EAI Endorsed Transactions
on Digital Transformation of Industrial Processes Research Article EAI.EU

A Two-Phase Hybrid Metaheuristic Framework for Engineering Optimization

A. Baskar[1], Justine Yasappan[1], M. Anthony Xavior[2,*], Anna Burduk[3] and Suthep Butdee[4]

[1]Loyola-ICAM College of Engineering and Technology (LICET), Chennai, India
[2]Vellore Institute of Technology, Vellore, India
[3]Wroclaw University of Science and Technology, Poland
[4]Rajamangala University of Technology Krungthep (RMUTK), Thailand

Abstract

Population-based optimization metaheuristic algorithms generate a pool of candidate solutions in the “Initialization” phase and these approximate solutions are iteratively refined further in the subsequent “Improvement” phase(s) towards the optimal/near-optimal solution. Any population-based algorithm may have a single or multiple “Improvement” phase(s). This paper analyses the impact of having two improvement phases. Different updating expressions are considered in each phase of the algorithm. In one case, the “C-Sine” algorithm, two new untested expressions are used, and performance is analysed. In the other case, the better performing Grey Wolf Optimizer (GWO) is applied in the first phase, and a new updating trigonometric expression is used in the second phase (termed as GWO:SineL algorithm) and analysis is carried out. The second phase applies the trigonometric "Sine" function over the random numbers generated using the Levy Flight Strategy. Mathematical functions, the CEC2019 dataset and a few real-world engineering problems are used for the analyses. Finally, the application of the “C-Sine” algorithm for solving multi-objective problems and the “GWO:SineL” algorithm for supply chain problems are studied. Codes are generated in MATLAB and run on an i5 PC with 4 GB RAM.

Keywords: Population-Based Algorithm, Constrained and Unconstrained Optimization, Two-Phase Framework, C-Sine Algorithm, GWO:SineL Algorithm, Levy Flight Strategy.

Received on 10 May 2026, accepted on 27 May 2026, published on 01 June 2026

doi: 10.4108/dtip.12972

1. Introduction

Optimization is an important domain of operations research that deals with finding better alternatives under the prevailing conditions and constraints. Though a few traditional methods like Newton’s and Gradient Descent are capable of solving specific non-linear problems, the probability of getting stuck in local optima is very high [1] for these algorithms. Exact methods are available that could find optimal solutions for smaller problems in polynomial time. However, computation time grows exponentially with problem size, and they will not converge in a reasonable time on many occasions. As a result of the significant developments taking place in industrialization and computation, real-world problems also become complex and larger with a huge number of variables and constraints. The optimization problems can be classified in many ways; one way is depicted in Fig. 1 [2].

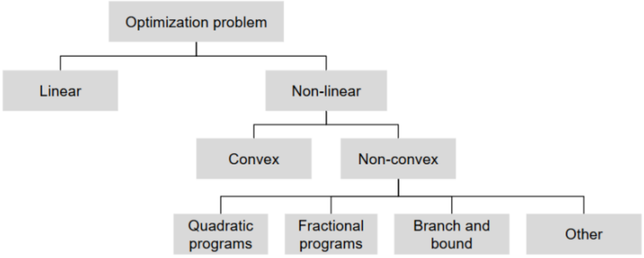


Figure 1. Classification of Optimization Problems

---

[*] Corresponding author. Email: manthonyxavior@vit.ac.in

Linear problems are convex since their objective functions and all constraints are linear. Non-linear problems can be further subdivided into convex or non-convex. Non-convex optimization problems are further grouped based on the programs and algorithms that can be applied to solve them.

Also, based on additional criteria, while classifying optimization problems, they can be grouped in different ways:

Continuous and Discrete Problems: based on the solution space, continuous or discrete.

Unconstrained and Constrained Problems: whether the solution space is unconstrained or constrained.

Single and Multi-Objective Problems: the objective is to optimize a single objective or multiple objectives.

Deterministic and Stochastic Problems: If the problems have deterministic variables, they fall under deterministic whereas, other problems have some uncertainty.

The general form of an optimization problem could be represented as:

Minimize $f(x)$ such that, $g(x) \leq 0$, $h(x) = 0$;
$x_{LB} \leq x \leq x_{UB} \ldots x = \{x_1, x_2, \ldots, x_n\} \epsilon R^n$ (1)

Where: $g(x)$ = non-equality constraints, $h(x)$ = equality constraints, x_{LB} = Lower Bound, x_{UB} = Upper Bound and, n = number of dimensions.

If the objective function has to be maximized, then

Minimize $-f(x)$ such that, $g(x) \leq 0$, $h(x) = 0$; $x_{LB} \leq x \leq x_{UB} \ldots$
$x = \{x_1, x_2, \ldots, x_n\} \epsilon R^n$ (2)

Several categories of algorithms exact, heuristic and metaheuristic are available in the literature for solving these problems. One possible classification of metaheuristic optimization algorithms (Table 1) in engineering and science [3] could be as follows:

- Bio-inspired algorithms: inspired by the biological processes and behaviours of living beings seen in nature.
- Mathematics-inspired algorithms: based on nonlinear functions, numerical methods, statistical behaviours, or distributed permutation flow.
- Physics-inspired Algorithms: inspired by physical observation and experimentation based on the behaviours observed in nature that are not related to biological processes.
- Socially inspired algorithms: search optimization methodologies that emulate human interactions in different environments.

The emergence of heuristic and metaheuristic algorithms has opened up new gates in optimization in recent times. Many of these algorithms have been proven effective at solving complex problems with acceptable accuracy in a reasonable time. They are capable of "exploring" new areas of the search space and "exploiting" the neighbourhoods of the already obtained approximate solution to further improve the solution towards the optimal solution.

Several algorithms propose a few "user-defined parameters" to modify the effectiveness, convergence speed, and ability of the algorithms. Tuning such parameters helps avoid local optima in multi-modal problem instances to a greater extent.

Other than the above-mentioned metaheuristics, many other algorithms find their place in the optimization domain. They can follow a specific strategy for convergence other than the one discussed above.

Table 1. Classification and Examples of Metaheuristic Optimization Algorithms

Category	Algorithms	Author(s)	Inspired By
Bio-inspired algorithms	Particle Swarm Optimization (PSO) [4]	Kennedy and Eberhar	Flocks of birds and fish
	Whale Optimization Algorithm (WOA) [5]	Mirjalili and Lewis	Flocks of whales
	Ant Colony Optimization (ACO) [6]	Dorigo et al.	Flocks of ants
Mathematics-inspired algorithms	Sine-Cosine Algorithm (SCA) [7]	Mirjalili	Trigonometric functions
	Sine (B) [8]	Baskar	Trigonometric functions
	Sine (AB), Cosine (AB) [9]	Baskar	Trigonometric functions
	TP-AB [10]	Baskar et al.	Trigonometric functions
Physics-inspired algorithms	Archimedes Optimization Algorithm (AOA) [11]	Hashim et al.	Archimedes' principle
	Black Hole Optimizer (BH) [12]	Hatamlou	The phenomenon of the black hole
	Simulated Annealing Algorithm (SA) [13]	Kirkpatrick et al.	Physical annealing
Socially inspired algorithms	Social Network Search Algorithm (SNS) [14]	Bayzidi et al.	Social network user's efforts
	Teaching-Learning-Based Optimization (TLBO) [15]	Rao et al.	Influence of a teacher on learners
	Nomadic People Optimizer (NPO) [16]	Salih and Alsewari	Behaviour of nomadic people

The Four-Point direction search algorithm (FP-AB) [17, 18] is a hybrid algorithm that uses an initial population followed by a local search to reach the optimal/near-optimal solution. The volleyball Premier League (VPL) Algorithm [19] is based on the modelling of behaviours of the players and decisions of the club managers in the football league and the Puzzle Optimization Algorithm (POA) [20] is inspired by puzzle-solving.

One of the popular algorithms of recent times is the Grey Wolf Optimizer (GWO) [21]. The advantage of GWO is the ease of implementation with very few parameters. However, it has the limitation of poor exploitation ability and being stuck in the local optimum while solving complex and challenging optimization problems [22].

Though population-based metaheuristics have many advantages, a number of parameter settings, getting stuck in the local optima, slow convergence rate, unbalanced exploration and exploitation capabilities, premature convergence, and poor performance in handling complex and multimodal problems are a few disadvantages of some algorithms. Developing hybrid and two-phase algorithms could address several disadvantages of such approaches. Several popular algorithms are combined to create new hybrid models, and many of these models perform better than their individual components.

Hybrid algorithms involving GWO are also widely applied in real-world problems.

The hybrid PSO-GWO [23] achieves superior convergence compared to standalone PSO (18.3% faster), GWO (12.7% faster), and five recent hybrid methods in multi-objective optimization with experimental validation in micro-hydro power plants.

A hybrid optimization scheme which is a combination of the Grey Wolf Optimization (GWO) and the Artificial Bee Colony (ABC) algorithm, termed GWBC was proposed [24] to optimize a Proportional-Integral-Derivative (PID) controller for a LLE platform.

BAGWO, a hybrid optimization algorithm that integrates the Beetle Antennae Search algorithm (BAS) and the Grey Wolf Optimizer (GWO) was proposed [25] to leverage their complementary strengths while enhancing their original strategies.

Sanjay et al. [26] proposed hybridisation of the GWO with operators from evolutionary algorithms for optimizing the configuration of distributed generator units. A novel hybrid GWO-SCA approach was proposed by Singh and Singh [27], and the results were compared with other popular optimization algorithms available in the literature.

GWO was integrated with cuckoo search (CS) [28] to overcome the limitation that GWO is prone to falling into a local optimum. A hybrid model of Ahmad et. al. [29] incorporates the onlooker and scout bee operators from the artificial bee colony algorithm (ABC) during the position-changing stage of the grey wolves.

Recently, Baskar [30] combined GWO and TP-AB algorithms to have a two-phase algorithm and claimed better performance over several algorithms. Wolpert and Macready [31] analysed a two-phase algorithm and showed that the performance of the hybrid variant performs better than the individual algorithms.

As there can be “no universal algorithm and strategy” for solving all kinds of optimization problems, as stated by the “No Free Lunch (NFL)” theorem, one algorithm that performs better in some problems may not repeat the same performance in some other problem datasets. This provides a way for a new strategy to come into play.

This paper analyses the impact of combining two phases in a single algorithm for solving various optimization problems. Each improvement phase has different updating expressions. Two such hybrid algorithms, “C-Sine” and “GWO:SineL” [32], are analysed using different datasets available in the literature. Each phase is considered separately and as a single algorithm, with the same number of function evaluations. The results are compared with those of a few other similar algorithms reported in the literature. Also, the possibility of using such algorithms for solving multi-objective and facility location problems is studied.

2. Strategy Behind the Work and Benchmark Functions Used

The objective is to analyse the impact of having more than one “Improvement” phase in a population-based optimization algorithm. The generalised strategy could be:

Select different datasets; mathematical functions with zero optimal functions, functions with negative optimal solutions, functions with positive optimal solutions, tough functions, constrained and unconstrained problem instances, standard datasets like that of CEC, real-world problems with and without constraints, single and multi-objective benchmarks and, possible operations research problems like Travelling Salesman (TSP), location science etc.

Have at least two improvement phases with diverse update expressions.

Keep the population size, number of function evaluations (NFEs), number of trials and computing conditions the same throughout the analyses.

Carry out the simulation independently, using the original algorithm with all improvement phases, and, separately for each improvement phase, keeping the NFEs the same.

Compute the statistical metrics like “minimum”, “mean”, “median” and “standard deviation” for each type of dataset.

Analyse and compare the statistical metrics.

The required benchmarks, search ranges and their optimal values used for the comparison are extracted from the available literature. The real-world problems (first 13 problems in Table 2) are taken from the Social Network Search (SNS) algorithm paper; the cantilever stepped beam (continuous) is available in the MATLAB Help Centre [33]. In the pressure vessel, RCC Beam and Helical Spring (discrete) problems, the first two are discrete variables. In the gear train problem, all four variables are discrete. The spring (discrete) problem is taken from Deb and Goyal [34]. The detail of the constrained special benchmark function of Himmelblau is available in the paper of Michalewicz and Schoenauer [35] and, it is to be noted that two constraints are active in the solution at the optimum. A penalty approach is implemented to handle the constraints with a penalty parameter of 109.

Another important dataset used in this work is the “100-digit challenge of CEC2019” (Table 3), the data, which are

Table 2. Constrained Engineering Problems

S.No.	Problem	No. of Variables	No. of Constraints	Optimal Value
1.	Cantilever Beam	5	1	1.3399576
2.	I-shaped Beam	4	2	0.0130741
3.	Three-Bar Truss	2	3	263.8958434
4.	Tubular Column	2	6	26.486361473
5.	Speed Reducer	7	11	2994.424466
6.	Piston Lever	4	4	8.41269832311
7.	Corrugated Bulkhead	4	6	6.8429580100808
8.	Pressure Vessel	4	4	6059.714335048436
9.	Tension/Compression Spring (Continuous)	3	4	0.01266051
10.	Welded Beam	4	7	1.724852308597366
11.	Gear Train	4	0	2.70085714e-12
12.	Reinforced Concrete Beam	3	2	359.2080
13.	Car Side Impact	11	10	22.84296954
14.	Cantilever Stepped Beam (Continuous)	10	11	63408.9
15.	Tension/Compression Spring (Discrete)	3	8	2.65855916
16.	Himmelblau's Function	5	5	−30665.5

Table 3. 100 Digit Challenge: Basic Test Functions (CEC2019)

S.No.	Function	Optimal Value	Dimension	Bounds
1	Storn's Chebyshev Polynomial Fitting Problem	1	9	[−8192, 8192]
2	Inverse Hilbert Matrix Problem	1	16	[−16384, 16384]
3	Lennard-Jones Minimum Energy Cluster	1	18	[−4, 4]
4	Rastrigin's Function	1	10	[−100, 100]
5	Griewangk's Function	1	10	[−100, 100]
6	Weierstrass Function	1	10	[−100, 100]
7	Modified Schwefel's Function	1	10	[−100, 100]
8	Expanded Schaffer's F6 Function	1	10	[−100, 100]
9	Happy Cat Function	1	10	[−100, 100]
10	Ackley Function	1	10	[−100, 100]

available in Price et al. [36]. Other unconstrained benchmarks used in this paper are extracted from the portals of Al-Roomi [37] and Simon Frazer University [38].

3. Two-Phase "C-Sine" Algorithm with Tuning Option

The structure of this algorithm is similar to other population-based metaheuristics, a typical example is the popular TLBO algorithm.

Apart from the initialization stage, the algorithm has two phases, resulting in two function evaluations (FEs) per iteration during execution. A tuning parameter is included as an optional parameter in the second phase whose value reduces from "a" to zero as the iteration progresses. Although the value of "a" can take any value, it is set to 1 in this paper. Phase I takes care of the "exploitation" part, whereas Phase II is responsible for the "exploration" requirement.

The features of the algorithm are given below:

Two-Phase Algorithm (two function evaluations per iteration)

$(\mu+\lambda)$ selection in phase-I

Greedy selection in phase-II

Updation in phase-I:

New X = C*Sin(2*pi*rand).*X … C = 1.5 in this work (3)

Updation in phase-II:

New X(i)=X(i)+R*Sin (2*pi*rand).*(X(j)-Mean Sol) (4)

R = a-(a*Current Iteration/Maximum Iterations);

a = 1; if i == N, j = 1 else j = i + 1, N = Population Size

There are three counters introduced in the algorithm;

Count 1: records the "Best Cost" updates in Phase-I

Count 2: tracks the "Population Cost" updates in phase II

Count 3: counts the "Best Cost" updates in Phase-II.

The pseudo-code for this Two-Phase Algorithm (with tuning) is presented in Fig. 2. "Sin(2*pi*rand)" (Baskar, 2024) expression is used in both phases which generate random values varying from [-1, 1]. Hence, the expression "C*Sin(2*pi*rand).*X" effectively searches the space in the neighbourhoods of the approximate solution "X". In the second phase, the updating expression is responsible for the "exploration" requirement due to the stochastic nature of the element "(X(j) – Mean Solution)".

This Two-Phase "C-Sine" algorithm is simulated as 5 different variants.

(1) Two-Phase algorithm without tuning [A1],

(2) Two-Phase algorithm with tuning [A2],

(3) Single-phase algorithm, only Phase-I [A3],

(4) Single-phase algorithm, only Phase-II without tuning [A4] and,

(5) Single-phase algorithm, only Phase-II with tuning [A5].

The updating expressions used in both phases are entirely different. Hence, the two-phase algorithm is split, and performance is analysed using different sets of benchmarks. To maintain the number of FEs the same, single-phase algorithms are run twice the number of iterations as that of a two-phase algorithm.

The complexity of the two new two-phase algorithms discussed is O(2*I*N*D).

N – Population Size, I - Number of iterations, D - Dimension (number of variables) and there are two function evaluations per iteration.

The parameter settings used is presented in Table 4.

```
Randomly initialize the solutions: X                                   % N Solutions
Compute the Costs
Select the "Best Cost" and "Best Solution"
C = 1.5; a = 1;                                                        % for this work
While (Current_Iteration < Maximum_Iteration) do                       % while loop starts
    R = a - (a*Current_Iteration/Maximum_Iteration)
    MeanSol = mean(X);
    % Phase-I
        for (i=1 to N) do                                              % for loop starts
            New Solution: NewX(i) = C*Sin(2*pi*rand).*X(i)             % N solutions
            Trim
        end for                                                        % for loop ends
Compute the Costs for NewX                                             % FE-I
(μ + λ) Selection, X                                                   % N Solutions
Select the "Best Cost" and "Best Solution"
    % Phase-II
        for (i=1 to N) do                                              % for loop starts
            if i==N, j=1
            else j=i+1
            end if
            Improved Solution: X1 = X(i) + R*Sin (2*pi*rand).*(X(j) - MeanSol)
            Trim
            Compute the cost of improved solution: Cost(X1)            % FE-II
Greedy Selection
        end for                                                        % for loop ends
        Select the "Best Cost" and "Best Solution"
Current_Iteration = Current_Iteration+1
end while                                                              % while loop ends
Return the "Best Cost"; "Best Solution"
```

Figure. 2. Pseudo Code of the Two-Phase "C-Sine" Algorithm [A2]

Table 4. Parameter Settings for C-Sine Algorithm

Algorithm (C-Sine)	Dataset	Population Size	NFEs	Trials	Remark
Two-Phase (No Tuning), Two-Phase (With Tuning), Phase-I, Phase-II (No Tuning), Phase-II (With Tuning)	Unconstrained benchmark	5	40000	30	Dimension: 30
		5	50000	30	Dimension: 50, 100
	CEC2019 Benchmark	5	15000	30	Dimension: 9 to 18
	Real-World Engineering Problems	5	3600 to 100000	30	Dimension: 2 to 11 Constraints: 1 to 11
	Un-constrained Functions with Negative Optimal	5	20000	30	Dimension: 2 to 30
	Tough Functions	5	20000	30	Dimension: 2 to 30
Two-Phase with tuning	Multi-Objective unconstrained problems	100	100	1	2 Objective - Dimension: 10 to 30 3 Objective - Dimension - 2

3.1. Analyses of Results: C-Sine Algorithm

The algorithms and benchmarks are coded in MATLAB and run on an i5 desktop PC with 4 GB RAM. Uniformly, a population size of 5 is used, and 30 trials are conducted to better analyse the statistical metrics. Table 5 presents the best results for the five variants of the "C-Sine" algorithm on the most important dataset of real-world constrained engineering problems. Two-Phase (With Tuning) [A2] performs well here with 9 best results, followed by Two-Phase (No Tuning) [A1] and Phase-II (With Tuning) [A5] with 7 results each. The best results among all 5 variants are compared with known results (Table 6) available in the Social Network Search (SNS) paper. Other algorithms used for the performance analyses include: AOS-Atomic Orbital Search [39], WCA-Water Cycle Algorithm [40], CGO-Chaos Game Optimization [41], MCEO- Multilevel Cross Entropy Optimizer [42]; CS-Cuckoo search [43]; WOA- Whale Optimization Algorithm and WSA- Water Strider Algorithm [44].

For the "Best" metric, both SNS and Phase-II (without tuning) [A4] perform equally well in 6 problems each, followed by Two-Phase algorithms (without and with tuning) [A1, A2] and Phase-II (with tuning) [A5] doing well in 5 cases each. AOS accounts for 2 better results and; WCA, MECO and CS for one result each.

Table 7 summarises all simulations involving the 5 variants. Phase-II (with tuning) [A5] is the best performer with 86 (out of 201) better results, followed by Two-Phase (with tuning) [A2] with 81 results. Phase-II (with tuning) [A5] performs very poorly on the unconstrained benchmark, with zero optimal values and only 1 better result out of 60. However, Phase II (with tuning) [A5] performs very well in all other datas

Table 5. Real-World Constrained Engineering Problems – Comparison of “Best” Results

Problem	Two-Phase (No Tuning) [A1]	Two-Phase (With Tuning) [A2]	Phase-I [A3]	Phase-II (No Tuning) [A4]	Phase-II (With Tuning) [A5]
Cantilever Beam	1.3399568	**1.3399565**	1.3739086	1.3399569	1.3399566
I-shaped beam	**0.0130741**	**0.0130741**	0.0131125	0.0130742	**0.0130741**
Three-bar truss	263.8958435	**263.8958434**	263.9656335	**263.8958434**	**263.8958434**
Tubular column	**26.486361472**	26.486361473	26.535499636	**26.486361472**	**26.486361472**
Speed reducer	2994.426164	2994.427985	3083.248451	**2994.424473**	2994.424609
Piston lever	**8.41269833579**	8.41271268842	9.31143088791	8.90848050492	8.84839851831
Corrugated bulkhead	6.8429587885612	6.8429620166011	7.63544884550006	**6.8429580145467**	6.8429580789110
Pressure vessel (with SNS bounds)	**6059.719047641514**	6059.978440276225	8484.025443545645	6061.935250705509	6059.815029146799
Tension/compression spring (continuous)	0.01266591	0.01266645	0.01287601	0.01268121	**0.01266541**
Welded beam	1.725218789352377	1.724959114593066	1.935785529791289	1.725636445825603	**1.724952116201391**
Gear train	**2.70085714e-12**	**2.7008571e-12**	6.6020899e-10	**2.7008571e-12**	**2.7008571e-12**
Reinforced concrete beam	**359.2080**	**359.2080**	362.4219	**359.2080**	**359.2080**
Car side impact	22.84658012	**22.84440873**	23.97411726	22.85301155	22.84498777
Cantilever stepped beam (Continuous)	63438.3	**63149.1**	76827.1	63239.6	63172.5
Tension/compression spring (Discrete)	2.65856009	**2.65855921**	3.75462205	2.65856755	2.65855933
Himmelblau’s Function	**-30665.53867**	**-30665.53867**	-30578.57081	-30665.53771	-30665.53777
Best in	**7**	**9**	**0**	**6**	**7**

Table 6. Results of Constrained Engineering Problems; “Best” Values

S.No.	Problem	Best	Algorithm (NFEs)	This Work
1.	Cantilever Beam	**1.339957**	AOS (100000)	**1.3399565 [A2]**
2.	I-shaped beam	**0.0130741**	SNS (3600)	**0.0130741 [A1, A2, A5]**
3.	Three-bar truss	**263.895843**	WCA (5250)	**263.8958434 [A2, A4, A5]**
4.	Tubular column	**26.48636147**	SNS (1250)	**26.486361472 [A1, A4, A5]**
5.	Speed reducer	**2994.443649**	CGO (100000)	**2994.424473 [A4]**
6.	Piston lever	**8.412698349**	SNS (5000)	**8.41269833579 [A1]**
7.	Corrugated bulkhead	**6.84295801**	AOS (100000)	**6.8429580145467 [A4]**
8.	Pressure vessel (with SNS bounds)	**6059.714335**	SNS (6000)	6059.719047641514 [A1]
9.	Tension/compression spring (continuous)	**0.01266051**	MCEO (2000)	0.01266541 [A5]
10.	Welded beam	**1.724852**	SNS (9000)	1.724952116201391 [A5]
11.	Gear train	**2.700857E − 12**	SNS (25000)	**2.70085714e-12 [A1, A2, A4, A5]**
12.	Reinforced concrete beam	**359.2080**	SNS (1000)	**359.2080 [A1, A2, A4, A5]**
13.	Car side impact	**22.84294**	CS (20000)	22.84440873 [A2]

Table 7. Summary of Simulations

Benchmark	Metric	Two-Phase (No Tuning) [A1]	Two-Phase (With Tuning) [A2]	Phase-I [A3]	Phase-II (No Tuning) [A4]	Phase-II (With Tuning) [A5]
Unconstrained-Zero Optimal	Best	14	14	**17**	1	0

	Mean	12	14	**17**	1	0
	SD	13	14	**18**	1	1
	Out of 60	**39**	**42**	**52**	**3**	**1**
Unconstrained-Negative Optimal	Best	9	**10**	4	5	**10**
	Mean	1	2	1	1	**9**
	SD	0	0	2	1	**8**
	Out of 33	**10**	**12**	**7**	**7**	**27**
Unconstrained-Tough Functions	Best	4	**7**	3	0	2
	Mean	0	3	3	0	**4**
	SD	2	0	1	3	**4**
	Out of 30	**6**	**10**	**7**	**3**	**10**
Constrained Real-World	Best	7	**9**	0	6	7
	Mean	0	0	1	2	**13**
	SD	0	0	1	3	**12**
	Out of 48	**7**	**9**	**2**	**11**	**32**
CEC 2019 Dataset	Best	3	**7**	1	2	4
	Mean	2	1	1	2	**7**
	SD	0	0	1	4	**5**
	Out of 30	**5**	**8**	**3**	**8**	**16**
Best in	**Out of 201**	**67**	**81**	**71**	**32**	**86**

Phase-I [A3] algorithm performs well for the unconstrained benchmark with zero optimal values with, 52/60 better results. In all other datasets, Phase-I [A3] performs below average.

It is concluded that Two-Phase (with tuning) [A2] and Phase-II (with tuning) [A5] perform well when different categories of problems are considered in the analysis. This demonstrates the varying performances of “Improvement” phases with diverse updating expressions for different datasets.

3.2. Preliminary Evaluation of Multi-Objective Benchmarks

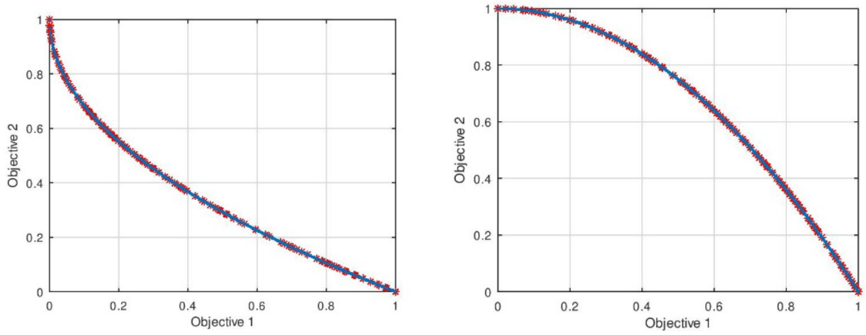
The suitability of the new algorithm for multi-objective optimization problems is primarily evaluated in this section. For this, five two-objective problem instances proposed by Zitzler et al. [45] and one three-objective problem of Viennet et al. [46] are used (Table 8). In this analysis, ZDT5 is not considered because it is a binary-encoded problem. For all six unconstrained problems, a single trial is conducted with a population size of 100 and runs for 100 iterations. One of the best-performing algorithms, the Two-Phase Algorithm with tuning [A2], is used in this simulation.

No in-depth analyses are conducted using any performance metrics such as Diversity, Spacing, Spread, or Hypervolume in this work. Only the true Pareto fronts (blue) are superimposed over the obtained Pareto fronts (red). The Pareto Fronts are reproduced in Fig. 3 for all six problems. The obtained Pareto Fronts closely match the true ones. Detailed analyses are reserved for future work.

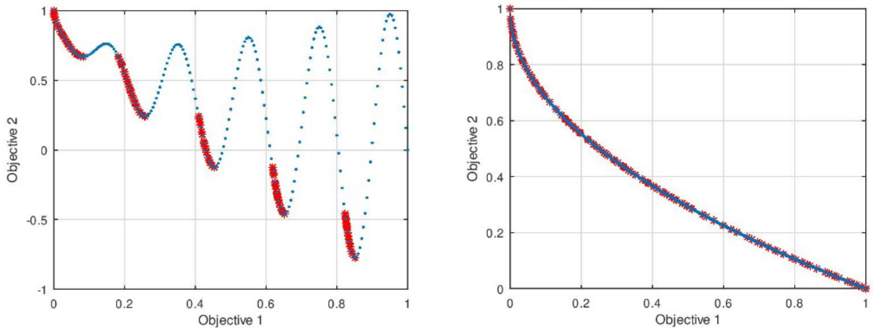
Table 8. Two Objective Unconstrained ZDT Problems

Problem	No. of Objectives	No. of Variables	Bounds (Domains)	Geometry	First Objective	Second Objective
ZDT1	2	30	[0, 1]	Convex	Uni-modal	Uni-modal
ZDT2	2	30	[0, 1]	Concave	Uni-modal	Uni-modal

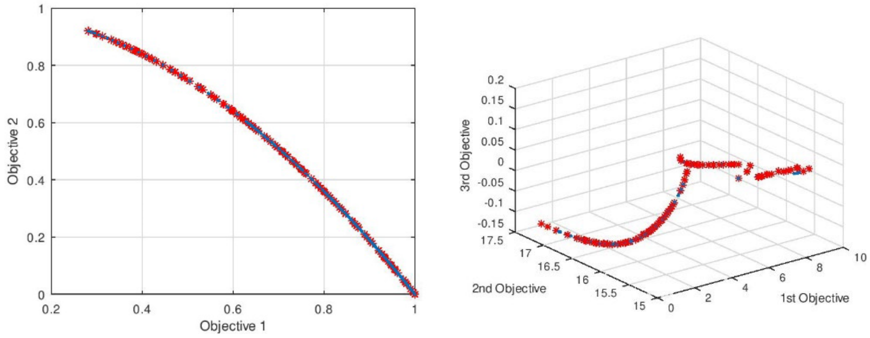
ZDT3	2	30	[0, 1]	Dis-connected	Uni-modal	Multi-modal
ZDT4	2	10	X_1 [0, 1]; X_i [-5, 5]	Convex, Multi-frontal	Uni-modal	Multi-modal
ZDT6	2	10	[0, 1]	Concave	Multi-modal	Multi-modal
Viennet	3	2	[-3 3]	Convex	--	--



ZDT1 (Two-Objective) ZDT2 (Two-Objective)



ZDT3 (Two-Objective) ZDT4 (Two-Objective)



ZDT6 (Two-Objective) Viennet (Three-Objective)

Figure 3. Superimposed True and Obtained Pareto Fronts

4. Two-Phase GWO:SineL Algorithm with Levy Flight Strategy

The GWO (with a linear reduction from 2 to 0) algorithm is combined with a new, untested updating expression using the Levy Flight Strategy (LFT), together termed “GWO:SineL,” and the two-phase algorithm is analysed on 50 benchmarks available in the literature. The updating expression in the second phase, “SineL” (Sine function combined with Levy distribution), also uses the "tuning" technique. The tuning parameter 1 (in this work) reduces from 1 to 0 as the iteration progresses. The updating expression used by "SineL" follows the simple concept,

New Solution = Old Solution ± Fraction of Old Solution [47].

$$\mathbf{New\ X(i) = X_i + R{*}Sine(LF){*}[X_{(i-1)} – X_{(i+1)}]} \quad (5)$$

Where X(i) is the current population, $X_{(i-1)}$ is the previous one and $X_{(i+1)}$ is the next population.

The flow chart of "GWO:SineL" is presented in Fig. 4.

In the flowchart,

R = 1 - (Current Iteration Number)/(Maximum Number of Iterations),

"LF" is the set of random numbers generated using LFT with a shape parameter of β = 1.5. LFT refers to the Levy distribution; it allows generating larger steps than the normal distribution. Using a mixture of larger and smaller steps avoids local minima, leading to better exploration of the search space.

The approximated Levy distribution [48] could be represented as:

$$\mathbf{L(s) \sim s^{(-1-\beta)}} \quad (6)$$

where 's' is the step size.

The trigonometric "Sine" function is used in the updating expression of "SineL" as "Sine(LF)".

The levy function, LF is represented as,

LF = Levy(Population Size, Dimension, Shape Parameter 1.5).

The Levy distribution generates a matrix (size: population size × dimension) of both positive and negative numbers that need not lie between [-1, +1]. However, the "Sine(LF)" function, when the inputs are in terms of radians, will produce values between [-1, +1}.

For example in one simulation, Levy(3, 4, 1.5) gives,

LF = [0.4012, -13.6224, 0.9730, 0.5806;
-0.6034, 0.6910, -1.0668, -0.1182;
2.2532, 0.1811, 15.3325, -0.3605] and the corresponding "Sine" values are,

Sine(LF) = [0.3905, -0.8704, 0.8266, 0.5486;
-0.5675, 0.6373, -0.8757, -0.1179;
0.7761, 0.1801, 0.3667, -0.3527].

This ensures good diversity within the search space.

The complexity of the discussed two-phase new algorithms is O(2*I*N*D).

N – Population Size, I - Number of iterations, D - Dimension (number of variables) and there are two function evaluations per iteration.

Time Complexity of NEH Algorithm is O(m*n3). n = number of jobs, m = number of machines.

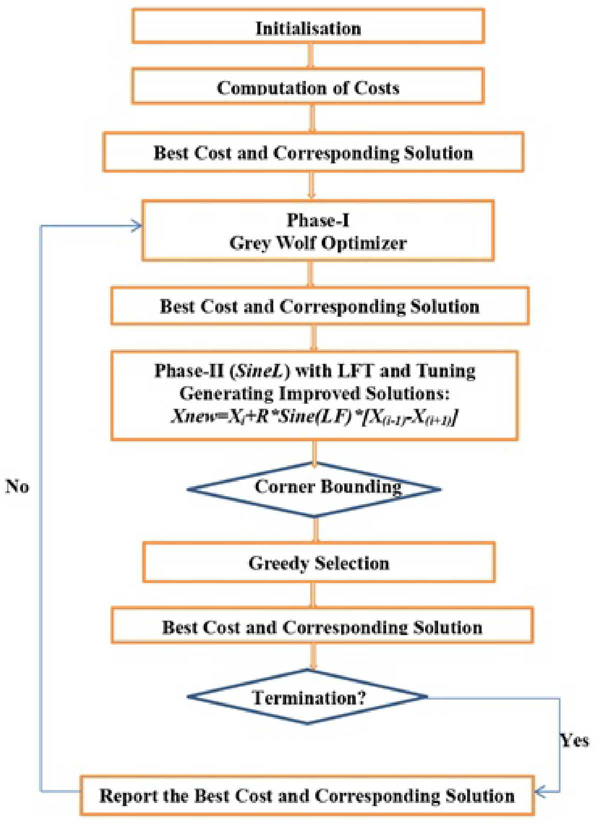


Figure 4. Flowchart of GWO:SineL Algorithm

4.1. Benchmarks Used and Parameter Settings

The first set of 19 problems is extracted from Abdullah and Ahmed [49]. It is a collection of unimodal (F1-F7), multimodal (F8-F13) and composite (F14-F19) functions. They are scalable functions, and in this work, the dimension is set to 30 for F1-F13 functions and to 10 for the composite functions. F8 has a negative optimal solution, whereas zero is the optimal value for the other 18 problems.

10 problems of CEC2019 (Table 3) and the third set contains constrained real-world engineering problems (Table 2). The fourth set comprises five problems with negative optimal costs.

The parameter settings of this hybrid GWO:SineL algorithm is presented in Table 9.

4.2. Performance of GWO:SineL Algorithm

GWO outperforms others in unimodal functions F1-F7, reporting best values in fifteen cases each, followed by GWO:SineL in 5 cases. SineL is a poor performer for this set. For the multimodal functions F8-F13 also GWO is good with eight metrics, closely followed by GWO:SineL in seven and SineL in three. For the composite functions

F14-F19, SineL is a clear winner, reporting the best results in 13 cases and GWO:SineL is a distant second. If only the "mean" values are considered, SineL is the winner in all six problems. However, for the tough CEC2029 dataset, the hybrid GWO:SineL is the best performer. It yields the best results in 19/30 metrics, followed by 11 by GWO. GWO:SineL is good in 8/10 mean results.

For the 16 real-world constrained engineering (Table 10) and problems with negative optimal also, GWO:SineL is a clear leader.

GWO:SineL is a better performer in 74/150, GWO in 42, closely followed by SineL in 39 cases. For the important "mean" metric also, GWO:SineL produces better results in 26/50 problems, SineL in 13 and GWO in 12 problems.

The results demonstrate the better performance of the hybrid GWO:SineL when compared with individual GWO and SineL for the same number of function evaluations.

The summary of results of 16 real-world engineering problems, scalable functions are presented in Table 11.

Table 9. Parameter Settings for GWO: SineL Algorithm

Algorithm	Dataset	Population Size	NFEs	Trials	Remark
GWO, SineL, GWO:SineL	Classic 19 Functions	5	10000	100	Dimension: 30
	CEC2019 Benchmark	30	15000	30	Dimension: 9 to 18
	Real-World Engineering Problems	5	5000	100	Dimension: 2 to 11 Constraints: 1 to 11
	Un-constrained Functions with Negative Optimal	5	5000	100	Dimension: 2 to 30
GWO:SineL	Random Dataset, Mini-Max and Mini-Sum	5	5000	1	10 Data Points
	Indian States and UTs, Mini-Max and Mini-Sum	5	10000	1	36 Data Points
	Indian States and UTs, TSP	5	200 and 10000	1	36 Data Points

Table 10. Real-World Engineering Problems, Best Values

Function	GWO	SineL	GWO:SineL
Cantilever Beam	1.340007124430873108877904087 4215	1.355824169938772527643777721 0958	**1.339989410637388855818130650 7322**
Car Side Impact	22.88676070297855957846877572 5923	22.87274637153647205423112609 4237	**22.85435534002374424744630232 4533**
Corrugated Bulk Head	6.849807480895721667479847383 2197	**6.843587785175042625951391528 3605**	6.843759068481729102018107369 0765
Gear Train	**2.70085714888651335106182952 7627e-12**	2.307815733312755151633545295 4697e-11	**2.70085714888651335106182952 7627e-12**
Himmelblau	-30664.4949416332165128551423 54965	**-30665.4443499466506182216107 84531**	-30665.3725135231798049062490 46326
I-Beam Deflection	0.013074120951945206528521659 095077	0.013074141724784027127070906 715289	**0.013074118905640096793829840 748913**
Piston Lever	8.423559594033925179701327579 096	8.829327836191318112923909211 5313	**8.412732100421097314324470062 2745**
Pressure Vessel	6061.529165486501369741745293 1404	6060.075636271614712313748896 122	**6059.778334508404441294260323 0476**
RCC Beam	359.2081033714569002768257632 8516	359.2080000001010375854093581 4381	**359.2080000000754580469219945 3712**
Speed Reducer	3003.382570269637653836980462 0743	**2994.820998069923462026054039 5975**	2995.553357496624357736436650 157

Helical Spring	0.0126791837823130183054 98708571122	0.0126699206132566705856 8900102955	**0.0126665283823893751874 44535697523**
Helical Spring (Discrete)	2.6585628435159875948556 873481721	2.6585592784267300103806 519473437	**2.6585591778530668349844 745534938**
Stepped Cantilever Beam (MATLAB)	63962.444981652944989036 76867485	**63835.197715529146080370 992422104**	63937.223246682631724979 728460312
Three-Bar Truss	263.89632630662140400090 720504522	**263.89585310215369418074 260465801**	263.89587386684462444463 861174881
Tubular Column	26.487066514922094739858 94874204	**26.486362150977605267598 846694455**	26.486362578809156786974 199349061
Welded Beam	1.7264484827726871429831 589921378	1.7265839611184765622908 798832214	**1.7256157610909406141530 55321367**
Best in	**1**	**6**	**10**

Table 11. Summary of Results

Algorithm	Min	Mean	SD	Algorithm	Min	Mean	SD
F1-F7				CEC2019			
GWO	5	5	5	GWO	5	3	3
SineL	0	0	0	SineL	2	1	2
GWO:SineL	2	1	2	GWO:SineL	6	8	5
F8-F13				Engineering			
GWO	3	2	3	GWO	1	1	1
SineL	1	1	1	SineL	6	4	4
GWO:SineL	2	3	2	GWO:SineL	10	11	11
F14-F19				Negative			
GWO	0	0	2	GWO	1	1	1
SineL	4	6	3	SineL	1	1	2
GWO:SineL	2	0	1	GWO:SineL	3	3	2
	Min	Mean	SD	Total			
GWO	15	12	15	42			
SineL	14	13	12	39			
GWO:SineL	25	26	23	74			

4.3. Validation to Supply Chain Problems

Finally, to evaluate the effectiveness of the GWO:SineL algorithm, three supply chain management problems are considered.

Initially, to validate the slightly modified algorithm, one random dataset [17] is considered (Table 12).

The compared results are presented in Table 13. "Mass Center" refers to the average of the latitudes and longitudes. Since the data points are located on the Earth's surface, the Great Circle Distance (GCD) is considered here instead of Euclidean Distance (ED). For the mini-sum problem, total distance reported is 67835.95306 km, which exactly matches the value reported by the Four Point Search Algorithm 'AB'. For the mini-max case, the optimal radius is 12455.35164 km, which is slightly better than that of the Four Point Search Algorithm 'AB'. In this case, the total distance reported is 78838.90454 km, which is higher than the mini-sum case. The facility locations are also presented in Table 13 in terms of degrees. This shows that the GWO:SineL algorithm performs satisfactorily for this category of optimization problems.

Table 12. Random Dataset

Location	Latitude, deg.	Longitude, deg.
Yukon, Canada	60.170638	-130.827364
Kamchatka Krai, Russia	62.424437	169.684973
Durazno, Uruguay	-33.195543	-55.429532
London, UK	51.5085300	-0.1257400
Ihosy, Madagascar	-22.488918	45.657826
Hamrin Mountain, Iraq	35.050944	43.636343
Nenets Autonomous Okrug, Russia	68.031820	61.372730
Thellai, India	12.776006	79.028060
Omakau, New Zealand	-45.062944	169.629765
Hulunbuir, Inner Mongolia, China	49.753488	124.590197
Mass Center		
Al Udayd Saudi Arabia	**23.8968458**	**50.7217258**

Table 13. Random Dataset – Four Point Search Algorithm ‘AB’ vs GWO:SineL

. Algorithm	Facility Latitude, deg.	Facility Longitude, deg.	Total Distance, km (GCD)
Mass Center: [23.89807, 50.72395] degrees			
Mini-Sum (Total Distance)			
Four Point Search Algorithm ‘AB’	67.29389	62.36073	67835.95306
GWO:SineL Algorithm	67.29389	62.36073	67835.95306
Mani-Max (Radius)			
Four Point Search Algorithm ‘AB’	7.60258	55.95105	12455.35575 (Total: 78838.91365)
GWO:SineL Algorithm	7.60258	55.95162	12455.35164 (Total: 78838.90454)

The compared results are presented in Table 13. “Mass Center” refers to the average of the latitudes and longitudes. Since the data points are located on the Earth’s surface, the Great Circle Distance (GCD) is considered here instead of Euclidean Distance (ED). For the mini-sum problem, total distance reported is 67835.95306 km, which exactly matches the value reported by the Four Point Search Algorithm ‘AB’. For the mini-max case, the optimal radius is 12455.35164 km, which is slightly better than that of the Four Point Search Algorithm ‘AB’. In this case, the total distance reported is 78838.90454 km, which is higher than the mini-sum case. The facility locations are also presented in Table 13 in terms of degrees. This shows that the GWO:SineL algorithm performs satisfactorily for this category of optimization problems.

4.4. Real-World Supply Chain Problems - States and Union Territories of India

Finally, one real-world supply chain problem is considered. The objectives are (i) to locate a facility that minimizes the total distance to connect the headquarters of 36 Indian states and union territories (data points) (ii) to locate another facility such that the radius from the optimal location is minimized to cover all the 36 data points and (iii) to connect all 36 data points by considering this as a travelling salesman problem (TSP).

The latitude and longitude of all 36 data points are collected (Table 14). The “Mass Center” is initially computed, which is the mean of all data points. This approximate center is iteratively moved towards the optimal facility location in “Mini-Sum” and “Mini-Max” cases separately.

The results are presented in Table 15.

The facility is located at Sirkhola, Chhattisgarh, for the mini-sum case and located at Padkidih, Chhattisgarh, for the mini-max case. These two centers are separated by a distance of 230.70 km (GCD).

Table 14. States and Union Territories of India, Headquarters and their Coordinates

State/UT	Headquarters	Latitude (degrees)	Longitude (degrees)
Andhra Pradesh	Amaravati	16.3923	80.1740
Arunachal Pradesh	Itanagar	27.9890	93.2202
Assam	Dispur	26.2014	91.2493
Bihar	Patna	25.2503	85.5827
Chhattisgarh	Raipur	21.6120	81.5032
Goa	Panaji	15.3896	73.7569
Gujarat	Gandhinagar	23.0902	72.6224
Haryana	Chandigarh	30.5157	76.4612
Himachal Pradesh	Shimla	31.2262	77.5498
Jharkhand	Ranchi	23.5142	85.3478
Karnataka	Bengaluru	12.9374	77.3608
Kerala	Thiruvananthapuram	8.5600	76.4841
Madhya Pradesh	Bhopal	23.3996	77.3321
Maharashtra	Mumbai	19.1368	72.5766
Manipur	Imphal	24.9351	93.6270
Meghalaya	Shillong	25.5596	91.7649
Mizoram	Aizawl	23.5027	92.3608
Nagaland	Kohima	25.8175	94.7443
Odisha	Bhubaneswar	20.2598	85.6572
Punjab	Chandigarh	30.5157	76.4612
Rajasthan	Jaipur	26.6253	75.3669
Sikkim	Gangtok	27.1754	88.4819
Tamil Nadu	Chennai	13.0634	80.3115
Telangana	Hyderabad	17.6356	78.1514
Tripura	Agartala	23.4913	91.5988
Uttar Pradesh	Lucknow	26.6941	80.6610
Uttarakhand	Dehradun	30.4470	77.9394
West Bengal	Kolkata	22.5917	88.3673
Andaman and Nicobar Islands	Port Blair	11.5394	92.6301
Chandigarh	Chandigarh	30.5157	76.4612
Dadra and Nagar Haveli and Daman and Diu	Daman	20.7010	72.7427
Delhi	New Delhi	28.6192	77.1087
Jammu and Kashmir	Srinagar (Summer)	34.3374	74.4845
Ladakh	Leh	34.6067	77.5899
Lakshadweep	Kavaratti	10.5653	72.5021
Puducherry	Puducherry	11.8373	79.2573

Table 15. States and Union Territories of India - Performance of GWO:SineL Algorithm

S. No.	Algorithm	Latitude, Longitude	Location	Distance, km (GCD)
Mass Center: [0.4006, 1.4251] radians (22.953 81.652) degrees, Location: Dharamdas, Madhya Pradesh				
Mini-Sum (Total Distance)				
	GWO:SineL	[23.68122, 81.68657] deg.	Sirkhola, Chhattisgarh	Sum: 35345.56740
Distance from mass center = 81.085 km (GCD)				
Mani-Max (Radius)				
	GWO:SineL	[21.62641, 81.99711] deg.	Padkidih, Chhattisgarh	Radius: 1592.98245 (Sum: 35996.95720)
Distance from mass center = 151.68 km (GCD)				
Distance between Radius center and Sum center = 230.70 km (GCD)				
Travelling Salesman (Total Distance, GCD)				
NEH Cost: 17740 km				
Total Distance (Cost) = 16123.9509997 km				
Mean Cost: 16147.0692212 km				
Standard Deviation: 115.82				
Number of Times the Cost betters the NEH Cost = 156/200 (78%)				
Optimal Sequence = [7-31-14-6-35-12-11-36-23-1-24-5-19-29-28-25-17-15-18-2-16-3-22-4-10-26-13-21-32-20-30-8-27-9 -34-33-7], Numbers refer to the serial number in Table 11				
Number of Times the Cost betters the NEH Cost = 156/200				

Next, the TSP problem is solved. In this case, the NEH algorithm [50] is used to compute the seed solution, which is refined further by conducting a larger number of trials.

For the TSP, the strategy followed is slightly different. The seed solution is the NEH cost. The NEH algorithm is mostly applied for makespan minimization in permutation flowshop scheduling problems. The modified strategy applied to TSP is:

(i) For each data point, the sum of distances from all other points are computed.

(ii) The sequence is sorted in non-increasing order of their total distance (cost).

(iii) Two points with highest costs are considered as the initial partial sequence.

(iv) Other points are inserted one by one at a place that minimizes the total cost.

(v) This is termed as the “NEH Cost” and taken as the “Best Cost”, initially. The tour is considered as the initial “Best Tour”.

Then the iteration starts:

(a) First two points are taken from the “Best Tour” and is the initial partial sequence.

(b) One sequence with a set of random numbers equal to the number of data points (n) is generated using the updating expression in the first phase. They are converted to integers from 1 to n.

(c) Other points, except the initial partial sequence, are inserted one by one from this random sequence at a place that minimizes the total cost.

(d) If the total cost is better than that of NEH, it is updated as the new “Best Cost” and the corresponding tour as the new “Best Tour”.

(e) Then the second phase starts and the steps from (a) to (d) are repeated.

(f) This process is continued till the termination condition is met.

The “Best Cost” and “Best Tour” are stored for each iteration. The simulation is continued for the fixed number of iterations and the statistical measures are computed for the “Best Costs”.

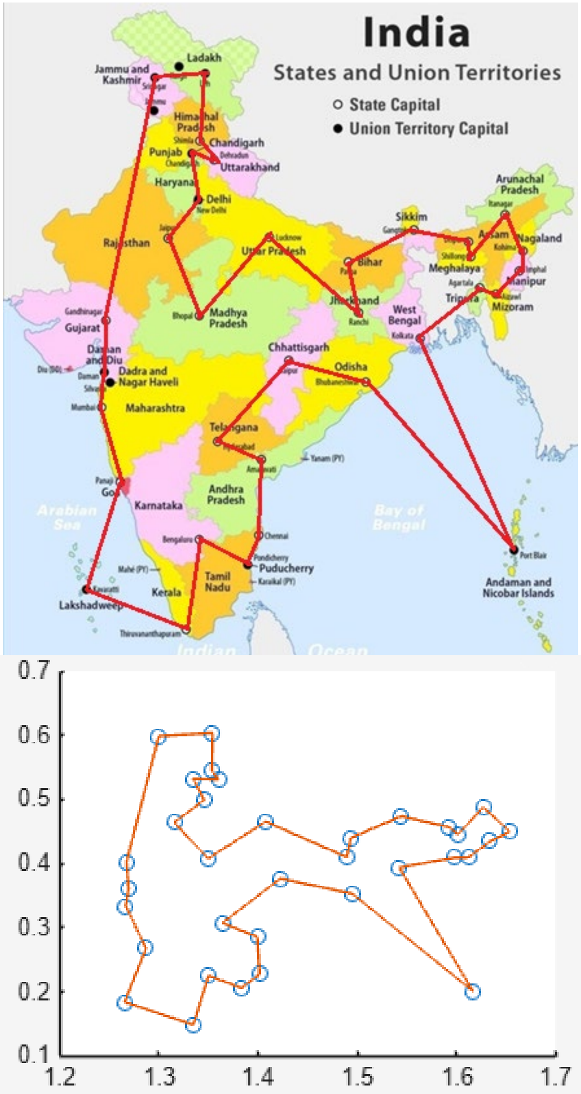


Figure 5. Connecting the Headquarters of the State and Union Territories of India (2026)

Two simulations are carried out, 200 trials and 10000 trials. In both the simulations, the minimum total cost is reported to be 16123.9509997 km (GCD). The mean cost comes down to 16125.3008910 km and the standard deviation to 29.77 when the trials are increased to 10000. Also, the number of times the results are better than those of NEH is 7565/10000 (76%). The graph obtained from the simulation is superimposed on the India map and is presented in Fig. 5.

5. Conclusion and Future Work

This paper analysed two two-phase population-based optimization algorithms, "C-Sine" and "GWO:SineL," across a wide range of benchmark and real-world problems. The "C-Sine" algorithm was evaluated on 67 benchmark datasets, including unconstrained, constrained, and CEC2019 problems, using five of its variants with and without tuning. The results showed that algorithm performance varies across datasets, and improvement strategies effective for one problem may not generalize to others. Among the variants, the "Two-Phase with tuning" and "Phase II with tuning" approaches demonstrated strong performance compared with existing high-performing algorithms, while preliminary experiments also highlighted their potential for multi-objective optimization. In addition, one new hybrid algorithm, "GWO:SineL" algorithm, which combines GWO in the first phase with the "SineL" update mechanism in the second phase, was proposed and tested on 50 benchmark and constrained engineering problems.

The results confirmed that the hybrid approach consistently outperformed standalone "GWO" and "SineL" under equal function evaluations. Its effectiveness was further validated through real-world supply chain applications involving Indian states and union territories, including mini-sum, mini-max, and TSP problems. Overall, the study demonstrates that carefully designed multi-phase, hybrid and diverse population updating strategies can significantly enhance search performance and solution quality by improving the exploration and exploitation capabilities. The combination can be updating strategies of two different popular algorithms, one popular and another new or both can be new and untested. The future work will focus on extending these approaches to broader benchmark suites, multi-objective or many-objective optimization problems and other domains of operations research.

References

[1] Hu G, Yang R, Qin X, Wei G. MCSA: Multi-strategy boosted chameleon-inspired optimization algorithm for engineering applications. Computer Methods in Applied Mechanics and Engineering. 2023 Jan 1;403:115676.

[2] Categorization of optimization problems. https://www.supplychaindataanalytics.com/categorization-of-mathematical-optimization-problems/ (Accessed on 17 October 2023).

[3] Montoya OD, Molina-Cabrera A, Gil-González W. A possible classification for metaheuristic optimization algorithms in engineering and science. Ingeniería. 2022 Dec;27(3).

[4] Kennedy J, Eberhart R. Particle swarm optimization (PSO). InProc. IEEE international conference on neural networks, Perth, Australia 1995 Nov 27 (Vol. 4, No. 1, pp. 1942-1948).

[5] Mirjalili S, Lewis A. The whale optimization algorithm. Advances in engineering software. 2016 May 1;95:51-67.

[6] Dorigo M, Birattari M, Stutzle T. Ant colony optimization. IEEE computational intelligence magazine. 2006 Nov 30;1(4):28-39.

[7] Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems. Knowledge-based systems. 2016 Mar 15;96:120-33.

[8] Baskar A. Sine (B): a single randomized population-based algorithm for solving optimization problems. Materials today: Proceedings. 2022 Jan 1;62:4745-51.

[9] Baskar A. New simple trigonometric algorithms for solving optimization problems. Journal of Applied Science and Engineering. 2022 Mar;25(6):1257-72.

[10] Baskar A, Xavior MA, Jeyapandiarajan P, Batako A, Burduk A. A novel two-phase trigonometric algorithm for solving global optimization problems. Annals of Operations Research. 2024 Mar 21:1-51.

[11] Hashim FA, Hussain K, Houssein EH, Mabrouk MS, Al-Atabany W. Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. Applied intelligence. 2021 Mar;51(3):1531-51.

[12] Hatamlou A. Black hole: A new heuristic optimization approach for data clustering. Information sciences. 2013 Feb 10;222:175-84.

[13] Kirkpatrick S, Gelatt Jr CD, Vecchi MP. Optimization by simulated annealing. science. 1983 May 13;220(4598):671-80.

[14] Bayzidi H, Talatahari S, Saraee M, Lamarche CP. Social network search for solving engineering optimization problems. Computational Intelligence and Neuroscience. 2021;2021(1):8548639.

[15] Rao RV, Savsani VJ, Vakharia DP. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Computer-aided design. 2011 Mar 1;43(3):303-15.

[16] Salih SQ, Alsewari AA. A new algorithm for normal and large-scale optimization problems: Nomadic People Optimizer. Neural Computing and Applications. 2020 Jul;32(14):10359-86.

[17] Baskar A, Anthony Xavior M. A four-point direction search heuristic algorithm applied to facility location on plane, sphere, and ellipsoid surfaces. Journal of the Operational Research Society. 2021 Dec 13;73(11):2385-94.

[18] Baskar A, Anthony Xavior M. A Simple Hybrid Local Search Algorithm for Solving Optimization Problems. In International Conference on Computer, Communication, and Signal Processing 2023 Jan 4 (pp. 246-259). Cham: Springer Nature Switzerland.

[19] Moghdani R, Salimifard K. Volleyball premier league algorithm. Applied Soft Computing. 2018 Mar 1;64:161-85.

[20] Zeidabadi FA, Dehghani M. POA: Puzzle Optimization Algorithm. International Journal of Intelligent Engineering & Systems. 2022 Jan 1;15(1).

[21] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Advances in engineering software. 2014 Mar 1;69:46-61.

[22] Chandran V, Mohapatra P. Enhanced opposition-based grey wolf optimizer for global optimization and engineering design problems. Alexandria Engineering Journal. 2023 Aug 1;76:429-67.

[23] Rajak MK, Meitei IC, Pudur R. Hybrid PSO-GWO optimization for converter-free grid integration of parallel induction generators with experimental validation in micro-hydro power plants. Scientific Reports. 2026 Apr 20.

[24] Raut N, Karapagale A, Gangwar A, Goud H, Bhargaw HN. A hybrid grey wolf bee colony optimized PID controller for improved transient response and stability of lower limb exoskeleton system. Discover Robotics. 2026 Dec;2(1):9.

[25] Zhang F, Liu C, Liu P, Ding S, Qiu T, Wang J, Du H. Multiple strategy enhanced hybrid algorithm BAGWO combining beetle antennae search and grey wolf optimizer for global optimization. Scientific Reports. 2025 May 2;15(1):15460.

[26] Sanjay R, Jayabarathi T, Raghunathan T, Ramesh V, Mithulananthan N. Optimal allocation of distributed generation using hybrid grey wolf optimizer. Ieee Access. 2017 Jul 21;5:14807-18.

[27] Singh N, Singh SB. A novel hybrid GWO-SCA approach for optimization problems. Engineering science and technology, an international journal. 2017 Dec 1;20(6):1586-601.

[28] Xu H, Liu X, Su J. An improved grey wolf optimizer algorithm integrated with Cuckoo Search. In2017 9th IEEE international conference on intelligent data acquisition and advanced computing systems: technology and applications (IDAACS) 2017 Sep 21 (Vol. 1, pp. 490-493). IEEE.

[29] Ahmad I, Qayum F, Rahman SU, Srivastava G. Using improved hybrid grey wolf algorithm based on artificial bee colony algorithm onlooker and scout bee operators for solving optimization problems. International Journal of Computational Intelligence Systems. 2024 May 6;17(1):111.

[30] Baskar A. Performance Analysis of a Two-Phase Algorithm with Tuning Option for Solving Optimization Problems. SSRN (April 25, 2024). DOI: 10.2139/ssrn.4807467.

[31] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE transactions on evolutionary computation. 2002 Aug 6;1(1):67-82.

[32] Baskar A. Improving the Grey Wolf Optimizer (GWO) by Hybrid Approach: An analysis, SSRN (May 30, 2025). DOI: 10.2139/ssrn.5275944..

[33] Solve a Mixed-Integer Engineering Design Problem Using the Genetic Algorithm. https://in.mathworks.com/help/gads/solving-a-mixed-integer-engineering-design-problem-using-the-genetic-algorithm.html (Accessed on 11 November 2022)

[34] Deb K, Goyal M. Optimizing Engineering Designs Using a Combined Genetic Search. In Icga 1997 Jul 19 (pp. 521-528).

[35] Michalewicz Z, Schoenauer M. Evolutionary algorithms for constrained parameter optimization problems. Evolutionary computation. 1996 Mar 1;4(1):1-32.

[36] Price KV, Awad NH, Ali MZ, Suganthan PN. The 100-digit challenge: problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization. Nanyang Technological University. 2018 Nov;1:1-21.

[37] Optimization. Benchmarks · Unconstrained. https://www.al-roomi.org/benchmarks/unconstrained (Accessed on 4 November 2023).

[38] Optimization Test Problems. https://www.sfu.ca/~ssurjano/optimization.html (Accessed on 7 November 2023).

[39] Azizi M, Talatahari S, Giaralis A. Optimization of engineering design problems using atomic orbital search algorithm. IEEE Access. 2021 Jul 12;9:102497-519.

[40] Eskandar H, Sadollah A, Bahreininejad A, Hamdi M. Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems. Computers & Structures. 2012 Nov 1;110:151-66.

[41] Talatahari S, Azizi M. Optimization of constrained mathematical and engineering design problems using chaos game optimization. Computers & Industrial Engineering. 2020 Jul 1;145:106560.

[42] MiarNaeimi F, Azizyan G, Rashki M. Multi-level cross entropy optimizer (MCEO): an evolutionary optimization algorithm for engineering problems. Engineering with Computers. 2018 Oct;34(4):719-39.

[43] Yang XS, Deb S. Engineering optimisation by cuckoo search. International Journal of Mathematical Modelling and Numerical Optimisation. 2010 Jan 1;1(4):330-43.

[44] Kaveh A, Eslamlou AD. Water strider algorithm: A new metaheuristic and applications. In Structures 2020 Jun 1 (Vol. 25, pp. 520-541). Elsevier.

[45] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary computation. 2000 Jun;8(2):173-95.

[46] Vlennet R, Fonteix C, Marc I. Multicriteria optimization using a genetic algorithm for determining a Pareto set. International Journal of Systems Science. 1996 Feb 1;27(2):255-60.

[47] Baskar A. Simple population-based algorithms for solving optimization problems. RAIRO-Operations Research. 2024 Jan 1;58(1):253-80.

[48] Chawla M, Duhan M. Levy flights in metaheuristics optimization algorithms–A review. Applied Artificial Intelligence. 2018 Nov 26;32(9-10):802-21.

[49] Abdullah JM, Ahmed T. Fitness dependent optimizer: inspired by the bee swarming reproductive process. IEEe Access. 2019 Mar 22;7:43473-86.

[50] Nawaz M, Enscore Jr EE, Ham I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. Omega. 1983 Jan 1;11(1):91-5.

.