FMDB *Transactions on Sustainable Structures and Materials*

Continuous U-Turning Ant Colony Optimization for Constrained Speed Reducer Design

Saman M. Almufti[1,*], Renas Rajab Asaad[2], Bhopendra Singh[3], Dinka Lale[4], Yuri Ryagin[5], Rahul Panakkal[6]

[1,2]Department of Information Technology, Technical College of Informatics-Akre, Akre University for Applied Sciences, Akre, Nineveh Governorate, Iraq.
[3]Department of Engineering, Amity University Dubai, Dubai, United Arab Emirates.
[4]Faculty of Electrical Engineering and Applied Computing, University of Dubrovnik, Dubrovnik, Croatia.
[5]Department of Low Temperature Physics and Nanoscale Systems, Ural Federal University, Yekaterinburg, Sverdlovsk Oblast, Russia.
[6]Department of Computer Science and Engineering, University of Illinois at Urbana, Champaign, Illinois, United States of America.
saman.almofty@gmail.com[1], renas.beda89@gmail.com[2], bhopendrasingh@yahoo.com[3], lariat@unidu.hr[4], yuri.ryagin@urfu.ru[5], rahulp8@illinois.edu[6]

Abstract: The speed reducer benchmark is a mixed-integer nonlinear engineering design problem in which gear geometry, bearing spans, and shaft diameters jointly determine structural weight and eleven stress, deflection, and geometric constraints. This paper develops a continuous U-Turning Ant Colony Optimization (U-TACO) formulation that transfers partial-decision pheromone learning from routing problems to a seven-variable constrained mechanical design. A U-ant samples only a subset of coordinates; the remaining coordinates are completed, the full reducer is decoded and evaluated, and pheromone credit is assigned only to the sampled subset. Complete ants subsequently construct full designs using the same decoder and feasibility-first comparison rule. Eight parameter configurations were evaluated with 50 matched random seeds, while nested samples of 10, 30, and 50 runs were retained for stability analysis. The experiment comprised 400 distinct runs. The precision-search configuration S8 obtained a best feasible weight of 2994.471066, a mean of 2994.471067, a standard deviation of 6.4123×10^{-7}, and 100% final feasibility. Friedman tests detected configuration effects at each run count; for 50 runs, $\chi^2=259.793658$, $p=2.277\times10^{-52}$, and Kendall's $W=0.742268$. Holm-adjusted paired tests favoured S8 over each alternative. The best solution was [3.5, 0.7, 17, 7.3, 7.715320, 3.350215, 5.286654], with four constraints active or numerically near active. The results show that partial-decision reinforcement is effective when combined with exact mixed-integer decoding, constraint-aware completion, and local refinement. Reported runtimes are Python/Numba validation times and are not MATLAB benchmarks.

Keywords: Speed Reducer Design; Gear Geometry; Bearing Spans; Mixed-Integer Decoding; Geometric Constraints; Local Refinement; Comparison Rule; Pheromone Credit.

Received on: 04/07/2025, **Revised on:** 19/10/2025, **Accepted on:** 30/10/2025, **Published on:** 07/10/2026

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

DOI: https://doi.org/10.69888/FTSSM.2026.000814

Cite as: S. M. Almufti, R. R. Asaad, B. Singh, D. Lale, Y. Ryagin, and R. Panakkal, "Continuous U-Turning Ant Colony Optimization for Constrained Speed Reducer Design," *FMDB Transactions on Sustainable Structures and Materials,* vol. 2, no. 2, pp. 70–87, 2026.

Copyright © 2026 S. M. Almufti *et al.,* licensed to Fernando Martins De Bulhão (FMDB) Publishing Company. This is an open access article distributed under CC BY-NC-SA 4.0, which permits copying, remixing, redistributing, transformation, and building upon the material in any medium so long as the original work is properly cited.

[*]Corresponding author.

1. Introduction

A speed reducer transfers rotary motion between an input and an output shaft while reducing speed and increasing torque. Its mass is governed by the coupled selection of gear dimensions, bearing spans, and shaft diameters. Reducing one component can increase bending stress, surface stress, shaft deflection, or geometric incompatibility. The classical benchmark therefore uses seven variables and eleven nonlinear inequalities, with the number of pinion teeth treated as an integer [1]; [3]. The benchmark is frequently used to test constrained optimizers because its feasible region is narrow near the best-known solution. Several constraints become active simultaneously, and small differences in bounds, integer handling, numerical precision, or constraint equations can alter the reported objective. Direct numerical comparisons are meaningful only when the complete formulation is identical. Ant Colony Optimization (ACO) was developed as a constructive pheromone-learning method for discrete problems [4].

Continuous ACO methods replace discrete edge choices with distributions over real-valued coordinates [5]. U-TACO introduces a separate partial-agent mechanism: a U-ant constructs only part of a candidate and returns feedback before complete ants construct final solutions [6]; [7]. In the original routing interpretation, U-ants guide the pheromone matrix but are not responsible for the final tour. The present work preserves this credit-assignment principle rather than treating a partial vector as a complete mechanical design. A direct transfer to speed reducer design requires three modifications. First, pheromone must be represented over bounded continuous intervals, while the tooth count remains an integer. Second, every partial vector must be completed before its quality is known. Third, candidate comparison must account for feasibility without allowing a small weight to compensate for a violated stress or geometry constraint [2]. The proposed formulation addresses these requirements with interval pheromone, model-based completion, a feasibility-first rule, and local refinement.

The main contributions are:

- A continuous mixed-integer U-TACO formulation is derived in which a partial coordinate set receives credit only after a complete reducer design has been evaluated.
- A model-based decoder uses the algebraic structure of the face-width and shaft constraints to reduce avoidable infeasibility while retaining explicit evaluation of all eleven residuals.
- Eight parameter configurations are evaluated with 50 matched seeds, and the same results are examined as nested samples of 10, 30, and 50 runs.
- Accuracy, feasibility, evaluations, runtime, convergence, diversity, pheromone entropy, constraint histories, and nonparametric paired tests are reported.

2. Related Work

2.1. Speed Reducer Optimization

Golinski [1] introduced nonlinear programming formulations for machine synthesis and the speed reducer problem. The reducer model was later examined using deterministic global techniques, harmony search, differential evolution, particle swarm optimization, bat algorithms, grey wolf optimization, whale optimization, and many newer population methods [10]; [12]. The deterministic study of Lin et al. [3] expressed the model as a generalized geometric program. It used convexification and piecewise linearization to obtain a globally certified approximation under its stated formulation. Metaheuristic results reported for this benchmark are not always interchangeable. Some studies treat the tooth count as a continuous variable and round it only after the search; others enforce integrality at every evaluation. Bounds for the second shaft span and numerical coefficients also differ in parts of the literature. For this reason, the present study fixes one model and uses a reference value only to quantify numerical agreement within that model.

2.2. Continuous ACO and U-Turn Learning

ACO uses repeated construction, pheromone reinforcement, and evaporation to increase the probability of decisions associated with good solutions [4]. Socha and Dorigo [5] showed that the same framework can be extended to continuous variables by sampling from distributions rather than selecting discrete graph components. The current interval representation provides a direct way to preserve dimension-specific pheromone and to identify which coordinates were selected by a U-ant.

The U-TACO mechanism was first described for symmetric travelling-salesman instances [6]; [7]. U-ants form partial tours and update the corresponding pheromone components; ordinary ants subsequently construct complete tours. The transfer used here retains the separation between guidance and complete-solution construction. A partial coordinate vector is never ranked as a reducer design [18]. It is completed, decoded, and evaluated before the original coordinate subset receives a deposit.

2.3. Experimental Gap

Many engineering-optimization papers report only the best design and one convergence curve [11]. Such reporting does not show whether the result is repeatable, whether final solutions are feasible, how parameter choices alter performance, or whether a small numerical difference persists across matched random seeds. The experimental design in this paper therefore combines scenario sensitivity, complete residual reporting, nested run counts, and paired nonparametric tests [13]; [17].

2.4. Speed Reducer Design Model

Figure 1 shows the adopted two-stage reducer representation (Table 1).

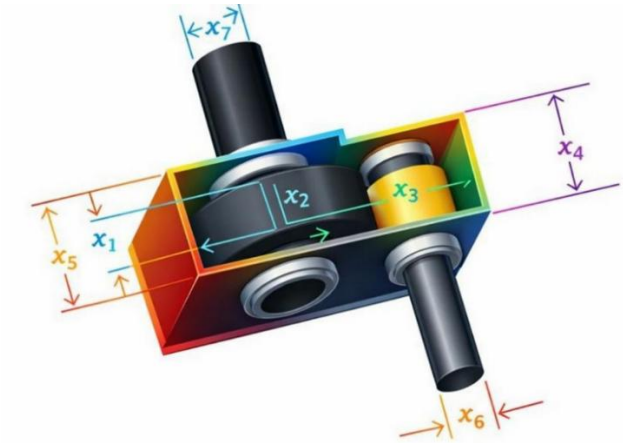


Figure 1: Speed reducer geometry and the seven design variables

The decision vector is:

$$\mathbf{x} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7]^T = [b, m, z, l_1, l_2, d_1, d_2]^T \quad (1)$$

Table 1: Decision variables and search bounds

Variable	Symbol	Physical Meaning	Bounds	Type
x_1	b	Face width	2.6-3.6	Continuous
x_2	m	Module of teeth	0.7-0.8	Continuous
x_3	z	Number of pinion teeth	17-28	Integer
x_4	l_1	First shaft bearing span	7.3-8.3	Continuous
x_5	l_2	Second shaft bearing span	7.3-8.3	Continuous
x_6	d_1	First shaft diameter	2.9-3.9	Continuous
x_7	d_2	Second shaft diameter	5.0-5.5	Continuous

2.5. Objective Function

The objective is the conventional surrogate for the total reducer weight:

$$f(\mathbf{x}) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \quad (2)$$

The first term represents the gear contribution, the second and third terms account for shaft-related corrections, and the last term includes the bearing-span contributions [14]. The objective is minimized without an additive constraint penalty [15].

2.6. Constraints

The first two constraints limit gear-tooth bending and surface loading:

$$g_1(\mathbf{x}) = \frac{27}{x_1 x_2^2 x_3} - 1 \le 0$$

$$g_2(\mathbf{x}) = \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \le 0 \quad (3)$$

The next two inequalities limit transverse deflection of the two shafts:

$$g_3(\mathbf{x}) = \frac{1.93 x_4^3}{x_2 x_3 x_6^4} - 1 \le 0$$

$$g_4(\mathbf{x}) = \frac{1.93 x_5^3}{x_2 x_3 x_7^4} - 1 \le 0 \quad (4)$$

The shaft-stress constraints are:

$$g_5(\mathbf{x}) = \frac{\sqrt{\left(\frac{745 x_4}{x_2 x_3}\right)^2 + 16.9 \times 10^6}}{110 x_6^3} - 1 \le 0$$

$$g_6(\mathbf{x}) = \frac{\sqrt{\left(\frac{745 x_5}{x_2 x_3}\right)^2 + 157.5 \times 10^6}}{85 x_7^3} - 1 \le 0 \quad (5)$$

The remaining geometric and dimensional conditions are:

$$g_7(\mathbf{x}) = \frac{x_2 x_3}{40} - 1 \le 0$$

$$g_8(\mathbf{x}) = \frac{5 x_2}{x_1} - 1 \le 0 \quad (6)$$

$$g_9(\mathbf{x}) = \frac{x_1}{12 x_2} - 1 \le 0$$

$$g_{11}(\mathbf{x}) = \frac{1.1 x_7 + 1.9}{x_5} - 1 \le 0 \quad (7)$$

All residuals follow the convention $g_i(x) \le 0$. A negative residual indicates a feasible margin; a value close to zero denotes an active or numerically near-active constraint [16].

2.7. Bounds and Mixed-Integer Condition

$$2.6 \le x_1 \le 3.6, \quad 0.7 \le x_2 \le 0.8, \quad x_3 \in \{17, \ldots, 28\}$$

$$7.3 \le x_4, x_5 \le 8.3, \quad 2.9 \le x_6 \le 3.9, \quad 5.0 \le x_7 \le 5.5 \quad (8)$$

The pinion-tooth variable is rounded and clipped during every decoding operation. It is therefore an integer throughout the search rather than being rounded only after optimization.

2.8. Feasibility Measure

A normalized total violation is used only to compare infeasible candidates:

$$\mathrm{CV}(\mathbf{x}) = \sum_{i=1}^{11} \max\left(0, \frac{g_i(\mathbf{x})}{s_i}\right) + \mathrm{CV}_{\mathrm{bounds}} + \mathrm{CV}_{\mathrm{domain}} \quad (9)$$

The scales s_i prevent residuals of different numerical magnitudes from dominating the infeasible comparison. Feasible candidates always dominate infeasible candidates; among feasible candidates, the lower weight is preferred [8]; [9].

3. Continuous U-TACO Method

Figure 2 shows the iterative U-TACO optimization process for restricted speed-reducer design. Phase 1: Sampling and evaluation of specified design coordinates under U-turn partial-decision guidance. Phase 2 is a constraint evaluation and a feasibility-first local refinement and a full-colony search. Finally, pheromone updating and convergence checking are repeated to find the optimum viable speed-reducer design.

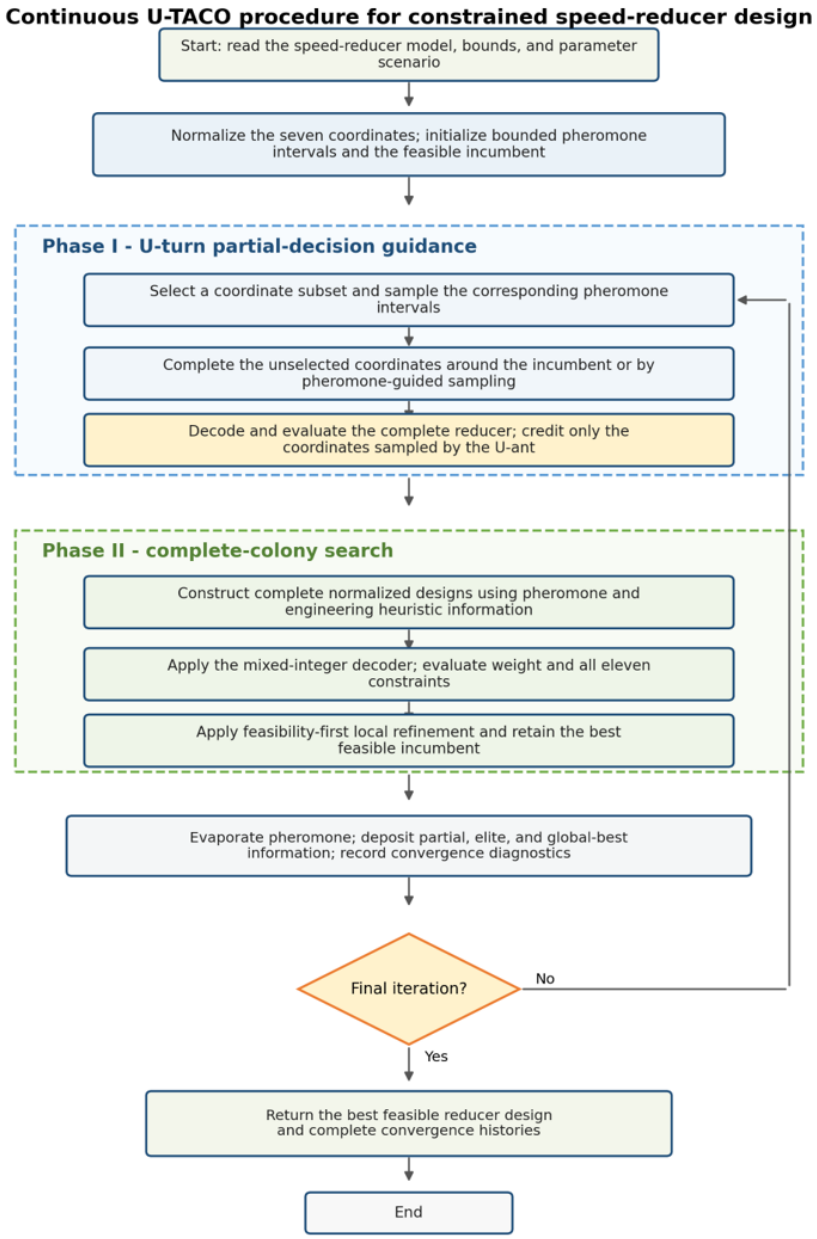


Figure 2: Continuous U-TACO procedure for constrained speed reducer design

3.1. Normalized Interval Representation

Each continuous coordinate is mapped to the unit interval. For coordinate j with lower bound l_j and upper bound $u_j$max:

$$v_j = \frac{x_j - l_j}{u_j - l_j}, \qquad 0 \le v_j \le 1 \tag{10}$$

The unit interval is divided into B bins. A separate pheromone vector is maintained for each dimension. The integer coordinate x_3 is sampled in normalized form and mapped to the nearest admissible tooth count during decoding.

3.2. Bin Selection and Engineering Heuristic

The probability of selecting bin b in dimension j is:

$$p_{jb} = \frac{\tau_{jb}^{\alpha} \eta_{jb}^{\beta}}{\sum_{q=1}^{B} \tau_{jq}^{\alpha} \eta_{jq}^{\beta}} \tag{11}$$

Where α controls pheromone influence and β controls heuristic influence; the bin heuristic is defined relative to the current best normalized coordinate:

$$\eta_{jb} = \frac{1}{\varepsilon + \left(c_{jb} - v_j^{\text{best}}\right)^2} \quad - \tag{12}$$

This heuristic increases local sampling pressure without eliminating nonlocal intervals. Scenario-dependent sampling widths and evaporation prevent the same exploitation level from being imposed on all configurations.

3.3. U-Ant Completion and Partial Credit

A U-ant selects a coordinate subset S and samples only those intervals. The unselected coordinates are completed around the incumbent or by pheromone-guided sampling:

$$\tilde{\mathbf{v}}_S = \mathbf{v}_S^U, \qquad \tilde{\mathbf{v}}_{\bar{S}} = \mathcal{C}\left(\mathbf{v}^{\text{best}}, \boldsymbol{\tau}\right) \tag{13}$$

The completed vector is decoded, repaired, locally refined, and evaluated using the full objective and all eleven constraints. Only the dimensions in S receive the U-ant deposit:

$$\Delta\tau_{jb}^{(U)} = I_{jb}^{U} \frac{q_U}{1 + \hat{f}(\tilde{\mathbf{v}}) + \lambda \mathrm{CV}(\tilde{\mathbf{v}})} \tag{14}$$

The indicator $I_{j\beta}^{U}$ equals one only when dimension j belongs to the U-ant subset and bin b was sampled; otherwise, it is zero. This rule preserves the original U-TACO distinction between partial guidance and final-solution construction [7].

3.4. Complete-Colony Reinforcement

Complete ants sample all seven coordinates. Elite and global-best candidates contribute complete deposits:

$$\Delta\tau_{jb}^{(C)} = \frac{q_E I_{jb}^{\text{elite}} + q_G I_{jb}^{\text{global}}}{1 + \hat{f} + \lambda \mathrm{CV}} \tag{15}$$

Pheromone is then evaporated and bounded:

$$\tau_{jb} \leftarrow \min\left(\tau_{\max}, \max\left(\tau_{\min}, (1-\rho)\tau_{jb} + \Delta\tau_{jb}^{(U)} + \Delta\tau_{jb}^{(C)}\right)\right) \tag{16}$$

3.5. Model-Based Decoder and Refinement

The decoder first enforces bounds and tooth-count integrality. It then uses the monotone inequalities g_8 and g_9 to identify a feasible interval for the face width. Lower bounds for d_1 and d_2 are derived from the corresponding deflection and stress constraints, while upper geometric limits are retained. If these intervals are empty, the candidate remains infeasible and is compared through total violation. All eleven original residuals are still evaluated after decoding. Local refinement tests coordinate and paired-coordinate moves in normalized space. A move is retained only when it is better under the feasibility-first rule. Step sizes contract when no accepted move is found. A deeper global refinement is applied periodically to the incumbent according to the scenario settings.

3.6. Search Diagnostics

Population diversity and normalized pheromone entropy are recorded at each iteration:

$$D_t = \frac{1}{7}\sum_{j=1}^{7} \mathrm{sd}\left(v_{1j}^{(t)}, \ldots, v_{Mj}^{(t)}\right) \quad (17)$$

$$H_t = -\frac{1}{7\log B}\sum_{j=1}^{7}\sum_{b=1}^{B} p_{jb}^{(t)} \log p_{jb}^{(t)} \quad (18)$$

Diversity measures the spread of complete-ant coordinates; entropy measures how evenly pheromone mass is distributed over bins.

3.7. Experimental Design

Table 2. Colony, pheromone and U-turn parameters for the eight experimental configurations (S1-S8). Each setup alters the number of ants, U-ants, iterations, pheromone bins and control parameters. These settings correspond to conservative, balanced, exploratory, exploitative and specialized search techniques. The parameter variations allow a systematic assessment of optimization performance, exploration, exploitation and convergence efficiency.

Table 2: Colony, pheromone, and U-turn parameters

Set	Description	Ants	U-Ants	Iter.	Bins	α	β	ρ	Partial
S1	Conservative	18	8	70	50	1.00	1.50	0.15	0.50
S2	Balanced	24	12	90	70	1.00	2.00	0.20	0.50
S3	Exploratory	28	16	100	80	0.80	1.20	0.30	0.67
S4	Exploitative	24	12	100	100	1.50	3.00	0.12	0.50
S5	Rapid evaporation	24	12	90	70	1.00	2.00	0.45	0.50
S6	Strong U-turn	24	20	90	70	1.00	2.00	0.20	0.67
S7	Large colony	40	20	90	80	1.00	2.00	0.20	0.50
S8	Precision local search	20	10	120	120	1.30	2.50	0.15	0.50

Table 3 shows the completion and local-refinement parameters for eight experimental configurations (S1–S8). Each configuration affects the heuristic strength, the completion-best weighting, the local elite selection, the refinement passes and the step-sizes. The contraction factors and global search intervals balance exploration and exploitation. These parameter adjustments allow a thorough evaluation of search precision, convergence efficiency and optimization performance.

Table 3: Completion and local-refinement parameters

Set	Heuristic	Completion Best	Local Elite	Local Passes	Initial Step	Contract	Global Interval	Global Passes	Global Step
S1	3.0	0.60	2	4	0.080	0.60	15	4	0.040
S2	4.0	0.65	3	6	0.060	0.55	10	8	0.025
S3	2.5	0.45	3	5	0.100	0.60	15	6	0.050
S4	5.5	0.80	4	10	0.040	0.50	5	10	0.020
S5	4.0	0.65	3	6	0.060	0.55	10	8	0.025
S6	4.0	0.60	3	7	0.055	0.55	10	8	0.025
S7	4.0	0.65	4	6	0.055	0.55	10	8	0.025
S8	6.0	0.85	5	14	0.030	0.45	5	15	0.015

3.7.1. Matched Repeated-Run Protocol

Each scenario was executed with seeds 1-50. The same seeds were used across all eight scenarios, producing 400 distinct optimization runs. The 10-run sample contains seeds 1-10, the 30-run sample contains seeds 1-30, and the 50-run sample contains seeds 1-50. These samples are nested; the algorithm was not redundantly re-executed for each sample size. The

canonical reference used for numerical agreement was 2994.47094338914. Absolute relative percentage deviation was computed as:

$$\mathrm{RPD}(\%) = 100\left|\frac{f - f_{\mathrm{ref}}}{f_{\mathrm{ref}}}\right| \tag{19}$$

Strict success was defined before comparing scenarios:

$$\mathrm{CV} \le 10^{-8} \quad \wedge \quad \mathrm{RPD} \le 10^{-4}\% \tag{20}$$

3.7.2. Statistical Analysis

Friedman tests compared the eight scenarios under matched seeds. Kendall's W quantified the strength of the scenario effect. After a significant omnibus result, the best-ranked S8 configuration was compared with each alternative using paired Wilcoxon signed-rank tests. Holm's step-down adjustment controlled the family-wise error rate [13]; [17]. Win-tie-loss counts and rank-biserial effects were reported with the adjusted probabilities.

3.7.3. Execution Environment

The numerical results were generated with a Python validation implementation using NumPy, SciPy, and Numba. The MATLAB package supplied with the study follows the same mathematical model, scenario definitions, seed numbering, constraint handling, and output structure. Because MATLAB was not available in the execution environment, the measured times in this paper are Python validation runtimes and must not be interpreted as MATLAB timing benchmarks. Function-evaluation counts provide a platform-independent measure of search effort.

4. Results and Discussions

4.1. Parameter Sensitivity and Final Accuracy

Table 4 shows that all scenarios returned feasible final designs in all 50 runs. S8 gave the smallest mean error and the tightest dispersion: its mean weight was 2994.471067, and its standard deviation was 6.4123×10^{-7}. S4 was the closest lower-cost alternative, with a mean of 2994.471086. S1 reached strict success in 44% of runs; every other configuration reached 100%. The difference is therefore mainly one of precision and search budget rather than final feasibility.

Table 4: Objective accuracy, success, and final feasibility for 50 runs

Set	Best	Mean	Std.	Mean Abs. RPD (%)	Success (%)	Feasible (%)
S1	2994.471096	2994.498235	4.274e-02	0.000911	44.0	100.0
S2	2994.471068	2994.471405	3.469e-04	0.000015	100.0	100.0
S3	2994.471070	2994.471945	6.028e-04	0.000033	100.0	100.0
S4	2994.471067	2994.471086	1.666e-05	0.000005	100.0	100.0
S5	2994.471068	2994.471423	3.839e-04	0.000016	100.0	100.0
S6	2994.471066	2994.471394	3.064e-04	0.000015	100.0	100.0
S7	2994.471067	2994.471270	1.545e-04	0.000011	100.0	100.0
S8	2994.471066	2994.471067	6.412e-07	0.000004	100.0	100.0

Computational effort, population diagnostics and composite ranks for the eight configurations (S1–S8) are reported in Table 5. The results are a comparison of execution time, function evaluations, convergence speed, feasibility rates and population diversity. The fastest convergence in median iterations is achieved for configuration S8 (2.0), whereas configuration S1 has the least computational time (0.015690 s). S4 and S8 have the best composite rank of 1.5, which means they are robust in overall optimization performance.

Table 5: Computational effort, population diagnostics, and composite ranks

Set	Time (s)	Evaluations	Median Iter. to Target	Raw Feas. (%)	Post Feas. (%)	Diversity	Composite Rank
S1	0.015690	10,328	32.0	70.3	70.3	0.0069	8.0
S2	0.037466	27,657	11.5	93.0	93.0	0.0044	3.5

S3	0.057673	26,710	29.0	82.2	82.2	0.0071	7.0
S4	0.059962	63,320	4.0	45.8	45.8	0.0031	1.5
S5	0.037620	27,657	15.0	93.0	93.0	0.0044	6.0
S6	0.049737	32,207	10.0	93.0	93.0	0.0047	3.5
S7	0.073877	37,617	10.0	77.0	77.0	0.0039	5.0
S8	0.080556	125,533	2.0	18.6	26.1	0.0029	1.5

Table 6 shows the results of the Friedman test for nested samples of 10, 30 and 50 runs for eight configurations. The results demonstrate statistically significant differences in performance with very low p-values for all sample sizes. Kendall's W values between 0.742268 and 0.786538 show excellent agreement in the configuration rankings. Configuration S8 always obtains the best rank, exhibiting better and statistically robust optimization performance.

Table 6: Friedman tests for the nested run samples

Runs	Statistic	df.	p-Value	Kendall's W	Best-Ranked
10	55.057692	7	1.452e-09	0.786538	S8
30	160.708333	7	2.276e-31	0.765278	S8
50	259.793658	7	2.277e-52	0.742268	S8

Table 7. Paired Wilcoxon-Holm comparisons of setup S8 against 7 other configurations (50 runs). Results show statistically significant differences in all comparisons with very low adjusted p-values. Rank-biserial correlation values were between 0.990588 and 1.000000, indicating very significant impact sizes. These results demonstrate that S8 is consistently and statistically significantly better than the other options.

Table 7: Paired Wilcoxon-Holm comparisons against S8 for 50 runs

Comparison	W-T-L	W Statistic	Raw p	Holm p	Rank-Biserial	Significant
S8 vs S1	50-0-0	0.0	1.776e-15	1.243e-14	1.000000	Yes
S8 vs S2	50-0-0	0.0	1.776e-15	1.243e-14	1.000000	Yes
S8 vs S3	50-0-0	0.0	1.776e-15	1.243e-14	1.000000	Yes
S8 vs S4	49-0-1	6.0	2.487e-14	2.487e-14	0.990588	Yes
S8 vs S5	50-0-0	0.0	1.776e-15	1.243e-14	1.000000	Yes
S8 vs S6	49-0-1	2.0	5.329e-15	1.243e-14	0.996863	Yes
S8 vs S7	50-0-0	0.0	1.776e-15	1.243e-14	1.000000	Yes

Table 8 shows the best practicable design of the speed-reducer obtained with configuration S8 in 50 trial runs. The optimum design contains seven design factors such as face width, module, pinion teeth, shaft spans and shaft diameters. The minimum goal weight of 2994.471066155764 is attained with an almost zero normalized constraint violation of 4.441e-16. The results indicate high optimization accuracy and feasibility, requiring 125,632 function evaluations and a validation runtime of 0.080305 seconds.

Table 8: Best feasible S8 design from the 50-run experiment

Quantity	Value
Face width, x_1	3.500000000000
Module, x_2	0.700000000000
Pinion teeth, x_3	17
First shaft span, x_4	7.300000000000
Second shaft span, x_5	7.715319911882
First shaft diameter, x_6	3.350214666096
Second shaft diameter, x_7	5.286654464980
Objective weight	2994.471066155764
Total normalized violation	4.441e-16
Function evaluations	125,632
Python validation runtime	0.080305 s

Table 9 gives the residual values of eleven constraints for the optimal S8 speed-reducer design. The findings show that all constraints satisfy the feasibility criteria, which proves the validity of the optimized solution. The constraints g5, g6, g8 and g11 are active or near-active; the other constraints are dormant. Such results indicate that S8 is effective in obtaining a feasible and optimal design with negligible constraint violations.

Table 9: Residuals of the eleven constraints for the best S8 design

Constraint	Residual $g_i(x)$	Status	Feasibility
g1	-7.391528e-02	Inactive	Feasible
g2	-1.979985e-01	Inactive	Feasible
g3	-4.991722e-01	Inactive	Feasible
g4	-9.046439e-01	Inactive	Feasible
g5	2.220446e-16	Active/Near-Active	Feasible
g6	2.220446e-16	Active/Near-Active	Feasible
g7	-7.025000e-01	Inactive	Feasible
g8	0.000000e+00	Active/Near-Active	Feasible
g9	-5.833333e-01	Inactive	Feasible
g10	-5.132575e-02	Inactive	Feasible
g11	-5.228074e-11	Active/Near-Active	Feasible

Table 10 displays the stability results of configuration S8 in nested tests with 10, 30 and 50 runs. The results show consistent best and mean objective values with very low standard deviations between 5.952e-07 and 6.412e-07. All trials obtain 100% success and feasibility rates, showing reliable optimization performance. The practically constant average execution time and the small objective deviations illustrate the computing efficiency, robustness and stability of S8.

Table 10: Nested 10-, 30-, and 50-run stability results for S8

Runs	Best	Mean	Std.	Mean RPD (%)	Success (%)	Feasible (%)	Mean Time	Total Time
10	2994.471066	2994.471067	5.952e-07	0.000004	100.0	100.0	0.081577	0.816
30	2994.471066	2994.471067	6.355e-07	0.000004	100.0	100.0	0.081129	2.434
50	2994.471066	2994.471067	6.412e-07	0.000004	100.0	100.0	0.080556	4.028

Figure 3 shows that the high-accuracy settings enter the reference region rapidly. S8 continues refining numerical digits after reaching the target, whereas S1 stabilizes at a visibly larger error.

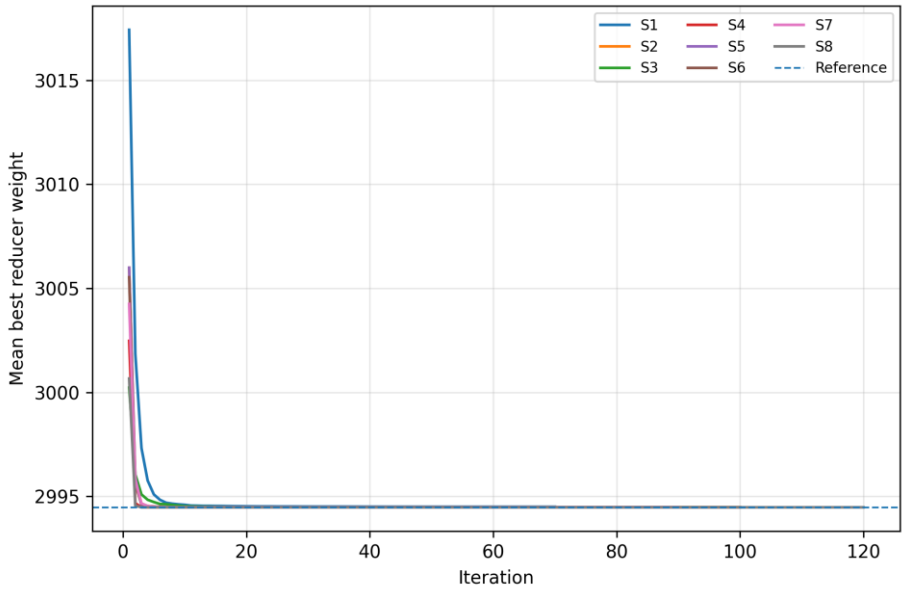


Figure 3: Mean best-so-far weight across 50 runs for the eight scenarios

The boxplots in Figure 4 separate S8 and S4 from the broader distributions of the remaining configurations. The extremely narrow S8 interval is consistent with its sub-micro-unit standard deviation.

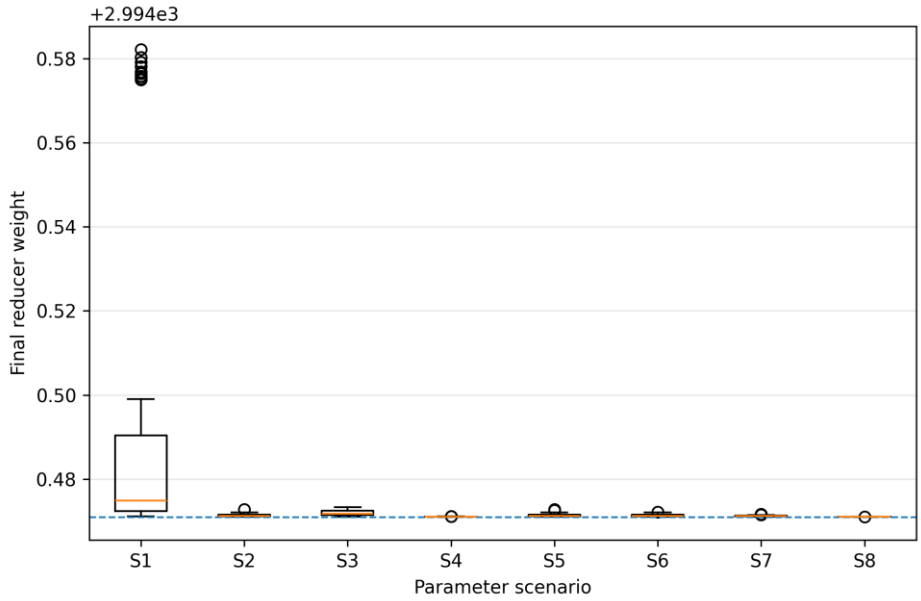


Figure 4: Distribution of final objective values over 50 runs

Figure 5 confirms that final feasibility was not the source of the ranking difference. The decoder and feasibility-first local search returned a feasible final incumbent in every run.

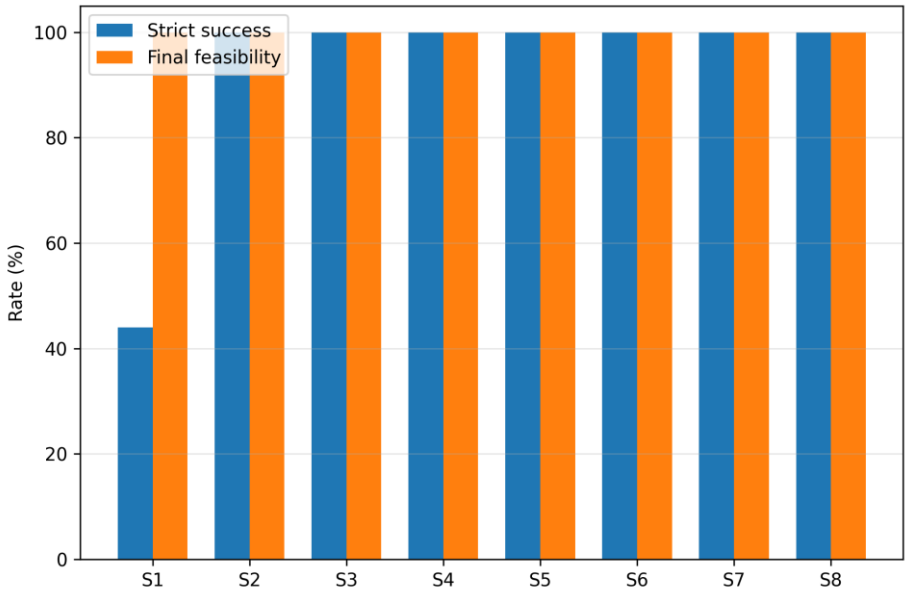


Figure 5: Strict success and final feasibility rates for 50 runs

4.2. Computational Effort and Trade-Off

Table 5 separates objective quality from computational cost. S8 used approximately 125,533 evaluations per run and had the largest mean validation runtime. S4 required about half as many evaluations and remained numerically close. S1 was fastest, but its smaller search budget produced the lowest success rate and the largest mean deviation. Figure 6 places S8 at the precision-oriented end of the trade-off. S4 is appropriate when a lower evaluation budget is preferred, while S2 and S6 provide intermediate choices.

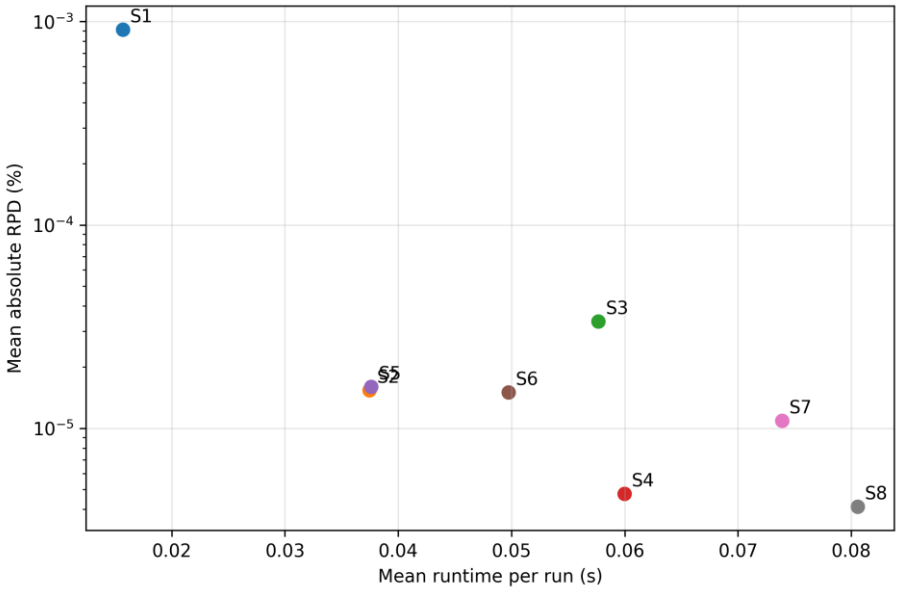


Figure 6: Mean accuracy-runtime trade-off for the eight scenarios

4.3. Statistical Comparison

The Friedman tests in Table 6 were significant for all three nested samples. At 50 runs, the statistic was 259.793658 with $p=2.277\times10^{-52}$ and Kendall's $W=0.742268$, indicating a strong parameter effect within the tested scenario set. The mean objective rank of S8 was 1.04, followed by S4 at 2.28. Table 7 shows that S8 remained significantly more accurate than every alternative after Holm correction. It won all 50 paired comparisons against S1, S2, S3, S5, and S7, and won 49 of 50 comparisons against S4 and S6. The rank-biserial effects were at least 0.990588.

4.4. Best Design and Active Constraints

The best result was obtained by S8 with seed 23.

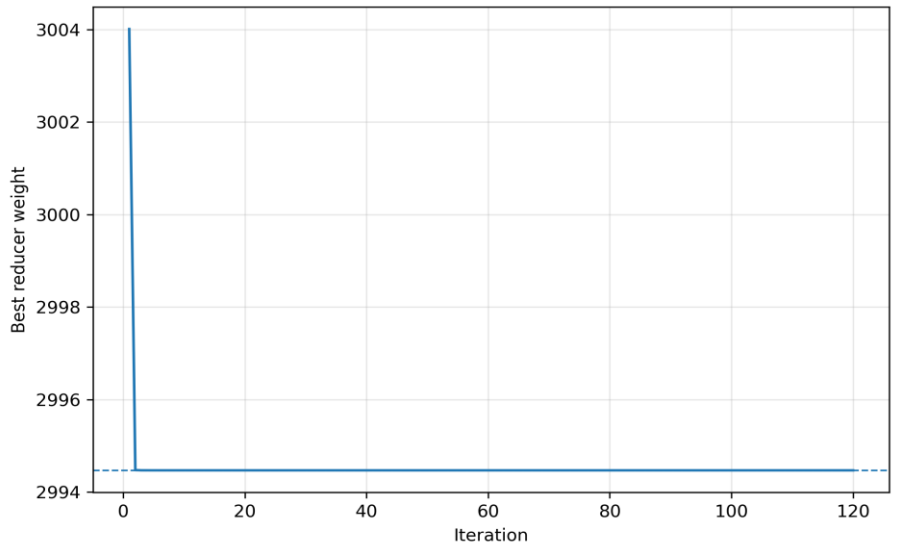


Figure 7: Best-so-far objective for the overall best S8 run

The design and its complete residual vector are reported in Tables 8 and 9. The face width x_1, the first shaft span x_4, and the module x_2 lie on their adopted bounds. Constraints g_5, g_6, g_8, and g_{11} are active or numerically near-active. The remaining constraints retain finite feasible margins.

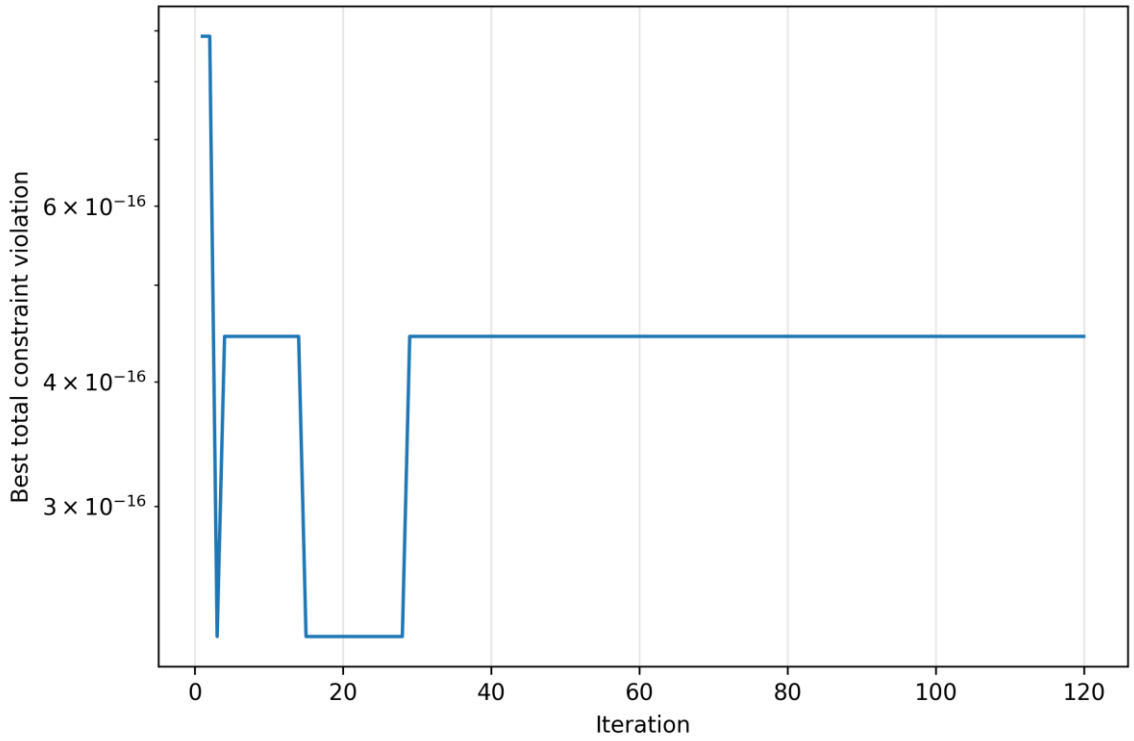


Figure 8: Total constraint-violation history for the overall best S8 run

Figures 7 and 8 show that the run reached the target region in iteration 3 and retained a feasible incumbent. Later iterations were used to refine the final digits and consolidate pheromone concentration (Figure 9).

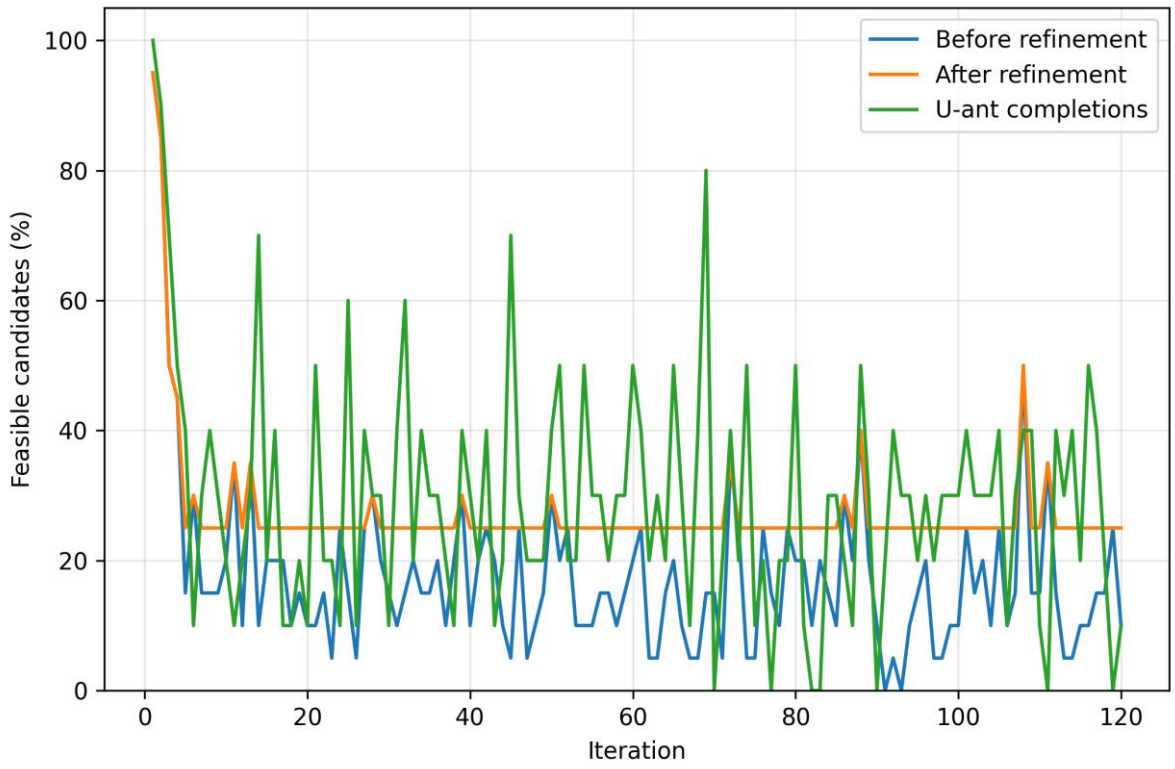


Figure 9: Raw and post-refinement feasible-candidate rates in the best run

The feasible-candidate rate is lower for S8 than for several balanced settings because its fine search is concentrated near active boundaries. Local refinement increases the feasible share while preserving the best feasible incumbent (Figure 10).

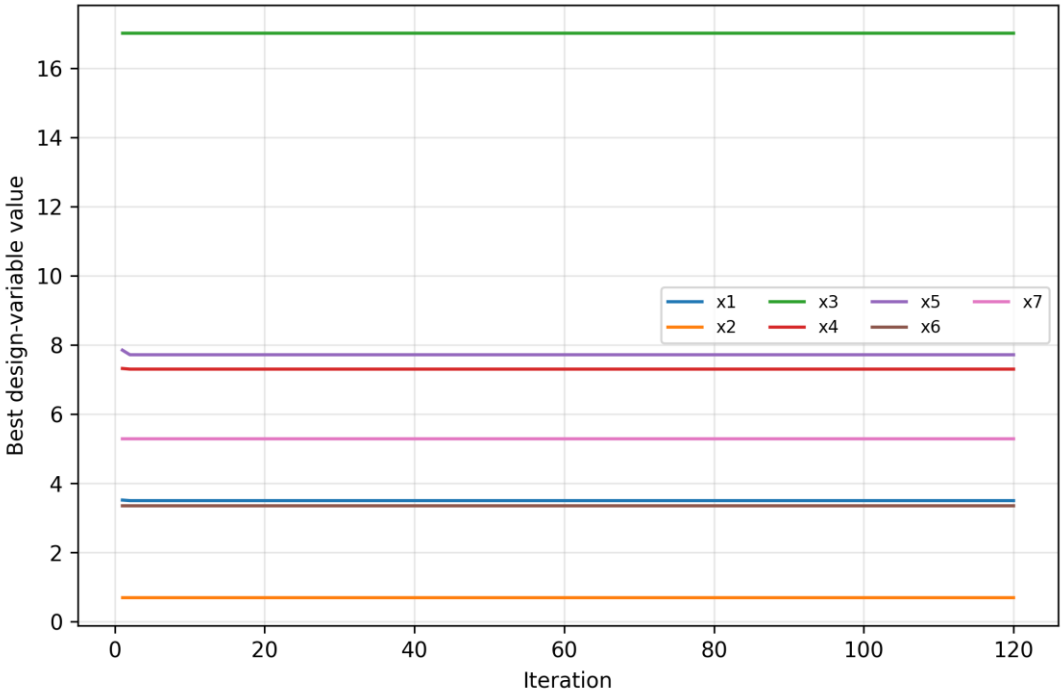


Figure 10: Convergence histories of the seven physical design variables

Figures 10 to 12 connect the objective trajectory to the mechanical variables and constraints. The shaft diameters approach the stress-controlled boundaries, while x_1 is fixed by g_8.

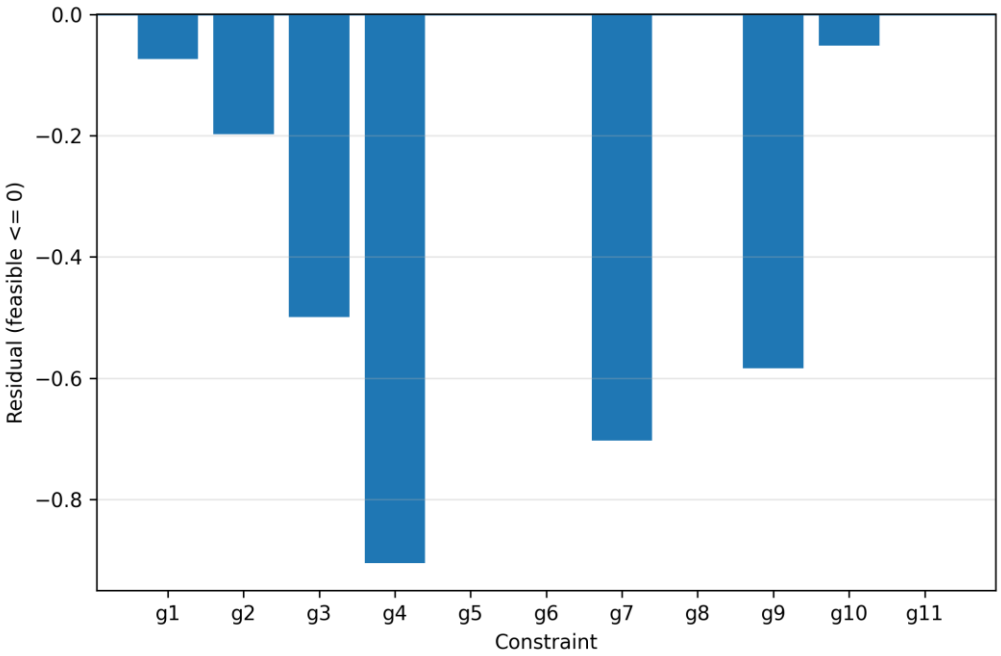


Figure 11: Final residuals of all eleven constraints for the best design

This explains why small changes in d_1, d_2, or the second shaft span produce measurable changes in the final weight (Figure 12).

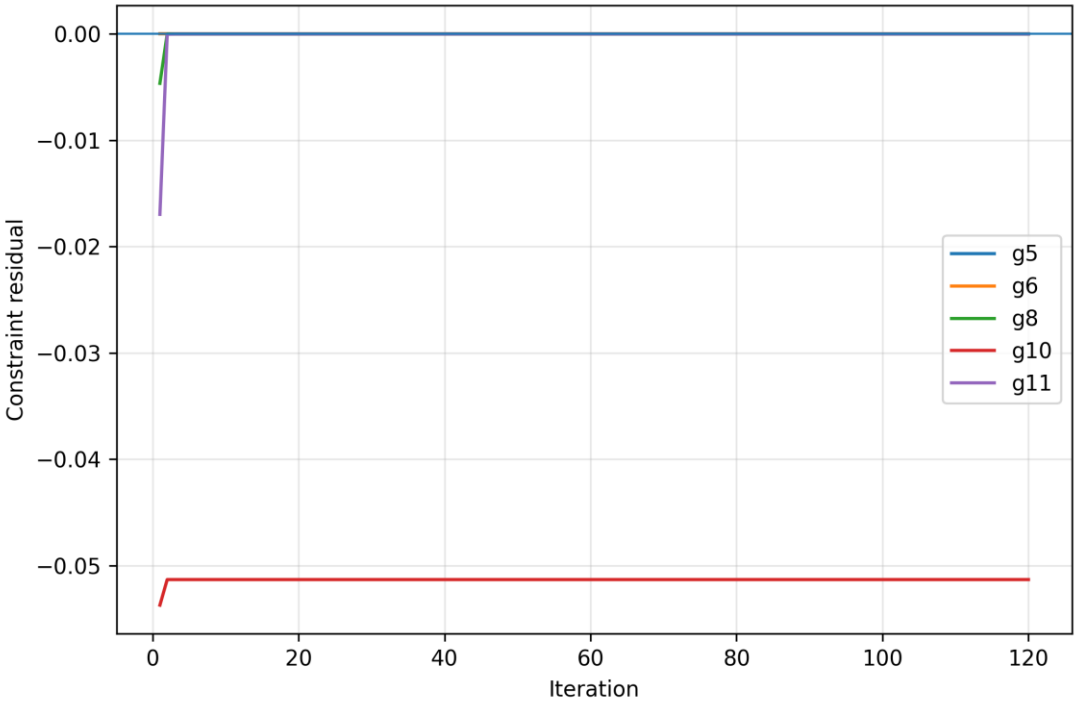


Figure 12: Histories of selected active and near-active constraints

4.5. Diversity, Pheromone Concentration, and Run-Count Stability

Figure 13 shows the evolution of the normalized diversity of the population during the 120 iterations of the optimization procedure. At first, the population variety reduces rapidly, which means the convergence toward promising solutions is fast. We observe minor oscillations in the first few iterations and persistently low variety for the rest of the optimization process. The results show an effective stabilized search, reduced population variance and convergence to an optimum solution.

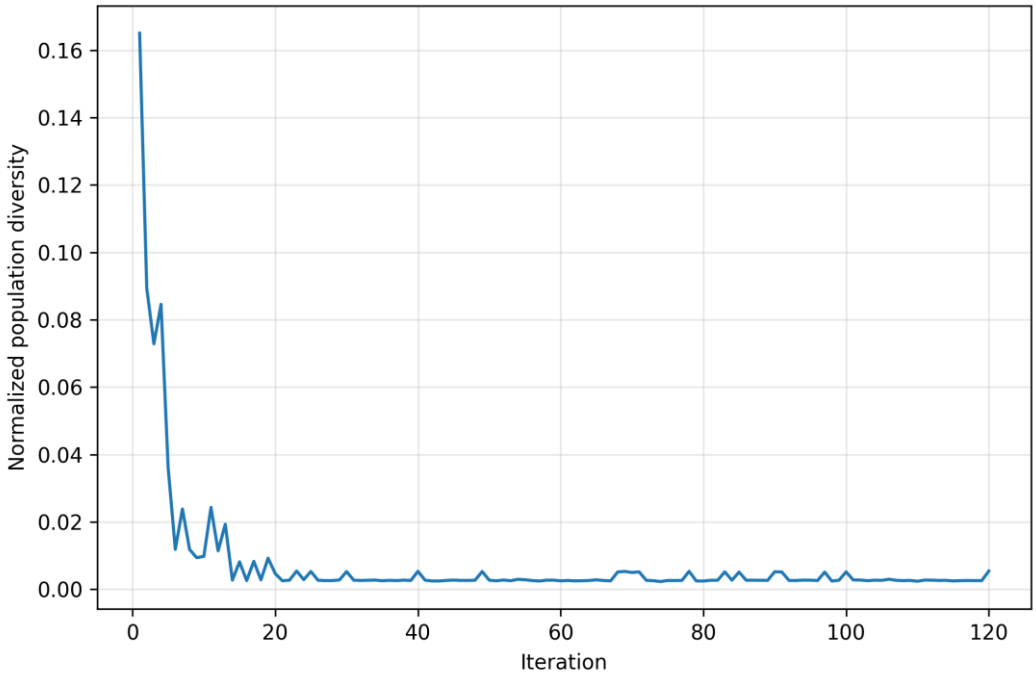


Figure 13: Normalized population diversity during the overall best run

Diversity and entropy decrease as the colony concentrates on the active-boundary region, but neither collapses immediately. The U-ant stage continues to sample partial combinations while complete ants and local refinement exploit the best interval (Figure 14).

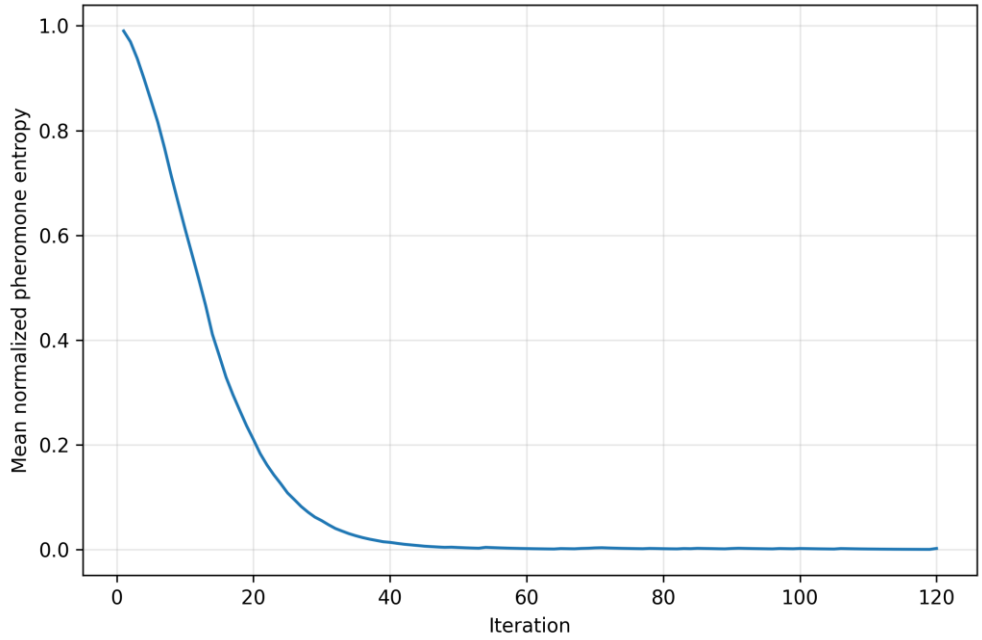


Figure 14: Normalized pheromone entropy during the overall best run

Table 10 and Figure 15 show that the best value, mean, feasibility, and success estimates were already stable at 10 runs. Increasing the sample to 30 and 50 runs mainly refined the dispersion estimate. Thirty matched runs are sufficient for the main parameter comparison under this model, although 50 runs provide stronger paired-test resolution.

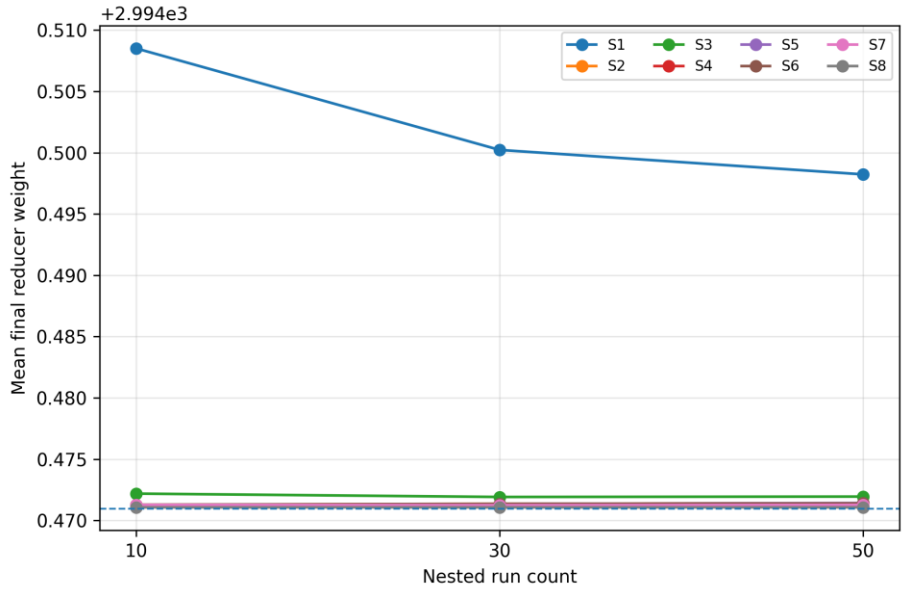


Figure 15: Stability of S8 statistics for nested samples of 10, 30, and 50 runs

4.6. Weight Composition and Literature Comparability

Figure 16 separates the objective components and shows why the optimum is sensitive to shaft diameter. Published values below or above the conventional reference cannot be ranked solely by their final number. The comparison must first establish

that the same eleven constraints, variable bounds, coefficients, and tooth-count integrality were used. The recent SDOA study is therefore cited as a related speed-reducer application rather than inserted into a direct numerical ranking under an assumed common convention.

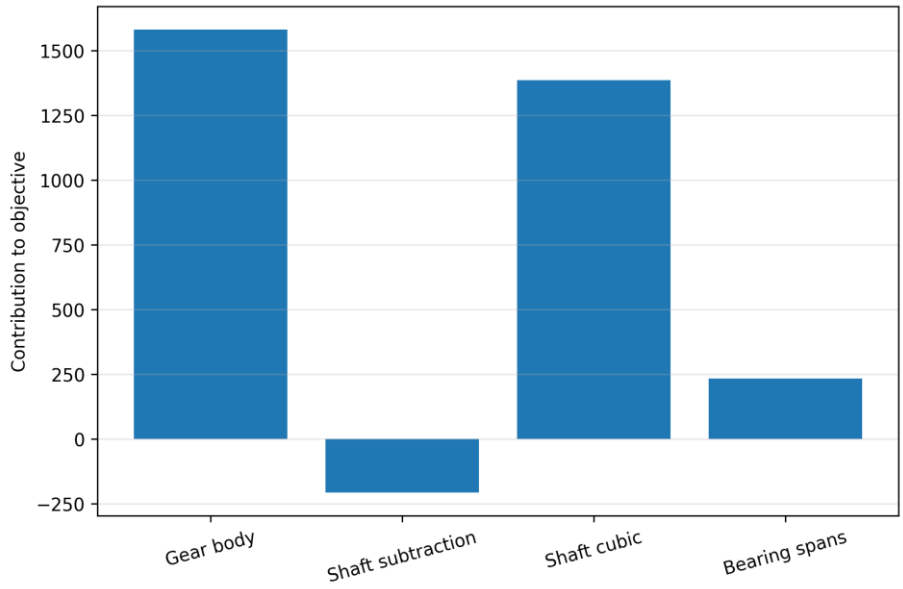


Figure 16: Decomposition of the objective terms for the best design

The main result is not that U-TACO changes the known engineering design of this benchmark, but that its partial-decision mechanism can be transferred to a mixed-integer continuous model without assigning pheromone to unevaluated partial vectors. The decoder makes the transfer technically coherent: completion, integrality, and feasibility are resolved before any U-ant credit is deposited. The scenario results also clarify the role of search structure. Strong exploitation alone is effective: S4 obtained the second-best objective rank with a smaller budget than S8. Strong U-turn guidance, represented by S6, improved the balanced setting but did not replace fine complete-colony refinement. S8 performed best because interval resolution, completion around the incumbent, local passes, and global refinement jointly reduced final numerical variation. All final designs were feasible, but population feasibility differed substantially. This distinction matters. Reporting only the feasibility of the returned incumbent would conceal whether the method explores mostly valid designs or relies on repair near active boundaries. The raw/post-refinement histories provide evidence about the constraint-handling process rather than only its final output.

5. Conclusion

A continuous mixed-integer U-TACO method was developed for constrained speed reducer design. Partial U-ant decisions were completed and evaluated before only the sampled coordinates received pheromone credit. Complete ants, a model-based decoder, feasibility-first refinement, and bounded interval pheromone formed the remainder of the search. Across 400 matched runs, S8 obtained the best mean accuracy and smallest dispersion. Its best feasible objective was 2994.471066, all 50 final designs were feasible, and paired nonparametric tests favored it over each alternative scenario. S4 remained a practical lower-cost option. Nested 10-, 30-, and 50-run results were stable, and the complete residual vector confirmed that four constraints were active or near-active. The study supports the use of partial-decision pheromone learning for continuous engineering design when the credit rule is based on a completed feasible model rather than on an isolated partial vector.

5.1. Limitations

The conclusions are specific to the adopted seven-variable, eleven-constraint benchmark. The algebraic decoder uses monotonic relations that may not exist in a general gearbox model. The experiment evaluates parameter sensitivity within eight designed scenarios and does not claim global optimality. The canonical reference is used for accuracy reporting, not as a proof of the global minimum. Finally, runtimes were measured in a Python/Numba validation environment; MATLAB timings should be regenerated on the target workstation before any platform-specific efficiency claim.

Acknowledgment: Conceptualization, S.M.A. and R.R.A.; methodology, S.M.A.; software, S.M.A.; validation, S.M.A. and R.R.A.; formal analysis, S.M.A.; investigation, S.M.A.; data curation, S.M.A.; writing-original draft preparation, S.M.A.; writing-review and editing, S.M.A. and R.R.A.; visualization, S.M.A.; supervision, R.R.A. All authors have read and agreed to the manuscript.

Data Availability Statement: The MATLAB implementation, Python validation implementation, scenario definitions, run-level CSV files, convergence histories, figures, and statistical outputs are included in the accompanying research package.

Funding Statement: No external funding was received for this study.

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

Ethics and Consent Statement: This research adhered to established ethical principles and relevant institutional guidelines. Informed consent was secured from participants wherever applicable, ensuring voluntary participation and ethical compliance.

References

1. J. Golinski, "Optimal synthesis problems solved by means of nonlinear programming and random methods," *Journal of Mechanisms*, vol. 5, no. 3, pp. 287–309, 1970.
2. J. Golinski, "An adaptive optimization system applied to machine synthesis," *Mechanism and Machine Theory*, vol. 8, no. 4, pp. 419–436, 1973.
3. M. H. Lin, J. F. Tsai, N. Z. Hu, and S. C. Chang, "Design optimization of a speed reducer using deterministic techniques," *Mathematical Problems in Engineering*, vol. 2013, no. 1, pp. 1–7, 2013.
4. M. Dorigo and T. Stützle, "Ant Colony Optimization." *MIT Press*, Cambridge, Massachusetts, United States of America. 2004.
5. K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1155–1173, 2008.
6. K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2–4, pp. 311–338, 2000.
7. C. A. C. Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11–12, pp. 1245–1287, 2002.
8. Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.
9. K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 36–38, pp. 3902–3933, 2005.
10. M. Jaberipour and E. Khorram, "Two improved harmony search algorithms for solving engineering optimization problems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 11, pp. 3316–3331, 2010.
11. 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.
12. F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.
13. S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979.
14. J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, no. 12, pp. 1–30, 2006.
15. S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization," *Journal of Heuristics*, vol. 15, no. 5, pp. 617–644, 2009.
16. K. V. Price, R. M. Storn, and J. A. Lampinen, "Differential evolution: a practical approach to global optimization," *Springer*, Berlin and Heidelberg, Germany, 2005.
17. J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Networks*, Perth, Western Australia, Australia, 1995.
18. J. Wang, S. Luo, and D. Su, "Multi-objective optimal design of cycloid speed reducer based on genetic algorithm," *Mechanism and Machine Theory*, vol. 102, no. 8, pp. 135–148, 2016.

Publisher's Note: *The publisher remains impartial concerning jurisdictional claims in published maps and institutional affiliations. Responsibility for the content rests entirely with the authors and does not necessarily reflect the publisher's perspectives.*