

A Hybrid Grey Wolf-Bat Algorithm for Global Optimization

Mohammed ElGayyar¹ , E. Emary², N. H. Sweilam¹ ,
and M. Abdelazeem³

¹ Department of Mathematics, Faculty of Science, Cairo University, Giza, Egypt
{melgayar, nsweilam}@sci.cu.edu.eg

² Faculty of Computers and Information, Cairo University, Giza, Egypt
eidemary@yahoo.com

³ National Research Institute of Astronomy and Geophysics (NRIAG),
Helwan, Egypt
maazeem03@yahoo.com

Abstract. Bio-inspired algorithms are now becoming powerful methods for solving many real-world optimization problems. In this paper, we propose a hybrid approach involving Grey Wolf optimizer (GWO) and Bat swarm optimizer (BA) for global function optimization problems. GWO is well known for its balanced exploration/exploitation behavior, while BA is known to be more exploitative due to its low exploration ability in some conditions. We use GWO exploration skills to explore the search space effectively and BA local search capabilities to refine the solution. In our hybrid algorithm, namely (GWOBA), GWO is used to explore the problem space alone and pass the best two solutions to BA to guide its local search, then BA digs deeper and find the best solution. The new proposed approach has been tested using 30 standard benchmark functions from CEC2017 benchmark suite. The performance of the hybrid algorithm has been compared to the original GWO, BA and the Whale optimization algorithm (WOA). We use a set of performance indicators to evaluate the efficiency of the method. Results over various dimensions show the superiority of the proposed algorithm.

Keywords: Optimization · Greywolf optimizer · Bat algorithm
CEC2017 · Hybridization

1 Introduction

In most real-world applications, there is always a need to minimize (cost, time or waste) or maximize (performance, benefits or profit). Both minimization and maximization are usually considered the main objectives of optimization problems. In many cases, traditional deterministic algorithms fail to solve optimization problems in practice. Real problems are usually nonlinear, complex and multimodal with many local optima, they might also suffer from complex constraints

and high number of dimensions, which motivated researchers and scientists to design non-deterministic optimization algorithms to solve such problems.

Almost all new optimization algorithms are inspired from Nature, hence they are referred to as nature-inspired algorithms. Some algorithms were inspired from chemical and physical systems of nature like simulated annealing [9] and harmony search [6], others were inspired from the success of biological systems in solving problems like genetic algorithm [7,8] and differential evolution (DE) [12], while the most popular algorithms were inspired from swarm intelligence like particle swarm optimization [5], ant colony optimization [3,4], cuckoo search [16], bat algorithm [15], Greywolf optimizer [10] and firefly algorithm [14].

In this paper, we propose a new hybrid Grey Wolf/Bat optimization algorithm (GWOBA). The proposed algorithm is designed to use the best of GWO and BA strategies in order to achieve better overall exploration and exploitation behavior. During the first half of iterations, GWO is used to explore the search space effectively using its high exploration behavior. After GWO is done, BA which has better exploitation capabilities, is initiated using the best set of solutions found by GWO and continues searching for the rest of iterations.

The performance of GWOBA is evaluated using 30 benchmark problems of CEC2017 [1] and is compared to GWO, BA and WOA performance. The results are evaluated using the guidelines provided in CEC2017 and they clearly indicate that GWOBA provides better results in most benchmark problems.

2 Preliminaries

In this section, we review the standard metaheuristics: Grey wolf optimizer (GWO) and bat algorithm. GWO simulates the leadership hierarchy and hunting procedure of grey wolves in nature proposed by Mirjalili et al. in 2014 [10], while Bat algorithm was proposed by Xin-She Yang in 2010 [15], inspired by the echolocation of microbats. Both algorithms have shown superiority over many other metaheuristics over wide range of applications.

2.1 Grey Wolf Optimizer

GWO algorithm is inspired from the hunting mechanism of grey wolves and their social hierarchy. The closest wolves (solutions) to the prey (optimum) are called α wolves. The β and δ wolves are the second and the third best solutions respectively. Their location is denoted in the search space as X_α , X_β and X_δ . The rest of the wolves follow these three wolves as shown in the following equations:

$$X(t+1) = X_p(t) - A \cdot |C \cdot X_p(t) - A \cdot X(t)| \quad (1)$$

where t is the current iteration, A and C are coefficient vectors, X_p is the position vector of the prey, and X indicates the position vector of a grey wolf. The vectors A and C are calculated as follows:

$$A = 2a \cdot r_1 - a, \quad C = 2 \cdot r_2 \quad (2)$$

where r_1, r_2 are random vectors in $[0, 1]$ and the exploration rate (a) is linearly decreased from 2 to 0 over the course of iterations as shown below:

$$a = 2 - t \cdot \frac{2}{MaxIter} \quad (3)$$

where t is the current iteration and $MaxIter$ is the total number of iterations allowed for the optimization. From Eqs. (2) and (3), the random vector A resides in the interval $[-a, a]$. Exploration and exploitation are guaranteed by the adaptive values of a [10] allowing GWO to transit smoothly between exploration and exploitation.

Supposing that the α, β and δ wolves have better knowledge about the potential location of prey, other agents are obliged to update their positions to follow them.

$$D_\alpha = |C_1 \cdot X_\alpha - X|, D_\beta = |C_2 \cdot X_\beta - X|, D_\delta = |C_3 \cdot X_\delta - X| \quad (4)$$

where $X_\alpha, X_\beta,$ and X_δ are the best three solutions at a given iteration.

$$X_1 = X_\alpha - A_1 \cdot (D_\alpha), X_2 = X_\beta - A_2 \cdot (D_\beta), X_3 = X_\delta - A_3 \cdot (D_\delta) \quad (5)$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \quad (6)$$

2.2 Bat Algorithm

Bats use echolocation to sense distance and hunt, they emit sound pulses and process the signal of the echo. They can adjust the wavelength, the emission rate and the loudness of the emitted pulses. In the Bat algorithm, bats move randomly with velocity V_i at position X_i with varying wavelength λ and loudness A_0 to search for prey. Virtual bats adjust their position according to the following equations:

$$F_i = F_{min} + (F_{max} - F_{min})\beta \quad (7)$$

$$V_i^t = V_i^{t-1} + (X_i^t - X_*)F_i \quad (8)$$

$$X_i^t = X_i^{t-1} + V_i^t \quad (9)$$

Where β is a random vector in the range $[0,1]$ drawn from uniform distribution. X_* is the current global best location. F_{min} and F_{max} represent the minimum and maximum frequency needed depending on the problem. V_i represents the velocity vector.

Probabilistically a local search is to be performed using a random walk as in the following equation:

$$X_{new} = X_* + \epsilon A^t \quad (10)$$

Where A^t is the average loudness of all bats at this time and ϵ is a random number uniformly drawn from $[-1, 1]$.

The updating of the loudness is performed using the following equation:

$$A_i^{t+1} = \delta A^t \quad (11)$$

Where δ is a constant selected experimentally.

The emission rate r_i controls the application of the local search and is updated using the equation:

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)] \quad (12)$$

Where r_i^0 is the initial pulse emission rate and is a constant greater than 0.

3 Proposed Hybrid Grey Wolf-Bat Optimization Algorithm

The hybrid GWOBA algorithm is about mixing the high exploration skills of GWO algorithm with the high exploitative properties of BA. First, GWO algorithm is initialized with a random set of solutions then it iterates to find a better set. After $MaxIter/2$ iterations, only the best two solutions (X_α, X_β) are passed to the bat algorithm (BA) as initial guess to help the algorithm to focus on them. BA then runs for $MaxIter/2$ iterations and returns the best solution which is considered the best solution of the hybrid algorithm.

In order to hybrid GWO and BA algorithms, we needed to make some modifications to both algorithms. The first modification was in GWO algorithm, and the question was: how many leading wolves do we need? By default GWO algorithm makes use of three leading wolves (alpha, beta and delta) which leads the rest of the herd to the prey location. In some problems, the three wolves are located in three different areas within the search space, which slows down the convergence of the algorithm. After many experiments, we decided to use the best two solutions only (alpha and beta) to lead the rest of the solutions to the prey location.

The reason behind this decision is that passing three good solutions to BA might create a divergence, bats are following the best solution and then keep following it with a very low chance to switch this best with another far one. In a three pole function, where one pole is the optimal solution and the other two poles are just local optimas, if GWO has three leaders, having each leader settling down in a pole far away from each other, BA will start searching around them and it might find a best solution around the worst leader and then it will keep searching around the fake best with a very low chance to get out of the pole. This is a typical scenario where GWOBA fails to jump out of a local optima. This modification raised the ability of the algorithm to locate where to search more precisely and helped our algorithm to have significance over the original GWO. Equations (4, 5, 6) are modified to be:

$$D_\alpha = |C_1 \cdot X_\alpha - X|, D_\beta = |C_2 \cdot X_\beta - X| \quad (13)$$

$$X_1 = X_\alpha - A_1 \cdot (D_\alpha), X_2 = X_\beta - A_2 \cdot (D_\beta) \quad (14)$$

$$X(t+1) = \frac{X_1 + X_2}{2} \quad (15)$$

The second modification affects Eq. (3), to enhance the exploration capabilities of the wolves, we forced $|A| > 1$ by decreasing the exploration rate a from two down to one, so Eq. (3) becomes:

$$a = 2 - \frac{t}{MaxIter} \quad (16)$$

where t is the iteration number and $MaxIter$ is the total number of iterations allowed for the optimization. The algorithm describing the hybrid GWOBA algorithm is outlined in Algorithm 1.

Algorithm 1. Algorithm for GWOBA optimizer

<p>Input : n Number of agents $MaxIter$ Maximum iterations r_i^0 Initial pulse emission rate A_i Loudness Result: The optimal solution (X_*)</p> <p>1 <i>Initialize the grey wolf population X_i</i> <i>($i = 1, 2, \dots, n$) randomly</i></p> <p>2 <i>Initialize a, A, and C</i></p> <p>3 <i>Evaluate the positions of wolves</i></p> <p>4 $X_\alpha =$ <i>The best search agent</i></p> <p>5 $X_\beta =$ <i>The second best search agent</i></p> <p>6 $t = 0$</p> <p>7 while ($t < MaxIter/2$) do</p> <p>8 for each X_i do</p> <p>9 Update X_i position Eq.(15).</p> <p>10 end</p> <p>11 Update a, A and C Eqs.(16,2)</p> <p>12 Evaluate the positions of wolves.</p> <p>13 Update X_α and X_β</p> <p>14 $t = t + 1$</p> <p>15 end</p> <p>16 <i>Initialize the first two bats $X_1 = X_\alpha$,</i> $X_2 = X_\beta$ <i>and initialize the rest ($n - 2$)</i> <i>bats X_i ($i = 3, 4, \dots, n$) randomly</i></p>	<p>17 <i>Initialize frequencies f_i, pulse rates r_i</i> <i>and the loudness A_i</i></p> <p>18 <i>Find the best solution based on fitness X_*</i></p> <p>19 while ($t < MaxIter$) do</p> <p>20 for each X_i do</p> <p>21 Generate new solution (X_i^{new}) by</p> <p> adjusting frequency Eqs.(7 to 9)</p> <p>22 if $rand > r_i$ then</p> <p>23 Generate a local solution</p> <p> (X_i^{new}) around the best</p> <p> solution X_* Eq.(10)</p> <p>24 end</p> <p>25 if $rand < A_i$ <i>and fitness</i> (X_i^{new}) $< fitness(X_i)$ then</p> <p>26 Update the position of X_i to</p> <p> X_i^{new}</p> <p>27 Reduce A_i Eq.(11)</p> <p>28 Increase r_i Eq.(12)</p> <p>29 end</p> <p>30 Update X_*</p> <p>31 end</p> <p>32 $t = t + 1$</p> <p>33 end</p> <p>34 return X_*</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

4 Experimental Results and Discussion

This section summarizes the results from applying the proposed GWOBA on 30 benchmark functions from the new CEC2017 test suit. The next two subsections contain the results and the analysis for the whole set of functions. We use a set of qualitative measures to analyze the results obtained by the methods we apply. The first four metrics (Mean Fitness, Best Fitness, Worst Fitness and Median

Fitness) give a measure of the mean, best, worst, and median expected performance of the algorithms. The fifth measure (Standard Deviation) is adopted to show the precision of each optimizer. The sixth metric (Root-Mean-Square Error (RMSE)) shows how accurate is the optimizer. The last two metrics (T-Test [2, 11] and Wilcoxon Rank Sum Test [13]) are used to directly comparing algorithms in pairs and show whether the difference between them is significant or not.

4.1 CEC2017 Benchmark Suite

In this suite [1], benchmark problems were developed with several novel features such as new basic problems, composing test problems by extracting features dimension-wise from several problems, graded level of linkages, rotated trap problems, and so on. All test functions are minimization problems with a shifted global optimum randomly distributed in the range $[-80, 80]$. The suite consists of 30 functions divided into different categories unimodal, simple multimodal, hybrid and composition functions to simulate different types of real life problems.

4.2 Results and Parameter Settings

In order to benchmark our hybrid optimizer, we compared it to its primitive algorithms GWO, BA and to the whale optimization algorithm (WOA) as well. All experiments were done using 20 agents and the standard parameters for all algorithms.

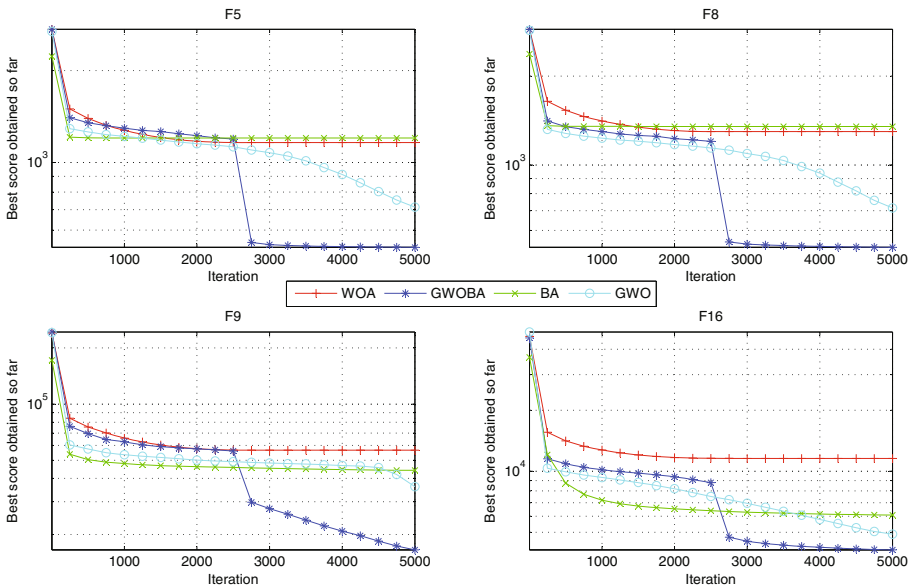


Fig. 1. Mean convergence curves of the compared algorithms ($D = 100$)

In our experiments, we used 30 minimization problems with four different dimensions: $D = 10, 30, 50, 100$ from CEC2017 benchmark suite. Each optimizer runs 51 runs per problem with maximum function evaluations ($MaxFES = 1000 * D$) (i.e. 10000 function evaluations for 10 dimensions' problem). All problems have the global optimum within the search range $[-100, 100]$. All four optimizers were initialized with uniform random initialization within the search space with a random seed based on time.

In Fig. 1, the four algorithms GWOBA, WOA, GWO and BA are compared using the same dimensions and against the same benchmark functions. We plotted the mean convergence curve for each algorithm along 51 runs for dimension ($D = 100$) and functions ($F5, F8, F9, F16$). As can be noticed, our hybrid algorithm mean convergence curve is diving down when it passes half the maximum number of iterations causing the results to be far better than the other algorithms.

In Table 1, mean error and standard deviation results are tabulated for all 30 benchmark functions ($F1$ through $F30$) along 51 runs for dimension ($D = 100$).

Table 1. Mean error and std. for dimension ($D = 100$)

Fun.	Mean				Std.			
	WOA	GWOBA	BA	GWO	WOA	GWOBA	BA	GWO
F01	1.472E+10	2.580e+04	4.253E+06	5.985E+10	3.810E+09	2.226e+04	3.190E+05	1.271E+10
F02	2.03E+168	9.65e+108	5.63E+126	2.79E+138	∞	6.89e+109	4.02E+127	1.99E+139
F03	8.674E+05	2.022e+05	4.719E+05	2.658E+05	1.450E+05	2.315E+04	1.745E+05	2.172e+04
F04	3.217E+03	2.868E+02	2.817e+02	5.970E+03	7.679E+02	3.794e+01	4.827E+01	1.897E+03
F05	1.162E+03	5.274e+02	1.202E+03	7.140E+02	1.008E+02	4.746e+01	1.620E+02	8.744E+01
F06	9.264E+01	4.522E+01	7.934E+01	4.343e+01	9.672E+00	3.617e+00	7.638E+00	4.83E+00
F07	2.834E+03	1.331e+03	5.980E+03	1.464E+03	1.414E+02	1.342E+02	1.508E+03	1.232e+02
F08	1.298E+03	5.265e+02	1.351E+03	7.150E+02	1.193E+02	6.864e+01	1.646E+02	8.472E+01
F09	5.665E+04	1.656e+04	4.415E+04	3.606E+04	1.583E+04	2.472e+03	1.007E+04	1.362E+04
F10	2.451E+04	1.452e+04	1.689E+04	1.544E+04	2.088E+03	1.320E+03	1.150e+03	3.634E+03
F11	1.282E+05	1.333e+03	2.308E+03	6.093E+04	6.306E+04	1.716e+02	6.149E+02	1.120E+04
F12	3.119E+09	3.490e+07	5.408E+07	1.475E+10	1.020E+09	1.149e+07	2.307E+07	8.534E+09
F13	3.088E+07	5.236e+04	2.385E+05	1.641E+09	2.149E+07	2.235e+04	4.783E+04	1.881E+09
F14	7.826E+06	5.537E+05	3.948e+05	6.976E+06	3.195E+06	3.467E+05	2.031e+05	3.991E+06
F15	5.848E+06	4.197e+04	1.437E+05	2.085E+08	7.563E+06	1.763e+04	5.095E+04	3.141E+08
F16	1.158E+04	4.100e+03	6.082E+03	4.895E+03	1.617E+03	7.989E+02	1.135E+03	7.242e+02
F17	6.806E+03	3.341e+03	4.910E+03	3.828E+03	1.536E+03	5.668e+02	8.656E+02	1.092E+03
F18	7.209E+06	7.814e+05	8.246E+05	5.193E+06	4.270E+06	3.911E+05	3.515e+05	2.90E+06
F19	3.903E+07	1.939e+06	3.953E+06	2.857E+08	2.872E+07	8.898e+05	1.187E+06	4.480E+08
F20	4.466E+03	2.972E+03	4.298E+03	2.826e+03	6.726E+02	5.469e+02	8.351E+02	7.996E+02
F21	2.067E+03	7.820e+02	2.485E+03	9.254E+02	2.607E+02	6.512e+01	2.548E+02	9.667E+01
F22	2.579E+04	1.576e+04	1.926E+04	1.699E+04	2.100E+03	1.225e+03	2.282E+03	2.245E+03
F23	2.697E+03	1.185e+03	3.399E+03	1.379E+03	2.426E+02	7.677e+01	3.764E+02	1.065E+02
F24	3.716E+03	1.873e+03	4.738E+03	1.997E+03	3.616E+02	1.469e+02	6.327E+02	1.838E+02
F25	2.552E+03	8.310E+02	7.933e+02	4.631E+03	3.216E+02	4.297e+01	6.283E+01	9.867E+02
F26	3.124E+04	1.271e+04	3.488E+04	1.457E+04	3.640E+03	1.189e+03	1.150E+04	1.40E+03
F27	2.737E+03	1.350e+03	2.445E+03	1.453E+03	7.615E+02	1.496e+02	6.718E+02	1.680E+02
F28	3.464E+03	6.096e+02	6.152E+02	6.586E+03	6.093E+02	4.194e+01	4.277E+01	1.707E+03
F29	1.298E+04	5.080e+03	7.636E+03	6.019E+03	1.676E+03	5.909e+02	1.236E+03	7.203E+02
F30	6.088E+08	1.033e+07	1.582E+07	1.773E+09	2.941E+08	3.123e+06	8.107E+06	1.625E+09

Table 2. Best and worst error for dimension ($D = 100$)

Fun.	Best				Worst			
	WOA	GWOBA	BA	GWO	WOA	GWOBA	BA	GWO
F01	9.395E+09	8.900e+03	3.605E+06	2.462E+10	2.542E+10	1.567e+05	4.979E+06	8.286E+10
F02	1.94E+142	7.532E+79	2.027e+63	2.30E+108	1.01E+170	4.92e+110	2.87E+128	1.42E+140
F03	6.257E+05	1.548e+05	2.686E+05	2.144E+05	1.338E+06	2.630e+05	9.213E+05	3.172E+05
F04	1.797E+03	1.869E+02	1.847e+02	2.840E+03	4.976E+03	3.618e+02	3.761E+02	1.219E+04
F05	9.415E+02	4.172e+02	8.985E+02	5.018E+02	1.386E+03	6.222e+02	1.560E+03	1.104E+03
F06	7.553E+01	3.854E+01	6.468E+01	2.917e+01	1.183E+02	5.450E+01	9.983E+01	5.267e+01
F07	2.485E+03	1.075e+03	3.189E+03	1.238E+03	3.107E+03	1.669e+03	1.133E+04	1.836E+03
F08	1.117E+03	3.930e+02	1.027E+03	5.268E+02	1.650E+03	6.683e+02	1.720E+03	9.315E+02
F09	3.478E+04	1.193e+04	3.183E+04	1.648E+04	1.088E+05	2.552e+04	6.718E+04	5.968E+04
F10	2.018E+04	1.121E+04	1.408E+04	1.089e+04	2.887E+04	1.718e+04	1.906E+04	2.905E+04
F11	5.518E+04	9.760e+02	1.341E+03	4.094E+04	3.581E+05	1.654e+03	4.412E+03	9.464E+04
F12	1.343E+09	1.556e+07	1.663E+07	3.668E+09	6.470E+09	5.518e+07	1.415E+08	4.545E+10
F13	8.183E+06	1.999e+04	1.605E+05	1.598E+08	1.036E+08	1.346e+05	3.932E+05	1.247E+10
F14	1.855E+06	1.076E+05	1.074e+05	1.152E+06	2.031E+07	1.907E+06	9.684e+05	1.527E+07
F15	3.192E+05	1.288e+04	7.604E+04	1.118E+06	3.197E+07	8.912e+04	3.067E+05	1.504E+09
F16	8.931E+03	2.496e+03	4.360E+03	3.135E+03	1.571E+04	5.841e+03	1.071E+04	6.606E+03
F17	4.676E+03	1.919e+03	2.986E+03	2.305E+03	1.270E+04	4.687e+03	7.127E+03	6.971E+03
F18	1.953E+06	2.386e+05	3.482E+05	1.580E+06	1.860E+07	2.242E+06	1.874e+06	1.320E+07
F19	3.510E+06	3.158e+05	1.336E+06	4.731E+06	1.311E+08	4.748e+06	6.881E+06	2.099E+09
F20	2.486E+03	2.100E+03	2.804E+03	1.982e+03	5.703E+03	4.352e+03	6.136E+03	5.695E+03
F21	1.513E+03	6.848e+02	1.896E+03	7.398E+02	2.713E+03	9.481e+02	3.085E+03	1.487E+03
F22	2.074E+04	1.311e+04	1.506E+04	1.388E+04	3.072E+04	1.847e+04	2.465E+04	2.995E+04
F23	2.145E+03	9.809e+02	2.631E+03	1.247E+03	3.184E+03	1.340e+03	4.240E+03	1.739E+03
F24	3.163E+03	1.634E+03	3.099E+03	1.562e+03	4.710E+03	2.310e+03	6.192E+03	2.492E+03
F25	1.945E+03	7.366E+02	6.503e+02	2.591E+03	3.598E+03	9.115e+02	9.286E+02	7.854E+03
F26	2.367E+04	1.035E+04	3.143e+02	1.179E+04	4.015E+04	1.485e+04	5.589E+04	1.862E+04
F27	1.726E+03	9.067e+02	1.316E+03	1.075E+03	5.046E+03	1.740e+03	4.517E+03	1.895E+03
F28	2.432E+03	5.474E+02	5.450e+02	3.317E+03	5.215E+03	7.107e+02	7.438E+02	1.136E+04
F29	1.013E+04	3.995e+03	5.487E+03	4.605E+03	1.677E+04	6.311e+03	1.187E+04	7.564E+03
F30	1.689E+08	3.561e+06	5.873E+06	1.278E+08	1.838E+09	1.951e+07	5.170E+07	5.499E+09

The results show that GWOBA mean error is better than the others for 25 out of 30 functions, even when it fails to achieve the best results, it loses with a very low order of magnitude. Regarding the standard deviation results, GWOBA lost the lead for 6 out of 30 functions, but without order of magnitude.

In Table 2, best and worst error values are tabulated for all 30 benchmark functions ($F1$ through $F30$) along 51 runs for dimension ($D = 100$). The results show that GWOBA found the best solution for 20 out of 30 functions. On the other hand, GWOBA has the lowest worst solutions for 27 out of 30 functions, which means that it might not get the best solution every time but it will not get the worst.

Wilcoxon rank sum test and T-test results with $\alpha = 0.05$ are tabulated in Table 3, the tests are calculated on different dimensions for all four optimizers. Since we are interested in the performance of GWOBA against the other algorithms, we report all the comparisons with GWOBA only. It is clear that

Table 3. Wilcoxon rank sum test and T-test for all dimensions

D	Wilcoxon rank sum test			T-test		
	WOA	BA	GWO	WOA	BA	GWO
10	7.70E-39	3.06E-40	4.25E-07	5.49E-09	1.09E-06	1.17E-02
30	9.81E-44	1.73E-19	3.44E-08	2.00E-13	1.02E-04	2.02E-10
50	1.96E-40	1.08E-15	3.08E-15	3.84E-17	2.75E-04	1.14E-14
100	2.35E-43	1.42E-13	9.51E-26	8.52E-18	3.81E-11	4.23E-19

GWOBA performs better than GWO, BA, and WOA as per T-test results at a significance level $\alpha = 0.05$ for all dimensions.

The last measure to test our hybrid algorithm performance is to sum all wins for the proposed algorithm against GWO, BA, and WOA for different dimensions and for every qualitative measure. Table 4 summarizes the overall performance of GWOBA, showing excellent success rates in mean, std., worst and RMSE measures. The results also show that the algorithm performance is very solid against higher dimensions, while other algorithms fail to get competitive results in the same conditions.

Table 4. Overall success rate of GWOBA for all set of qualitative measures

D	Mean	Std.	Best	Worst	Median	RMSE
10	27	23	13	22	21	26
30	22	27	18	25	20	24
50	22	24	18	21	21	22
100	25	24	20	27	25	25
Success Rate	80%	82%	58%	79%	73%	81%

The mean error and standard deviation results for all dimensions proves that GWOBA continues its great job defeating other algorithms even for the highest dimensions. Having both low mean error and low standard deviation for all dimensions means that GWOBA has the best precision and accuracy compared to the other algorithms. As a result, GWOBA showed excellent performance against its primitive algorithms GWO and BA as well as WOA.

5 Conclusion and Future Work

In this paper, a hybrid GWO/BA algorithm is proposed. The proposed GWOBA algorithm makes use of both GWO and BA strengths in exploration and exploitation. GWOBA is compared to GWO, BA, and WOA on CEC2017 benchmark suite. Results were assessed using a set of performance indicators. Significant

improvement was observed in the performance of GWOBA against other methods. GWOBA proved excellent precision and accuracy in different search space terrains with overall best mean value, standard deviation, median, best, worst, and RMSE. The different tests show that GWOBA is very solid regarding higher dimensions as well as lower ones. The algorithm passed two significance tests on four different dimensions. In the near future, we will test GWOBA in other applications to measure its performance in real life problems.

References

1. Awad, N., Ali, M., Liang, J., Qu, B., Suganthan, P.: Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective real-parameter numerical optimization (2016)
2. Derrac, J., García, S., Molina, D., Herrera, F.: 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), 3–18 (2011)
3. Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. *IEEE Comput. Intell. Mag.* **1**(4), 28–39 (2006)
4. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **26**(1), 29–41 (1996)
5. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: *Proceedings of the Sixth International Symposium on Micro Machine and Human Science, MHS 1995*, pp. 39–43. IEEE (1995)
6. Geem, Z.W., Kim, J.H., Loganathan, G.: A new heuristic optimization algorithm: harmony search. *Simulation* **76**(2), 60–68 (2001)
7. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston (1989)
8. Holland, J.H.: *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor (1975)
9. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., et al.: Optimization by simulated annealing. *Science* **220**(4598), 671–680 (1983)
10. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61 (2014)
11. Rice, J.: *Mathematical Statistics and Data Analysis*. Nelson Education (2006)
12. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **11**(4), 341–359 (1997)
13. Wilcoxon, F.: Individual comparisons by ranking methods. *Biom. Bull.* **1**(6), 80–83 (1945)
14. Yang, X.S.: Firefly algorithms for multimodal optimization. In: *International Symposium on Stochastic Algorithms*, pp. 169–178. Springer (2009)
15. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, pp. 65–74 (2010)
16. Yang, X.S., Deb, S.: Cuckoo search via lévy flights. In: *World Congress on Nature & Biologically Inspired Computing, NaBIC 2009*, pp. 210–214. IEEE (2009)