Journal of Control, Automation and Electrical Systems (2020) 31:970–978
https://doi.org/10.1007/s40313-020-00600-0

Novel Exploration Coefficient Update for the Grey Wolf Optimizer

Frederico F. Panoeiro[1] · Gustavo Rebello[1] · Vinicius Cabral[1] · Ivo C. S. Junior[1] · Francisco C. R. Coelho[2] · Edmarcio A. Belati[3]

Received: 25 July 2019 / Revised: 20 February 2020 / Accepted: 17 April 2020 / Published online: 22 May 2020

Abstract

Avoiding stagnation at local optimum values is one of the greatest challenges faced by computational intelligence techniques when solving nonconvex optimization problems. The transition between global and local search may not be effective and can compromise the performance of optimization algorithms. This work presents a novel manner to update the exploration coefficient of the meta-heuristic known as grey wolf optimizer (GWO), by replacing the linear update of the exploration coefficient by a triangular-shaped function, enabling the algorithm to escape from local optima. In order to validate the proposed grey wolf optimizer (PGWO) methodology, its performance is compared to the original version of GWO and its chaotic version, as well as to the well-known genetic algorithm, bat algorithm and particle swarm optimization techniques, in solving 10 nonconvex benchmark functions. Also, in order to verify the proposed methodology's ability in solving a more realistic engineering problem, the authors implemented the PGWO to solve the wind farm layout optimization (WFLO) problem, which is a large-sized optimization problem, of combinatorial nature and nonconvex solution region. The results indicate that the PGWO improved the performance of the original GWO, as well as all investigated methodologies for the benchmark functions optimization and for the WFLO problem.

Keywords Grey wolf optimizer · Exploration coefficient · Computational intelligence · Benchmark functions · Wind farm layout optimization

1 Introduction

Optimization problems can be classified according to the type of constraints (linear or nonlinear), optimization variables (continuous, discrete or binary) and solution region convexity, which is an important factor when determining the methodology implemented in its resolution (Calafiore and El Ghaoui 2014).

Classic optimization methods, when applied to solve nonconvex problems, tend to converge to local optima, since they are highly susceptible to initial conditions (Bazaraa et al. 2013). In order to solve this issue, the union of concepts from optimization and artificial intelligence made it possible to develop more efficient and intelligent heuristic strategies, named as meta-heuristics or computational intelligence (Neri and Cotta 2012). Computational intelligence is defined as the science that develops methodologies that are able to mimic similar behaviors of intelligent systems (human or bioinspired), aiming at solving complex optimization problems.

Given this context, in the last years, several researchers have been developing new approaches to solve optimization problems through computational intelligence techniques, being inspired by natural existing biological adaptation mechanisms. Algorithms inspired in the evolution of species theory, such as the genetic algorithm (GA) (Holland 1992), swarm intelligence, such as the particle swarm optimization (PSO) (Eberhart and Kennedy 1995), bats echolocation, such as the bat algorithm (BA) (Yang 2010), among others have been developed as a result of such an effort. These algorithms have presented great efficiency in solving not only engineering-related problems, but also in other fields of knowledge. Currently, these bioinspired algorithms are a part of a set of tools associated with a great number of high-technology projects.

✉ Gustavo Rebello
gustavo.rebello@engenharia.ufjf.br

[1] Department of Electrical Energy, Federal University, Juiz de Fora, MG, Brazil

[2] Electrical Engineering Department, Federal University, São João del-Rei, MG, Brazil

[3] Electrical Energy Department, Federal University of ABC, Santo André, SP, Brazil

One of the greatest challenges faced by computational intelligence techniques is the balance between conducting local or global search throughout the iterative process, generally performed through exploration coefficients that are updated over the course of iterations. This inherent limitation to the computational intelligence techniques may negatively affect the efficiency of these algorithms in solving large-sized optimization problems, making them stagnate at local optimum values.

In this sense, this paper presents an adaptation never seen in the literature, to perform the update of the exploration coefficient of the bioinspired computational intelligence technique known as grey wolf optimizer (GWO) (Seyedali et al. 2014), being this the main contribution of this work. The methodology validation will be performed through the comparison of the proposed approach with the GA, BA and PSO techniques in 10 benchmark functions. Besides that, to prove the effectiveness of the proposed methodology on solving real large-sized engineering optimization problems, a study case applied to the optimization of the layout of an offshore wind farm will be presented.

The content of this paper is divided as follows: Sect. 2 presents the computational intelligence technique known as GWO, as well its chaotic (Kohli and Arora 2018) and the proposed version by the authors. In Sect. 3, the proposed methodology is presented and validated on several benchmark functions and compared to the original and chaotic version of GWO, and also to the GA, BA and PSO techniques. Section 4.1 presents the wind farm layout optimization (WFLO) problem formulation, and the results obtained by the proposed methodology are discussed and compared to the aforementioned techniques. A conclusion is presented in Sect. 5.

2 Grey wolf optimizer: novel exploration coefficient

The GWO is a bioinspired meta-heuristic presented by Seyedali et al. (2014) in 2014, and it is an algorithm that is being applied in many researches (El-Fergany and Hasanien 2015; Routray et al. 2019; Faris et al. 2018; Yan et al. 2019; Mostafa et al. 2018; Verma et al. 2017; Hernndez et al. 2019; Panoeiro et al. 2020). The algorithm mathematically models the hunting and social behavior of grey wolves to solve optimization problems. Grey wolves generally live on a pack, that is basically composed of four types of wolves: alpha (α), beta (β), delta (δ) and omega (ω).

The hunting strategy of grey wolves is divided into three steps, as follows: (1) tracking, chasing and approaching the prey; (2) chasing and encircling until stationary situation and (3) attacking the prey.

In the optimization context, the three best candidate solutions obtained by the algorithm are depicted by the α, β and δ wolves. These solutions are used in the algorithm to govern the searching process. The remaining solutions are called ω, and over the course of iterations they encircle the α, β and δ wolves in order to attempt to reach better solutions.

Algorithm 1 depicts a pseudocode of the GWO method. At first, a random population composed of N_{wolves} is created and evaluated through an objective function (OBF). Then, the hierarchy is established and the iterative process starts, until a stopping criteria are met. Then, from lines 7–11 the algorithm emulates the hunting behavior of grey wolves, and more details of the presented pseudocode can be found in reference (Seyedali et al. 2014).

Algorithm 1 Grey Wolf Optimizer

```
1: Initialize Wolves Population: ($X_i$, with $i = 1, 2, ...N_{\text{wolves}}$)
2: Evaluate Pack: OBF($X_i$) (Eq. (11))
3: Define Hierarchy: ($X_\alpha, X_\beta, X_\delta$)
4: While stopping criteria not met, do:
5:     Update exploration coefficient $\vec{a}_t$ linearly or through equations (1), (2) and (3)
6:     For each wolf ∈ ($i = 1, 2, ...N_{\text{wolves}}$) do:
7:         $\vec{A} = 2\vec{a}_t \cdot \vec{r_1} - \vec{a}_t$,        $\vec{r_1} \in [0, 1]$
8:         $\vec{C} = 2 \cdot \vec{r_2}$,        $\vec{r_2} \in [0, 1]$
9:         $\vec{D_\alpha} = |\vec{C_1} \cdot \vec{X_\alpha}^t - \vec{X_i}^t|$,    $\vec{D_\beta} = |\vec{C_2} \cdot \vec{X_\beta}^t - \vec{X_i}^t|$,    $\vec{D_\delta} = |\vec{C_3} \cdot \vec{X_\delta}^t - \vec{X_i}^t|$
10:        $\vec{X_1} = |\vec{X_\alpha}^t - \vec{A_1} \cdot \vec{D_\alpha}|$,    $\vec{X_2} = |\vec{X_\beta}^t - \vec{A_2} \cdot \vec{D_\beta}|$,    $\vec{X_3} = |\vec{X_\delta}^t - \vec{A_3} \cdot \vec{D_\delta}|$
11:        $\vec{X_i}^{t+1} = \frac{\vec{X_1}+\vec{X_2}+\vec{X_3}}{3}$
12:        Evaluate Pack and Update Hierarchy: ($X_\alpha, X_\beta, X_\delta$)
13:    end For
14: end While
15: Return $X_\alpha$
```

As stated in Seyedali et al. (2014) and depicted in Fig. 1, the algorithm performs local search, i.e., attacks the prey, if $|\vec{A}| < 1$, and global search, i.e., searches for other preys, if $|\vec{A}| > 1$. As for the coefficient $\vec{C}$ its value attenuates or increases the magnitude of the best solution during the searching mechanism. Vectors $\vec{r_1}$ and $\vec{r_2}$ are randomly distributed values ∈ [0,1] and help the algorithm to avoid stagnation at local minima. Parameters $\vec{D_\alpha}$, $\vec{D_\beta}$ and $\vec{D_\delta}$ represent the distance between an omega wolf and the dominant wolves. Parameters $\vec{X_1}$, $\vec{X_2}$ and $\vec{X_3}$ weigh this distance input by the parameter $\vec{A}$ and defines if a local/global search will be performed. At last, line 11 depicts the position update taking into account the information from the dominant wolves α, β and δ. At the end of the iterative process, the best solution X_α is returned.

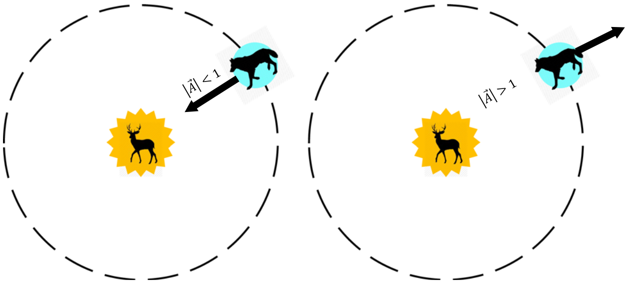

Fig. 1 Global and local searching components of GWO

Line 5 highlights the importance of the exploration coefficient $\vec{a}_t$, since it governs whether the algorithm will perform a local or global search. In Seyedali et al. (2014), the authors proposed a linear decaying function in order to update the exploration coefficient $\vec{a}_t$. In this paper, however, the authors propose a triangular-shaped function to update the exploration coefficient, giving rise to a version called proposed grey wolf optimizer (PGWO). Also, the concept of chaotic optimization is taken into account by the authors and two chaotic maps are implemented to replace the linear-decaying exploration coefficient $\vec{a}_t$.

Swarm intelligence algorithms have proven themselves as an excellent tool to solve optimization problems. However, one of the greatest challenges faced by these algorithms is related to the stagnation in local optimum values after a certain number of iterations. To avoid such thing and improve algorithm's performance, several works in the literature are proposing modifications in the original versions of the meta-heuristics. Within the scope of GWO, this is no different: in Kaur and Narang (2019), a hybrid version of the GWO, introducing three mutation strategies, is proposed. In Wen (2016) and Mittal et al. (2016), the authors propose a nonlinear adjustment of the exploration coefficient $\vec{a}_t$. In Zhang et al. (2017), the opposite-learning technique is used to obtain a better set of initial solutions for the studied problem.

In Kohli and Arora (2018) the authors propose the use of chaotic maps to update parameters $\vec{A}$, $\vec{C}$ and $\vec{a}_t$ to explore the search space more dynamically. The use of chaotic maps in the exploration parameters update of computational intelligence techniques has been widely studied by researchers in the past few years. In Gandomi and Yang (2014), chaotic maps are used to update parameters of the BA. In Gandomi et al. (2013), the authors evaluate the performance of 12 chaotic maps to update parameters of the firefly optimization (FO) algorithm. In Amir Hossein et al. (2013), the update is performed in the PSO method. At last, in Emary and Zawbaa (2016) the authors analyze the impact of chaotically updating the exploration coefficient in different meta-heuristics, among them the GWO, the ant lion optimizer (ALO) and the moth-flame optimizer (MFO). In general, by analyzing the works that introduce chaos in meta-heuristics, it can be observed that the updates following chaotic maps happen in a pseudorandom fashion, indicating a more dynamic searching process, contributing to avoid stagnation.

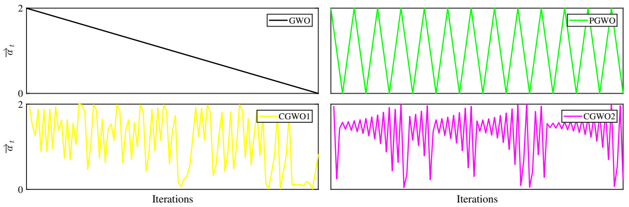

Fig. 2 Exploration coefficient update

Following this line of research, this work implements 2 different chaotic maps, taken from Gandomi and Yang (2014), to update the exploration coefficient $\vec{a}_t$. The algorithms following these chaotic updates are named chaotic grey wolf optimizer (CGWO) in this paper and are represented by Eqs. (1) (CGWO1) and (2) (CGWO2). Figure 2 graphically depicts the differences between the chaotic and linear updates.

$$x_{t+1} = \cos(t\cos^{-1}(x_t)) \tag{1}$$

$$x_{t+1} = 1{,}07\left(7{,}86x_t - 23{,}31x_t^2 + 28{,}75x_t^3 - 13{,}3x_t^4\right). \tag{2}$$

Besides the chaotic version of GWO, this work proposes the update of the exploration coefficient $\vec{a}_t$ in a novel manner, not yet seen in the specialized literature. This novel approach aims at introducing to the GWO a more dynamic transition between local and global search, enabling it to perform local and global searches throughout the entire execution of the algorithm, improving its capability of avoiding local optimal values even in the last few iterations. This modification improves the algorithm's performance, and it is the major contribution of this paper. The proposed modification gives rise to a version named PGWO by the authors and represents the main novelty and contribution of this paper. The proposed function consists on a triangle-pulsed wave, shown in Eq. (3). At this point, it is important to highlight that the authors have tested different similar functions, but chose to present this one at this work, since it presented the best results obtained so far. Figure 2 also depicts the proposed approach in comparison with the traditional linear update of the exploration coefficient originally proposed in Seyedali et al. (2014).

$$x_{t+1} = \frac{2}{\pi}|\sin^{-1}[\sin(\pi t)]|. \tag{3}$$

3 Method validation

In order to validate the PGWO method, the authors compared its performance with the traditional version of the GWO algorithm, as well as its chaotic version, both presented in Sect. 2. Besides that, a comparison with respect to the well-known GA, BA and PSO techniques is also presented, in the optimization of 10 nonconvex benchmark functions.

Table 1 depicts the parameters adopted for each optimization method. All techniques populations were composed of 40 individuals, and the stopping criterion adopted for all algorithms was the maximum allowed number of 40 iterations. The remaining parameters shown in Table 1 are values recommended by the authors of the methods. It should be noticed that the GWO is the method with the least number of parameters to adjust.

In order to evaluate the PGWO robustness to solve a wide range of optimization problems, the authors performed 1000 simulations of each methodology to solve each one of the 10 benchmark functions. It should be highlighted that all methodologies started from the same initial solutions, in order to make the comparison with other techniques fairer. The mathematical functions as well as the coordinates of the global optimal results and the variables boundaries can be obtained from Ali et al. (2005).

The methodologies assertiveness rates, i.e., the percentage of simulations that achieved the global optimum result for each function, are depicted in Table 2, and it can be seen that the PGWO method obtained higher assertiveness rates for almost every benchmark function tested. In five opportunities, the proposed methodology reached the global optimum value for all 1000 simulations performed. The mean value of assertiveness shows that the PGWO has shown itself as a very competitive tool with respect to other methodologies to solve optimization problems, reaching the global optimum value in 97.4% of times, outperforming the original GWO version. It can also be highlighted that, for the functions under consideration, the chaotic versions of GWO did not have a positive impact in the algorithm's overall performance.

Figure 3 depicts, in a form of a boxplot, the results shown in Table 2. From Fig. 3 it can be observed that the results obtained from the proposed method were less dispersed with respect to all other investigated methodologies, reinforcing the idea that this is a fairly robust method to solve nonconvex optimization problems. It can also be highlighted that the PGWO method presented the highest median value. This improvement is explained by the greater dynamics introduced by the novel exploration coefficient update, enabling the algorithm to escape from stagnating at local optimum values over the course of iterations.

It can also be emphasized that the GWO, CGWO1 and CGWO2 techniques have also outperformed the GA, BA and PSO algorithms, indicating that the use of the three best solutions to govern the searching process is indeed effective, as stated in Seyedali et al. (2014).

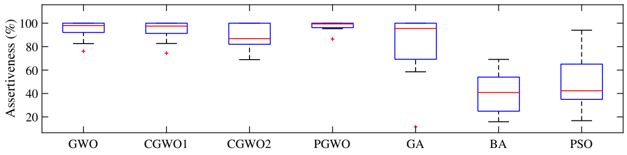

Fig. 3 Methodologies assertiveness over 10 benchmark functions

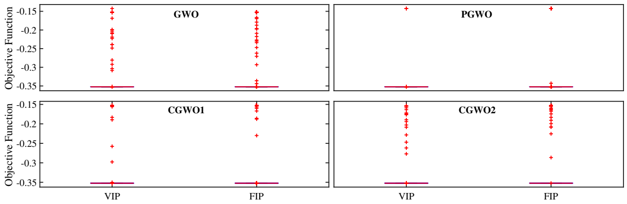

Fig. 4 Methods robustness regarding initial solutions. (*Aluffi-Pentini* function)

3.1 Proposed grey wolf optimizer robustness regarding initial solution

Besides the capacity of finding good solutions, it is desirable that a computational intelligence technique performs well regardless of its initial set of solutions (Vasant et al. 2016). This characteristic may be understood as a way to measure a method's robustness with respect to the initial set of candidate solutions.

To evaluate this important aspect, 1000 simulations of the 4 GWO-based methodologies were performed to minimize the *Aluffi-Pentini* function in two situations. For the first situation, all techniques (GWO, PGWO, CGWO1 and CGWO2) start from a fixed initial population, the same way as for the study regarding all benchmark functions previously presented. For the second situation, the algorithm's repetition was made starting from different random initial solutions, coming from an uniform distribution.

Figure 4 depicts the boxplot of all methodologies in obtaining the optimal result of -0.35 for the *Aluffi-Pentini* function. In this context, FIP stands for fixed initial population and VIP for variable initial population.

In order to verify whether the samples depicted in Fig. 4 are statistically equal, a test hypothesis was performed. The *t test for two samples*, described in Montgomery and Runger (2010), allows one to verify the probability that the differences between two samples are not caused by chance, and it is called p value. It is usual to adopt a significance level of 5%, meaning that if the p value is less than 5%, the *null hypothesis* that establishes equality between samples is rejected.

Considering this significance level, the authors applied this hypothesis test for the samples obtained for each one of

Table 1 Parameters adopted for the compared meta-heuristics

Method	Parameter	Value	Description
GWO, CGWO & PGWO	*ITE*	40	Number of iterations
	N	40	Population size
GA	ite	40	Number of iterations
	n	40	Population size
	C_p	0.8	Crossover probability
	M_p	0.01	Mutation probability
	R_p	0.05	Population reproduction probability
BA	ite	40	Number of iterations
	n	40	Population size
	α	0.9	Amplitude decay coeff.
	γ	0.8	Pulse emission coeff.
	r^0	1	Maximum pulse emission rate
	$f_{\min}$	0	Minimum frequency
	$f_{\max}$	0.15	Maximum frequency
PSO	ite	40	Number of iterations
	n	40	Population size
	c_1	0.3	Local info weighing
	c_2	0.7	Global info weighing
	θ	1	Speed inertia factor

Table 2 Methodologies assertiveness rates

Function	GWO (%)	CGWO1 (%)	CGWO2 (%)	PGWO (%)	GA (%)	BA (%)	PSO (%)
Ackley	99.5	99.2	84.1	**100**	11.5	52.3	43.0
Aluffi-Pentini	92.8	93.3	82.0	**96.2**	100	41.4	41.6
Becker and Lago	98.9	99.3	99.4	**86.5**	100	24.9	23.0
Bohachevsky 1	98.5	95.9	87.9	**100**	88.3	37.1	38.9
Bohachevsky 2	92.6	91.3	85.7	**95.3**	91.0	15.9	35.0
Camel Back 3	77.8	74.4	68.9	**98.8**	69.2	55.6	65.1
Cosine Mixture	100	100	100	**100**	100	54.0	67.2
Easom	99.8	100	100	**97.4**	100	24.3	16.8
Eggcrate	81.1	82.7	79.1	**100**	58.5	40.3	44.5
Exponential	100	100	100	**100**	100	69.1	94.0
Mean	94.1	93.6	88.7	**97.4**	81.8	38.3	44.9

Bold values highlights the solutions found by PGWO

the 4 methodologies. The p values obtained were: (1) GWO: 0.09%; (2) PGWO: 90.37%; (3) CGWO1: 32.64%; and (4) CGWO2: 40.46%. For the PGWO, CGWO1 and CGWO2, the p values obtained were greater than 5%, meaning that the null hypothesis cannot be rejected.

The results from the *t test for two samples* indicate that, for the proposed method, the samples do not have significant differences, indicating the method's robustness. Furthermore, the *p value* for the proposed method was the largest among all techniques studied, indicating its greatest robustness regarding initial solution conditions.

In addition, the authors make available the codes to evaluate all tested benchmark functions through the tested versions of the GWO via the following link: https://github.com/ufcod/pgwocodes.git.

4 Real problem: Wind Farm Layout Optimization

4.1 Problem Data

The development of renewable energy sources increases year after year worldwide. In this context, the use of wind energy has gained a lot of importance in the past decade. According to the last report produced by the Global Wind Energy Council (GWEC) (2018), in 2018 51.3 GW of power was added, totaling an installed capacity of 591 GW worldwide.

One of the factors that most affects the energy extraction of a wind farm is the turbines layout (Elkinton et al. 2005). The layout optimization of wind farms is normally given through the division of the terrain in cells, where each cell represents a possible position to allocate a wind turbine. Given the combinatorial nature of determining the optimal location of the wind turbines, the WFLO problem has been resolved through the application of computational intelligence techniques, able to solve large-sized combinatorial optimization problems in a reasonable computational time (Le and Vo 2017; Wu et al. 2020; Yang et al. 2019; Park et al. 2019; Pillai et al. 2018; Wilson et al. 2018; Feng and Shen 2015).

In this work, the authors implemented the 4 GWO versions explained in Sect. 2, as well as the GA, BA and PSO methods to determine the optimal layout of a wind farm of 500 MW of installed capacity. The wind farm under study has its area represented by a (10 × 10) matrix, at which each cell has dimensions of 900 m × 900 m, equivalent to 5 times the rotor diameter. This problem modeling is of binary nature, at which the total number of possible combinations is of 2^{100}, since each one of the 100 cells in the matrix can contain ('1') or not ('0') a wind turbine (Sittichoke et al. 2013). This huge number of combinations possibility demonstrates this problem complexity, justifying the use of computational intelligence techniques to solve it.

The wind turbine model used in this work is based on the DTU 10 MW (Bak et al. 2013), of 10-MW rated power. Since this work will optimize a wind farm of 500 MW of installed capacity, 50 units of the DTU10MW turbine will be considered in this study. Equation (4) describes the power curve, i.e., the electrical power extracted by this turbine in function of the speed of incident wind (u).

$$P_{\mathrm{kW}}(u) = \begin{cases} 0, & \text{if } u < 4 \\ 7.0564u^3, & \text{if } 4 \leq u < 11.4 \\ 10{,}000, & \text{if } 11.4 \leq u < 25 \\ 0, & \text{if } u \geq 25 \end{cases} \tag{4}$$

Regarding wind speed and direction, the authors consider a mean speed of 11.23 m/s, which corresponds to the rated wind speed of the DTU10MW turbine, and an incidence direction from north to south. Additional data necessary to model the problem, such as the rotor diameter, height of the hub, impulse coefficient (C_T) and roughness of the ground, are depicted in Table 3.

4.2 Wake effect

The conversion of the kinetic energy from the winds into electrical energy through the wind turbine is not ideal, i.e., there is a reduction of the wind's speed when passing through the turbine. In a wind farm, composed of many turbines, the ones located upstream reduce the wind velocity that will reach downstream turbines. This phenomenon is known as wake effect. According to Barthelmie and Jensen (2010), in offshore wind farms this effect can reduce the energetic potential of the venture between 10 and 15%.

Table 3 Additional data

Parameter	Value
Rotor diameter (m)	180
Rated power (MW)	10
C_T	0.88
Hub height (z) (m)	120
Soil roughness (z_0)	0.0002

Given this context, Jensen (1983) presented a simplified model of the wake effect, modeling the wind behavior after interacting with a wind turbine. According to Jensen, the interaction between wind and turbine is based upon the principle of movement conservation, as depicted in Eq. (5), and it creates a conical region that has the same radius of the turbine and linearly expands with distance.

$$\pi r_{\mathrm{turb}}^2 u + \pi(r_{\mathrm{est}}^2 - r_{\mathrm{turb}}^2)u_0 = \pi r_{\mathrm{est}}^2 u_j \tag{5}$$

where r_{turb} depicts the turbine radius and r_{est} the radius at a distance x from the wind turbine. The mean wind speed at the region is represented by u_0, while the wind speed right after interacting with a turbine is represented by u. The wind speed at a distance x from the wind turbine is represented by the variable u_j.

Figure 5 depicts the wake effect and its impact at the wind speed after interacting with the wind turbine i. The wind, with initial speed u_0, has its velocity reduced at the moment of energy extraction, being the wind velocity that reaches a downstream turbine j given by Eq. (6).

$$u_j = u_0 \left[1 - \left(\frac{2a}{1 + \alpha \left(\frac{x_{ij}}{r_i} \right)^2} \right) \right], \tag{6}$$

where x_{ij} is the position of the wind turbine, and r_i is the effective radius of the upstream rotor that is related to the downstream rotor radius r_j according to (7). Besides that, a is related to the impulse coefficient C_T, as shown in (8) and α is a drag constant obtained from (9), where z is the hub height and z_0 depicts the soil rugosity. This work considers $z_0 = 0.0002$, corresponding to an offshore region.

$$r_i = r_j \sqrt{\frac{1-a}{1-2a}} \tag{7}$$

$$C_T = 4a(1-a) \tag{8}$$

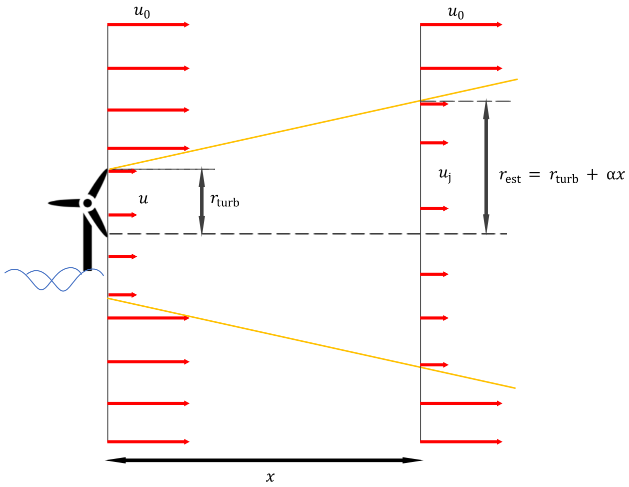

Fig. 5 Wake effect

$$\alpha = \frac{0,5}{\ln\left(\frac{z}{z_0}\right)}. \tag{9}$$

In case there is more than one turbine impacting the wind velocity, the energy decrease at each upstream turbine must be taken into account, as shown in (10).

$$u_i = u_0 \left[1 - \sqrt{\sum_{i=1}^{N} \left(1 - \frac{u_{ij}}{u_0}\right)^2} \right]. \tag{10}$$

4.3 Objective function

At the present work, the goal is to optimize the location of 50 DTU10MW wind turbines in order to maximize the extracted power of the wind farm, considering the wake effect previously explained. Thus, the OBF adopted is shown in Eq. (11), where P_{kW} represents the turbine extracted power and u_{n_T} depicts the wind velocity that reaches the turbine n_T (m/s) considering the wake effect.

$$\text{Max} \left(\sum_{n_T=1}^{50} P_{\text{kW}}(u_{n_T}) \right). \tag{11}$$

4.4 Simulation results

To obtain the statistical results presented in this work, 50 simulations of each technique were performed. In order to perform an unbiased comparison, all methodologies started the iterative process from the same initial solutions. In this study, the algorithm's stopping criteria adopted were the number of iterations, set to 300, for a population of 1000 individuals.

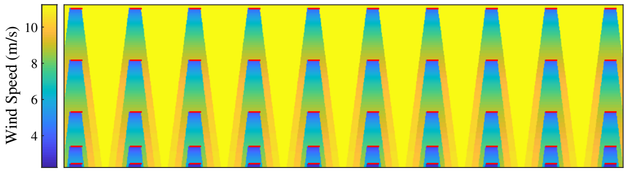

Fig. 6 Optimal layout obtained by the PGWO method

The optimal layout obtained for this study case was obtained by the PGWO method and is depicted in Fig. 6, where the wind turbines are represented by the red lines. In this graph it can also be observed the impacts in the wind velocity caused by the wake effect. For this layout, the extracted power is of 246.953864 MW, corresponding to a capacity factor of 49.39%.

Analyzing Table 4, it can be seen that the author's proposed grey wolf optimizer was the only method that reached the best OBF for this study (246.953864 MW). The CGWO1 method reached the best STD value. All GWO-based methods reached the same median value of 246.658266 MW. By comparing the PGWO and GWO methods, it can be observed that the OBF value of the simulations was also higher for the proposed methodology, with a difference of 1.72% in terms of STD. The PGWO overreaches the traditional GWO in almost every parameter analyzed, fact explained by the greater dynamics introduced by the nonlinear behavior in the exploration coefficient's $\vec{a}_t$ update, improving the algorithm's transition between local and global searching. Another factor to be highlighted is the fact that the only methodology that was able to obtain the optimal OBF for this real problem was the novel version proposed by the authors.

Given the high complexity of this problem, mainly due to its extremely high number of possible combinations, the assertiveness obtained by the PGWO was of 4%. By comparing the results obtained by the proposed methodology and the well-known GA, BA and PSO algorithms, it can be seen that the PGWO produces better results in all aspects analyzed, such as OBF, median and STD. From the results, however, it can be seen that the BA produced good results, obtaining the third best OBF value in this study.

5 Conclusion

At this work, the authors presented a novel way of updating the exploration coefficient of the GWO meta-heuristic. The proposed methodology was tested and validated at several benchmark functions and was compared to the chaotic and traditional versions of the GWO, as well as well-known methodologies such as the GA, BA and PSO. In general, for all 10 benchmark functions, the PGWO version obtained better results than all other techniques tested in this work,

Table 4 Statistical results

Method	OBF (MW)	Median (MW)	STD (MW)	Assertiveness (%)
GWO	246.879965	**246.658266**	0.113202	0
CGWO1	246.879965	**246.658266**	**0.112149**	0
CGWO2	246.879965	**246.658266**	0.114251	0
PGWO	**246.953864**	**246.658266**	0.115189	**4**
GA	241.448307	235.134939	3.116312	0
BA	246.572281	245.863347	0.324331	0
PSO	232.526499	227.746058	1.223905	0

Best results highlighted

with higher assertiveness rates and less dispersed results. This improvement highlights the importance of determining an well-adjusted exploration coefficient update to perform a more efficient global and local searching procedure over the course of iterations.

Given the high assertiveness and efficiency of the proposed methodology, the novel exploration coefficient was applied to solve a realistic engineering problem, consisting at determining the optimal layout of the wind turbines of an offshore wind farm, which is a highly complex multimodal optimization problem, of combinatorial nature and nonconvex solution region. The best result obtained in this paper for the WFLO problem was reached by the PGWO methodology. The PGWO overreached all other methodologies investigated (GWO, CGWO1, CGWO2, GA, BA and PSO) in most aspects analyzed (OBF, median and mean values).

For the WFLO problem, each OBF evaluation takes about 0.02 s. Therefore, it is important to highlight the impossibility of using exhaustive search mechanisms to obtain the global optimal value for this problem, since its immense number of possible combinations (2^{100}) would require a computational time of approximately 2.5×10^{28} days to be completed, reinforcing the benefits of using computational intelligence algorithms such as the proposed grey wolf optimizer to solve problems of such magnitude in an acceptable computational time.

References

Ali, M. M., Khompatraporn, C., & Zabinsky, Z. B. (2005). A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *Journal of Global Optimization*, *31*(4), 635. https://doi.org/10.1007/s10898-004-9972-2.

Amir Hossein, G., Gun Jin, Y., Xin-She, Y., & Siamak, T. (2013). Chaos-enhanced accelerated particle swarm optimization. *Communications in Nonlinear Science and Numerical Simulation*, *18*(2), 327. https://doi.org/10.1016/j.cnsns.2012.07.017.

Bak, C., Zahle, F., Bitsche, R., Kim, T., Yde, A., Henriksen, L., Andersen, P., Natarajan, A., & Hansen, M. (2013). The DTU 10-MW reference wind turbine. In Danish wind power research, 1, 1–111.

Barthelmie, R. J., & Jensen, L. (2010). Evaluation of wind farm efficiency and wind turbine wakes at the nysted offshore wind farm. *Wind Energy*, *13*(6), 573.

Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (2013). *Nonlinear programming: Theory and algorithms*. New York: Wiley.

Calafiore, G. C., & El Ghaoui, L. (2014). *Optimization models*. Cambridge: Cambridge University Press.

Eberhart, R., & Kennedy, J. (1995). Particle swarm optimization. In *IEEE international conference on neural networks* (Vol. 4, p. 1942).

El-Fergany, A. A., & Hasanien, H. M. (2015). Single and multi-objective optimal power flow using grey wolf optimizer and differential evolution algorithms. *Electric Power Components and Systems*, *43*(13), 1548–1559.

Elkinton, C. N., Manwell, J., & Mcgowan, J. (2005). Offshore Wind Farm Layout Optimization (OWFLO) Project: An introduction. *FME Transactions*, *38*, 1–26.

Emary, E., & Zawbaa, H. M. (2016). Impact of chaos functions on modern swarm optimizers. *PLOS ONE*, *11*(7), 1. https://doi.org/10.1371/journal.pone.0158738.

Faris, H., Aljarah, I., Al-Betar, M. A., & Mirjalili, S. (2018). Grey wolf optimizer: A review of recent variants and applications. *Neural Computing and Applications*, *30*(2), 413–435.

Feng, J., & Shen, W. Z. (2015). Solving the wind farm layout optimization problem using random search algorithm. *Renewable Energy*, *78*, 182–192.

Gandomi, A. H., & Yang, X. S. (2014). Chaotic bat algorithm. *Journal of Computational Science*, *5*(2), 224.

Gandomi, A. H., Yang, X. S., Talatahari, S., & Alavi, A. H. (2013). Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation*, *18*(1), 89. https://doi.org/10.1016/j.cnsns.2012.06.009.

Global Wind Energy Council (GWEC). (2018). Global wind report. https://gwec.net/members-area-market-intelligence/reports/. Accessed 17 Dec 2018.

Hernndez, E., Castillo, O., & Soria, J. (2019). Optimization of fuzzy controllers for autonomous mobile robots using the grey wolf optimizer. In *2019 IEEE international conference on fuzzy systems (FUZZ-IEEE)* (pp. 1–6). IEEE.

Holland, J. H. (1992). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. Cambridge: MIT Press.

Jensen, N. O. (1983). A note on wind generator interaction. *Riso National Laboratory*, *2*, 1–26.

Kaur, A., & Narang, N. (2019). Optimum generation scheduling of coordinated power system using hybrid optimization technique. *Electrical Engineering*, *101*, 379–408.

Kohli, M., & Arora, S. (2018). Chaotic grey wolf optimization algorithm for constrained optimization problems. *Journal of Computational Design and Engineering*, *5*(4), 458.

Le, T. T., & Vo, D. N. (2017). Optimal layout for off-shore wind farms using metaheuristic search algorithms. *GMSARN International Journal*, *11*, 1–15.

Mittal, N., Singh, U., & Sohi, B. S. (2016). Modified grey wolf optimizer for global engineering optimization. *Applied Computational Intelligence and Soft Computing*, *2016*, 8.

Montgomery, D. C., & Runger, G. C. (2010). *Applied statistics and probability for engineers*. New York: Wiley.

Mostafa, E., Abdel-Nasser, M., & Mahmoud, K. (2018). Application of mutation operators to grey wolf optimizer for solving emission-economic dispatch problem. In *2018 international conference on innovative trends in computer engineering (ITCE)* (pp. 278–282). IEEE.

Neri, F., & Cotta, C. (2012). Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation*, *2*, 1.

Panoeiro, F. F., Rebello, G., Cabral, V. A., Moraes, C. A., da Silva Junior, I. C., Oliveira, L. W., et al. (2020). Frontier applications of nature inspired computation. In *Application of recent metaheuristic techniques for optimizing power generation plants with wind energy*. Springer Nature Singapore Pte Ltd.

Park, J. W., An, B. S., Lee, Y. S., Jung, H., & Lee, I. (2019). Wind farm layout optimization using genetic algorithm and its application to Daegwallyeong wind farm. *JMST Advances*, *1*(4), 249–257.

Pillai, A. C., Chick, J., Johanning, L., & Khorasanchi, M. (2018). Offshore wind farm layout optimization using particle swarm optimization. *Journal of Ocean Engineering and Marine Energy*, *4*(1), 73–88.

Routray, A., Mistry, K. D., & Arya, S. (2019). Analysis of wake effect on wind power generation for loss minimization in distribution system. In *2019 IEEE 1st international conference on energy, systems and information processing (ICESIP)* (pp. 1–6). IEEE.

Seyedali, M., Seyed Mohammad, M., & Andrew, L. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46. https://doi.org/10.1016/j.advengsoft.2013.12.007.

Sittichoke, P., & Weerakorn, O. (2013). Optimal placement of wind turbines within wind farm using binary particle swarm optimization with time-varying acceleration coefficients. *Renewable Energy*, *55*, 266. https://doi.org/10.1016/j.renene.2012.12.005.

Vasant, P. M., Rahman, I., Singh Mahinder Singh, B., & Abdullah-Al-Wadud, M. (2016). Optimal power allocation scheme for plug-in hybrid electric vehicles using swarm intelligence techniques. *Cogent Engineering*, *3*(1), 1203083.

Verma, S. K., Yadav, S., & Nagar, S. K. (2017). Optimization of fractional order PID controller using grey wolf optimizer. *Journal of Control, Automation and Electrical Systems*, *28*(3), 314–322.

Wen, L. (2016). Grey wolf optimizer based on nonlinear adjustment control parameter. In *International conference on sensors, mechatronics and automation* (pp. 643–648).

Wilson, D., Rodrigues, S., Segura, C., Loshchilov, I., Hutter, F., Buenfil, G. L., et al. (2018). Evolutionary computation for wind farm layout optimization. *Renewable Energy*, *126*, 681–691.

Wu, Y., Zhang, S., Wang, R., Wang, Y., & Feng, X. (2020). A design methodology for wind farm layout considering cable routing and economic benefit based on genetic algorithm and GeoSteiner. *Renewable Energy*, *146*, 687–698.

Yan, C., Chen, J., & Ma, Y. (2019). Grey Wolf optimization algorithm with improved convergence factor and position update strategy. In *2019 11th international conference on intelligent human–machine systems and cybernetics (IHMSC)* (Vol. 1, pp. 41–44). IEEE.

Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65–74).

Yang, K., Kwak, G., Cho, K., & Huh, J. (2019). Wind farm layout optimization for wake effect uniformity. *Energy*, *183*, 983–995.

Zhang, S., Luo, Q., & Zhou, Y. (2017). Hybrid grey wolf optimizer using elite opposition-based learning strategy and simplex method. *International Journal of Computational Intelligence and Applications*, *16*(02), 1750012.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.