IJIO
(*International Journal of Industrial Optimization*)
ISSN 2723-3022 (online) | 2714-6006 (printed)
VOL 4, No. 2, 115-130
https://doi.org/10.12928/ijio.v4i2.7651

A new health-based metaheuristic algorithm: cholesterol algorithm

Serap Ulusam Seçkiner *, Şeyma Yilkici Yüzügüldü
Department of Industrial Engineering, Gaziantep University, Turkey
*Corresponding Author: seckiner@gantep.edu.tr

ARTICLE INFO

Article history
Received: February 9, 2023
Revised: March 30, 2023
Accepted: June 25, 2023

Keywords

Cholesterol Algorithm;
Optimization;
Meta-heuristics;
Continuous functions.

ABSTRACT

High levels of cholesterol can result in the hardening of arteries and the formation of fat deposits known as cholesterol plaques in blood vessels. These deposits can obstruct blood flow, leading to various health issues over time. Therefore, this research is proposed to explore the effectiveness of a new health-based metaheuristic algorithm inspired by the cholesterol metabolism of the human body. In the study, the main idea is to focus on the cholesterol algorithm's performance on unconstrained continuous optimization problems. The performances of the proposed cholesterol algorithm are evaluated based on 23 comparison tests and results were compared with Particle Swarm Optimization, Genetic Algorithm, Grey Wolf Optimization, Whale Optimization Algorithm, Harris Hawks Optimization, Differential Evolution, FireFly Algorithm, Cuckoo Search, Multi-Verse Optimizer, and JAYA algorithms. Results showed that this novel cholesterol algorithm implementation could compete effectively with the best-known solution to test functions. The CA algorithm was tested only in continuous problems and did well in simple functions but it had some issues with convergence and exploration-exploitation control. Despite these problems, it showed promise by finding good results quickly in continuous optimization problems. Future research will test the CA algorithm in different problem types and real-world applications to improve its effectiveness. Only a few algorithms like ANN and AIS have been inspired by the health field. This study presents a novel health-based algorithm that utilizes the levels of cholesterol, a crucial element in the human body.

This is an open access article under the CC–BY-SA license.

INTRODUCTION

An optimizer generally finds solution techniques to relatively unknown mathematical functions, which are derived from real-world problems. Through algorithmic approaches, she/he arrives at the research problem and asks, 'How do these functions are solved?' Optimization is as old as the history of the universe. It can be seen anywhere from the farthest point of the universe to the atomic particle. Optimization can sometimes be seen in an engineering design and sometimes in an ant's search for food. The main purpose of all is to find the best possible result/s under certain conditions. The universe, nature, and animals solve their problems easily by instinct. For this reason, people try to solve complex problems by imitating the universe and nature. They developed metaheuristics to solve complex optimization problems. Metaheuristics are a general-purpose heuristic method toward promising regions of the search space containing high-quality solutions [1]. They aim to find the global best solution with the minimum time and effort. Many algorithms in the literature are inspired by animals and nature. For example, the inspirations of the algorithms used as compared to algorithms in this study are as follows: Genetic Algorithm [2] and Differential Evolution [3] were inspired by the principles of survival of the bests. Particle Swarm Optimization [4] was inspired by the social behavior of bird flocking or fish schooling. Cuckoo Search [5], Firefly Algorithm [6], Whale Optimization Algorithm [7], Harris Hawks Optimization [8], and Grey Wolf Optimization [9] were inspired by behaviors of cuckoos, fireflies, humpback whales, Harris Hawks birds, and grey wolves, respectively. Multiverse Optimization [10] was inspired by three concepts in cosmology a white hole, a black hole, and a wormhole. Also, JAYA [11] was inspired by a move toward the best solution and avoiding the worst solution.

The number of algorithms inspired by the human body such as artificial neural networks and artificial immune systems is very few. The cholesterol Algorithm (CA) is a new approach to a problem-solving tool that takes inspiration from the cholesterol structure found in the body tissues. To the best of our knowledge, this is the first study that shows how to optimize continuous functions inspired by biochemical processes of the human body. Before presenting CA, we need to understand the cholesterol mechanism in the human body.

Cholesterol is a lipid that is a waxy lipid that is naturally produced by all cells in the blood of people and the liver. Cholesterol has a waxy texture found in the blood [12]. For the human body to be formed healthy, it needs a certain amount of cholesterol to be used in the formation of hormones and D vitamins as well as cell membranes [13]. However, cholesterol is insoluble in water, so it cannot pass into the blood on its own and be distributed throughout the body alone. Lipoproteins are produced by the liver to help transport cholesterol [14]. Cholesterol-carrying lipoproteins in the human body can be divided into two groups low-density lipoprotein, LDL, and high-density lipoprotein, HDL. While LDL carries cholesterol in the bloodstream, HDL carries cholesterol within the liver and tissues. When high levels of cholesterol levels can lead to the hardening of the arteries and the development of fat deposits in the blood vessels. These deposits are also called cholesterol plaques. Together over time, these deposits make it difficult to flow back blood from the veins and cause various health problems. LDL is also called "bad cholesterol". Its job is to carry cholesterol and triglycerides, the type of fat the body uses for energy, in the arteries. If an individual's LDL cholesterol level is too high, it can build up on the walls of the arteries over time. HDL is also called "good cholesterol". It helps LDL cholesterol go back to the liver to be removed from the body. In this way, it prevents the accumulation of cholesterol plaques in the arteries. Triglycerides are another type of lipid that is different from cholesterol [14]. While the human body uses cholesterol to form cell walls, certain hormones, and vitamin D, it uses triglycerides as an energy source. Total cholesterol (TC) is the sum of HDL, LDL, and one-fifth of triglycerides (Trig) [15]. Total cholesterol is calculated by using Equation (1).

$$TC = HDL + LDL + Trig/5 \tag{1}$$

Cholesterol levels in the human body [15] are shown in Table 1. Cholesterol levels that should be in a healthy person are HDL 60 and above, LDL less than 100, Triglycerides less than 150, and Total Cholesterol less than 200. Otherwise, cardiovascular diseases are seen [16].

Table 1. Cholesterol levels in a human body

	Low	Good	High
HDL	Men: Less than 40 Women: Less than 50	60 or higher	n/a
LDL	n/a	Less than 100	Greater than 160
Triglycerides	n/a	Less than 150	Greater than 200
Total Cholesterol	n/a	Less than 200	240 or higher

 10.12928/ijio.v4i2.7651

The objective of this research is to introduce a cholesterol optimization algorithm and to show its most notable continuous function optimizations. The main contribution is to develop a novel algorithm to solve mathematical functions inspired by the state of being healthy. Also, Cholesterol Algorithm is contributing to the world of metaheuristic algorithms by offering an alternative solution approach.

The organization of this paper is as follows: Section II presents the steps of the CA algorithm, Section III presents the experimental design of CA, Section IV presents the results of CA with the most notable theoretical results of some other meta-heuristics, and Section V highlights some current hot research topics and concludes the article.

METHOD

1. Cholesterol Algorithm (CA)

CA takes inspiration from the biochemical processes of the human body. HDL, LDL, and Triglycerides increase and decrease within the body. When LDL rises too high, the vessel starts to block. In the meantime, HDL has a protective feature. When the level of triglyceride increases, the risk of vascular occlusion increases. In that case, the blocking of a vessel may indicate a bad solution for the optimization problem, and the openness of the vessel may indicate a good solution. If good status at HDL and LDL levels may represent the local or global best condition. In the optimization problem, at the same time, having HDL and LDL ratios at a certain level will lead to good solutions, and a high-level solution will deteriorate. Where pheromone hormone accumulates in good solutions in the ant colony algorithm, we will assume that we are in a good or bad solution according to the algorithm HDL, LDL, and non-steroidal lipid Triglyceride accumulation in the vessel. All indicators that show that the vessel is healthy will inspire the solution of an optimization problem. When these human body vessels deposit steroid lipids, blood flow is restricted, and favorable vessels (path) should be searched in other healthy vessels. CA exploits a similar mechanism for solving optimization problems.

In CA, each optimization problem is considered as a human body. The transport of nutrients and oxygen between tissues in the human body occurs through vessels. In an optimization problem, the quality of each solution candidate is considered as cholesterol status in each vessel. After the function values of the solution candidates are calculated, the quality of the solution candidate or cholesterol levels in the vessel is obtained by using the worst and best function values.

Cholesterol levels of each solution are calculated by using Equation (2) - Equation (5). Function values are transformed cholesterol levels through these equations. Cholesterol levels in a person should be within certain ranges. So the level of HDL, good cholesterol, should be above 40, and the level of LDL, bad cholesterol, should be below 160. These levels and the good-bad cholesterol relationship are two of the issues that the CA algorithm is inspired by cholesterol metabolism. The CA algorithm uses ranges to determine the quality of the solution and decides whether solutions are good or bad relative to these ranges.

The cholesterol ranges to be used in the CA algorithm is [$HDL_{LowerBound}$, 2*$HDL_{LowerBound}$] for the range of HDL, [0, $LDL_{UpperBound}$] for the range of LDL, [0, $Triglycerides_{UpperBound}$] for the range of triglycerides. In this study, it is preferred to use the real cholesterol levels in the human body in Table 1. In other words, 40 for $HDL_{LowerBound}$, 160 for $LDL_{UpperBound}$, and 200 for $Trigliycerides_{UpperBound}$ were used. So the cholesterol ranges of the CA algorithm are [40, 80] for the range of HDL, [0, 160] for the range of LDL, and [0, 200] for the range of Triglycerides. It is more important to use only these ranges for determining levels of HDL, LDL, and Triglycerides in the proposed CA algorithm. In other words, according to the results obtained from the observations, levels of HDL, LDL, and Triglycerides in the proposed CA algorithm can be determined at any level as long as under ranges. A range determination is necessary as it constitutes the working principle of the CA algorithm.

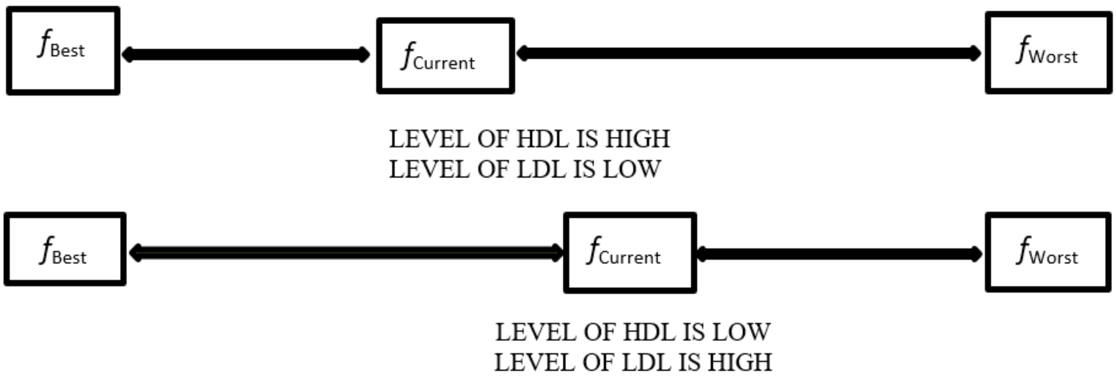
$$HDL = HDL_{LowerBound} + HDL_{LowerBound} * (f_{Current} - f_{Worst})/(f_{Worst} - f_{Best}) \quad (2)$$

$$LDL = LDL_{UpperBound} * (f_{Best} - f_{Current}) / (f_{Worst} - f_{Best}) \quad (3)$$

$$Triglycerides = Triglycerides_{UpperBound} * random(0,1) \quad (4)$$

$$Total\ Cholesterol = HDL + LDL + Triglycerides/5 \quad (5)$$

In Figure 1, the relationship between the distance of the current solution from the best-worst solutions and the HDL-LDL levels has been shown. That is, the closer the current solution is to the best-known solution, the higher the HDL value and the lower the LDL value. Another of the issues that the CA algorithm is inspired by cholesterol metabolism is that levels of lipoprotein affect the blood flow through the vessel. If the level of LDL is high in the vessel, there is a blockage in the vessel because there is excess fat in the vessel. The amount of HDL reduces the amount of fat in the vessel. If we adopt this situation to an optimization problem, the amount of HDL increases when the solution gets better, and the amount of LDL increases when it gets worse. When levels of HDL and LDL in the vessel, in the solution path, go beyond the predetermined ranges, the vessel is blocked. Vascular occlusion shows that there are no good solutions in the search area of solution space, so this solution path is blocked. These solutions are forgotten because optimization problems are focused on maximization or minimization. The search continues with the neighboring solutions of the best solution found instead of them. The exploitation process in the optimization algorithms is performed in this way in CA. As long as cholesterol levels continue to be within specified limits, blood flow in the vessel continues. This situation shows that the searched solution space is suitable for exploration and that the search will continue in this solution space until occlusion occurs.



.

Figure 1. Relationship of function values and levels of HDL and LDL

New solutions are obtained based on the cholesterol levels of current solutions or best solutions. According to Equation (6) – Equation (9) and Figure 2, if the amount of LDL exceeds $LDL_{GoodLevel}$ and the amount of HDL falls below $HDL_{GoodLevel}$, this solution is bad and is forgotten due to the blockage of this solution (vessel). The neighboring solutions of the best solution are focused on improving the best solution. If the cholesterol level of the solution is within the desired range, this solution is good and the search continues in the area where the solution is located. $Total\_Chol_{GoodLevel}$, $LDL_{GoodLevel}$, and $HDL_{GoodLevel}$ must be in the range of Total_Chol, LDL, and HDL, respectively. In this study, values in the 'Good' Column of Table 1 were used for these values. Total_CholGoodLevel, $LDL_{GoodLevel}$, and HDLGoodLevel were taken as 200, 100, and 60, respectively.

In Figure 2, Total Cholesterol, LDL, and HDL are used in the decision-making of cholesterol levels in the CA algorithm. Since total cholesterol consists of the sum of LDL, HDL, and triglycerides (see Equation (1)), the number of triglycerides indirectly affects total cholesterol. Therefore, it was not considered necessary to use triglycerides in decision-making. CA algorithm is focused on solving minimization problems. To solve maximization problems, if the objective function of the problem is $f(x)$ and the constraints of it are $g(x) \leq 0$, they are taken as $- f(x)$, $-g(x) \geq 0$ respectively.

 10.12928/ijio.v4i2.7651

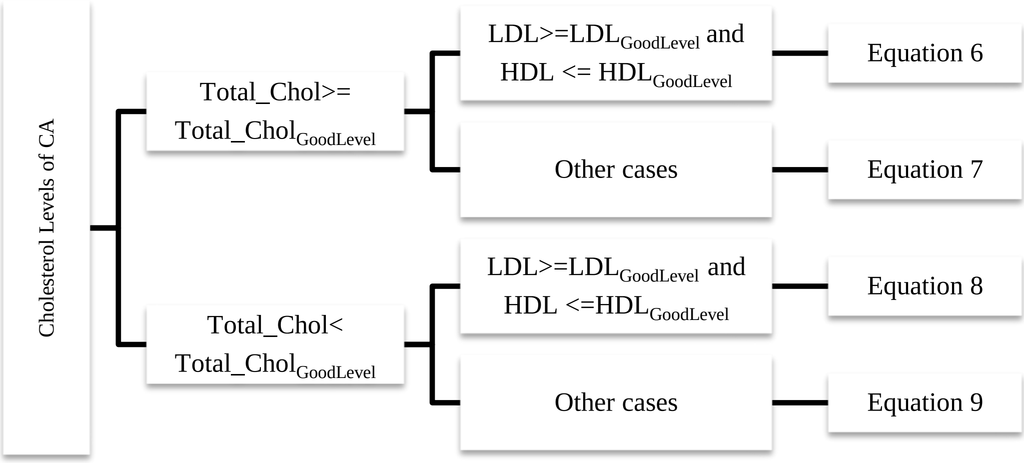


Figure 2. Cholesterol levels of CA

$$x_{new} = x_{best} - x_{best} * Trig\ of\ x_{best} * random(-1,1) \tag{6}$$

$$x_{new} = x_{current} - (x_{current} * LDL\ of\ x_{current} / HDL\ of\ x_{current} - Trig\ of\ x_{current}) \tag{7}$$

$$x_{new} = x_{best} + x_{best} * Trig\ of\ x_{best} * random(-1,1) \tag{8}$$

$$x_{new} = x_{current} + (x_{current} * LDL\ of\ x_{current} / HDL\ of\ x_{current} - Trig\ of\ x_{current}) \tag{9}$$

The process of CA is as follows:

Step 1: The initial solution set is randomly generated in uniform distribution (see Equation (10)) and the function values are calculated (see Equation (11)).

$$x = lowerbound + (upperbound - lowerbound) * Random \tag{10}$$

$$Function\ Value = f(x) \tag{11}$$

Step 2: The solution set has the best and worst values according to function values. Cholesterol levels are calculated according to these values, which show the quality of the solution. The equations for calculating the cholesterol levels of the problem are presented in Equations (2) – Equation (5).
Step 3: The new sets of solutions are calculated based on cholesterol levels according to Figure 2. Function values of these solution sets are calculated.
Step 4: Step 2 proceeds until the specified stopping criterion is terminated. Steps 2 and Step 3 are repeated. Figure 3 presents the pseudocode of our algorithm. For the flowchart of CA can be shown in Figure 4.

2. The experimental design of CA

All tests and performance analyze were performed using a computer with Intel Core i5 10th Gen, 1.60GHz processor, 8GB Ram, and Microsoft Windows 10 operating system. CA was coded in Python, the most popular programming language of recent times. CA and 10 other algorithms (Particle Swarm Optimization, Grey Wolf Optimization, Whale Optimization Algorithm, Harris Hawks Optimization, Differential Evolution, FireFly Algorithm, Cuckoo Search, Multi-Verse Optimizer, Genetic Algorithm, and JAYA) were run in 23 benchmark tests using Evolopy Framework [17,18]. This framework allows us to test functions at one time. Test functions with the name of the function, equation, and variable search range, dimension, and the best-known global optimum value are described in Table 2.

```
Set the termination criteria and vessel number
Generate the initial solution set in each vessel randomly and calculate the function values of the
initial solution set
For(termination criteria)
        Determine local_best, global_best, local_worst
        For(vessels)
                Calculate the cholesterol levels of each vessel
                Update solution set
                Calculate function values of the new solution set
        End for
End For
```

Figure 3. Pseudocode of CA

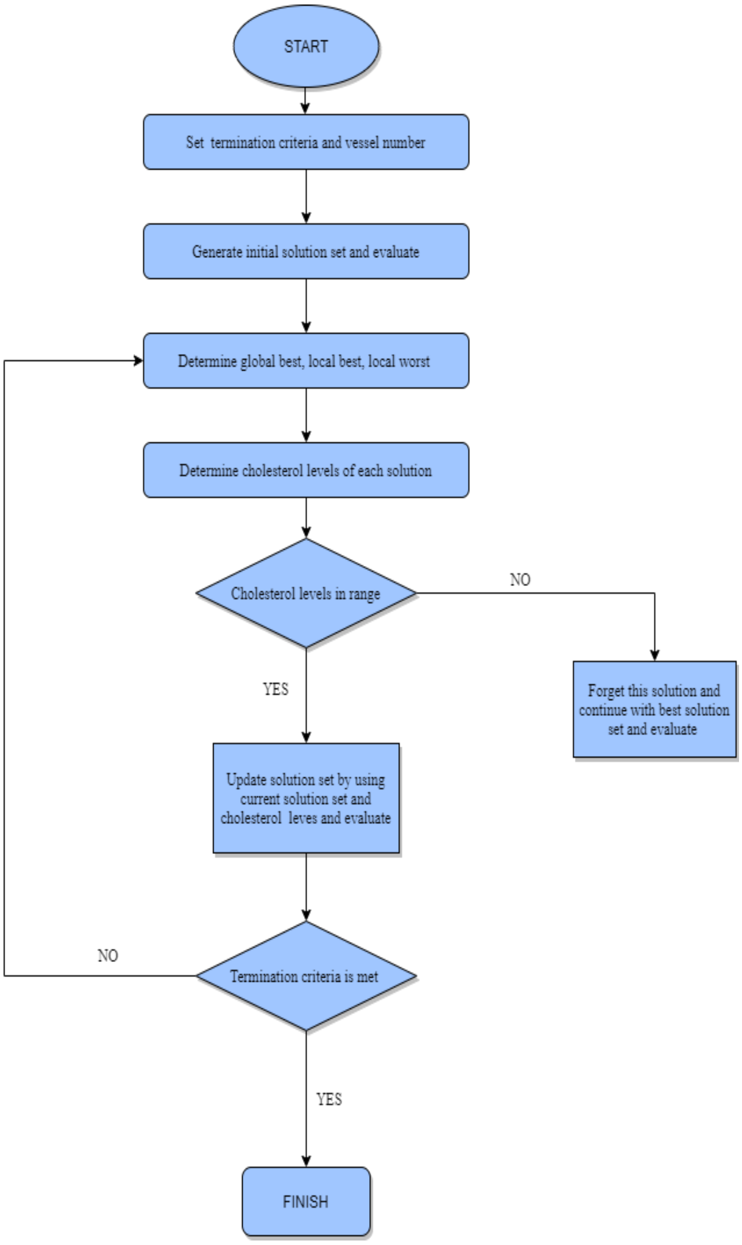


Figure 4. Flowchart of CA

10.12928/ijio.v4i2.7651

In this study, before analyzing the performance of the algorithm, all tests worked under equal conditions as follows: (1) compared algorithms had been applied to the same problems, 23 benchmark tests, (2) the properties of the applied problems were determined as the same in all tests, (3) compared algorithms were run 500 iteration numbers as stopping criteria, (4) the numbers of algorithms were taken 30 runs independently of each other, (5) initial candidate solutions were started randomly between the lower and upper limits specified for each function, (6) the population number was taken as 50, and (6) algorithms' specific parameters were kept independent from problems.

Compared algorithms contained in the Evolopy Framework were used. They are PSO, FFA, GWO, WOA, MVO, CS, HHO, JAYA, DE, and GA. In GA, the crossover probability is 0.8, the mutation probability is 0.05, elitism ratio is 0.5. In FFA, α is 0.5 as randomness, the minimum value of β (the variation of attractiveness) is 0.20, and the absorption coefficient is 1. In the CS algorithm, the discovery rate of alien eggs/solutions is 0.25. In MWO, the wormhole existence probability (WEP) is between 0.2 and 1. The parameters of PSO are v_{max} is 6, w_{max} is 0.9, w_{Min} is 0.2, and c_1 and c_2 are 2.

The quality of the solutions found by CA and other compared algorithms was applied to evaluate statistical results, such as the best solution, mean, and standard deviation values achieved by algorithms in all studies used. Furthermore, Mean Square Error (MSE) and Root Mean Square Error (RMSE) had been calculated by using the differences between the best-known solutions and the best solutions found. These values were calculated by Equation (12) and Equation (13).

$$MSE = \frac{\sum_{f=1}^{n}(evaluated(f) - predicted(f))^2}{n} \quad (12)$$

$$RMSE = \sqrt{\frac{\sum_{f=1}^{n}(evaluated(f) - predicted(f))^2}{n}} \quad (13)$$

Table 2. Test functions

	Name	Function	Search Range	Dimension	Feature	Global optimum
1	Sphere	$f(x)=\sum_{i=1}^{n} x_i^2$	[-100,100]	30	US	0
2	Schwefel2.22	$f(x)=\sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	[-10,10]	30	UN	0
3	Schwefel1.2	$f(x)=\sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j\right)^2$	[-100,100]	30	UN	0
4	Schwefel2.21	$f(x)=max_i\{\lvert x_i\rvert, 1 \le i \le n\}$	[-100,100]	30	UN	0
5	Rosenbrock	$f(x)=\sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$	[-30,30]	30	UN	0
6	Step	$f(x)=\sum_{i=1}^{n} ([x_i + 0.5])^2$	[-100,100]	30	US	0
7	Quartic	$f(x)=\sum_{i=1}^{n} ix_i^4 + random[0,1)$	[-1.28,1.28]	30	US	0
8	Beale	$f(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	[-4.5,4.5]	5	UN	0
9	Rastrigin	$f(x)=\sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	[-5.12,5.12]	30	MS	0

	Name	Function	Search Range	Dimension	Feature	Global optimum
10	Ackley	$f(x)=\sum_{i=1}^{n} -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e$	[-32,32]	30	MN	0
11	Griewank	$f(x)=\frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600,600]	30	MN	0
12	Penalized	$f(x)=\frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i-1)^2[1+10sin^2(\pi y_{i+1})] + (y_n-1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10{,}100{,}4)$ $y_i = 1+\frac{x_i+1}{4} u(x_i,a,k,m) = f(x) = \begin{cases} k(x_i-a)^m & x_i > a \\ 0-a & < x_i < a \\ k(-x_i-a)^m & x_i < -a \end{cases}$	[-50,50]	30	MN	0
13	Penalized2	$f(x)=0.1\{sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i-1)^2[1+sin^2(3\pi x_i+1)] + (x_n-1)^2[1+sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5{,}100{,}4)$	[-50,50]	30	MN	0
14	Booth	$f(x)=(x_1+2x_2-7)^2+(2x_1+x_2-5)^2$	[-10,10]	2	MS	0
15	Kowalik	$f(x)=\sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2+b_i x_2}{b_i^2+b_i x_3+x_4}\right]^2$	[-5,5]	4	MN	0.000307486
16	Six Hump Camel Back	$f(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	[-5,5]	2	MN	-1.01316
17	Branin	$f(x)=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi})\cos x_1+10$	[-5,10]x[0,15]	2	MS	0.398
18	GoldStein-Price	$f(x)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]\times[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	[-2,2]	2	MN	3
19	Hartman 3	$f(x)=-\sum_{i=1}^{4} c_i exp(-\sum_{j=1}^{3} a_{ij}(x_j-p_{ij})^2$	[1,3]	3	MN	-3.86
20	Hartman 6	$f(x)=-\sum_{i=1}^{4} c_i exp(-\sum_{j=1}^{6} a_{ij}(x_j-p_{ij})^2$	[0,1]	6	MN	-3.32
21	Shekel 5	$f(x)=-\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	[0,10]	4		-10.1532
22	Shekel 7	$f(x)=-\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	[0,10]	4		-10.4028
23	Shekel 10	$f(x)=-\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	[0,10]	4		-10.5363

Here, $evaluated\ (f)$ (see Equation (12) and Equation (13)) is the optimum value calculated when the algorithm reaches the stopping criterion. $predicted\ (f)$ is the global optimum value expected from the algorithm and n is the function number. Wilcoxon signed-rank test was used

 10.12928/ijio.v4i2.7651

to statistically compare the quality of the solutions obtained from all independent studies given. The run time from the beginning of the algorithms to reaching the stopping criterion was calculated to evaluate the speed at which the algorithms reach the solution.

RESULTS AND DISCUSSION

The field of metaheuristic optimization algorithms has witnessed remarkable progress with the introduction of various innovative techniques. Particle Swarm Optimization (PSO) takes inspiration from the social behavior of animals like birds and fish to achieve efficient global optimization. Kennedy and Eberhart [19] proposed PSO, which has been widely used in diverse applications due to its simplicity and effectiveness. Another notable approach, the Grey Wolf Optimization (GWO), mimics the hierarchical hunting behavior of grey wolves to strike a balance between exploration and exploitation. Mirjalili, Mirjalili, and Lewis [20] introduced GWO, which has demonstrated promising results in solving complex optimization problems.

The Whale Optimization Algorithm (WOA) imitates the bubble-net hunting tactic of humpback whales to efficiently navigate through the search space. Its ability to balance exploration and exploitation makes it an attractive choice for optimization tasks. Mirjalili and Lewis [21] proposed WOA, showing its efficacy in handling various optimization challenges. Similarly, the Harris Hawks Optimization (HHO) draws inspiration from the collaborative hunting abilities of Harris's hawks, which allows it to effectively explore the solution space. Heidari et al. [22] introduced HHO, showcasing its capabilities in solving real-world optimization problems. Differential Evolution (DE) is a powerful evolutionary algorithm that efficiently explores and exploits the search space by generating new candidate solutions through mutation and crossover operations. Storn and Price [23] originally presented DE, which has been widely adopted for its versatility and success in optimization tasks.

These metaheuristic optimization algorithms have significantly advanced the state-of-the-art in optimization techniques and have been extensively applied across diverse domains. Their continued development and application are likely to yield further improvements and provide valuable solutions to increasingly complex real-world problems. Researchers and practitioners can benefit from the insights gained through these algorithms' exploration of biological and natural phenomena, opening up new avenues for future research and advancements.

Our algorithm has been compared with Particle Swarm Optimization, Grey Wolf Optimization, Whale Optimization Algorithm, Harris Hawks Optimization, Differential Evolution, FireFly Algorithm, Cuckoo Search, Multi-Verse Optimizer, Genetic Algorithm, and JAYA. All these algorithms have been tested in 23 benchmark tests described in Table 2. In Table 3, statistical results such as the best solution, mean, and standard deviation values achieved by algorithms for each function with 30 independent runs are shown. The mean values for the function are indicated in bold colors. If the data in Table 3 is examined in detail on a function-based, CA is the best algorithm in F1, F2, F3, and F4 unimodal functions. Unimodal functions are used to test the exploitation capability of the algorithm. The CA algorithm gave the best results in these four functions, and the exploitation capability of the CA algorithm can be mentioned. The algorithms that give the best results in F5, F6, F8 unimodal functions are HHO, PSO, and DE, respectively.

Multimodal functions have many local minima, they are used to test the exploration capabilities of algorithms. If the data in Table 3 is examined in detail on a function-based, CA and HHO gave the best results for F9, F10, and F16 multimodal functions. CA, WOA, and HHO had the best result in the F11 function. HHO is the best algorithm in F12, F13, and F15 functions. For the F14 function, the best algorithms are DE, PSO, and JAYA. For F16, F17, F18, and F19 functions, most algorithms' means are the same, but standard deviations are different. Therefore, algorithms with minimum standard deviation are DE and PSO in the F16 function, DE and PSO algorithms in the F17 function, CS algorithm in the F18 function, and DE and CS algorithm in the F19 function. For the F20 function, the best algorithms are DE and CS algorithms. For F22 and F23 multimodal functions, the best result is in the CS algorithm. CA algorithm gave approximate results to known correct answers in most multimodal functions, but the CA algorithm did not have better results than other algorithms in other multimodal functions except F9, F10, F11, and F16. The exploration capability of the CA algorithm is not as good as its exploitation capability of it. So, this capability needs to be improved a little more.

The success of the algorithms varies according to the dimension and characteristics of the problem and the parameters it uses. So, the conditions under which each algorithm is successful are different from each other. To compare the success of algorithms, each algorithm must be run under the same conditions such as termination criteria, population number, and dimension and characteristics of problems. So, in this study, the dimension of the problem was taken as 30 for multi-dimensional functions. The population number of all algorithms was taken as 50 and termination criteria were taken as 500 iteration number. Therefore, the performance of some algorithms was worse than other algorithms for some problems under these conditions. Table 4 shows the MSE and RMSE values of algorithms. The values of MSE and RMSE show how far the results of the algorithms are from the true value. HHO algorithm has the minimum value of MSE and RMSE. CA has the second minimum value. Table 5 shows the results of the Wilcoxon signed-rank test. For problem-based pairwise comparisons of test algorithms, all independent studies are. It was carried out using the globally optimum values.

The hypothesis thesis is as follows:

- H_0 (null hypothesis): 'There is no significant difference between the results of the two algorithms.
- H_1 (alternative hypothesis) is the opposite.

In the tests, the significance level was used as $\alpha = 0.05$. The *p*-value is the estimated probability of rejecting the H_0 hypothesis when the hypothesis is true. A small p-value indicates strong evidence against the null hypothesis. The '+' value refers to the sums of rank values in the results, where the first algorithm is better than the second. The '–' refers to the sums of rank values in the results, where the second algorithm is better. '=' indicates that there is no statistically significant difference between the success of the two algorithms in solving the problem. T represents the smallest of these sums. The last lines of Table 6 shows three statistical meanings in binary comparison and show the total numbers for the status (marked with '+', '=', or '-').

Table 5 shows that the CA algorithm was better than other algorithms in special problems such as F1, F2, F3, and F4 unimodal functions, but it could not have a noticeable success compared to other algorithms in F6 and F8 unimodal functions. CA algorithm was not as good as other algorithms in multimodal functions except F9, F10, F11, and F15. Table 3 was examined in detail, the best and mean values of the CA algorithm were the same as the other best algorithms for these multimodal functions. The standard deviation values of the CA algorithm were higher than other algorithms. Table 6 shows the average run time of the algorithms in seconds. DE has the best run time; CA is the second-best run time. JAYA is third. Especially, in F1, F2, F4, F5, F6, F7, F9, F10, F11, F12, and F13 functions, the CA algorithm has faster than other algorithms The CA algorithm has demonstrated effective exploitation capabilities when dealing with unimodal functions.

 doi 10.12928/ijio.v4i2.7651

Table 3. The best solution, mean, and std. dev. values of functions

		CA	GWO	WOA	HHO	DE	FFA	CS	PSO	MVO	JAYA	GA
F1	best	6.00E-112	3.52E-38	8.52E-95	1.12E-85	1.11E-03	3.02E-03	1.06E+01	1.63E-07	2.99E-01	1.82E-07	2.32E+01
	mean	**2.13E-109**	7.91E-37	8.58E-85	1.42E-69	4.44E-03	6.40E-03	2.14E+01	8.60E-06	7.43E-01	1.12E-05	4.87E+01
	std	3.44E-109	1.13E-36	3.60E-84	7.75E-69	2.19E-03	2.14E-03	5.98E+00	9.55E-06	2.25E-01	1.73E-05	1.55E+01
F2	best	2.46E-57	1.28E-22	3.28E-60	1.38E-43	2.54E-02	1.83E-01	8.65E+00	5.04E-04	3.94E-01	1.44E-05	1.16E+00
	mean	**1.78E-55**	6.99E-22	8.65E-54	5.55E-37	4.77E-02	9.59E+00	1.35E+01	4.00E+00	7.46E-01	8.14E-05	2.06E+00
	std	2.38E-55	5.11E-22	2.32E-53	2.69E-36	1.55E-02	1.17E+01	3.32E+00	6.75E+00	7.83E-01	4.73E-05	4.35E-01
F3	best	3.46E-111	1.47E-11	1.15E+04	5.80E-73	1.90E+04	3.10E+02	2.63E+03	1.81E+01	3.73E+01	5.86E+03	4.28E+03
	mean	**2.12E-108**	2.71E-07	3.32E+04	2.67E-48	2.85E+04	1.32E+03	3.48E+03	4.21E+01	9.49E+01	1.84E+04	1.10E+04
	std	5.02E-108	1.06E-06	1.13E+04	1.46E-47	4.57E+03	9.03E+02	7.26E+02	1.51E+01	4.73E+01	7.98E+03	2.88E+03
F4	best	5.76E-56	1.75E-09	3.87E-06	8.85E-44	8.35E+00	9.03E-02	9.23E+00	5.09E-01	4.78E-01	6.84E+00	9.38E+00
	mean	**5.57E-55**	7.26E-08	3.52E+01	7.03E-36	1.65E+01	2.18E-01	1.36E+01	8.47E-01	1.32E+00	1.67E+01	1.33E+01
	std	4.41E-55	7.90E-08	3.12E+01	2.69E-35	4.90E+00	7.97E-02	1.62E+00	2.05E-01	5.50E-01	6.31E+00	2.05E+00
F5	best	2.70E+01	2.53E+01	2.68E+01	3.54E-05	2.97E+01	2.28E+01	5.40E+02	1.32E+01	3.19E+01	2.57E+01	5.88E+02
	mean	2.85E+01	2.66E+01	2.76E+01	**6.58E-03**	4.68E+01	5.09E+02	1.04E+03	5.98E+01	4.33E+02	7.18E+01	1.58E+03
	std	5.46E-01	7.69E-01	3.94E-01	7.62E-03	2.86E+01	1.26E+03	4.66E+02	5.49E+01	6.80E+02	9.14E+01	9.57E+02
F6	best	2.62E+00	3.10E-05	1.98E-02	1.24E-07	1.02E-03	2.98E-03	9.37E+00	5.28E-07	2.91E-01	3.00E+00	2.13E+01
	mean	4.02E+00	4.33E-01	8.24E-02	4.27E-05	5.21E-03	6.81E-03	2.35E+01	**2.32E-05**	7.40E-01	3.77E+00	5.11E+01
	std	9.15E-01	2.85E-01	7.16E-02	5.26E-05	1.97E-03	2.37E-03	7.23E+00	8.33E-05	2.06E-01	5.36E-01	2.69E+01
F7	best	1.07E-05	3.31E-04	4.12E-05	1.09E-06	3.51E-02	1.38E-01	3.75E-02	3.49E-02	8.97E-03	6.62E-03	7.36E-02
	mean	1.29E-04	1.63E-03	2.10E-03	**6.88E-05**	4.88E-02	2.63E-01	7.98E-02	1.25E+00	2.26E-02	4.12E-02	1.63E-01
	std	1.04E-04	9.60E-04	1.75E-03	6.51E-05	9.04E-03	7.15E-02	2.38E-02	2.31E+00	8.06E-03	3.82E-02	5.03E-02
F8	best	5.82E-08	1.08E-08	4.90E-17	0.00E+00	0.00E+00	4.35E-11	5.07E-23	0.00E+00	5.21E-10	0.00E+00	7.88E-03
	mean	1.44E-01	1.12E-07	4.84E-10	1.58E-12	**0.00E+00**	5.68E-10	1.50E-17	**0.00E+00**	7.62E-02	1.62E-03	4.59E-01
	std	2.66E-01	9.99E-08	2.27E-09	3.45E-12	0.00E+00	5.04E-10	3.33E-17	0.00E+00	2.33E-01	6.63E-03	3.00E-01
F9	best	0.00E+00	1.14E-13	0.00E+00	0.00E+00	1.68E+02	3.88E+01	8.34E+01	4.39E+01	3.94E+01	2.82E+01	8.20E+00
	mean	**0.00E+00**	8.14E+00	5.68E-15	**0.00E+00**	1.88E+02	7.69E+01	1.09E+02	1.02E+02	1.14E+02	8.86E+01	1.14E+01
	std	0.00E+00	8.16E+00	1.73E-14	0.00E+00	6.98E+00	2.20E+01	1.38E+01	3.42E+01	2.93E+01	4.04E+01	2.16E+00
F10	best	4.44E-16	3.24E-14	4.44E-16	4.44E-16	1.57E-02	1.66E-02	4.57E+00	6.08E-04	3.29E-01	4.17E-04	2.50E+00
	mean	**4.44E-16**	3.82E-14	3.76E-15	**4.44E-16**	2.72E-02	6.90E-01	5.88E+00	2.32E-03	1.96E+00	1.34E-03	2.99E+00
	std	1.50E-31	3.30E-15	1.85E-15	1.50E-31	8.17E-03	6.50E-01	7.06E-01	1.71E-03	3.34E+00	1.15E-03	2.83E-01
F11	best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.92E-03	8.51E-03	1.07E+00	5.63E-08	6.01E-01	1.84E-06	1.22E+00

	mean	**0.00E+00**	2.05E-03	**0.00E+00**	**0.00E+00**	4.79E-02	2.00E-02	1.20E+00	8.13E-03	7.77E-01	5.66E-02	1.41E+00
	std	0.00E+00	4.77E-03	0.00E+00	0.00E+00	8.79E-02	8.21E-03	6.53E-02	7.65E-03	8.07E-02	1.08E-01	1.19E-01
	best	3.53E-01	2.86E-06	1.91E-03	1.03E-09	1.61E-02	3.94E-05	3.01E+00	1.69E-08	6.80E-01	7.02E-01	2.58E-01
F12	mean	5.69E-01	7.60E-02	1.93E-02	**1.89E-06**	8.93E-02	3.64E-01	3.83E+00	3.80E-02	1.94E+00	1.17E+00	6.02E-01
	std	1.41E-01	6.13E-02	3.91E-02	2.33E-06	7.51E-02	2.64E-01	5.59E-01	7.93E-02	8.88E-01	4.11E-01	2.68E-01
	best	1.76E+00	6.00E-05	2.83E-02	4.06E-08	4.77E-03	3.64E-04	4.31E+00	2.75E-07	4.72E-02	1.67E+00	1.64E+00
F13	mean	2.57E+00	3.85E-01	1.46E-01	**2.60E-05**	2.53E-02	5.27E-03	9.01E+00	5.15E-03	1.09E-01	3.54E+02	2.87E+00
	std	3.76E-01	2.13E-01	9.86E-02	3.90E-05	1.96E-02	6.82E-03	2.67E+00	5.56E-03	5.32E-02	1.90E+03	9.31E-01
	best	2.48E-08	5.11E-09	1.10E-06	9.65E-11	0.00E+00	4.99E-10	1.24E-26	0.00E+00	1.12E-08	0.00E+00	4.91E-05
F14	mean	5.42E-06	2.40E-07	1.98E-04	2.66E-07	**0.00E+00**	6.35E-09	1.74E-22	**0.00E+00**	5.58E-07	**0.00E+00**	4.88E-02
	std	7.15E-06	1.84E-07	2.06E-04	2.98E-07	0.00E+00	7.23E-09	5.86E-22	0.00E+00	5.37E-07	0.00E+00	5.30E-02
	best	3.08E-04	3.07E-04	3.08E-04	3.08E-04	3.07E-04	7.06E-04	3.35E-04	3.09E-04	5.13E-04	3.22E-04	7.89E-04
F15	mean	3.48E-04	2.32E-03	7.95E-04	**3.10E-04**	3.69E-04	1.08E-03	5.13E-04	4.77E-03	4.05E-03	1.08E-03	6.19E-03
	std	7.19E-05	6.12E-03	5.21E-04	2.43E-06	2.32E-04	2.39E-04	1.08E-04	7.94E-03	7.42E-03	5.08E-04	8.33E-03
	best	0.00E+00	1.14E-13	0.00E+00	0.00E+00	1.68E+02	3.88E+01	8.34E+01	4.39E+01	3.94E+01	2.82E+01	8.20E+00
F16	mean	0.00E+00	8.14E+00	5.68E-15	0.00E+00	1.88E+02	7.69E+01	1.09E+02	1.02E+02	1.14E+02	8.86E+01	1.14E+01
	std	0.00E+00	8.16E+00	1.73E-14	0.00E+00	6.98E+00	2.20E+01	1.38E+01	3.42E+01	2.93E+01	4.04E+01	2.16E+00
	best	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
F17	mean	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	**-1.03E+00**	-1.03E+00	-1.03E+00	**-1.03E+00**	-1.03E+00	-1.03E+00	-1.03E+00
	std	2.08E-07	1.57E-08	2.13E-10	2.76E-10	6.78E-16	3.66E-09	1.59E-13	6.78E-16	2.52E-07	2.17E-05	2.89E-03
	best	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01
F18	mean	3.98E-01	3.98E-01	3.98E-01	3.98E-01	**3.98E-01**	3.98E-01	3.98E-01	**3.98E-01**	3.98E-01	3.99E-01	4.02E-01
	std	2.08E-05	1.48E-06	1.61E-06	3.48E-07	1.13E-16	3.91E-09	3.20E-09	1.13E-16	3.36E-07	4.59E-03	6.70E-03
	best	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
F19	mean	3.00E+00	3.00E+00	3.00E+00	5.70E+00	3.00E+00	3.00E+00	**3.00E+00**	3.00E+00	3.00E+00	3.04E+00	7.77E+00
	std	2.63E-04	1.23E-05	4.73E-05	8.24E+00	4.85E-15	3.36E-08	4.52E-16	1.95E-15	2.84E-06	1.64E-01	1.08E+01
	best	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.85E+00	-3.86E+00
F20	mean	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	**-3.86E+00**	-3.86E+00	**-3.86E+00**	-3.86E+00	-3.86E+00	-3.72E+00	-3.86E+00
	std	2.71E-04	2.50E-03	2.70E-03	1.36E-05	1.36E-15	8.34E-10	1.36E-15	2.00E-03	8.12E-07	1.04E-01	2.16E-04
	best	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-2.94E+00	-3.32E+00

 10.12928/ijio.v4i2.7651

F21	**mean**	-3.27E+00	-3.24E+00	-3.27E+00	-3.26E+00	-3.24E+00	-3.27E+00	**-3.32E+00**	-3.25E+00	-3.27E+00	-2.16E+00	-3.27E+00
	std	7.74E-02	6.66E-02	7.40E-02	6.06E-02	5.54E-02	6.21E-02	7.69E-07	8.65E-02	5.99E-02	4.74E-01	5.92E-02
	best	-1.01E+01	-1.01E+01	-1.01E+01	-1.01E+01	-1.01E+01	-1.01E+01	-1.01E+01	-1.01E+01	-1.01E+01	-3.57E+00	-1.01E+01
F22	**mean**	-9.43E+00	-9.43E+00	-9.60E+00	-5.21E+00	-9.94E+00	-7.67E+00	**-1.01E+01**	-6.59E+00	-6.99E+00	-1.58E+00	-5.99E+00
	std	1.75E+00	1.74E+00	1.54E+00	9.24E-01	9.17E-01	2.68E+00	1.03E-07	2.84E+00	2.62E+00	8.11E-01	3.63E+00
	best	-1.02E+01	-1.02E+01	-1.02E+01	-5.06E+00	-1.02E+01	-1.02E+01	-1.02E+01	-1.02E+01	-1.02E+01	-4.20E+00	-1.02E+01
F23	**mean**	-6.64E+00	-9.66E+00	-7.98E+00	-5.06E+00	-9.42E+00	-7.67E+00	**-1.02E+01**	-7.74E+00	-6.14E+00	-1.82E+00	-4.40E+00
	std	3.70E+00	1.54E+00	3.00E+00	5.46E-05	1.99E+00	3.21E+00	1.00E-06	3.11E+00	2.83E+00	9.50E-01	3.01E+00

However, it has not performed as well in handling multimodal functions compared to other algorithms. One of its limitations lies in convergence issues with certain multi-no separable problems, and it lacks a well-controlled exploration-exploitation mechanism. Nonetheless, its simplicity allows it to compete favorably with other algorithms. In the context of unconstrained continuous optimization problems, the CA algorithm has not shown significant superiority over other approaches. Nevertheless, it has exhibited the ability to find results close to the best-known answers within a relatively shorter time compared to many alternative algorithms [12-14].

CONCLUSION

The search for the optimal solution to a problem has been a longstanding challenge for researchers considering the possible solutions. Over the years, numerous methods have been developed to address this issue. However, these methods have proven inadequate when faced with problems of varying sizes, constraints, and objectives. Interestingly, the universe, nature, and animals possess inherent problem-solving abilities that are executed effortlessly and swiftly, even without consciousness. This phenomenon has captured the attention of researchers, prompting them to explore and develop new methods inspired by the universe, nature, and animals, and they continue to do so.

Table 4. MSE and RMSE values of algorithms

	MSE	**RMSE**
HHO	6.70	2.59
CA	38.93	6.24
PSO	872.26	29.53
MVO	2.87E+04	169.27
FFA	1.88E+05	433.49
GWO	3.87E+05	622.12
CS	6.05E+05	777.66
JAYA	1.76E+07	4197.76
GA	5.78E+06	2.40E+03
DE	3.62E+07	6015.39
WOA	5.34E+07	7307.14

Table 5. The results of the wilcoxon signed-rank test

	CA - GWO			CA - WOA			CA - HHO			CA - DE			CA-GA		
F	p	T	W	p	T	W	p	T	W	p	T	W	p	T	W
F1	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+
F2	1.83E-06	0	+	3.88E-01	190	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+
F3	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+
F4	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+
F5	2.02E-06	1	-	1.18E-05	19	-	1.83E-06	0	-	1.83E-06	0	+	1.83E-06	0	+
F6	1.83E-06	0	-	1.83E-06	0	-	1.83E-06	0	-	1.83E-06	0	-	1.83E-06	0	+
F7	1.83E-06	0	+	4.08E-06	8	+	6.23E-03	99	-	1.83E-06	0	+	1.83E-06	0	+
F8	4.97E-06	10	-	1.83E-06	0	-	1.83E-06	0	-	1.83E-06	0	-	6.89E-04	67	+
F9	1.81E-06	0	+	1.49E-01	0	=	1.00E+0	0	=	1.83E-06	0	+	1.83E-06	0	+
F10	8.30E-07	0	+	1.96E-06	0	+	1.00E+0	0	=	1.83E-06	0	+	1.83E-06	0	+
F11	5.91E-02	0	=	1.00E+00	0	=	1.00E+00	0	=	1.83E-06	0	+	1.83E-06	0	+
F12	1.83E-06	0	-	1.83E-06	0	-	1.83E-06	0	-	1.83E-06	0	-	9.02E-01	226	=
F13	1.83E-06	0	-	1.83E-06	0	-	1.83E-06	0	-	1.83E-06	0	-	2.02E-01	170	=
F14	2.26E-05	26	-	2.24E-06	2	+	5.08E-05	35	-	1.83E-06	0	-	1.83E-06	0	+
F15	2.02E-01	170	=	1.88E-05	24	+	5.32E-02	138	=	3.73E-04	59	-	1.83E-06	0	+
F16	2.97E-05	29	-	1.83E-06	0	-	1.83E-06	0	-	1.83E-06	0	-	1.83E-06	0	+
F17	5.54E-05	36	-	4.26E-05	33	-	5.48E-06	11	-	1.83E-06	0	-	2.02E-06	1	+
F18	1.83E-06	0	-	2.02E-06	1	-	2.86E-03	87	-	1.83E-06	0	-	2.02E-06	1	+
F19	5.51E-01	203	=	2.26E-05	26	+	4.97E-06	10	-	1.83E-06	0	-	7.69E-02	146	=
F20	8.53E-01	223	=	7.73E-01	218	=	5.79E-01	205	=	4.11E-01	192	=	2.58E-01	177	=
F21	4.72E-01	197	=	4.83E-02	136	+	4.97E-06	10	+	2.26E-05	26	-	1.43E-05	21	+
F22	4.53E-03	94	-	2.10E-01	171	=	1.21E-02	110	+	1.20E-04	45	-	7.50E-03	102	+
F23	3.19E-04	57	-	1.28E-02	111	-	7.69E-02	146	=	1.83E-06	0	-	6.07E-01	207	=
+/=/-	**7/5/11**			**10/4/9**			**6/6/11**			**9/1/13**			**18/5/0**		

	CA - FFA			CA - CS			CA - PSO			CA - MVO			CA - JAYA		
F	p	T	W	p	T	W	p	T	W	p	T	W	p	T	W
F1	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+
F2	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+
F3	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+
F4	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+
F5	6.39E-04	66	+	1.83E-06	0	+	5.85E-02	140	=	1.83E-06	0	+	1.15E-03	74	+
F6	1.83E-06	0	-	1.83E-06	0	+	1.83E-06	0	-	1.83E-06	0	-	3.24E-01	184	=
F7	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+
F8	1.83E-06	0	-	1.83E-06	0	-	1.83E-06	0	-	2.18E-03	83	-	2.67E-03	86	-
F9	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+
F10	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+
F11	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+
F12	2.67E-03	86	-	1.83E-06	0	+	1.83E-06	0	-	2.48E-06	3	+	2.24E-06	2	+
F13	1.83E-06	0	-	1.83E-06	0	+	1.83E-06	0	-	1.83E-06	0	-	3.34E-06	6	+
F14	1.83E-06	0	-	1.83E-06	0	-	1.83E-06	0	-	5.93E-04	65	-	1.83E-06	0	-
F15	1.83E-06	0	+	4.08E-06	8	+	1.83E-06	0	+	1.83E-06	0	+	1.83E-06	0	+
F16	3.69E-06	7	-	1.83E-06	0	-	1.83E-06	0	-	5.24E-01	201	=	3.03E-06	5	+
F17	2.02E-06	1	-	1.83E-06	0	-	1.83E-06	0	-	3.69E-06	7	-	6.81E-01	212	=
F18	1.83E-06	0	-	1.83E-06	0	-	1.83E-06	0	-	1.83E-06	0	-	1.90E-02	118	+
F19	1.83E-06	0	-	1.83E-06	0	-	3.73E-04	59	-	1.83E-06	0	-	1.83E-06	0	+
F20	2.41E-01	175	=	1.83E-06	0	-	9.34E-01	228	=	9.17E-02	150	=	1.83E-06	0	+
F21	3.76E-01	189	=	1.83E-06	0	-	1.80E-02	117	+	1.80E-02	122	+	1.83E-06	0	+
F22	2.01E-02	119	-	1.83E-06	0	-	4.60E-02	135	-	7.89E-01	219	=	8.07E-06	15	+
F23	8.07E-06	15	-	1.83E-06	0	-	4.71E-04	62	-	9.55E-03	106	-	8.88E-06	16	+
+/=/-	10/2/11			13/0/10			10/2/11			12/3/8			19/2/2		

 10.12928/ijio.v4i2.7651

Table 6. The average run time of the algorithms in seconds (CPU time)

(Sn)	CA	GWO	WOA	HHO	DE	FFA	CS	PSO	MVO	JAYA	GA
F1	**0.67**	16.43	17.36	1.47	3.64	18.02	12.84	16.05	16.07	2.6	4.39
F2	**1.21**	16.52	14.07	2.19	2.63	17.66	13.5	15.96	16.4	3.42	4.33
F3	8.03	19.64	18.03	7.14	**5.83**	21.24	19.51	18.78	19.49	7.31	6.02
F4	**0.62**	16.25	14.2	1.19	2.24	17.55	13.04	15.45	16.37	3.05	4.09
F5	**1.08**	16.57	14.48	1.51	2.5	17.71	13.28	15.74	16.45	3.29	4.37
F6	**0.77**	16.31	14.4	1.28	3.21	17.7	13.21	15.64	16.72	3.16	4.25
F7	**1.45**	16.32	14.75	1.72	3.07	17.67	13.72	15.67	17.09	3.24	4.48
F8	0.63	1.53	1.33	1.55	1.08	18.04	1.82	1.25	1.42	**0.32**	3.31
F9	**0.89**	15.79	15.85	1.59	2.36	17.7	12.71	16.14	16.94	3.06	4.09
F10	**1.48**	16.34	17.77	1.96	2.73	18.36	13.59	16.41	18.52	3.56	4.52
F11	**2.00**	16.53	17.69	2.3	2.88	18.66	13.85	16.52	18.29	3.74	4.66
F12	**2.96**	16.95	18.67	2.99	3.42	19.15	14.79	17.15	18.88	4.35	5.05
F13	**2.73**	17.96	18.07	2.91	3.38	19.38	14.72	17.4	18.36	4.29	5.00
F14	0.58	1.72	1.52	1.45	1.34	19.09	2.05	1.24	1.8	**0.4**	3.24
F15	1.70	3.36	3.46	2.28	1.59	20.49	3.21	3.2	3.07	**1.06**	3.96
F16	0.66	1.5	1.54	1.3	0.98	21.08	1.51	1.41	1.45	**0.36**	3.19
F17	0.66	1.53	1.59	1.37	1.02	16.74	1.52	1.51	1.51	**0.38**	3.27
F18	1.01	1.71	1.78	1.66	1.16	16.82	1.88	1.67	1.72	**0.55**	3.34
F19	2.90	3.49	3.77	3.68	2.14	17.73	4.02	3.61	3.4	**1.84**	4.20
F20	2.90	5.32	5.75	3.66	2.33	17.81	5.3	5.11	5.11	**2.1**	4.23
F21	8.27	7.67	8.82	9.53	**5.14**	20.72	9.83	7.58	8.1	5.35	7.35
F22	11.80	9.38	9.7	12.51	**6.79**	22.39	13.35	9.46	10.4	7.51	8.85
F23	16.78	12.23	12.49	17.13	**9.38**	27.47	18.58	12.23	13.69	9.98	10.79
Sum	**71.77**	**251.05**	**247.09**	**84.37**	**70.84**	**439.18**	**231.83**	**245.18**	**261.25**	**74.92**	**110.99**
Mean	**3.12**	**10.92**	**10.74**	**3.67**	**3.08**	**19.09**	**10.08**	**10.66**	**11.36**	**3.26**	**4.83**

These approaches are referred to as metaheuristics. While they do not guarantee exact results, they can generate approximate solutions within a shorter timeframe. The adaptability of metaheuristics to a wide range of problem domains is also crucial. Contemporary problems are increasingly complex and multidimensional, necessitating methods that can navigate this complexity and uncertainty to approach the correct solution. Various algorithms have been developed inspired by many disciplines in literature. However, an algorithm with all the above advantageous characteristics has yet to be discovered. Hence, new algorithms with distinct features continue emerging in the literature as researchers keep exploring different methods within existing algorithms to find the ideal solution.

This study aims to contribute to the existing literature by introducing a novel algorithm. Currently, only a limited number of algorithms, such as ANN and AIS, have been developed with inspiration from the health field. In this research, a new health-based algorithm has been developed using cholesterol levels, which is vital in the human body.

The study begins with a brief introduction to optimization and metaheuristics. The algorithm development process is then discussed, accompanied by examples from existing algorithms in the literature. The functions of cholesterol in the human body and the characteristics of the lipoproteins that transport them are highlighted. The study describes how the Cholesterol Algorithm (CA) is inspired by the behavior of lipoproteins in the human body. Lipoproteins such as HDL, LDL, and triglycerides transport cholesterol in the human body. Maintaining specific levels of these lipoproteins is crucial for maintaining healthy blood flow in the vessels. Exceeding certain thresholds for LDL and falling below certain thresholds for HDL can lead to vessel blockage. The CA mimics this behavior by adjusting the number of LDL and HDL in response to the quality of the solution. When the solution improves, the number of LDL decreases and HDL increases, and vice versa when the solution deteriorates. The CA algorithm's performance in 23 unconstrained continuous optimization problems is compared to 10 algorithms from the literature. The evaluation includes measuring the best, average, and standard deviation values obtained by the algorithms for the given functions and runtime. In addition, Wilcoxon signed-rank test was performed to make binary comparisons of the CA algorithm with other algorithms.

In this study, the CA algorithm was tested only in unconstrained continuous problems. Its performance in other problem types, like discrete and real-world problems, is unknown. CA algorithm had proven its exploitation capability in unimodal functions, but multimodal functions did not succeed more than other algorithms. It has a convergence problem in some multi-non-separable problems and does not have a more controlled exploration-exploitation mechanism. However, it can compete with other algorithms with its simple structure. In addition, although it did not outperform other algorithms in unconstrained continuous optimization problems, the results were close to best-known answers in a shorter time than most of the algorithms.
In terms of future research, it is recommended to assess the performance of the CA algorithm in various problem types and real-world applications. This could include evaluating its efficacy in cost or profit functions within production or service environments to enhance managerial effectiveness. The parallel structure of the CA and its hybrid states, combining good features of other algorithms, should also be tested.

REFERENCES

[1] K. Sörensen and F. Glover, "Metaheuristics," Encycl. Oper. Res. Manage. Sci., vol. 62, pp. 960-970, 2013. doi: https://doi.org/10.1007/978-1-4419-1153-7_1167.
[2] J. H. Holland, "Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence," in MIT Press, 1975, doi: https://doi.org/10.7551/mitpress/1090.001.0001
[3] R. Storn and K. Price, “Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces,” J. Glob. Optim., Vol 11, pp 341-359, 1997, doi: https://doi.org/10.1023/A:1008202821328
[4] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proc. ICNN'95 - Int. Conf. on Neural Networks, 1995, pp. 1942-1948. doi: https://doi.org/10.1109/ICNN.1995.488968.
[5] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in Proc. 2009 World Congress on Nature & Biologically Inspired Computing (NABIC), Coimbatore, India, 2009, pp. 210-214. doi: https://doi.org/10.1109/NABIC.2009.5393690.
[6] X. S. Yang, "Firefly algorithms for multimodal optimization," in Nature Inspired Cooperative Strategies for Optimization (NISCO), L. C. Jain and P. N. Suganthan, Eds., Springer, Berlin, Heidelberg, 2009, pp. 169-178. doi: 10.1007/978-3-642-04944-6_14.
[7] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," Adv. Eng. Softw., vol. 95, pp. 51-67, 2016. doi: https://doi.org/10.1016/j.advengsoft.2016.01.008.
[8] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," Future Gener. Comput. Syst., vol. 97, pp. 849-872, 2019. doi: https://doi.org/10.1016/j.future.2019.02.028.
[9] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," Adv. Eng. Softw., vol. 69, pp. 46-61, 2014. doi: https://doi.org/10.1016/j.advengsoft.2013.12.007.
[10] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-Verse Optimizer: a nature-inspired algorithm for global optimization," Neural Comput. Appl., vol. 27, no. 2, pp. 495-513, 2016. doi: https://doi.org/10.1007/s00521-015-1870-7.
[11] R. Venkata Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems," Int. J. Ind. Eng. Comput., 2016. doi: https://doi.org/10.5267/j.ijiec.2015.8.004.
[12] C. M. M. Lawes, S. V. Hoorn, M. R. Law, and A. Rodgers, "Chapter 7 High Cholesterol," in Comparative Quantification of Health Risks: Global and Regional Burden of Disease Attributable to Selected Major Risk Factors, M. Ezzati, A. D. Lopez, A. Rodgers, and C. J. Murray, Eds., vol. 1, Geneva: World Health Organization, 2004. [Online]. Available: https://hsepedia.com/wp-content/uploads/2018/04/High-Cholesterol.pdf
[13] K. Bloch, "Chapter 12 Cholesterol: Evolution of structure and function," New Compr. Biochem., vol. 20, no. C, pp. 363–381, 1991. doi: https://doi.org/10.1016/S0167-7306(08)60340-3

 10.12928/ijio.v4i2.7651

[14] R. L. Jackson, J. D. Morrisett, and A. M. Gotto, "Lipoprotein structure and metabolism," Physiol. Rev., vol. 56, no. 2, pp. 259-318, 1976. doi: https://doi.org/10.1152/physrev.1976.56.2.259

[15] K. K. Birtcher and C. M. Ballantyne, "Measurement of Cholesterol," Circulation, vol. 110, no. 11, pp. 296–297, 2004, doi: https://doi.org/10.1161/01.cir.0000141564.89465.4e.

[16] P. Barter et al., "HDL Cholesterol, Very Low Levels of LDL Cholesterol, and Cardiovascular Events," N. Engl. J. Med., 2007. doi: https://doi.org/10.1056/NEJMoa064278.

[17] R. A. Khurma, I. Aljarah, A. Sharieh, and S. Mirjalili, "EvoloPy-FS: An Open-Source Nature-Inspired Optimization Framework in Python for Feature Selection," in Evolutionary Machine Learning Techniques, S. Mirjalili, H. Faris, I. Aljarah (eds.), Algorithms for Intelligent Systems. Springer, Singapore, 2020. doi: https://doi.org/10.1007/978-981-32-9990-0_8.

[18] H. Faris, I. Aljarah, S. Mirjalili, P. A. Castillo, and J. J. Merelo, "EvoloPy: An open-source nature-inspired optimization framework in python," in Proc. 8th Int. Joint Conf. Comp. Intelligence - Volume 0IJCCI, pp. 171-177, Porto, Portugal, 2016. doi: https://doi.org/10.5220/0006048201710177.

[19] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in Proc. IEEE Int. Conf. Neural Networks, pp. 1942-1948, 1995. doi: https://doi.org/10.1109/ICNN.1995.488968.

[20] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," Adv. Eng. Softw., vol. 69, pp. 46-61, 2014, doi: https://doi.org/10.1016/j.advengsoft.2013.12.007

[21] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," Adv. Eng. Softw., vol. 95, pp. 51-67, 2016, doi: https://doi.org/10.1016/j.advengsoft.2016.01.008

[22] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," Future Gener. Comput. Syst., vol. 97, pp. 849-872, 2019, doi: https://doi.org/10.1016/j.future.2019.02.028

[23] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," J. Glob. Optim., vol. 11, no. 4, pp. 341-359, 1997. doi: https://doi.org/10.1023/A:1008202821328.