

Dipper Throated Optimization Algorithm for Unconstrained Function and Feature Selection

Ali E. Takieldean¹, El-Sayed M. El-kenawy^{1,2}, Mohammed Hadwan^{3,4,5,*} and Rokaia M. Zaki^{6,7}

¹Faculty of Artificial Intelligence, Delta University for Science and Technology, Mansoura, 35712, Egypt

²Department of Communications and Electronics, Delta Higher Institute of Engineering and Technology, Mansoura, 35111, Egypt

³Department of Information Technology, College of Computer, Qassim University, Buraydah, 51452, Saudi Arabia

⁴Department of Computer Science, College of Applied Sciences, Taiz University, Taiz, Yemen

⁵Intelligent Analytics Group (IAG), College of Computer, Qassim University, Buraydah, Saudi Arabia

⁶Higher Institute of Engineering and Technology, Kafrelsheikh, Egypt

⁷Department of Electrical Engineering, Shoubra Faculty of Engineering, Benha University, Egypt

*Corresponding Author: Mohammed Hadwan. Email: m.hadwan@qu.edu.sa

Received: 13 December 2021; Accepted: 13 January 2022

Abstract: Dipper throated optimization (DTO) algorithm is a novel with a very efficient metaheuristic inspired by the dipper throated bird. DTO has its unique hunting technique by performing rapid bowing movements. To show the efficiency of the proposed algorithm, DTO is tested and compared to the algorithms of Particle Swarm Optimization (PSO), Whale Optimization Algorithm (WOA), Grey Wolf Optimizer (GWO), and Genetic Algorithm (GA) based on the seven unimodal benchmark functions. Then, ANOVA and Wilcoxon rank-sum tests are performed to confirm the effectiveness of the DTO compared to other optimization techniques. Additionally, to demonstrate the proposed algorithm's suitability for solving complex real-world issues, DTO is used to solve the feature selection problem. The strategy of using DTOs as feature selection is evaluated using commonly used data sets from the University of California at Irvine (UCI) repository. The findings indicate that the DTO outperforms all other algorithms in addressing feature selection issues, demonstrating the proposed algorithm's capabilities to solve complex real-world situations.

Keywords: Metaheuristic optimization; swarm optimization; feature selection; function optimization

1 Introduction

Optimization is the process of obtaining the greatest or least objective function value for a set of inputs. It is the subject of various machine techniques that draw on artificial neural networks. Hundreds of famous optimization algorithms have become accessible, and dozens of technologies are available in major scientific code libraries. Given the problems of optimization, selecting what methods



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

can thus be challenging [1]. Optimization is how a function has the lowest or highest output of input parameters or arguments. In machines where the input parameters of the tasks like the floating-point values are numerical, continuous functions optimization frequently arises. The function returns an evaluation of the parameter of real-life [2].

Continuous function optimization may be used to distinguish between such problems with discrete variables, known as combined optimization problems [3]. Different techniques may be resolved, organized, and called to optimize the problems involving continuous functions. The information about the objective function utilized and used throughout the optimization process depends on one technique of optimization classification [4]. The more information about the target function is generally understood, the easier it is to optimize because knowledge can be applied effectively [5]. Perhaps the significant difference between optimization techniques is identifying the destination function in one location [6]. It means that the first derivative of the feature may be used to identify a possible solution (gradient or route). It distinguishes itself from other not-calculated gradient data [7]. Metaheuristic optimization is the optimization process utilizing metaheuristic techniques. Almost every area of life is involved, much from engineering to business, holiday preparation to internet travel [8]. The use of those readily available resources must be maximized due to the continuous scarcity of money, resources, and time. The vast majority are non-linear, multimodal, and quite restrictive in real-life problems [9]. Different objectives frequently collide. Even if one goal is set, Optimum solutions are not always available [10]. Usually, a faultless or failed response is not simple to find. Many metaheuristic algorithms have been published, including swarm intelligence, anthrax optimization, optimization of the particulate swarm. In previous articles [11]. The feature selection issue may be seen as a multi-objective optimization problem in which two conflicting objectives must be met: picking the fewest possible features while attaining maximum classification accuracy [12]. The solution with the most distinctive features and the maximum classification accuracy is deemed optimal [13].

2 Literature Review

Meta-heuristics refers to generic methods that normally used to solve complex and challenging combinatorial search problems. Generally, the problems solved by metaheuristic algorithms are challenging for computer scientists due to the need to examining a huge number of combinations that usually exponential with conflicting objectives [14]. Many metaheuristic algorithms have been proposed to tackle real-world situation such as image segmentation [15], water allocation and crop planning [16], Nurse Rostering [17], power load dispatch [18], and Parkinson diagnosis [19]. Several survey papers are available for more information about metaheuristic algorithms [20–23].

Nature-inspired metaheuristic algorithms to solve complex real-world situation have attracted the attention of the researchers in the scientific community. Many new nature-inspired metaheuristic algorithms have been developed, including Symbiotic Organisms Search [24], Bat Algorithm (BA) [25], Bacterial Foraging Opt [26], Gravitational Search Algorithm [27], Firefly Algorithm (FA) [28], Krill Herd [29], Grey Wolf Optimization (GWO) algorithm [30,31], Cuckoo Search [32], Harmony search algorithm [33], Whale optimization [34], Social spider optimization [35], and Biogeography-based Opt [36].

Several research paper can be found in the literature tackling feature selection as in [37–40]. When it comes to feature selection, metaheuristic algorithms are instrumental because they deal with the dimensions of the data set to make predictions [38]. However, when the dimensionality of the data sets is increased, the performance of classification methods suffers because of this. Furthermore, high-dimensional data sets have several drawbacks, including a long model creation time, redundant data,

and reduced performance, making data analysis very challenging [40]. The feature selection step is a major preprocessing step that is used to resolve this problem. Its goal is to select a subset of features from an extensive data set while also increasing the accuracy of the classification or clustering model, resulting in the removal of noisy, extraneous, and ambiguous data. The following section present the proposed DTO optimizer.

3 Proposed Dipper Throated Optimization Algorithm

Dipper Throated bird is a member of the genus *Cinclus* in the bird family *Cinclidae*, so-called because of their bobbing or dipping movements see Fig. 1. They are unique among passerines for their ability to dive, swim, and hunts underwater. Besides, it can fly rapidly and straight without pauses or glides because it has short and flexible wings. Dipper Throated bird has its unique hunting technique, it performs rapid bowing movements, enhanced by the pure white of the breast. Once the prey is detected, it dives headfirst into the water, even into the turbulent and fast-flowing water. When it became on the bottom, it turns up stones and pebbles, to disturb aquatic invertebrates, aquatic insects, and small fish. the Dipper walks on the bottom by grasping stones. It often walks against the current, with the head downwards to locate prey, it can be stable for a long time with its strong feet, also, it can walk into the water and deliberately submerge, by using its wings effectively and walk along with the bottom keeping its head well down and its body oblique to secures its food.

3.1 Mathematical Formulation

Mathematically, the Dipper Throated Optimization (DTO) algorithm assumes the birds are swimming and flying to search for food resources N_{fs} available for n birds. The birds' locations, BP , and velocities, BV , can be represented by the following matrices:

$$BP = \begin{bmatrix} BP_{1,1} & BP_{1,2} & BP_{1,3} & \dots & BP_{1,d} \\ BP_{2,1} & BP_{2,2} & BP_{2,3} & \dots & BP_{2,d} \\ BP_{3,1} & BP_{3,2} & BP_{3,3} & \dots & BP_{3,d} \\ \dots & \dots & \dots & \dots & \dots \\ BP_{n,1} & BP_{n,2} & BP_{n,3} & \dots & BP_{n,d} \end{bmatrix} \tag{1}$$

$$BV = \begin{bmatrix} BV_{1,1} & BV_{1,2} & BV_{1,3} & \dots & BV_{1,d} \\ BV_{2,1} & BV_{2,2} & BV_{2,3} & \dots & BV_{2,d} \\ BV_{3,1} & BV_{3,2} & BV_{3,3} & \dots & BV_{3,d} \\ \dots & \dots & \dots & \dots & \dots \\ BV_{n,1} & BV_{n,2} & BV_{n,3} & \dots & BV_{n,d} \end{bmatrix} \tag{2}$$

where BP_{ij} indicates i^{th} bird in the j^{th} dimension for $i \in 1, 2, 3, \dots, n$ and $j \in 1, 2, 3, \dots, d$. BV_{ij} indicates i^{th} bird velocity in the j^{th} dimension for $i \in 1, 2, 3, \dots, n$ and $j \in 1, 2, 3, \dots, d$. The initial locations of BP_{ij} are uniform distribution within lower and upper bounds. The fitness values $f = f_1, f_2, f_3, \dots, f_n$ are calculated for each bird as in the following array

$$f = \begin{bmatrix} f_1(BP_{1,1}, BP_{1,2}, BP_{1,3}, \dots, BP_{1,d}) \\ f_2(BP_{2,1}, BP_{2,2}, BP_{2,3}, \dots, BP_{2,d}) \\ f_3(BP_{3,1}, BP_{3,2}, BP_{3,3}, \dots, BP_{3,d}) \\ \dots \\ f_n(BP_{n,1}, BP_{n,2}, BP_{n,3}, \dots, BP_{n,d}) \end{bmatrix} \quad (3)$$

where the fitness value indicates the quality of food source searched by each bird. The optimal value means mother bird. These values are then sorted in ascending order. The first best solution is declared to be BP_{best} . The remaining solutions are supposed to be normal birds BP_{nd} for follower birds. The global best solution is declared to be BP_{Gbest} .

First DTO mechanism by this optimizer to update the swimming bird position is based on the following equation:

$$BP_{nd}(t+1) = BP_{best}(t) - C_1 \cdot |C_2 \cdot BP_{best}(t) - BP_{nd}(t)| \quad (4)$$

where $BP_{nd}(t)$ is a normal bird position at iteration t and $BP_{best}(t)$ is the best bird position. The “.” is pairwise multiplication. $BP_{nd}(t+1)$ is the updated bird position for the solution.

The C_1 and C_2 are updated within the iterations by the following

$$\begin{aligned} C_1 &= 2c \cdot r_1 - c, \\ C_2 &= 2r_1, \\ c &= 2 \left(1 - \left(\frac{t}{T_{max}} \right)^2 \right) \end{aligned} \quad (5)$$

where c changes from 2 to 0 exponentially, r_1 is a random value in $[0, 1]$ and T_{max} in the total number of iterations.

Second DTO mechanism is based on updating the flying bird position and velocity by the following equations. The flying birds' positions are updated as

$$BP_{nd}(t+1) = BP_{nd}(t) + BV(t+1) \quad (6)$$

where $BP_{nd}(t+1)$ is the new bird position for normal birds, and the updated velocity of each bird $BV(t+1)$ is calculated as

$$BV(t+1) = C_3 BV(t) + C_4 r_2 (BP_{best}(t) - BP_{nd}(t)) + C_5 r_2 (BP_{Gbest} - BP_{nd}(t)) \quad (7)$$

where C_3 is a weight value, C_4 and C_5 are constants. BP_{Gbest} is the global best position and r_2 is a random number in $[0; 1]$.

The DTO algorithm can be described by this equation

$$BP_{nd}(t+1) = \begin{cases} BP_{best}(t) - C_1 \cdot |M| & \text{if } R < 0.5 \\ BP_{nd}(t) + BV(t+1) & \text{otherwise} \end{cases} \quad (8)$$

where $M = C_2 \cdot BP_{best}(t) - BP_{nd}(t)$ and R is a random value in $[0, 1]$.



Figure 1: White throated dipper

Algorithm 1: The DTO Algorithm

Initialization positions $BP_i (i = 1, 2, \dots, n)$ with size n ,
 velocities $BV_i (i = 1, 2, \dots, n)$, total number of iterations T_{max} ,
 fitness function $f_n, c, C_1, C_2, C_3, C_4, C_5, r_1, r_2, R, t = 1$
Calculate objective function f_n for each bird BP_i
Find best bird BP_{best}
While $t \leq T_{max}$ **do**
 for $(i = 1 : i < n + 1)$ **do**
 if $(R < 0.5)$ **then**
 Update position of current swimming bird as
 $BP_{nd}(t + 1) = BP_{best}(t) - C_1 \cdot |C_2 \cdot BP_{best}(t) - BP_{nd}(t)|$
 else
 Update velocity of current flying bird as
 $BV(t + 1) = C_3 BV(t) + C_4 r_2 (BP_{best}(t) - BP_{nd}(t)) + C_5 r_2 (BP_{Gbest} - BP_{nd}(t))$
 Update position of current flying bird as
 $BP_{nd}(t + 1) = BP_{nd}(t) + BV(t + 1)$
 end if
 end for
Calculate objective function f_n for each bird BP_i
Update c, C_1, C_2, R
Find best bird BP_{best}
Set $BP_{Gbest} = BP_{best}$
Set $t = t + 1$
Return best bird BP_{Gbest}

3.2 Complexity Analysis

The computational complexity of the DTO algorithm can be expressed as follow. For population n and iterations t_{max} , the time complexity will be defined as follows:

- Initialization $BP_i (i = 1, 2, \dots, n), BV_i (i = 1, 2, \dots, n), T_{max}, c, C_1, C_2, C_3, C_4, C_5, r_1, r_2, R, t = 1: O(1)$.

- Calculate objective function f_n for each bird BP_i : $O(n)$.
- Finding best bird BP_{best} : $O(n)$.
- Updating position of current swimming bird: $O(t_{max} \times n)$.
- Updating velocity of current flying bird: $O(t_{max} \times n)$.
- Updating position of current flying bird: $O(t_{max} \times n)$.
- Calculating objective function f_n for each bird BP_i : $O(t_{max})$.
- Updating c, C_1, C_2, R : $O(t_{max})$.
- Finding best bird BP_{best} : $O(t_{max})$.
- Setting $BP_{Gbest} = BP_{best}$: $O(t_{max})$.
- Setting $t = t + 1$: $O(t_{max})$.
- Producing the best bird BP_{Gbest} : $O(1)$

From this analysis, the complexity of computations is $O(t_{max} \times n)$ and $O(t_{max} \times n \times d)$ with d dimension.

4 Experimental Results

The experiments in this section are explained in two sets. The first set of experiments is designed to evaluate the proposed DTO algorithm performance. The proposed DTO algorithm is tested compared to the algorithms of Particle Swarm Optimization (PSO), Whale Optimization Algorithm (WOA), Grey Wolf Optimizer (GWO), and Genetic Algorithm (GA) based on the seven unimodal benchmark functions [19]. Then, ANOVA and Wilcoxon rank-sum tests are performed to confirm the effectiveness of the proposed algorithm compared to other optimization techniques. The second is experimental in feature selection [20].

4.1 Evaluation of DTO Algorithm Unconstrained Function

Tab. 1 shows a list of the unimodal benchmark function tested in the first experiment. Tab. 1 shows the compared algorithms configuration. To be fair in the comparison, all the algorithms start with 20 agents, same objective function, 100 iterations, same dimensions, and boundaries.

Table 1: Descriptions of unimodal benchmark functions used in our experiments

Benchmark function	D	Range	f_{min}
$f_{01}(x) = \sum_{i=1}^n x^2$	30	[-100, 100]	0
$f_{02}(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[-10, 10]	0
$f_{03}(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	30	[-100, 100]	0
$f_{04}(x) = \max_i \{ x_i , 1 \leq i \leq D\}$	30	[-100, 100]	0
$f_{05}(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-30, 30]	0
$f_{06}(x) = \sum_{i=1}^D ([x_i + 0.5])^2$	30	[-100, 100]	0

(Continued)

Table 1: Continued

Benchmark function	D	Range	f_{min}
$f_{07}(x) = \sum_{i=1}^D ix_i^4 + random[0, 1]$	30	$[-1.28, 1.28]$	0

Fig. 2 shows the convergence curves of the proposed DTO algorithm compared to the PSO, WOA, and GWO algorithms for the benchmark mathematical functions. Note that, the best convergence is achieved by the DTO algorithm. Tab. 2 shows the mean and the standard deviation results based on the benchmark function, F1-F7, for different algorithms.

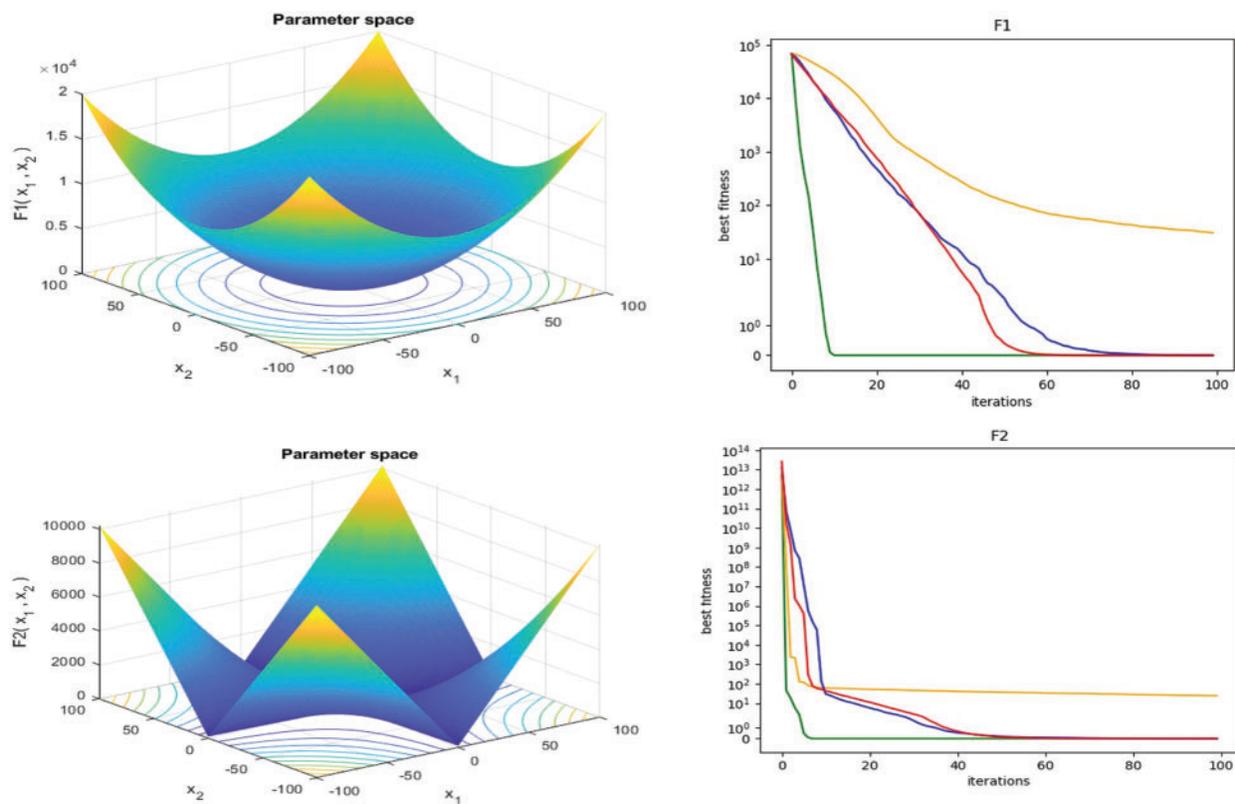


Figure 2: (Continued)

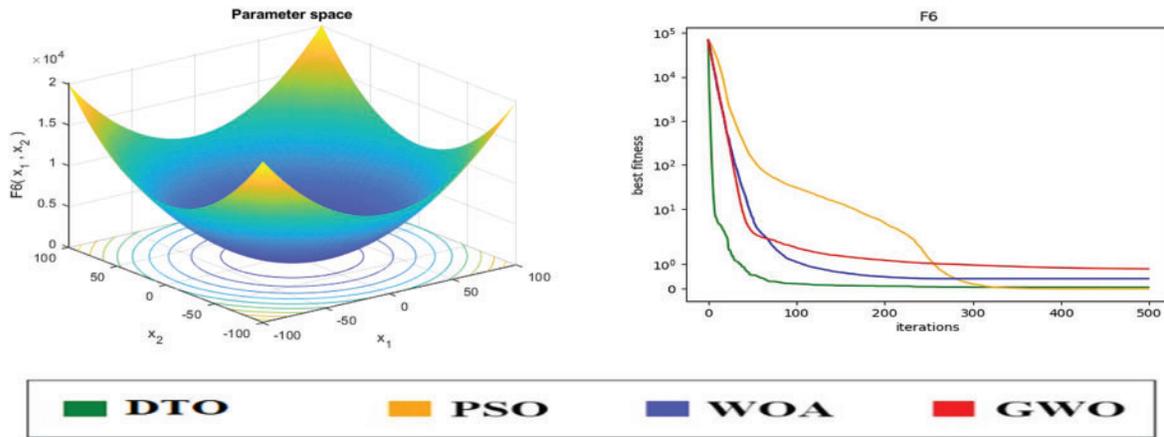


Figure 2: The sample functions' graphical representations and convergence curves

Table 2: Mean and standard deviation results based on the benchmark function F1: F7

Function	Algorithm	DTO	PSO	WOA	GWO	GA
F1	Mean	0	0.000136	1.41E-30	6.59E-28	4.6E-172
	StDev	0	0.000202	4.91E-30	6.34E-05	0
F2	Mean	0	0.042144	1.06E-21	7.18E-17	3.44E-90
	StDev	0	0.045421	2.39E-21	0.029014	6.13E-90
F3	Mean	0	70.12562	5.39E-07	3.29E-06	1.7E-127
	StDev	0	22.11924	2.93E-06	79.14958	8.6E-127
F4	Mean	0	1.086481	0.072581	5.61E-07	1.15E-75
	StDev	0	0.317039	0.39747	1.315088	2.45E-75
F5	Mean	0	96.71832	27.86558	26.81258	28.37287
	StDev	0	60.11559	0.763626	69.90499	0.582802
F6	Mean	0.0001	0.000102	3.116266	0.816579	3.932626
	StDev	0.000121	8.28E-05	0.532429	0.000126	0.431755
F7	Mean	0.0009	0.122854	0.001425	0.002213	0.022992
	StDev	0.001439	0.044957	0.001149	0.100286	0.021966

To test the statistical difference between the proposed DTO algorithm and the compared algorithms, a one-way analysis of variance (ANOVA) test is applied. In this test, two hypotheses are considered; the first is null hypothesis ($H_0: \mu_{DTO} = \mu_{PSO} = \mu_{WOA} = \mu_{GWO} = \mu_{GA}$) and the alternate hypothesis (H_1 : Means are not all equal). The ANOVA test results are shown in Tab. 3. Fig. 3 shows the ANOVA test for proposed and the compared algorithms vs. the objective function. From the results, the alternate hypothesis H_1 is accepted. The ANOVA test confirms the effectiveness of the proposed algorithm compared to other optimization techniques.

Table 3: ANOVA test results based on the benchmark function F1: F7

F1					
ANOVA table	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	2.42E-07	4	6.05E-08	F (4, 145) = 12.04	P < 0.0001
Residual (within columns)	7.29E-07	145	5.03E-09		
Total	9.71E-07	149			
F2					
ANOVA table	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	269.7	4	67.43	F (4, 145) = 7.730	P < 0.0001
Residual (within columns)	1265	145	8.723		
Total	1535	149			
F3					
ANOVA table	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	4.25E+10	4	1.06E+10	F (4, 145) = 186.8	P < 0.0001
Residual (within columns)	8.24E+09	145	56830905		
Total	5.07E+10	149			
F4					
ANOVA table	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	61550	4	15387	F (4, 145) = 92.40	P < 0.0001
Residual (within columns)	24147	145	166.5		
Total	85697	149			
F5					
ANOVA table	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	59376	4	14844	F (4, 145) = 42.46	P < 0.0001
Residual (within columns)	50691	145	349.6		
Total	110067	149			

(Continued)

Table 3: Continued

F6					
ANOVA table	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	355.1	4	88.78	F (4, 145) = 1261	P < 0.0001
Residual (within columns)	10.21	145	0.07039		
Total	365.3	149			
F7					
ANOVA table	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	566.5	4	141.6	F (4, 145) = 16.82	P < 0.0001
Residual (within columns)	1221	145	8.422		
Total	1788	149			

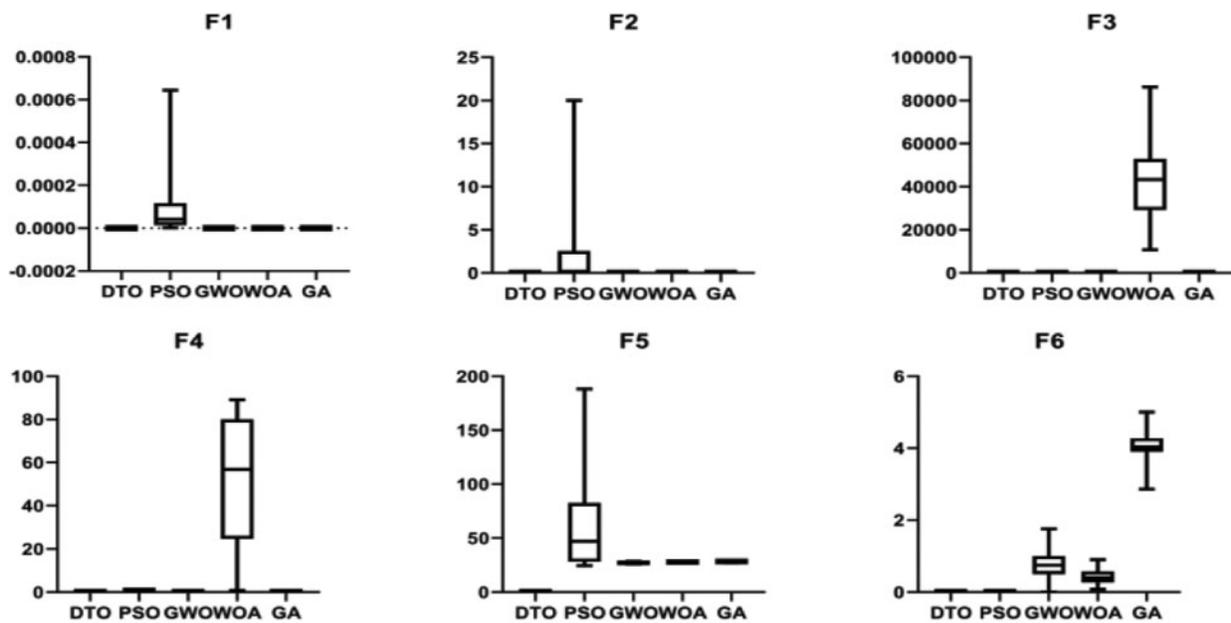


Figure 3: (Continued)

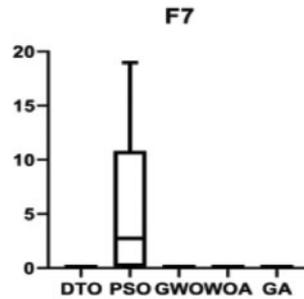


Figure 3: ANOVA test for the proposed DTO and compared algorithms based on the benchmark functions

Wilcoxon’s rank-sum test is employed between every two algorithms to get the p-values between the proposed DTO algorithm and other algorithms to show that DTO has a significant difference. The two hypotheses are the null hypothesis ($H_0: \mu_{DTO} = \mu_{PSO}, \mu_{DTO} = \mu_{GWO}, \mu_{DTO} = \mu_{WOA}, \text{ and } \mu_{DTO} = \mu_{GA}$) and the alternate hypothesis (H_1 : Means are not all equal). Tab. 4 shows that the p-values are less than 0.05 between the proposed algorithm and other algorithms. This confirms the superiority of the DTO algorithm and that it is statistically significant. Thus, the alternate hypothesis H_1 is accepted.

Table 4: Wilcoxon’s rank-sum test results based on the benchmark function F1: F7

	DTO vs. PSO	DTO vs. GWO	DTO vs. WOA	DTO vs. GA
F1	1.12E−05	1.12E−05	1.12E−05	1.12E−05
F2	1.12E−05	1.12E−05	1.12E−05	1.12E−05
F3	1.12E−05	1.12E−05	1.12E−05	1.12E−05
F4	1.12E−05	1.32E−05	1.12E−05	1.42E−05
F5	1.12E−05	1.12E−05	1.12E−05	1.12E−05
F6	1.12E−05	1.12E−05	1.22E−05	1.12E−05
F7	1.12E−05	1.12E−05	1.12E−05	1.12E−05

4.2 Evaluation of DTO Algorithm on Feature Selection Problem

In the case of the feature selection issue, the output solution should be modified from a continuous solution to a binary solution utilizing the numbers 0 or 1. This function is often used to convert the continuous solution of an optimizer to a binary solution in an optimization problem.

$$X_d^{(t+1)} = \begin{cases} 0 & \text{if } Sigmoid(X_{Best}) < 0.5 \\ 1 & \text{otherwise} \end{cases},$$

$$Sigmoid(X_{Best}) = \frac{1}{1 + e^{-10(X_{Best}-0.5)}} \tag{9}$$

4.2.1 Fitness Function

The quality of an optimizer's solutions is determined by the fitness function that has been given to it. The error rate of classification and regression, as well as the features that have been picked from the input dataset, are the primary determinants of the function. It is advisable to choose a solution based on the collection of characteristics that can provide the bare minimum of features with the lowest classification error rate. The following equation is used in this study to evaluate the quality of the solutions provided.

$$F_n = \alpha \text{Err}(O) + \beta \frac{|s|}{|f|} \quad (10)$$

As part of the experiments and comparative findings, the DTO evaluated our proposed algorithm against six datasets from the UCI machine learning library to determine its effectiveness. The datasets were chosen because they had a diverse range of features and occurrences that were reflective of the many types of problems that the proposed approach would be evaluated against. For more details about the UCI datasets, refer to [Tab. 5](#).

Algorithm 2: The bDTO Algorithm

Initialization DTO Algorithm configuration, including population and parameters

Change solutions to binary (0 or 1)

Evaluate fitness function

While $t \leq \text{iters}_{\max}$ **do**

Calculate objective function f_n for each bird BP_i

Find best bird BP_{best}

While $t \leq T_{max}$ **do**

Apply DTO Algorithm

Convert solutions to binary (0 or 1) using [Eq. \(9\)](#)

Update parameters and best solution

end while

Return best solution

Table 5: Datasets description

Data-set name	Number of rows	Number of columns (Features)
Australian	690	14
Breast_Cancer	699	8
WaveformEW	5000	21
Towonorm	7400	20
Seeds	210	7
Ring	7400	20

4.2.2 Evaluation Metrics

The evaluation metrics used in this research are presented in [Tab. 6](#) as follows:

Table 6: Evaluation metrics

Metrics	
Average error	$AvgPref = \frac{1}{M} \sum_{j=1}^M \frac{1}{N} \sum_{i=1}^N mse(C_i, L_i)$
Best fitness	$best = \min_{i=1}^M g_*^i$
Worst fitness	$worst = \max_{i=1}^M g_*^i$
Average fitness size	$AVG Selection Size = \frac{1}{M} \sum_{i=1}^M \frac{size(g_*^i)}{D}$
Mean	$Mean = \frac{1}{M} \sum_M g_*^i$
Std (standard deviation)	$Std = \sqrt{\frac{1}{M-1} \sum (g_1^i - Mean)^2}$

The average error of several algorithms is shown in [Tab. 7](#). Due to the decreased error, the optimizer has identified the optimal collection of features that can train the classifier while also producing a smaller error on the concealed test data. [Tab. 8](#) below shows the average features selected. [Tab. 9](#) for average fitness, [Tab. 10](#) for standard deviation fitness, [Tab. 11](#) for best fitness, and [Tab. 12](#) for worst fitness. The DTO has been able to find the superiority fitness for all datasets.

Table 7: Average error

Datasets	bDTO	bGWO	bGA	bWOA	bPSO
Australian	0.152173913	0.155072464	0.157971014	0.168115942	0.156521739
Breast cancer	0.025751073	0.028612303	0.027181688	0.025751073	0.027181688
WaveformEW	0.116809117	0.131054131	0.122507123	0.131054131	0.11965812
Twonorm	0.0324412	0.044201135	0.057853474	0.0324412	0.062719654
Seeds	0.2	0.2	0.2	0.2	0.2
Ring	0.15963774	0.16044877	0.162341173	0.163557718	0.161665315

Table 8: Average select size

Datasets	bDTO	bGWO	bGA	bWOA	bPSO
Australian	0.452380952	0.619047619	0.571428571	0.523809524	0.5
Breast cancer	0.75	0.708333333	0.791666667	0.75	0.625
WaveformEW	0.343434343	0.454545455	0.444444444	0.606060606	0.404040404
Twonorm	0.7	0.85	0.766666667	1	1
Seeds	0.428571429	0.571428571	0.523809524	0.619047619	0.571428571
Ring	0.316666667	0.35	0.316666667	0.333333333	0.316666667

Table 9: Average fitness

Datasets	bDTO	bGWO	bGA	bWOA	bPSO
Australian	0.152173913	0.155072464	0.157971014	0.168115942	0.156521739
Breast cancer	0.025751073	0.028612303	0.027181688	0.025751073	0.027181688
WaveformEW	0.116809117	0.131054131	0.122507123	0.131054131	0.11965812
Twonorm	0.0324412	0.044201135	0.057853474	0.0324412	0.062719654
Seeds	0.2	0.2	0.2	0.2	0.2
Ring	0.15963774	0.16044877	0.162341173	0.163557718	0.161665315

Table 10: Standard deviation fitness

Datasets	bDTO	bGWO	bGA	bWOA	bPSO
Australian	0.006641414	0.018101443	0.008695652	0.010040874	0.011503267
Breast cancer	0.002477898	0.004291845	0.002477898	0.004291845	0.006555902
WaveformEW	0.004934618	0.0215095	0.027474788	0.013055771	0.022613259
Twonorm	0	0.003535198	0.00529245	0	0.014424787
Seeds	0	0	0	0	0
Ring	0.004102187	0.010800208	0.011769254	0.007611737	0.008567154

Table 11: Best fitness

Datasets	bDTO	bGWO	bGA	bWOA	bPSO
Australian	0.134782609	0.152173913	0.143478261	0.156521739	0.147826087
Breast cancer	0.021459227	0.025751073	0.025751073	0.021459227	0.021459227
WaveformEW	0.085470085	0.111111111	0.11965812	0.11965812	0.094017094
Twonorm	0.0324412	0.040145985	0.051905921	0.0324412	0.04703974
Seeds	0.2	0.2	0.2	0.2	0.2
Ring	0.151257097	0.156934307	0.152068127	0.158961882	0.154095702

Table 12: Worst fitness

Datasets	bDTO	bGWO	bGA	bWOA	bPSO
Australian	0.165217391	0.169565217	0.160869565	0.173913043	0.169565217
Breast cancer	0.030042918	0.030042918	0.030042918	0.030042918	0.034334764
WaveformEW	0.136752137	0.153846154	0.128205128	0.145299145	0.136752137
Twonorm	0.0324412	0.046634225	0.062043796	0.0324412	0.075425791
Seeds	0.1982	0.2	0.2	0.2	0.2
Ring	0.164233577	0.172343877	0.175182482	0.172343877	0.169505272

5 Conclusion

In this paper, a novel Dipper Throated Optimization (DTO) is introduced which is inspired by the throated dipper bird. Six UCI machine learning database datasets and unconstrained function are used to prove the consistency of the suggested optimizer and guarantee that the proposed solution is dependable and stable to evaluate its quality and effectiveness. ANOVA and Wilcoxon rank-sum tests are used to compare the proposed algorithm to different optimization methods. The results showed clearly that, DTO outperformed all other compared methods. For future work, DTO needs more investigation by applying it to solve other well-known real-world combinatorial optimization problems. Based on the great success of DTO, the researchers can investigate the hybridizations between DTO and other metaheuristic optimization algorithms as it is proved that hyper metaheuristic algorithms perform well compared to single metaheuristic algorithms.

Acknowledgement: The researcher would like to thank the Deanship of Scientific Research, Qassim University for funding the publication of this project.

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

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

References

- [1] A. Ibrahim, H. A. Ali, M. M. Eid and E. -S. M. El-kenawy, "Chaotic harris hawks optimization for unconstrained function optimization," in *16th Int. Computer Engineering Conf. (ICENCO)*, Cairo, Egypt, pp. 153–158, 2020.
- [2] M. M. Eid, E. -S. M. El-kenawy and A. Ibrahim, "A binary sine cosine-modified whale optimization algorithm for feature selection," in *National Computing Colleges Conf. (NCCC)*, Taif, Saudi Arabia, pp. 1–6, 2021.
- [3] E. -S. M. El-Kenawy, S. Mirjalili, A. Ibrahim, M. Alrahmawy, M. El-Said *et al.*, "Advanced meta-heuristics, convolutional neural networks, and feature selectors for efficient COVID-19 X-ray chest image classification," *IEEE Access*, vol. 9, pp. 36019–36037, 2021.
- [4] E. S. M. El-Kenawy, S. Mirjalili, S. S. M. Ghoneim, M. M. Eid, M. El-Said *et al.*, "Advanced ensemble model for solar radiation forecasting using sine cosine algorithm and Newton's laws," *IEEE Access*, vol. 9, pp. 115750–115765, 2021.
- [5] A. Ibrahim, S. Mirjalili, M. El-Said, S. S. M. Ghoneim, M. Alharthi *et al.*, "Wind speed ensemble forecasting based on deep learning using adaptive dynamic optimization algorithm," *IEEE Access*, vol. 9, pp. 125787–125804, 2021.
- [6] A. A. Salamai, E. -S. M. El-kenawy and A. Ibrahim, "Dynamic voting classifier for risk identification in supply chain 4.0," *Computers, Materials & Continua*, vol. 69, no. 3, pp. 3749–3766, 2021.
- [7] E. -S. M. El-Kenawy, M. M. Eid, M. Saber and A. Ibrahim, "MbGWO-SFS: Modified binary grey wolf optimizer based on stochastic fractal search for feature selection," *IEEE Access*, vol. 8, pp. 107635–107649, 2020.
- [8] E. -S. M. El-kenawy, H. F. Abutarboush, A. W. Mohamed and A. Ibrahim, "Advance artificial intelligence technique for designing double T-shaped monopole antenna," *Computers, Materials & Continua*, vol. 69, no. 3, pp. 2983–2995, 2021.
- [9] M. M. Fouad, A. I. El-Desouky, R. Al-Hajj and E. -S. M. El-Kenawy, "Dynamic group-based cooperative optimization algorithm," *IEEE Access*, vol. 8, pp. 148378–148403, 2020.
- [10] E. -S. M. El-Kenawy, A. Ibrahim, S. Mirjalili, M. M. Eid and S. E. Hussein, "Novel feature selection and voting classifier algorithms for COVID-19 classification in CT images," *IEEE Access*, vol. 8, pp. 179317–179335, 2020.

- [11] A. Ibrahim, M. Noshay, H. A. Ali and M. Badawy, "PAPSO: A power-aware VM placement technique based on particle swarm optimization," *IEEE Access*, vol. 8, pp. 81747–81764, 2020.
- [12] E. M. Hassib, A. I. El-Desouky, L. M. Labib and E. -S. M. T. El-Kenawy, "Woa + brnn: An imbalanced big data classification framework using whale optimization and deep neural network," *Soft Computing*, vol. 24, pp. 5573–5592, 2020.
- [13] S. Mirjalili, "Sca: A sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.
- [14] J. Hao and C. Solnon, "Meta-heuristics and artificial intelligence," in P. Marquis, O. Papini, H. Prade, (Eds.), *A Guided Tour of Artificial Intelligence Research: AI Algorithms*, 1st ed., vol. 2. Cham, Denmark: Springer, pp. 27–52, 2020.
- [15] M. Ahmadi, K. Kazemi, A. Aarabi, T. Niknam and M. S. Helfroush, "Image segmentation using multilevel thresholding based on modified bird mating optimization," *Multimedia Tools Applications*, vol. 78, pp. 23003–23027, 2019.
- [16] O. Mohammadrezapour, I. YoosefDoost and M. Ebrahimi, "Cuckoo optimization algorithm in optimal water allocation and crop planning under various weather conditions (case study: Qazvin plain, Iran)," *Neural Computing and Applications*, vol. 31, pp. 1879–1892, 2017.
- [17] M. Hadwan, M. Ayob, N. R. Sabar and R. Qu, "A harmony search algorithm for nurse rostering problems," *Information Sciences*, vol. 233, pp. 126–140, 2013.
- [18] K. B. O. Medani, S. A. Sayah and A. Bekrar, "Whale optimization algorithm based optimal reactive power dispatch: A case study of the Algerian power system," *Electric Power Systems Research*, vol. 163, pp. 696–705, 2018.
- [19] D. Gupta, S. Sundaram, A. Khanna, A. E. Hassanien, and V. H. C. Albuquerque, "Improved diagnosis of Parkinson's disease using optimized crow search algorithm," *Computer and Electrical Engineering*, vol. 68, pp. 412–424, 2018.
- [20] K. Hussain, M. N. Mohd Salleh and S. Cheng, "Metaheuristic research: A comprehensive survey," *Artificial Intelligence Reviews*, vol. 52, pp. 2191–2233, 2019.
- [21] E. N. Dragoi and V. Dafinescu, "Review of metaheuristics inspired from the animal kingdom," *Mathematics*, vol. 9, no. 18, pp. 1–52, 2021.
- [22] R. Rajakumar, P. Dhavachelvan and T. Vengattaraman, "A survey on nature inspired meta-heuristic algorithms with its domain specifications," in *2016 Int. Conf. on Communication and Electronics Systems (ICCES)*, Coimbatore, India, pp. 1–6, 2016.
- [23] T. El-Ghazali, "Machine learning into metaheuristics: A survey and taxonomy of data-driven metaheuristics," *ACM Computer Survey*, vol. 54, pp. 1–32, 2020.
- [24] C. Min-Yuan and D. Prayogo, "Symbiotic organisms search: A new metaheuristic optimization algorithm," *Computers and Structures*, vol. 139, pp. 98–112, 2014.
- [25] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, 1st ed., vol. 284. Berlin, Heidelberg: Springer, pp. 65–74, 2010.
- [26] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, vol. 22, pp. 52–67, 2002.
- [27] E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, "Gsa: A gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.
- [28] X. -S. Yang, "Firefly algorithm, stochastic test functions and design optimization," arXiv preprint arXiv:1003.1409, 2010.
- [29] A. H. Gandomi and A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831–4845, 2012.
- [30] E. -S. El-Kenawy and M. Eid, "Hybrid gray wolf and particle swarm optimization for feature selection," *International Journal of Innovative Computing, Information and Control*, vol. 16, no. 3, pp. 831–844, 2020.
- [31] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

- [32] X. -S. Yang and S. Deb, "Cuckoo search via lévy flights," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, IEEE, Coimbatore, India, pp. 210–214, 2009.
- [33] M. Hadwan, M. Ayob, M. A. Rassam and E. A. Hezam, "Deluge harmony search algorithm for nurse rostering problems," in *2019 First Int. Conf. of Intelligent Computing and Engineering (ICOICE)*, Hadhramout, Yemen, pp. 1–5, 2019.
- [34] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [35] E. Cuevas, M. Cienfuegos, D. Zaldívar and M. Pérez-Cisneros, "A swarm optimization algorithm inspired in the behavior of the social-spider," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6374–6384, 2013.
- [36] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.
- [37] P. Agrawal, H. F. Abutarboush, T. Ganesh and A. W. Mohamed, "Metaheuristic algorithms on feature selection: A survey of one decade of research (2009–2019)," *IEEE Access*, vol. 9, pp. 26766–26791, 2021.
- [38] B. Xue, M. Zhang, W. N. Browne and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Transaction on Evolutionary Computing*, vol. 20, no. 4, pp. 606–626, 2016.
- [39] P. Y. Lee, W. P. Loh and J. F. Chin, "Feature selection in multimedia: The state-of-the-art review," *Image and Vision Computing*, vol. 67, pp. 29–42, 2017.
- [40] M. Sharma and P. Kaur, "A comprehensive analysis of nature-inspired meta-heuristic techniques for feature selection problem," *Archives of Computational Methods in Engineering*, vol. 28, pp. 1–25, 2020.