A novel meta-heuristic optimization algorithm based on cell division: Cell Division Optimizer

Sehej Jain and Kusum Kumari Bharti[†]

PDPM Indian Institute of Information Technology Design and Manufacturing, Jabalpur,
Madhya Pradesh, India.

*Corresponding author(s). E-mail(s): sehejjain@outlook.com;

Contributing authors: kusum@iiitdmj.ac.in;

[†]These authors contributed equally to this work.

Abstract

A novel meta-heuristic algorithm named as the Cell Division Optimizer (CDO) is proposed. The proposed algorithm is inspired by the reproduction methods at the cellular level, which is formulated by the well-known cell division process known as mitosis and meiosis. In the proposed model Meiosis and Mitosis govern the exploration and exploitation aspects of the optimization algorithm, respectively. In the proposed method, the solutions are updated in two phases to achieve the global optimum solution. The proposed algorithm can be easily adopted to solve the combinatorial optimization method. To evaluate the proposed method, 50 well-known benchmark test functions and also 2 classical engineering optimization problems including 1 mechanical engineering problem and 1 electrical engineering problem are employed. The results of the proposed method are compared with the latest versions of state-of-the-art algorithms like Particle Swarm Optimization, Cuckoo Search, Grey Wolf Optimizer, FruitFly Optimization, Whale Optimizer, Water-Wave Optimizer and recently proposed variants of top-performing algorithms like SHADE (success history-based adaptive differential evolution) and CMAES (Covariance matrix adaptation evolution strategy). Moreover, the convergence speed of the proposed algorithm is better than the considered competitive methods in most cases.

Keywords: Optimization, Cell division optimization, Meta-heuristics , Mitosis , Meiosis, Engineering optimization problem

1 Introduction

Many real-world design problems are defined as a decision-making process to build products that satisfy defined requirements. Usually, these design problems include complicated objective functions with a large number of decision variables. A solution that satisfies the specified constraints of the different decision variables along with attaining the better possible value of the objective function is considered as a potential solution. In the

existing literature, different algorithms have been developed to solve optimization problems. The majority of the algorithms are based on linear and nonlinear methods. These methods require substantial gradient information. Moreover, they are often stuck at local optimum solutions. These optimization algorithms are good for simple problems. However, most real-world problems are very complex in nature. The complexity of these models makes it difficult to apply these algorithms to solve real-world problems. The limitation of

the traditional numerical methods has motivated to explore the possibility of using meta-heuristic algorithms to solve the real-world problem.

Over the past few decades, meta-heuristic algorithms have been explored to solve a wide variety of problems. Researchers have contributed with a number of algorithms inspired by natural phenomena such as Particle Swarm Optimization [1], inspired by birds flocking and fish schooling. John Holland [2] proposed the Genetic Algorithm which was inspired by the Darwinian concepts of evolution. Karaboga et al [3] proposed the Artificial Bee Colony Algorithm inspired by the foraging behaviour of bees in a honey bee swarm. Yang et al [4] have proposed Cuckoo Search Algorithm inspired by the egg-laying behaviour of Cuckoo Birds. Seyedali Mirjalili [5] has proposed whale optimization algorithms inspired by the hunting behaviour of whales.

From their development, these algorithms have since been used to solve a wide variety of real-life problems like the Travelling Salesman Problem [6–8], Job Scheduling Problem [9, 10], Pressure Vessel Design [11], Optical Buffer Design [12, 13] etc. Some of the important reasons why meta-heuristics are widely used [12] are mentioned below:

1. **Simplicity:** Meta-heuristic Algorithms are generally very simple in their concepts. They draw inspiration from the phenomena in nature or evolutionary concepts. This makes the resultant algorithms easier to understand and implement these algorithms in diverse problems.
2. **Flexibility:** Meta-heuristics are extremely flexible in the sense that they can be easily modified to suit a specific problem without any major changes in the algorithm itself.
3. **Derivation Free Mechanism:** Due to the random initialization in these algorithms, there is no need to calculate derivatives of the search spaces to find an optimum.
4. **Local Optimum Avoidance:** Unlike other algorithms, meta-heuristics fare better when they encounter a local optimum. Due to the randomness incorporated within these algorithms, they have a better chance at getting out of local optima.

It is of note that, a meta-heuristic algorithm is not suitable for all optimization problems. The NFL Theorem, or No Free Lunch Theorem which

has proved that a meta-heuristic algorithm can give very promising solutions on some optimization problems, but cannot show good performance on all optimization problems. This is the reason why new meta-heuristic algorithms and improvements of existing meta-heuristic algorithms is imperative. Therefore, it is the aim of this paper to propose a novel meta-heuristics that is inspired by the cell division behaviour of all cells. The architecture of the various agents, the iterative steps and as well as the optimization strategy in the proposed algorithm contribute to the reason of its superiority over other well known and recently proposed meta-heuristics.

2 Contributions and Novelty

The main contributions of our work are summarized below:

1. This work proposes a novel meta-heuristic optimization algorithm that works better for complex global search.
2. The proposed algorithm automatically balances exploration and exploitation innately and efficiently, better than other algorithms (Refer Section 6).
3. The proposed algorithm is able to find a near-global optimum solution in case of most multimodal problems.
4. The algorithm uses the concept of Genetic Diversity inside a colony of cells which is the reason for its efficiency and better performance.
5. The proposed algorithm can be easily adopted for any kind of optimization problems.

The proposed algorithm has taken inspiration from the process of Cell Division that happens in cells of any and all living organisms on earth, including us humans. The algorithm has the following characteristics:

- The algorithm has no total population parameter thus, the total population is variable. This way, the total computation resources that are being used over the course of running the algorithm is lesser when compared to existing algorithms.
- The Process of mitosis and meiosis govern both exploration and exploitation. Both are managed innately by these two processes; this makes the convergence of the algorithm more efficient.

- Stagnation and getting stuck at a local optima is innately avoided by adding in genetic diversity.
 1. As soon as the population of cells in the algorithm reaches the limit, a colony level disaster is simulated. This disaster not only curbs the low fitness cells however, it also introduces genetic diversity.
 2. The process of Meiosis and Mitosis also add to the genetic diversity of the colony.
- This Genetic Diversity is the reason why CDO does not get trapped into local optima and still has a good balance of Exploration and Exploitation capabilities.

Paper Organisation

The rest of this paper is organized as follows. Section 3 presents a brief description of the meta-heuristic algorithm proposed in the literature. Section 4 provides a background of the proposed methodology. Section 5 discusses the detailed descriptions of the proposed meta-heuristic algorithms. In Section 6, we describe and discuss the extensive experimental results. The results and discussion of benchmark functions, and two real applications are presented in Sections 6 Results and discussion. Section 8 discusses the possible reasons for its superiority even in Composite Benchmarks with complex landscapes and many local optima. Finally, the conclusions and future works are presented in Section 9.

3 Review of literature

Swarm Intelligence is a meta-heuristic approach that is widely used to solve optimization problems. The core of Swarm Intelligence lies in the collective behaviours of social creatures. Algorithms have been developed to solve various types of optimization problems [6, 9] including but not limited to non-linear [14], non-convex [15], multi-modal [16] or combinatorial optimization problems [17]. The two most important properties for adapting the behaviour of social creatures are self-organisation and division of labour [18–21].

1. Self-organisation: This feature defines the structure of the many organisms that form the swarm in consideration. Since there is no central authority, the global level response of the swarm is highly dependent on the local interactions of the swarm members. There are four

important characteristics that self-organisation is based upon which have been defined by Bonabeau et al [22].

- (a) Positive Feedback: Similar to backpropagation, feedback is the output of the system to the current input which acts as an input for further iterations of the meta-heuristic. This provides diversity and brings the system to a new stable state.
 - (b) Negative Feedback: This helps to stabilize the current system and compensates the effect of positive feedback.
 - (c) Fluctuations: It defines the rate or magnitude of changes in the system. As randomness helps to locate new solutions and dealing with local optimum solutions, it is considered a crucial element for meta-heuristic algorithms.
 - (d) Multiple Interactions: These interactions provide a way of learning from the other members of a swarm.
2. Division of Labour: Performance of simultaneous tasks is believed to be more efficient than sequential performance, thus division of labour marks another feature of immense importance for Swarm Intelligence[22].

Meta-heuristics are mainly classified into three categories:

1. Evolutionary Algorithms(EA)
2. Physics-based Algorithms
3. Swarm Intelligence based Algorithms

Pictorial representation of the different optimization algorithms along with their category is given in the Fig. 1.

Evolutionary Algorithms are inspired by the concept of evolution in Nature. One of the most important algorithms in this category is the Genetic Algorithm [2] which is widely used today for a plethora of problems [23–25]. This algorithm simulates concepts of Darwinian evolution. Engineering applications of the Genetic Algorithm were investigated by Goldberg [26]. In evolutionary algorithms, the optimization starts with a randomly generated solution. In each successive iteration, new members are generated with selection, crossover, and mutation operations. The solutions are evolved with the view of generating better solutions in each iteration. A few Evolutionary Algorithms are Differential Evolution [27],

Fig. 1 Types of Optimization Algorithms

Evolutionary Programming, [28, 29], Evolution Strategy [30, 31] and Genetic Programming [32]

The second type of meta-heuristics is one of the *Physics-Based algorithms* which have taken inspiration from concepts in physics like the phenomenon of . Some of the most popular algorithms are Gravitational Local Search (GLSA) [33], Big-Bang Big-Crunch (BBBC) [34], Gravitational Search Algorithm (GSA) [35], Charged System Search (CSS) [36], Central Force Optimization (CFO) [37], Artificial Chemical Reaction Optimization Algorithm (ACROA) [38], Black Hole (BH) algorithm [39], Ray Optimization (RO)

algorithm [40], Small-World Optimization Algorithm (SWOA) [41], Galaxy-based Search Algorithm (GbSA) [42], and Curved Space Optimization (CSO) [43]. The difference between Evolutionary and Physics-based algorithms is that in Physics-based algorithms a random set of agents communicate and move in the search space according to the physical rules.

The third sub-class is the one containing *Swarm Intelligence methods*. These meta-heuristics mimic the social behaviour of swarms, flocks, herds etc. The simulated individuals act as agents for exploring a search space.

Some SI meta-heuristics are:

- Particle Swarm Optimization [1]

Table 1 Applications of some Meta-Heuristic Algorithms

Algorithm	References	Applications
Particle Swarm Optimization	[44] [45] [46, 47]	Evolving Neural Network metal removal operation in manufacturing Antennas, BioMedical, Communication Networks, Clustering and Classification, Combinatorial optimisation among others
Cuckoo Search	[48, 49] [50, 51] [52–54]	Medical applications Clustering and data mining applications Image Processing, Engineering Design, Energy and Economic Load Problems among others
Grey Wolf Optimizer	[55–58] [59–61] [62–65]	Machine learning applications Engineering applications Robotics, Scheduling, Wireless sensor network, Medical among others
Artificial Bee Colony	[66, 67] [68, 69] [70–72]	Engineering applications Data Mining Bioinformatics, Clustering, Image Processing among others
Genetic Algorithm	[73, 74] [75, 76] [77–79]	Engineering Design Medical applications Image Processing, Energy Load Problems among others

- Ant Colony Optimization [80]
- Cuckoo Search [4]
- Artificial Bee Colony [3]
- Grey Wolf Optimization [12]
- Spider Monkey Optimization [18]

Tabulated summary of these algorithms and its variations is given in Table 1.

Other well-known meta-heuristics include:

- Marriage in Honey Bees Optimization Algorithm [81]
- Artificial Fish Swarm Algorithm [82]
- Termite Algorithm [83]
- Wasp Swarm Algorithm [84]
- Monkey Search [85]
- Bee Collecting Pollen [86]
- Dolphin Partner Optimization [87]
- Firefly Algorithm [88]
- Bird Mating Optimizer [89]
- Fruit Fly Algorithm [90]

- Krill Herd Optimization [91]
- Whale Optimizer [5]

4 Background

In this section the inspiration of the proposed method is first discussed. Then, the mathematical model is provided.

4.1 Cell Division

All living beings are composed of cells and Cell Division [92, 93] is the process in which a cell divides into two or more daughter cells. In all living organisms cells continuously divide and new cells replace old cells every second.

There are mainly two processes by which a cell can undergo cell division: Mitosis [93–95] and Meiosis [96, 97]. Mitosis is a form of asexual reproduction which results in the birth of two identical

daughter cells. Meiosis is a process that facilitates sexual reproduction, in which a diploid cell divides into four different haploid cells, called gametes, which further combine with other haploid cells to produce a diploid daughter cell.

4.2 Mitosis

Mitosis [93–95] is the process in which a parent cell divides into two identical daughter cells. In multi-cellular organisms, mitosis serves as a way for growth and replacing worn-out cells. Thus, mitosis is extremely crucial for life to go on. Defects formed during mitosis can potentially lead to genetic disorders. Mitosis mainly consists of 5 stages based on the physical state of the chromosomes and spindle. These phases are *prophase*, *prometaphase*, *metaphase*, *anaphase*, and *telophase*. Interphase is a stage that takes place before mitosis, hence, it is not counted as a part of mitosis.

- *Prophase*: In this stage, chromosomes recruit a protein complex called condensin and start to undergo a condensation process. As the two pairs of centrioles travel to opposite poles during prophase, the spindle begins to develop.
- *Prometaphase*: Microtubules coming from both poles of the spindle pull and tug chromosomes in opposite directions during prometaphase.
- *Metaphase*: The chromosomes align on a theoretical line called the metaphase plate during this stage. The centrosomes complete their work and are placed at the cell's opposite ends.
- *Anaphase*: Sister chromatids are pulled to opposing ends of the cell at this stage. The chromatids in the centromere are broken and moved to opposite poles of the cell as the spindle fibres compress. The cell will be elongated by spindle fibres that are not linked to chromatids in order to prepare it for division.
- *Telophase*: The cell has lengthened and is nearly completed its division at this point. Cell-like traits, such as the reconstruction of two nuclei, begin to resurface (one for each cell). The chromosomes condense and the mitotic spindle's fibres break down.

4.3 Meiosis

Meiosis [96, 97] is a type of cell division exclusive to sexually reproducing organisms. In this process,

there are two rounds of cell division which result in the birth of four daughter cells with each having half the number of chromosomes of the parent cell. These haploid cells then further fuse with haploid cells produced by other organisms to give birth to a new zygote of that organism's species. The two rounds of meiosis are termed as Meiosis I and Meiosis II.

4.3.1 Meiosis I

Meiosis I segregates homologous chromosomes and produces two haploid cells. It consists of 4 stages: *prophase I*, *metaphase I*, *anaphase I*, and *telophase I*.

- *Prophase I*: Similar to mitosis Prophase, chromosomes start to undergo a condensation process. The difference here is that the chromosomes do not pair up. This step does not take place in mitosis and is the reason why the resulting cells are haploid.
- *Metaphase I*: Here the homologous pairs of chromosomes line up on the metaphase plate, near the centre of the cell.
- *Anaphase I*: Similar to Mitosis, in this stage chromosomes are pulled towards the centrioles at the sides of the cell.
- *Telophase I*: The chromosomes are pulled apart completely in this stage and new nuclear envelopes are formed.

4.3.2 Meiosis II

This process involves *equational segregation*, i.e., separation of sister chromatids (a chromatid is one of two identical halves of a replicated chromosome). This is similar to mitosis however, the difference lies in the results. Mitosis produces two diploid offsprings (resulting cells are complete cells and contain paired chromosomes) while meiosis produces four diploid gametes (gametes are like incomplete cells, they have unpaired chromosomes). It consists of 4 stages: *prophase II*, *metaphase II*, *anaphase II*, and *telophase II*.

- *Prophase II*: Similar to Prophase I, this stage prepares for the cell division of the two haploid cells produced in Prophase I. The nuclear envelopes disappear and centrioles are formed. The chromosomes begin to get pulled toward the metaphase plate.

Fig. 2 Mitosis

- *Metaphase II*: The cycle now resembles mitosis. The chromosomes line up with their centromeres on the metaphase plate.
- *Anaphase II*: The sister chromatids separate. They are now called sister chromosomes and are pulled toward the centrioles. This separation marks the final division of Nuclear Matter.
- *Telophase II*: As in Telophase I, the cells are now divided into two and the chromosomes are at extreme ends of the cell.

4.4 Levy Flight

Levy Flight [98] is a random walk algorithm in which the step lengths have a heavy-tailed Levy Distribution. Named after the French Mathematician Paul Lévy, the term was coined by Benoît Mandelbrot. Levy flight operates under the following function.

$$Levy(\lambda) = \left| \frac{\Gamma(1 + \lambda) * \sin(\frac{\pi * \lambda}{2})}{\Pi \frac{(1+\lambda)}{2} * \lambda * 2^{\frac{\lambda-1}{2}}} \right|^{\frac{1}{\lambda}} \quad (1)$$

The equation to find the new position is:

$$X_{t+1} = X_t + S * Levy(\lambda) * \alpha \quad (2)$$

where X_t, X_{t+1} are the current and new positions respectively, S is the step size, λ is a constant ($1 \leq \lambda \leq 3$) and α is a random number generated between $[-1,1]$.

Figure 4 shows a simulation of Levy Flight in a 2-dimensional plane.

Fig. 4 Simulation of Levy Flight ¹

5 Cell Division Optimization

The algorithm proposed by this paper works by simulating a colony of cells. By Darwin's law of Survival of the fittest, the algorithm iteratively weeds out solutions with bad fitness. The algorithm starts with an initial population of cells that act as the progenitors of the colony. These cells are randomly generated within the given bounds. Each iteration in this algorithm represents the lifespan of the cells. In each iteration, the cells undergo one of the two cell divisions, Mitosis and Meiosis. In mitosis, the parent cell divides to produce two identical daughter cells. Mutation occurs in the Prophase of the mitosis process and thus mitosis becomes a very powerful tool to exploit a given region to find out a local optimum. In meiosis, the DNA of two-parent cells merges through a

¹Source: <https://en.wikipedia.org/wiki/File:LevyFlight.svg>

Fig. 3 Meiosis

crossover and thus producing an entirely new set of genes. Factoring in mutation, meiosis becomes a tool for exploration and to get out of local optima. The elementary objects of the proposed model is mentioned in the subsections.

5.1 Cells

Each cell in the colony represents two possible solutions for the given problem. The inspiration for this is taken from homologous chromosomes that occur in each of our bodies. Humans have 23 pairs of chromosomes and for each pair, there is a maternal and a paternal chromosome. Thus, each cell in the algorithm has a paternal and a maternal influence which is generated through meiosis.

5.2 Gametes

Gametes are the products of Meiosis. They are haploid, which means they contain half the number of chromosomes as the parent cells. This implies that each gamete represents one possible solution. The solutions contained in Gametes are intermediary and they are not evaluated. As Gametes are haploid, i.e., do not contain a complete solution, each gamete needs to be paired with another gamete of a different cell. These two gametes are used to get a new cell, which has equal contributions from both the gametes. Algorithm description of this model is presented in Algorithm 5.2.5.

In each iteration, cells either go for mitosis or meiosis, based on a Cell Division Probability given to the algorithm while instantiating it. This probability governs the exploration and exploitation capabilities of the proposed algorithm. A higher mitosis probability (lower chance of meiosis) focuses more on exploitation while a low mitosis probability (higher chance of meiosis) focuses more on exploration.

5.2.1 Interphase

For each cell in the colony, the two chromosomes are duplicated and stored with the cell object. These four chromosomes further go on to make the new cells. As interphase occurs before the process and prepares the cell for division, it not considered as a part of both meiosis and mitosis.

5.2.2 Mitosis

The process of mitosis starts after the interphase stage. As, for each cell the mitosis has to focus on the exploitation of a region, we rely on the mutation factor.

5.2.3 Levy Flight

Levy Flight has been used in mitosis to facilitate the random alteration capability (usually known as mutation). Breaking down Levy Flight into its bare essentials,

- There's a high probability that after doing levy flight, the object in question would be somewhere close to its original position.
- There's a very low probability that after doing levy flight, the object would cover a far greater distance and abandon the neighbourhood it was in.

The mathematical formulation of the step size of a single gene for Levy Flight is determined by:

$$step = rand(.) \times X_i \times 0.5 \quad (3)$$

where $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ represents one of the four chromosomes of the cell after Interphase. $Rand(.)$ represents a random number between 0 and 1. Levy flight is performed for each gene in all four of the chromosomes in the cell. This represents the prophase to metaphase of the mitosis process. After these steps, the chromosomes are interchanged and paired together. These two pairs are then used to create the new daughter cells. This is the anaphase and the telophase of the mitosis process.

Algorithm 1 Mitosis

```

cell ← Cell that has to undergo Mitosis
Interphase_Duplication(cell)
SET new_chromosomes to empty list
for chromosome ∈ (cell → chromosomes) do
  SET new_genes to empty list
  pbest ← xi
  for i ← (0 → length of chromosome) do
    Perform Levy Flight
  end for
  APPEND new_genes to new_chromosomes
end for
Generate 2 cells from the 2 pairs of chromosomes in
new_chromosomes

```

5.2.4 Meiosis

In each iteration, cells are chosen to perform meiosis are randomly paired together and meiosis is carried out for each of them. The gametes produced from this are then made to reproduce new cells with their pairs. Like mitosis, meiosis also requires interphase to happen. Thus again, the chromosomes are duplicated. In meiosis, a crossover of genes occurs in prophase I. This crossover is a facilitating factor for exploration in a region. After interphase duplication, two intermediate daughter cells are birthed.

These intermediary cells then further divide via Meiosis II to produce four gametes.

Algorithm 2 Meiosis

```

cell ← Cell that has to undergo Meiosis
Interphase_Duplication(cell) comment: Meiosis I
SET new_daughters to empty list
Perform Crossover of genes
Separate out the odd and even pairs from the four chromo-
somes in the cell
ADD all daughters to new_daughters comment: Meiosis II
SET gametes to empty list
for daughter_cell ∈ (new_daughters) do
  SET gamete_1 to empty list
  SET gamete_2 to empty list
  k ← random_integer(0, cell → dimensions)
  ADD daughter_cell → chromosome1 [0 → k] to
  gamete_1
  ADD daughter_cell → chromosome2 [k → length of
  chromosome1] to gamete_1
  ADD daughter_cell → chromosome2 [0 → k] to
  gamete_2
  ADD daughter_cell → chromosome1 [k → length of
  chromosome1] to gamete_2
  ADD gamete_1 and gamete_2 to gametes
end for

```

5.2.5 Reproduction of cells after Meiosis

Since the gametes produced by Meiosis are haploid, in this case, contain only one solution. For a cell that contains two solutions, two gametes must fuse to produce the zygote or the new cell.

Algorithm 3 Fusion of Gametes

```

gamete_1 ← A Gamete from a cell
gamete_2 ← A Gamete from a different cell
RETURN Cell(solution1 = gamete_1, solution2 =
gamete_2)

```

5.2.6 Population Control

Like a cell colony, the colony in this algorithm has an exponential growth rate. Since each cell divides in each iteration, the population almost doubles in each iteration. The concept of lifespan cannot be introduced in this context since the previous cell dies to produce new daughter cells.

Hence, an inspiration is taken from nature for keeping a check on the population of the colony. Natural disasters and other factors have been simulated in which only the fittest of the cells survive. The distribution of survival probability is such that the better the fitness of the cell, the higher is the survival probability.

Fig 5 represents the flowchart and Algorithm 4 represents the pseudo-code of the proposed cell division inspired optimization (CDO) algorithm.

Algorithm 4 Proposed Algorithm

```

1:  $max\_itr \leftarrow$  Number of iterations
2:  $init\_pop \leftarrow$  Initial Population
3:  $max\_pop \leftarrow$  Max Population
4:  $lower\_bounds \leftarrow$  Lower Bounds
5:  $upper\_bounds \leftarrow$  Upper Bounds
6:  $dim \leftarrow$  Dimensions
7:  $offsprings \leftarrow$  Number of Offsprings
8:  $Pro \leftarrow$  Meiosis Probability
9:  $disaster\_Pro \leftarrow$  Disaster Probability
10:  $obj\_func \leftarrow$  Function
11: SET colony to empty list
12: for  $i \leftarrow (1 \text{ to } population)$  do
13:   SET genes1 to empty list
14:   SET genes2 to empty list
15:   for  $j \leftarrow (1 \text{ to } dim)$  do
16:     Append random value in the given bounds to
     genes1
17:     Append random value in the given bounds to
     genes2
18:   end for
19:   Append genes1 and genes2 to colony
20: end for
21: SET best = 0
22: for cell  $\in$  colony do
23:   EVALUATE cell with obj_func
24:   if cell  $\rightarrow$  fitness < best then
25:     Update best
26:   end if
27: end for
28: for  $t \leftarrow (0 \text{ to } max\_itr)$  do
29:   SET new_cells to empty list
30:   SET mitosis_list to empty list
31:   SET meiosis_list to empty list
32:   for cell  $\in$  colony do
33:     Use Pro to decide if cell does Mitosis or Meiosis
34:     Add cell to mitosis_list or meiosis_list
35:   end for
36:   if Length(meiosis_list) is odd then
37:     Move random cell from mitosis_list and to
     meiosis_list
38:   end if
39:   for cell  $\in$  mitosis_list do
40:     Mitosis(cell)
41:     APPEND daughters to new_cells
42:   end for
43:   for cell1 & cell2  $\in$  meiosis_list do
44:     Meiosis(cell1, cell2)
45:     Fuse Gametes of cell1 & cell2
46:     APPEND daughters to new_cells
47:   end for
48:    $m \leftarrow$  RandomNumber(1, Length(new_cells))
49:   ADD m best cells to colony
50: end for
51: Update best
52: if Length(colony) >  $max\_pop$  then
53:   Use disaster_Pro to eliminate cells
54: end if
55: RETURN best

```

6 Experimental evaluation and results

In this section, In this section, the comparative performance of the proposed methods in comparison to the state-of-the-art algorithms is presented. All algorithms were implemented using Python 3.7.9 and performed on a 3.1 GHz Dual-Core Intel Core i5 with 8GB of memory running Mac OS 11.2.

6.1 Benchmark Functions

In this paper, the 20 classical benchmark functions and 30 benchmark functions with varying landscapes and modalities have been used for empirical analysis. The second 30 functions have been taken from CEC2017's [99]¹² repository of Benchmark Functions, including various unimodal, multimodal, simple, hybrid, and composite functions. The functions used to test the algorithm make use of various simple functions in them. Tables 2 and 3 show functions, f_1 to f_{19} which are the simple functions used by other complex functions.

Table 4 and 5 represent the 20 classical benchmark functions. Tables 6 shows basic functions, F_1 to F_{10} . Function F_2 has been declared as deprecated and thus not taken.

Tables 7 and 8 show the Functions F_{11} to F_{20} . These are hybrid Functions. In Hybrid Functions, sub-components of the variables have different properties. The mathematical formulation of these hybrid functions is given by the following equation.

$$F(\mathbf{x}) = g_1(\mathbf{M}_1 z_1) + g_2(\mathbf{M}_2 z_2) + \dots + g_N(\mathbf{M}_N z_N) + F^*(\mathbf{x}) \quad (4)$$

where,

$F(\mathbf{x})$: hybrid function

$g_i(x)$: i^{th} basic function used to construct the hybrid function

N : number of basic functions

$z = [z_1, z_2, \dots, z_N]$

$\mathbf{y} = \mathbf{x} - \mathbf{o}_i$

$S = \text{randperm}(1 : D)$

¹<https://github.com/P-N-Suganthan/CEC2017-BoundConstrained>

²<https://github.com/tilleyd/cec2017-py>

Fig. 5 Flowchart for the proposed Algorithm

p_i : used to control the percentage of $g_i(\mathbf{x})$
 n_i : dimension for each basic function
 $\sum_{i=1}^N n_i = D$
 $n_1 = [p_1 D], n_2 = [p_2 D], \dots, n_{N-1} =$

$$[p_{N-1} D], n_N = D - \sum_{i=1}^{N-1} n_i$$

The tabulated summaries of the complex functions F_{21} to F_{30} are presented in Tables 9 and 10.

The mathematical formulation of these composite functions is given by the following equation.

$$F(\mathbf{x}) = \sum_{i=1}^N \{\omega_i^* [\lambda_i g_i(\mathbf{x}) + \text{bias}_i]\} + F^* \quad (5)$$

$F(\mathbf{x})$: composition function

$g_i(\mathbf{x})$: i^{th} basic function used to construct the composition function

N : number of basic functions

o_i : new shifted optimum position for each $g_i(\mathbf{x})$, define the global and local optima's position

bias_i : defines which optimum is global optimum

σ_i : used to control each $g_i(\mathbf{x})$'s coverage range, a small σ_i give a narrow range for that $g_i(\mathbf{x})$

λ_i : used to control each $g_i(\mathbf{x})$'s height

w_i : weight value for each $g_i(\mathbf{x})$, calculated as below:

$$w_i = \frac{1}{\sqrt{\sum_{j=1}^D (x_j - o_{ij})^2}} \exp\left(-\frac{\sum_{j=1}^D (x_j - o_{ij})^2}{2D\sigma_i^2}\right)$$

Then normalize the weight $\omega_i = w_i / \sum_{i=1}^n w_i$

Fig 6 represents the plots of 12 out of all the classical benchmark functions considered. Fig 7 represents the plots of all the composite functions(F_{21} to F_{30}).

In addition to these functions, two real-world problems, pressure vessel design and frequency modulated sound wave, have been used to verify the performance of the proposed meta-heuristic algorithm for real world problems. Refer to Section 7.

The performance of the proposed algorithm is tested with the ten well-known meta-heuristic optimization algorithms Particle Swarm Optimization(PSO) (1999) [1], Cuckoo Search(CS) (2009, 2018)[4, 100], Fruit Fly Algorithm (2012) [90], Krill Herd (2012) [91], Whale Optimizer(WOA) (2016) [5], Water-Wave Optimizer(WWO) (2015) [101], Artificial Bee Colony (2007) [3], Grey Wolf Optimizer(GWO) (2014) [12], IL-SHADE (2016)[102], WCMA-ES (2020)[103]

6.2 Parameters

Algorithms considered in this paper were initialized randomly and allowed to run 10 times independently with maximum 500 iterations, while the size of the population in each algorithm is kept at 10 to 500 . Table 11 shows the parameters that govern the experiments performed. The parameters have been set up in such a way that a fair comparison of the algorithms under consideration can be made

6.3 Results and Discussion

Tables 12 and 13 represents the outputs of the considered meta-heuristics on the 20 different classical benchmark functions. The algorithms are run 10 times on each benchmark function and the best, mean and standard deviation are recorded. Similar experiments have been performed for the 30 benchmark functions from the CEC2017 repository [99].

Table 14 shows the results for the basic functions F1 to F10. Table 15 shows the results of the hybrid functions, F11 to F20. Table 16 shows the results on Composite Benchmarks, F21 to F30.

CDO offers consistently competitive results when considering classical Benchmark Functions in both the Best and the Mean values of the optima it finds.

When considering Unimodal Functions (14), CDO does not outperform the other algorithms since best optima found by CDO and the other algorithms are same. However in this case, CDO consistently gives us solutions closed to its best value, which is where the other algorithms fail. This analysis is clearly evident from the Standard Deviation of the results computed for the 10 independent run. On Schaffer2 test function, the recorded standard deviation for CDO algorithm is $0.00E + 00$, $4.53E - 06$ is recorded for PSO, $2.04E - 07$ is recorder for CS, $1.01E - 13$ is recorded for FruitFly, $2.58E - 08$ is recorded for Krill Herd, $4.33E - 04$ is recorder for WOA , and $9.25E - 03$ is recorded for WWO.

In case of Multimodal and Composite Functions (refer, Table 16, CDO excels and outperforms the other algorithms. This is due to its local optima avoiding capabilities and hence, does not get stuck in the complex landscapes of multimodal functions. Other algorithms considered get

Fig. 6 Plots of 12 Benchmark Functions **Fig. 7** Plots of Composition Functions

Table 2 Basic Functions

Function Name	Function
Bent Cigar Function	$f_1(x) = x_1^2 + 10^6 \sum_{i=2}^D x_i^2$
Zakharov Function	$f_3(\mathbf{x}) = \sum_{i=1}^D x_i^2 + \left(\sum_{i=1}^D 0.5x_i\right)^2 + \left(\sum_{i=1}^D 0.5x_i\right)^4$
Rosenbrock's Function	$f_4(\mathbf{x}) = \sum_{i=1}^{D-1} \left(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right)$
Rastrigin's Function	$f_5(\mathbf{x}) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$
Expanded Schaffer's F6 Function	$g(x, y) = 0.5 + \frac{(\sin^2(\sqrt{x^2 + y^2}) - 0.5)}{(1 + 0.001(x^2 + y^2))^2}$ $f_6(\mathbf{x}) = g(x_1, x_2) + g(x_2, x_3) + \dots + g(x_{D-1}, x_D) + g(x_D, x_1)$
Lunacek bi-Rastrigin Function	$f_7(x) = \min \left(\sum_{i=1}^D (x_i - \mu_0)^2, dD + s \sum_{i=1}^D (x_i - \mu_1)^2 \right) + 10 \left(D - \sum_{i=1}^D \cos(2\pi z_i) \right)$ $\mu_0 = 2.5, \mu_1 = -\sqrt{\frac{\mu_0^2 - d}{s}}, s = 1 - \frac{1}{2\sqrt{D+20} - 8.2}, d = 1$ $y = \frac{10(\mathbf{x} - \mathbf{o})}{100}, \hat{x}_i = 2 \operatorname{sign}(x_i^*) y_i + \mu_0, \text{ for } i = 1, 2, \dots, D$ $z = \Lambda^{100}(\mathbf{x} - \mu_0)$
Non-continuous Rotated Rastrigin's Function	$f_8(x) = \sum_{i=1}^D \left(z_i^2 - 10 \cos(2\pi z_i) + 10 \right) + f_{13*}$ $x = \mathbf{M}_1 \frac{5.12(\mathbf{x} - \mathbf{o})}{100}, y_i = \begin{cases} x_i & \text{if } x_i \leq 0.5 \\ \operatorname{round}(2x_i)/2 & \text{if } x_i > 0.5 \end{cases} \text{ for } i = 1, 2, \dots, D$ $z = \mathbf{M}_1 \Lambda^{10} \mathbf{M}_2 T_{asy}^{0.2}(T_{osz}(y))$ <p>Where Λ^α : a diagonal matrix in D dimensions with the i^{th} diagonal element as $\lambda_{ii} = \alpha^{\frac{i-1}{2(D-1)}}$</p> $i = 1, 2, \dots, D$ $T_{asy}^\beta : \text{ if } x_i > 0, x_i = x_i^{1+\beta \frac{i-1}{D-1} \sqrt{x_i}}, \text{ for } i = 1, \dots, D$ $T_{osz} : \text{ for } x_i = \operatorname{sign}(x_i) \exp(\hat{x}_i + 0.049(\sin(c_1 \hat{x}_i) + \sin(c_2 \hat{x}_i))), \text{ for } i = 1 \text{ and } D[4] \text{ where}$ $\hat{x}_i = \begin{cases} \log(x_i) & \text{if } x_i \neq 0 \\ 0 & \text{otherwise} \end{cases}, \operatorname{sign}(x_i) = \begin{cases} -1 & \text{if } x_i < 0 \\ 0 & \text{if } x_i = 0 \\ 1 & \text{otherwise} \end{cases}, c_1 = \begin{cases} 10 & \text{if } x_i > 0 \\ 5.5 & \text{otherwise} \end{cases}, \text{ and}$ $c_2 = \begin{cases} 7.9 & \text{if } x_i > 0 \\ 3.1 & \text{otherwise} \end{cases}$
Levy Function	$f_9(\mathbf{x}) = \sin^2(\pi w_1) + \sum_{i=1}^{D-1} (w_i - 1)^2 [1 + 10 \sin^2(\pi w_i + 1)] + (w_D - 1)^2 [1 + \sin^2(2\pi w_D)]$ <p>Where $w_i = 1 + \frac{x_i - 1}{4}, \forall i = 1, \dots, D$</p>
High Conditioned Elliptic Function	$f_{11}(x) = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} x_i^2$

stuck relatively easier and hence do not offer good solutions consistently.

The conducted analysis demonstrates that the CDO performs better than the other competitive methods when dealing with the functions of complex landscapes and real life problems.

Section 8 discusses the possible reasons behind the performance of the proposed algorithm.

The results on various benchmark functions show that the proposed algorithm is generally superior to other algorithms in most cases. Section 6.5 shows the results of Statistical Tests done on these results which shows the significant difference

Table 3 Basic Functions Continued

Function Name	Function
Discus Function	$f_{12}(x) = 10^6 x_1^2 + \sum_{i=2}^D x_i^2$
Ackley's Function	$f_{13}(\mathbf{x}) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e$
Modified Schwefel's Function	$f_{10}(x) = 418.9829 \times D - \sum_{i=1}^D g(z_i)$ $z_i = x_i + 4.209687462275036e + 002$ $g(z_i) = \begin{cases} z_i \sin(z_i^{1/2}) & \text{if } z_i \leq 500 \\ (500 - \text{mod}(z_i, 500)) \sin \left(\sqrt{500 - \text{mod}(z_i, 500)} \right) - \frac{(z_i - 500)^2}{10000D} & \text{if } z_i > 500 \\ (\text{mod}(z_i, 500) - 500) \sin \left(\sqrt{\text{mod}(z_i, 500) - 500} \right) - \frac{(z_i + 500)^2}{10000D} & \text{if } z_i < -500 \end{cases}$
Weierstrass Function	$f_{14}(\mathbf{x}) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{\max}} \left[a^k \cos(2\pi b^k (x_i + 0.5)) \right] \right) - D \sum_{k=0}^{k_{\max}} \left[a^k \cos(2\pi b^k \cdot 0.5) \right]$ $a = 0.5, b = 3, k_{\max} = 20$
Griewank's Function	$f_{15}(\mathbf{x}) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$
Katsuura Function	$f_{16}(\mathbf{x}) = \frac{10}{D^2} \prod_{i=1}^D \left(1 + i \sum_{j=1}^{32} \frac{2^j x_i - \text{round}(2^j x_i)}{2^j} \right)^{\frac{10}{D^{1.2}}}$
HappyCat Function	$f_{17}(\mathbf{x}) = \sum_{i=1}^D x_i^2 - D^{1/4} + \left(0.5 \sum_{i=1}^D x_i^2 + \sum_{i=1}^D x_i \right) / D + 0.5$
HGBat Function	$f_{18}(x) = \left \left(\sum_{i=1}^D x_i^2 \right)^2 - \left(\sum_{i=1}^D x_i \right)^{1/2} + \left(0.5 \sum_{i=1}^D x_i^2 + \sum_{i=1}^D x_i \right) / D + 0.5 \right $
Expanded Rosenbrock's Function	$f_{19}(x) = f_7(f_4(x_1, x_2)) + f_7(f_4(x_2, x_3)) + \dots + f_7(f_4(x_{D-1}, x_D)) + f_7(f_4(x_D, x_1))$
Schaffer's F7 Function	$f_{20}(\mathbf{x}) = \left[\frac{1}{D-1} \sum_{i=1}^{D-1} \left(\sqrt{s_i} \cdot \left(\sin(50.0 s_i^{0.2}) + 1 \right) \right) \right]^2$ $s_i = \sqrt{x_i^2 + x_{i+1}^2}$

Table 4 Benchmark Functions

Name	Function	Type	Domain	Minimum
Sphere	$f(x_1 \cdots x_n) = \sum_{i=1}^n x_i^2$	Unimodal Continuous	$-100 \leq x_i \leq 100$	$f(0, \cdots, 0) = 0$
Ackley	$f(\mathbf{x}) = f(x_0 \cdots x_n) = -20e^{(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2})} - \exp(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	Multimodal Continuous	$-32 \leq x_i \leq 32$	$f(0, \cdots, 0) = 0$
Schwefel	$f(\mathbf{x}) = f(x_1 \cdots x_n) = \sum_{i=1}^n (-x_i \sin(\sqrt{x_i})) + \alpha \cdot n$	Multimodal Non-Separable	$\alpha = 418.982887$ $-512 \leq x_i \leq 512$	$f(\mathbf{x}^*) = 0$ $\mathbf{x}^* = (420.968 \cdots 420.968)$
Schwefel 2.22	$f(\mathbf{x}) = f(x_0 \cdots x_n) = \sum_{i=0}^n x_i + \prod_{i=0}^n x_i$	Multimodal Separable	$-10 \leq x_i \leq 10$	$f(0, \cdots, 0) = 0$
Xin She Yang N2	$f(\mathbf{x}) = \frac{f(x_1, \dots, x_n)}{(\sum_{i=1}^n x_i) \exp(-\sum_{i=1}^n \sin(x_i^2))}$	Multimodal Non-Separable	$-2\pi \leq x_i \leq 2\pi$	$f(0, \cdots, 0) = 0$
Rosenbrock	$f(\mathbf{x}) = f(x_1 \cdots x_n) = \sum_{i=1}^{n-1} (100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2)$	Multimodal Non-Separable	$-100 \leq x_i \leq 100$	$f(1, 1, \cdots, 1) = 0$
Griewank	$f(\mathbf{x}) = f(x_1, \dots, x_n) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}})$	Unimodal Non-Separable	$-600 \leq x_i \leq 600$	$f(0, 0, \cdots, 0) = 0$
Levy13	$f(x, y) = \frac{\sin^2(3\pi x) + (x-1)^2(1 + \sin^2(3\pi y))}{1^2(1 + \sin^2(2\pi y))} + (y - 1)^2(1 + \sin^2(2\pi y))$	Multimodal Non-Separable	$-10 \leq x \leq 10$ $-10 \leq y \leq 10$	$f(1, 1) = 0$
Rastrigin	$f(\mathbf{x}) = f(x_1 \cdots x_n) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$	Multimodal Separable	$-5.12 \leq x_i \leq 5.12$	$f(0, \cdots, 0) = 0$
Easom	$f(x, y) = -\cos x \cos y \times \exp(-(x - \pi)^2 - (y - \pi)^2)$	Multimodal Separable	$-100 \leq x \leq 100$ $-100 \leq y \leq 100$	$f(\pi, \pi) = -1$
Three-Hump Camel	$f(x, y) = 2x^2 - 1.05x^4 + \frac{x^6}{6} + xy + y^2$	Unimodal Non-Separable	$-5 \leq x_i \leq 5$	$f(0, \cdots, 0) = 0$

Table 5 Benchmark Functions Continued

Name	Function	Type	Domain	Minimum
Bukin 6	$f(x, y) = 100\sqrt{y-0.01x^2} + 0.01 \times x + 10$	Multimodal Non-Separable	$-15 \leq x \leq 5$ $-3 \leq y \leq -3$	$f(-10, 1) = 0$
Shubert	$f(x_1, \dots, x_n) = \prod_{i=1}^n \left(\sum_{j=1}^5 \cos((j+1)x_i + j) \right)$	Multimodal Non-Separable	$-10 \leq x_i \leq 10$	$f(\mathbf{x}^*) = 186.7309$
Six-Hump Camel	$f(x_1, x_2) = (4 - 2.1 \times x_1^2 + \frac{x_1^4}{3}) \times x_1^2 + x_1 \times x_2 + (-4 + 4x_2^2) \times x_2^2$	Multimodal Non-separable.	$-3 \leq x_1 \leq 3$ $-2 \leq x_2 \leq 2$	$f(0.0898, -0.7126) =$ $f(-0.0898, 0.7126) =$ -1.0316
Schaffer2	$f(x, y) = 0.5 + \frac{\sin^2(x^2 - y^2) - 0.5}{(1 + 0.001(x^2 + y^2))^2}$	Unimodal Non-Separable	$-100 \leq x_i \leq 100$ $i = 1, 2$	$f(0, 0) = 0$
Stybtang	$f(\mathbf{x}) = f(x_1, \dots, x_n) = \frac{1}{2} \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i)$	Multimodal Continuous	$-5 \leq x_i \leq 5$	$f(-2.903534 \dots$ $2.903534) = -39.16599n$
Michalewicz	$f(x) = -\sum_{i=1}^d \sin(x_i) \sin^{2m} \left(\frac{ix_i^2}{\pi} \right)$	Multimodal Separable	$0 \leq x_i \leq \pi$	$f(2.2, 1.57) = -1.8013$
Holder table	$f(x, y) = -\sin(x) \cos(y) \exp \left(1 - \frac{\sqrt{x^2 + y^2}}{\pi} \right)$	Multimodal Non-Separable	$-10 \leq x \leq 10$ $-10 \leq y \leq 10$	$f(\pm 8.05502, \pm 9.66459) =$ -1.8013
Egg Holder	$f(x, y) = -(y + 47) \sin \sqrt{x} \sin \sqrt{y + 47} - x \sin \sqrt{x} \sin \sqrt{y + 47}$	Multimodal Non-Separable	$-512 \leq x_i \leq 512$	$f(512, 404.2319) =$ -959.6406627106155
Crossit	$f(x, y) = -0.0001(\sin(x) \sin(y) \exp(100 - \frac{\sqrt{x^2 + y^2}}{\pi}) + 1)^{0.1}$	Multimodal Non-Separable	$-10 \leq x, y \leq 10$	$f(\pm 1.34941, \pm 1.34941) =$ -2.06261

Table 6 Basic Functions

Function	Type	Function Name	Function	Domain	Minimum
F1	Unimodal	Shifted and Rotated Bent Cigar Function	$F_1(\mathbf{x}) = f_1(\mathbf{M}(\mathbf{x} - \mathbf{o}_1)) + F_1^*$	$[-100, 100]^D$	100
F3	Unimodal	Shifted and Rotated Zakharov Function	$F_3(\mathbf{x}) = f_3(\mathbf{M}(\mathbf{x} - \mathbf{o}_3)) + F_3^*$	$[-100, 100]^D$	200
F4	Simple Multimodal	Shifted and Rotated Rosenbrock's Function	$F_4(\mathbf{x}) = f_4\left(\mathbf{M}\left(\frac{2.048(\mathbf{x} - \mathbf{o}_4)}{100}\right) + 1\right) + F_4^*$	$[-100, 100]^D$	300
F5	Simple Multimodal	Shifted and Rotated Rastrigin's Function	$F_5(\mathbf{x}) = f_5(\mathbf{M}(\mathbf{x} - \mathbf{o}_5)) + F_5^*$	$[-100, 100]^D$	400
F6	Simple Multimodal	Shifted and Rotated Schaffer's F7 Function	$F_6(\mathbf{x}) = f_{20}\left(\mathbf{M}\left(\frac{0.5(\mathbf{x} - \mathbf{o}_6)}{100}\right)\right) + F_6^*$	$[-100, 100]^D$	500
F7	Simple Multimodal	Shifted and Rotated Lunacek Bi-Rastrigin's Function	$F_7(\mathbf{x}) = f_7\left(\mathbf{M}\left(\frac{600(\mathbf{x} - \mathbf{o}_7)}{100}\right)\right) + F_7^*$	$[-100, 100]^D$	600
F8	Simple Multimodal	Shifted and Rotated Non-Continuous Rastrigin's Function	$F_8(\mathbf{x}) = f_8\left(\frac{5.12(\mathbf{x} - \mathbf{o}_8)}{100}\right) + F_8^*$	$[-100, 100]^D$	700
F9	Simple Multimodal	Shifted and Rotated Levy Function	$F_9(\mathbf{x}) = f_9\left(\mathbf{M}\left(\frac{5.12(\mathbf{x} - \mathbf{o}_9)}{100}\right)\right) + F_9^*$	$[-100, 100]^D$	800
F10	Simple Multimodal	Shifted and Rotated Schwefel's Function	$F_{10}(\mathbf{x}) = f_{10}\left(\frac{1000(\mathbf{x} - \mathbf{o}_{10})}{100}\right) + F_{10}^*$	$[-100, 100]^D$	900

between the proposed and currently used algorithms. Section 8 explores the reasons why CDO is performing better based on its inspiration and implementation.

6.4 Convergence Evaluation

Convergence is evaluated by recording the progress of each algorithm in each iteration. Horizontal lines in the graphs represent the algorithm not progressing. Having no progress for multiple consecutive iterations implies that the algorithm

is stuck in a local optima. Fig 8 and Fig 9 show the convergence characteristics of the algorithms in comparison to the proposed CDO algorithm on Ackley and Xin She Yang N2 benchmark functions.

It can be observed in these two cases that except Cuckoo Search and CDO other algorithms considered converge before the earlier. In both cases, CDO consistently improves without horizontal lines, which implies it improves itself in

Table 7 Hybrid Benchmark Functions

Function	Function	Domain	Minimum
F11	$N = 3$ $p = [0.2, 0.4, 0.4]$ g_1 : Zakharov Function f_3 g_2 : Rosenbrock Function f_4 g_3 : Rastrigin's Function f_5	$[-100, 100]^D$	1000
F12	$N = 3$ $p = [0.3, 0.3, 0.4]$ g_1 : High Conditioned Elliptic Function f_{11} g_2 : Modified Schwefel's Function f_{10} g_3 : Bent Cigar Function f_1	$[-100, 100]^D$	1100
F13	$N = 3$ $p = [0.3, 0.3, 0.4]$ g_1 : Bent Cigar Function f_1 g_2 : Rosenbrock Function f_4 g_3 : Lunache Bi-Rastrigin Function f_7	$[-100, 100]^D$	1200
F14	$N = 4$ $p = [0.2, 0.2, 0.2, 0.4]$ g_1 : High Conditioned Elliptic Function f_{11} g_2 : Ackley's Function f_{13} g_3 : Schaffer's F7 Function f_{20} g_4 : Rastrigin's Function f_5	$[-100, 100]^D$	1300
F15	$N = 4$ $p = [0.2, 0.2, 0.3, 0.3]$ g_1 : Expanded Schaffer F6 Function f_6 g_2 : HGBat Function f_{18} g_3 : Rosenbrock's Function f_4 g_4 : Modified Schwefel's Function f_{10}	$[-100, 100]^D$	1400
F16	$N = 4$ $p = [0.2, 0.2, 0.3, 0.3]$ g_1 : Expanded Schaffer F6 Function f_6 g_2 : HGBat Function f_{18} g_3 : Rosenbrock's Function f_4 g_4 : Modified Schwefel's Function f_{10}	$[-100, 100]^D$	1500
F17	$N = 5$ $p = [0.1, 0.2, 0.2, 0.2, 0.3]$ g_1 : Katsuura Function f_{16} g_2 : Ackley's Function f_{13} g_3 : Expanded Griewank's plus Rosenbrock's Function f_{19} g_4 : Modified Schwefel's Function f_{10} g_5 : Rastrigin's Function f_5	$[-100, 100]^D$	1600

every iteration. It can be inferred from this analysis that the CDO can easily avoid getting stuck in local optima and does not converge prematurely.

Similar performance is recorded on the other test functions also (Fig. 10, 11, 12 and 13).

6.5 Statistical Significance

It is important to compare the statistical significance of the data collected using the proposed algorithm to the algorithms considered in its competition. Student's T-Test has been used to determine this importance. This statistical test is used to assess if two population means differ where the variances are known and the sample size is high. The results are listed in Table 17

and 18. These scores have been calculated with a confidence level ($\alpha = 0.95$). P-values have been calculated for each pair among the considered variants of PSO. P-values less than 0.05 represent that the difference between the means of the two algorithms is statistically significant. Here, + denotes the proposed algorithm performs significantly better than the compared algorithm on the other hand – denotes the algorithm is not significantly better than the competitive methods.

On CEC2017 benchmark function, for F1 function CDO performs better than the all competitive methods except Krill Herd. For F14 and F27, it perform better than the all competitive methods. As shown in tables 17 and 18, CDO shows

Table 8 Hybrid Benchmark Functions Continued

Function	Function	Domain	Minimum
F18	$N = 5$ $p = [0.2, 0.2, 0.2, 0.2, 0.2]$ g_1 : High Conditioned Elliptic Function f_1 g_2 : Ackley's Function f_{13} g_3 : Rastrigin's Function f_5 g_4 : HGBat Function f_{18} g_4 : Discus Function f_{12}	$[-100, 100]^D$	1700
F19	$N = 5$ $p = [0.2, 0.2, 0.2, 0.2, 0.2]$ g_1 : Bent Cigar Function f_1 g_2 : Rastrigin's Function f_5 g_3 : Expanded Griewank's plus Rosenbrock's Function f_{19} g_4 : Weierstrass Function f_{14} g_5 : Expanded Schaffer's F6 Function f_6	$[-100, 100]^D$	1800
F20	$N = 6$ $p = [0.1, 0.1, 0.2, 0.2, 0.2, 0.2]$ g_1 : Happycat Function f_{17} g_2 : Katsuura Function f_{16} g_3 : Ackley's Function f_{13} g_4 : Rastrigin's Function f_5 g_5 : Modified Schwefel's Function f_{10} g_6 : Schaffer's F7 Function f_{20}	$[-100, 100]^D$	1900

a significant improvement in comparison to the competitive methods in most of the cases.

7 Real-Life Applications

In this section, the proposed algorithm is evaluated in solving two constrained real engineering problems, namely pressure vessel design problem and frequency modulated sound wave problem. These problems contain several inequality constraints. For both these problems, all algorithms have been run for 500 iterations and the same parameters as defined in Section 6.2.

7.1 Pressure Vessel Design

The pressure vessel design problem is to minimize the total cost of material, forming and welding of a cylindrical vessel [104]. There are four design variables.

1. $x_1 \rightarrow T_s$, shell thickness
2. $x_2 \rightarrow T_h$, spherical head thickness
3. $x_3 \rightarrow R$, Radius of cylindrical shell
4. $x_4 \rightarrow L$, shell length

The mathematical formulation becomes:

$$f(X) = 0.6224 \times x_1 \times x_3 \times x_4 + 1.7781 \times x_2 \times x_3^2 + 3.1611 \times x_1^2 \times x_4 + 19.84 \times x_1^2 \times x_3$$

subject to

$$g_1(X) = 0.0193 \times x_3 - x_1 \leq 0 \quad (6)$$

$$g_2(X) = 0.00954 \times x_3 - x_3 \leq 0 \quad (7)$$

$$g_3(X) = 750 \times 1728 - \pi \times x_3^2 \left(x_4 + \frac{4}{3} \times x_3 \right) \leq 0 \quad (8)$$

Table 19 shows the results of the algorithms on the Pressure Vessel Optimization Problem. The performance of the proposed algorithm is tested in comparison to the recently proposed algorithms Warm Starting CMA-ES [103] and iL-SHADE [102]. It is evident from Table 19 that the CDO gives results at par with the currently used algorithms however the consistency of the algorithm are better than the algorithm.

7.2 Frequency-Modulated sound wave

FM Sound Wave synthesis has an important role in the modern music industry and the problem here is to optimize the six parameters governing this which are $[a_1, a_2, a_3, w_1, w_2, w_3]$.

The problem is to generate a sound similar to a given sound. The expressions of the estimated sound and the target sound wave are given as:

Table 9 Composite Functions

Function	Function	Domain	Minimum
F21	$N = 3$ $\sigma = [10, 20, 30]$ $\lambda = [1, 1e - 6, 1]$ $\text{bias} = [0, 100, 200]$ g_1 : Rosenbrock's Function F_4 g_2 : High Conditioned Elliptic Function F_{11} g_3 Rastrigin's Function F_4 ?	$[-100, 100]^D$	2000
F22	$N = 3$ $\sigma = [10, 20, 30]$ $\lambda = [1, 10, 1]$ $\text{bias} = [0, 100, 200]$ g_1 : Rastrigin's Function F_5 g_2 : Griewank's Function F_{15} g_3 Modified Schwefel's Function F_{10}	$[-100, 100]^D$	2100
F23	$N = 4$ $\sigma = [10, 20, 30, 40]$ $\lambda = [1, 10, 1, 1]$ $\text{bias} = [0, 100, 200, 300]$ g_1 : Rosenbrock's Function F_4 g_2 : Ackley's Function F_{13} g_3 : Modified Schwefel's Function F_{10} g_4 : Rastrigin's Function F_5 ,	$[-100, 100]^D$	2200
F24	$N = 4$ $\sigma = [10, 20, 30, 40]$ $\lambda = [10, 1e - 6, 10, 1]$ $\text{bias} = [0, 100, 200, 300]$ g_1 : Ackley's Function F_{13} g_2 : High Conditioned Elliptic Function F_{11} g_3 : Griewank Function F_{15} g_4 : Rastrigin's Function F_5	$[-100, 100]^D$	2300
F25	$N = 5$ $\sigma = [10, 20, 30, 40, 50]$ $\lambda = [10, 1, 10, 1e - 6, 1]$ $\text{bias} = [0, 100, 200, 300, 400]$ g_1 : Rastrigin's Function F_5 g_2 : Happycat Function F_{17} g_3 : Ackley Function F_{13} ? g_4 : Discus Function F_{12} g_5 : Rosenbrock's Function F_4 ,	$[-100, 100]^D$	2400

$$y(t) = a_1 \times \sin(w_1 \times t \times \theta) + a_2 \times \sin(w_2 \times t \times \theta + a_3 \times \sin(w_3 \times t \times \theta))$$

$$y_0(t) = (1.0) \times \sin((5.0)t \times \theta - (1.5) \times \sin((4.8) \times t \times \theta + (2.0) \times \sin((4.9) \times t \times \theta))) \quad (9)$$

where $\theta = 2\pi/100$ and the domain of the variables is $[-6.4, 6.35]$.

The final function to optimize becomes:

$$f(X) = \sum_{i=0}^{100} (y(t) - y_0(t))^2 \quad (10)$$

Table 20 shows the results of the algorithms for FM Sound Wave Optimization. CDO gives a better Best and Mean value when compared with currently used algorithms. The iL-SHADE algorithm has the same best value as CDO but CDO outperforms it in terms of the mean and standard deviation values.

8 Discussion

8.1 Exploration and exploitation evaluation

Unimodal Benchmark functions provide suitable verification for the exploitation ability of the algorithm to find an optimal solution. To verify the

Table 10 Composite Functions Continued

Function	Function	Domain	Minimum
F26	$N = 5$ $\sigma = [10, 20, 20, 30, 40]$ $\lambda = [1e - 26, 10, 1e - 6, 10, 5e - 4]$ $\text{bias} = [0, 100, 200, 300, 400]$ g_1 : Expanded Scaffer's F6 Function F_6 g_2 : Modified Schwefel's Function F_{10} g_3 : Griewank's Function F_{15} g_4 : Rosenbrock's Function F_4 g_5 : Rastrigin's Function F_5	$[-100, 100]^D$	2500
F27	$N = 6$ $\sigma = [10, 20, 30, 40, 50, 60]$ $\lambda = [10, 10, 2.5, 1e - 26, 1e - 6, 5e - 4]$ $\text{bias} = [0, 100, 200, 300, 400, 500]$ g_1 : HGBat Function F_{18} g_2 : Rastrigin's Function F_5 g_3 : Modified Schwefel's Function F_{10} g_4 : Bent-Cigar Function F_{11} g_4 : High Conditioned Elliptic Function F_{11} g_5 : Expanded Scaffer's F6 Function F_6	$[-100, 100]^D$	2600
F28	$N = 6$ $\sigma = [10, 20, 30, 40, 50, 60]$ $\lambda = [10, 10, 1e - 6, 1, 1, 5e - 4]$ $\text{bias} = [0, 100, 200, 300, 400, 500]$ g_1 : Ackley's Function F_{13} g_2 : Griewank's Function F_{15} g_3 : Discus Function F_{12} g_4 : Rosenbrock's Function F_4 g_4 : HappyCat Function F_{17} g_5 : Expanded Scaffer's F6 Function F_6	$[-100, 100]^D$	2700
F29	$N = 3$ $\sigma = [10, 30, 50]$ $\lambda = [1, 1, 1]$ $\text{bias} = [0, 100, 200]$ g_1 : Hybrid Function $5F_5$ g_2 : Hybrid Function $6F_6$ g_3 : Hybrid Function $7F_7$	$[-100, 100]^D$	2800
F30	$N = 3$ $\sigma = [10, 30, 50]$ $\lambda = [1, 1, 1]$ $\text{bias} = [0, 100, 200]$ g_1 : Hybrid Function $5F_5$ g_2 : Hybrid Function $8F_8$ g_3 : Hybrid Function $9F_9$	$[-100, 100]^D$	2900

Table 11 Parameters used in algorithms

Method	Parameters
PSO	$c1=c2=2$
CS	$P_a = 0.5$
	$\alpha = 0.3$
GWO	a decreases linearly from 2 to 0
WWO	$\alpha = 1.0026$
	$hmax = 6$
KH	$Nmax = 0.01$
	$V_f = 0.02$
	$D_{max} = 0.002$
	$C_t=0.93$
	$w_n=0.42$
	$w_f=0.38$
	$nn=5$
	Crossover rate=0.2
	Mutation Rate=0.05
WOA	$a = [2 \ 0]$, $b = 1$

Table 12 Results on Classical Benchmark Functions

		CDO	PSO	CS	FruitFly	Krill Herd	WOA	WVO
Ackley	Best	4.44E-16	1.89E-02	2.07E-11	3.24E-04	9.95E-04	4.44E-16	1.12E-05
	Mean	2.22E-15	8.18E-02	2.50E-09	3.29E-04	2.62E-03	3.29E-15	1.50E+00
	STD	4.51E-15	5.54E-02	4.62E-09	3.25E-06	1.54E-03	2.80E-15	1.44E+00
Sphere	Best	2.11E-40	1.52E-05	1.75E-42	1.40E-09	6.54E-09	4.77E-52	1.27E-07
	Mean	5.27E-38	8.13E-04	4.43E-38	1.45E-09	1.09E-07	4.19E-44	3.00E-01
	STD	1.04E-37	1.27E-03	9.28E-38	3.48E-11	1.34E-07	1.32E-43	5.19E-01
Schwefel	Best	1.29E-03	1.80E-04	2.55E-05	1.39E-09	2.55E-05	2.61E-04	1.44E-03
	Mean	7.55E-03	1.02E+02	2.55E-05	1.43E-09	1.44E+02	5.93E+01	5.74E-01
	STD	4.69E-03	9.76E+01	0.00E+00	2.46E-11	6.94E+01	6.24E+01	6.05E-01
Schwefel 2.22	Best	1.41E-21	4.10E-03	3.86E-21	5.34E-06	3.60E-05	2.84E-08	4.56E-05
	Mean	2.54E-20	2.00E-02	2.38E-20	5.39E-06	1.11E-03	2.36E-07	1.03E+00
	STD	6.45E-20	1.22E-02	2.07E-20	3.44E-08	9.24E-04	2.89E-07	1.26E+00
Xin She Yang	Best	6.31E-22	5.53E-03	1.34E-19	1.04E-03	2.84E-08	1.06E-40	5.77E-07
	Mean	4.32E-19	2.08E-02	1.65E-16	1.08E-03	2.36E-07	3.30E-02	1.12E-02
	STD	1.29E-18	1.53E-02	3.59E-16	1.54E-05	2.89E-07	1.04E-01	1.93E-02
Rosenbrock	Best	4.47E-06	2.95E-04	1.40E-09	6.03E-01	2.83E-03	5.75E-09	2.03E-05
	Mean	3.55E-05	9.15E-01	4.40E-03	6.72E-01	2.74E-02	3.63E-05	1.76E-03
	STD	3.54E-05	2.86E+00	8.75E-03	3.51E-02	2.54E-02	3.62E-05	2.74E-03
Griewank	Best	0.00E+00	1.56E-04	1.48E-14	1.39E-11	2.31E-04	7.40E-03	9.85E-05
	Mean	8.07E-03	4.73E-03	1.76E-04	1.45E-11	2.82E-03	2.49E-02	2.05E-01
	STD	4.77E-03	4.11E-03	5.54E-04	4.87E-13	2.97E-03	1.15E-02	1.55E-01
Levy13	Best	5.87E-08	8.70E-04	4.48E-04	1.46E-01	9.24E-07	8.07E-06	6.23E-08
	Mean	3.07E-06	1.04E-02	3.31E-03	6.56E-01	3.33E-06	4.64E-04	6.45E-02
	STD	2.45E-06	1.28E-02	4.19E-03	3.62E-01	3.70E-07	8.55E-04	7.00E-02
Rastrigin	Best	0.00E+00	1.63E-02	0.00E+00	1.04E-04	1.62E-09	0.00E+00	9.22E-07
	Mean	0.00E+00	2.38E-01	0.00E+00	1.09E-04	4.00E-01	9.53E-08	2.12E-01
	STD	0.00E+00	2.42E-01	0.00E+00	3.88E-06	6.45E-01	2.91E-07	4.16E-01
Easom	Best	-1.00E+00	-1.00E+00	-1.00E+00	-2.25E-06	2.84E-08	-1.00E+00	-1.00E+00
	Mean	-1.00E+00	-9.99E-01	-9.99E-01	-5.66E-07	2.36E-07	-9.00E-01	-9.49E-01
	STD	6.73E-06	8.96E-04	2.18E-03	6.06E-07	2.89E-07	3.16E-01	1.10E-01

Table 13 Results on Classical Benchmark Functions

		CDO	PSO	CS	FruitFly	Krill Herd	WOA	WVO
Camel3	Best	1.11E-83	1.86E-06	3.60E-41	1.07E-06	4.66E-12	1.36E-47	1.03E-06
	Mean	2.62E-43	5.58E-04	8.10E-37	1.14E-06	7.34E-11	2.99E-02	2.23E-03
	STD	8.29E-43	6.48E-04	2.46E-36	3.67E-08	7.86E-11	9.44E-02	4.57E-03
Bukin6	Best	3.45E-02	1.10E-01	5.23E-02	3.26E-01	3.98E-02	1.48E-02	8.94E-04
	Mean	1.20E-02	1.58E+00	1.14E-01	1.09E+00	1.57E-01	5.27E-02	2.48E-02
	STD	4.97E-02	8.24E-01	4.43E-02	6.78E-01	9.07E-02	2.14E-02	3.16E-02
Shubert	Best	-1.87E+02	-1.87E+02	-1.87E+02	-6.16E+01	-1.87E+02	-1.87E+02	-1.87E+02
	Mean	-1.87E+02	-1.84E+02	-1.87E+02	-2.53E+01	-1.87E+02	-1.87E+02	-1.82E+02
	STD	8.97E-04	2.48E+00	2.86E-03	1.45E+01	2.05E-05	4.81E-02	6.17E+00
Camel6	Best	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	Mean	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	STD	1.42E-07	4.66E-03	0.00E+00	5.24E-03	2.33E-09	4.66E-08	1.26E-03
Schaffer2	Best	0.00E+00	6.62E-08	2.36E-11	3.87E-12	1.47E-09	0.00E+00	1.48E-06
	Mean	0.00E+00	3.48E-06	1.24E-07	4.01E-12	2.94E-08	1.37E-04	5.41E-03
	STD	0.00E+00	4.53E-06	2.04E-07	1.01E-13	2.58E-08	4.33E-04	9.25E-03
Stybtang	Best	-7.83E+01	-7.83E+01	-7.83E+01	-4.90E+01	-7.83E+01	-7.83E+01	-7.83E+01
	Mean	-7.83E+01	-7.83E+01	-7.83E+01	-3.59E+01	-7.82E+01	-7.83E+01	-7.83E+01
	STD	3.81E-05	2.49E-02	0.00E+00	6.18E+00	3.77E-01	6.35E-04	4.01E-02
CrossIt	Best	-2.06E+00	-2.06E+00	-2.06E+00	-2.06E+00	-2.06E+00	-2.06E+00	-2.06E+00
	Mean	-2.06E+00	-2.06E+00	-2.06E+00	-2.00E+00	-2.06E+00	-2.06E+00	-2.06E+00
	STD	4.14E-08	1.13E-04	1.09E-13	3.72E-02	1.27E-09	7.61E-10	4.72E-05
Holder Table	Best	-1.92E+01	-1.92E+01	-1.92E+01	-4.41E+00	-1.92E+01	-1.92E+01	-1.92E+01
	Mean	-1.92E+01	-1.90E+01	-1.92E+01	-2.52E+00	-1.69E+01	-1.92E+01	-1.92E+01
	STD	1.04E-04	5.00E-01	3.21E-10	1.18E+00	3.84E+00	6.82E-07	4.07E-02
Egg Holder	Best	-9.59E+02	-8.89E+02	-9.60E+02	-2.72E+01	-9.04E+02	-9.60E+02	-9.44E+02
	Mean	-9.59E+02	-6.56E+02	-9.60E+02	-2.63E+01	-6.95E+02	-9.29E+02	-9.23E+02
	STD	3.70E-01	1.45E+02	7.15E-04	5.07E-01	1.38E+02	3.45E+01	2.46E+01
Michalewicz	Best	-1.80E+00	-1.80E+00	-1.80E+00	-6.76E-01	-1.80E+00	-1.80E+00	-1.80E+00
	Mean	-1.80E+00	-1.79E+00	-1.80E+00	-2.49E-01	-1.80E+00	-1.80E+00	-1.80E+00
	STD	1.42E-05	6.87E-03	0.00E+00	2.55E-01	2.14E-09	2.15E-05	1.14E-03

Table 14 Results of Basic Benchmark Functions F1 to F10

Function	CDO	PSO	WOA	CS	GWO	FruitFly	SSO	KrillHerd	WVO	ABC
F1	Best	1.01E+02	1.01E+02	1.90E+02	1.01E+02	1.01E+02	1.01E+02	1.00E+02	1.45E+02	1.01E+02
	Mean	1.19E+02	4.12E+02	2.47E+03	1.00E+02	2.19E+03	7.50E+10	7.50E+10	1.60E+02	1.34E+03
	STD	1.37E+01	3.79E+02	2.23E+03	4.64E-11	2.08E+03	5.77E+02	2.33E+03	1.02E+02	1.52E+03
F2	Best	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02
	Mean	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02
	STD	1.11E-05	4.17E-03	3.49E-02	4.66E-13	2.14E-04	1.55E+01	6.30E-06	1.49E-05	1.45E-01
F3	Best	3.00E+02	3.00E+02	3.00E+02	3.00E+02	3.00E+02	3.00E+02	3.00E+02	3.00E+02	3.00E+02
	Mean	3.00E+02	3.00E+02	3.00E+02	3.00E+02	3.00E+02	4.87E+06	3.00E+02	3.00E+02	3.00E+02
	STD	1.90E-06	7.80E-04	6.67E-07	1.59E-07	2.07E-05	2.47E+05	4.24E-07	4.28E+00	6.44E-09
F4	Best	4.00E+02	4.00E+02	4.00E+02	4.00E+02	4.00E+02	4.00E+02	4.00E+02	4.00E+02	4.00E+02
	Mean	4.00E+02	4.00E+02	4.00E+02	4.00E+02	4.00E+02	4.74E+02	4.00E+02	4.00E+02	4.00E+02
	STD	2.14E-06	3.49E-06	4.80E-05	0.00E+00	1.83E-03	2.11E+00	1.09E-06	1.26E-02	4.53E-03
F5	Best	5.00E+02	5.00E+02	5.00E+02	5.00E+02	5.00E+02	5.00E+02	5.00E+02	5.00E+02	5.00E+02
	Mean	5.00E+02	5.00E+02	5.01E+02	5.00E+02	5.00E+02	5.14E+02	5.00E+02	5.00E+02	5.00E+02
	STD	1.29E-04	5.14E-01	5.14E-01	0.00E+00	3.15E-01	4.59E-07	0.00E+00	1.59E-08	0.00E+00
F6	Best	6.00E+02	6.00E+02	6.00E+02	6.00E+02	6.00E+02	6.00E+02	6.00E+02	6.00E+02	6.00E+02
	Mean	6.00E+02	6.00E+02	6.00E+02	6.00E+02	6.00E+02	8.08E+02	6.00E+02	6.01E+02	6.00E+02
	STD	1.12E-02	1.45E-02	3.33E-02	5.01E-11	5.63E-03	8.34E+01	1.27E-03	1.20E+00	7.72E-06
F7	Best	7.00E+02	7.00E+02	7.01E+02	7.00E+02	7.00E+02	7.02E+02	7.01E+02	7.01E+02	7.00E+02
	Mean	7.00E+02	7.02E+02	7.02E+02	7.00E+02	7.02E+02	7.02E+02	7.02E+02	7.02E+02	7.01E+02
	STD	1.02E-01	8.56E-01	4.22E-01	7.91E-01	9.18E-01	4.84E-02	3.23E-01	7.93E-01	8.17E-01
F8	Best	8.00E+02	8.00E+02	8.00E+02	8.00E+02	8.00E+02	8.11E+02	8.00E+02	8.00E+02	8.00E+02
	Mean	8.00E+02	8.01E+02	8.00E+02	8.00E+02	8.00E+02	8.13E+02	8.00E+02	8.00E+02	8.00E+02
	STD	8.71E-04	4.29E-01	4.92E-01	5.54E-04	1.93E-05	1.44E+00	2.47E-01	5.01E-01	4.27E-01
F9	Best	9.00E+02	9.00E+02	9.00E+02	9.00E+02	9.00E+02	9.00E+02	9.00E+02	9.00E+02	9.00E+02
	Mean	9.00E+02	9.00E+02	9.00E+02	9.00E+02	9.00E+02	9.00E+02	9.00E+02	9.00E+02	9.00E+02
	STD	7.27E-06	1.07E-04	3.03E-04	0.00E+00	3.19E-07	4.75E-05	0.00E+00	3.00E-01	2.87E-09
F10	Best	1.00E+03	1.00E+03	1.00E+03	1.00E+03	1.00E+03	1.66E+03	1.00E+03	1.00E+03	1.00E+03
	Mean	1.00E+03	1.01E+03	1.00E+03	1.00E+03	1.01E+03	2.03E+03	1.01E+03	1.00E+03	1.00E+03
	STD	1.35E-01	1.84E+01	6.84E+00	1.29E-01	1.74E+01	1.83E+02	6.95E+00	1.19E+00	1.56E-01

Table 15 Results on Hybrid Benchmark Functions

Function	CDO	PSO	WOA	CS	GWO	FruitFly	SSO	KrillHerd	WVO	ABC
F11	Best	1.02E+03	1.11E+03	1.10E+03	1.11E+03	4.85E+07	1.12E+03	1.11E+03	1.16E+03	1.11E+03
	Mean	1.12E+03	1.25E+03	1.11E+03	1.14E+03	5.78E+07	1.24E+03	1.16E+03	1.29E+03	1.11E+03
	STD	3.54E+01	9.10E+01	1.98E+00	2.06E+01	3.61E+06	8.52E+01	4.26E+01	8.61E+01	6.25E+00
F12	Best	1.96E+05	4.92E+04	1.92E+03	6.90E+04	2.36E+10	4.99E+05	1.80E+04	2.32E+06	1.71E+05
	Mean	3.15E+06	5.89E+06	3.49E+03	1.70E+06	2.42E+10	8.61E+06	2.51E+06	2.19E+08	3.33E+05
	STD	2.32E+06	7.55E+06	1.58E+03	2.53E+06	3.44E+08	8.06E+06	2.78E+06	3.27E+08	2.91E+05
F13	Best	1.12E+03	2.93E+03	1.32E+03	6.94E+03	2.55E+10	1.92E+03	3.03E+03	6.38E+03	2.01E+03
	Mean	6.96E+03	1.54E+04	1.35E+03	2.57E+04	2.69E+10	9.21E+03	1.21E+04	5.12E+06	7.22E+03
	STD	6.88E+03	1.31E+04	3.22E+01	1.77E+04	8.75E+08	7.03E+03	4.66E+03	1.40E+07	5.56E+03
F14	Best	1.41E+03	1.69E+03	1.42E+03	1.57E+03	9.41E+08	1.60E+03	1.55E+03	1.71E+03	1.43E+03
	Mean	1.56E+03	2.64E+03	1.44E+03	3.54E+03	1.99E+09	2.26E+03	4.83E+03	2.95E+03	2.04E+03
	STD	4.22E+01	1.49E+03	1.08E+01	1.95E+03	3.71E+08	8.45E+02	4.07E+03	1.33E+03	5.56E+02
F15	Best	1.72E+03	2.80E+03	2.20E+03	2.04E+03	5.02E+09	2.43E+03	8.04E+03	4.67E+03	1.54E+03
	Mean	1.94E+03	1.94E+04	1.51E+03	1.31E+04	6.44E+09	3.75E+04	2.54E+04	2.83E+04	2.60E+03
	STD	1.59E+02	1.91E+04	4.80E+00	1.05E+04	6.00E+08	4.68E+04	1.91E+04	1.97E+04	1.48E+03
F16	Best	2.20E+03	1.71E+03	1.60E+03	1.65E+03	3.28E+03	1.62E+03	1.85E+03	1.70E+03	2.20E+03
	Mean	1.63E+03	1.96E+03	1.62E+03	1.78E+03	3.37E+03	1.89E+03	2.01E+03	1.84E+03	1.67E+03
	STD	2.23E+01	1.38E+02	2.43E+01	1.23E+02	4.28E+01	2.02E+02	8.30E+01	1.27E+02	5.99E+01
F17	Best	1.72E+03	1.75E+03	1.73E+03	1.74E+03	3.09E+03	1.74E+03	1.75E+03	1.75E+03	2.20E+03
	Mean	1.75E+03	1.81E+03	1.75E+03	1.77E+03	3.20E+03	1.81E+03	1.82E+03	1.82E+03	1.71E+03
	STD	8.46E+00	6.15E+01	1.19E+01	3.74E+01	4.67E+01	5.05E+01	9.53E+01	4.80E+01	1.00E+01
F18	Best	2.80E+03	3.31E+03	1.82E+03	7.59E+03	1.22E+10	1.17E+04	2.59E+03	4.79E+03	2.62E+03
	Mean	9.40E+03	1.24E+04	1.83E+03	2.85E+04	1.39E+10	2.46E+04	9.11E+03	5.32E+04	7.21E+03
	STD	5.03E+03	6.57E+03	9.64E+00	1.59E+04	6.43E+08	1.29E+04	4.58E+03	5.92E+04	4.56E+03
F19	Best	1.90E+03	5.59E+03	1.90E+03	1.91E+03	1.11E+11	2.74E+03	2.44E+03	1.95E+03	1.94E+03
	Mean	2.24E+03	8.00E+03	1.91E+03	4.88E+04	1.17E+11	1.91E+04	2.31E+05	2.11E+04	2.58E+03
	STD	3.01E+02	6.95E+03	1.05E+00	1.05E+05	3.20E+09	3.27E+04	4.72E+05	2.39E+04	4.51E+02
F20	Best	2.00E+03	2.05E+03	2.10E+03	2.03E+03	3.02E+03	2.08E+03	2.03E+03	2.06E+03	2.00E+03
	Mean	2.04E+03	2.14E+03	2.19E+03	2.04E+03	3.10E+03	2.17E+03	2.19E+03	2.12E+03	2.00E+03
	STD	1.86E+01	6.73E+01	5.80E+01	8.39E+00	2.90E+01	6.87E+01	8.75E+01	4.66E+01	3.11E+00

Table 16 Results of Composite Benchmark Functions

Function	CDO	PSO	WOA	CS	GWO	FruitFly	SSO	KrillHerd	WVO	ABC
F21	Best	2.10E+03	2.20E+03	2.25E+03	2.20E+03	2.20E+03	2.20E+03	2.20E+03	2.21E+03	2.10E+03
	Mean	2.10E+03	2.32E+03	2.34E+03	2.21E+03	2.30E+03	2.28E+03	2.30E+03	2.28E+03	2.10E+03
	STD	1.35E-02	6.26E+01	3.92E+01	6.08E+00	3.70E+01	1.20E+01	5.92E+01	6.05E+01	3.60E-09
F22	Best	2.20E+03	2.31E+03	2.23E+03	2.23E+03	2.30E+03	2.25E+03	2.30E+03	2.30E+03	2.20E+03
	Mean	2.20E+03	2.46E+03	2.31E+03	2.28E+03	2.31E+03	2.32E+03	2.30E+03	2.49E+03	2.27E+03
	STD	1.58E-02	4.68E+02	3.23E+01	3.57E+01	1.12E+01	2.57E+01	3.85E+00	9.31E+01	5.16E+01
F23	Best	2.30E+03	2.66E+03	2.63E+03	2.61E+03	2.61E+03	2.62E+03	2.63E+03	2.63E+03	2.30E+03
	Mean	2.30E+03	2.71E+03	2.67E+03	2.62E+03	2.62E+03	2.66E+03	2.66E+03	2.66E+03	2.31E+03
	STD	4.25E-02	4.82E+01	2.57E+01	5.97E+00	1.20E+01	3.63E+01	1.88E+01	1.40E+01	4.09E+00
F24	Best	2.40E+03	2.41E+03	2.77E+03	2.52E+03	2.74E+03	2.40E+03	2.50E+03	2.63E+03	2.31E+03
	Mean	2.40E+03	2.69E+03	2.80E+03	2.65E+03	2.75E+03	2.69E+03	2.75E+03	2.76E+03	2.37E+03
	STD	1.37E-01	1.91E+02	3.53E+01	9.11E+01	9.74E+00	1.60E+02	9.23E+01	7.20E+01	4.32E+01
F25	Best	2.50E+03	2.90E+03	2.74E+03	2.90E+03	2.90E+03	2.60E+03	2.90E+03	2.97E+03	2.50E+03
	Mean	2.50E+03	2.93E+03	2.94E+03	2.90E+03	2.93E+03	2.90E+03	2.94E+03	3.04E+03	2.50E+03
	STD	2.79E-01	2.38E+01	7.23E+01	1.49E+01	2.17E+01	1.09E+02	1.93E+01	6.69E+01	4.93E+00
F26	Best	2.60E+03	2.84E+03	3.02E+03	2.61E+03	2.83E+03	2.80E+03	2.80E+03	3.11E+03	2.60E+03
	Mean	2.60E+03	3.43E+03	3.37E+03	2.86E+03	3.28E+03	3.07E+03	3.54E+03	3.30E+03	2.39E+03
	STD	1.14E-01	5.52E+02	3.22E+02	9.48E+01	5.24E+02	2.24E+02	4.73E+02	1.22E+02	1.42E+02
F27	Best	2.70E+03	3.13E+03	3.10E+03	3.09E+03	3.09E+03	3.09E+03	3.18E+03	3.11E+03	2.70E+03
	Mean	2.70E+03	3.19E+03	3.15E+03	3.09E+03	3.11E+03	3.11E+03	3.22E+03	3.13E+03	2.70E+03
	STD	5.56E-01	4.46E+01	4.25E+01	1.50E+00	3.02E+01	2.56E+01	3.75E+01	1.74E+01	2.59E-02
F28	Best	2.80E+03	2.86E+03	3.28E+03	3.10E+03	3.23E+03	3.10E+03	3.40E+03	3.31E+03	2.80E+03
	Mean	2.81E+03	3.23E+03	3.48E+03	3.12E+03	3.35E+03	3.30E+03	3.51E+03	3.36E+03	2.82E+03
	STD	3.84E+00	1.77E+02	1.24E+02	2.88E+01	8.50E+01	2.16E+02	8.08E+01	5.16E+01	1.76E+01
F29	Best	3.16E+03	3.18E+03	3.28E+03	3.16E+03	3.17E+03	3.19E+03	3.21E+03	3.23E+03	3.16E+03
	Mean	3.18E+03	3.28E+03	3.39E+03	3.20E+03	3.23E+03	3.31E+03	3.38E+03	3.30E+03	3.19E+03
	STD	7.32E+00	8.73E+01	1.19E+02	2.70E+01	5.88E+01	8.69E+01	1.47E+02	5.48E+01	2.04E+01
F30	Best	1.17E+05	6.25E+04	2.00E+05	4.34E+03	1.30E+04	2.01E+04	8.71E+05	1.14E+06	4.44E+03
	Mean	2.45E+05	6.92E+05	4.72E+06	6.63E+04	1.96E+06	3.20E+06	2.27E+07	9.09E+06	1.86E+05
	STD	7.63E+04	5.83E+05	4.42E+06	7.83E+04	3.02E+06	3.36E+06	2.59E+07	6.23E+06	2.75E+05

Table 17 Result of Student's T Test on classical benchmarks

Functions	PSO	CS	FruitFly	Krill Herd	WOA	WVO						
Ackley	1.17E-03	+	1.21E-01	-	1.43E-19	+	4.33E-04	+	5.91E-01	-	9.46E-03	+
Sphere	7.38E-02	-	8.22E-01	-	4.35E-16	+	3.00E-02	+	1.42E-01	-	1.01E-01	-
Schwefel	9.15E-03	+	6.69E-04	+	6.53E-04	+	1.02E-04	+	1.48E-02	+	1.60E-02	+
Schwefel 2.22	5.89E-04	+	9.22E-01	-	2.85E-21	+	4.14E-03	+	2.93E-02	+	3.00E-02	+
Xin She Yang	1.96E-03	+	1.80E-01	-	3.86E-18	+	2.93E-02	+	3.43E-01	-	1.01E-01	-
Rosenbrock	3.38E-01	-	1.50E-01	-	4.55E-13	+	7.79E-03	+	6.11E-01	-	7.35E-02	=
Griewank	1.40E-01	-	5.35E-04	+	4.67E-04	+	2.45E-02	+	2.40E-03	+	3.22E-03	+
Levy13	3.00E-02	+	3.43E-02	+	2.82E-04	+	3.50E-02	+	1.22E-01	-	1.72E-02	+
Rastrigin	1.24E-02	+	2.22E-02	+	1.45E-14	+	8.15E-02	-	3.28E-01	-	1.42E-01	-
Easom	1.33E-03	+	2.72E-01	-	4.30E-48	+	4.77E-48	+	3.43E-01	-	1.75E-01	-
Camel3	2.36E-02	+	3.25E-01	-	6.10E-15	+	1.62E-02	+	3.43E-01	-	1.58E-01	-
Bukin6	4.48E-04	+	6.74E-01	-	1.31E-03	+	2.10E-01	-	6.53E-03	+	1.11E-03	+
Shubert	7.79E-03	+	7.39E-01	-	5.82E-11	+	1.22E-02	+	1.98E-01	-	3.08E-02	+
Camel6	4.60E-02	+	2.07E-02	+	4.08E-03	+	2.22E-02	+	8.03E-02	=	8.32E-02	=
Schaffer2	3.78E-02	+	8.79E-02	=	6.43E-16	+	5.67E-03	+	3.43E-01	-	9.71E-02	-
Stybtang	4.37E-03	+	1.01E-02	+	4.46E-09	+	1.68E-01	-	6.23E-02	=	3.12E-01	-
CrossIt	4.07E-02	+	1.66E-03	+	2.78E-04	+	1.89E-03	+	1.76E-03	+	1.23E-01	-
Holder Table	1.65E-01	-	5.91E-04	+	7.01E-12	+	9.35E-02	-	5.89E-04	+	7.04E-02	=
Egg Holder	9.82E-05	+	2.77E-05	+	1.34E-29	+	1.94E-04	+	2.45E-02	+	1.31E-03	+
Michalewicz	1.60E-03	+	3.41E-04	+	1.25E-07	+	3.41E-04	+	3.17E-01	-	1.20E-01	-

Table 18 Student's TTest Results for CEC2017 Benchmarks

	PSO	WOA	CS	GWO	FruitFly	SSA	KrillHerd	WVO	ABC
F1	4.09E-02	+ 8.88E-03	+ 1.67E-03	+ 1.20E-02	+ 1.58E-74	+ 9.56E-03	+ 2.45E-01	- 3.16E-02	+ 3.61E-02
F2	4.87E-03	+ 4.20E-04	+ 5.07E-05	+ 1.07E-04	+ 7.50E-13	+ 5.08E-05	+ 5.03E-05	+ 1.17E-01	- 7.41E-01
F3	3.46E-03	+ 3.24E-03	+ 3.21E-03	+ 4.83E-03	+ 3.59E-13	+ 3.20E-03	+ 3.23E-03	+ 2.80E-01	- 3.20E-03
F4	4.63E-01	- 2.32E-01	- 2.24E-03	+ 3.46E-01	- 2.10E-15	+ 2.24E-03	+ 3.48E-02	+ 1.33E-02	+ 5.32E-02
F5	3.68E-02	+ 5.13E-03	+ 3.92E-03	+ 3.44E-01	- 8.80E-47	+ 3.92E-03	+ 3.93E-03	+ 2.76E-01	- 3.92E-03
F6	2.80E-03	+ 6.90E-02	- 1.85E-01	- 2.49E-01	- 2.50E-05	+ 1.85E-01	- 3.71E-01	- 1.86E-02	+ 1.85E-01
F7	2.78E-04	+ 5.49E-08	+ 1.68E-01	- 2.87E-04	+ 6.68E-14	+ 2.10E-02	+ 1.81E-08	+ 1.86E-05	+ 2.41E-02
F8	2.03E-04	+ 8.18E-02	- 3.89E-01	- 7.03E-03	+ 3.30E-10	+ 1.69E-01	- 1.74E-01	- 1.56E-02	+ 1.56E-01
F9	2.96E-02	+ 4.08E-02	+ 3.49E-02	+ 4.26E-02	+ 1.07E-61	+ 3.49E-02	+ 3.53E-02	+ 1.31E-01	- 3.50E-02
F10	9.32E-02	- 1.25E-01	- 3.53E-01	- 2.81E-02	+ 2.53E-08	+ 2.99E-01	- 1.59E-04	+ 1.57E-01	- 3.73E-01
F11	2.48E-01	- 4.50E-03	+ 1.78E-01	- 2.61E-01	- 2.28E-12	+ 3.72E-03	+ 5.01E-02	- 4.07E-04	+ 5.01E-01
F12	4.93E-01	- 3.50E-01	- 2.01E-03	+ 4.22E-02	+ 3.83E-18	+ 9.54E-02	- 5.91E-01	- 6.69E-02	- 5.33E-03
F13	1.62E-04	+ 5.84E-02	- 2.99E-02	+ 2.13E-02	+ 6.52E-15	- 3.39E-01	- 8.04E-02	- 2.78E-01	- 9.34E-01
F14	3.18E-02	+ 4.78E-02	+ 4.07E-06	+ 1.05E-02	+ 3.83E-08	+ 3.02E-02	+ 3.13E-02	+ 1.04E-02	+ 3.10E-02
F15	4.99E-04	+ 1.81E-02	+ 1.18E-05	+ 8.06E-03	+ 8.20E-11	+ 3.98E-02	+ 3.71E-03	+ 2.14E-03	+ 1.83E-01
F16	1.41E-05	+ 2.29E-05	+ 1.90E-01	- 3.42E-03	+ 5.25E-16	+ 3.71E-03	+ 1.29E-07	+ 1.23E-03	+ 1.39E-01
F17	1.10E-02	+ 2.30E-02	+ 6.09E-01	- 1.12E-01	- 7.28E-15	+ 3.61E-03	+ 4.02E-02	+ 1.99E-03	+ 1.73E-06
F18	2.39E-01	- 3.24E-01	- 1.03E-03	+ 1.08E-02	+ 1.54E-13	+ 4.51E-03	+ 9.00E-01	- 4.61E-02	+ 2.96E-01
F19	2.56E-02	+ 2.34E-01	- 6.34E-03	+ 1.93E-01	- 1.41E-15	+ 1.37E-01	- 1.60E-01	- 3.45E-02	+ 6.29E-02
F20	9.03E-04	+ 5.49E-05	+ 5.10E-01	- 7.16E-05	+ 2.15E-14	+ 4.66E-04	+ 1.23E-04	+ 1.18E-03	+ 1.47E-04
F21	1.55E-06	+ 1.02E-08	+ 1.10E-12	+ 2.91E-08	+ 1.75E-17	+ 5.06E-05	+ 1.91E-06	+ 5.22E-06	+ 1.43E-05
F22	1.17E-01	- 1.52E-06	+ 6.62E-05	+ 1.25E-10	+ 1.81E-18	+ 1.72E-07	+ 2.13E-14	+ 4.30E-06	+ 1.34E-03
F23	6.19E-10	+ 6.56E-12	+ 4.43E-17	+ 2.17E-14	+ 2.94E-18	+ 1.73E-10	+ 4.23E-13	+ 3.02E-14	+ 9.66E-04
F24	8.95E-04	+ 5.27E-11	+ 1.24E-05	+ 1.67E-15	+ 3.78E-25	+ 2.56E-04	+ 8.12E-07	+ 7.48E-08	+ 7.86E-02
F25	7.90E-13	+ 1.18E-08	+ 2.09E-14	+ 3.44E-13	+ 1.89E-22	+ 1.01E-06	+ 1.03E-13	+ 1.07E-09	+ 3.97E-01
F26	1.06E-03	+ 3.60E-05	+ 1.13E-05	+ 2.77E-03	+ 8.66E-24	+ 9.03E-05	+ 1.42E-04	+ 2.30E-08	+ 1.31E-03
F27	7.35E-11	+ 9.22E-11	+ 4.72E-23	+ 1.05E-11	+ 3.37E-15	+ 2.20E-12	+ 8.65E-12	+ 4.97E-14	+ 5.33E-05
F28	3.57E-05	+ 3.49E-08	+ 4.49E-11	+ 7.13E-09	+ 8.96E-19	+ 5.83E-05	+ 5.43E-10	+ 6.11E-11	+ 9.39E-02
F29	6.21E-03	+ 3.12E-04	+ 5.79E-02	- 2.26E-02	+ 2.72E-11	+ 8.07E-04	+ 1.72E-03	+ 3.80E-05	+ 1.41E-01
F30	3.69E-02	+ 1.06E-02	+ 5.06E-04	+ 1.08E-01	- 1.24E-08	+ 2.08E-02	+ 2.28E-02	+ 1.57E-03	+ 5.86E-01

Fig. 8 Convergence with Ackley Function

Fig. 9 Convergence with Xin She Yang N2 Function

Table 19 Results for Pressure Vessel Design Problem

	CDO	PSO	CS	WOA	WWO	WCMA-ES	iL-SHADE
Best	6.78E+03	6.88E+03	6.78E+03	6.78E+03	8.71E+03	6.78E+03	6.78E+03
Mean	6.81E+03	6.89E+03	6.78E+03	6.84E+03	1.07E+04	6.90E+03	6.91E+03
STD	3.62E+01	2.83E+02	1.86E-07	1.56E+02	1.50E+03	1.73E+02	1.25E+02

exploration ability of the algorithm, Multimodal functions are useful due to their high complexity and multiple local optima.

Tables 14, 15 and 16 represent the results of various Unimodal, Multimodal and Complex Functions. When compared to other algorithms, the results show the superiority of the algorithm in terms of either their best optimum value or the standard deviation in the results. Due to the Genetic Diversity Factor that is added due to its inspiration from the Cell division process, CDO shows competitive exploration abilities while using

Levy Flight and Mutation to exploit a particular region.

8.2 Local Optima Avoidance

Local Optima Avoidance can be directly linked to the exploration ability of an algorithm. Thus in this case, Multimodal, Hybrid and Composite Functions provide a way to test the proposed algorithm. Moreover, the balance between exploitation and exploration can concurrently be benchmarked by these functions. The Convergence Graphs in Section 6.4 shows how the proposed algorithm

Fig. 10 Convergence with Schaffer2 Function

Fig. 11 Convergence with Schwefel Function

Fig. 12 Convergence with Camel6 Function

Fig. 13 Convergence with Schwefel Function

converges to the optima while balancing exploration and exploitation and not getting stuck in local optima.

9 Conclusion and Future Work

This work proposed a novel Meta-heuristic called the Cell Division Optimizer. The proposed algorithm takes inspiration from the reproduction methods of cells in a colony. The 50 well-known benchmark functions were used to benchmark the performance of the algorithm. The results show that CDO was able to provide highly competitive results when compared to other meta-heuristics such as PSO, CS, Krill Herd, FruitFly algorithm, Whale Optimizer, Artificial Bee Colony, Grey

Wolf Optimier and Water-wave Optimizer. The benchmarks considered included Unimodal, Multimodal and Composite functions, and the results show that the superiority of the proposed algorithm. Convergence analysis was performed on these algorithms, which asserted the convergence of this algorithm.

The algorithm was tested on two real life-engineering problems which showed its versatility in tackling different problems. Finally, the statistical analysis was performed using Student's T-Test which proved the significance of the algorithm.

For future work, we are going to test the algorithm to solve the different real-life problems to assess its capabilities.

Table 20 Results for FM Sound Wave Problem

	CDO	PSO	CS	FruitFly	Krill Herd	WOA	WCMA-ES	iL-SHADE
Best	1.13E+01	2.30E+01	1.36E+01	2.49E+01	2.39E+01	1.37E+01	1.17E+01	1.13E+01
Mean	1.43E+01	2.43E+01	1.74E+01	2.73E+01	2.56E+01	2.26E+01	2.03E+01	1.69E+01
STD	3.06E+00	7.54E-01	2.07E+00	1.51E+00	1.33E+00	3.42E+00	3.64E+00	6.17E+00

Statements and Declarations

The authors have no relevant financial or non-financial interests to disclose.

Data Availability

All data generated or analysed during this study are included in this published article.

References

- [1] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95-international Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995). IEEE
- [2] Holland, J.H.: Genetic algorithms. Scientific american **267**(1), 66–73 (1992)
- [3] Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. Journal of global optimization **39**(3), 459–471 (2007)
- [4] Yang, X.-S., Deb, S.: Cuckoo search via lévy flights. In: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), pp. 210–214 (2009). Ieee
- [5] Mirjalili, S., Lewis, A.: The whale optimization algorithm. Advances in engineering software **95**, 51–67 (2016)
- [6] Wang, K.-P., Huang, L., Zhou, C.-G., Pang, W.: Particle swarm optimization for traveling salesman problem. In: Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693), vol. 3, pp. 1583–15853 (2003). <https://doi.org/10.1109/ICMLC.2003.1259748>
- [7] Gupta, D.: Solving tsp using various meta-heuristic algorithms. International Journal of Recent Contributions from Engineering, Science & IT (iJES) **1**(2), 22–26 (2013)
- [8] Rokbani, N., Abraham, A., Alimi, A.M.: Fuzzy ant supervised by pso and simplified ant supervised pso applied to tsp. In: 13th International Conference on Hybrid Intelligent Systems (HIS 2013), pp. 251–255 (2013). IEEE
- [9] Sha, D.Y., Lin, H.-H.: A multi-objective pso for job-shop scheduling problems. Expert Systems with Applications **37**(2), 1065–1070 (2010). <https://doi.org/10.1016/j.eswa.2009.06.041>
- [10] Sajedi, H., Rabiee, M.: A metaheuristic algorithm for job scheduling in grid computing. International Journal of Modern Education & Computer Science **6**(5) (2014)
- [11] Hassan, S., Kumar, K., Raj, C.D., Sridhar, K.: Design and optimisation of pressure vessel using metaheuristic approach. In: Applied Mechanics and Materials, vol. 465, pp. 401–406 (2014). Trans Tech Publ
- [12] Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Advances in engineering software **69**, 46–61 (2014)
- [13] Dhiman, G., Kaur, A.: Optimizing the design of airfoil and optical buffer problems using spotted hyena optimizer. Designs **2**(3), 28 (2018)
- [14] N, C.-e., Aungkulanon, P., Luangpaiboon, P.: Bees and firefly algorithms for noisy non-linear optimisation problems. Lecture Notes in Engineering and Computer Science **2** (2011)

- [15] Henrion, D., Lasserre, J.-B.: Solving non-convex optimization problems. *IEEE Control Systems Magazine* **24**(3), 72–83 (2004)
- [16] Wang, H., Moon, I., Yang, S., Wang, D.: A memetic particle swarm optimization algorithm for multimodal optimization problems. *Information Sciences* **197**, 38–52 (2012)
- [17] Colnari, A., Dorigo, M., Maffioli, F., Maniezzo, V., Righini, G., Trubian, M.: Heuristics from nature for hard combinatorial optimization problems. *International Transactions in Operational Research* **3**(1), 1–21 (1996)
- [18] Bansal, J.C., Sharma, H., Jadon, S.S., Clerc, M.: Spider monkey optimization algorithm for numerical optimization. *Memetic computing* **6**(1), 31–47 (2014)
- [19] De Castro, L.N., Von Zuben, F.J.: Artificial immune systems: Part i–basic theory and applications. *Universidade Estadual de Campinas, Dezembro de, Tech. Rep* **210**(1) (1999)
- [20] Jeanne, R.L.: The evolution of the organization of work in social insects. *Monitore Zoologico Italiano-Italian Journal of Zoology* **20**(2), 119–133 (1986)
- [21] Oster, G., Wilson, E.: Caste and ecology in the social insects: Princeton univ pr (1979)
- [22] Bonabeau, E., Marco, D.d.R.D.F., Dorigo, M., Théraulaz, G., Theraulaz, G., *et al.*: *Swarm Intelligence: from Natural to Artificial Systems* vol. 1. Oxford university press, ??? (1999)
- [23] Shin, K.-S., Lee, Y.-J.: A genetic algorithm application in bankruptcy prediction modeling. *Expert systems with applications* **23**(3), 321–328 (2002)
- [24] Srivastava, P.R., Kim, T.-h.: Application of genetic algorithm in software testing. *International Journal of software Engineering and its Applications* **3**(4), 87–96 (2009)
- [25] Wang, Q.: The genetic algorithm and its application to calibrating conceptual rainfall-runoff models. *Water resources research* **27**(9), 2467–2471 (1991)
- [26] Goldberg, D.E.: *Genetic algorithms in search. Optimization, and MachineLearning* (1989)
- [27] Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization* **11**(4), 341–359 (1997)
- [28] Fogel, L.J., Owens, A.J., Walsh, M.J.: *Artificial intelligence through simulated evolution* (1966)
- [29] Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. *IEEE Transactions on Evolutionary computation* **3**(2), 82–102 (1999)
- [30] Hansen, N., Müller, S.D., Koumoutsakos, P.: Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation* **11**(1), 1–18 (2003)
- [31] Rechenberg, I.: *Evolution strategy: Nature’s way of optimization*, pp. 106–126. Springer, ??? (1989)
- [32] Koza, J.R., Koza, J.R.: *Genetic Programming: on the Programming of Computers by Means of Natural Selection* vol. 1. MIT press, ??? (1992)
- [33] Webster, B., Bernhard, P.J.: A local search optimization algorithm based on natural principles of gravitation. Technical report (2003)
- [34] Erol, O.K., Eksin, I.: A new optimization method: big bang–big crunch. *Advances in Engineering Software* **37**(2), 106–111 (2006)
- [35] Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: Gsa: a gravitational search algorithm. *Information sciences* **179**(13), 2232–2248 (2009)

- [36] Kaveh, A., Talatahari, S.: A novel heuristic optimization method: charged system search. *Acta Mechanica* **213**(3), 267–289 (2010)
- [37] Formato, R.A.: Central force optimization. *Prog Electromagn Res* **77**, 425–491 (2007)
- [38] Van Tran, T., Wang, Y.: Artificial chemical reaction optimization algorithm and neural network based adaptive control for robot manipulator. *Journal of Control Engineering and Applied Informatics* **19**(2), 61–70 (2017)
- [39] Hatamlou, A.: Black hole: A new heuristic optimization approach for data clustering. *Information sciences* **222**, 175–184 (2013)
- [40] Kaveh, A., Khayatazad, M.: A new metaheuristic method: ray optimization. *Computers & structures* **112**, 283–294 (2012)
- [41] Du, H., Wu, X., Zhuang, J.: Small-world optimization algorithm for function optimization. In: *International Conference on Natural Computation*, pp. 264–273 (2006). Springer
- [42] Shah-Hosseini, H.: Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation. *International Journal of Computational Science and Engineering* **6**(1-2), 132–140 (2011)
- [43] Moghaddam, F.F., Moghaddam, R.F., Cheriet, M.: Curved space optimization: a random search based on general relativity theory. *arXiv preprint arXiv:1208.2214* (2012)
- [44] Eberhart, R.C., Shi, Y.: Evolving artificial neural networks. In: *Proceedings of the International Conference on Neural Networks and Brain*, vol. 1, pp. 5–3 (1998). PRC
- [45] Tandon, V.: Closing the gap between cad/cam and optimized cnc end milling. Master thesis, Purdue School of Engineering and Technology, Indiana University Purdue University Indianapolis (2000)
- [46] Poli, R.: An analysis of publications on particle swarm optimization applications. London, UK: Essex, department of computer science, university of Essex (2007)
- [47] Shi, Y., *et al.*: Particle swarm optimization: developments, applications and resources. In: *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546)*, vol. 1, pp. 81–86 (2001). IEEE
- [48] Prabukumar, M., Agilandeewari, L., Ganesan, K.: An intelligent lung cancer diagnosis system using cuckoo search optimization and support vector machine classifier. *Journal of Ambient Intelligence and Humanized Computing* **10**(1), 267–293 (2019)
- [49] Chatterjee, S., Dzitac, S., Sen, S., Rohatnovici, N.C., Dey, N., Ashour, A.S., Balas, V.E.: Hybrid modified cuckoo search-neural network in chronic kidney disease classification. In: *2017 14th International Conference on Engineering of Modern Electric Systems (EMES)*, pp. 164–167 (2017). IEEE
- [50] Abbas, A.K., Sadeq, A.T.: Database clustering using intelligent techniques. *Al-Nahrain Journal of Science* **17**(3), 195–203 (2014)
- [51] Amsaleka, R., Latha, M.: A optimally enhanced fuzzy kc means (oefkcm) for clustering algorithm medical image segmentation. *International Journal of Advanced Research in Computer and Communication Engineering* **3**(3), 5678–5682 (2014)
- [52] Tiwari, V.: Face recognition based on cuckoo search algorithm. *image* **7**(8), 9 (2012)
- [53] Kaveh, A., Bakhshpoori, T., Ashoory, M.: An efficient optimization procedure based on cuckoo search algorithm for practical design of steel structures. **2**(1), 1–14 (2012)
- [54] Shehab, M., Khader, A.T., Al-Betar, M.A.: A survey on applications and variants of the

- cuckoo search algorithm. *Applied Soft Computing* **61**, 1041–1059 (2017). <https://doi.org/10.1016/j.asoc.2017.02.034>
- [55] Emary, E., Zawbaa, H.M., Grosan, C., Hassenian, A.E.: Feature subset selection approach by gray-wolf optimization. In: *Afro-European Conference for Industrial Advancement*, pp. 1–13 (2015). Springer
 - [56] Emary, E., Zawbaa, H.M., Hassanien, A.E.: Binary grey wolf optimization approaches for feature selection. *Neurocomputing* **172**, 371–381 (2016)
 - [57] Mirjalili, S.: How effective is the grey wolf optimizer in training multi-layer perceptrons. *Applied Intelligence* **43**(1), 150–161 (2015)
 - [58] Mosavi, M.R., Khishe, M., Ghamgosar, A.: Classification of sonar data set using neural network trained by gray wolf optimization. *Neural Network World* **26**(4), 393 (2016)
 - [59] Li, S.-X., Wang, J.-S.: Dynamic modeling of steam condenser and design of pi controller based on grey wolf optimizer. *Mathematical Problems in Engineering* **2015** (2015)
 - [60] Yadav, S., Verma, S., Nagar, S.: Optimized pid controller for magnetic levitation system. *Ifac-PapersOnLine* **49**(1), 778–782 (2016)
 - [61] Sharma, Y., Saikia, L.C.: Automatic generation control of a multi-area st-thermal power system using grey wolf optimizer algorithm based classical controllers. *International Journal of Electrical Power & Energy Systems* **73**, 853–862 (2015)
 - [62] Faris, H., Aljarah, I., Al-Betar, M.A., Mirjalili, S.: Grey wolf optimizer: a review of recent variants and applications. *Neural computing and applications* **30**(2), 413–435 (2018)
 - [63] Negi, G., Kumar, A., Pant, S., Ram, M.: Gwo: a review and applications. *International Journal of System Assurance Engineering and Management* **12**, 1–8 (2021)
 - [64] Al-Aboody, N., Al-Raweshidy, H.: Grey wolf optimization-based energy-efficient routing protocol for heterogeneous wireless sensor networks. In: *2016 4th International Symposium on Computational and Business Intelligence (ISCBI)*, pp. 101–107 (2016). IEEE
 - [65] Zeng, C., Huang, S., Yang, Y., Zhou, G.: Influence of end structure on electromagnetic forces on end winding of a 1550 mw nuclear generator. *International Journal of Rotating Machinery* **2017** (2017)
 - [66] Rao, R., Patel, V.: Optimization of mechanical draft counter flow wet-cooling tower using artificial bee colony algorithm. *Energy Conversion and Management* **52**(7), 2611–2622 (2011)
 - [67] Gómez-Iglesias, A., Vega-Rodríguez, M.A., Castejón, F., Cárdenas-Montes, M., Morales-Ramos, E.: Artificial bee colony inspired algorithm applied to fusion research in a grid computing environment. In: *2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing*, pp. 508–512 (2010). IEEE
 - [68] Karaboga, D., Ozturk, C.: A novel clustering approach: Artificial bee colony (abc) algorithm. *Applied soft computing* **11**(1), 652–657 (2011)
 - [69] Udgata, S.K., Sabat, S.L., Mini, S.: Sensor deployment in irregular terrain using artificial bee colony algorithm. In: *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pp. 1309–1314 (2009). IEEE
 - [70] Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N.: A comprehensive survey: artificial bee colony (abc) algorithm and applications. *Artificial Intelligence Review* **42**(1), 21–57 (2014)
 - [71] Khader, A.T., Al-betar, M.A., Mohammed, A.A.: Artificial bee colony algorithm, its variants and applications: a survey (2013)
 - [72] Ma, M., Liang, J., Guo, M., Fan, Y., Yin,

- Y.: Sar image segmentation based on artificial bee colony algorithm. *Applied Soft Computing* **11**(8), 5205–5214 (2011)
- [73] Kumar, A.S., Khan, M.A., Thiraviam, R., Sornakumar, T.: Machining parameters optimization for alumina based ceramic cutting tools using genetic algorithm. *Machining science and Technology* **10**(4), 471–489 (2006)
- [74] Wisniewski, J.: Optimal design of reinforcing fibres in multilayer composites using genetic algorithms. *FIBRES AND TEXTILES IN EASTERN EUROPE* **12**(3), 58–63 (2004)
- [75] Pareek, N.K., Patidar, V.: Medical image protection using genetic algorithm operations. *Soft Computing* **20**(2), 763–772 (2016)
- [76] Ghaheri, A., Shoar, S., Naderan, M., Hoseini, S.S.: The applications of genetic algorithms in medicine. *Oman medical journal* **30**(6), 406 (2015)
- [77] Haldurai, L., Madhubala, T., Rajalakshmi, R.: A study on genetic algorithm and its applications. *International Journal of computer sciences and Engineering* **4**(10), 139 (2016)
- [78] Bhoskar, M.T., Kulkarni, M.O.K., Kulkarni, M.N.K., Patekar, M.S.L., Kakandikar, G., Nandedkar, V.: Genetic algorithm and its applications to mechanical engineering: A review. *Materials Today: Proceedings* **2**(4-5), 2624–2630 (2015)
- [79] Kumar, M., Husain, M., Upreti, N., Gupta, D.: Genetic algorithm: Review and application. Available at SSRN 3529843 (2010)
- [80] Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. *IEEE computational intelligence magazine* **1**(4), 28–39 (2006)
- [81] Abbass, H.A.: Mbo: Marriage in honey bees optimization-a haplometrosis polygynous swarming approach. In: *Proceedings of the 2001 Congress on Evolutionary Computation* (IEEE Cat. No. 01TH8546), vol. 1, pp. 207–214 (2001). IEEE
- [82] Li, X.: A new intelligent optimization-artificial fish swarm algorithm. Doctor thesis, Zhejiang University of Zhejiang, China, 27 (2003)
- [83] Roth, M.: Termite: A swarm intelligent routing algorithm for mobile wireless ad-hoc networks (2005)
- [84] Pinto, P.C., Runkler, T.A., Sousa, J.M.: Wasp swarm algorithm for dynamic max-sat problems. In: *International Conference on Adaptive and Natural Computing Algorithms*, pp. 350–357 (2007). Springer
- [85] Mucherino, A., Seref, O.: Monkey search: a novel metaheuristic search for global optimization. In: *AIP Conference Proceedings*, vol. 953, pp. 162–173 (2007). American Institute of Physics
- [86] Lu, X., Zhou, Y.: A novel global convergence algorithm: bee collecting pollen algorithm. In: *International Conference on Intelligent Computing*, pp. 518–525 (2008). Springer
- [87] Shiqin, Y., Jianjun, J., Guangxing, Y.: A dolphin partner optimization. In: *2009 WRI Global Congress on Intelligent Systems*, vol. 1, pp. 124–128 (2009). IEEE
- [88] Yang, X.-S.: Firefly algorithm, stochastic test functions and design optimisation. *International journal of bio-inspired computation* **2**(2), 78–84 (2010)
- [89] Askarzadeh, A.: Bird mating optimizer: an optimization algorithm inspired by bird mating strategies. *Communications in Non-linear Science and Numerical Simulation* **19**(4), 1213–1228 (2014)
- [90] Pan, W.-T.: A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowledge-Based Systems* **26**, 69–74 (2012)
- [91] Gandomi, A.H., Alavi, A.H.: Krill herd:

- a new bio-inspired optimization algorithm. *Communications in nonlinear science and numerical simulation* **17**(12), 4831–4845 (2012)
- [92] Martin, E., Hine, R.: *A Dictionary of Biology*. Oxford University Press, ??? (2008). <https://doi.org/10.1093/acref/9780199204625.001.0001>
- [93] Mazia, D.: Mitosis and the physiology of cell division. In: *The Cell*, pp. 77–412. Elsevier, ??? (1961)
- [94] Mitchison, T.J., Salmon, E.D.: Mitosis: a history of division. *Nature Cell Biology* **3**(1), 17–21 (2001). <https://doi.org/10.1038/35050656>
- [95] Nicklas, R.B.: Mitosis. *Advances in cell biology*, 225–297 (1971)
- [96] John, B.: *Meiosis. Developmental and Cell Biology Series*. Cambridge University Press, ??? (2005). https://books.google.co.in/books?id=dM_Kf7maJkQC
- [97] Kleckner, N.: Meiosis: how could it work? *Proceedings of the National Academy of Sciences* **93**(16), 8167–8174 (1996). <https://doi.org/10.1073/pnas.93.16.8167>
- [98] Chechkin, A., Metzler, R., Klafter, J., Gonchar, V.: *Introduction to the Theory of Lévy Flights*, pp. 129–162. Wiley, ??? (2008). <https://doi.org/10.1002/9783527622979.ch5>
- [99] PNSNH, A., Ali, M., Liang, J., Qu, B.: Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective real-parameter numerical optimization. *Int. J. Intell. Comput. Cybern* (2016)
- [100] Yang, X.-S., Deb, S., Mishra, S.K.: Multi-species cuckoo search algorithm for global optimization. *Cognitive Computation* **10** (2018). <https://doi.org/10.1007/s12559-018-9579-4>
- [101] Zheng, Y.-J.: Water wave optimization: A new nature-inspired metaheuristic. *Computers & Operations Research* **55**, 1–11 (2015). <https://doi.org/10.1016/j.cor.2014.10.008>
- [102] Brest, J., Maučec, M.S., Bošković, B.: il-shade: Improved l-shade algorithm for single objective real-parameter optimization. In: *2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1188–1195 (2016). <https://doi.org/10.1109/CEC.2016.7743922>
- [103] Nomura, M., Watanabe, S., Akimoto, Y., Ozaki, Y., Onishi, M.: Warm starting cma-es for hyperparameter optimization. *arXiv preprint arXiv:2012.06932* (2020)
- [104] Vesterstrom, J., Thomsen, R.: A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In: *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753)*, vol. 2, pp. 1980–1987 (2004). IEEE