FMDB *Transactions on Sustainable Structures and Materials*

Discrete Gear-Train Ratio Optimization using the Scuba Diver Optimization Algorithm: Exact Optimum Validation and Search Dynamics Analysis

Saman M. Almufti[1,*], Amira Bibo Sallow[2]

[1]Department of Information Technology, Technical College of Informatics-Akre, Akre University for Applied Sciences, Akre, Nineveh Governorate, Iraq.
[1,2]Department of Information Technology, Technical College of Duhok, Duhok Polytechnic University, Duhok, Iraq.
saman.almofty@gmail.com[1], amira.bibo@nawroz.edu.krd[2]

Abstract: The compound gear-train benchmark is a discrete engineering optimisation problem in which four integer tooth counts must approximate a prescribed transmission ratio. Although the mathematical model is compact, the attainable objective values are highly discontinuous, and several recent reports have produced artificially smaller values by treating tooth counts as continuous variables. This study develops an integer-preserving Scuba Diver Optimisation Algorithm (SDOA) for the benchmark and evaluates ten parameter configurations through 30 independent runs per configuration. Every movement operator is followed by rounding and bound repair, and the best-known solution is verified by exhaustive enumeration of the 49^4 admissible designs. The strongest setting, S10, used 150 divers and 600 iterations. It obtained a mean objective of 3.935622e-10, a median of 2.307816e-11, and the lowest mean rank of 3.617. It reached f <= 1e-9 in 86.67% of runs, f <= 1e-10 in 60.00%, and the exact discrete optimum in 13.33%. The best design [19, 16, 43, 49] is symmetric to [16, 19, 43, 49] and gives f = 2.700857148886513e-12. A Friedman test detected a configuration effect (chi-square (9) = 120.972634, p = 8.447286e-22, Kendall W = 0.448047). Holm-adjusted Wilcoxon tests confirmed that S10 outperformed the three lowest-budget settings, while several higher-budget alternatives remained statistically competitive. Convergence and search-stage records show that the selected configuration allocated 30.90% of diver-iterations to fine-tuning and limited resets to 13.76%. The results establish a reproducible discrete SDOA implementation and show that parameter selection, integer repair, and exact-optimum verification materially affect conclusions for this benchmark.

Keywords: Discrete Optimisation; Engineering Design; Gear Train; Integer Repair and Metaheuristics; Parameter Sensitivity; Scuba Diver Optimisation Algorithm; Nonparametric Statistics.

Received on: 07/05/2025, **Revised on:** 22/08/2025, **Accepted on:** 13/09/2025, **Published on:** 05/06/2026

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSSM

DOI: https://doi.org/10.69888/FTSSM.2026.000753

Cite as: S. M. Almufti and A. B. Sallow, "Discrete Gear-Train Ratio Optimisation using the Scuba Diver Optimisation Algorithm: Exact Optimum Validation and Search Dynamics Analysis," *FMDB Transactions on Sustainable Structures and Materials*, vol. 2, no. 1, pp. 1–16, 2026.

1. Introduction

Gear trains transmit rotational motion and modify angular speed and torque through meshing toothed wheels. In the preliminary design, the required overall ratio may be prescribed before detailed decisions on modules, face width, strength, lubrication, and manufacturing are made. The benchmark examined here isolates the ratio-selection task: four integer tooth counts must be

*Corresponding author.

chosen so that a compound gear train approximates 1/6.931 as closely as possible. This problem was introduced in early work on nonlinear integer and discrete mechanical design [1] and later used to assess mixed-variable optimisation procedures [2]; [3]. The search domain contains 49^4 = 5,764,801 ordered tooth combinations. Direct enumeration is feasible for verification but is an unsuitable general strategy when the gear model is extended to include additional stages, catalogue restrictions, interference conditions, centre-distance requirements, load capacity, efficiency, noise, or reliability. The benchmark therefore remains useful as a controlled test of how an optimiser represents and searches integer decisions. Its objective is inexpensive to evaluate, but the landscape is discontinuous: a small continuous displacement has no engineering meaning until it changes at least one rounded tooth count. Population-based metaheuristics are often applied to engineering problems because they do not require derivatives and can incorporate problem-specific encodings and repair rules [4]; [5]. Their performance, however, depends on representation, operator scale, boundary handling, elitism, termination criteria, and parameter values. Reviews of metaheuristic development have also emphasised the large number of algorithmic metaphors and the need to distinguish genuinely useful search mechanisms from nominal novelty [6]; [7].

No-free-lunch results further imply that an algorithm cannot be expected to dominate across all optimisation classes [8]. A particular difficulty in the gear-train benchmark is the fidelity of the formulation. Tooth counts are integers. If an implementation evaluates non-integer positions without rounding, it solves a continuous relaxation and may report zero or values smaller than the verified discrete minimum. Such values are not improved gear designs; they belong to a different mathematical problem. Several recent studies state integer bounds but subsequently present non-integer tooth counts [9]; [10]. This inconsistency makes explicit repair, exact verification, and transparent reporting necessary. This paper applies the Scuba Diver Optimisation Algorithm (SDOA) to the discrete gear-train ratio problem. SDOA represents candidate solutions as divers, whose oxygen states determine five search stages: global exploration, moderate exploration, exploitation, fine-tuning, and reset diversification. The implementation preserves the integer domain after every operator. Ten parameter configurations are compared under 30 matched-seed runs, and the analysis integrates objective distributions, precision thresholds, convergence histories, exact-optimum hit rates, stage allocation, runtime, Friedman ranks, and Holm-adjusted pairwise tests. The study makes four contributions. First, it provides a complete integer-preserving formulation of SDOA for gear-train design. Second, it verifies the discrete global optimum by exhaustive enumeration and uses it to define exact-hit and accuracy criteria. Third, it separates the best isolated solution from repeated-run reliability through a ten-configuration sensitivity experiment. Fourth, it links outcome differences to the internal use of SDOA search stages, providing evidence about why the selected configuration behaves differently from lower-budget alternatives.

2. Related Work and Methodological Context

2.1. Discrete Gear-Train Optimisation

Sandgren formulated nonlinear integer and discrete mechanical design problems and used the compound gear train as an illustrative benchmark [1]. Kannan and Kramer subsequently applied an augmented Lagrange multiplier framework to mixed integer, discrete, and continuous mechanical design [2]. These studies established the central representation issue: the number of teeth is not a continuous geometric variable and must remain within a finite set of integers.

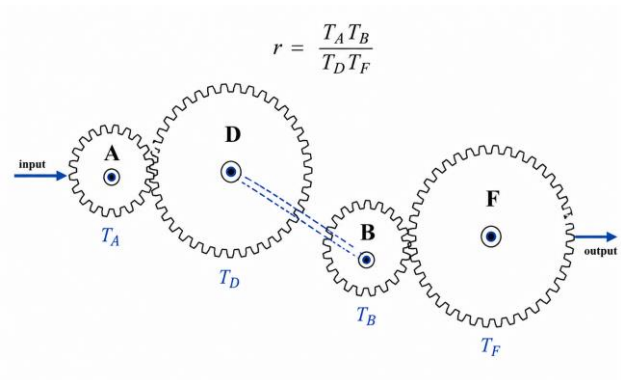

Figure 1: Compound gear-train benchmark and decision variables

Sharif et al. [3] developed a discrete mode-pursuing sampling method for expensive black-box functions and used the gear-train case to demonstrate search in a discrete domain. Their discussion is directly relevant to the present study because integer problems combine combinatorial structure with nonconvex response behaviour. For the current four-variable benchmark, exhaustive enumeration can verify the optimum; for more expensive extensions, the same combinatorial structure motivates efficient stochastic search.

2.2. Population-Based Optimisation for Engineering Design

The principal population-based methods used in engineering optimisation include particle swarm optimisation (PSO) [11], differential evolution (DE) [12], grey wolf optimisation (GWO) [13], whale optimisation (WOA) [14], sine cosine optimisation (SCA) [15], salp swarm optimisation (SSA) [16], moth-flame optimisation (MFO) [17], multi-verse optimisation (MVO) [18], and Harris hawks optimisation (HHO) [19]. Their update equations differ, but each method must coordinate a broad search with local refinement while preserving valid engineering decisions. Algorithm surveys classify these methods as evolutionary, swarm, physics-based, human-based, or other-inspired, while noting that classification does not determine performance [6]; [7]; [20]. Constrained and discrete engineering applications additionally require feasibility or representation mechanisms. Work on constrained mechanical design has shown that the treatment of variable types and constraints can influence results as strongly as the nominal search algorithm [21]. Parameter investigations on other mechanical benchmarks also show that population size, movement scale, and iteration budget should be evaluated rather than adopted without evidence [22].

2.3. Recent Gear-Train Results and Formulation Consistency

Recent studies continue to use the four-gear benchmark. Rao and Zinzuvadia [21] compared Rao variants with CS, ALO, MFO, and MVO and explicitly noted that a result smaller than the discrete optimum was infeasible because the number of teeth had been treated as continuous [23]. Biswas et al. [6] evaluated enhanced prairie dog optimisation (EPDO) with PDO, SCA, WOA, MFO, and RSA while retaining integer bounds [24]. Ghasemi et al. [22] reported the verified optimum for an improved firefly algorithm [25], and Eker compared the Capuchin Search Algorithm with GA and a hybrid SCA-GWO approach [26]. Other recent papers state integer limits but report non-integer tooth counts, including studies of an improved exponential distribution optimiser and a hybrid coati-DE optimiser [9]; [10]. These results illustrate why the mathematical statement alone is insufficient: the computational implementation must round or encode the variables before objective evaluation. The present work therefore explicitly reports the repair operation, verifies the reference optimum independently, and does not interpret continuous-relaxation values as valid discrete improvements.

2.4. Statistical Evaluation of Stochastic Optimisers

A best run demonstrates possibility, not reliability. Stochastic algorithms should be evaluated over repeated runs using measures of central tendency, dispersion, success probability, and convergence [27]. When the distributional assumptions of parametric tests are uncertain, rank-based procedures are appropriate. The Friedman test evaluates multiple paired algorithms or parameter settings [28], the Wilcoxon signed-rank test evaluates paired differences [29], and the Holm procedure controls the familywise error rate in multiple comparisons [30]. These methods are used in the present parameter study.

2.5. Gear-Train Design Formulation

The compound train contains two meshing pairs. Gears D and B are mounted on a common intermediate shaft. With the notation used in Figure 1, the achieved transmission ratio is:

$$r(\mathbf{x}) = \frac{T_A T_B}{T_D T_F} \tag{1}$$

Where T_A, T_B, T_D, and T_F are integer tooth counts. The design vector is:

$$\mathbf{x} = [T_A, T_B, T_D, T_F] \tag{2}$$

And the target ratio is:

$$r_t = \frac{1}{6.931} = 0.144279324772760 \ldots \tag{3}$$

The objective minimises the squared ratio error:

$$\min f(\mathbf{x}) = [r_t - r(\mathbf{x})]^2 \tag{4}$$

The variable domains are:

$$T_i \in \{12,13, \ldots ,60\}, \qquad i \in \{A, B, D, F\} \tag{5}$$

The problem has no additional inequality constraint, but the integer and bound requirements are hard feasibility conditions. A candidate with a fractional tooth count is not physically admissible (Table 1).

Table 1: Gear-train decision variables and admissible domains

Variable	Description	Type	Lower bound	Upper bound
T_A	Teeth on input gear A	Integer	12	60
T_B	Teeth on intermediate gear B	Integer	12	60
T_D	Teeth on intermediate gear D	Integer	12	60
T_F	Teeth on output gear F	Integer	12	60

A repaired continuous vector y is mapped to the feasible domain by:

$$R(y_j) = \min\left(60, \max\left(12, \text{round}(y_j)\right)\right) \quad (6)$$

The objective is symmetric under exchange of the two numerator gears and under exchange of the two denominator gears:

$$f(T_A, T_B, T_D, T_F) = f(T_B, T_A, T_D, T_F) = f(T_A, T_B, T_F, T_D) \quad (7)$$

Exhaustive enumeration of all 5,764,801 ordered designs gives the product pair T_A T_B = 304 and T_D T_F = 2107. Representative optimum [16, 19, 43, 49] gives r = 0.144280968201234, absolute error 1.643428473918629e-06, and f* = 2.700857148886513e-12. The four symmetric permutations are equivalent global solutions.

3. Scuba Diver Optimisation Algorithm

3.1. Population Representation and Initialisation

Each diver stores an integer design vector, an objective value, an oxygen level, and a depth-stage label. A stratified initialisation distributes candidates across each tooth-count interval and replaces approximately one-third of the population with uniformly sampled integer designs. For a temporary continuous sample u_ij, initialisation is written as:

$$u_{ij} = L_j + \frac{(i-1) + \text{rand}}{N}(U_j - L_j) \quad (8)$$

Followed by the integer repair in Eq. (6). The best objective defines the global guide. Elite divers are copied before movement and restored after each iteration, which prevents deterioration of the best solutions already found.

3.2. Adaptive Movement and Oxygen State

Two global control variables decrease with iteration. The mutation probability and current strength are:

$$\mu(t) = \mu_0\left(1 - \frac{t}{T}\right) \quad (9)$$

$$c(t) = c_0\left(1 - \frac{t}{T}\right)^2 \quad (10)$$

Where t is the current iteration, and T is the maximum iteration count. The oxygen state of diver i is updated by:

$$O_i(t) = O_i(t-1)\exp\left(-\frac{\alpha t}{T}\right) \quad (11)$$

Improvement adds five oxygen units up to the initial level. The resulting oxygen value selects one of five stages: D1 for O > 75, D2 for 55 < O <= 75, B1 for 35 < O <= 55, D3 for 15 < O <= 35, and reset for O <= 15.

3.3. Stage-Specific Operators

3.3.1. D1 Global Exploration

D1 applies a heavy-tailed displacement to permit large changes in gear teeth:

$$\mathbf{y}_i = \mathbf{x}_i + c(t)(\mathbf{U} - \mathbf{L})\tan[\pi(\text{rand} - 0.5)]\left(1 - \frac{t}{T}\right) \quad (12)$$

The candidate is repaired before evaluation. The heavy tail creates occasional long jumps even when most changes are moderate.

3.3.2. D2 Moderate Exploration

D2 chooses crossover with the global best with probability 0.7; otherwise, it performs a bounded random walk. Crossover is:

$$\mathbf{y}_i = \lambda\mathbf{x}_i + (1 - \lambda)\mathbf{x}_{\text{best}}, \qquad \lambda \sim \mathcal{U}(0,1) \quad (13)$$

Whereas the random walk is:

$$\mathbf{y}_i = \mathbf{x}_i + \boldsymbol{\delta}, \qquad \delta_j \sim \mathcal{U}\left[-s_j(t), s_j(t)\right], \qquad \mathbf{s}(t) = \max[1, c(t)(\mathbf{U} - \mathbf{L})] \quad (14)$$

3.3.3. B1 Exploitation

B1 first tests one-tooth positive and negative moves in each coordinate. The best improving neighbour is retained. With probability 0.6, the locally improved design is crossed with a randomly selected elite diver. This stage is particularly suitable for the gear problem because the final refinement must compare adjacent integer combinations rather than arbitrarily small continuous steps.

3.3.4. D3 Fine-Tuning

D3 uses a nonuniform mutation whose magnitude decreases with iteration:

$$\Delta(t) = 1 - \text{rand}^{(1-t/T)^b}, \qquad b = 5 \quad (15)$$

The mutation moves a variable toward either its upper or lower bound, after which integer repair is applied. Although the pre-repair displacement is continuous, objective evaluation is always performed using a feasible integer design.

3.3.5. Reset Diversification

When oxygen becomes low, a nonelite diver is reinitialised with probability 0.4. Reset prevents the population from remaining indefinitely around the same product pair. The oxygen level of a reset diver returns to its initial value.

3.4. Communication, Acceptance, and Elitism

A candidate is accepted only when its objective is lower than the stored objective. In addition, a diver may communicate with the global best with probability:

$$p_i(t) = 0.6\frac{O_i(t)}{O_0} \quad (16)$$

Communication samples a repaired design within approximately 10% of the variable range around the current global best. The greedy rule protects improvements, while elite restoration guarantees that the best elite members survive the iteration unchanged.

3.5. Algorithm Sequence and Complexity

For N divers, T iterations, and d = 4 variables, the population updates require O(NTd) arithmetic operations, excluding sorting. Elite identification adds O(TN log N) to the overall complexity in the current implementation. Objective evaluation is constant time. Memory use is O(Nd + NT) when convergence and stage histories are retained; without histories, working memory is O(Nd) (Figure 2).

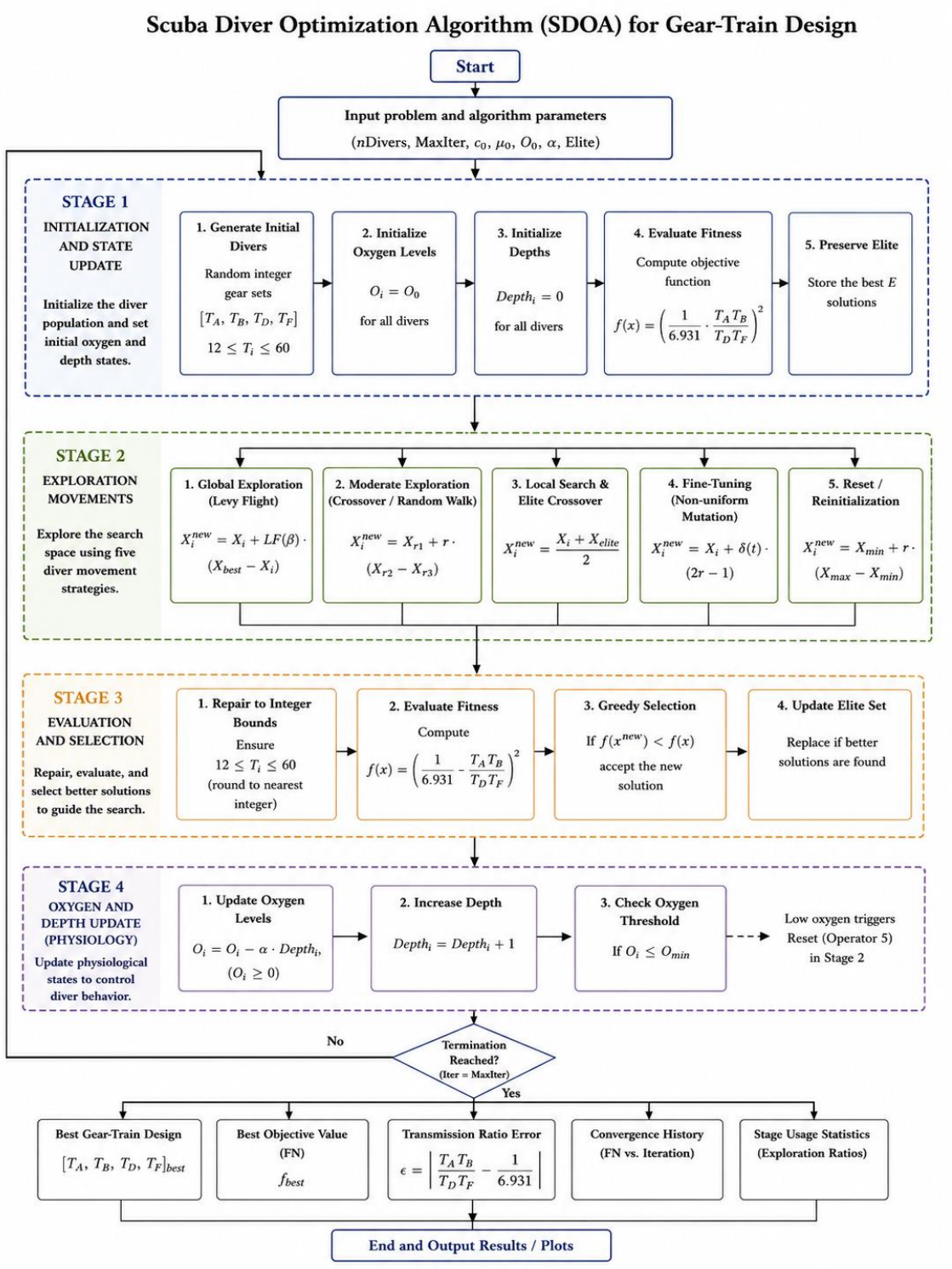

Figure 2: Integer-preserving SDOA procedure for gear-train design

4. Experimental Design

4.1. Parameter Configurations and Run Protocol

Ten settings were constructed to examine population size, iteration budget, current strength, mutation rate, oxygen decay, and elite size. Each setting was executed 30 times. Matched run indices used the same seed sequence across configurations, enabling paired tests. The experiment therefore contains 300 final solutions (Table 2).

Table 2: SDOA parameter configurations

Set	nDivers	maxIter	Initial Current Strength	Initial Mutation Rate	Initial Oxygen Level	Oxygen Decay Alpha	Elite Size
S1	25	50	0.20	0.50	100	0.30	5
S2	40	100	0.25	0.60	100	0.40	8
S3	50	200	0.30	0.70	100	0.50	10
S4	75	300	0.35	0.80	100	0.60	15
S5	100	400	0.30	0.80	100	0.50	20
S6	125	300	0.20	0.90	100	0.70	25
S7	150	400	0.30	0.80	100	0.50	30
S8	100	600	0.40	0.60	100	0.40	20
S9	200	300	0.25	0.90	100	0.70	40
S10	150	600	0.50	0.50	100	0.30	30

All runs used the same tooth bounds, objective, stage thresholds, communication probability, local search, and repair operation. No stopping condition based on the known optimum was used; every run consumed its assigned iteration budget. The global optimum was used only after execution to compute accuracy and exact-hit statistics.

4.2. Performance Indicators

The final performance was summarised as best, mean, median, worst, and standard deviation. Three success indicators were used: f <= 10^-9, f <= 10^-10, and exact equality to the verified optimum within a numerical tolerance of max (10^-18, 10^-8 f*). The first iteration reaching each criterion was also recorded. The mean runtime was measured for the compiled Python/Numba implementation used in the experiment and is reported only as a relative cost across configurations. Convergence curves report the mean incumbent objective. Because iteration budgets differ, the ten-setting comparison uses normalised progress from 0% to 100%. The selected setting is additionally reported by iteration using its mean, median, and interquartile range. Stage usage is the share of all diver-iterations assigned to D1, D2, B1, D3, or reset.

4.3. Statistical Analysis

The objective values were paired by run index and ranked within each run. Lower rank indicates better performance. The Friedman test examined the null hypothesis that all 10 settings have the same performance distribution; Kendall's W quantified agreement in the paired rankings. After rejecting the omnibus null, S10 was compared with the other configurations using two-sided Wilcoxon signed-rank tests and Holm adjustment at alpha = 0.05. Rank-biserial correlations describe paired effect direction and magnitude.

5. Results

5.1. Descriptive Performance

S10 obtained the lowest mean rank (3.617) and the lowest mean objective (3.935622e-10). Its median objective was 2.307816e-11, approximately one order of magnitude above the verified optimum, while its worst run ended at 2.357641e-09 (Table 3).

Table 3: Repeated-run performance of the ten parameter configurations

Set	Best	Mean	Median	Worst	Std.	Mean rank	Success <=1e-9 (%)	Success <=1e-10 (%)	Exact optimum (%)	Mean runtime (s)
S10	2.701e-12	3.936e-10	2.308e-11	2.358e-09	6.515e-10	3.617	86.67	60.00	13.33	0.011545
S8	2.701e-12	6.503e-10	9.940e-11	2.358e-09	8.424e-10	4.100	73.33	56.67	3.33	0.008383
S9	2.701e-12	5.133e-10	1.356e-10	2.358e-09	5.884e-10	4.183	86.67	36.67	3.33	0.008305
S5	2.701e-12	6.418e-10	4.074e-10	2.358e-09	7.327e-10	4.250	80.00	40.00	6.67	0.005747
S7	2.701e-12	7.086e-10	6.602e-10	2.358e-09	7.194e-10	4.317	80.00	30.00	6.67	0.008866
S6	2.701e-12	1.046e-09	7.745e-10	4.503e-09	1.195e-09	4.683	66.67	36.67	10.00	0.005223

S4	2.701e-12	1.122e-09	8.888e-10	4.503e-09	1.233e-09	5.050	63.33	33.33	13.33	0.003170
S3	2.308e-11	3.685e-09	2.358e-09	2.726e-08	5.543e-09	7.517	23.33	6.67	0.00	0.001522
S2	3.068e-10	4.304e-09	2.829e-09	2.726e-08	5.623e-09	7.983	26.67	0.00	0.00	0.000598
S1	1.263e-09	2.911e-08	6.655e-09	2.505e-07	5.581e-08	9.300	0.00	0.00	0.00	0.000218

The mean is larger than the median because a limited number of runs terminated at weaker discrete ratios. The three lowest-budget configurations were clearly less reliable (Figure 3).

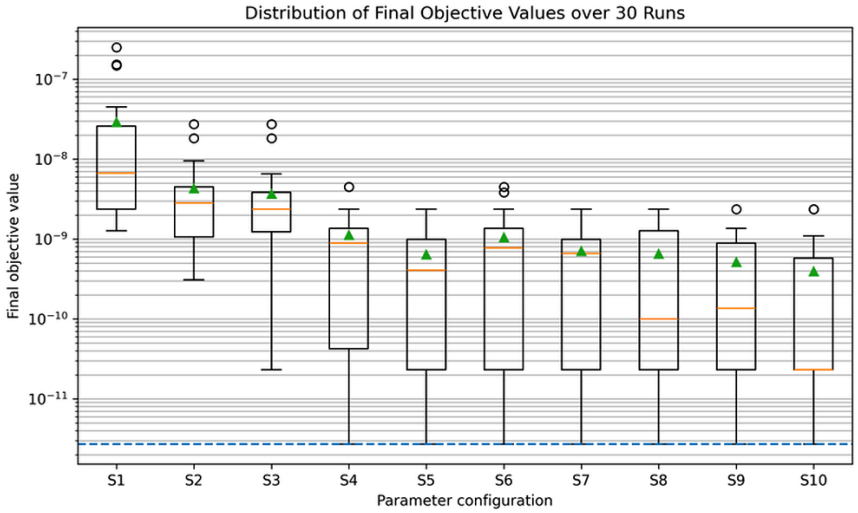

Figure 3: Final-objective distributions over 30 runs

S1 never reached 10^-9; S2 and S3 did not reach the exact optimum. Every configuration from S4 to S10 found an optimum in at least one run, but the distributions across repeated runs remained different. This result demonstrates that reporting only the best value would obscure substantial parameter sensitivity.

5.2. Best Designs and Symmetry

The best S10 design was [19, 16, 43, 49], with a ratio of 0.144280968201234, an absolute error of 1.643428473918629e-06, and an objective of 2.700857148886513e-12 (Figure 4).

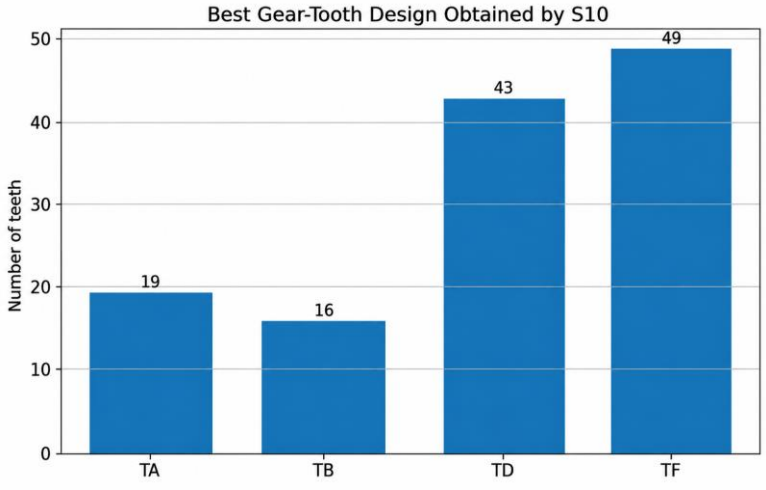

Figure 4: Tooth counts of the best S10 design

It is one of the four symmetric optimum designs. S4-S10 all reached an optimum, whereas the best S3 design [15, 26, 53, 51] remained at 2.307816e-11 (Table 4).

Table 4: Best gear design obtained by each configuration

Set	TA	TB	TD	TF	Ratio	Absolute error	Objective	Reference gap
S1	12	33	49	56	0.144314868805	3.554e-05	1.263e-09	1.261e-09
S2	21	15	59	37	0.144296839212	1.751e-05	3.068e-10	3.041e-10
S3	15	26	53	51	0.144284128746	4.804e-06	2.308e-11	2.038e-11
S4	19	16	43	49	0.144280968201	1.643e-06	2.701e-12	0.000e+00
S5	16	19	49	43	0.144280968201	1.643e-06	2.701e-12	0.000e+00
S6	16	19	43	49	0.144280968201	1.643e-06	2.701e-12	0.000e+00
S7	16	19	43	49	0.144280968201	1.643e-06	2.701e-12	0.000e+00
S8	16	19	49	43	0.144280968201	1.643e-06	2.701e-12	0.000e+00
S9	19	16	43	49	0.144280968201	1.643e-06	2.701e-12	0.000e+00
S10	19	16	43	49	0.144280968201	1.643e-06	2.701e-12	0.000e+00

5.3. Success Rates and Convergence

S10 reached 10^-9 in 86.67% of runs and 10^-10 in 60.00%, the strongest combination of the two thresholds. S9 tied the 10^-9 rate but reached 10^-10 in only 36.67%. S4 had the same exact-optimum percentage as S10, but its mean, median, and rank were weaker. Exact-hit percentage should therefore be interpreted with the full distribution rather than as a standalone criterion (Table 5).

Table 5: Accuracy-based success rates and attainment iterations

Set	Success <=1e-9 (%)	Success <=1e-10 (%)	Exact Optimum (%)	Mean Iter <=1e-9	Mean Iter <=1e-10	Mean Exact-Hit Iter
S10	86.67	60.00	13.33	74.42	126.94	148.00
S8	73.33	56.67	3.33	98.55	108.12	1.00
S9	86.67	36.67	3.33	28.69	53.27	99.00
S5	80.00	40.00	6.67	46.79	72.67	19.00
S7	80.00	30.00	6.67	44.88	75.89	77.50
S6	66.67	36.67	10.00	40.90	50.18	32.00
S4	63.33	33.33	13.33	72.58	91.00	61.75
S3	23.33	6.67	0.00	26.14	43.50	-
S2	26.67	0.00	0.00	13.25	-	-
S1	0.00	0.00	0.00	-	-	-

Figure 5 shows the success rates for ten parameter settings (S1–S10) at three accuracy levels: f 1e9, f 1e10 and exact optimum. The results indicate that the success rate is highest for the condition f <= 1e-9, with S9 and S10 at 87%, S7 at 80%, and S5 at 80%.

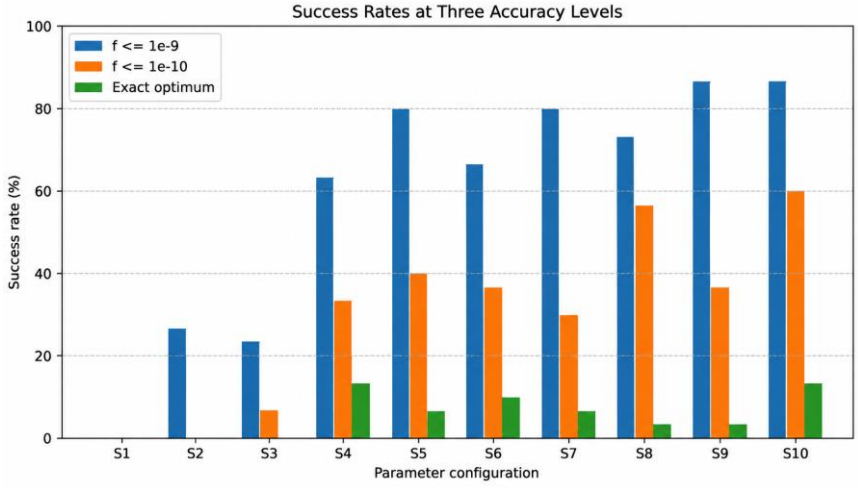

Figure 5: Success rates at three accuracy levels

The success rates are lower with the tighter condition of f <= 1e-10. S10 has the highest success rate at 60%, and S8 has 56%. The Exact optimum condition has the lowest success rates, with S4 and S10 reaching around 13% (Figure 6).

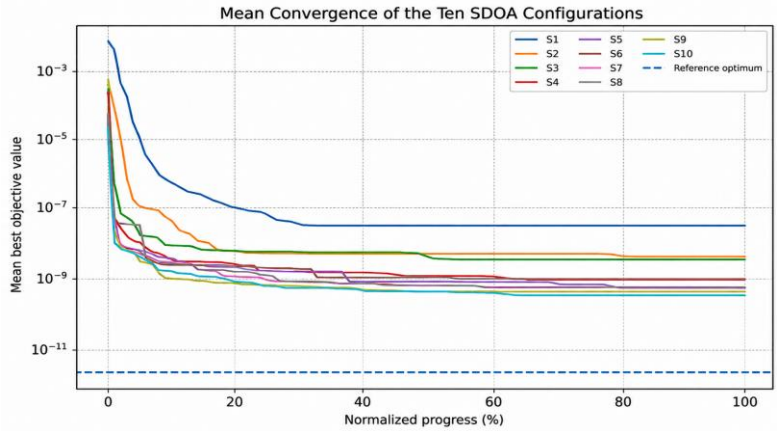

Figure 6: Mean convergence normalised by iteration budget

Overall, the chart shows that the higher the accuracy requirement, the more difficult it is to achieve successful results, while some configurations, such as S9 and S10, show relatively good performance (Figure 7).

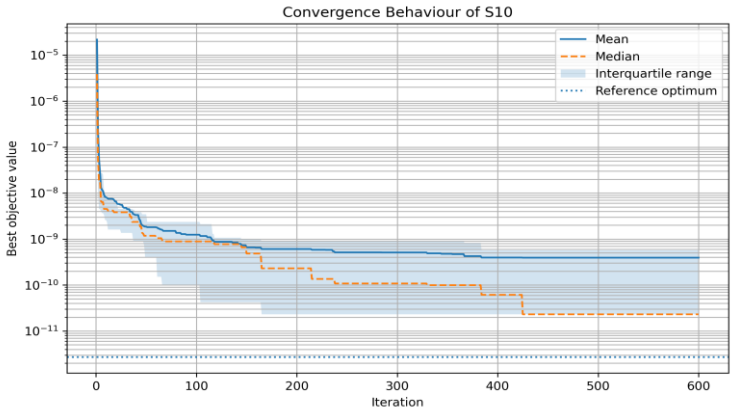

Figure 7: S10 mean, median, and interquartile convergence

The normalised convergence curves show rapid early improvement followed by discrete plateaus. Low-budget settings stabilise earlier and at higher values. S8 and S10 continue to improve as they progress through their longer budgets.

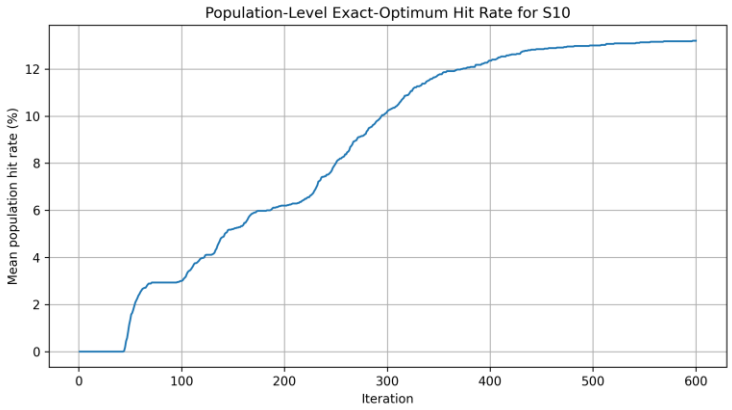

Figure 8: Mean population-level exact-optimum hit rate for S10

For S10, the median curve reaches the near-optimum region earlier than the mean. The interquartile band contracts as the run progresses, indicating increased agreement among runs, but it does not collapse completely because some runs remain at alternative near-optimal product pairs. Figure 8 illustrates the population-level exact-optimum hit rate for parameter configuration S10 over 600 iterations. In the initial iterations, the average population hit rate is close to 0%, and no solutions with an exact optimum are found. The hit rate quickly increases to around 3% at iteration 50, then slowly improves to around 6-7% between iterations 100 and 220. The increase is steadier from around iteration 230 to 350, with the hit rate going up to about 11.5%. After 400 iterations, the improvement slows, and the curve approaches a stable value of about 13% around iteration 600. Figure 8 shows that S10 gradually improves its ability to reach the exact optimum, most of the improvement being between iterations 200 and 400, after which it plateaus.

5.4. Rank-Based Statistical Comparison

The Friedman test rejected the null of equal performance: chi-square (9) = 120.972634, p = 8.447286e-22. Kendall W = 0.448047 indicates substantial agreement in the run-wise ordering (Table 6).

Table 6: Friedman omnibus test

Test	Number of sets	Paired runs	Chi-square	df.	p-value	Kendall W
Friedman	10	30	120.972634	9	8.447286e-22	0.448047

S10 ranked first, followed by S8, S9, S5, S7, S6, S4, S3, S2, and S1. After Holm adjustment, S10 was significantly better than S1, S2, and S3. Comparisons with S4-S9 were not significant at 0.05 (Figure 9).

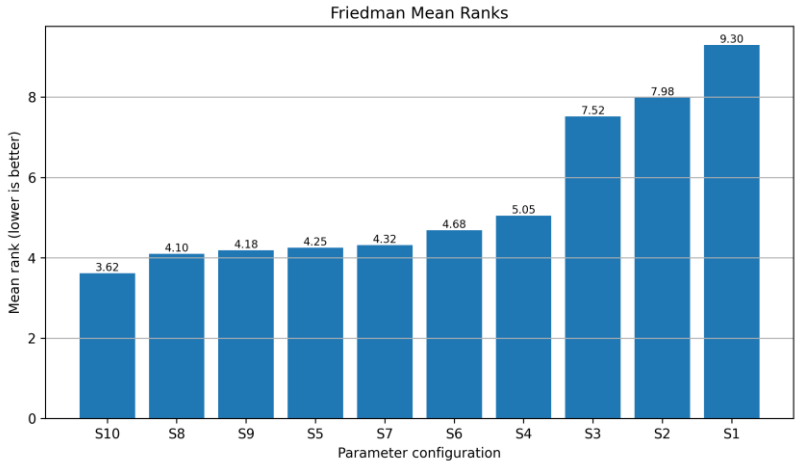

Figure 9: Friedman mean ranks; lower is better

The effect estimates against the three weakest sets were strongly negative because lower S10 objectives dominated the paired differences. The statistical conclusion is therefore that S10 is the best-ranked configuration and clearly improves on insufficient search budgets, while several larger configurations remain plausible alternatives (Table 7).

Table 7: Wilcoxon-Holm comparisons between S10 and the other configurations

Selected set	Comparison set	Wilcoxon W	Raw p	Holm p	Rank-biserial (selected-other)	Significant at 0.05
S10	S1	1.0	0.000002	0.000017	-0.996	True
S10	S3	10.5	0.000005	0.000040	-0.955	True
S10	S2	14.0	0.000007	0.000049	-0.940	True
S10	S4	80.5	0.015805	0.094829	-0.541	False
S10	S6	110.5	0.035150	0.175748	-0.456	False
S10	S7	102.0	0.103443	0.413774	-0.372	False
S10	S8	83.5	0.162292	0.486876	-0.340	False
S10	S5	116.0	0.331286	0.662571	-0.227	False
S10	S9	138.0	0.340434	0.662571	-0.214	False

5.5. Search-Stage Behaviour

Table 8 shows the percentage distribution across the different stages for the 10 SDOA configurations (S1–S10). D1 Global and D2 Moderate are the global and moderate search phases, and B1 Exploitation and D3 Fine-tune are the exploitation and fine-tuning activities. The Reset (%) column shows the percentage of reset operations. Overall, the configurations give a larger share to D3 Fine-tune, and S10 has the highest fine-tuning percentage (30.90%), while S1 has the highest D1 Global percentage (24.55%).

Table 8: Share of diver-iterations assigned to each SDOA stage

Set	D1 Global (%)	D2 Moderate (%)	B1 Exploitation (%)	D3 Fine-tune (%)	Reset (%)
S1	24.55	21.07	21.76	21.88	10.73
S2	16.51	18.04	23.71	25.61	16.13
S3	11.65	18.29	23.29	26.88	19.89
S4	9.19	17.08	20.76	29.27	23.71
S5	9.96	17.12	22.73	29.46	20.72
S6	8.16	14.66	21.18	29.33	26.66
S7	9.91	17.06	22.93	29.20	20.90
S8	11.08	17.91	22.78	30.63	17.60
S9	8.07	14.40	21.09	29.54	26.90
S10	14.24	18.89	22.21	30.90	13.76

S10 assigned 30.90% of diver-iterations to D3 fine-tuning, 22.21% to B1 exploitation, 18.89% to D2 moderate exploration, 14.24% to D1 global exploration, and 13.76% to reset (Figure 10).

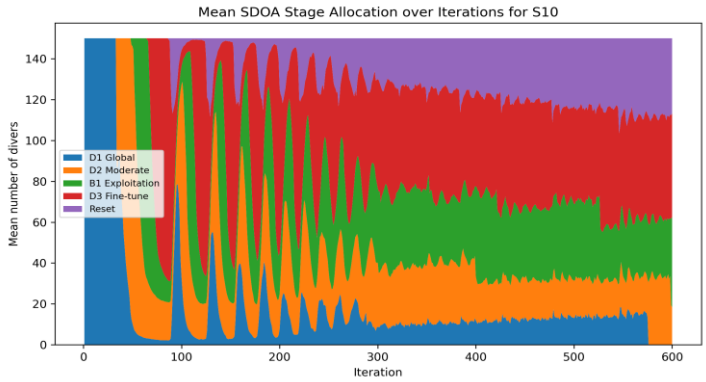

Figure 10: Mean stage allocation over S10 iterations

The reset share was lower than in most configurations with faster oxygen decay. The selected schedule consequently preserved a larger proportion of productive neighbourhood and refinement activity (Figure 11).

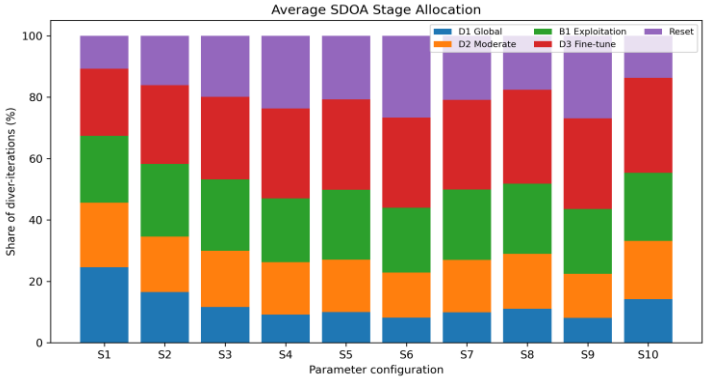

Figure 11: Average stage allocation across configurations

5.6. Computational Cost

S10 required a mean of 0.011545 s per run in the compiled experiment. S1 was fastest but least accurate (Figure 12).

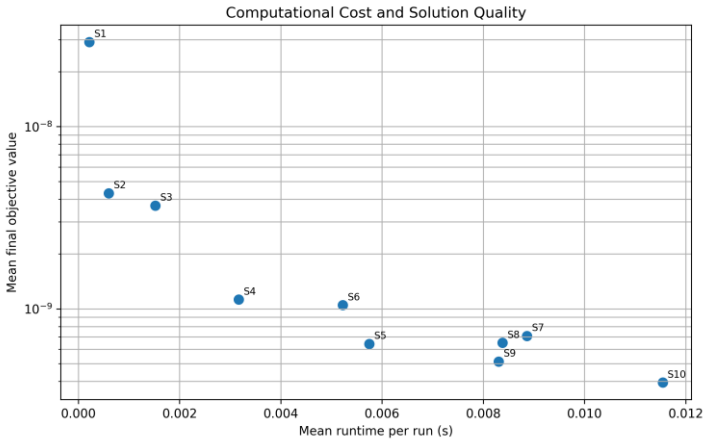

Figure 12: Mean runtime and mean final objective

S5 offered a lower-cost compromise, while S10 improved the mean and median at a larger population-iteration budget. Absolute times depend on the implementation and hardware and are not presented as MATLAB benchmarks.

5.7. Comparison with Previously Reported Results

Table 9 compares the present SDOA result with 24 previously reported algorithms. SDOA attains the verified discrete optimum, as do GeneAS, PSO-GA, MBA, CS, APSO, IAPSO, DE, DMPS, ALO, H-GA-FA, the nine Rao-family variants, and CL-CSA. Once this known optimum is reached, the best FN cannot by itself establish superiority. The relevant evidence from the present study is therefore the repeated-run distribution, accuracy success rates, convergence behaviour, and statistical rank under a stated evaluation budget. The comparison also preserves the benchmark's discrete character. Results with non-integer tooth counts, missing design variables, or objective values below the verified integer optimum were not included. Cross-study differences in mean performance cannot be inferred from this Table 9 because the cited studies used different population sizes, stopping criteria, function-evaluation budgets, random seeds, and numerical precision.

Table 9: Comparison of SDOA with 24 previously reported algorithms for the discrete gear-train design problem

#	Algorithm	TA	TB	TD	TF	FN	Ref.
1	Branch-and-Bound (BB)	18	22	45	60	5.712E-06	Sandgren [1]
2	Augmented Lagrange (AL)	13	15	33	41	2.146E-08	Kannan and Kramer [2]
3	Gene Adaptive Search (GeneAS)	19	16	49	43	2.701E-12	El-Shorbagy and El-Refaey [29]
4	Hybrid PSO-GA	19	16	43	49	2.70085E-12	El-Shorbagy and El-Refaey [29]
5	Mine Blast Algorithm (MBA)	19	16	43	49	2.70085E-12	El-Shorbagy and El-Refaey [29]
6	Cuckoo Search (CS)	19	16	43	49	2.701E-12	El-Shorbagy and El-Refaey [29]
7	Accelerated PSO (APSO)	19	16	43	49	2.700857E-12	El-Shorbagy and El-Refaey [29]
8	Improved APSO (IAPSO)	19	16	43	49	2.700857E-12	El-Shorbagy and El-Refaey [29]
9	Differential Evolution (DE)	19	16	43	49	2.700857E-12	El-Shorbagy and El-Refaey [29]
10	Discrete Mode-Pursuing Sampling (DMPS)	19	16	49	43	2.7009E-12	Sharif et al. [3]
11	Ant Lion Optimiser (ALO)	19	16	49	43	2.7009E-12	Mouhayen [30]

12	Original Firefly Algorithm (FA)	15	13	33	41	2.1469E-08	Mouhayen [30]
13	Original Genetic Algorithm (GA)	14	17	33	50	1.362E-09	Mouhayen [30]
14	Hybrid GA-FA (H-GA-FA)	19	16	49	43	2.7009E-12	Mouhayen [30]
15	Rao1	19	16	49	43	2.70086E-12	Ekker [23]
16	Rao2	19	16	49	43	2.70086E-12	Ekker [23]
17	Rao3	19	16	49	43	2.70086E-12	Ekker [23]
18	SAMPE Rao1	19	16	49	43	2.70086E-12	Ekker [23]
19	SAMPE Rao2	19	16	49	43	2.70086E-12	Ekker [23]
20	SAMPE Rao3	19	16	49	43	2.70086E-12	Ekker [23]
21	Chaotic Rao1	19	16	49	43	2.70086E-12	Ekker [23]
22	Chaotic Rao2	19	16	49	43	2.70086E-12	Ekker [23]
23	Chaotic Rao3	19	16	49	43	2.70086E-12	Ekker [23]
24	Comprehensive Learning CSA (CL-CSA)	16	19	43	49	2.7008571E-12	Mouhayen [30]
25	Scuba Diver Optimisation Algorithm (SDOA)	19	16	43	49	2.700857148886513E-12	Present study

Note: T_A, T_B, T_D, and T_F follow the variable order used in the present objective function. When a source used the equivalent ratio T_B T_D/(T_A T_F), the reported tooth counts were reordered without changing the physical gear combination or FN. FN values are reproduced at the precision reported by the cited studies. Symmetric exchanges T_A <-> T_B and T_D <-> T_F produce the same FN.

5.8. Discussion

5.8.1. Interpretation of the Selected Configuration

S10 combines a relatively large initial current strength with a moderate mutation rate, slow oxygen decay, 150 divers, and 600 iterations. The setting does not simply maximise every parameter. Slow oxygen decay reduces repeated resets and keeps more divers in D1-D3, while the long budget gives local one-tooth search and nonuniform mutation sufficient opportunities to test nearby product pairs. This interpretation is supported by the stage record rather than inferred only from the final objective. The median S10 result is considerably closer to the optimum than its mean. The skew is expected in a discrete-ratio problem: many runs converge to one of a few very accurate product pairs, while occasional runs remain on a less accurate plateau. Increasing the budget improves the probability of moving between plateaus but cannot produce smooth continuous convergence.

5.8.2. Importance of Integer Repair and Exact Verification

Integer repair is not a minor implementation detail. Without repair, crossover, Levy-like movement, mutation, and communication, fractional tooth counts result. Evaluating those values changes the feasible set and can produce an objective arbitrarily close to zero. Rounding only at the end is also inappropriate because the optimiser is guided throughout the run by values that cannot be manufactured. The repair-before-evaluation rule used here ensures that selection pressure is based on admissible designs. Exhaustive verification is possible for this benchmark and provides an unusually strong reference. It confirms both the optimum value and the existence of four symmetric optimum designs. The verification also prevents misinterpretation of numerical precision: the target ratio is not exactly attainable within the specified integer range. Hence, a reported zero indicates relaxation or insufficient output precision, not a superior discrete design.

5.8.3. Parameter Sensitivity and Statistical Meaning

The sensitivity experiment shows a transition from inadequate to sufficient search coverage. S1-S3 use small populations or limited iterations and rarely enter the highest-precision region. S4-S10 all reach the optimum, but their distributional performance remains different. This finding supports using mean, median, dispersion, ranks, and success thresholds together. The non-significant Holm-adjusted comparisons among several high-budget settings should not be ignored. A lower mean rank does not imply certain dominance. S10 is recommended because it gives the strongest combined numerical evidence, but S5 or S8 may be preferred under different cost constraints. The result is consistent with the no-free-lunch perspective: parameter choice is part of matching an algorithm to a problem and budget [8].

5.8.4. Engineering Interpretation

The benchmark represents a preliminary ratio-selection problem rather than a complete gearbox design. The optimum identifies tooth products that approximate the target ratio. Still, it does not address module, pressure angle, contact ratio, bending stress, pitting resistance, face width, shaft layout, centre distance, efficiency, noise, or manufacturability beyond integer tooth counts.

The selected combination should therefore be treated as an input to later mechanical verification, not as a final production gearbox.

6. Conclusion and Future Work

An integer-preserving Scuba Diver Optimisation Algorithm was developed for the compound gear-train ratio benchmark. Ten parameter configurations were evaluated through 300 independent runs. S10 achieved the lowest mean rank and mean objective, reached f <= 10^-9 in 86.67% of runs, and reproduced the verified global optimum [19, 16, 43, 49] with f = 2.700857148886513e-12. The Friedman test confirmed a configuration effect, and Holm-adjusted Wilcoxon tests showed that S10 significantly outperformed the three lowest-budget settings. The stage analysis indicates that the selected setting benefited from slow oxygen decay, sustained fine tuning, and limited reset activation. More importantly, the study demonstrates that integer repair and exact-optimum verification are necessary for valid conclusions. Values below the verified minimum obtained from fractional tooth counts do not solve the stated gear-train problem. Future work will extend SDOA to multi-stage and planetary gear systems with tooth interference, center-distance constraints, load-capacity constraints, efficiency constraints, and reliability constraints. Adaptive oxygen thresholds, learned operator selection, and hybrid exact-metaheuristic procedures should also be examined. For broader algorithm comparisons, equal function-evaluation budgets and shared reference implementations are recommended.

6.1. Limitations

The study has four limitations. First, it considers one low-dimensional discrete benchmark, so the findings do not establish general SDOA superiority. Second, the experiment compares parameter configurations of SDOA rather than reimplementing all algorithms under an identical function-evaluation budget. Third, the runtime measurements come from a compiled Python/Numba port and cannot be interpreted as MATLAB execution times. Fourth, the fixed-stage thresholds and operator probabilities were not included in the sensitivity analysis. Future work should test larger discrete gear systems, catalog-constrained tooth sets, multiobjective gearbox formulations, equal function-evaluation budgets, and adaptive stage thresholds.

Acknowledgment: The authors would like to express their sincere gratitude to Technical College of Informatics-Akre and Technical College of Duhok for their valuable support, encouragement, and academic assistance throughout this research.

Data Availability Statement: The MATLAB source code, parameter configurations, complete run-level results, summary statistics, convergence histories, and statistical-test outputs generated for this study are available in the accompanying supplementary package.

Funding Statement: This research received no external funding.

Conflicts of Interest Statement: The authors declare no conflicts of interest.

Ethics and Consent Statement: The study was conducted in accordance with established ethical guidelines, and participants were assured that their responses would remain confidential and anonymous**.**

References

1. E. Sandgren, “Nonlinear integer and discrete programming in mechanical design optimization,” *Journal of Mechanical Design*, vol. 112, no. 2, pp. 223–229, 1990.
2. B. K. Kannan and S. N. Kramer, “An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design,” *Journal of Mechanical Design*, vol. 116, no. 2, pp. 405–411, 1994.
3. B. Sharif, G. G. Wang, and T. Y. ElMekkawy, “Mode pursuing sampling method for discrete variable optimization on expensive black-box functions,” *Journal of Mechanical Design*, vol. 130, no. 2, p. 021402, 2008.
4. I. Boussaid, J. Lepagnot, and P. Siarry, “A survey on optimization metaheuristics,” *Information Sciences*, vol. 237, no. 7, pp. 82–117, 2013.
5. X. S. Yang, “Nature-inspired optimization algorithms: Challenges and open problems,” *Journal of Computational Science*, vol. 46, no. 10, p. 101104, 2020.
6. S. Biswas, A. Shaikh, A. E. S. Ezugwu, J. Greeff, S. Mirjalili, U. K. Bera, and L. Abualigah, “Enhanced prairie dog optimization with Levy flight and dynamic opposition-based learning for global optimization and engineering design problems,” *Neural Computing and Applications*, vol. 36, no. 3, pp. 11137–11170, 2024.

7. W. Liu, W. Yan, T. Li, G. Han, and T. Ren, "A multi-strategy improved Grasshopper Optimization Algorithm for solving global optimization and engineering problems," *International Journal of Computational Intelligence Systems*, vol. 17, no. 7, pp. 1–28, 2024.
8. E. H. Houssein, M. K. Saeed, G. Hu, and M. M. Al-Sayed, "An efficient improved exponential distribution optimizer: Application to the global, engineering and combinatorial optimization problems," *Cluster Computing*, vol. 27, no. 4, pp. 9345–9380, 2024.
9. S. Biswas, B. Maiti, G. Singh, A. E. Ezugwu, K. Saleem, L. Abualigah, A. Smerat, and U. K. Bera, "A novel hybrid optimizer based on Coati Optimization Algorithm and Differential Evolution for global optimization and constrained engineering problems," *International Journal of Computational Intelligence Systems*, vol. 18, no. 6, pp. 1–70, 2025.
10. R. Storn and K. Price, "Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 12, pp. 341–359, 1997.
11. S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, no. 3, pp. 46–61, 2014.
12. S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, no. 5, pp. 51–67, 2016.
13. S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, no. 3, pp. 120–133, 2016.
14. S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, no. 12, pp. 163–191, 2017.
15. S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, no. 11, pp. 228–249, 2015.
16. S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-Verse Optimizer: A nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, no. 3, pp. 495–513, 2016.
17. A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Generation Computer Systems*, vol. 97, no. 8, pp. 849–872, 2019.
18. X. S. Yang, C. Huyck, M. Karamanoglu, and N. Khan, "True global optimality of the pressure vessel design problem: A benchmark for bio-inspired optimisation algorithms," *International Journal of Bio-Inspired Computation*, vol. 5, no. 6, pp. 329–335, 2013.
19. S. Wang, L. Cao, Y. Chen, C. Chen, Y. Yue, and W. Zhu, "Gorilla optimization algorithm combining sine cosine and Cauchy variations and its engineering applications," *Scientific Reports*, vol. 14, no. 3, pp. 1–20, 2024.
20. J. Sui, Z. Tian, and Z. Wang, "Multiple strategies improved spider wasp optimization for engineering optimization problem solving," *Scientific Reports*, vol. 14, no. 11, pp. 1–29, 2024.
21. R. V. Rao and N. Zinzuvadia, "Design optimization of selected mechanical engineering components using variants of Rao algorithms," *Jordan Journal of Mechanical and Industrial Engineering*, vol. 16, no. 5, pp. 835–863, 2022.
22. M. Ghasemi, S. K. Mohammadi, M. Zare, S. Mirjalili, M. Gil, and R. Hemmati, "A new firefly algorithm with improved global exploration and convergence with application to engineering optimization," *Decision Analytics Journal*, vol. 5, no. 12, pp. 1–18, 2022.
23. E. Eker, "Performance Evaluation of Capuchin Search Algorithm through Non-linear Problems, and Optimization of Gear Train Design Problem," *European Journal of Technique (EJT)*, vol. 13, no. 2, pp. 142–149, 2023.
24. J. Derrac, S. Garcia, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.
25. M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.
26. F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.
27. S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979.
28. M. Mendez, D. A. Rossit, B. Gonzalez, and M. Frutos, "Proposal and comparative study of evolutionary algorithms for optimum design of a gear system," *IEEE Access*, vol. 8, no. 12, pp. 3482–3497, 2020.
29. M. A. El-Shorbagy and A. M. El-Refaey, "A hybrid genetic-firefly algorithm for engineering design problems," *Journal of Computational Design and Engineering*, vol. 9, no. 2, pp. 706–730, 2022.
30. K. Mouhayen, S. El Moumen, and H. Lakhbab, "Comprehensive Learning Crow Search Algorithm for Global Optimization," *Mathematical Modeling and Computing*, vol. 13, no. 1, pp. 136–146, 2026.