Applied Soft Computing Journal 161 (2024) 111776

Contents lists available at ScienceDirect

Applied Soft Computing

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

Potential corrections to grey wolf optimizer

Hsing-Chih Tsai, Jun-Yang Shi *

Department of Civil and Environmental Engineering, National University of Kaohsiung, 700, Kaohsiung University Rd., Nanzih District, Kaohsiung 81148, Taiwan, Republic of China

HIGHLIGHTS

- This paper fixes GWO flaws via three key changes.
- Removal of the coefficient factor C improves algorithm performance.
- Eliminating the absolute sign for factor D rectifies potential biases.
- A novel strategy curbs greed in GWO by using current-to-prey search.
- The improved GWO excels at high-dimensional function optimization.

ARTICLE INFO

Keywords:
Grey wolf optimizer
Continuous optimization
Metaheuristics
Biased performance

ABSTRACT

Grey wolf optimizer (GWO), a well-known powerful algorithm that simulates the leadership hierarchy and hunting mechanisms of grey wolves in nature, has garnered significant attention from researchers recently. However, parts of GWO formulations have been shown to be unfit. Moreover, GWO generates outstanding results only for functions with optimal values of 0. Thus, in this paper, the inherent flaws of GWO are discussed and corrected variants are proposed to resolve its inherent flaws. The three corrections to the original GWO proposal made in this study include eliminating coefficient vector C, eliminating the absolute sign for factor D, and introducing a current-to-prey approach. Based on a numerical validation using CEC2005 and CEC2019 benchmark functions, one of the proposed corrected variants performs comparably with other popular optimization algorithms and handles high-dimensional functions exceptionally well. Numerical simulations have elucidated the efficacy of the suggested corrections in mitigating the inherent flaws present in the original GWO. The corrected variants of GWO proposed in this study may be useful in developing future GWO applications and other GWO variants.

1. Introduction

Bio-inspired metaheuristics, which draw inspiration from the behavior of biological organisms, have been applied extensively to address optimization problems that are complex, nonlinear, and non-differentiable. Considerable research interest has been focused in recent decades on further refining the ability of these metaheuristics to emulate biological processes and developing practical applications.

Bio-inspired algorithms replicate specific biological processes and behaviors [1], such as evolution, swarming, foraging, hunting, reproduction, chemotaxis, and mimicry. In general, the bio-inspired algorithms can be classified into six main groups: evolution-based algorithms [2–6], swarm-based algorithms [7–29], plant-based algorithms [30–34], ecology-based algorithms [35–39], physics-based algorithms [40–44], and human intelligence-based algorithms [45–50]. Notably, most of the 278 bio-inspired algorithms identified by Molina *et al.* [1] in 2020 were derivatives of classical methods. Each methodology possesses distinct advantages and disadvantages, and no universal optimization technique can effectively address all categories of practical challenges [51]. Fig. 1 shows the taxonomy of novel bio-inspired algorithms.

Grey wolf optimizer (GWO) is a widely known swarm-based algorithm that draws inspiration from the natural leadership hierarchy and hunting behaviors of grey wolves. GWO is highly customizable to various optimization problems due to its unique advantages of using minimal parameters and not requiring initial derivation information.

* Corresponding author.
E-mail address: jyshi@nuk.edu.tw (J.-Y. Shi).

https://doi.org/10.1016/j.asoc.2024.111776
Received 4 November 2023; Received in revised form 7 April 2024; Accepted 16 May 2024
Available online 21 May 2024

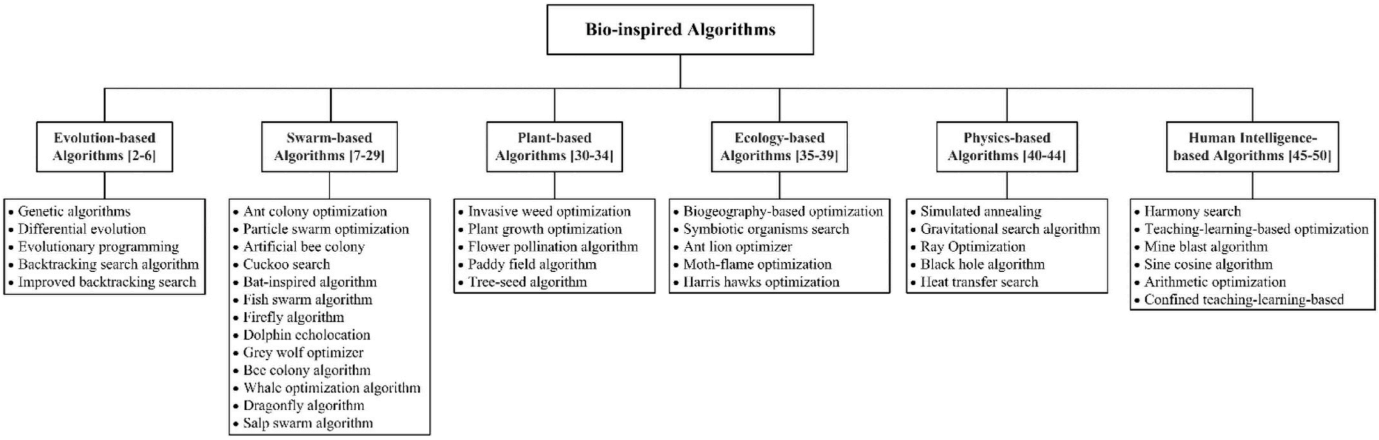

Fig. 1. Taxonomy of novel bio-inspired algorithms [1,52].

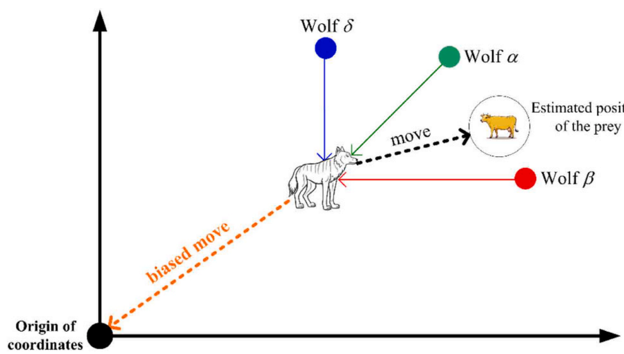

Fig. 2. Illustrations for position updating with search bias in GWO.

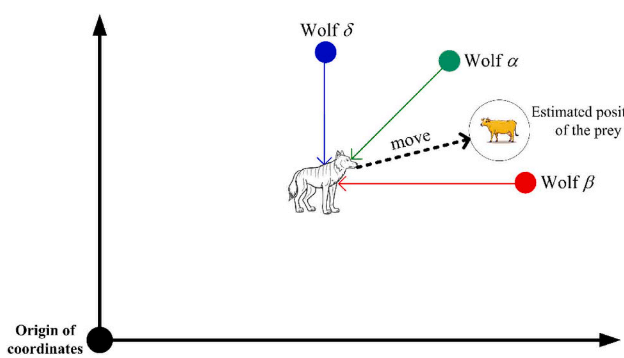

Fig. 3. Illustrations for position updating in the proposed corrected version I (GWO-C1).

Mirjalili *et al.* [20] further developed GWO in 2014 and demonstrated its great potential for solving constrained engineering design problems. Mirjalili [21] pioneered GWO applications for training multi-layer perceptrons. Their groundbreaking research involved comparing GWO comprehensively against several other optimization algorithms, including GA, particle swarm optimization (PSO), ant colony optimization, and evolution strategy, with the results demonstrating the superior performance of GWO. Gholizadeh [22] introduced a new variant of GWO specifically tailored to address the optimization challenge posed by double-layer grids with nonlinear characteristics, finding GWO outperformed alternative algorithms in terms of identifying the optimal design for these grids. Saremi *et al.* [23] integrated the original GWO with the evolutionary population dynamic, which is an evolutionary mechanism designed to eliminate weaker individuals from a population.

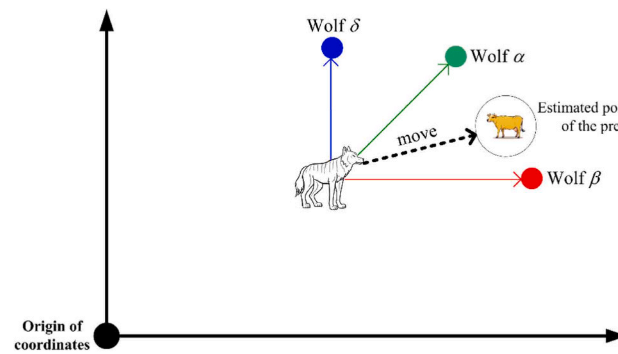

Fig. 4. Illustrations for position updating in the proposed corrected version II (GWO-C2).

This hybrid GWO demonstrated superior performance to the original GWO in terms of conversion rate, exploration, and avoiding local optima. Sulaiman *et al.* [24] applied GWO to solve an optimal reactive power dispatch problem, and compared the results to other optimization techniques, including SI, particle swarm optimization (PSO), evolutionary computation, the harmony search algorithm, gravity search algorithm, invasive weed optimization, and a modified imperialist competitive algorithm. Their findings showed GWO consistently produced superior optimal solutions to the other methods.

Recently, GWO has garnered considerable interest among researchers in various domains with regard to its unique ability to balance exploration and exploitation to achieve favorable convergence. Faris *et al.* [25] reported that GWO has been applied in global optimization, power engineering, bioinformatics, environmental studies, machine learning, networking, and image processing. Also, the findings of a comprehensive literature review by Sharma *et al.* [26] highlighted the need to modify or hybridize the original GWO algorithm to address diverse problem-solving challenges. Despite its popularity, Pan *et al.* [27] reported that, despite its reputation as a global search method, the original GWO often struggles to resolve high-dimensional problems with local optima. They subsequently modified the original GWO by incorporating two innovative search strategies, which significantly improved the ability of the algorithm to efficiently explore high-dimensional data for feature selection.

However, some flaws in GWO have been highlighted in more recent studies. Niu *et al.* [53] pointed out a significant flaw in the GWO algorithm: its performance progressively declines as the optimal solution of a function moves away from the 0. Moreover, Luo [54] confirmed that GWO has a significant search bias toward the origin of coordinates (the 0). Using numerical analysis, Tsai [55] demonstrated GWO often

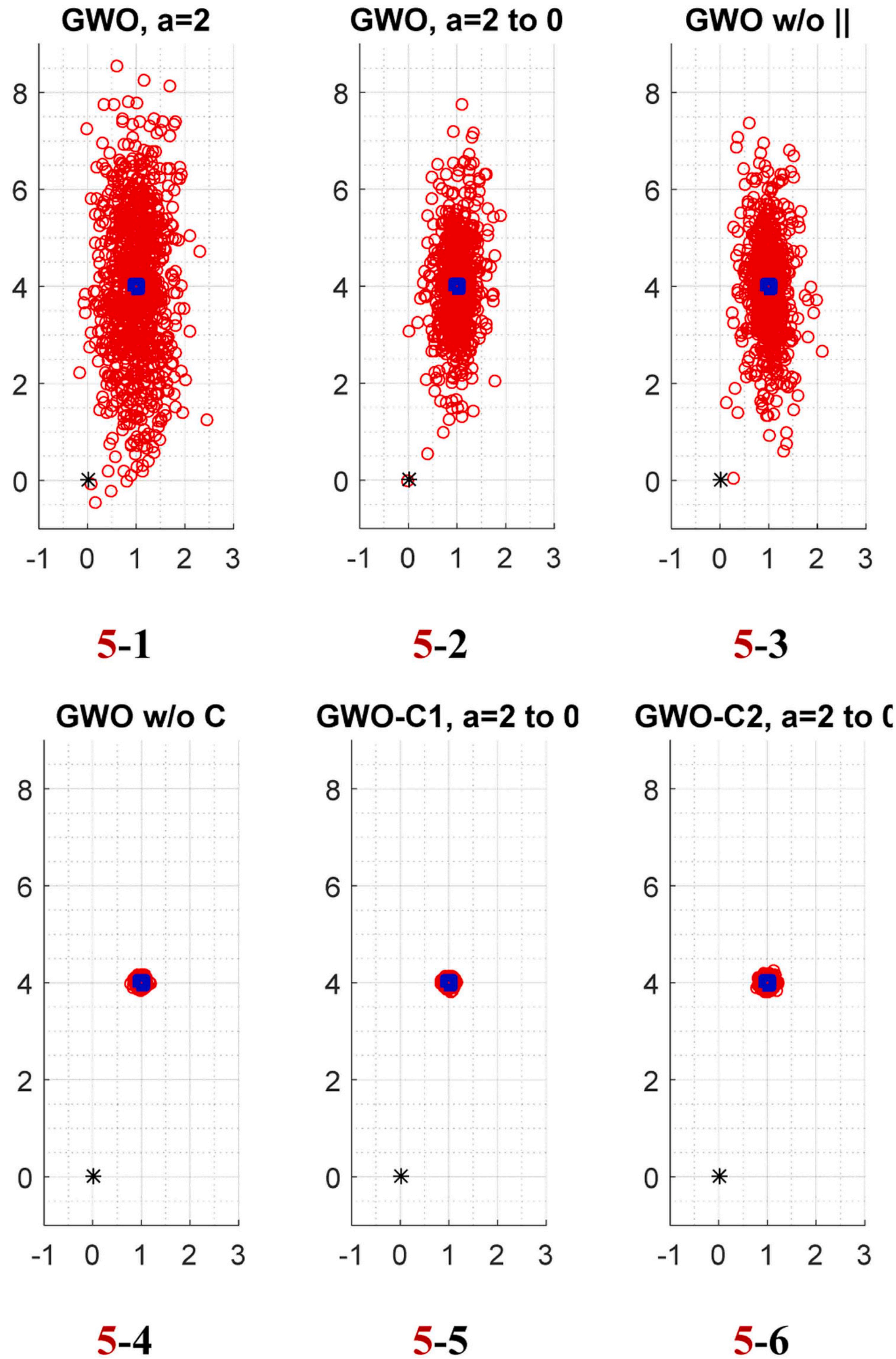

Fig. 5. When wolf agents reaching a high consensus.

prioritizes the origin of coordinates when generating solutions, which generates exceptional outcomes when applied to benchmark functions with optimal solutions at the origin. Likewise, Hu *et al.* [56] noted structural flaws and unstable performance in GWO, particularly when tackling complex problems like multimodality and hybrid functions. In addition, GWO exhibits limitations in terms of exploitation, as additional iterations do not necessarily yield improved solutions when trapped in a local optimum. Meidani *et al.* [57] also identified other limitations in GWO, including a lack of adaptive balance between exploration and exploitation. Thus, in light of the above research, the performance of GWO has been shown to fall short of the original claims.

Thus, although GWO and its variants have been shown to perform exceptionally well in various optimization problems, more recent studies have identified structural defects in the original GWO, including solution candidates generated close to the origin of coordinates and limited exploitation in the solution space. However, the underlying reasons for those defects have yet to be adequately investigated. Therefore, the novelty of this study lies in its quantitative analysis and subsequent corrections to rectify the defects of GWO. Fundamental questions to be solved in this study include:

Table 1
Unimodal functions.

No.	Name	Range	Optimum	Position	Function
F_1	Sphere	$[-100,100]^D$	0	$(0)^D$	$F_1(\vec{x}) = \sum_{i=1}^{D} x_i^2$
F_2	Schwefel2.22	$[-10,10]^D$	0	$(0)^D$	$F_2(\vec{x}) = \sum_{i=1}^{D} \lvert x_i\rvert + \prod_{i=1}^{D} \lvert x_i\rvert$
F_3	RotatedHyper-Ellipsoid	$[-65.536, 65.536]^D$	0	$(0)^D$	$F_3(\vec{x}) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$
F_4	Schwefel2.21	$[-100,100]^D$	0	$(0)^D$	$F_4(\vec{x}) = \max\{\lvert x_i\rvert, 1 \le i \le D\}$
F_5	Rosenbrock	$[-10,10]^D$	0	$(1)^D$	$F_5(\vec{x}) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
F_6	Step	$[-100,100]^D$	0	$(-0.5)^D$	$F_6(\vec{x}) = \sum_{i=1}^{D} (x_i + 0.5)^2$
F_7	QuarticWN	$[-1.28, 1.28]^D$	0	$(0)^D$	$F_7(\vec{x}) = \sum_{i=1}^{D} i \cdot x_i^4 + random[0,1]$

Table 2
Multimodal functions.

No.	Name	Range	Optimum	Position	Function
F_8	Schwefel2.26	$[-500,500]^D$	0	$(420.96)^D$	$F_8(\vec{x}) = 418.982887272433799807913601398D - \sum_{i=1}^{D} - x_i \sin(\sqrt{\lvert x_i\rvert})$
F_9	Rastrgin	$[-5.12, 5.12]^D$	0	$(0)^D$	$F_9(\vec{x}) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$
F_{10}	Ackley	$[-32,32]^D$	0	$(0)^D$	$F_{10}(\vec{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + \exp(1)$
F_{11}	Grewank	$[-600,600]^D$	0	$(0)^D$	$F_{11}(\vec{x}) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
F_{12}	Penalized1	$[-50,50]^D$	0	$(-1)^D$	$F_{12}(\vec{x}) = \frac{\pi}{D}\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_D - 1)^2\}$ $+ \sum_{i=1}^{D} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & , x_i > a \\ 0 & , -a \le x_i \le a \\ k(-x_i - a)^m & x_i < -a \end{cases}$
F_{13}	Penalized2	$[-50,50]^D$	0	$(1)^D$	$F_{13}(\vec{x}) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{D}(x_i - 1)^2[1 + \sin^2(3\pi x_i)]$ $+ (x_D - 1)^2[1 + \sin^2(2\pi x_D)]\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$

Table 3
Experimental results for shift functions.

	GWO		GWO'		GWO-C1		GWO-C1'		GWO-C2		GWO-C2'	
	FEs	*Suc*	*FEs*	*Suc*	*FEs*	*Suc*	*FEs*	*Suc*	*FEs*	*Suc*	*FEs*	*Suc*
F_1	3120	100	3000,000	0	3000,000	0	3000,000	0	94,042	100	93,096	100
F_2	3657	100	3000,000	0	3000,000	0	3000,000	0	127,734	99	184,115	97
F_3	10,932	100	3000,000	0	3000,000	0	3000,000	0	1218,183	74	1269,640	72
F_4	7196	100	3000,000	0	3000,000	0	3000,000	0	3000,000	0	3000,000	0
F_5	3000,000	0	3000,000	0	3000,000	0	3000,000	0	3000,000	0	3000,000	0
F_6	3000,000	0	3000,000	0	3000,000	0	3000,000	0	93,880	100	93,364	100
F_7	20,715	100	3000,000	0	3000,000	0	3000,000	0	1852,536	90	1870,466	88
F_8	3000,000	0	3000,000	0	3000,000	0	3000,000	0	3000,000	0	3000,000	0
F_9	6286	100	3000,000	0	3000,000	0	3000,000	0	2748,858	10	2847,739	6
F_{10}	3807	100	3000,000	0	3000,000	0	3000,000	0	175,303	98	205,828	97
F_{11}	4053	100	3000,000	0	3000,000	0	3000,000	0	2516,482	17	2481,564	18
F_{12}	2992,820	3	3000,000	0	3000,000	0	3000,000	0	96,742	100	155,369	98
F_{13}	193,116	94	3000,000	0	3000,000	0	3000,000	0	256,060	94	178,827	97
SUM	12,245,702	897	39,000,000	0	39,000,000	0	39,000,000	0	18,179,821	782	18,380,008	773

- Why do the potential biases exist in GWO?
- What types of appropriate corrections can be implemented to enhance GWO?

The remainder of this paper is structured as follows: Section 2 introduces the fundamental concepts of the GWO algorithm; Section 3 elucidates the inherent flaws of the original GWO and offers suggestions to enhance performance; Section 4 proposes two corrected variants of GWO to enhance its performance and discusses the results of our evaluation of biased algorithms; Section 5 compares the performance of GWO and the proposed variants; and Section 6 presents conclusions and suggests directions for future studies.

2. Grey Wolf Optimizer (GWO)

Mirjalili *et al.* [20] created the population-based GWO based on the social hierarchy of grey wolves. In GWO, four wolf types, namely alpha (α), beta (β), delta (δ), and omega (ω), are deployed to simulate this hierarchical structure, with alpha, beta, and delta, respectively representing the current first, second, and third best solutions as pseudo wolf agents. The last type, omega, are candidate solutions that follow the first three types in the search for global optimization.

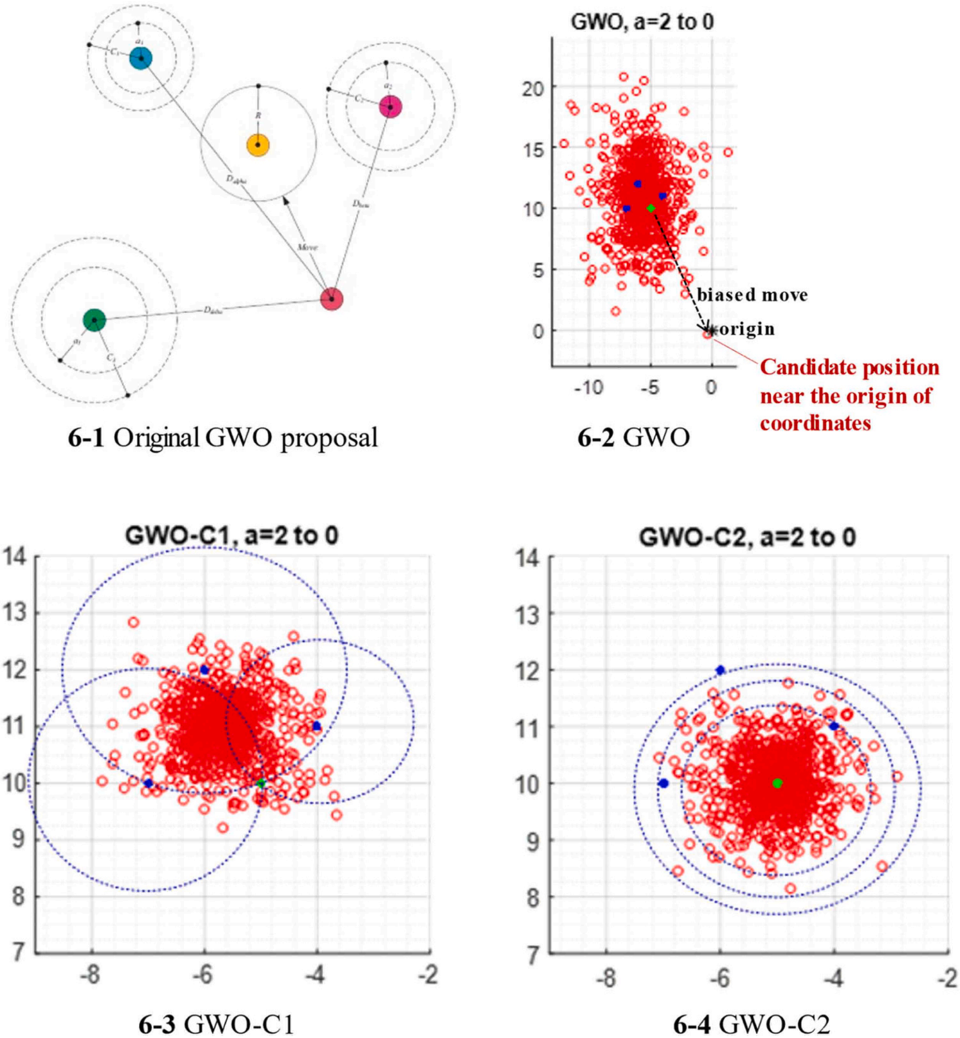

Fig. 6. Destinations updating by GWO and corrected versions.

2.1. Encircling prey

The following equations were suggested by GWO to simulate the encircling behavior of grey wolves during hunting:

$$\vec{D}(t) = \left|\vec{C}\cdot\vec{X}_p(t) - \vec{X}(t)\right|, \tag{1}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A}\cdot\vec{D}(t), \tag{2}$$

where t denotes the current iteration; $\vec{X}$ indicates the position vector of a grey wolf; $\vec{X}_p$ is the position vector of a prey animal; $\vec{D}$ is a vector formed primarily using the differences among $\vec{X}_p$ and $\vec{X}$; $\vec{A}$ and $\vec{C}$ are coefficient vectors that are respectively calculated as:

$$\vec{A} = 2a\cdot\vec{r}_1 - a, \tag{3}$$

$$\vec{C} = 2\vec{r}_2, \tag{4}$$

where a is a coefficient that decreases linearly from 2 to 0 over the course of iterations and the two random vectors, *i.e.*, $\vec{r}_1$ and $\vec{r}_2$, are generated in [0,1]. Therefore, $\vec{A}$ is varied from (−2,2) in the beginning iterations, (−1,1) midway through the iterations, and near 0 at the end of the iterations. $\vec{C}$ has values uniformly distributed in [0,2]. $\vec{A}$ and $\vec{C}$ play two important roles in achieving a good balance between exploitation and exploration. When the absolute value of $\vec{A}$ is greater than 1, GWO executes a divergent search that favors exploration. When the value of $\vec{C}$ is greater than 1, GWO emphasizes the effect of prey in defining the vector differences. $\vec{C}$ is associated with a random value at all times in order to emphasize exploration [58].

2.2. Hunting

In GWO, the "grey wolves" use the above equations to identify the location of prey and encircle them, with the hunt typically led by the α wolf, and the β and δ wolves occasionally assisting with hunting responsibilities. Assuming that the three leader wolves have better knowledge of prey positions, the top three current best solutions are treated as the three leader wolves. Eqs. (1–2) may be rewritten using α, β, and δ to replace the prey.

$$\vec{D}_\alpha = \left|\vec{C}_1\cdot\vec{X}_\alpha - \vec{X}\right|, \quad \vec{D}_\beta = \left|\vec{C}_2\cdot\vec{X}_\beta - \vec{X}\right|, \quad D_\delta = \left|\vec{C}_3\cdot\vec{X}_\delta - \vec{X}\right|, \tag{5}$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1\cdot\vec{D}_\alpha, \quad \vec{X}_2 = \vec{X}_\beta - \vec{A}_2\cdot\vec{D}_\beta, \quad \vec{X}_3 = \vec{X}_\delta - \vec{A}_3\cdot\vec{D}_\delta, \tag{6}$$

where the definitions of $\vec{C}_1$, $\vec{C}_2$, and $\vec{C}_3$ are the same as $\vec{C}$; the definitions of $\vec{A}_1$, $\vec{A}_2$, and $\vec{A}_3$ are similar to $\vec{A}$; and $\vec{X}_1$, $\vec{X}_2$, and $\vec{X}_3$ are transient vectors used to generate the next position for the wolf agent in the next iteration, *i.e.,*

Table 4
Summary of the 25 benchmark functions of CEC2005.

Function	D	Range	$f_{\min}$
F_1: Shifted Sphere Function	10,30,50	[−100, 100]	−450
F_2: Shifted Schwefel's Problem 1.2	10,30,50	[−100, 100]	−450
F_3: Shifted Rotated High Conditioned Elliptic Function	10,30,50	[−100, 100]	−450
F_4: Shifted Schwefel's Problem 1.2 with Noise in Fitness	10,30,50	[−100, 100]	−450
F_5: Schwefel's Problem 2.6 with Global Optimum on Bounds	10,30,50	[−100, 100]	−310
F_6: Shifted Rosenbrock's Function	10,30,50	[−100, 100]	390
F_7: Shifted Rotated Griewank's Function without Bounds	10,30,50	[−∞, ∞]	−180
F_8: Shifted Rotated Ackley's Function with Global Optimum on Bounds	10,30,50	[−32, 32]	−140
F_9: Shifted Rastrigin's Function	10,30,50	[−5, 5]	−330
F_{10}: Shifted Rotated Rastrigin's Function	10,30,50	[−5, 5]	−330
F_{11}: Shifted Rotated Weierstrass Function	10,30,50	[−0.5, 0.5]	90
F_{12}: Schwefel's Problem 2.13	10,30,50	[−π, π]	−460
F_{13}: Expanded Extended Griewank's plus Rosenbrock's Function (F8F2)	10,30,50	[−3, 1]	−130
F_{14}: Shifted Rotated Expanded Scaffer's F6	10,30,50	[−100, 100]	−300
F_{15}: Hybrid Composition Function	10,30,50	[−5, 5]	120
F_{16}: Rotated Hybrid Composition Function	10,30,50	[−5, 5]	120
F_{17}: Rotated Hybrid Composition Function with Noise in Fitness	10,30,50	[−5, 5]	120
F_{18}: Rotated Hybrid Composition Function	10,30,50	[−5, 5]	10
F_{19}: Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum	10,30,50	[−5, 5]	10
F_{20}: Rotated Hybrid Composition Function with the Global Optimum on the Bounds	10,30,50	[−5, 5]	10
F_{21}: Rotated Hybrid Composition Function	10,30,50	[−5, 5]	360
F_{22}: Rotated Hybrid Composition Function with High Condition Number Matrix	10,30,50	[−5, 5]	360
F_{23}: Non-Continuous Rotated Hybrid Composition Function	10,30,50	[−5, 5]	360
F_{24}: Rotated Hybrid Composition Function	10,30,50	[−5, 5]	260
F_{25}: Rotated Hybrid Composition Function without Bounds	10,30,50	[−∞,∞]	260

$$\vec{X}\left(t+1\right)=\frac{\vec{X}_1+\vec{X}_2+\vec{X}_3}{3}. \tag{7}$$

The three leader wolves in GWO are guided by three fundamental formulations, namely Eqs. (5–7), respectively. In other words, the α, β, and δ dominate the GWO search to update wolf positions around their prey.

3. Potential corrections and suggestions for GWO

Niu *et al.* [53] and Tsai [55] verified GWO is a biased algorithm that performs well only when solving problems with optimal targets of 0. The candidate position generated by GWO with search bias is given in Fig. 2. This point is clarified and suggestions for potential corrections to GWO are made in this paper. In the original GWO, the $\vec{C}$ vector is a significant parameter that controls the degree to which the effect of prey is emphasized, with priority given to this effect when the value of $\vec{C}$ approaches 2. Moreover, assigning random values for $|\vec{C}|$ increases the difficulty faced by wolves in approaching prey [20], with $|\vec{C}|$ values of 3 or larger resulting in extremely low degrees of emphasis. Assuming the absolute sign in Eq. (1) is eliminated, $|\vec{C}|$ is 2, and $|\vec{A}|$ is 1, Eq. (2) becomes:

$$\vec{X}\left(t+1\right)=\vec{X}_p(t)-\left[2\vec{X}_p(t)-\vec{X}(t)\right]=\vec{X}(t)-\vec{X}_p(t). \tag{8}$$

If wolf $\vec{X}$ is very close to prey $\vec{X}_p$, the position of the wolf approaches 0 in the next iteration and, in this situation, candidate positions near 0 are generated automatically as solutions. This reflects the effect of $\vec{C}$ rather than the situations claimed in the original proposal. Furthermore, in line with the vector calculation terminology, while $(\vec{X}_p-\vec{X})$ represents the difference between the two vectors, $(2\vec{X}_p-\vec{X})$ is not clear for any mathematical purposes concerning the two vectors. Therefore, this paper suggests withdrawing $\vec{C}$ from GWO.

The absolute sign in Eq. (1) was not discussed in the original GWO proposal because $\vec{A}$ is equally split between positive and negative values. Based on this, two potential corrected variants of GWO were proposed in this study.

3.1. Corrected version I

Eliminating the absolute sign and C from GWO resulted in the following GWO equations:

$$\vec{X}_1=\vec{X}_\alpha-\vec{A}_1\cdot\left(\vec{X}_\alpha-\vec{X}\right), \tag{9}$$

$$\vec{X}_2=\vec{X}_\beta-\vec{A}_2\cdot\left(\vec{X}_\beta-\vec{X}\right), \tag{10}$$

$$\vec{X}_3=\vec{X}_\delta-\vec{A}_3\cdot\left(\vec{X}_\delta-\vec{X}\right), \tag{11}$$

$$\vec{X}(t+1)=\frac{\vec{X}_1+\vec{X}_2+\vec{X}_3}{3}, \tag{12}$$

where Eqs. (9–11) are prey-to-current search strategies that use their base position at prey $\vec{X}_\alpha$ to evolve to current position $\vec{X}$. Although the search factors of prey-to-current strategies are usually confined to [0,2], the current value of $\vec{A}$ is in [−2,2]. However, the original $\vec{A}$ settings are acceptable for searching in the reverse direction to increase exploration capabilities. Although prey-to-current strategies have been explored only rarely in the literature [6], strategies using a base vector with random perturbation vectors and current-to-prey strategies are more common. Fig. 3 displays the candidate position generated by the corrected version I (GWO-C1).

3.2. Corrected version II

Both prey-to-current and current-to-prey scenarios are retrievable in the original GWO. Therefore, a current-to-prey version was also proposed in this study, *i.e.,*

$$\vec{X}_1=\vec{X}+\vec{A}_1\cdot\left(\vec{X}_\alpha-\vec{X}\right), \tag{13}$$

$$\vec{X}_2=\vec{X}+\vec{A}_2\cdot\left(\vec{X}_\beta-\vec{X}\right), \tag{14}$$

$$\vec{X}_3=\vec{X}+\vec{A}_3\cdot\left(\vec{X}_\delta-\vec{X}\right), \tag{15}$$

$$\vec{X}(t+1)=\frac{\vec{X}_1+\vec{X}_2+\vec{X}_3}{3}, \tag{16}$$

where Eqs. (13–15) have their base vectors at the current position of a wolf. The minus operator before the $\vec{A}$ parameter is changed to a plus operator. Because the destination of the strategy is the position of the prey when $|\vec{A}|=1$, the values of $\vec{A}$ is generally set in [0,2]. However, in this paper, the original settings of $|\vec{A}|$ in [−2,2] are retained to increase the exploration abilities of the algorithm. Fig. 4 shows the candidate position generated by the corrected version II (GWO-C2).

Table 5
Comparisons for GWO and corrected versions on CEC2005 (D=10).

Function	GWO			GWO-C1			GWO-C2	
	Mean	Std		Mean	Std		Mean	Std
F_1	22.400	49.945	$\prec$	17.876	75.325	$\prec$	4.40E-10	2.20E-09
F_2	454.573	795.486	$\succ$	1756.767	1260.918	$\prec$	0.317	1.051
F_3	6.23E+05	5.40E+05	$\succ$	5.70E+06	8.06E+06	$\approx$	5.53E+05	5.60E+05
F_4	808.686	1221.538	$\succ$	5809.341	5118.759	$\prec$	2.764	12.498
F_5	195.304	260.876	$\succ$	3221.202	1668.860	$\approx$	147.436	247.684
F_6	9.94E+04	2.12E+05	$\approx$	1.03E+05	4.16E+05	$\prec$	3.54E+02	1.39E+03
F_7	7.739	9.539	$\succ$	47.602	33.598	$\prec$	0.540	0.271
F_8	20.339	0.082	$\approx$	20.343	0.128	$\approx$	20.376	0.058
F_9	12.178	6.077	$\succ$	16.823	7.982	$\prec$	2.985	2.632
F_{10}	16.719	7.679	$\succ$	27.611	11.875	$\approx$	13.920	7.703
F_{11}	3.638	1.279	$\succ$	6.778	1.787	$\approx$	3.870	1.419
F_{12}	2802.282	2920.219	$\succ$	7757.866	5508.469	$\approx$	1453.580	2026.725
F_{13}	1.053	0.649	$\succ$	1.611	0.701	$\prec$	0.739	0.338
F_{14}	3.016	0.494	$\succ$	3.814	0.434	$\approx$	2.854	0.435
F_{15}	278.393	193.638	$\succ$	453.480	146.522	$\approx$	283.972	154.399
F_{16}	123.397	26.057	$\succ$	174.451	52.880	$\approx$	119.183	15.355
F_{17}	142.623	34.772	$\succ$	210.459	50.571	$\approx$	125.279	20.249
F_{18}	844.121	160.643	$\succ$	1037.174	43.940	$\approx$	770.117	272.623
F_{19}	853.755	127.761	$\succ$	1008.865	71.203	$\approx$	851.808	153.648
F_{20}	874.655	142.933	$\succ$	1026.105	59.914	$\approx$	823.097	200.488
F_{21}	921.455	301.213	$\succ$	1160.064	167.209	$\approx$	818.364	320.953
F_{22}	773.103	160.734	$\succ$	875.222	63.533	$\approx$	791.906	47.866
F_{23}	1095.446	185.898	$\succ$	1186.562	161.444	$\prec$	929.973	284.276
F_{24}	358.288	269.048	$\succ$	712.691	407.169	$\approx$	399.809	251.575
F_{25}	518.531	255.083	$\succ$	1027.955	342.253	$\prec$	386.896	23.960
Rank	1.84			2.92			1.24	
$\succ/\approx/\prec$	–			22/2/1			0/16/9	

Table 6
Comparisons for GWO and corrected versions on CEC2005 (D=30).

Function	GWO			GWO-C1			GWO-C2	
	Mean	Std		Mean	Std		Mean	Std
F_1	1579.533	1495.227	$\succ$	5333.120	2899.407	$\prec$	178.893	348.125
F_2	1.07E+04	3.77E+03	$\succ$	3.90E+04	1.07E+04	$\prec$	1.37E+03	2.04E+03
F_3	1.43E+07	1.07E+07	$\succ$	1.24E+08	6.90E+07	$\prec$	6.50E+06	3.29E+06
F_4	1.62E+04	5.42E+03	$\succ$	5.42E+04	1.37E+04	$\prec$	3.23E+03	4.80E+03
F_5	5.25E+03	3.58E+03	$\succ$	1.54E+04	2.38E+03	$\approx$	4.46E+03	1.10E+03
F_6	1.07E+08	3.33E+08	$\succ$	1.14E+09	1.24E+09	$\prec$	2.03E+07	7.95E+07
F_7	427.117	176.781	$\succ$	2535.404	535.930	$\prec$	0.680	0.352
F_8	20.950	0.036	$\approx$	20.889	0.113	$\approx$	20.957	0.048
F_9	95.930	24.554	$\succ$	145.723	30.228	$\prec$	19.395	9.812
F_{10}	140.633	61.264	$\succ$	236.493	60.676	$\approx$	122.848	60.923
F_{11}	18.033	3.120	$\succ$	33.061	2.864	$\succ$	24.421	5.261
F_{12}	7.48E+04	3.69E+04	$\succ$	2.94E+05	1.07E+05	$\prec$	3.23E+04	2.34E+04
F_{13}	5.721	2.643	$\succ$	21.296	9.719	$\prec$	3.941	2.885
F_{14}	11.901	0.572	$\succ$	13.362	0.429	$\succ$	12.220	0.551
F_{15}	448.122	100.717	$\succ$	622.381	116.659	$\prec$	306.391	128.205
F_{16}	305.794	181.921	$\succ$	450.900	115.192	$\approx$	229.727	122.161
F_{17}	294.584	174.612	$\succ$	538.247	151.311	$\approx$	260.294	123.358
F_{18}	948.264	20.209	$\succ$	1071.276	40.796	$\prec$	910.036	3.861
F_{19}	949.317	16.542	$\succ$	1049.800	38.255	$\prec$	909.108	2.416
F_{20}	943.634	20.861	$\succ$	1047.062	42.037	$\prec$	910.921	3.874
F_{21}	913.818	207.894	$\succ$	1207.371	55.484	$\prec$	562.423	137.497
F_{22}	970.716	36.949	$\succ$	1152.143	77.589	$\prec$	914.010	34.742
F_{23}	835.391	154.591	$\succ$	1177.174	98.382	$\prec$	550.190	51.055
F_{24}	865.451	363.578	$\succ$	1263.460	71.628	$\prec$	664.425	354.782
F_{25}	448.586	274.050	$\succ$	1296.543	33.884	$\prec$	258.104	190.780
Rank	1.92			2.92			1.16	
$\succ/\approx/\prec$	–			24/1/0			2/5/18	

4. Search destinations for GWO and the proposed corrected versions

The proposed corrected variants were respectively named GWO-C1 and GWO-C2. Rather than seeking to propose an "ultimate" version of GWO able to outperform most optimization algorithms, this paper aimed to investigate the fundamental issues of GWO and propose corrected variants with minimal changes. Prior to comparing performance, two tests should be conducted to verify whether a GWO version favors specific search targets or to demonstrate whether the obtained destination is suitable.

4.1. *Biased favors for specific search targets*

A scenario is designed that generates all agent information within a tiny region, with wolf agents in the next iteration assigned to regions

Table 7
Comparisons for GWO and corrected versions on CEC2005 (D=50).

Function	GWO			GWO-C1			GWO-C2	
	Mean	Std		Mean	Std		Mean	Std
F_1	6.00E+03	3.73E+03	≻	2.55E+04	1.01E+04	≺	1.05E+03	1.31E+03
F_2	2.85E+04	7.98E+03	≻	1.14E+05	2.75E+04	≺	8.12E+03	5.79E+03
F_3	7.59E+07	4.45E+07	≻	4.16E+08	1.88E+08	≺	1.99E+07	5.35E+06
F_4	3.65E+04	1.21E+04	≻	1.55E+05	3.71E+04	≺	1.94E+04	8.98E+03
F_5	1.21E+04	2.33E+03	≻	2.63E+04	3.41E+03	≺	9.37E+03	2.04E+03
F_6	3.47E+08	4.17E+08	≻	6.81E+09	3.86E+09	≺	1.19E+08	2.47E+08
F_7	900.711	205.417	≻	5702.719	975.307	≺	1.915	1.698
F_8	21.136	0.040	≺	21.006	0.099	≈	21.148	0.028
F_9	218.763	29.340	≻	353.301	44.694	≺	54.113	26.220
F_{10}	299.979	101.522	≻	604.393	111.917	≈	322.145	119.718
F_{11}	34.750	8.505	≻	58.787	4.248	≻	55.675	11.455
F_{12}	4.09E+05	1.54E+05	≻	1.38E+06	3.96E+05	≺	1.97E+05	9.63E+04
F_{13}	15.666	10.614	≻	94.536	45.661	≈	13.740	7.564
F_{14}	21.129	0.537	≻	23.203	0.453	≻	22.539	0.494
F_{15}	466.875	80.720	≻	729.736	87.222	≺	324.720	148.413
F_{16}	294.118	145.115	≻	470.786	118.250	≈	299.770	86.262
F_{17}	300.248	132.440	≻	722.115	122.475	≈	319.177	87.680
F_{18}	992.428	20.965	≻	1166.213	29.839	≺	936.400	11.273
F_{19}	987.930	22.288	≻	1163.966	51.726	≺	936.863	15.301
F_{20}	986.313	22.398	≻	1173.331	45.279	≺	936.070	9.487
F_{21}	969.260	163.630	≻	1287.254	34.872	≈	1015.561	3.742
F_{22}	1029.827	25.424	≻	1259.115	71.231	≺	943.003	29.063
F_{23}	990.141	136.712	≻	1281.896	33.569	≈	999.610	95.958
F_{24}	1213.301	30.566	≻	1342.703	39.020	≺	989.053	85.833
F_{25}	1107.142	279.518	≻	1374.435	39.700	≺	570.150	476.532
Rank	1.72			2.92			1.36	
≻/≈/≺	–			24/0/1			2/7/16	

Table 8
CEC2019 test functions.

Function	Name	D	Range	f_{min}
F_1	Storn's Chebyshev polynomial fitting problem	9	[−8192, 8192]	1
F_2	Inverse Hilbert matrix problem	16	[−16,384, 16384]	1
F_3	Lennard-Jones minimum energy cluster problem	18	[−4, 4]	1
F_4	Shifted and rotated Rastrigin's function	10	[−100, 100]	1
F_5	Shifted and rotated Griewank's function	10	[−100, 100]	1
F_6	Shifted and rotated Weierstrass function	10	[−100, 100]	1
F_7	Shifted and rotated Schwefel's function	10	[−100, 100]	1
F_8	Shifted and rotated expanded Schaffer's F_6 function	10	[−100, 100]	1
F_9	Shifted and rotated happy cat function	10	[−100, 100]	1
F_{10}	Shifted and rotated ackley function	10	[−100, 100]	1

Table 9
Parameter settings for all of the algorithms in CEC2005 and CEC2019 tests.

Algorithm	Parameter settings
GWO-C2	N=30, a=2–0 linearly
PSO	N=30, c_1=c_2=2.0, w=0.9–0.2 linearly
ABC	N_e=30, N_o=30, N_s=1, $limit$=$N\times D$
HTS	N=30, CDF=2, COF=10, RDF=2
MFO	N=30, N_{Flame}=30–1 linearly
SCA	N=30, r_1=2–0 linearly, r_2=[0,2π], r_3=[0,2], r_4=[0,1]
HHO	N=30, E_1=[−1,1], E_2=2–0 linearly
AOA	N=30, C_1=2, C_2=6, C_3=1, C_4=2, u=0.9, l=0.1

highly associated with the tiny region [55]. Using a two-dimensional problem as an example, after separately generating $\overrightarrow{X}$, $\overrightarrow{X}_\alpha$, $\overrightarrow{X}_\beta$, and $\overrightarrow{X}_\delta$ 1000 times in a square region centered on (1,4) with a length 0.1 (i.e., the blue square region in Fig. 5), 1000 corresponding $\overrightarrow{X}$ samples in the next iteration may be obtained (the red circles in Fig. 5) while the a value is available. Assuming a=2, most of the $\overrightarrow{X}$ samples should be obtained in the region centered on (1,4) and bounded by (0,0) (Fig. 5–1). In other words, point (0,0) is always a candidate solution for GWO. Considering that a linearly decreases from 2 to 0 over the course of 1000 samples, $\overrightarrow{X}$ samples increasingly converge on (1,4), with point (0,0) still a candidate solution for GWO (Fig. 5–2). Using the same setting as in Fig. 5–2, Fig. 5–3 considers a GWO without the absolute sign, while Fig. 5–4 uses a GWO without the vector $\overrightarrow{C}$. As previously discussed, no significant impacts of the absolute sign were found. To remove the vector $\overrightarrow{C}$ from GWO, the $\overrightarrow{X}$ samples are increasingly associated with the assigned region (i.e., the knowledge held by wolf agents). The $\overrightarrow{X}$ samples for the proposed GWO-C1 and GWO-C2 are presented, respectively, in Figs. 5–5 and 5–6, demonstrating that both of the corrected variants generate X samples that are highly associated with their assigned regions, regardless of region location.

As 0 is always included in the pool of GWO solutions, GWO has been previously shown to have an advantage in solving problems with optimal targets of 0. To verify this, 13 functions, including seven unimodal (Table 1) and six multimodal (Table 2), were employed in this study. Most of the 13 functions had global optimal values of 0, and a successful run was considered to be a near-optimal result within a gap of 10^{-3} to the global optimum in 300,000 function evaluations (*FEs*) [59, 60]. Forty individuals and 30 dimensional (D=30) functions were used, with each function executed 100 times. GWO is expected to solve problems with a global optimal value of 0 extremely fast. Shifting each of the 30 functions to the right-hand side with the length of the entire search space in each dimension and shifting the search space using the same number of functions are logically the same problem, although involving different spaces. Therefore, the shift F_1 has a global optimum of $(200)^D$ and a search range of $[100,300]^D$, with, for example, the shift F_2 solving $(20)^D$ in $[10,30]^D$. The experimental results are given in Table 3. The results of GWO' reflect those based on shift functions and the GWO analysis of the initial 13 functions. *FEs* is the average number

Table 10
Comparisons for GWO-C2 and other optimization algorithms on CEC2005 (D=10).

Function	GWO-C2 Mean	PSO	ABC	HTS	MFO	SCA	HHO	AOA
F_1	0.000	1599.488	0.000	0.000	88.097	1109.469	3.945	5128.928
F_2	0.317	1509.425	4.936	0.000	502.054	2702.737	6428.601	8453.716
F_3	5.53E+05	1.08E+07	5.30E+05	1.11E+05	3.41E+06	1.40E+07	5.63E+06	3.22E+07
F_4	2.76E+00	1.34E+03	9.70E+02	2.60E+01	4.22E+02	3.55E+03	1.48E+04	9.43E+03
F_5	1.47E+02	1.44E+03	7.01E+01	2.76E-10	3.31E+00	2.30E+03	7.79E+03	1.21E+04
F_6	3.54E+02	9.99E+07	1.11E+00	4.83E+01	1.03E+06	1.64E+07	6.76E+04	8.14E+08
F_7	0.540	508.995	0.289	5.009	1.118	150.910	52.212	432.605
F_8	20.376	20.375	20.331	20.358	20.252	20.384	20.266	20.244
F_9	2.985	46.877	0.000	0.955	29.046	47.548	33.762	24.853
F_{10}	13.920	60.169	31.136	20.638	31.749	68.719	63.815	52.826
F_{11}	3.870	7.157	5.690	5.093	6.126	9.085	9.130	8.443
F_{12}	1.45E+03	1.49E+04	5.10E+01	1.36E+03	6.39E+03	2.62E+04	1.66E+04	2.16E+04
F_{13}	0.739	5.817	0.200	0.635	1.339	4.902	4.236	2.401
F_{14}	2.854	3.599	3.281	3.047	4.021	3.982	4.087	3.494
F_{15}	283.972	606.078	0.001	245.862	422.015	627.752	567.666	429.902
F_{16}	119.183	249.398	149.699	150.959	173.415	265.722	305.595	281.624
F_{17}	125.279	275.171	160.296	166.195	183.212	301.390	363.720	284.446
F_{18}	770.117	986.367	598.807	828.779	848.937	1020.113	900.000	1118.979
F_{19}	851.808	999.479	537.947	859.446	892.893	968.308	900.000	1120.346
F_{20}	823.097	974.634	561.444	893.967	921.816	996.569	900.000	1130.394
F_{21}	818.364	1206.381	378.084	931.849	941.713	1215.358	1203.797	1307.475
F_{22}	791.906	924.142	711.081	826.243	836.270	919.484	1001.985	1067.055
F_{23}	929.973	1165.414	530.983	957.969	1002.721	1246.525	1253.595	1308.666
F_{24}	399.809	1069.576	200.000	854.181	726.334	878.888	1257.454	1292.702
F_{25}	386.896	947.580	291.893	503.730	386.373	926.131	1127.336	1362.149
Rank	2.36	6.00	1.76	2.56	3.92	6.56	6.12	6.72
$\succ/\approx/\prec$	–	23/2/0	7/4/14	9/10/6	18/5/2	24/1/0	21/3/1	24/0/1

Table 11
Comparisons for GWO-C2 and other optimization algorithms on CEC2005 (D=30).

Function	GWO-C2 Mean	PSO	ABC	HTS	MFO	SCA	HHO	AOA
F_1	1.79E+02	2.68E+04	1.05E-13	5.64E-04	5.66E+03	2.10E+04	3.85E+01	5.15E+04
F_2	1.37E+03	5.27E+04	3.68E+03	3.03E+02	2.25E+04	3.57E+04	3.37E+04	4.00E+04
F_3	6.50E+06	2.13E+08	5.85E+06	3.59E+06	4.26E+07	3.43E+08	5.78E+07	7.72E+08
F_4	3.23E+03	6.32E+04	3.47E+04	1.31E+04	6.24E+04	4.24E+04	6.23E+04	4.28E+04
F_5	4.46E+03	1.79E+04	1.09E+04	5.72E+03	1.09E+04	2.05E+04	2.62E+04	3.17E+04
F_6	2.03E+07	1.07E+10	2.52E+00	1.26E+07	2.73E+09	4.29E+09	1.89E+05	2.03E+10
F_7	0.680	6649.513	0.024	2.498	2.969	2618.811	149.639	2646.088
F_8	20.957	20.955	20.827	20.934	20.474	20.960	20.583	20.870
F_9	19.395	271.689	0.000	43.221	168.799	297.490	226.552	201.638
F_{10}	122.848	406.983	356.978	308.537	215.421	418.312	458.394	438.404
F_{11}	24.421	34.973	28.195	31.475	31.296	39.829	38.349	35.711
F_{12}	3.23E+04	6.66E+05	7.14E+03	4.06E+04	1.83E+05	6.29E+05	3.64E+05	7.71E+05
F_{13}	3.941	139.602	0.970	3.480	29.085	49.208	32.612	21.423
F_{14}	12.220	13.183	12.926	12.602	13.592	13.703	13.464	13.044
F_{15}	306.391	877.996	4.397	371.541	536.269	826.888	738.895	1005.300
F_{16}	229.727	641.491	292.871	430.908	287.116	486.785	508.725	917.228
F_{17}	260.294	585.332	357.874	448.927	321.962	563.021	595.045	996.888
F_{18}	910.036	1077.576	916.604	962.904	933.558	1086.308	900.000	911.415
F_{19}	909.108	1043.722	920.855	973.750	931.292	1090.017	900.000	900.000
F_{20}	910.921	1047.093	915.517	972.582	949.291	1085.984	900.000	900.000
F_{21}	562.423	1158.420	489.243	1240.240	1115.242	1246.756	1286.311	1335.444
F_{22}	914.010	1118.379	1072.973	1027.235	1000.639	1217.185	1309.941	1382.674
F_{23}	550.190	1161.549	531.872	1181.260	1118.860	1244.579	1308.308	1324.687
F_{24}	664.425	1007.110	275.699	1203.348	939.263	1256.761	1370.015	1386.205
F_{25}	258.104	1513.515	1284.636	1207.122	323.927	1437.124	1360.686	1416.416
Rank	2.16	6.32	2.4	3.44	3.76	6.52	5.08	6.24
$\succ/\approx/\prec$	–	24/1/0	13/1/10	18/5/2	23/1/1	24/1/0	19/0/6	22/0/3

of function evaluations and *Suc* is the number of runs used to reach the designed minima before the maximum 300,000 *FEs*. The last raw adds the above 13 values together, revealing GWO required an average of 11, 741,867 function evaluations to solve the 13 functions and reached the designed minima in 911 successful runs. GWO' shows the results of the shift functions, which used significantly more function evaluations and fewer successful runs than GWO. The results of GWO' are more suitable than those of GWO for evaluating the performance of the original GWO in practice, as the former does not favor solutions at the position of 0. Thus, the original GWO performed less outstandingly in handling the task to have 101 successful runs.

The first corrected variant does not favor solutions at 0. The results of GWO-C1 and GWO-C1' were the same, with both failing with 0 successful runs and equal performance unable to be fully confirmed. The original GWO suggested three leader agents only to guide the whole population, and its search always begins with the three leader agents, giving these agents the ability to reduce population diversity rapidly. It has been frequently argued that PSO converges too fast to be trapped

Table 12
Comparisons for GWO-C2 and other optimization algorithms on CEC2005 (D=50).

Function	GWO-C2 Mean	PSO	ABC	HTS	MFO	SCA	HHO	AOA
F_1	1.05E+03	7.69E+04	2.14E-13	4.51E+02	2.21E+04	5.89E+04	1.24E+02	1.08E+05
F_2	8.12E+03	1.43E+05	1.80E+04	2.22E+03	8.11E+04	9.26E+04	7.39E+04	1.12E+05
F_3	1.99E+07	1.18E+09	1.36E+07	1.28E+07	1.70E+08	1.16E+09	1.31E+08	3.48E+09
F_4	1.94E+04	1.82E+05	9.86E+04	3.87E+04	1.68E+05	1.25E+05	1.34E+05	1.17E+05
F_5	9.37E+03	2.88E+04	2.59E+04	1.26E+04	2.55E+04	3.08E+04	3.21E+04	3.60E+04
F_6	1.19E+08	3.74E+10	7.12E+00	6.38E+08	1.83E+10	1.52E+10	6.95E+05	4.07E+10
F_7	1.92E+00	1.37E+04	3.97E-03	6.01E+01	4.78E+01	6.95E+03	1.74E+02	4.89E+03
F_8	21.148	21.124	20.910	21.137	20.471	21.134	20.762	21.094
F_9	54.113	554.853	0.000	136.001	325.459	613.251	419.777	504.541
F_{10}	322.145	827.620	1024.307	785.263	519.288	873.231	912.857	916.336
F_{11}	55.675	64.520	57.126	60.921	59.724	72.620	72.181	66.884
F_{12}	1.97E+05	3.91E+06	3.88E+04	1.78E+05	1.21E+06	2.99E+06	2.22E+06	4.80E+06
F_{13}	13.740	998.478	1.794	8.022	167.202	187.406	69.088	53.963
F_{14}	22.539	22.966	22.718	22.274	23.310	23.412	23.164	22.760
F_{15}	324.720	961.424	0.170	382.389	610.183	909.261	778.998	1159.070
F_{16}	299.770	699.377	389.846	569.397	389.775	623.729	557.548	1033.043
F_{17}	319.177	730.960	461.557	537.679	493.423	670.874	672.208	1097.719
F_{18}	936.400	1136.276	970.167	980.983	1025.931	1199.126	900.000	900.000
F_{19}	936.863	1148.877	978.227	1025.654	1038.756	1213.171	900.000	900.000
F_{20}	936.070	1138.213	970.379	991.984	1010.853	1212.062	900.000	900.000
F_{21}	1015.561	1156.528	500.000	1343.627	1036.543	1322.223	1331.250	1413.816
F_{22}	943.003	1201.224	1160.243	1127.809	1010.913	1315.927	1385.412	1575.558
F_{23}	999.610	1192.523	539.123	1285.677	1043.614	1322.573	1361.204	1409.995
F_{24}	989.053	1139.958	1337.302	1365.441	1054.717	1377.813	1424.410	1429.667
F_{25}	570.150	1659.149	1391.809	1396.621	1276.156	1659.164	1424.023	1481.514
Rank	2.24	6.24	2.60	3.60	4.00	6.44	4.64	6.12
$\succ/\approx/\prec$	–	24/0/1	13/1/11	14/6/5	23/1/1	24/1/0	19/1/5	20/0/5

Table 13
Comparisons for GWO-C2 and other optimization algorithms on standard CEC2019.

Function	GWO-C2 Mean	PSO	ABC	HTS	MFO	SCA	HHO	AOA
F_1	6.10E+06	5.86E+07	1.39E+06	9.45E+04	7.94E+06	3.87E+07	0.00E+00	8.44E-16
F_2	855.839	6858.672	2908.812	155.664	741.197	9350.706	4.000	3.901
F_3	1.171	7.093	0.663	1.450	5.835	9.389	2.934	3.205
F_4	8.837	45.589	9.566	10.948	25.924	60.304	56.222	48.114
F_5	0.167	19.784	0.029	0.255	0.133	29.476	6.903	40.807
F_6	0.995	7.425	1.609	1.085	3.030	8.695	8.751	6.350
F_7	370.500	1119.309	368.239	783.890	978.287	1532.983	1316.291	1456.543
F_8	2.313	3.486	2.394	2.550	3.341	3.674	3.768	3.499
F_9	0.179	1.045	0.130	0.315	0.323	1.514	0.411	1.238
F_{10}	20.393	20.353	18.948	20.180	20.134	20.407	20.126	20.080
Rank	2.90	6.10	2.20	3.20	4.20	7.60	4.90	4.90
$\succ/\approx/\prec$	–	9/1/0	2/4/4	6/2/2	7/1/2	9/1/0	6/1/3	6/1/3

Table 14
Comparisons for GWO-C2 and other optimization algorithms on shift CEC2019.

Function	GWO-C2 Mean	PSO	ABC	HTS	MFO	SCA	HHO	AOA
F_1’	6.38E+06	6.20E+07	1.46E+06	3.29E+04	6.65E+06	1.13E+09	3.34E+06	2.97E+06
F_2’	7.52E+02	8.63E+03	3.02E+03	1.07E+02	9.39E+02	2.65E+04	4.04E+02	1.38E+03
F_3’	1.704	7.113	0.708	1.806	5.231	10.665	4.464	0.345
F_4’	9.769	45.582	9.915	19.264	25.826	143.593	59.915	60.629
F_5’	0.169	17.389	0.028	0.488	0.142	195.331	21.740	41.994
F_6’	0.798	7.780	1.635	1.092	3.322	13.149	10.017	6.639
F_7’	470.734	1115.011	370.125	1018.147	1041.085	2223.509	1502.143	1697.372
F_8’	2.193	3.465	2.499	2.618	3.487	4.325	3.955	3.650
F_9’	0.159	0.975	0.120	0.244	0.315	4.578	0.731	1.455
F_{10}’	20.390	20.363	19.987	20.179	20.158	20.721	20.319	20.208
Rank	2.80	5.80	2.10	2.70	4.10	8.00	5.30	5.20
$\succ/\approx/\prec$	–	9/1/0	3/2/5	6/2/2	7/2/1	9/1/0	7/1/2	7/1/2

into the local optimum. However, PSO has one global guidance and one personal guidance for each individual, with the number of guidance greater than three. Another observation was that GWO’ had 101 runs and both GWO-C1 and GWO-C1’ failed completely due to GWO’ searching wider regions than the first corrected variant. Sometimes, using wider search regions enhances the exploration ability of GWO’.

Results of GWO-C2 and GWO-C2’ were quite similar to each other on each function. Therefore, the second corrected variant is not recommended for solving problems with an optimum of 0 and is able to partly handle the designed problem. Based on this finding, using current-to-prey formulations is recommended to fundamentally change the GWO proposal.

4.2. Destinations guided by the three prey

The picture shown in Fig. 6-1 was included in the original GWO proposal [20]. Each wolf agent is respectively guided by the three prey toward a central region. Assuming the three prey are at (−7,10), (−6, 12), and (−4,11; *i.e.*, the blue dots in the plots) and current $\overrightarrow{X}$=(−5,10; *i. e.*, the green dot in the plots), 1000 destinations samples (i.e., the red circles in the plots) are generated that progressively and linearly decrease with sample number from 2 to 0. The destination results for GWO are given in Fig. 6–2. Similarly, (0,0) was always on the edge of destination zone for the GWO results.

Under the same scenario for GWO-C1 and GWO-C2, the destination results for GWO-C1 are given in Fig. 6–3, which are more congruent than the idea in the original GWO proposal. The destination results comprise three circle searches centered on the prey. The three circle searches are shown as the dash-line circles in Fig. 6–3 **and** are only for $|\overrightarrow{A}|$ in [−1,1]. When $|\overrightarrow{A}|$ is in [−2,2], the circle searches have twice the number of radiuses than in Fig. 6–3. The destination results for GWO-C2, which follows the current-to-prey formula, are given in Fig. 6–4. Rather than [0,2], GWO-C2 adopts search factor $|\overrightarrow{A}|$ in [−2,2] in compliance with the suggestions in the original GWO proposal. Therefore, the three circle searches of GWO-C2 are centered on the current wolf rather than the position of the prey. However, GWO-C2 retains base vectors at the current wolf to prevent greedy search strategies. Notably, the dash-lined circles considering $|\overrightarrow{A}|$ in [−2,2] may be twice the size in terms of radius than those shown in Fig. 6–4. Consequently, GWO does not perform as described in the original proposal. Therefore, the proposed GWO-C1 and GWO-C2 may be considered as potentially superior alternatives.

5. Performance comparisons using CEC2005 benchmark functions

Rather than proposing an "ultimate" version of GWO able to outperform most optimization algorithms, this paper aimed to develop alternatives to GWO that adhere to the concept of the original GWO. Although both GWO-C1 and GWO-C2 were verified to assign agents correctly to suitable positions, their optimization performances still needed to be quantified. Twenty-five benchmark functions of CEC2005 were used to investigate the respective performances of these two new GWO versions. The CEC2005 benchmark suite contains 25 test functions involving 5 unimodal functions, 7 multimodal functions, 2 expanded functions, and 11 hybrid composition functions. Each test function uses 30 search agents for 10,000×D function evaluations, as suggested by Suganthan *et al.* [61]. The obtained results were based on 25 independent trials with random initial populations. In Table 4, the Function column shows the objective fitness function, the D column shows the dimensionality of each function, the Range column shows the boundaries of the search space, and f_{min} shows the optimal value. Because F_7 and F_{25} search the entire D-dimensional real-number space, their initializations are accomplished in [0600] and [−2, 5], respectively, while the remaining 23 functions adopt their search ranges for their initialization as well.

5.1. Comparative performance of GWO vs. corrected versions

Twenty-five benchmark functions of CEC2005 were used to investigate the respective performances of the suggested GWO versions. The average function error values (Mean) and their standard deviation (Std) are given in Tables 5, 6, and 7, respectively, for D settings of 10, 30, and 50. The row labelled *Rank* represents the average ranking considering the rank over 25 functions. Wilcoxon's rank sum tests at a 0.05 significance level were conducted to assess the comparative performance of different pairs of algorithms, as suggested by Derrac *et al.* [62], Cui *et al.* [63], and Beiranvand *et al.* [64]. '$\succ$','$\approx$'and'$\prec$' indicate, respectively, that the compared version is significantly worse than, almost the same as, or significantly better than the leftmost algorithm [6].

As shown in Tables 5 – 7, GWO-C1 did not outperform GWO over the low- and high-dimensional benchmark functions on the CEC2015 tests for most functions, with GWO-C1 frequently being identified as the inferior algorithm with a *Rank* of 2.92, and the worst *Rank* value being 3.0. Although GWO-C1 sends agents to suitable positions, its search strategies are too greedy to send wolves to the positions of the prey quickly. Also, GWO always searches a wide region with edges around 0, which is an unfavorable condition that reduces convergence speeds and increases GWO's exploration abilities. By contrast, GWO-C2 frequently outperformed GWO. Although the length of search regions for GWO-C2 and GWO-C1 were similar, the current-to-prey search strategies of GWO-C2 were not greedy because wolf agents must evolve to reach the position of the prey. Therefore, GWO-C2 is recommended as a potential alternative for GWO that removes the absolute sign of $\overrightarrow{D}$ and eliminates $\overrightarrow{C}$ as well as treats strategy forms as current-to-prey modes.

5.2. Performance of GWO-C2 vs. other optimization algorithms

In this section, the CEC2005 and CEC2019 suites were used to compare the performance of the proposed GWO-C2 against other optimization algorithms, including PSO [65], artificial bee colony (ABC) [66], heat transfer search (HTS) [47], moth-flame optimization (MFO) [41], Harris hawks optimization (HHO) [42], sine cosine (SCA) [51], and arithmetic optimization (AOA) [52]. The benchmark test problems (named F_1 to F_{10}) in CEC2019 are summarized in Table 8, known as "The 100-digit challenge." The benchmarks F_4 to F_{10} are rotated and shifted, whereas the F_1 to F_3 are minimization problems of different dimensions and ranges. The benchmark functions F_4 to F_{10} also have minimization problems with 10 dimensions in the range [−100,100]. All benchmarks in CEC2019 have a global optimum at 1.

Major parameter settings for the above algorithms are listed in Table 9, and the experimental results for the 25 CEC2005 functions are given in Tables 10 – 12. The test results for the CEC2019 functions are listed in Table 13. The CEC2019 test results were obtained after 30 runs utilizing 50 searching agents over a 5000 ×D maximum number of function evaluations as per the suggestions from Price *et al.* [67]. Although no single algorithm achieved optimal results for all applications [51], most of the results presented in Tables 10 – 13 support GWO-C2 and ABC as the best algorithms of those listed. GWO-C2 performed especially well in terms of handling high-dimensional functions. ABC is a three-phase algorithm with onlooker and scout phases designed, respectively, to enhance exploitation and exploration abilities. Different from ABC, GWO-C2 evaluates each wolf agent once in one iteration to strike a good balance between exploitation and exploration. Although the latter five algorithms were proposed relatively recently (within the last decade), their performances were disappointingly similar to that of the original GWO, and significant additional effort will be necessary to improve them. Corrections such as those made in this paper represent the first step toward making such improvements.

In addition, in case the global optimum in CEC2019 is much closer to the origin of coordinates, Table 14 further displays the results by moving all CEC2019 functions to the right side of the search space, with the length of the entire search space in each dimension and shifting the search space by the same number of functions. Table 13 shows that HHO and AOA performed best in finding the global optimum for the function F_1 in CEC2019. However, after moving the optimum far away from the origin of the coordinates, the two algorithms exhibit similar performance levels compared to other algorithms, as indicated in Table 14. In contrast, GWO-C2, PSO, and ABC demonstrate invariant performance when solving standard and shift benchmark problems. Although CEC functions are responsible for creating a fair comparison platform, the test results show that few benchmark functions may exhibit advantageous characteristics for specific optimization algorithms. Hence, the joint test outcomes presented in Tables 13 and 14 offer a more robust means of evaluating the efficacy of the corrected GWO compared to other algorithms.

6. Conclusions and recommendations

The strong performance of GWO and its variants on various optimization problems has been showcased in many studies. This paper further clarified the underlying causes of potential biases in GWO. Subsequently, two corrected variants are proposed to enhance the performance of GWO. The performance of the corrected variants has also been validated through numerical experiments using CEC2005 and CEC2019 benchmark functions.

Based on the numerical results, GWO performed exceptionally well on functions with optimal values of 0. The underlying reasons for the potential biases present in GWO can be outlined as follows.

- The absolute sign of the factor D is invalid.
- Effects of the coefficient vector $\overrightarrow{C}$ just assign wolf agents to positions along the direction toward the origin of the coordinates.
- GWO's performance is found to be subpar due to the excessive greediness of its prey-to-current approach.

This paper proposes three corrections to address the potential biases in GWO, as outlined below. Note that there is no interdependence among the three corrections.

- Eliminate the absolute sign for factor D to keep the algorithm simple.
- Remove the vector $\overrightarrow{C}$ to rectify the search bias.
- Use a current-to-prey search strategy to mitigate excessive greediness in GWO.

The proposed GWO-C2 integrated all three proposed corrections as a potential alternative to GWO. Based on numerical validation using CEC2005 benchmark functions, GWO-C2 proved comparable to other optimization algorithms and performed better than these other algorithms in dealing with high-dimensional functions. Nevertheless, the proposed variant of GWO did not exhibit a significantly superior performance compared to other well-known metaheuristic algorithms such as ABC. The GWO-C2 not only preserves the fundamental concepts of the original GWO but also integrates necessary corrections to improve its performance. In forthcoming research endeavors, there is potential to hybridize GWO-C2 with other optimization algorithms to significantly enhance its functionality in addressing intricate engineering challenges such as multimodal multiobjective optimization problems.

CRediT authorship contribution statement

Jun-Yang Shi: Investigation, Visualization, Writing – original draft, Writing – review & editing, Validation. **Hsing-Chih Tsai:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The study was financially supported by the National Science and Technology Council, Taiwan, under grant NSTC 112-2221-E-390-008.

Code availability

The source code of the corrected GWO variants can be found at the link below.

https://www.mathworks.com/matlabcentral/fileexchange/162131-corrected-grey-wolf-optimizer-for-continuous-optimization

References

[1] D. Molina, J. Poyatos, J.D. Ser, S. García, A. Hussain, F. Herrera, Comprehensive taxonomies of nature- and bio-inspired optimization: inspiration versus algorithmic behavior, critical analysis recommendations, Cogn. Comput. 12 (5) (2020) 897–939.
[2] J.H. Holland, Genetic algorithms, Sci. Am. 267 (1) (1992) 66–73.
[3] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (1997) 341–359.
[4] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. Evol. Comput. 3 (2) (1999) 82–102.
[5] P. Civicioglu, Backtracking search optimization algorithm for numerical optimization problems, Appl. Math. Comput. 219 (15) (2013) 8121–8144.
[6] H.C. Tsai, Improving backtracking search algorithm with variable search strategies for continuous optimization, Appl. Soft Comput. 80 (2019) 567–578.
[7] X.S. Yang, R. Xiao, M. Karamanoglu, Z. Cui, A.H. Gandomi (Eds.), Swarm Intelligence and Bio-Inspired Computation: Theory and Applications. Elsevier, 2013.
[8] J. Kennedy, The behavior of particles. in Proceedings of the International Conference on Evolutionary Programming, Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.
[9] H.C. Tsai, Y.H. Lin, Modification of the fish swarm algorithm with particle swarm optimization formulation and communication behavior, Appl. Soft Comput. 11 (8) (2011) 5367–5374.
[10] X.S. Yang, A New Metaheuristic Bat-Inspired Algorithm, Springer, 2010, pp. 65–74.
[11] I. Bojic, V. Podobnik, I. Ljubi, G. Jezic, M. Kusek, A self-optimizing mobile network: auto-tuning the network with firefly-synchronized agents, Inform. Sci. 182 (1) (2012) 77–92.
[12] M. Dorigo, V. Maniezzo, A. Colorni, The ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern. B 26 (1) (1996) 29–41.
[13] M. Dorigo, M. Birattari, Thomas Stützle, Ant colony optimization, IEEE Comput. Intell. Mag. 1 (4) (2006) 28–39.
[14] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Glob. Optim. 39 (3) (2007) 459–471.
[15] H.C. Tsai, Integrating the artificial bee colony and bees algorithm to face constrained optimization problems, Inform. Sci. 258 (2014) 80–93.
[16] R. Murugan, M.R. Mohan, C.C. Asir Rajan, P.D. Sundari, S. Arunachalam, Hybridizing bat algorithm with artificial bee colony for combined heat and power economic dispatch, Appl. Soft Comput. 72 (2018) 189–217.
[17] C.M. Rahman, T.A. Rashid, A. Alsadoon, N. Bacanin, P. Fattah, S. Mirjalili, A survey on dragonfly algorithm and its applications in engineering, Evol. Intell. 16 (1) (2023) 1–21.
[18] Y. Meraihi, A. Ramdane-Cherif, D. Acheli, M. Mahseur, Dragonfly algorithm: a comprehensive review and applications, Neural Comput. Appl. 32 (2020) 16625–16646.
[19] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[20] S. Mirjalili, How effective is the grey wolf optimizer in training multi-layer perceptron, Appl. Intell. 43 (1) (2015) 150–161.
[21] S. Gholizadeh, Optimal design of double layer grids considering nonlinear behaviour by sequential grey wolf algorithm, J. Optim. Civ. Eng. 5 (4) (2015) 511–523.
[22] S. Saremi, S.Z. Mirjalili, S.M. Mirjalili, Evolutionary population dynamics and grey wolf optimizer, Neural Comput. Appl. 26 (5) (2015) 1257–1263.
[23] M.H. Sulaiman, Z. Mustaffa, M.R. Mohamed, O. Aliman, Using the grey wolf optimizer for solving optimal reactive power dispatch problem, Appl. Soft Comput. 32 (2015) 286–292.
[24] H. Faris, I. Aljarah, M.A. Al-Betar, S. Mirjalili, Grey wolf optimizer: a review of recent variants and applications, Neural Comput. Appl. 30 (2018) 413–435.
[25] I. Sharma, V. Kumar, S. Sharma, A comprehensive survey on grey wolf optimization, Recent Adv. Comput. Sci. Commun. 15 (3) (2022) 323–333.
[26] H. Pan, S. Chen, H. Xiong, A high-dimensional feature selection method based on modified Gray Wolf Optimization, Appl. Softw. Comput. 135 (2023) 110031.

[27] A. Kaveh, N. Farhoudi, A new optimization method: Dolphin echolocation, Adv. Eng. Softw. 59 (2013) 53–70.
[28] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[29] X.S. Yang, S. Deb, Cuckoo search via Lévy flights. 2009 World congress on nature & biologically inspired computing (NaBIC), IEEE, 2009, pp. 210–214.
[30] A.R. Mehrabian, C. Lucas, A novel numerical optimization algorithm inspired from weed colonization, Ecol. Inform. 1 (4) (2006) 355–366.
[31] W. Cai, W. Yang, X. Chen, A global optimization algorithm based on plant growth theory: Plant growth optimization, in: 2008 International conference on intelligent computation technology and automation (ICICTA), Vol. 1, IEEE, 2008, pp. 1194–1199.
[32] U. Premaratne, J. Samarabandu, T. Sidhu, A new biologically inspired optimization algorithm. 2009 international conference on industrial and information systems (ICIIS), IEEE, 2009, pp. 279–284.
[33] X.S. Yang, Flower pollination algorithm for global optimization. In International conference on unconventional computing and natural computation, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 240–249.
[34] M.S. Kiran, TSA: Tree-seed algorithm for continuous optimization, Expert Syst. Appl. 42 (19) (2015) 6686–6698.
[35] D. Simon, Biogeography-based optimization, IEEE Trans. Evolut. Comput. 12 (6) (2008) 702–713.
[36] M.Y. Cheng, D. Prayogo, Symbiotic organisms search: a new metaheuristic optimization algorithm, Comput. Struct. 139 (2014) 98–112.
[37] S. Mirjalili, The ant lion optimizer, Adv. Eng. Softw. 83 (2015) 80–98.
[38] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowl. -Based Syst. 89 (2015) 228–249.
[39] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872.
[40] S. Kirkpatrick, C.D. Gelatt Jr, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[41] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.
[42] A. Kaveh, M. Khayatazad, A new meta-heuristic method: ray optimization, Comput. Struct. 112 (2012) 283–294.
[43] A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering, Inf. Sci. 222 (2013) 175–184.
[44] V.K. Patel, V.J. Savsani, Heat transfer search (HTS): a novel optimization algorithm, Inform. Sci. 324 (2015) 217–246.
[45] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2) (2001) 60–68.
[46] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, Comput. -Aided Des. 43 (3) (2011) 303–315.
[47] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm for optimization of truss structures with discrete variables, Comput. Struct. 102 (2012) 49–63.
[48] S. Mirjalili, SCA: A Sine Cosine Algorithm for solving optimization problems, Knowl. -Based Syst. 96 (2016) 120–133.
[49] L. Abualigah, A. Diabat, S. Mirjalili, M.A. Elaziz, A.H. Gandomi, The arithmetic optimization algorithm, Comput. Methods Appl. Mech. Eng. 376 (1) () (2021) 113609.
[50] H.C. Tsai, Confined teaching-learning-based optimization with variable search strategies for continuous optimization, Info Sci. 500 (2019) 34–47.
[51] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67–82.
[52] H.M. Dubey, M. Pandit, B.K. Panigrahi, An overview and comparative analysis of recent bio-inspired optimization techniques for wind integrated multi-objective power dispatch, Swarm Evol. Comput. 38 (2018) 12–34.
[53] P. Niu, S. Niu, L. Chang, The defect of the Grey Wolf optimization algorithm and its verification method, Knowl. -Based Syst. 171 (2019) 37–43.
[54] K. Luo, Enhanced grey wolf optimizer with a model for dynamically estimating the location of the prey, Appl. Softw. Comput. 77 (2019) 225–235.
[55] H.C. Tsai, Potential bias when creating a differential-vector movement algorithm, Appl. Softw. Comput. 113 (2021) 107925.
[56] J. Hu, H. Chen, A.A. Heidari, M. Wang, X. Zhang, Y. Chen, Z. Pan, Orthogonal learning covariance matrix for defects of grey wolf optimizer: insights, balance, diversity, and feature selection, Knowl. -Based Syst. 213 (2021) 106684.
[57] K. Meidani, A. Hemmasian, S. Mirjalili, A. Barati Farimani, Adaptive grey wolf optimizer, Neural Comput. Appl. 34 (10) (2022) 7711–7731.
[58] S. Mirjalili, S. Saremi, S.M. Mirjalili, L.S. Coelho, Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization, Expert Syst. Appl. 47 (2016) 106–119.
[59] T. Weise, R. Chiong, K. Tang, J. Lässig, S. Tsutsui, W. Chen, Z. Michalewicz, X. Yao, Benchmarking optimization algorithms: an open source framework for the traveling salesman problem, IEEE Comput. Intell. Mag. 9 (2014) 40–52.
[60] N. Hansen, A. Auger, D. Brockhoff, D. Tušar, T. Tušar, arXiv preprint, arXiv: 1605.03560 (. COCO: performance assessment, CoRR, 2016.
[61] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.P. Chen, A. Auger, S. Tiwari, Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization, Technical Report, Nanyang Technological University, Singapore, 2005.
[62] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.
[63] L. Cui, G. Li, Y. Luo, F. Chen, Z. Ming, N. Lu, J. Lu, An enhanced artificial bee colony algorithm with dual-population framework, Swarm Evol. Comput. 43 (2018) 184–206.
[64] V. Beiranvand, W. Hare, Y. Lucet, Best practices for comparing optimization algorithms, Optim. Eng. 18 (4) (2017) 815–848.
[65] H.C. Tsai, Y.Y. Tyan, Y.W. Wu, Y.H. Lin, Isolated particle swarm optimization with particle migration and global best adoption, Eng. Optim. 44 (12) (2012) 1405–1424.
[66] H.C. Tsai, Artificial bee colony directive for continuous optimization, Appl. Soft Comput. 87 (2020) 105982.
[67] K.V. Price, N.H. Awad, M.Z. Ali, P.N. Suganthan, Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization. In Technical report, Nanyang Technological University, Singapore, 2018.