optimal capturing.

Motivated from the above discussion, a quantum behaved bonobo optimization algorithm (QBOA) is proposed. Whereby, quantum mechanics are adopted in this paper to integrate a quantum behavior in the bonobo optimization algorithm. For which, the proposed quantum-behaved bonobo optimization exploitation mechanism absorbs the character of Quantum-behaved method. Aiming to evaluate the robustness and coherence of the QBOA, the proposed QBOA performance is evaluated on CEC2019 and CEC2005 benchmark. Furthermore, the QBOA algorithm was adapted to optimize the dynamic photovoltaic models parameters.

The major contributions of this paper are as follows:

1. A quantum behaved bonobo optimization algorithm is proposed (QBOA), which is combining the advantages of the BOA and quantum mechanics to direct the search agents through the search space.
2. Several tests are conducted over unimodal and multimodal benchmark functions that are adopted for assessing the effectiveness of the proposed QBOA algorithm.
3. The proposed QBOA algorithm is used for optimizing integral and fractional dynamic photovoltaic models. The experimental results ensure that the QBOA algorithm is efficient enough in the identification of the dynamic PV model parameters.

The remainder of the paper is organized as follows: In Sec. 2, the Bonobo Optimization algorithm brief description is given. In Sec. 3, the proposed QBOA is detailed. In Sec. 5, the efficiency of the proposed QBOA algorithm on CEC2019 and CEC2005 benchmarks, as well as comparative analysis of the QBOA versus several optimization algorithms are presented. The dynamic photovoltaic models are provided in Sec. 4. The simulation results and analysis of the QBOA of dynamic PV models are discussed in Sec. 5.4. Lastly, in Sec 6, The paper's key findings are discussed.

2. Bonobo optimization algorithm

Das and Pratihar presented the Bonobo Optimizer (BO) as a new metaheuristic algorithm [6]. The BO algorithm simulates the reproductive approaches social behavior of bonobos. BO mimics the fission-fusion process, which focuses on segmenting the community into multiple subgroups of varying compositions and sizes, then rejoining them with the rest of the community. Restricting, promiscuous, consortship and extra-group mating are the four types of bonobo strategies. The BO algorithm's working principle is depicted in detail as follows:

Non-user initial parameters are set: the positive and negative phase count $ppc = 0, npc = 0$, the change in phase $cp = 0$, the extra-group mating probability $p_{xgm} = p_{xgm-intial}$, the sizing factor of the temporary sub-group $tsgs_{factor} = tsgs_{factor-initial}$, the directional probability $p_d = 0.5$ and the phase probability $p_p = 0.5$.

Inspired by the fission-fusion social group technique [7], the p^{th} bonobo is chosen for pairing with the i^{th} bonobo. For which, the temporary subgroup maximum size $tsgs_{max}$ is calculated using the following equation:

$$tsgs_{max} = max(2, tsgs_{factor} \times N) \tag{1}$$

Where N is the size of the population. According to equation 1,the temporary subgroup size is between 2 and $tsgs_{max}$; and is constructed by randomly selecting non-repeats bonobos from the $N-1$ population, eliminating the i^{th} bonobo. The optimal solution from the subgroup is picked as the p^{th} bonobo if its fitness is higher than the i^{th} bonobo; contrarily, a randomly bonobo from the subgroup is selected as the p^{th} bonobo. The chosen p^{th} bonobo then begins mating in order to generate the offspring.

In the bonobo society, four different types of mating have been observed: extra-group mating, promiscuous, consortship and restrictive mating. The mating method differs according to whether the phase is positive or negative. The possibility of restricted and promiscuous matings is high during a positive phase. On the other hand, in a negative phase, extra-group and consortship matings are perceived as high. A mating method is utilized using the phase probability parameter p_p. Whereby, a random number $r \in [1,0]$ is generated, and determined to be either equal or less than p_p. A new_bonobo is created using the following equation:

$$\begin{aligned} new\_bonobo_k = bonobo_k^i + r_1 \times scab \times (\alpha_k^{bonobo} - bonobo_k^i) + \\ (1-r_1) \times flag \times scsb \times (bonobo_k^i - bonobo_k^p) \end{aligned} \tag{2}$$

where r_1 is a random number in [0,1] and k is the optimization problem decision variable number. While, scsb and scab are the p^{th} bonobo and the alpha bonobo sharing coefficients, respectively. The parameters scab and scsb are predetermined constants that affect the balance between explorative and exploitative tendencies. The flag parameter can have two possible values: 1 or -1 for promiscuous mating or restrictive mating, respectively.

Sharing coefficients for the alpha-bonobo and pth-bonobo are represented using scab and scsb, respectively.

On the contrary, extra-group mating is employed to create an offspring when r is bigger than p_p, as shown below:

$$\beta_1 = e^{\left(r_4 + r_4^2 - 2/r_4\right)} \tag{3}$$

$$\beta_2 = e^{\left(2r_4 - r_4^2 - 2/r_4\right)} \tag{4}$$

$$new\_bonobo_k = bonobo_k^i + \beta_1 \times (UB_k - bonobo_k^i) \tag{5}$$

$$new\_bonobo_k = bonobo_k^i - \beta_2 \times (bonobo_k^i - LB_k) \tag{6}$$

$$new\_bonobo_k = bonobo_k^i - \beta_1 \times (bonobo_k^i - LB_k) \tag{7}$$

$$new\_bonobo_k = bonobo_k^i + \beta_2 \times (UB_k - bonobo_k^i) \tag{8}$$

where, LB_k and UB_k are the lower and upperboundary, respectively; while, $r4 \neq 0$ is a random number.

If a random number $r2$ is greater than $pxgm$, the consortship mating method is utilized to generate an offspring, as follows:

$$new\_bonobo_k = \begin{cases} bonobo_k^i + flag \times e^{-r_5} \times \left(bonobo_k^i - bonobo_k^p\right), \\ \qquad\qquad\qquad \text{if } (r_6 \leq p_d || \text{ flag } = 1\ ) \\ \\ bonobo_k^p, \qquad\qquad \text{otherwise} \end{cases} \tag{9}$$

When the new bonobo's fitness is discovered to be better than the parent's, or when a random number $r \in [0, 1]$ is equal to or less than p_{xgm}, the new bonobo is accepted. Furthermore, if the new_bonobo fitness is shown to be superior to the alpha fitness, the new_bonobo is designated as the alpha-bonobo.

The BO's controlling parameters are modified as follows when the newly obtained alpha bonobo in the current iteration is detected to be an improved solution.

$$npc = 0,\ cp = min(0.5, ppc \times rcpp),\ ppc = ppc + 1,$$

$$p_{xgm} = p_{xgm-initial},\ p_d = p_p,\ p_p = cp + 0.5,$$

$$tsgs_{factor} = min(tsgs_{factor-max}, (tsgs_{factor-initial} + ppc \times rcpp^2)).$$

Where $rcpp$ is the change rate of the phase probability. On the other hand, The following updates have been made to the controlling parameters:

$$npc = 1 + npc,\ ppc = 0,\ cp = -min(0.5, rcpp \times npc),$$

$$p_p = cp + 0.5, p_{xgm} = min(0.5, (p_{xgm-initial} - rcpp^2 \times npc)),$$

$$tsgs_{factor} = min(0, (tsgs_{factor-initial} - npc \times rcpp^2)), p_d = p_p,.$$

3. Proposed quantum-behaved bonobo optimization algorithm (QBOA)

In Bonobo optimizer, the bonobos are characterized by their location and position vector, which constitute the bonobo particle's trajectory. In accordance with Newtonian mechanism particles moves along a predetermined trajectory. However, due to the principle of uncertainty, it is not possible to estimate both distance and position simultaneously in reality.

Accordingly, quantum mechanics are adopted in this study to integrate quantum behaviour in the bonobo optimization algorithm. For which, the proposed quantum-behaved bonobo optimization exploitation mechanism absorbs the character of Quantum-behaved method. The mechanism setting places the bonobo in quantum mechanics space, utilizes the wave function to describe the bonobo's position and regulated the bonobo's state change process according to the Schrodinger equation [18].

In quantum mechanics, the fundamental time dependent Schrodinger equation is defined by:

$$i\hbar \frac{\partial \Psi}{\partial t} = -\hat{H}(X)\Psi \tag{10}$$

The wave function Ψ described the quantum state of the bonobo and only depends on its position. $\hat{H}(X)$ is a time independent Hamiltonian operator given by:

$$\hat{H}(X) = -\frac{\hbar^2}{2m}\nabla^2 + V(X) \tag{11}$$

Where $\hbar$ is the Planck's constant,$V(X)$ is the potential energy distribution and m is the bonobo mass. In a three dimensional space, the probability density of the bonobo in a position to appear is given by:

$$|\Psi|^2 dxdydz = Qdxdydz \tag{12}$$

Where, Q is the probability density function that meets the normalization condition:

$$\int_{-\infty}^{+\infty} |\Psi|^2 dxdydz = \int_{-\infty}^{+\infty} Qdxdydz = 1 \tag{13}$$

Moreover, the positions of local attractor for each bonobo can be defined as:

$$b_k = scab \times r \times \alpha_k^{bonobo} + scsb \times (1-r) \times bonobo_k^p \tag{14}$$

The statistical justification for the wave function is shown by Equations 12 and 13 where the integration is carried out over the full space. Each bonobo in the proposed algorithm has assumed a spin-less movement with a specific potential energy in D-dimensional Hilbert space. The bonobo is pulled using this field in accordance with a position specified by Equation 14.

For D-dimensional Hilbert space, where each bonobo position is bounded by delta potential well; a new_bonobo is created using the following equation:

$$new\_bonobo_k = b_k + flag \times \alpha_k^{bonobo} \cdot |bonobo_k^i - W_k| \times \ln\left(\frac{1}{r}\right) \tag{15}$$

Where r is a random number in [0,1] and W is the average positions of the bonobo at iteration t. The pseudo code of the QBOA algorithm is presented in Algorithm 1.

Algorithm 1 Pseudocode of the QBOA

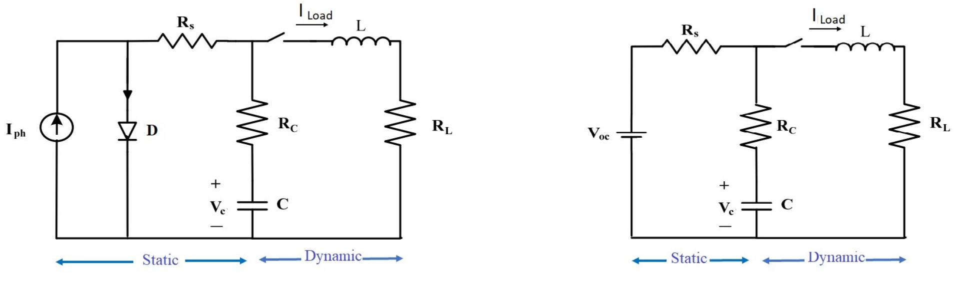
Input:
Total population number N_{pop}
Optimization iterations number Max_Iter
Output:
Optimal alpha Bonobo

```
Initialize the bonobo parameters
Initialize the bonobo population positions randomly.
Calculate the objective values for each search agent and dictate the α bonobo
while t ≤ Max_Iter do
    Calculate tsgs_max using equation 1
    for i=1:N_pop do
        Determine the temporary subgroup size
        #Apply the fission-fusion Technique
        Select Flag value
        if r <= p_p then
            #Update the position of the bonobo using the quantum mechanism
            Calculate the positions of local attractor for each bonobo using equation 14
            Create new_bonobo using equation 15
            Apply the boundary limiting conditions
        else
            for k=1:d do
                if r_2 < p_xgm then
                    if α_k^bonobo >= bonobo_k then
                        if r_3 <= p_p then
                            Create new_bonobo_k using equation 5
                        else
                            Create new_bonobo_k using equation 6
                        end if
                    else
                        if r_4 <= p_p then
                            Create new_bonobo_k using equation 7
                        else
                            Create new_bonobo_k using equation 8
                        end if
                    end if
                else
                    Create new_bonobo_k using equation 9
                end if
                Apply the boundary limiting conditions
            end for
            Evaluate the new_bonobo_k fitness value
            if fitness(new_bonobo_k) < fitness(α^bonobo) then
                α^bonobo = new_bonobo_k
            end if
        end if
    end for
    Update the controlling parameters
    t=t+1
end while
return α^bonobo
```

4. Dynamic photovoltaic models

4.1. Integral dynamic photovoltaic model

The considered integral dynamic PV model [8] is a second-order model that account the junction capacitance and conductance, along with the inductive effects, as shown in Figure 1a. The PV model and its associated load are valid in the area of the current-voltage curve which lies between the near constant voltage region and the open circuit voltage [9,15]. The circuit in Figure 1a has a linear behavior; accordingly, it is possible to reduce the PV static part to a series resistance Rs and a constant voltage source Voc, yielding the circuit in Figure 1b. While, the dynamic part of the integral PV model is represented by conductance Rc, capacitor C for junction capacitance, and inductance L for cabling and connection inductance.



(a) PV model

(b) Equivalent PV model in the linear voltage region

Figure 1. Integral-dynamic PV model

4.2. Fractional dynamic photovoltaic model

The inductor and capacitor in the fractional dynamic PV model are alternated with fractional equivalents of orders β and α, respectively, as illustrated in Figure 2.

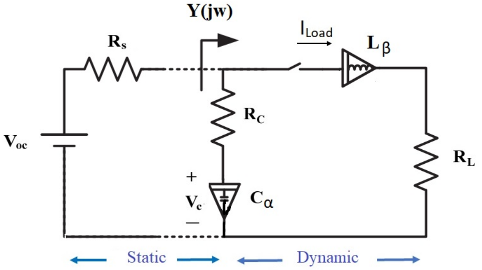


Figure 2. Fractional-dynamic PV model

Whereby, the fractional capacitor's effect is visible in low value of the resistor R_c; due to, the real frequency dependence on fractional capacitance impedance [26].

5. Simulation results and analysis

With the goal of examining the performance and capabilities of the proposed Quantum-Behaved Bonobo Optimization Algorithm QBOA, Matlab R2018b was adjusted for simulation purposes. All evaluation experiments were conducted on: Intel(R), Core i7 4910MQ CPU@2.90GHz and 16GB RAM.

5.1. Performance estimation with CEC 2005

With the aim to evaluate the proposed QBOA algorithm performance, multiple optimization test problems solved over various runs to obtain a reliable conclusion. QBOA is estimated on 23 test functions extracted from the CEC 2005 [19]. Accordingly, the test functions are separated into two categories based on their characteristics:($F1-F7$) unimodal and ($F8-F23$) multimodal functions . Whereby, the test objective functions are denoted as: "differentiable, non-differentiable, discontinuous, continuous, scalable, non-scalable,non-separable and separable". Since they include no local optima and a single global optimum, unimodal functions are used to estimate the exploitative potential of the meta-heuristic algorithm. Multimodal functions, on the other hand, have several local optimal and one global optimum. As a result, they can be used to assess the meta-heuristic algorithm's capability for exploration and escape from local optima.

The BOA and proposed QBOA internal parameters are adapted as: total population number $N = 50$, stopping criterion of $30000 * d$, where d is the optimization problems dimension, $p_{xgm-initial} = 1/d$, rcpp=0.0036, $tsgs_{factor-max} = 0.02$, and the sharing coefficients scab=1.3 and scsb=1.4. To conduct an unbiased comparison, the statistical results for each test function are calculated across 30 independent runs with completely random initial conditions. Through which, four distinct evaluation factors are taken into account: the minimum (best) solution, the average (mean) solution, the maximum (worst) solution and the standard deviation (St-dev). The worst, best and mean metrics examine the accuracy of the solution, while the St.dev estimates the obtained solution's robustness.

The experimental finding of the BOA and proposed QBOA; on the unimodal and multimodal test optimization functions are recorded in Table 1. From Table 1, it is clear that the proposed quantum-behaved BO surpass the BO algorithm in term of mean, best and worst results, except for test function F12. Both QBOA and BOA could continuously attain the global optimal for F5 and F6. Furthermore, compared to the BOA, the QBOA was able to locate the global optima with less standard deviation for all test functions, which demonstrate the QBOA's robustness in locating the global optimal.

Table 1
Statistical results of BOA and QBOA algorithm on CEC 2005

Function	BOA				QBOA			
	Best	Mean	Worst	St.dev	Best	Mean	Worst	St.dev
F1	9.5176E-57	4.51667E-46	1.35358E-44	2.4712E-45	1.3325E-141	2.0989E-118	5.3473E-117	9.7943E-118
F2	9.18109E-29	3.80198E-26	2.25682E-25	6.34863E-26	5.29453E-72	3.15854E-60	9.45349E-59	1.72583E-59
F3	3.01466E-56	5.08291E-50	6.43562E-49	1.33984E-49	9.6229E-136	1.8777E-109	5.633E-108	1.0284E-108
F4	4.27144E-28	3.31868E-25	2.49976E-24	6.22758E-25	9.13827E-72	6.93219E-57	2.05336E-55	3.74755E-56
F5	0	0	0	0	0	0	0	0
F6	0	0	0	0	0	0	0	0
F7	1.17E-04	1.49E-03	4.23E-03	1.09E-03	2.84E-06	7.58E-04	2.34E-03	5.87E-04
F8	-418.9828873	-411.0869983	-300.5445527	30.0487686	-4189.982887	-4182.982887	-4111.982869	3.31728E-06
F9	0	0.033165302	0.994959057	0.18165384	0	0	0	0
F10	8.88178E-16	8.88178E-16	8.88178E-16	0	8.88178E-16	8.88178E-16	8.88178E-16	0
F11	0	0.001644112	0.009864672	0.003739194	0	0	0	0
F12	4.71163E-31	4.71163E-31	4.71163E-31	8.90784E-47	7.81257E-26	3.69146E-08	1.10744E-06	2.0219E-07
F13	4.69E-27	3.0545E-06	6.62661E-05	1.28046E-05	1.34978E-32	1.34978E-32	1.34978E-32	5.5674E-48
F14	0.998003838	1.776170636	12.67050581	2.961408668	0.998003838	0.998003838	0.998003838	1.00999E-16
F15	3.08E-04	4.58E-03	2.04E-02	7.51E-03	3.09E-04	3.30E-03	2.04E-02	6.81E-03
F16	-1.031628453	-1.031628453	-1.031628453	5.21556E-16	-1.03	-1.03	-1.03	0
F17	0.397887358	0.397887358	0.397887358	0	0.397887358	0.39788736	0.397887	0
F18	3	3.9	30	4.929503018	3	3	3	1.30E-15
F19	-3.862779787	-3.837012599	-3.089764134	0.141132703	-3.86	-3.86	-3.86	7.05E-15
F20	-3.042457738	-3.017878829	-2.981002427	0.030617435	-3.042423011	-3.025796432	-2.975483131	0.026197581
F21	-10.15319876	-9.984784532	-5.100772055	0.922442691	-10.15319876	-10.1531982	-10.15318981	2.12097E-16
F22	-10.40282204	-10.05122207	-5.128822495	1.338056572	-10.40282204	-10.40232042	-10.39505238	0.001910066
F23	-10.5362903	-9.998653738	-5.128480623	1.640498374	-10.5362903	-10.53629027	-10.53628927	1.87452E-17

The proposed QBOA was analyzed versus six well known met-heuristic algorithms: Coyote Optimization Algorithm (COA) [24], Dragonfly Algorithm (DA) [21], Particle Swarm Optimization (PSO) [10], Gravitational Search Algorithm (GSA) [27], Moth-Flame Optimization (MFO) [20], Genetic algorithm (GA) [17], as shown in Table 2. The initial controlling parameters of the optimization algorithms are listed in Table 3.

Table 2

Comparison results attained for QBOA and different optimization algorithms

Function		QBOA	COA	PSO	DA	GSA	GA	MFO
F1	Mean	**2.0989E-118**	25.32456	1.36E-04	5.30E-01	2.53E-16	8.00E-04	1.65E-31
	St.dev	**9.7943E-118**	9.284834	2.02 e-7	1.318	9.67E-17	8.70E-04	4.91E-31
F2	Mean	**3.15854E-60**	0.7051718	0.042144	2.392	0.055655	3.00E-03	2.69E-19
	St.dev	**1.72583E-59**	0.1208514	0.045421	3.912	0.194074	1.80E-03	6.22E-19
F3	Mean	**1.8777E-109**	2252.63547	70.12562	215.45	896.5347	13.213	2.05E-11
	St.dev	**1.0284E-108**	825.3839	22.11924	935.17	318.9559	8.042	4.21E-11
F4	Mean	**6.93219E-57**	24.4519057	1.086481	1.153	7.35487	0.209	5.79E-06
	St.dev	**3.74755E-56**	3.6646211	0.317039	2.702	1.741452	5.80E-02	3.17E-05
F5	Mean	**0**	2592.44628	96.71832	6784.5	67.54309	66.9	133.11
	St.dev	**0**	1925.75963	60.11559	21974.5	62.22534	22.6	555.57
F6	Mean	**0**	27.835087	0.000102	2.2023	2.50E-16	7.50E-04	4.78E-32
	St.dev	**0**	12.9163617	8.28E-05	5.528	1.74E-16	7.20E-04	1.27E-31
F7	Mean	**7.58E-04**	6.72E-02	1.23E-01	6.90E-03	0.089441	8.10E-04	1.20E-03
	St.dev	5.87E-04	2.40E-02	4.50E-02	7.60E-03	0.04339	**5.50E-04**	7.20E-04
F8	Mean	-4182.982887	**-12299.311**	-4841.29	-3213.66	-2821.07	-3692.39	-3329.13
	St.dev	**3.31728E-06**	105.679038	1152.814	431.748	493.0375	182.42	288.317
F9	Mean	**0**	20.11984	46.70423	11.561	25.96841	3.80E-04	12.8372
	St.dev	**0**	2.22484365	11.62938	10.177	7.470068	3.20E-04	7.352
F10	Mean	**8.88E-16**	4.0391476	2.76E-01	3.14E-05	6.21E-02	**8.88E-16**	**8.88E-16**
	St.dev	**0**	0.95061505	5.09E-01	1.70E-04	2.36E-01	1.00E-31	1.00E-31
F11	Mean	**0**	1.1976333	9.22E-03	0.3846	27.70154	8.88E-16	1.78E-01
	St.dev	**0**	7.64E-02	7.72E-03	0.3826	5.040343	1.00E-31	8.43E-02
F12	Mean	**3.69146E-08**	2.0935602	6.92E-03	0.5296	1.799617	5.73E-05	3.11E-02
	St.dev	**2.0219E-07**	0.809141215	2.63E-02	0.6912	0.95114	1.40E-04	9.49E-02
F13	Mean	**1.34978E-32**	11.91776	6.68E-03	0.5292	8.899084	6.21E-05	1.10E-03
	St.dev	**5.5674E-48**	3.7510395	8.91E-03	0.7173	7.126241	1.10E-04	3.33E-03
F14	Mean	**0.998**	**0.998**	3.627168	1.1	3.4	**0.998**	1.03
	St.dev	1.00999E-16	**4.12E-17**	2.560828	0.303306	2.578637	8.83E-14	0.181483682
F15	Mean	3.30E-03	**3.07E-04**	5.77E-04	1.34E-03	1.80E-03	8.40E-04	8.37E-04
	St.dev	6.81E-03	**7.70E-09**	2.22E-04	5.11E-04	4.90E-04	2.90E-04	2.54E-04
F16	Mean	**-1.03**	**-1.03**	**-1.03**	**-1.03**	**-1.03**	**-1.03**	**-1.03**
	St.dev	**0**	6.58E-16	6.25E-16	2.55E-11	**0**	5.02E-10	**0**
F17	Mean	**0.39789**	**0.39789**	**0.39789**	**0.39789**	**0.39789**	**0.39789**	**0.39789**
	St.dev	**0**	**0**	**0**	7.60E-13	**0**	4.73E-07	1.13E-16
F18	Mean	**3**	**3**	**3**	**3**	**3**	**3**	**3**
	St.dev	**1.30E-15**	1.36E-15	1.33E-15	1.38E-06	4.17E-15	1.21E-08	1.95E-15
F19	Mean	**-3.86**	-3.86277	-3.8628	**-3.86**	-3.8628	**-3.86**	**-3.86**
	St.dev	7.05E-15	3.12E-15	**2.58E-15**	1.59E-03	2.29E-15	2.20E-03	2.71E-15

Table 2 cont.

F20	Mean	-3.025796432	-3.04245773	-3.26634	-3.25	-3.31778	**-3.32**	**-3.22**
	St.dev	2.62E-02	**2.21E-13**	6.05E-02	6.72E-02	2.31E-02	2.17E-02	4.51E-02
F21	Mean	-10.1531982	-10.153199	-9.31	-9.81	-9.95512	**-10.2**	-7.56
	St.dev	**2.12097E-16**	7.23E-15	1.925505	1.280913	3.737079	4.84E-04	3.323037
F22	Mean	**-10.40232042**	**-10.40232042**	-9.52	-10.4	-9.68447	-9.93	-9.35
	St.dev	**1.91E-03**	0.962917757	2.00228	0.192434	2.014088	1.822252	2.423664
F23	Mean	**-10.536**	**-10.536**	**-10.536**	**-10.536**	**-10.536**	-9.61	-10.3
	St.dev	**1.87452E-17**	1.22342651	1.635722	1.060781	2.60E-15	2.405191	1.39948

Table 3
Parameter settings of the Optimization algorithms

Optimization Algorithm	Parameter Setting
Quantum-Behaved Bonobo Optimization Algorithm (QBOA)	rcpp=0.0036, $tsgs_{factor-max} = 0.02$, scab=1.25, scsb=1.3
Particle Swarm Optimization (PSO)	c1=c2=2, $w_{max} = 0.9, w_{min} = 0.2$
Coyote Optimization Algorithm (COA)	packs number$N_p = 10$, coyotes number $N_c = 10$
Gravitational Search Algorithm (GSA)	$G_0 = 100, alpha = 20$
Genetic algorithm (GA)	Roulette wheel selection, crossover=0.7, mutation=0.3
Dragonfly Algorithm (DA)	$\beta = 0.5$
Moth-Flame Optimization (MFO)	b=1,a decreased linearly from -1 to -2

As reported in Table 2, the proposed QBOA algorithm surpassed the comparison algorithms for all optimization test functions except for F8, where COA presents better mean and QBOA presents the second best; and for F15 where COA presents better standard deviation and mean measure. Compared to the GA and MFO algorithm, QBOA finds the second best results for function F20. Moreover,For functions F5, F6, F9, and F11, the theoretical global optimum could be consistently located through QBOA. While, for test function F10 QBOA was able to attain the theoretical global optimal similar to GA and MFO algorithms, however with the best St-dev.

5.2. Performance estimation with CEC 2019

Extra examinations of the proposed quantum behaved BO on the CEC2019 benchmarks are undertaken in this sub-section. CEC2019 [25] signifies a test environment, including ten different functions with various characteristics. CEC01, CEC02, and CEC03 are the first three functions, with dimensions of: $[-8192, 8192]$, $[-16384, 16384]$ and $[-4, 4]$, respectively. While, the test functions CEC04 to CEC10 are shifted and rotated in the range $[-100, 100]$. The evaluation findings are reported in Table 4. From Table 4, the proposed QBOA algorithm provides the best results in term of best, worst, mean and standard deviation for all test functions.

Table 4
Statistical results of BOA and QBOA algorithm on CEC 2019

Function	BOA				QBOA			
	Best	Mean	Worst	St.dev	Best	Mean	Worst	St.dev
CEC01	5.05E+09	4.39E+10	8.01646E+11	1.45922E+11	9.44E+07	3.82E+10	1.67726E+11	35343962073
CEC02	17.54313222	29.24788963	147.0449426	27.2284062	17.34285715	17.34394919	17.37353053	0.000158971
CEC03	12.70240428	12.70240486	12.70241987	2.88462E-06	12.70240422	12.70240826	12.70242195	1.13E-07
CEC04	196.933194	1224.912295	4116.441557	835.165023	32.78727204	188.1486484	1342.768471	24.1737793
CEC05	1.462741151	1.963403865	2.282512869	0.18092835	1.041741434	1.105880945	1.935413404	0.021160053
CEC06	8.32434066	11.14097169	13.38913543	1.333579119	8.028343674	9.275948809	13.29664279	1.024430079
CEC07	230.2766639	841.5797016	1358.830208	290.1988289	105.5116356	410.001414	1263.376387	186.1911856
CEC08	5.80548777	6.799254532	7.18450629	0.990889863	3.71901424	5.099504448	7.585492774	0.401191137
CEC09	5.030062803	53.47166279	205.4756953	52.53316654	2.075989319	2.163408587	5.849287551	0.008198
CEC10	20.2806579	20.57014903	20.78783798	0.135932697	20.26194419	20.13350873	20.77914636	0.115771177

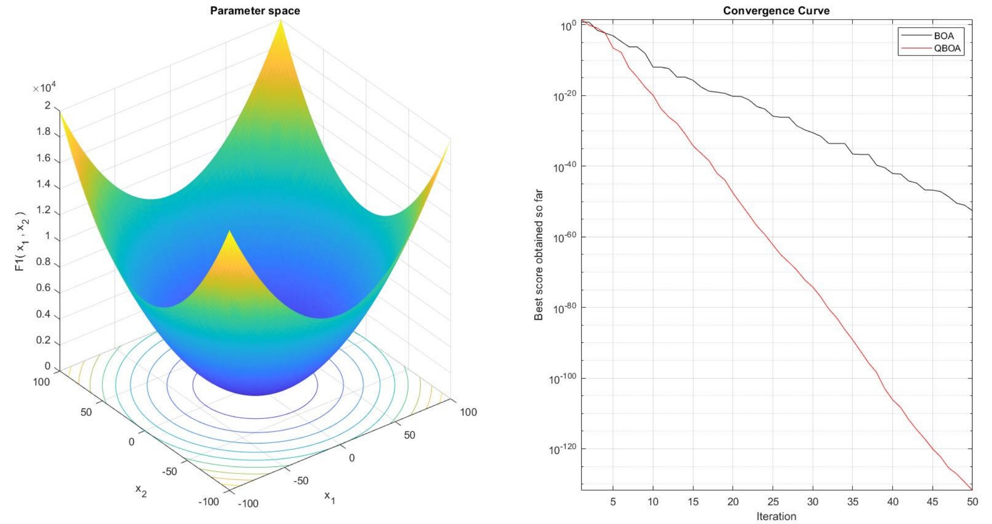
In addition, the QBOA is examined against four well known algorithms in the literature: Particle Swarm Optimization (PSO) [10], Dragonfly Algorithm (DA) [21], Whale optimization algorithm (WOA) [23] and Salp swarm algorithm (SSA) [22], Table 5. From Table 5, the proposed QBOA Obtains better results for CEC02, CEC05, CEC06, CEC08 and CEC09. Furthermore, QBOA provides the better mean for CEC10, and produces the second-best outcomes for test function CEC07 in comparison to the PSO algorithm.

Table 5
Comparison results attained for QBOA and different optimization algorithms on CEC2019

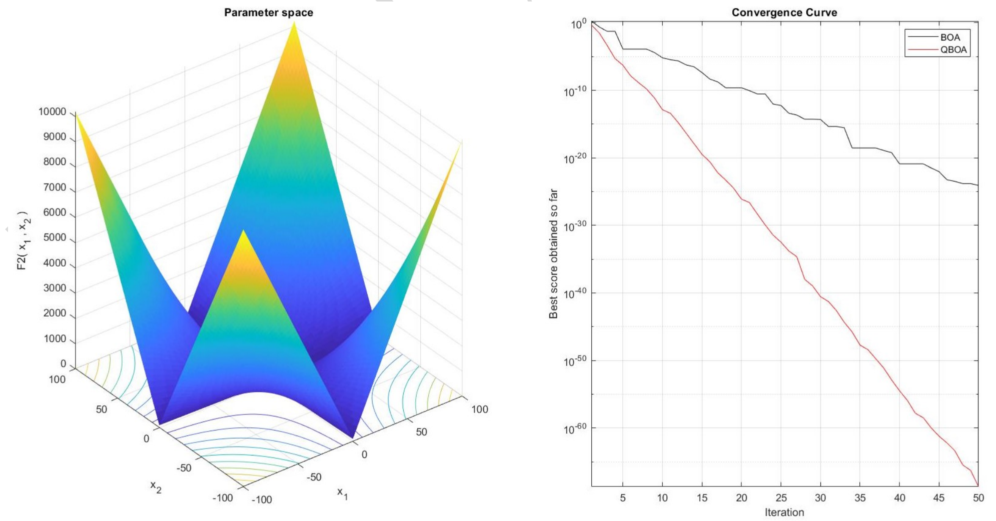
Function		QBOA	PSO	DA	WOA	SSA
CEC01	Mean	3.82E+10	1.471E + 12	5.43E+10	4.11E+10	**6.05E+09**
	St.dev	3.53E+10	1.324E + 12	6.69E+10	5.42E+10	**4.75E+09**
CEC02	Mean	**17.34394919**	15183.91348	78.0368	17.3495	18.3434
	St.dev	**0.000158971**	3729.553229	87.7888	0.0045	0.0005
CEC03	Mean	**12.7024**	**12.7024**	13.7026	13.7024	13.7025
	St.dev	1.13E-07	9.03E-15	0.0007	**0**	0.0003
CEC04	Mean	188.1486484	**16.80077558**	344.3561	394.6754	41.6936
	St.dev	24.1737793	**8.199**	414.0982	248.5627	22.2191
CEC05	Mean	**1.105880945**	1.138264	2.5572	2.7342	2.2084
	St.dev	**0.021160053**	0.08938	0.3245	0.2917	0.1064
CEC06	Mean	**9.275948809**	9.30531	9.8955	10.7085	6.0798
	St.dev	**1.024430079**	1.69	1.6404	1.0325	1.4873
CEC07	Mean	410.001414	**160.686**	578.9531	490.6843	410.3964
	St.dev	186.1911856	**104.203**	329.3983	194.8318	290.5562
CEC08	Mean	**5.099504448**	5.2241	6.8734	6.909	6.3723
	St.dev	**0.401191137**	0.7867	0.5015	0.4269	0.5862
CEC09	Mean	**2.163408587**	2.373279	6.0467	5.9371	3.6704
	St.dev	**0.008198**	0.01843	2.871	1.6566	0.2362
CEC10	Mean	**20.13350873**	20.2806	21.2604	21.2761	21.04
	St.dev	0.115771177	0.12853	0.1715	**0.1111**	0.078

5.3. Convergence analysis

The proposed QBOA and BOA's convergence behaviour are explored. The cost function for F1–F4, F7, F9, F10, F13, F15, and F21 test problems, as well as their associated convergence curves, are shown in Figures 3–6. As shown in Figure 3, the QBOA algorithm exhibits abrupt changes in the early stages of iterations, which decreased gradually throughout the course of iterations. This behaviour demonstrates that, the proposed QBOA algorithm gains from a well balance of exploitation and exploration, accordingly enables the QBOA to avoid being locked into local optimals.

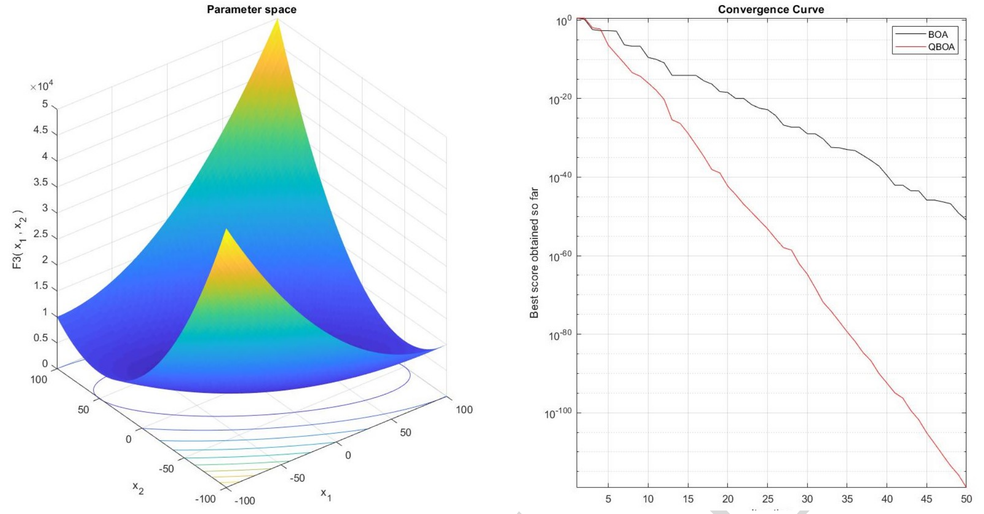


(a) $F1$

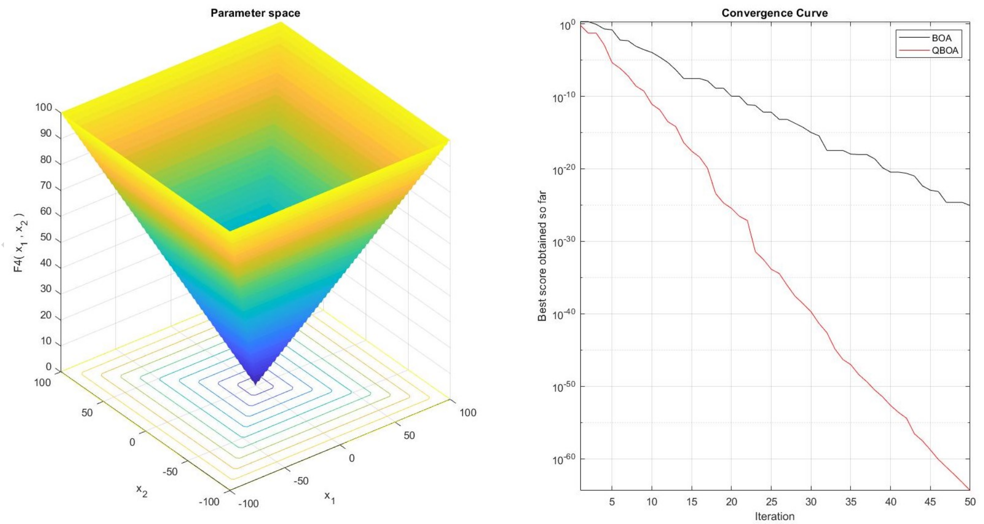


(b) $F2$

Figure 3. Best fitness convergence curves

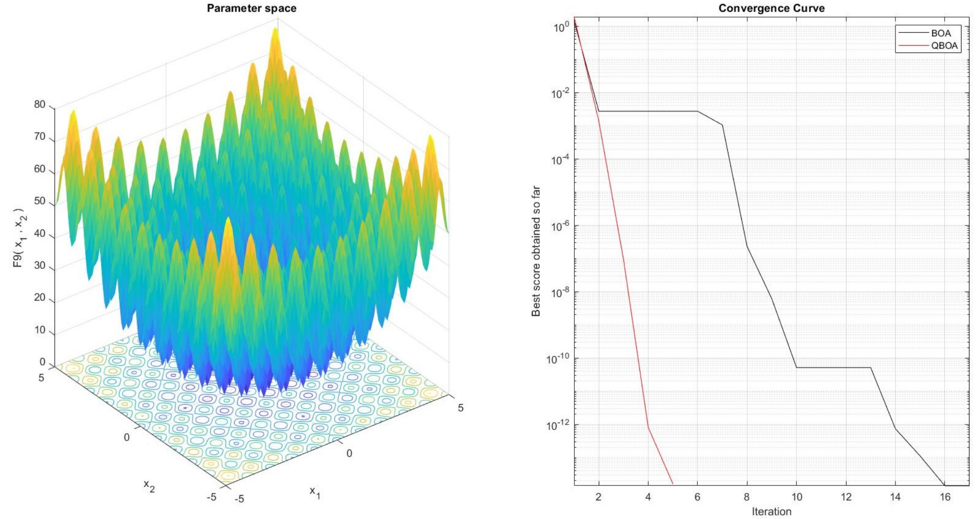


(a) $F3$

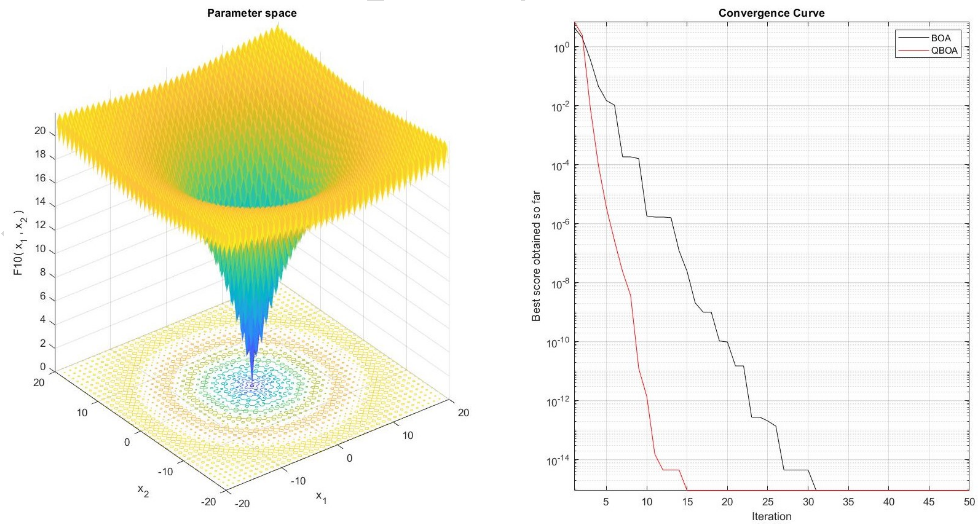


(b) $F4$

Figure 4. Best fitness convergence curves (cont.)

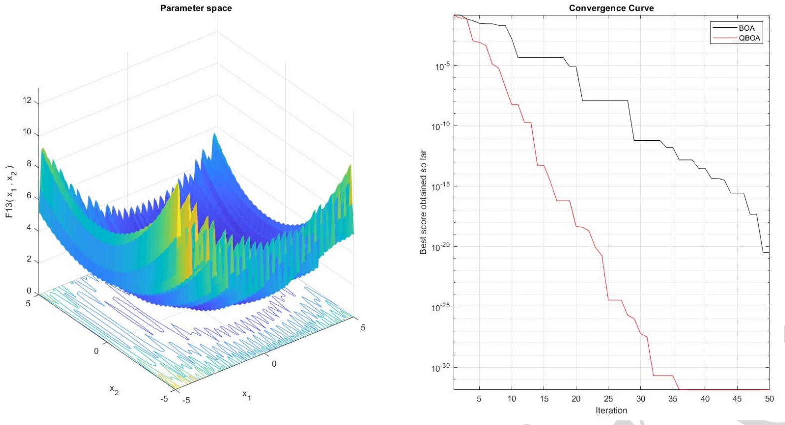


(a) $F9$

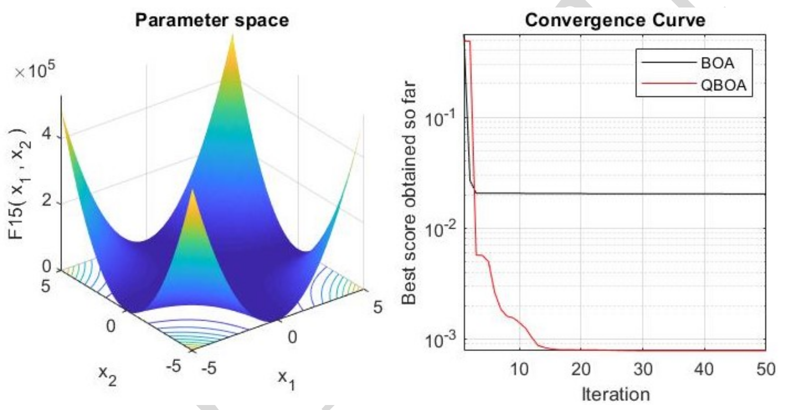


(b) $F10$

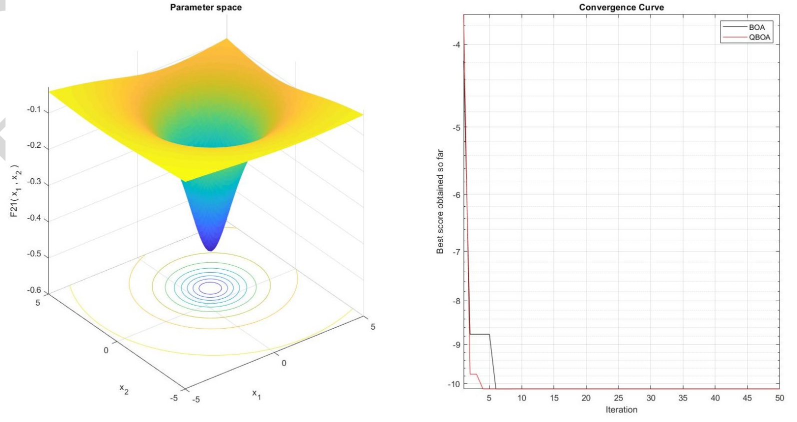
Figure 5. Best fitness convergence curves (cont.)



(a) $F13$



(b) $F15$



(c) $F21$

Figure 6. Best fitness convergence curves (cont.)

5.4. Dynamic PV models' parameters estimation

This subsection is concerned with the application of the proposed QBOA for parameter optimization of the integral and fractional dynamic PV models. The dynamic data sets were collected from PV module at an irradiance level of 655 W/m^2 and a temperature of 25^oC, with connected load $R_L = 23.1\Omega$ [8]. Whereby, for the integral dynamic PV model, three unknown parameters R_c, L and C should be estimated, while in the fractional dynamic PV model five parameters $R_c, L_\beta, C_\alpha, \beta$ and α should be estimated.

Thirty independent runs of the PV optimization problems were performed. The BOA and proposed QBOA internal parameter values for optimizing the dynamic PV models were kept the same as follows: $N = 50$, $p_{xgm-initial} = 1/d$, $rcpp = 0.0036$, $tsgs_{factor-max} = 0.02$, $scab = 1.3$ and $scsb = 1.4$. The upper and lower boundaries of the fractional and integral dynamic PV models are given in Table 6.

Table 6
Dynamic PV model parameters boundaries

Model	Parameters	Lower bound (LB)	Upper bound (UB)
Integral PV	R_c	0	20
	C	2.0E-08	600 E-7
	L	5.0 E-6	100 E-6
Fractional PV	R_c	0	20
	C_α	2.0E-08	600 E-7
	L_β	5.0 E-6	100 E-6
	α	0.8	1.1
	β	0.8	1.1

The three optimized parameters $(R_c, L, C,)$ for the integral PV model, the five optimized parameters $(R_c, C_\alpha, \alpha, L_\beta, \beta)$ for the fractional PV model and their corresponding objective function (RMSE) are reported in Table 7 and Table 8 . Moreover, the proposed QBOA's RMSE findings are compared to BOA algorithm and various well-known and recently developed optimizers, including Gradient-Based Optimizer (GBO) [2], artificial ecosystem based optimization (AEO) [30]and jellyfish search optimizer (JS) [5]. The tabulated results revealed that, the proposed QBOA produced the best RMSE values.

For more comprehensive validation of the proposed QBOA, the load current curve estimated by QBOA is generated and contrasted with that of the measured data for the integral and fractional PV model in Figure 7 and Figure 8, respectively. Additionally, the absolute error curves between the measured and estimated load current curves are given in Figure 9 and Figure 10. The visual comparisons validate the efficiency of the proposed QBOA in the identification of the dynamic PV model parameters.

Table 7
Identified parameters of integral dynamic PV model

Algorithm	R_c	C	L	RMSE
GBO	5.624748753	8.16E-06	7.47 E-6	0.008493067
AEO	5.624748647	8.16E-06	7.47 E-6	0.0084931
JS	5.624749	8.15726 E-6	7.47323 E-6	0.008493067
BOA	7.314974895	3.81307E-07	7.3251E-06	0.0084805
proposed QBOA	7.314974219	3.81307E-07	7.3251E-06	0.0084505

Table 8
Identified parameters of fractional dynamic PV model

Algorithm	R_c	C_α	L_β	α	β	RMSE
GBO	5.00598	5.04 E-6	1.35 E-5	1.026120535	0.957165925	0.0082360
AEO	4.55020	1.46 E-5	1.73 E-5	0.917230623	0.940654537	0.0081960
JS	4.69892	4.08 E-5	1.44 E-5	0.833404373	0.953192785	0.007995872
BOA	3.91281E-05	1.80215E-06	7.20724E-05	0.947239708	0.846072368	0.006168565
proposed QBOA	1.00E-05	1.83738E-06	6.60382E-05	0.94299238	0.852227908	0.006155832

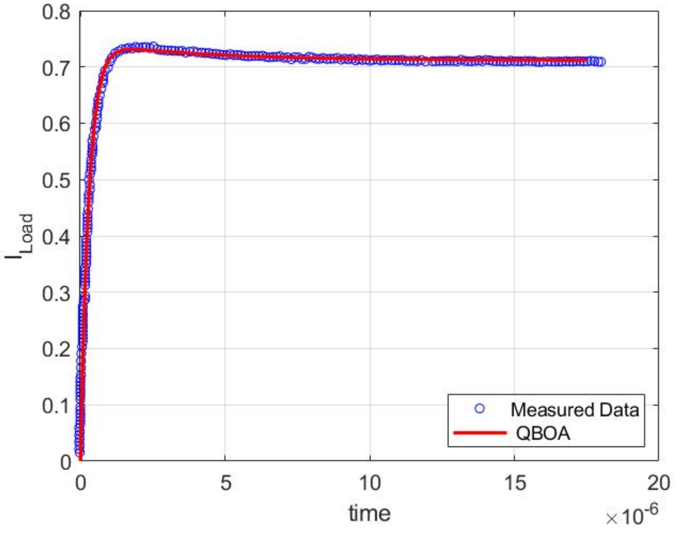


Figure 7. Integral dynamic PV model load current fitting

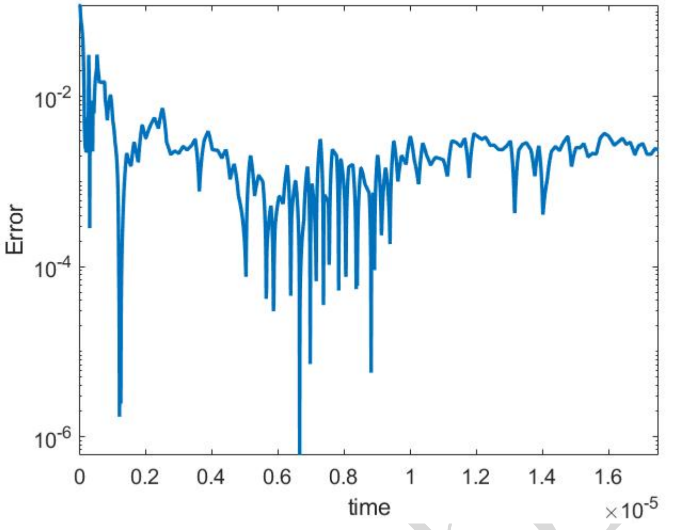


Figure 8. Integral dynamic PV model absolute error

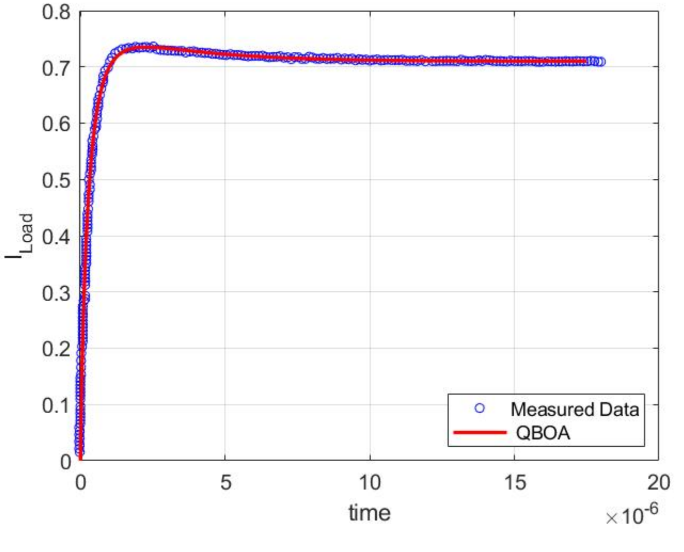


Figure 9. Fractional dynamic PV model load current fitting

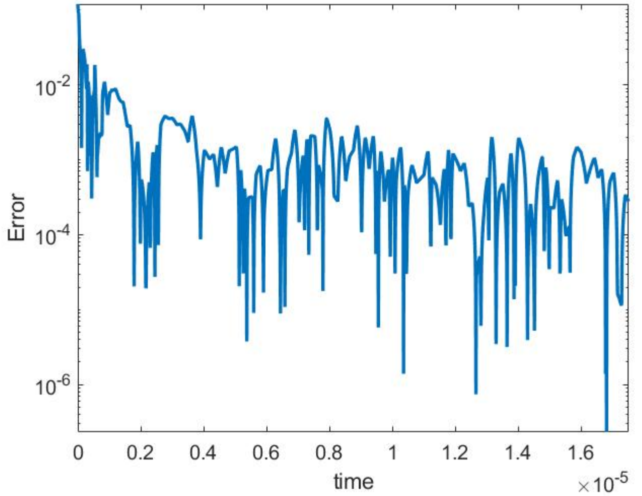


Figure 10. Fractional dynamic PV model absolute error

6. Conclusion

The finite supply of non-renewable resources combined with the world population's rapid increase is driving up demand for energy. Due to this circumstance, there is a risk of environmental pollution and climate change. Therefore, researchers' attention towards renewable energy sources, especially solar energy, have taken the spotlight. Several photovoltaic models have been proposed, where designing a high performance photovoltaic system requires addressing the problem of simulating a solar module and identifying its parameter. This paper employs a quantum behaved meta-heuristic named QBOA for addressing various optimization problems and dynamic photovoltaic models parameter identification. QBOA simulates the reproductive strategies and social behavior of bonobos. Quantum mechanics incorporated into the QBOA algorithm to direct the search agents through the search space. Correspondingly, the exploitation potential of the proposed QBOA algorithm is promoted by this integration. To determine the robustness and coherence of the QBOA, its performance has been examined using the CEC2019 and CEC2005 benchmarks. Additionally, the proposed QBOA is presented to optimize dynamic photovoltaic model's parameter. From the experimental and simulations results, it can be designated that the quantum behaved QBOA sustains the competitiveness on solving different optimization problems and optimizing dynamic photovoltaic models.

References

[1] AbdelAty A.M., Radwan A.G., Elwakil A.S., Psychalinos C.: Transient and Steady-State Response of a Fractional-Order Dynamic PV Model Under Different Loads, *Journal of Circuits, Systems and Computers*, vol. 27(02), 1850023, 2018. doi: 10.1142/S0218126618500238.

[2] Ahmadianfar I., Bozorg-Haddad O., Chu X.: Gradient-based optimizer: A new metaheuristic optimization algorithm, *Information Sciences*, vol. 540, pp. 131–159, 2020. doi: 10.1016/j.ins.2020.06.037.

[3] Ali M.Z., Awad N.H., Suganthan P.N., Duwairi R.M., Reynolds R.G.: A novel hybrid Cultural Algorithms framework with trajectory-based search for global numerical optimization, *Information Sciences*, vol. 334–335, pp. 219–249, 2016. doi: 10.1016/j.ins.2015.11.032.

[4] Ayang A., Wamkeue R., Ouhrouche M., Djongyang N., Salomä N.E., Pombe J.K., *et al.*: Maximum likelihood parameters estimation of single-diode model of photovoltaic generator, *Renew Energy*, vol. 130, pp. 111–121, 2019. doi: 10.1016/j.renene.2018.06.039.

[5] Chou J.S., Truong D.N.: A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean, *Applied Mathematics and Computation*, vol. 389, 125535, 2021. doi: 10.1016/j.amc.2020.125535.

[6] Das A.K., Pratihar D.K.: A new Bonobo Optimizer (BO) for real-parameter optimization. In: *2019 IEEE Region 10 Symposium (TENSYMP)*, pp. 108–113, 2019. doi: 10.1109/tensymp46218.2019.8971108.

[7] De Waal F.B.: Bonobo Sex and Society, *Scientific American*, vol. 272, pp. 82–88, 1995. doi: 10.1038/scientificamerican0395-82.

[8] Di Piazza M.C., Luna M., Vitale G.: Dynamic PV Model Parameter Identification by Least-Squares Regression, *IEEE Journal of Photovoltaics*, vol. 3, pp. 799–806, 2013. doi: 10.1109/jphotov.2012.2236146.

[9] Di Piazza M.C., Vitale G.: Photovoltaic field emulation including dynamic and partial shadow conditions, *Applied Energy*, vol. 87, pp. 814–823, 2010.

[10] Eberhart R., Kennedy J.: A new optimizer using particle swarm theory. In: *Proc. of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43, Nagoya, Japan, 1995.

[11] Eid H.F., Abraham A.: Solving unconstrained, constrained optimization and constrained engineering problems using reconfigured water cycle algorithm, *Evolutionary Intelligence*, 2022. doi: 10.1007/s12065-021-00688-6.

[12] Eid H.F., Muda A.K.: Adjustive Reciprocal Whale Optimization Algorithm for Wrapper Attribute Selection and Classification, *International Journal of Image, Graphics and Signal Processing*, vol. 11, pp. 18–26, 2019. doi: 10.5815/ijigsp.2019.03.03.

[13] El-Fergany A.: Efficient tool to characterize photovoltaic generating systems using mine blast algorithm, *Electric Power Components and Systems*, vol. 43(8–10), pp. 890–901, 2015. doi: 10.1080/15325008.2015.1014579.

[14] Et-torabi K., Nassar-eddine I., Obbadi A., Errami Y., Rmaily R., Sahnoun S., El fajri A., Agunaou M.: Parameters estimation of the single and double diode photovoltaic models using a Gauss–Seidel algorithm and analytical method: A comparative study, *Energy Convers Manage*, vol. 148, pp. 1041–1054, 2017. doi: 10.1016/j.enconman.2017.06.064.

[15] Gil-Arias O., Ortiz-Rivera E.I.: General purpose tool for simulating the behavior of PV solar cells modules and arrays. In: *2008 11th Workshop on Control and Modeling for Power Electronics*, pp. 1–5, 2008. doi: 10.1109/compel.2008.4634686.

[16] Go S.I., Choi J.H.: Design and Dynamic Modelling of PV-Battery Hybrid Systems for Custom Electromagnetic Transient Simulation, *Electronics*, vol. 9, 1651, 2020. doi: 10.3390/electronics9101651.

[17] Holland J.H.: Genetic algorithms, *Scholarpedia*, vol. 267, pp. 66–72, 1992. doi: 10.4249/scholarpedia.1482.

[18] Levin F.S.: *An introduction to quantum theory*, Cambridge University Press, 2002.

[19] Liang J.J., Suganthan P.N., Deb K.: Novel composition test functions for numerical global optimization. In: *Proceedings 2005 IEEE Swarm Intelligence Symposium, SIS 2005*, pp. 68–75, 2005. doi: 10.1109/SIS.2005.1501604.

[20] Mirjalili S.: Moth-flame optimization algorithm: a novel natureinspired heuristic paradigm, *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015. doi: 10.1016/j.knosys.2015.07.006.

[21] Mirjalili S.: Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete and multi-objective problems, *Neural Comput Appl*, vol. 27, pp. 1053–1073, 2016.

[22] Mirjalili S., Gandomi A.H., Mirjalili S.Z., Saremi S., Faris H., Mirjalili S.M.: Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017. doi: 10.1016/j.advengsoft.2017.07.002.

[23] Mirjalili S., Lewis A.: The whale optimization algorithm, *Adv Eng Softw*, vol. 95, pp. 51–67, 2016. doi: 10.1016/j.advengsoft.2016.01.008.

[24] Pierezan J., LS. C.: Coyote optimization algorithm: a new metaheuristic for global optimization problems. In: *IEEE World Conference on Computational Intelligence, Congress on Evolutionary Computation*, pp. 2633–2640, Brazil, Rio de Janeiro, 2018. doi: 10.1109/cec.2018.8477769.

[25] Price K.V., Awad N.H., Ali M.Z., Suganthan P.N.: *Problem Definitions and Evaluation Criteria for the 100-Digit Challenge Special Session and Competition on Single Objective Numerical Optimization*, Tech. rep., Nanyang Technological University, Singapore, 2018.

[26] Radwan A.G., Salama K.N.: Fractional-order, *RC* and and *RL* circuits, *Circuits, Systems, and Signal Processing*, vol. 31, pp. 1901–1915, 2012. doi: 10.1007/s00034-012-9432-z.

[27] Rashedi E., Nezamabadi-Pour H., Saryazdi S.: GSA: a gravitational search algorithm., *Information Sciences*, vol. 179(13), pp. 2232–2248, 2009. doi: 10.1016/j.ins.2009.03.004.

[28] Ridha H.M., Heidari A.A., Wang M., Chen H.: Boosted mutation-based Harris hawks optimizer for parameters identification of single-diode solar cell models, *Energy Conversion and Management*, vol. 209, p. 112660, 2020. doi: 10.1016/j.enconman.2020.112660.

[29] Tossa A.K., Soro Y., Azoumah Y., Yamegueu D.: A new approach to estimate the performance and energy productivity of photovoltaic modules in real operating conditions, *Solar Energy*, vol. 110, pp. 543–560, 2014. doi: 10.1016/j.solener.2014.09.043.

[30] Zhao W., Wang L., Zhang Z.: Artificial ecosystem-based optimization: a novel nature-inspired meta-heuristic algorithm, *Neural Computing & Applications*, vol. 32, pp. 9383–9425, 2020. doi: 10.1007/s00521-019-04452-x.

Affiliations

Heba F. Eid
Al-Azhar University, Faculty of Science, Cairo, Egypt, heba.fathy@azhar.edu.eg

Erik Cuevas
University of Guadalajara, Department of Electronics, Mexico, erik.cuevas@cucei.udg.mx

Received: 30.07.2023
Revised: 02.04.2024
Accepted: 02.04.2024