A Tutorial on Child Drawing Development Optimization

Azad A. Ameen[1,2](✉), Tarik A. Rashid[3], and Shavan Askar[1]

[1] Information Systems Engineering Department, Erbil Technical Engineering College, Erbil Polytechnic University, Erbil, Iraq
Azad.ameen@charmouniversity.org

[2] Applied Computer Department, College of Medicals and Applied Sciences, Charmo University, Sulaymaniyah, Chamchamal, Iraq

[3] Computer Science and Engineering Department, University of Kurdistan Hewler, Erbil, Iraq

Abstract. In 2021, a novel metaheuristic algorithm inspired by the child's learning behavior and cognitive development employs the golden ratio. The golden ratio was first presented by the renowned scientist Fibonacci. The ratio of two consecutive numbers in the Fibonacci sequence is alike, and it is named the golden ratio, which is predominant in nature, architecture, design, and art. The Child Drawing Development Optimization Algorithm implements the golden ratio and mimics cognitive learning and the child's drawing development stages starting from the scribbling stage to the advanced pattern-based stage. This aids children with developing, refining their intelligence, and cooperatively achieving shared goals. The Child Drawing Development Optimization Algorithm is developed for single objective optimization problems, and it is also compared to Particle Swarm Optimization, Differential Evolution, Whale Optimization Algorithm, Gravitational search algorithm, and Fast Evolutionary Programming, Child Drawing Development Optimization Algorithm demonstrated its ability to handle complex optimization problems. The main contribution of this paper is to present and explains the Child Drawing Development Optimization Algorithm and uses them as models in a case study to minimize a fitness function. The initial population has been successfully fully improved as a result, and the best solution has been obtained by the Child Drawing Development Optimization Algorithm.

Keywords: Metaheuristic · Optimization Algorithm · Child Drawing Development Optimization Algorithm

1 Introduction

Optimization is the process of determining the variables of a function that will maximize or minimize the function value. The variables of the functions are frequently governed by constraints that limit the search space for determining the optimal values for the function. Optimization problems arise in the real world whenever there is a limited supply of resources, such as time, money, raw materials, and energy, that must be used to maximize or minimize certain outcomes. The study of optimization techniques is critical for solving

N. Bacanin and H. Shaker (Eds.): ICIITB 2022, ACSR 104, pp. 153–167, 2023.
https://doi.org/10.2991/978-94-6463-110-4_12

a wide range of problems in many fields, including engineering, economics, artificial intelligence, sociology, geology, and genetics [1]. Global optimization and nonlinear modeling are often viable applications for practically all metaheuristic techniques [2].

A metaheuristic is a problem-independent, high-level algorithmic framework that provides a set of recommendations or techniques for developing heuristic optimization algorithms. Among the many instances of metaheuristics are genetic/evolutionary algorithms, tabu search, simulated annealing, and ant colony optimization. Additionally, a metaheuristic is a problem-specific application of a heuristic optimization algorithm according to the guidelines provided by a metaheuristic framework [3]. The advantage of using these algorithms to solve complex problems is that they produce approximate solutions quickly, even for very complex problems [4]. Metaheuristics can be an effective method for producing acceptable solutions to complex problems through trial and error in a reasonable amount of time. Because of the complexity of the problem of interest, it is impossible to search for every possible solution or combination; instead, the goal is to find a good feasible solution within an acceptable timeframe. There is no guarantee that the best solutions will be found, and we don't even know whether an algorithm will work or why it will work if it does [5, 6]. The goal is to create an efficient and practical algorithm that works most of the time and produces high-quality results. Some of the discovered quality solutions are likely to be nearly optimal, though there is no guarantee of such optimality [7].

This has led to the development of heuristic optimization procedures, which can be used to address issues that cannot be resolved by derivative approaches. Therefore, many heuristic and metaheuristic algorithms have been presented [2]. Astonishing advances have been made in the field of metaheuristics, which aims to solve intractable optimization issues. There have been significant developments since the first metaheuristic was proposed, and many new algorithms continue to be proposed daily. Research in this area will surely advance further soon. Although, it is clear that there is a need to select the highest-performing metaheuristics that are anticipated to last [8]. These algorithms are either inspired by or based on human or natural populations [9].

The majority of cutting-edge metaheuristics were created before 2000 when these algorithms were referred to as "classical" metaheuristic algorithms [10, 11]. Thus, some popular classical metaheuristics algorithms, that are historically sorted, are Pattern Search (also known as direct search) [12], Simulated Annealing (SA) [13], Genetic Algorithms (GA) [14], Differential Evolution (DE) [15], Genetic Programming (GP) [16], Tabu search (TS) [17], Iterated Local Search (ILS) [18], Particle Swarm Optimization (PSO) [19], Artificial Bee Swarm (ABS) [19], Firefly Algorithm (FA) [20], Ant Colony Optimization (ACO) [21], Artificial Immune Algorithm (AIA) [22], Krill Herd Algorithm (KHA) [23], Harmony Search Algorithm (HAS) [24]. Despite the accomplishments of classical metaheuristic algorithms, new and novel evolutionary approaches have recently emerged successfully. During this era, research on metaheuristic algorithms introduces a significant number of new metaheuristics inspired by evolutionary or behavioral processes. This new wave of metaheuristic approaches frequently yields the best solutions for some of the unsolved benchmark problem sets [8].

Developing metaheuristics algorithms and applying them to the solution of optimization problems in a variety of scientific disciplines has emerged as a key area to fortify the active role of metaheuristic algorithms in the solution of real-world optimization problems [25]. Here, we highlight current works that analyze the use of novel or Hybridized optimization algorithms for various optimization issues to improve the feature selection process, engineering design, and other fields. X-rays and computerized tomography (CT) scans play a crucial role in the diagnosis of pneumonia and the COVID-19 virus. Monitoring the severity of lung infection with a technique that relies on image processing data from chest CT and X-rays is both time-consuming and inefficient. The proposed work addresses these problems with four interconnected steps, based on the Enhanced Whale with Salp Swarm Feature Classification [26]. Online transactions and credit card payments have increased significantly due to developments in online payment technologies and the COVID-19 epidemic. Naturally, credit card fraud has increased, affecting banks, credit card issuers, vendors, and merchants. Thus, there is an urgent need to adopt and establish adequate systems to safeguard online card transactions. To overcome this issue, the study has proposed hybrid machine learning and the novel, enhanced firefly algorithm, named group search firefly algorithm to address the challenge of credit card fraud detection [27]. After the COVID-19 pandemic, most countries had to take severe steps to contain the virus. Choosing measures requires predicting new cases. Upgraded COVID-19 forecasting, several improved forecasting approaches have been presented, such as a hybridization approach between machine learning, an adaptive neuro-fuzzy inference system, and enhanced beetle antennae search swarm intelligence metaheuristics [28], and hybridization between machine learning adaptive neuro-fuzzy inference system and enhanced genetic algorithm metaheuristics [29].

In 2021, Sabat Abdulhameed and Tarik A. Rashid published a Child Drawing Development Optimization Algorithm (CDDO) [30] to primarily address single objects. This paper's main contribution is to apply CDDO to a case study to minimize and obtain optimal solutions. For this purpose, a simple step-by-step guide is demonstrated. The paper can also be used by researchers to develop, improve, or hybridize the algorithm.

The remainder of the paper is divided into sections, with Sect. 2 providing a brief explanation of CDDO. A case study has been designed in Sect. 3 to evaluate CDDO. Finally, the results are summarized in Sect. 4.

2 Child Drawing Development Optimization Algorithm

The Child Drawing Development Optimization or CDDO algorithm is inspired by the child's learning behavior and cognitive development, and it uses the golden ratio to optimize the beauty of their art. Similar proportions between two consecutive numbers in the Fibonacci sequence are known as the "golden ratio," and they appear in nature, art, architecture, and design. The golden ratio is used by CDDO, and it also imitates cognitive learning and the stages of a child's development as a drawer, from scribbling to skilled pattern-based drawing. To achieve better results, the child's drawing's hand pressure width, length, and the golden ratio are adjusted. This assists children in evolving, improving their intelligence, and working together to achieve shared goals. This demonstrates CDDO's exceptional tenacity in seeking new solutions. It also demonstrates the algorithm's capability to avoid local minima by thoroughly covering promising regions within the design space and exploiting the best solution [30]. The CDDO algorithm operates in the manner seen in CDDO Pseudocode.

CDDO Pseudocode [30].

```
Begin
    Initialize child's drawing population X (i = 1, 2, ...,j)
    Compute each drawing's fitness
    Set personal best and global
    best Calculate the golden ratio of each drawing eq.(2) or (3)
    Create pattern memory array
    Randomly choose an index of pattern memory
        While (t < maximum number of iterations)
            Calculate RHP using eq. (4)
            Randomly choose hand pressure P1 Length P2, Width P3 For each
            drawing
            if (hand pressure was low)
                Update the drawings using eq. (5)
                Set LR and SR to HIGH (0.6-1) eq. using (6) and (7)
            Elseif (XiGR is near to golden ratio)
                Consider the learnt patterns, LR and SR using eq. (8)
                Set LR and SR to LOW (0-0.5) using (9) and (10)
            Endif
            Evaluate the cost values
            Update Personal best
            Update global best
            Update pattern memory
            Store the Best Cost Value
            increment t
        End While
    Return Global best
End
```

3 A Case Study Implementation

After exploring a variety of linear or nonlinear optimization problems, researchers need to have learned multiple lessons on how to successfully formulate models and what kind of algorithm will handle these challenges rapidly and reliably [31]. Although a single example cannot be used to deduce these lessons, it can be used to illustrate them. Train concerns for the CDDO approach are defined and summarized so that future researchers can state them concisely. For a minimization type of optimization (the context of an optimization strategy that seeks to achieve the lowest possible cost) as opposed to a maximization (the context of an optimization strategy that seeks to achieve the highest possible cost) type; consider the following function:

$$\boldsymbol{f}(\boldsymbol{x}),\ \text{where } \mathbf{f}(\mathbf{x}) = \mathbf{x_1^2} + \mathbf{x_2^2};\ \text{for integer x},\ 1 \le \mathrm{x}_1 \le 10 \text{ and } 1 \le \mathrm{x}_2 \le 10.$$

3.1 Calculating First Iteration

Step 1: Initialize the parameters of CDD
We set the parameters of CDDO as follows:-

$$\textit{Fitness Function} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x_1}, \boldsymbol{x_2}) = \boldsymbol{x_1^2} + \boldsymbol{x_2^2} \tag{1}$$

Lower Bound of Decision Variables (LB) = **1**
Upper Bound of Decision Variables (UB) = **10**
Dimension or Number of Unknown (Decision) Variables (dim) = **2**
Child Level Rate (LR) = **0.01**
Child Skill Rate (SR) = **0.9**
Pattern Matrix Size (PS) = **4**
Creativity Rate (CR) = **0.1**

Step 2: Generate the first population randomly and evaluate the fitness values of random solutions
Let's randomly create a CDDO population with two dimensions (x_1 *and* x_2) and CDDO consists of **10** drawings. Then evaluate the fitness of all CDDO drawings by the fitness equation $\mathbf{f(x) = {x_1}^2 + {x_2}^2}$. Calculate the summation and find the minimum fitness form CDDO as shown in Table 1, we try to evaluate the fitness values of random solutions (Cost) using Eq. (1).

In the beginning, for both dimensions x1 and x2, we manually generated a random number. The second power of all the numbers generated in the first stage is then discovered. Next, we calculate the total of the numbers we derived in stage two. Finding fitness function optimization by the given equation must be the final step as shown in Table 1 for example, randomly generate (5) and (7) for x1 and x2 then we find:

$$5^2 + 7^2 = 25 + 49 = 74$$

Step 3: Calculate the Drawing's Golden Ratio
Another factor that is used to update the solution and enhance its performance is the Golden Ratio (GR). The length and width of a child's drawing are the two elements that make up the solution's selected ratio, GR (see Eqs. (2) or (3)). Each of these two factors is chosen at random from all the other problem factors utilized as shown in Table 1.

$$\text{Golden Ratio(GRi)} = \frac{\boldsymbol{x1} + \boldsymbol{x2}}{\boldsymbol{x1}} \tag{2}$$

$$\text{Golden Ratio(GRi)} = \frac{\boldsymbol{x2} + \boldsymbol{x1}}{\boldsymbol{x2}} \tag{3}$$

Step 4: Find Personal Best
We set Global minima to an infinite number to reach a minimum value since this type of example that is supposed to be used is minimization.

Table 1. Randomly Generate the first population with Evaluate fitness values and Golden Ratio

CD i	x1	x2	Fitness Function (Cost)	Golden Ratio (GR)
CD 01	5	7	25 + 49 = **74**	(x2+x1)/x2 = (7+5)/7 = **1.714285**
CD 02	3	3	9 + 9 = **18**	(x1+x2)/x1 = (3+3)/3 = **2**
CD 03	8	4	64 + 16 = **80**	(x2+x1)/x2 = (4+8)/4 = **3**
CD 04	5	10	25 + 100 = **125**	(x2+x1)/x2 = (10+5)/10 = **1.5**
CD 05	6	8	36 + 64 = **100**	(x2+x1)/x2 = (8+6)/8 = **1.75**
CD 06	2	10	4 + 100 = **104**	(x2+x1)/x2 = (10+2)/10 = **1.2**
CD 07	2	9	4 + 81 = **85**	(x2+x1)/x2 = (9+2)/9 = **1.222222**
CD 08	10	4	100 + 16 = **116**	(x1+x2)/x1 = (10+4)/10 = **1.4**
CD 09	9	5	81 + 25 = **106**	(x1+x2)/x1 = (9+5)/9 = **1.555555556**
CD 10	5	3	25 + 9 = **34**	(x1+x2)/x1 = (5+3)/5 = **1.6**
Sum fitness = 842			**Average fitness = 84.2**	**Min fitness = 18**

Global Best = infinite number. And: -

Personal Best (Local Minima) = fitness function for each child drawing $f(x_1, x_2)$, which we named *(Cost)*.

a. Persona Best Fitness (i) = Fitness Function of each drawing.
b. Persona Best Drawing (i) = Employees Drawing (i), as shown in Table 2.

Step 5: Find Global Best

After analyzing the results of each drawing, we update the global best by comparing it to the local best.

IF (Local best < **Global Best**) **THEN** (**Global Best** = Local best)

Or

Global Best = minimization of Persona Best Fitness (i)

= minimization of (74, 18, 80, 125, 100, 104, 85, 116, 106, 34)

= (18)

Therefore, the Global Best of this iteration is (18), as shown in Table 2.

3.2 Calculating Second Iteration

Step 1: Set an array of the learning pattern

During this stage, the drawings are more standardized and copied, and the child compares his or her drawing to the best pattern he or she learned and defines the best sketched drawing so far. Additionally, the child recreates new scribbles by imitating the best surrounding artist and comparing their drawing to the group's best sketched drawing so far. After ordering local bests from least to greatest, the Pattern Matrix Size (PS) was determined to be the first five (4) drawings with the least amount of fitness.

During this stage, we do the following: -

1. Sort the result of personal best fitness from the lowest to the highest value.

Table 2. Finding Personal Best and Global Best

CD i	Personal (Local) Best			Global Best		
	x1	x2	Cost	x1	x2	Cost
CD 01	5	7	74			
CD 02	**3**	**3**	**18**	**3**	**3**	**18**
CD 03	8	4	80			
CD 04	5	10	125			
CD 05	6	8	100			
CD 06	2	10	104			
CD 07	2	9	85			
CD 08	10	4	116			
CD 09	9	5	106			
CD 10	5	3	34			

Table 3. Setting an array of Pattern Matrix Size (PS)

Pattern Matrix Size (PS)			
PS i	X1	X2	Cost
PS 1	3	3	18
PS 2	5	3	34
PS 3	8	4	80
PS 4	2	9	85

2. Then, depending on the pattern size, which in this example is 4, we copy the first five fitness to the pattern size array. as shown in Table 3**.**

Step 2: Set Random Hand Pressure (RMP)
Then, utilizing Eq. (4), we determined the Random Hand Pressure (RHP). Which is to generate a random number somewhere between the bottom boundary and the upper boundary for each drawing.

$$RHP = rand(LB, UP) \tag{4}$$

For instance, if we place each miner's Random Hand Pressure (RHP) reading anywhere between the lower border (1) and the upper boundary (10) of the range, we will obtain the following results that are shown in Table 4.

Step 3: Generate a new Solution
During this stage, a new drawing is generated and updated by randomly selecting between (x_1) and (x_2) for each drawing, followed by a comparison with the RHP in Table 5.

A. If xi is less than and equal to RHP, then: -

Table 4. Set Random Hand Pressure

CD i	Personal (Local) Best			Random Hand Pressure
	x1	x2	Cost	RHP i
CD 01	5	7	74	RHP 1 =rand (1,10) = **4**
CD 02	3	3	18	RHP 2 =rand (1,10) = **5**
CD 03	8	4	80	RHP 3 =rand (1,10) = **10**
CD 04	5	10	125	RHP 4 =rand (1,10) = **10**
CD 05	6	8	100	RHP 5 =rand (1,10) = **4**
CD 06	2	10	104	RHP 6 =rand (1,10) = **9**
CD 07	2	9	85	RHP 7 =rand (1,10) = **1**
CD 08	10	4	116	RHP 8 =rand (1,10) = **5**
CD 09	9	5	106	RHP 9 =rand (1,10) = **7**
CD 10	5	3	34	RHP 10=rand (1,10) = **2**

a. we update the drawing by using Eq. (5)

$$X_{i+1} = GR + SR * (X_{ilbest} - X_i) + LR * (X_{igbest} - X_i) \tag{5}$$

It means: -

New Drawing (i) = GR (i) + SR * (Persona Best Drawing (i) - Employees Drawing (i)) + LR *(Global Best Drawing – Persona Best Drawing (i)).

b. we update the Child Level Rate (LR) and Child Skill Rate (SR) by Eqs. (6) and (7).

$$\text{Child Level Rate(LR)} = \text{rand}[6 - 10]/10 \tag{6}$$

$$\text{Child Skill Rate(SR)} = \text{rand}[6 - 10]/10 \tag{7}$$

B. Else if the value of the Golden Ratio (GR) lies between (1.5) and (2) then: -

a. we update the drawing by using Eq. (8)

$$Xi + 1 = \mathrm{X}iMP + CR * (Xigbest) \tag{8}$$

It means: -

New Drawing (i) = Pattern Size Drawing (i) + CR * Global Best Drawing (i).

b. we update the Child Level Rate (LR) and Child Skill Rate (SR), using Eqs. (9) and (10).

$$\text{Child Level Rate(LR)} = \text{rand}[0 - 0.5] \tag{9}$$

$$\text{Child Skill Rate(SR)} = \text{rand}[0 - 0.5] \tag{10}$$

Table 5. Comparison between first Iteration drawings with Random Hand Pressure.

CDi	First Iteration Drawing		RHP i	GR	Comparison
	X1	X2			
CD 01	**5**	7	4	1.714285	**5 > 4**
CD 02	3	**3**	5	2	**3 < 5**
CD 03	**8**	4	10	3	**8 < 10**
CD 04	5	**10**	10	1.5	**10 = 10**
CD 05	6	**8**	4	1.75	**8 > 4**
CD 06	2	**10**	9	1.2	**10 > 9**
CD 07	**2**	9	1	1.222222	**2 > 1**
CD 08	**10**	4	5	1.4	**10 > 5**
CD 09	**9**	5	7	1.555555	**9 > 7**
CD 10	5	**3**	2	1.6	**3 > 2**

The schematics in this study can be updated using one of three implantation methods, as shown in Table 5. The drawings that were selected at random are indicated by the numbers that are highlighted above.

1. To update and obtain new drawings for (CD 02, CD 03, and CD 04), we are applying Eq. (5) as stated in Table 5. After that, we randomly update the values of Child Level Rate (LR) and Child Skill Rate (SR) between (0.6, 1.0), as given in Table 6.
2. By using equation No. (8) and taking into account the first iteration drawings, Pattern Size, and Global Best value, we update and acquire new drawings for (CD 01, CD 05, CD 09, and CD 10), as shown in Tables 7 and 8.

After that, we at random alter the Child Level Rate (LR) and Child Skill Rate (SR) values between (0.0, 0.5), as shown in Table 8.

3. As mentioned in Table 5, there are no new updates for (CD 06, CD 07, and CD 08).

Step 4: Evaluate the fitness function of the second iteration

c. Compute the value of the fitness function using the same fitness function and assess it using the value of the updated drawings; the resulting Table 9 displays the following:

Step 5: Find Personal Best

Then, identical to the previous iteration, a new state for Personal Best could be discovered; the following formula is required: -

Table 6. Finding new drawings by using Eq. (5) and randomly updating the values of Child Level Rate and Child Skill Rate

CD i	**1st Iteration Drawing**		**New drawing Updating**	**2nd Iteration Drawing**		**Randomly generate (LR) = rand(0.6,1.0)**	**Randomly generate (SR) = rand(0.6,1.0)**
	X1	X2		X1	X2		
CD 02	3	**3**	$Xi{+}1 = GR + SR \cdot{*} (Xilbest - Xi) + LR \cdot{*} (Xigbest - Xi)$ X2 =2 + 0.9 * (3-3) + 0.01 * (3-3) = **2**	3	**2**	**0.9**	**0.8**
CD 03	**8**	4	X1 =3+ 0.9 * (8-8) + 0.01 * (3-8) = **2.95**	**2.95**	4	**0.6**	**0.5**
CD 04	5	**10**	X2 =1.5 + 0.9 * (10-10) + 0.01 (3-10) = **1.43**	5	**1.43**	**0.7**	**0.9**

Table 7. The first iteration drawings, Pattern Size, and Global Best value

CD i	**First Iteration**	
	X1	**X2**
CD 01	5	7
CD 05	6	8
CD 09	9	5
CD 10	5	3

PS i	**Pattern Matrix Size (PS)**	
	X1	**X2**
PS 1	3	3
PS 2	5	3
PS 3	8	4
PS 4	2	9

Global Best	
X1	**X2**
3	*3*

Table 8. Finding new drawings by using Eq. (8) and randomly updating the values of Child Level Rate and Child Skill Rate

CD i	**1st Iteration Drawing**		**New drawing Updating**	**2nd Iteration Drawing**		**Randomly generate (LR) = rand(0.0,0.5)**	**Randomly generate (SR) = rand(0.0,0.5)**
	X1	X2		X1	X2		
CD 01	**5**	7	$Xi{+}1 = XiMP + CR \cdot{*} (Xigbest\ )$ X1 =2 + 0.1 * 3 = **2.3**	**2.3**	7	0.2	0.1
CD 05	6	**8**	X2 =3 + 0.1 * 3 = **3.3**	6	**3.3**	0.4	0.2
CD 09	9	**5**	X2 =4 + 0.1 * 3 = **4.3**	9	**4.3**	0.5	0.4
CD 10	**5**	3	X1 =3 + 0.1 * 3 = **3.3**	**3.3**	3	0.1	0.3

Table 9. Second iteration steps to Find values of the fitness function

CD i	x1	x2	Fitness Function (Cost)
CD 01	2.3	7	5.29 + 49 = **54.29**
CD 02	3	2	9 + 4 = **13**
CD 03	2.95	4	8.7025 + 16 = **24.7025**
CD 04	5	1.43	25 + 2.0449 = **27.0449**
CD 05	6	3.3	36 + 10.89 = **46.89**
CD 06	2	10	4 + 100 = **104**
CD 07	2	9	4 + 81 = **85**
CD 08	10	4	100 + 16 = **116**
CD 09	9	4.3	81 + 18.49 = **99.49**
CD 10	3.3	3	10.89 + 9 = **19.89**
Sum fitness = 590.3074		**Average fitness = 59.03074**	**Min fitness = 13**

Table 10. Updating Second iteration Personal Best

2^{nd} iteration drawing		Comparison	Old Personal Best (1^{st} Iteration)		New Personal Best for 2^{nd} Iteration		
CD i	Fitness Function		Cost		Drawing		Cost
					X 1	X 2	
CD 01	54.29	<	74	→	2.3	7	54.29
CD 02	13	<	18	→	3	2	13
CD 03	24.7025	<	80	→	2.95	4	24.7025
CD 04	27.0449	<	125	→	5	1.43	27.0449
CD 05	46.89	<	100	→	6	3.3	46.89
CD 06	104	=	104		2	10	104
CD 07	85	=	85		2	9	85
CD 08	116	=	116		10	4	116
CD 09	99.49	<	106	→	9	4.3	99.49
CD 10	19.89	<	34	→	3.3	3	19.89

IF fitness function value < Personal (Local) Best **THEN**

1. Personal (Local) Best = fitness function value
2. Personal Best drawing = 2^{nd} iteration drawing

As indicated in Table 10, several Personal Bests are revised after application of the above formula.

Table 11. The result of 2^{nd} Iteration

2^{nd} iteration drawing								Personal (Local) Best		
CD i	**Drawing**		**Cost**	**GR**	**LR**	**SR**	**CR**	**Drawing**		**Cost**
	x1	**x2**						**x1**	**x2**	
CD 01	2.3	7	54.29	1.714285	0.2	0.1	0.1	2.3	7	54.29
CD 02	3	2	13	2	0.9	0.8	0.1	**3**	**2**	**13**
CD 03	2.95	4	24.7025	3	0.6	0.5	0.1	2.95	4	24.7025
CD 04	5	1.43	27.0449	1.5	0.7	0.9	0.1	5	1.43	27.0449
CD 05	6	3.3	46.89	1.75	0.4	0.2	0.1	6	3.3	46.89
CD 06	2	10	104	1.2	0.01	0.9	0.1	2	10	104
CD 07	2	9	85	1.222222	0.01	0.9	0.1	2	9	85
CD 08	10	4	116	1.4	0.01	0.9	0.1	10	4	116
CD 09	9	4.3	99.49	1.555555	0.5	0.4	0.1	9	4.3	99.49
CD 10	3.3	3	19.89	1.6	0.1	0.3	0.1	3.3	3	19.89

Sum fitness = 590.3074 Average fitness = 59.03074 Min fitness = 13

Global Best = 13

Step 6: Find Global Best

Therefore, it must also be Updated Global Best for every drawing by comparing it to the Local best values.

IF (Local best < Global Best) **THEN** (Global Best = Local best) Or

Global Best = minimum value of Persona Best Fitness (i)

= minimum value of (54.29, 13, 2, 07025, 27.0449, 46.89, 104, 85, 116, 99.49, 19.89) = (**13**)

As stated in Table 11, the Global Best for this iteration is **13**.

Finally, as shown in Table 11, updates to drawings, personal best, and global best have been implemented.

4 Result

As a result of the implementation of this study, it is obvious from the outcomes that the drawings have been enhanced and that the productivity of individuals has improved. Table 12. Displays a comparison between the sum, the average, and the optimal value for each iteration. This new approach enhances both exploration and exploitation, as evidenced by the findings. As a result, the time complexity and convergence are enhanced by these modifications. Readers interested in assessing this study might look at Tables 2 and 11 to see how the numbers changed.

Table 12. Comparison between the results of 2 iterations

Iteration N	Sum	Average	Min
Iteration 1	842	84.2	18
Iteration 2	590.3074	59.03074	13

5 Conclusion

This study proposes an algorithm for optimizing the child's drawing development. A case study is intended to describe the golden ratio and replicates cognitive development and a child's drawing phases utilizing factors such as hand pressure width, length, and golden ratio that may mislead readers of this method. In the experimental outcomes, CDDO showed its capacity for enhancing and enhancing features, as well as locating the ideal answer. CDDO improves and iteratively obtains superior solutions. The authors suggest that this algorithm's processing time could be decreased by enhancing it. In addition, modifying the randomization parameter with other randomization techniques or mathematical equations from other competitive optimization algorithms may improve CDDO's performance.

In future work, CDDO will be used to adapt and test binary and multi-objective optimization problems. Finally, future studies may investigate combining CDDO with other algorithms and incorporating evolutionary operators.

References

1. P. P. Cruz, A. M. Gutierrez, R. A. Ramırez-Mendoza, E. M. Flores, A. A. O. Espinoza, and D. C. B. Silva, *A Practical Approach to Metaheuristics using LabVIEW and MATLAB*. Taylorn & Francis Group, LLC, 2020.
2. A. H. Gandomi, X.-S. Yang, S. Talatahari, and A. H. Alavi, *Metaheuristic applications in structures and infrastructures*. Newnes, 2013.
3. F. Glover, "1 Definition 2 A taxonomy of metaheuristics," *springer, New York*, vol. 1, 1986.
4. S. Ahmed and T. A. Rashid, "Learner Performance-based Behavior Optimization Algorithm : A Functional Case Study," pp. 1–12.
5. X.-S. Yang, *Nature-inspired metaheuristic algorithms*. Luniver press, 2010.
6. X.-S. Yang, *Engineering optimization: an introduction with metaheuristic applications*. John Wiley & Sons, 2010.
7. A. H. Gandomi, X. S. Yang, S. Talatahari, and A. H. Alavi, *Metaheuristic Algorithms in Modeling and Optimization*, no. December. 2013.
8. T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, "A survey on new generation metaheuristic algorithms," *Comput. Ind. Eng.*, vol. 137, no. August, p. 106040, 2019, doi: https://doi.org/10.1016/j.cie.2019.106040.
9. S. Shadravan, H. R. Naji, and V. K. Bardsiri, "The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 80, pp. 20–34, 2019, doi: https://doi.org/10.1016/j.engappai.2019.01.001.
10. G. R. Raidl, J. Puchinger, and C. Blum, "Metaheuristic Hybrids BT - Handbook of Metaheuristics," M. Gendreau and J.-Y. Potvin, Eds. Cham: Springer International Publishing, 2019, pp. 385–417.

11. A. E. Ezugwu *et al.*, "Metaheuristics: a comprehensive overview and classification along with bibliometric analysis," *Artif. Intell. Rev.*, vol. 54, no. 6, pp. 4237–4316, 2021, doi: https://doi.org/10.1007/s10462-020-09952-0.
12. E. Baeyens, A. Herreros, and J. R. Perán, "A Direct Search Algorithm for Global Optimization," *Algorithms* , vol. 9, no. 2. 2016, doi: https://doi.org/10.3390/a9020040.
13. P. J. M. van Laarhoven and E. H. L. Aarts, "Simulated annealing BT - Simulated Annealing: Theory and Applications," P. J. M. van Laarhoven and E. H. L. Aarts, Eds. Dordrecht: Springer Netherlands, 1987, pp. 7–15.
14. D. E. Goldberg and J. H. Holland, "Genetic Algorithms and Machine Learning," *Mach. Learn.*, vol. 3, no. 2, pp. 95–99, 1988, doi: https://doi.org/10.1023/A:1022602019183.
15. 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.
16. Wolfgang Banzhaf, *Genetic programming: an introduction: on the automatic evolution of computer programs and its applicationse*, 1st ed. addison-wesley publishing company inc., 1998.
17. F. Glover and M. Laguna, "Tabu Search BT - Handbook of Combinatorial Optimization: Volume1–3," D.-Z. Du and P. M. Pardalos, Eds. Boston, MA: Springer US, 1998, pp. 2093–2229.
18. H. R. Lourenço, O. C. Martin, and T. Stützle, "Iterated Local Search BT - Handbook of Metaheuristics," F. Glover and G. A. Kochenberger, Eds. Boston, MA: Springer US, 2003, pp. 320–353.
19. J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995, vol. 4, pp. 1942–1948 vol.4, doi: https://doi.org/10.1109/ICNN.1995.488968.
20. X.-S. Yang, "Firefly Algorithms for Multimodal Optimization BT - Stochastic Algorithms: Foundations and Applications," 2009, pp. 169–178.
21. M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, 2006, doi: https://doi.org/10.1109/MCI.2006.329691.
22. D. Dasgupta, "An Overview of Artificial Immune Systems and Their Applications BT - Artificial Immune Systems and Their Applications," D. Dasgupta, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 3–21.
23. G. G. Wang, L. Guo, A. H. Gandomi, G. S. Hao, and H. Wang, "Chaotic Krill Herd algorithm," *Inf. Sci. (Ny).*, vol. 274, no. January, pp. 17–34, 2014, doi: https://doi.org/10.1016/j.ins.2014.02.123.
24. X. Z. Gao, V. Govindasamy, H. Xu, X. Wang, and K. Zenger, "Harmony Search Method: Theory and Applications," *Comput. Intell. Neurosci.*, vol. 2015, p. 258491, 2015, doi: https://doi.org/10.1155/2015/258491.
25. E. H. Houssein, M. A. Mahdy, D. Shebl, and W. M. Mohamed, "A Survey of Metaheuristic Algorithms for Solving Optimization Problems BT - Metaheuristics in Machine Learning: Theory and Applications," D. Oliva, E. H. Houssein, and S. Hinojosa, Eds. Cham: Springer International Publishing, 2021, pp. 515–543.
26. N. Budimirovic *et al.*, "COVID-19 Severity Prediction Using Enhanced Whale with Salp Swarm Feature Classification," *Comput. Mater. Contin.*, vol. 72, no. 1, pp. 1685–1698, 2022, doi: https://doi.org/10.32604/cmc.2022.023418.
27. D. Jovanovic, M. Antonijevic, M. Stankovic, M. Zivkovic, M. Tanaskovic, and N. Bacanin, "Tuning Machine Learning Models Using a Group Search Firefly Algorithm for Credit Card Fraud Detection," *Mathematics*, vol. 10, no. 13, pp. 1–30, 2022, doi: https://doi.org/10.3390/math10132272.

28. M. Zivkovic *et al.*, "COVID-19 cases prediction by using hybrid machine learning and beetle antennae search approach," *Sustain. Cities Soc.*, vol. 66, no. November 2020, p. 102669, 2021, doi: https://doi.org/10.1016/j.scs.2020.102669.
29. M. Zivkovic *et al.*, "Hybrid Genetic Algorithm and Machine Learning Method for COVID-19 Cases Prediction BT - Proceedings of International Conference on Sustainable Expert Systems," 2021, pp. 169–184.
30. S. Abdulhameed and T. A. Rashid, "Child Drawing Development Optimization Algorithm Based on Child's Cognitive Development," *Arab. J. Sci. Eng.*, 2021, doi: https://doi.org/10.1007/s13369-021-05928-6.
31. Y. Bengio, A. Lodi, and A. Prouvost, "Machine learning for combinatorial optimization: A methodological tour d'horizon," *Eur. J. Oper. Res.*, vol. 290, no. 2, pp. 405–421, 2021, doi: https://doi.org/10.1016/j.ejor.2020.07.063.