Computers, Materials & Continua
DOI:10.32604/cmc.2022.021517
Article

Tech Science Press

Performance of Gradient-Based Optimizer for Optimum Wind Cube Design

Alaa A. K. Ismaeel[1,2], Essam H. Houssein[3], Amir Y. Hassan[4] and Mokhtar Said[5,*]

[1]Faculty of Computer Studies (FCS), Arab Open University (AOU), Muscat, 130, Sultanate of Oman
[2]Faculty of Science, Minia University, Minia, Egypt
[3]Faculty of Computers and Information, Minia University, Minia, Egypt
[4]Electronics Research Institute, Giza, Egypt
[5]Electrical Engineering Department, Faculty of Engineering, Fayoum University, Fayoum, Egypt
[*]Corresponding Author: Mokhtar Said. Email: msi01@fayoum.edu.eg
Received: 05 July 2021; Accepted: 15 August 2021

Abstract: Renewable energy is a safe and limitless energy source that can be utilized for heating, cooling, and other purposes. Wind energy is one of the most important renewable energy sources. Power fluctuation of wind turbines occurs due to variation of wind velocity. A wind cube is used to decrease power fluctuation and increase the wind turbine's power. The optimum design for a wind cube is the main contribution of this work. The decisive design parameters used to optimize the wind cube are its inner and outer radius, the roughness factor, and the height of the wind turbine hub. A Gradient-Based Optimizer (GBO) is used as a new metaheuristic algorithm in this problem. The objective function of this research includes two parts: the first part is to minimize the probability of generated energy loss, and the second is to minimize the cost of the wind turbine and wind cube. The Gradient-Based Optimizer (GBO) is applied to optimize the variables of two wind turbine types and the design of the wind cube. The metrological data of the Red Sea governorate of Egypt is used as a case study for this analysis. Based on the results, the optimum design of a wind cube is achieved, and an improvement in energy produced from the wind turbine with a wind cube will be compared with energy generated without a wind cube. The energy generated from a wind turbine with the optimized cube is more than 20 times that of a wind turbine without a wind cube for all cases studied.

Keywords: Wind turbine; wind cube; gradient-based optimizer; metaheuristics; energy source

1 Introduction

Quality of life improvements are necessary as an economy and society develop. One corresponding challenge is to decrease environmental pollution. Replacing fossil fuels with clean energy is one of the main components to decrease environmental pollution. Renewable energy utilized at a large scale can help to meet daily energy demands [1–5]. Industries and academic institutions alike are interested in developing electricity from renewable and clean energy sources, and with the advancement of existing technology, these factors justify the importance of wind energy in recent years [6,7]. A wind cube is a modern wind turbine built to absorb and amplify more kilowatt-hours (kWh) of wind. When wind hits the wind cube, it concentrates and generates the speed, in turn producing more power. The modern wind cube system has been designed to solve low wind speed problems and collect wind power under these circumstances [8]. Wind cubes are used to improve the efficiency of wind turbines and result in great productivity [8]. Because of the wind's nonlinear nature, optimization techniques are essential. Optimizing the layout is achieved through soft computing technology [7]. The meta-heuristic optimization algorithms are used to extract the optimum solution for several problems. One of these problems is the estimation of parameters in photovoltaic models such as the Harris Hawks optimization [9], the Marine Predators Algorithm [10], the multi-strategy success-history-based adaptive differential evolution [11], the bacterial foraging algorithm [12], the differential evolution algorithms [13], Enhanced leader particle swarm optimization (ELPSO) [14], Time varying acceleration coefficients particle swarm optimization (TVACPSO) [15], and the shuffled frog leaping algorithm [16]. Meta-heuristic optimization applied to a wind farm layout is one of the main tools used to determine optimum wind farm position and maximize the generated power. The optimization algorithms used for these are modified genetic algorithms based on a Boolean code [17], the Monte Carlo method [18], a genetic algorithm-based local search [19], a new pseudo-random number generation method [20], and a multi-level extended pattern search algorithm [21].

The following items summarize the contributions of this paper:

- Increasing the power generated from the wind turbine over a year using the wind cube.
- The inner and outer radius of the wind cube, roughness factor and the height of the wind turbine hub are the decision variables extracted using a new optimization algorithm (Gradient-Based Optimizer).
- Comparison between the proposed GBO algorithm with Tunicate swarm algorithm (TSA) and Chimp optimization algorithm (ChOA) is performed for the same wind turbine.
- Minimizing the probability generated energy loss.
- Minimizing the wind turbine and wind cube cost using the meter cubic function.
- Comparison between the power generated from the wind turbine with and without a wind cube.

The paper organization is as follows, Section two explains the problem formulation and metrological data. The objective function is illustrated also in Section 2. Section 3 dissects the Gradient-Based Optimizer algorithm. The study cases are illustrated in Section 4. The conclusion and future work is in Section 5.

2 Problem Formulation and Metrological Data

2.1 Wind Turbine Analysis

The variation of wind speed is the main factor affecting the power generated from the wind turbine. The characteristics of wind turbine output power are dependent on the boundaries of the wind speed (cut-in speed V_{ci}, rated speed V_r and cut-off speed V_{co}) as in the following equation [22,23]:

$$P_{wind} = \begin{cases} P_r\left(\frac{V^3 - V_{ci}^3}{V_r^3 - V_{ci}^3}\right) & V_{ci} \le V \le V_r \\ P_r & V_r \le V \le V_{co} \\ 0 & V_{co} \le V \, or \, V \le V_{ci} \end{cases} \tag{1}$$

where P_r and V_r are the rated power and rated speed, respectively. The hub height of the wind turbine is affected by the stream speed to the wind turbine, so that the stream velocity is changed according to the hub height with the following equation [24]:

$$\frac{V_{h.2}}{V_{h.1}} = \left(\frac{h_2}{h_1}\right)^{\propto} \tag{2}$$

where $V_{h.2}$ is the velocity at the new hub height h_2 and $V_{h.1}$ is the reference velocity at the reference hub of height for the wind turbine h_1. For the neutral stability condition, α is the roughness ingredient factor which ranges from 0.14 to 0.25 [25]. The improvement of the power generated from the wind turbine is processed using the wind cube. The principle theory for wind cube design is the Bernoulli theory. The wind cube size is changed to achieve the optimal design. The configuration of the wind cube is explained in Fig. 1.

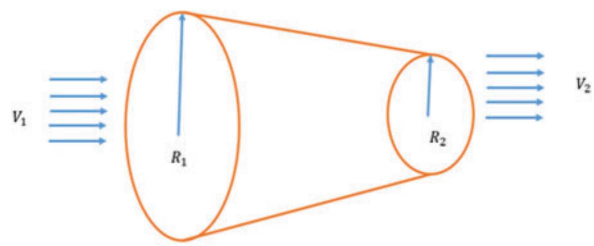

Figure 1: Wind cube configuration

Based on Bernoulli's theory and continuity equation, the main equation for the wind cube is shown as follows:

$$A_1 V_1 = A_2 V_2 \tag{3}$$

where V_1 is the wind speed input to the wind cube from the air side, V_2 is the wind speed output from the wind cube, A_1 is the area of the wind cube from the air side and A_2 is the area of the wind cube from the wind turbine side. The power generated from the wind turbine with the wind cube is given in this equation:

$$P_{wind} = \begin{cases} P_r\left(\frac{\left(\frac{R_1^2 V_1}{R_2^2}\right)^3 - V_{ci}^3}{V_r^3 - V_{ci}^3}\right) & V_{ci} \le V_1 \le V_r \\ P_r & V_r \le V_1 \le V_{co} \\ 0 & V_{co} \le V_1 \, or \, V_1 \le V_{ci} \end{cases} \tag{4}$$

where R_1 is the input radius of the wind cube from the air side and R_2 is the output radius of the wind cube from the wind turbine side.

2.2 Metrological Data

Hurghada City in the Red Sea governate of Egypt is the site used in this work to simulate the power output improvement. The latitude is $27°15'26.57''N$ and the longitude is $33°48'46.48''E$. The metrological data of the average wind speed for each month is explained in Fig. 2[1]., and the average wind speed over the year is shown in Fig. 3[1].

2.3 Analysis of Objective Function

The improvement to the power generated from the wind turbine corresponds with decreasing the loss of energy generated probability (LEGP) with the maximum speed out from the outer radius of the wind cube not increasing 80% of the cut-off speed for the turbine. The decision variables required for optimal sizing are the two radiuses of the wind cube, the wind turbine hub's height, and the roughness factor. The mathematical equation for the LEGP is as follows:

$$LEGP = \frac{E_{g.rated} - E_{g.actual}}{E_{g.rated}} \tag{5}$$

where $E_{g.rated}$, is the energy generated at rated power from the wind turbine, $E_{g.rated}$. The proposed algorithm is applied to an independent run for the objective function before another objective function (meter cubic function) is applied to choose the optimal solution that achieves the minimum parameters. The minimum parameters indicate that the cost of the wind turbine and wind cube is at its minimum. The second objective function is as follows:

$$f_{obj2} = R_1 \times R_1 \times h_2 \tag{6}$$

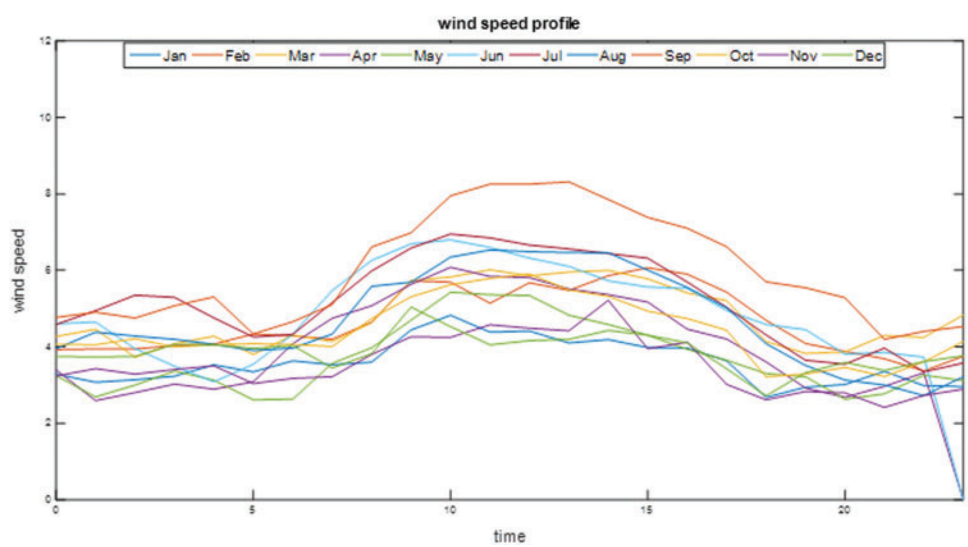

Figure 2: The average wind speed for each month

[1] http://www.wunderground.com

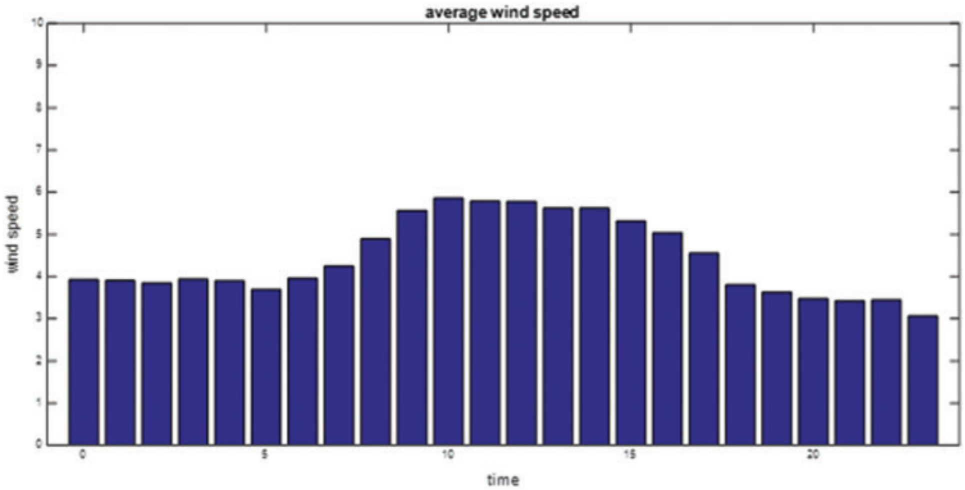

Figure 3: The wind speed average over the year

3 Gradient-Based Optimizer (GBO)

Recently, Ahmadianfar et al. [26–28] have proposed a new metaheuristic algorithm called the Gradient-Based Optimizer (GBO), which mimics the gradient and population-based methods together. In the GBO, in order to explore the search domain utilizing a set of vectors and two main operators (like local escaping operators and the gradient search rule), Newton's method is utilized to specify the search direction. The main process of the GBO are as follows:

3.1 The Initialization Process

In the GBO, the control parameters α and the probability rate are used to balance and switch from exploration to exploitation. Furthermore, the population size and iteration numbers are due to the problem's complexity. In the GBO, N vectors in a D-dimensional search space can be defined as:

$$X_{n,d} = \left[X_{n,d}, X_{n,d}, \ldots X_{n,d}\right], \quad n = 1, 2, \ldots N; \quad d = 1, 2, \ldots \tag{7}$$

Usually, the initial vectors of the GBO are randomly generated in the D-dimensional search domain, which can be defined as:

$$X_n = X_{\min} + rand\,(0, 1) \times (X_{\max} - X_{\min}) \tag{8}$$

where $X_{\min}$, and $X_{\max}$ are the bounds of decision variables X, and $rand(0, 1)$ is a random number in [0, 1].

3.2 Gradient Search Rule (GSR) Process

In the GBO algorithm, to guarantee a balance between exploration of significant search space regions and exploitation to reach near optimum and global points, a significant factor ρ_1 is employed as follows:

$$\rho_1 = 2 \times rand \times \alpha - \alpha \tag{9}$$

$$\alpha = \left|\beta \sin\left(\frac{3\pi}{2} + \sin\left(\beta \times \frac{3\pi}{2}\right)\right)\right| \tag{10}$$

$$\beta = \beta_{\min} + (\beta_{\max} - \beta_{\min}) \times \left(1 - \left(\frac{m}{M}\right)^3\right)^2 \quad (11)$$

where β_{min} and $\beta_{max}$are constant values 0.2 and 1.2, respectively, m represents the current iteration number, while M represents the total number of iterations. Particularly, the parameter ρ_1 is responsible for balancing the exploration and exploitation based on sine function α. This parameter value changes through iterations, starting with a large value through first the optimization iterations to improve population diversity. Then the value decreases through iterations to accelerate population convergence. The parameter value increases through defined iterations within a range [550, 750] to increase solution diversity and convergence around the best obtained solution. This also allows for exploration of more solutions, therefore enabling the algorithm to avoid local sub-regions. Thus, GSR can be determined as follows:

$$GSR = randn \times \rho_1 \times \frac{2\Delta x \times x_n}{x_{worst} - x_{best} + \varepsilon} \quad (12)$$

The concept of GSR is to provide the GBO algorithm with a random behavior through iterations, therefore strengthening exploration behavior and escape from local optima. In Eq. (12), it is defined by the factor Δx that delivers the difference between the best solution x_{best} and a randomly selected solution x_{r1}^m. The parameter δ is changed through iterations due to Eq. (15). In addition, a random number *randn* is included to improve exploration as follows:

$$\Delta x = rand(1:N) \times |\text{step}| \quad (13)$$

$$\text{step} = \frac{\left(x_{best} - x_{r1}^m\right) + \delta}{2} \quad (14)$$

$$\delta = 2 \times rand \times \left(\frac{\left|x_{r1}^m + x_{r2}^m + x_{r3}^m + x_{r4}^m\right|}{4} - x_n^m\right) \quad (15)$$

where $rand(1:N)$ is a vector of N random values $\in [0, 1]$. Also, four random integers are chosen from [1, N] which are r_1–r_4 such that $r_1 \neq r_2 \neq r_3 \neq r_4 \neq n$, and the parameter *step* represents a step size which is determined by x_{best} and x_{r1}^m.

Moreover, Direction Movement (DM) is employed to converge around the solution area x_n. To provide a convenient local search tendency with a significant effect on the GBO convergence, the term DM uses the best vector and moves the current vector x_n in the direction of $x_{best} - x_n$ and is computed as follows:

$$\text{DM} = rand \times \rho_2 \times (x_{best} - x_n) \quad (16)$$

where, rand is a uniform distributed number within the range [0, 1], and ρ_2 is a random parameter employed to modify step size of each vector agent. The ρ_2 parameter is considered a significant parameter of the GBO exploration process. The ρ_2 parameter is computed as follows:

$$\rho_2 = 2 \times rand \times \alpha - \alpha \quad (17)$$

Finally, depending on these terms GSR and DM, Eqs. (18) and (19) are used to update the position of the current vector x_n^m.

$$X1_n^m = x_n^m - GSR + DM \quad (18)$$

where, $X1_n^m$ is the new vector generated by updating x_n^m. According to Eqs. (12) and (16), $X1_n^m$ can be reformulated as:

$$X1_n^m = x_n^m - randn \times \rho_1 \times \frac{2\Delta x \times x_n}{x_{worst} - x_{best} + \varepsilon} + rand \times \rho_2 \times (x_{best} - x_n) \quad (19)$$

where yp_n^m and yq_n^m are equal to $y_n + \Delta x$ and $y_n - \Delta x$, respectively, and y_n is a vector equal to the average of the two vectors: current solution x_n and the vector z_{n+1} that are calculated as follows:

$$z_{n+1} = x_n - randn \times \frac{2\Delta x \times x_n}{x_{worst} - x_{best} + \varepsilon} \quad (20)$$

while x_n represents the current solution vector, *randn* is a random solution vector of dimension *n*, x_{worst} and x_{best} represent the worst and best solutions, and Δx is given by Eq. (13). Based on the previous formula, when replacing the best solution vector x_{best} with the current solution vector x_n^m, we get $X2_n^m$ as follows:

$$X2_n^m = x_{best} - randn \times \rho_1 \times \frac{2\Delta x \times x_n^m}{x_{worst} - x_{best} + \varepsilon} + rand \times \rho_2 \times \left(x_{r1}^m - x_{r2}^m\right) \quad (21)$$

Specifically, the GBO algorithm aims to enhance the exploration and exploitation phases using Eq. (19) to improve the global search for the exploration phase, while Eq. (21) is used to improve the local search capability for the exploitation phase. Finally, the new solution for the next iteration is as follows:

$$x_n^{m+1} = r_a \times \left(r_b \times X1_n^m + (1 - r_b) \times X2_n^m\right) + (1 - r_\alpha) \times X3_n^m \quad (22)$$

where r_a, and r_b are random numbers determined in the range [0, 1], and $X3_n^m$ is defined as:

$$X3_n^m = X_n^{m+1} - \rho 1 \times (X2_n^m - X1_n^m) \quad (23)$$

3.3 The Local Escaping Operator (LEO) Process

The LEO is introduced to strengthen the performance of the optimization algorithm to solve complex problems. The LEO can effectively update the position of the solution, to assist an algorithm to exit local optima points and speed the convergence of the optimization algorithm. The LEO targets generate a new solution with a superior performance X_{LEO}^m by several solutions (X_{best} best solution, the solutions $X1_n^m$ and $X2_n^m$ are randomly selected from population, X_{r1}^m, $X1_{r2}^m$ are randomly generated solutions) to update the current solution effectively. This process is performed based on following scheme:

If rand < pr

$$X_{LEO}^m = \begin{cases} X_n^{m+1} + f_1\left(u_1 x_{best} - u_2 x_k^m\right) + f_2 \rho_1 (u_3 \left(X2_n^m - X1_n^m\right) + u_2\left(x_{r1}^m - x_{r2}^m\right))/2, & \text{if rand} < 0.5 \\ x_{best} + f_1\left(u_1 x_{best} - u_2 x_k^m\right) + f_2 \rho_1 (u_3 \left(X2_n^m - X1_n^m\right) + u_2\left(x_{r1}^m - x_{r2}^m\right))/2, & \text{otherwise} \end{cases} \quad (24)$$

End

where pr is a probability value, $pr = 0.5$, the values f_1 and f_2 are uniform distribution random numbers $\in[-1, 1]$, and u_1, u_2 and u_3 are random values generated as follows:

$$u_1 = \begin{cases} 2 \times rand, & \text{if } \mu_1 < 0.5 \\ 1, & \text{otherwise} \end{cases} \tag{25}$$

$$u_2 = \begin{cases} 2 \times rand, & \text{if } \mu_1 < 0.5 \\ 1, & \text{otherwise} \end{cases} \tag{26}$$

$$u_3 = \begin{cases} rand, & \text{if } \mu_1 < 0.5 \\ 1, & \text{otherwise} \end{cases} \tag{27}$$

where $rand$ represents a random number $\in [0, 1]$ and μ_1 is a number in range $[0, 1]$. The previous equations for u_1, u_2 and u_3 can be simply explained as follows:

$$u_1 = L_1 \times 2 \times rand + (1 - L_1) \tag{28}$$

$$u_2 = L_1 \times rand + (1 - L_1) \tag{29}$$

$$u_3 = L_1 \times rand + (1 - L_1) \tag{30}$$

where L_1 is a binary parameter take value 0 or 1, such as if parameter $\mu_1 < 0.5$, then value of $L_1 = 1$, otherwise $L_1 = 0$. Where the solution x_k^m is generated as follows:

$$x_k^m = \begin{cases} x_{rand}, & \text{if } \mu_2 < 0.5 \\ x_p^m, & \text{otherwise} \end{cases} \tag{31}$$

where x_{rand} is a random generated solution according to the following formula:

$$x_{rand} = X_{\min} \times rand(0, 1) \times (X_{\max} - X_{\min}) \tag{32}$$

and x_p^m is a randomly selected solution from the population, μ_2 is a random number $\in [0, 1]$. For more details about GBO see [24].

4 Analysis of Results and Discussion

For fair benchmark comparison, the simulation settings are the same for all algorithms. furthermore, the algorithm parameter is set to their default values. This section presents the analysis of the proposed algorithm's results for the wind turbine explained in Tab. 1. The boundary limits for the decision variable of each turbine are explained in Tab. 2. The selection of boundaries for each turbine is dependent on the rotor blades radius and the height of the turbine hub. The wind cube radius from the turbine side is not almost less than the rotor blades radius. The wind cube radius from the airside is more than the rotor blade radius and smaller than the hub height with a specific distance according to each turbine to ensure that the cube does not touch the ground.[2,3]

[2] https://en.wind-turbine-models.com/turbines/493-aeolian-aeo-20

[3] https://en.wind-turbine-models.com/turbines/1380-aerolite-a-11

Table 1: Types of wind turbines

Type	Rated power	Rotor radius	Cut-in speed	Cut-off speed	Rated speed
6 kW	6000	3.05	3.6	25	9.8
30 kW	30000	5.5	3.6	25	13.4

Table 2: Boundary limits for the decision variable

Type	Lower boundaries		Upper boundaries	
	6 kW	30 kW	6 kW	30 kW
R_1	4.96	6.5	6	15
R_2	3.1	5.55	3.5	5.95
h_2	30	30	60	60
α	0.14	0.14	0.25	0.25

4.1 Wind Turbine of 6 kW

Based on analysis described in Section 2 and data reported in Tabs. 1 and 2, the proposed GBO algorithm is applied to the 6-kW wind turbine. Tab. 3 explains the proposed GBO algorithm's results for a 6-kW wind turbine based on minimizing the loss of the probability of a generated energy loss. The optimum solution for these results is determined according to the second objective function that satisfies the minimum meter cubic function.

Tab. 3 shows that all runs achieve zero probability of generated energy loss; the result in run 30 is the optimum solution due to this solution achieving the minimum objective function of minimizing the meter cubic function. Tab. 4 show the best solution from the proposed GBO algorithm in compared with Tunicate swarm algorithm (TSA) [29] and Chimp optimization algorithm (ChOA) [30] for 6-kW wind turbine. Based on this results the proposed GBO achieve the best meter cubic function compared with the other competitor algorithms. Fig. 4 shows the power demand output from the wind turbine without a wind cube at the same height as the wind turbine hub and the optimum solution's roughness coefficient. Fig. 5 shows the power demand output from the wind turbine with wind cube at the optimum solution for run 30 of the proposed GBO algorithm.

4.2 Wind Turbine of 30 kW

Based on analysis described in Section 2 and data reported in Tabs. 1 and 2, the proposed GBO algorithm is applied to the 30-kW wind turbine. Tab. 5 explains the proposed GBO algorithm's results for a 30-kW wind turbine based on minimizing the loss of the probability of a generated energy loss. The optimum solution for these results is determined according to the second objective function that satisfies the minimum meter cubic function.

Table 3: Results of the decision variable for 6 kW from a GBO based on the LEGP

Run	h_2	A	R_2	R_1	f_{obj2}
1	57.40128	0.157887	3.311413	5.942579	1129.561463
2	45.26694	0.152597	3.226155	5.8442	853.4763615
3	48.98618	0.222489	3.157694	5.918605	915.509757
4	32.27097	0.194206	3.217166	5.97835	620.6787187
5	57.46627	0.214215	3.119393	5.748713	1030.513524
6	40.41741	0.157574	3.114831	5.703932	718.0874355
7	50.6326	0.169845	3.297265	5.90975	986.6271951
8	54.04965	0.220564	3.195539	5.836084	1007.995303
9	46.48875	0.245294	3.189204	5.851694	867.5845803
10	51.28276	0.179077	3.178343	5.976067	974.0640628
11	56.23774	0.152181	3.256936	5.863454	1073.966373
12	42.95625	0.174857	3.102174	5.894946	785.5472124
13	52.33389	0.155133	3.33319	5.951809	1038.226451
14	54.72452	0.221956	3.376278	5.98674	1106.141125
15	47.29105	0.191712	3.128933	5.646466	835.510537
16	48.00857	0.166523	3.295603	5.905486	934.3494325
17	56.98286	0.218793	3.285613	5.866266	1098.303595
18	56.9222	0.144987	3.125514	5.91272	1051.938866
19	46.19192	0.193511	3.111921	5.582244	802.4229603
20	31.96327	0.160249	3.125179	5.952396	594.5903487
21	52.14464	0.229889	3.328248	5.888502	1021.951462
22	49.90867	0.233568	3.133448	5.86657	917.4507097
23	49.84812	0.140284	3.116203	5.631328	874.7529775
24	50.54733	0.226675	3.122786	5.923438	935.0056141
25	57.52945	0.163827	3.268732	5.759444	1083.053909
26	32.30148	0.157218	3.139406	5.784851	586.6271031
27	56.02531	0.201424	3.100072	5.971138	1037.081919
28	51.44141	0.168626	3.161804	5.712825	929.1776024
29	31.19217	0.169806	3.171518	5.947532	588.3686953
30	**31.08461**	**0.168752**	**3.15661**	**5.929514**	**581.8158272**

Table 4: The best solution for 6-kW from a GBO, TSA and ChOA algorithms based on the LEGP

Algorithm	h_2	A	R_2	R_1	f_{obj2}
GBO	31.08461	0.168752	3.15661	5.929514	**581.8158272**
TSA	33.95042	0.151096	3.190784	5.859134	634.7109261
ChOA	32.71091	0.164894	3.130075	5.811248	594.9997839

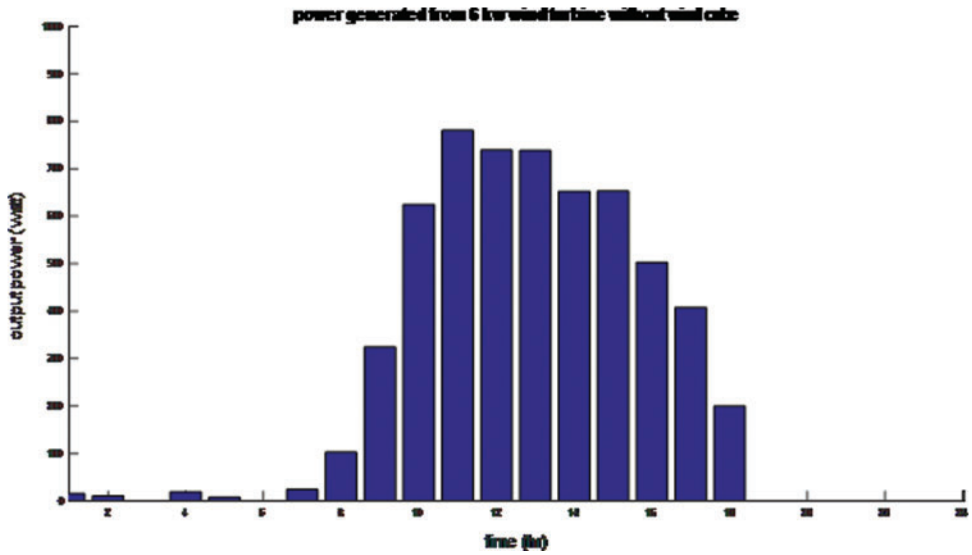

Figure 4: Power generated from a 6-kW wind turbine without a wind cube

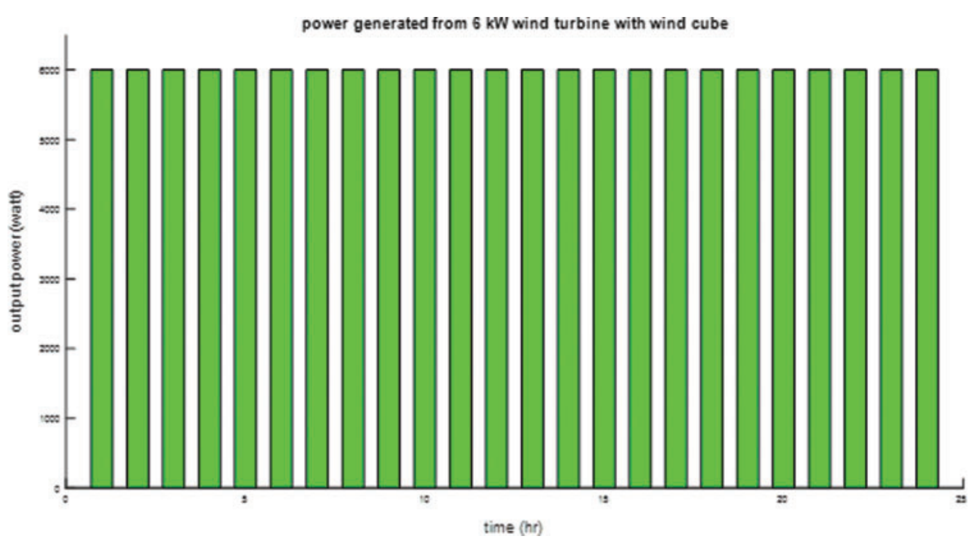

Figure 5: Output power of a 6-kW wind turbine with an optimized wind cube

Tab. 5 shows that all runs achieve 0.079240007 probability of a generated energy loss; the result in run 23 is the optimum solution because this solution achieves the minimum objective function of minimizing the meter cubic function. Tab. 6 show the best solution from the proposed GBO algorithm in compared with Tunicate swarm algorithm (TSA) and Chimp optimization algorithm (ChOA) for 30-kW wind turbine. Based on this results the proposed GBO achieve the best meter cubic function compared with the other competitor algorithms. Fig. 6 shows the power demand output from the wind turbine without a wind cube at the same height as the wind turbine hub and the optimum solution's roughness coefficient. Fig. 7 shows the power demand output from the wind turbine with a wind cube at the optimum solution for run 30 of the proposed GBO algorithm.

Table 5: Results of the decision variable for 30 kW from a GBO based on the LEGP

Run	h_2	α	R_2	R_1	f_{obj2}
1	33.9903	0.168467	5.576217	10.58872	2006.95769
2	41.23584	0.176445	5.90803	11.04344	2690.430679
3	59.99437	0.249611	5.947948	10.66068	3804.191615
4	34.07007	0.150989	5.593785	10.59017	2018.280502
5	41.17549	0.160371	5.915031	11.0462	2690.349584
6	59.99185	0.24995	5.94891	10.66202	3805.125321
7	56.3486	0.17893	5.787966	10.52267	3431.904241
8	46.87051	0.153597	5.65338	10.44837	2768.576657
9	59.66264	0.243826	5.732932	10.28951	3519.441572
10	46.07201	0.21587	5.72789	10.60664	2799.044078
11	56.52102	0.155246	5.551946	10.11289	3173.441056
12	51.64603	0.24933	5.552197	10.13934	2907.444836
13	33.72516	0.205779	5.799582	11.08859	2168.836732
14	35.23807	0.212581	5.807014	11.06361	2263.924172
15	53.90231	0.147021	5.881671	10.75907	3411.007878
16	30.02507	0.223701	5.841212	11.34799	1990.24277
17	45.81277	0.17039	5.893233	10.91312	2946.381392
18	39.48589	0.229741	5.575304	10.51012	2313.7594
19	34.81782	0.249944	5.751105	11.03274	2209.204985
20	36.73943	0.15563	5.635884	10.61535	2198.004659
21	49.66252	0.155886	5.887991	10.83304	3167.716883
22	32.88174	0.18592	5.763045	11.00836	2086.072849
23	**31.16894**	**0.199724**	**5.550141**	**10.68478**	**1848.380943**
24	39.46348	0.149229	5.55141	10.39238	2276.742437
25	50.89265	0.169445	5.929976	10.88363	3284.594438
26	59.15301	0.175359	5.66672	10.26178	3439.786336
27	44.19609	0.154358	5.8718	10.90138	2829.023782
28	59.96046	0.24965	5.844689	10.47629	3671.418304
29	56.1891	0.216415	5.757975	10.43581	3376.356052
30	35.94489	0.190275	5.571909	10.56166	2115.30706

Table 6: The best solution for 30-kW from a GBO, TSA and ChOA algorithms based on the LEGP

Algorithm	h_2	A	R_2	R_1	f_{obj2}
GBO	31.16894	0.199724	5.550141	10.68478	**1848.380943**
TSA	30.36002	0.170384	5.73696	11.00257	1916.363658
ChOA	30	0.25	5.640492	11.02396	1865.416509

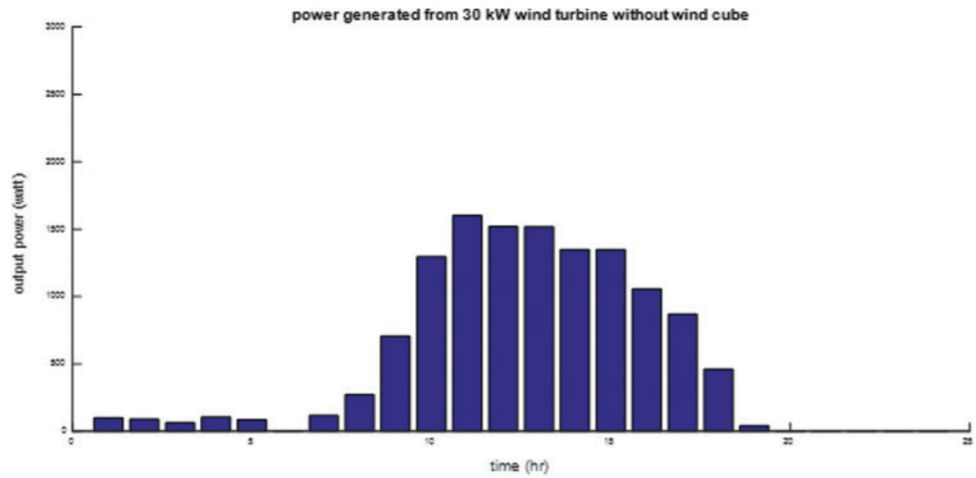

Figure 6: Power generated from 30-kW wind turbine without a wind cube

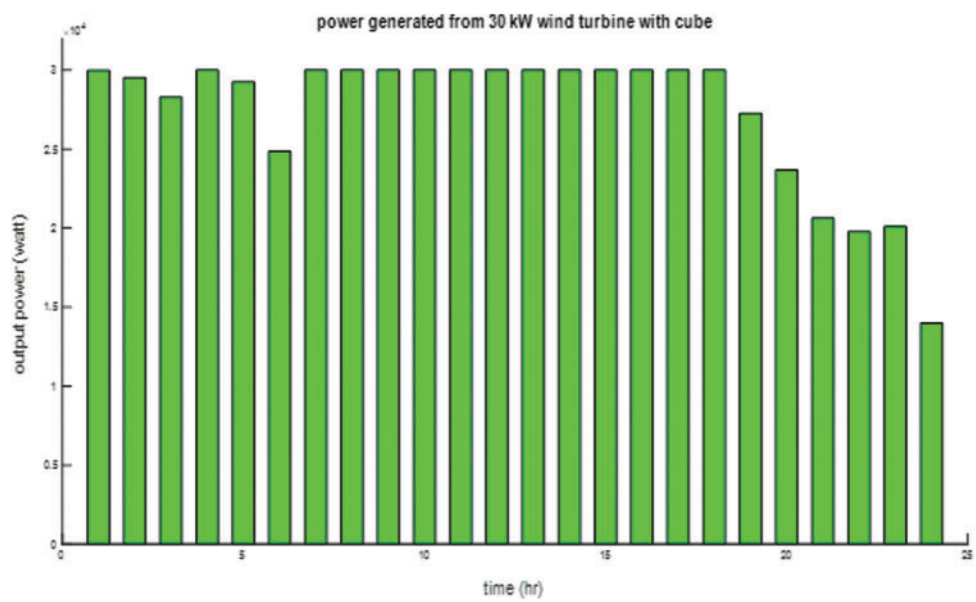

Figure 7: Output power of a 30-kW wind turbine with the optimized wind cube

5 Conclusion

Increasing the generated energy from the wind turbine is important work. Wind cubes improve wind turbine output using an effective optimization technique. A GBO is used to estimate the roughness factor's decision variable, the inner radius of the wind cube, the wind turbine hub's height, and the outer radius of the wind cube. The extraction of these parameters is dependent on minimizing the probability of a loss of generated energy and decreasing the decision variable to make these variables more cost-efficient. There is a tolerance of a 20% air speed increase for the site as compared with the speed recorded in this work. A comparison between wind turbine output power with and without the optimized wind cube was performed. Based on this comparison, the energy generated from a 30-kW wind turbine with the optimized wind cube as 55.7317 times the energy generated without the wind cube. The energy generated from a 6-kW wind turbine with the optimized wind cube is 23.8123 times the energy generated without the wind cube. The future work will concentrate on apply GBO for several problems such as power flow in power system, wind farm layout problem.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] X. Xu, W. Hu, D. Cao, Q. Huang, C. Chen *et al.,* "Optimized sizing of a standalone pv-wind-hydropower station with pumped-storage installation hybrid energy system," *Renewable Energy*, vol. 147, no. 4, pp. 1418–1431, 2020.
[2] A. Makhsoos, H. Mousazadeh, S. S. Mohtasebi, M. Abdollahzadeh and H. Jafarbiglu, "Design, simulation and experimental evaluation of energy system for an unmanned surface vehicle," *Energy*, vol. 148, no. 5, pp. 362–372, 2018.
[3] H. Hou, T. Xu, X. Wu, H. Wang and A. Tang, "Optimal capacity configuration of the wind-photovoltaic-storage hybrid power system based on gravity energy storage system," *Applied Energy*, vol. 271, no. 2, pp. 115052, 2020.
[4] D. Zhang, J. Wang, Y. Lin, Y. Si, C. Huang *et al.,* "Present situation and future prospect of renewable energy in china," *Renewable and Sustainable Energy Reviews*, vol. 76, pp. 865–871, 2017.
[5] B. Sun, Y. Yu and C. Qin, "Should china focus on the distributed development of wind and solar photovoltaic power generation? A comparative study," *Applied Energy*, vol. 185, no. 2, pp. 421–439, 2017.
[6] K. Balasubramanian, S. B. Thanikanti, U. Subramaniam, N. Sudhakar and S. Sichilalu, "A novel review on optimization techniques used in wind farm modelling," *Renewable Energy Focus*, vol. 35, no. 4, pp. 84–96, 2020.
[7] W. Sabry, A. Eliwa, U. AbouZayed and R. Youssef, "Design and implementation of the 10 kw wind cube," in *Proc. 21st Int. Conf. on Electricity Distribution*, Frankfurt, Germany, pp. 1–4, 2011.
[8] A. F. Abdelgawad, W. Sabry, U. R. Abouzayed, R. Y. Abdellateef and M. E. Fathy, "The 30 kva permanent magnet synchronous generator wind-cube design, implementation, control; and performance measurements and analysis," in *Proc. 20th Int. Middle East Power Systems Conf. (MEPCON)*, Cairo, Egypt, IEEE, pp. 260–265, 2018.
[9] M. H. Qais, H. M. Hasanien and S. Alghuwainem, "Parameters extraction of three-diode photovoltaic model using computation and harris hawks optimization," *Energy*, vol. 195, no. 12, pp. 117040, 2020.
[10] M. A. Soliman, H. M. Hasanien and A. Alkuhayli, "Marine predators algorithm for parameters identification of triple-diode photovoltaic models," *IEEE Access*, vol. 8, pp. 155832–155842, 2020.
[11] Q. Hao, Z. Zhou, Z. Wei and G. Chen, "Parameters identification of photovoltaic models using a multi-strategy success-history-based adaptive differential evolution," *IEEE Access*, vol. 8, pp. 35979–35994, 2020.
[12] B. Subudhi and R. Pradhan, "Bacterial foraging optimization approach to parameter extraction of a photovoltaic module," *IEEE Transactions on Sustainable Energy*, vol. 9, no. 1, pp. 381–389, 2017.
[13] V. J. Chin, Z. Salam and K. Ishaque, "An accurate modelling of the two-diode model of pv module using a hybrid solution based on differential evolution," *Energy Conversion and Management*, vol. 124, no. 8, pp. 42–50, 2016.
[14] A. Rezaee Jordehi, "Enhanced leader particle swarm optimisation (ELPSO): An efficient algorithm for parameter estimation of photovoltaic (PV) cells and modules," *Solar Energy*, vol. 159, pp. 78–87, 2018.

[15] A. R. Jordehi, "Time varying acceleration coefficients particle swarm optimisation (TVACPSO): A new optimisation algorithm for estimating parameters of PV cells and modules," *Energy Conversion and Management*, vol. 129, no. 4, pp. 262–274, 2016.

[16] H. M. Hasanien, "Shuffled frog leaping algorithm for photovoltaic model identification," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 2, pp. 509–515, 2015.

[17] Q. Yang, J. Hu and S. -S. Law, "Optimization of wind farm layout with modified genetic algorithm based on boolean code," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 181, no. 1, pp. 61–68, 2018.

[18] L. Wang, J. Yuan, M. E. Cholette, Y. Fu, Y. Zhou *et al.,* "Comparative study of discretization method and monte carlo method for wind farm layout optimization under weibull distribution," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 180, no. 1, pp. 148–155, 2018.

[19] A. M. Abdelsalam and M. El-Shorbagy, "Optimization of wind turbines siting in a wind farm using genetic algorithm based local search," *Renewable Energy*, vol. 123, no. 1, pp. 748–755, 2018.

[20] S. Zergane, A. Smaili and C. Masson, "Optimization of wind turbine placement in a wind farm using a new pseudo-random number generation method," *Renewable Energy*, vol. 125, no. 51, pp. 166–171, 2018.

[21] B. DuPont, J. Cagan and P. Moriarty, "An advanced modeling system for optimization of wind farm layout and wind turbine sizing using a multi-level extended pattern search algorithm," *Energy*, vol. 106, no. 1, pp. 802–814, 2016.

[22] G. E. -B. M. S. El-Negamy and A. Galal, "Controller energy management for hybrid renewable energy system," *International Journal of Engineering and Applied Sciences IJEAS*, vol. 3, no. 9, pp. 72–77, 2016.

[23] M. Said and A. Yassin, "Operation study of a combined system of diesel engine generators and wind turbine with battery storage system," in *Proc. 5th Int. Conf. on New Paradigms in Electronics & information Technology (PEIT)*, Egypt, IEEE, Hurgada, pp. 1–5, 2019.

[24] A. A. Masud, "An optimal sizing algorithm for a hybrid renewable energy system," *International Journal of Renewable Energy Research*, vol. 7, pp. 1595–1602, 2017.

[25] S. Moghaddam, M. Bigdeli, M. Moradlou and P. Siano, "Designing of stand-alone hybrid pv/wind/battery system using improved crow search algorithm considering reliability index," *International Journal of Energy and Environmental Engineering*, vol. 10, no. 4, pp. 429–449, 2019.

[26] I. Ahmadianfar, O. Bozorg-Haddad and X. Chu, "Gradient-based optimizer: A new metaheuristic optimization algorithm," *Information Sciences*, vol. 540, pp. 131–159, 2020.

[27] A. A. K. Ismaeel, E. H. Houssein, D. Oliva and M. Said, "Gradient-based optimizer for parameter extraction in photovoltaic models," *IEEE Access*, vol. 9, pp. 13403–13416, 2021.

[28] S. Deb, D. S. Abdelminaam, M. Said and E. H. Houssein, "Recent methodology-based gradient-based optimizer for economic load dispatch problem," *IEEE Access*, vol. 9, pp. 44322–44338, 2021.

[29] S. Kaur, L. K. Awasthi, A. L. Sangal and G. Dhiman, "Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization," *Engineering Applications of Artificial Intelligence*, vol. 90, no. 2, pp. 103541, 2020.

[30] M. Khishe and M. R. Mosavi, "Chimp optimization algorithm," *Expert Systems with Applications*, vol. 149, pp. 1–26, 2020.