BAREKENG: Journal of Mathematics and Its Applications
June 2026 Volume 20 Issue 2 Page 1229–1250
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol20iss2pp1229-1250

ENHANCED GIANT TREVALLY OPTIMIZER FOR ENGINEERING DESIGN AND EPIDEMIOLOGICAL MODEL

Ikhsan Rizqi Az-Zukruf As-Shidiq[1*], **E. Andry Dwi Kurniawan**[2], **Kuntjoro Adji Sidarto**[3]

[1,2,3]*Department of Mathematics, Faculty of Mathematics and Natural Sciences, Institut Teknologi Bandung*
Jln. Ganesa 10, Bandung, 40132, Indonesia

*Corresponding author's e-mail: * **ikhsanrizx@gmail.com***

Article Info

Article History:

Received: 17th May 2025
Revised: 11th July 2025
Accepted: 23rd September 2025
Available online: 26th January 2026

Keywords:

Benchmark functions;
Engineering design;
Giant Trevally Optimizer;
Metaheuristics;
Optimization;
Sobol sequence.

ABSTRACT

Metaheuristic algorithms are widely used for solving complex optimization problems, but their performance often depends on the initialization strategy. This study proposes an enhanced Giant Trevally Optimizer (GTO) by introducing quasi-random Sobol sequences in the initialization phase, yielding the Sobol-initialized Giant Trevally Optimizer (SGTO). The algorithm was tested on forty benchmark functions, five engineering design problems, and an epidemiological model case study. Experimental results show that SGTO consistently outperforms the original GTO in terms of achieving optimal solutions, convergence, and its ability to maintain a consistent solution across multiple independent runs. Furthermore, the epidemiological case study demonstrates the adaptability of SGTO for tackling more complex real-world problems. This work is the first to adapt Sobol sequences for the GTO and apply it to an epidemiological model. These findings confirm that quasi-random initialization substantially improves exploration and exploitation, establishing SGTO as a versatile and reliable optimization tool.

How to cite this article:

I. R. A.-Z. As-Shidiq, E. A. D. Kurniawan, and K. A. Sidarto, "ENHANCED GIANT TREVALLY OPTIMIZER FOR ENGINEERING DESIGN AND EPIDEMIOLOGICAL MODEL", *BAREKENG: J. Math. & App.*, vol. 20, no. 2, pp. 1229-1250, Jun, 2026.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article · Open Access

1. INTRODUCTION

Optimization plays a crucial role in a wide range of scientific and engineering domains, including structural design, control systems, machine learning, and scheduling problems [1]. These problems typically require finding the best configuration of decision variables that either minimizes or maximizes an objective function, often under a set of constraints. However, in many real-world applications, the objective functions involved are often found to be nonlinear, non-differentiable, high-dimensional, multimodal, or non-convex, which have proven difficult to solve using analytical or classical mathematical methods such as gradient descent [2], [3]. To overcome these challenges, researchers have turned to more flexible and adaptive optimization strategies, such as metaheuristic algorithms [4], [5].

Metaheuristic algorithms have gained widespread attention due to their ability to search complex, nonlinear, and multimodal landscapes without relying on gradient information or strict mathematical assumptions such as convexity or differentiability. Inspired by natural processes such as biological evolution, swarm intelligence, and physical phenomena, metaheuristic algorithms offer a balance between exploration and exploitation that enables them to find near-optimal solutions even in challenging search spaces. Several well-known examples include Genetic Algorithms (GA) [6], Particle Swarm Optimization (PSO) [7], Teaching-Learning Based Optimization (TLBO) [8], and Gravitational Search Algorithm (GSA) [9], each drawing inspiration from different natural phenomena to tackle diverse optimization problems. The success of these approaches has led to a growing domain of research dedicated to developing novel and more specialized metaheuristics in order to solve increasingly complex and high-dimensional problem domains.

Among the new generation of bio-inspired optimization algorithms, one of the recent developments is the Giant Trevally Optimizer (GTO), a swarm-based metaheuristic algorithm inspired by the intelligent hunting behavior of *Caranx ignobilis*, a predatory fish known for its strategic attacks on seabirds [10]. GTO simulates this behavior through three main phases: extensive search, area selection, and prey attack. In the extensive search phase, the algorithm exhaustively investigates various regions of the search space. And then, during the area selection phase, the search agents concentrate on the neighborhood of higher-quality solutions inside the featured space. Finally, in the prey attack phase, agents exploit the selected regions to refine solutions and converge toward the global optimum. By modeling these strategies mathematically, GTO offers an effective balance between exploration and exploitation, enabling it to effectively solve complex optimization problems.

One of the main concerns regarding the procedure of seeking optimality is the randomness involved in the search space [11]. In many metaheuristic algorithms, including GTO, the initial random distribution of search agents plays a critical role in determining the quality of the search process. A poorly initialized population may fail to cover the search space adequately, leading to premature convergence and suboptimal solutions. Motivated by this observation, this study proposes an enhancement to GTO's initialization phase by incorporating a quasi-random low-discrepancy sequence, aiming to generate an initial population that is more uniformly distributed across the search space. The modified GTO is expected to improve its exploration capability, accelerate convergence, and achieve higher solution accuracy.

The novelty and contribution of this research lie in the design of an improved version of the GTO, named the Sobol-initialized Giant Trevally Optimizer (SGTO), which is capable of providing a more uniform and effective exploration of the search space, reducing the likelihood of poor initial agent distribution and improving overall optimization performance which results in better convergence to optimal solutions and faster optimization. The proposed SGTO enhances the original GTO by incorporating a quasi-random low-discrepancy sequence during the initialization phase to generate search agents that are evenly distributed across the search space. The mathematical formulation of the modified initialization is provided, and the original flowchart of GTO is rewritten with indentation for clarity. Forty objective functions of unimodal, multimodal, separable, and non-separable types have been utilized to evaluate the effectiveness of the proposed SGTO in optimization. Furthermore, SGTO is applied to five complex engineering design problems as well as adapted to solve a system of differential equations arising in epidemiological modeling. Finally, the performance of SGTO is compared against the original GTO, demonstrating faster convergence and higher solution accuracy across the tested problems.

The paper is organized as follows: Section 2 presents research methods, along with the details of the proposed algorithm, proposed flow chart, and the pseudo code; Section 3 explains the numerical results and discussion. Finally, concluding remarks are provided in section 4.

2. RESEARCH METHODS

To investigate the impact of the proposed modification, this section presents the research methodology adopted in the study. First, the structure and working mechanism of the original GTO are briefly reviewed to establish a baseline understanding. Then, the proposed improvement focusing on the initialization phase is described in detail, including the mathematical incorporation of quasi-random principles. Finally, the experimental setup, benchmark problems, and evaluation procedures used to assess and compare the performance of the modified algorithm are outlined.

2.1 Giant Trevally Optimizer (GTO)

The Giant Trevally Optimizer (GTO) is a bio-inspired metaheuristic algorithm designed to solve global optimization problems by mimicking the predatory behavior of giant trevally (*Caranx ignobilis*), which are found in the Indian and Pacific Oceans. These fish select prey-dense areas and start to stalk their prey, then leap out of the water to catch the prey mid-air. Their hunting behaviors serve as the biological foundation for modeling the three main phases of GTO: extensive search, choosing an area, and chasing and attacking prey.

The optimization process begins with a random initialization of the population, where each agent (giant trevallies) represents a candidate solution to the optimization problem. Each agent's position is initialized using Eq. (1):

$$X_{i,j}(0) = Minimum_j + \left(Maximum_j - Minimum_j\right) \times R, \quad (1)$$

where $Minimum_j, Maximum_j$ represent the lower and upper bound values on the defined problem for the j^{th} dimension, and R is a random number in the interval [0,1] that is generated by a uniform distribution.

The extensive search phase is the first step in GTO's iterative process, where search agents perform a broad exploration of the search space. This phase simulates the foraging behavior of giant trevallies as they travel long distances in search of prey, which can be done by using the Levy flight that is often exhibited by marine predators [12], [13]. Thus, allowing agents to explore distant areas and avoid becoming trapped in local optima. Mathematically, the position update during this phase can be expressed as Eq. (2):

$$X(t+1) = X_{best}(t) \times R + Levy(d) \times (Minimum + (Maximum - Minimum) \times R), \quad (2)$$

where $X(t+1)$ denotes the next position of the agents, $X_{best}(t)$ is the best obtained solution during the last search iteration, and $Levy(d)$ is a random step vector drawn from the Levy distribution in d-dimensional space.

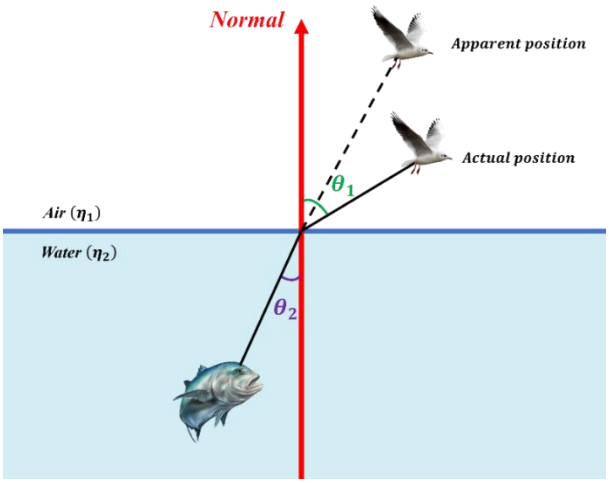


Figure 1. Visual Distortion in GTO Following the Principle of Light Refraction

In the choosing area phase, each agent selects promising regions within the search space based on the current knowledge of the population. This phase simulates the behavior of giant trevallies in identifying and selecting the best area in terms of the amount of prey where they can hunt for prey. Thus, directing agents toward areas that contain better solutions, enhancing the balance between diversification and intensification. This behavior is mathematically simulated in Eq. (3).

$$X(t+1) = X_{best}(t) \times \mathcal{A} \times R - X_i(t) \times R + \frac{1}{N}\sum_{j=1}^{N} X_j(t), \quad (3)$$

where $\mathcal{A}$ is a special parameter to control the rate of changing position, $X_i(t)$ is the location of the i^{th} giant trevally, and N is the number of giant trevallies. The value of $\mathcal{A} = 0.4$ was chosen based on empirical studies and systematic tuning to balance exploration and exploitation effectively during the search process.

Finally, the attacking phase simulates the Giant Trevally's attack by leaping out of the water to catch the prey mid-air. The mathematical model involves visual distortion due to light refractions, launching speed, and the jumping slope function. Fig. 1 demonstrates how the giant trevally perceives its prey due to light refraction at the interface between water and air. The bird's actual position is shown above the water, while the bird's position perceived by the giant trevally from below is displaced due to the bending of light rays. This perceptual distortion alters the trajectory of the jump. The giant trevally does not aim directly at the position of the prey, but rather at the distorted position as perceived through the refraction angle. The inclusion of this behavior in the GTO algorithm mimics how the solution agent adjusts its movement direction not only based on the best-known solution but also under a distortion term that introduces nonlinearity and adaptive search variability. This mechanism contributes to diversifying the search path, improving the algorithm's ability to escape local optima, and mimicking nature's imperfect yet effective decision-making processes. The following Eq. (4) represents a compact form of the original GTO update mechanisms, obtained by merging multiple expressions for brevity without altering their computational structure.

$$X(t+1) = X_i(t) \times \sin(\theta_2) \times F_{obj}\big(X_i(t)\big) + \frac{\eta_2}{\eta_1}\sin(\theta_2) \times |X_{best}(t) - X_i(t)| + \mathcal{H}, \quad (4)$$

where $F_{obj}\big(X_i(t)\big)$ is the fitness value of $X_i(t)$, $\eta_1 = 1.00029$, and $\eta_2 = 1.33$ represents the absolute refractive index of air and water, θ_2 is the angle of refraction given by a random number in the interval $[0°, 360°]$, and $\mathcal{H}$ represents the jumping slope function that gradually decreases over time to control the algorithm's transition from the exploration phase to the exploitation phase, which is given in Eq. (5).

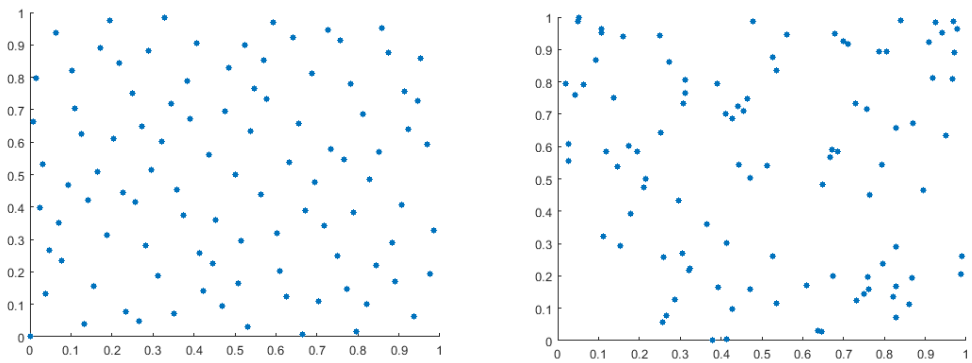
$$\mathcal{H} = R \times \left(2 - t \times \frac{2}{T}\right), \quad (5)$$

where T is the maximum number of iterations allowed during the optimization process.

Overall, GTO dynamically transitions between exploration and exploitation, enabling robust search capabilities over complex landscapes. Further detailed mathematical modeling of each phase and parameter setting guidelines can be found in the original paper.

2.2 Proposed Modification

In the first step of GTO, an initial population of agents is generated randomly. One of the difficulties with using random numbers is their tendency to generate unevenly spaced points in the search space. While such numbers may follow a uniform probability distribution $U[0, 1]$ in a stochastic sense, they often fail to achieve equidistribution, a more desirable form of uniformity in the context of optimization. This lack of uniformity can lead to poorly spread initial populations, especially in high-dimensional search spaces, resulting in clustering and inadequate coverage. Therefore, it is beneficial to generate a population of points in the search region such that the deviation from uniformity is minimum. This deviation is called discrepancy, which is measured as follows.



(a) **(b)**

Figure 2. Scatter Plot of the first 100 Points of (a) Sobol Sequence of Points, and (b) Pseudo-Random of Points

Let $Q \subseteq [0,1]^n$ be an arbitrary axially parallel n-dimensional rectangle in the unit cube $[0,1]^n \subset \mathbb{R}^n$. and let $x_1, x_2, \dots, x_N \in [0,1]^n$ be a set of N points. The idea of discrepancy is that for an evenly distributed point set, the proportion of points lying inside the rectangle Q should correspond to the volume of Q. Let # denote the number of points and vol denote the volume of n-dimensional rectangle, then we expect that

$$\frac{\#\ of\ x_i \in Q}{\#\ of\ all\ points} \approx \frac{vol(Q)}{vol([0,1]^n)}, \tag{6}$$

for as many rectangles as possible. Then, the discrepancy of the point set is defined by

$$D_N := \sup_Q \left| \frac{\#\ of\ x_i \in Q}{N} - vol(Q) \right|. \tag{7}$$

A lower value of D_N indicates a more uniform distribution of points, that is $D_N \to 0$ as $N \to \infty$. Next, a sequence of points $x_1, x_2, \dots, x_N, \dots \in \mathbb{R}^n$ is called low-discrepancy sequence if there is a constant C_n such that for all N,

$$D_N \leq \frac{C_n (\ln N)^n}{N}. \tag{8}$$

Sequences with low discrepancy are also referred to as quasi-random numbers. Among the various constructions of such sequences, the Sobol sequence is one of the most widely used due to its ability to produce uniformly distributed points, especially in high-dimensional spaces [14]-[16].

Initialize the number of giant trevallies N and the maximum number of iterations T
Initialize objective function $F_{obj}(x)$, dimension D, and its respective bounds
Generate Sobol sequence of points $\{S_i \in [0,1]^D \mid i = 1, 2, \dots, N\}$
Initialize the populations of giant trevallies by using (9)
***for** each giant trevally $i = 1$ to N*
    Calculate objective function $F_{obj}(X)$ of the current giant trevally
end for
Sort the value of $F_{obj}(X_i)$ and the population X_i
Determine the global best solution as F_{best}
Determine the best location as X_{best}
***while** $t < T$ (maximum number of iterations)*
    ***for** each giant trevally $i = 1$ to N*
        Calculate new best position for extensive search using (2)
        ***if** $F_{obj}(X_i(t)) < F_{obj}(X_{best})$*
            Replace X_{best} with $X_i(t)$
            ***if** $F_{obj}(X_i(t)) < F_{best}$*
                Replace F_{best} with $F_{obj}(X_i(t))$
            end if
        end if
        Calculate $X_i(t)$ for choosing area using (3)
        ***if** $F_{obj}(X_i(t)) < F_{obj}(X_{best})$*
            Replace X_{best} with $X_i(t)$
            ***if** $F_{obj}(X_i(t)) < F_{best}$*
                Replace F_{best} with $F_{obj}(X_i(t))$
            end if
        end if
        Switch from exploration to exploitation using (5)
        Calculate $X_i(t)$ for attacking phase using (4)
        ***if** $F_{obj}(X_i(t)) < F_{obj}(X_{best})$*
            Replace X_{best} with $X_i(t)$
            ***if** $F_{obj}(X_i(t)) < F_{best}$*
                Replace F_{best} with $F_{obj}(X_i(t))$
            end if
        end if
    $t = t + 1$
end while

Figure 3. Detailed Pseudo-Code of SGTO

Sobol sequences are a type of low-discrepancy quasi-random sequences characterized by their ability to produce points that are evenly and uniformly distributed within a multi-dimensional space. Unlike purely random sequences that may exhibit clustering or gaps, Sobol sequences minimize discrepancy, ensuring that points fill the search space more systematically. This property makes them especially valuable in high-dimensional optimization and integration tasks. By employing Sobol sequences during the initialization phase of GTO, the algorithm benefits from enhanced exploration of the search space due to better spatial coverage of initial agents. This systematic coverage improves both the speed of convergence and the likelihood of escaping local optima, thereby increasing the overall efficiency and robustness of the optimization process.

In the proposed modification, the Sobol sequence is employed during the initialization phase of GTO. Specifically, instead of generating each search agent's position using uniformly distributed random numbers, a sequence of quasi-random vectors is generated using the Sobol sequence. These vectors, defined within the unit hypercube $[0,1]^n$ are then linearly scaled to match the predefined lower and upper bounds of the problem's search space. This modification aims to enhance the algorithm's early-stage exploration by improving the spatial coverage of the search agents. To distinguish this variant, the algorithm is hereafter referred to as the Sobol-initialized Giant Trevally Optimizer (SGTO). In SGTO, the initial position of each search agent in the population is generated as follows.

Let $S_i \in [0,1]^j$ denote the i^{th} point generated by the Sobol sequence, where j is the number of decision variables (dimensions) and $i = 1, 2, \ldots, N$, with N being the population size. Similar to Eq. (1), each Sobol point is scaled to the problem's search space by the following transformation:

$$X_{i,j}(0) = Minimum_j + \left(Maximum_j - Minimum_j\right) \times S_{i,j}, \tag{9}$$

where $X_{i,j}$ represents the initial position of the i^{th} agent in the j^{th} dimension. Fig. 2 illustrates the distribution of 100 two-dimensional points generated by using the Sobol sequence and 100 points generated by using a pseudo-random number generator. Visual inspection reveals that the Sobol sequence provides more even coverage of the space, while the pseudo-random points exhibit clustering and leave noticeable gaps.

Aside from the modified initialization, the core behavior and mathematical model of the original GTO algorithm remain unchanged, allowing a focused assessment of the impact of modified initialization on convergence and accuracy. Additionally, the pseudo-code of the proposed SGTO is presented in Fig. 3.

2.3 Experimental Setup

To evaluate the effectiveness of the proposed SGTO algorithm, a set of numerical experiments was carried out across three problem domains: benchmark optimization functions, classical engineering design problems, and an epidemiological model case study. These experiments aim to assess SGTO's performance in terms of convergence behavior, solution quality, and consistency across multiple independent runs. In all test cases, SGTO was compared exclusively against the original GTO with identical control parameters to ensure a fair and controlled comparison. The case study serves as a demonstration of SGTO's practical applicability to dynamic real-world scenarios. Details of the problems, parameter settings, and performance metrics are provided in the following subsections.

2.3.1 Benchmark Function

The first group of tests comprises 40 benchmark functions with four characteristics: unimodal, multimodal, separable, and non-separable. For each function, both SGTO and GTO were executed for 300 independent runs to capture variability in performance. The population size was set to 30 giant trevallies, and the maximum number of iterations was fixed at 1000 iterations.

For each test case, four primary performance indicators were recorded: the best objective function value achieved, the average (mean) of objective function values across 300 runs, the standard deviation of the objective function values obtained across 300 runs, and the worst objective function value obtained. The mean and standard deviations are formulated in Eqs. (10) and (11).

$$Mean = \frac{1}{Run}\sum_{i=1}^{Run} F_{best}\,, \tag{10}$$

$$Std = \sqrt{\frac{1}{Run}(F_{best} - Mean)^2}, \tag{11}$$

where F_{best} is the global solution and Run is the number of independent runs. It is obvious that the smaller the values of the four criteria, the algorithm are capable to produce more stable and reliable solutions. Additionally, convergence behavior was analyzed by plotting the evolution of the best-found solution over iterations for selected problems. Execution time for each test case was also monitored to evaluate computational efficiency.

Each function is defined within a bounded search domain, and optimization aims to minimize the objective function value. The detailed mathematical formulations, dimensions, bounds, and optimal values for each benchmark function are presented in the original paper and fully described in [17] and [18].

2.3.2 Engineering Design Problem

To evaluate the practical applicability of SGTO, five well-known engineering design problems were implemented in the second group of tests. These problems are frequently used in the optimization literature due to their nonlinear, constrained nature and real-world relevance. The objective in each case is to find the optimal design parameters that minimize a specific cost function while satisfying a set of structural, mechanical, or manufacturing constraints.

For each problem, both algorithms were executed for 300 independent runs, the population size was set to 30 giant trevallies, and the maximum number of iterations was fixed at 3000 iterations. The mean and standard deviations of the best obtained solutions are then calculated by using Eqs. (10) and (11). Each function is defined within a bounded search domain, and optimization aims to minimize the objective function value. The detailed problems and illustrations are fully described in [19]-[24].

2.3.3 Epidemiological Model Case Study

To further evaluate the practical applicability of the proposed SGTO algorithm, a COVID-19 model case study was implemented in the third test. Specifically, the dataset and the SEAR (Susceptible, Exposed, Acute, Recovery) model presented in [25] are utilized without modification. The SEAR model, as depicted in Fig. 4, captures the compartmental dynamics of COVID-19 transmission, dividing the population into Susceptible (S), Exposed (E), Acute (A), and Recovered (R) compartments.

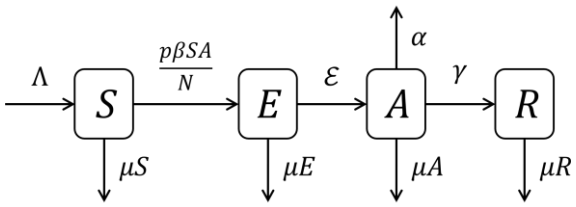


Figure 4. Transmission Diagram of the SEAR Mathematical Model for COVID-19 Spread in Indonesia

The interactions among these compartments are governed by the system of differential equations shown in Eqs. (12), (13), (14), (15), which describe the time evolution of each population group based on disease transmission rates, recovery rates, and natural death rates.

$$\frac{dS}{dt} = \Lambda - \frac{p\beta SA}{N} - \mu S, \tag{12}$$

$$\frac{dE}{dt} = \frac{p\beta SA}{N} - (\mu + \varepsilon)E, \tag{13}$$

$$\frac{dA}{dt} = \varepsilon E - (\alpha + \gamma + \mu)A, \tag{14}$$

$$\frac{dR}{dt} = \gamma A - \mu R, \tag{15}$$

where S is the number of healthy individuals who are vulnerable to the disease, E is the number of individual who have been exposed to the virus but are not yet infectious, A is the number of individuals who are actively infected and infectious, R is the number of individuals who have recovered and are assumed to have permanent immunity, N is the total population size at a given time, p is the migration factor, Λ is the

recruitment rate of new susceptible, β is the transmission rate of the virus, ε is rate of progression from exposure to acute, α is disease-induced mortality rate, γ is the recovery rate, and μ is the natural death rate.

In this study, the migration parameter p is set to 1, consistent with the assumption in the referenced paper that the migration rate remains at a normal level. The natural death rate μ is set to 0.0000384037 based on the average life expectancy in Indonesia, to account for background mortality in all compartments. This choice ensures methodological consistency, allowing for a direct comparison of algorithm performance using the same model structure and data.

Based on the models above, the objective of this problem is to minimize the error percentage determined by the value of five model parameters: $\beta, \varepsilon, \alpha, \gamma$, and Λ. The objective function of this problem is formulated in Eq. (16) as follows.

Minimize:

$$f(X) = \frac{1}{4n}\sum_{i=1}^{n}\left(\left|\frac{w_i - w_i^*}{w_i^*}\right| + \left|\frac{x_i - x_i^*}{x_i^*}\right| + \left|\frac{y_i - y_i^*}{y_i^*}\right| + \left|\frac{z_i - z_i^*}{z_i^*}\right|\right), \tag{16}$$

where n is the data size, w_i, x_i, y_i, z_i is the number of cases in i^{th}-day obtained from the model solutions for each variable, and $w_i^*, x_i^*, y_i^*, z_i^*$ is the number of cases in i^{th}-day based on the real data for each variable, namely S_i, E_i, A_i, R_i, and $S_i^*, E_i^*, A_i^*, R_i^*$ respectively.

The purpose of this case study is to illustrate how real-world problems can be treated as optimization tasks, thus allowing the application of advanced optimization techniques. Particularly, the test is employed to check the validity of SGTO and GTO to be applied to a real-world problem involving a dataset. To ensure consistency and comparability, this paper will use the same COVID-19 dataset used in the literature. This approach not only aligns comparative analysis with the existing literature but also maintains methodological rigor by utilizing identical data and model structures.

3. RESULTS AND DISCUSSION

This section presents the comparative performance of SGTO and GTO based on their optimization results across three problem domains: benchmark functions, engineering design problems, and an epidemiological model. The primary objective is to examine the influence of incorporating a Sobol sequence in the initialization phase on the convergence behavior, solution quality, and computational efficiency across different problem types.

3.1 Results on Benchmark Functions

First, the descriptive statistical analysis presents the best (minimum), worst (maximum), mean, and standard deviation of optimization results for each benchmark function. As shown in Table 1, SGTO consistently outperformed GTO in terms of best and mean objective function values across all 40 benchmark functions, indicating the efficacy of quasi-random initialization in achieving superior global search coverage. This improvement is particularly evident in complex multimodal functions, suggesting that quasi-random initialization enhances the algorithm's early-stage exploration capability.

Table 1. Comparison of Optimization Results Obtained for 40 Benchmark Functions

Function	Initialization	Indicator			
		Best	Mean	Std.	Worst
F1	Uniform	0	0	0	0
	Sobol	0	0	0	0
F2	Uniform	0	0	0	0
	Sobol	0	0	0	0
F3	Uniform	0	0	0	0
	Sobol	0	0	0	0
F4	Uniform	3.38525e-07	2.92321e-05	2.59196e-05	0.000144772
	Sobol	1.25941e-07	2.78514e-05	2.26393e-05	0.000111131
F5	Uniform	2.798e-08	7.11114e-05	0.000163472	0.001400477
	Sobol	5.8661e-09	6.77195e-05	0.000163353	0.001496631

Function	Initialization	Indicator			
		Best	Mean	Std.	Worst
F6	Uniform	-0.999999999	-0.999994359	8.07969e-06	-0.999942243
	Sobol	-1	-0.999994844	7.40329e-06	-0.999937562
F7	Uniform	0	0	0	0
	Sobol	0	0	0	0
F8	Uniform	5.26122e-11	3.34209e-06	8.71266e-06	9.89232e-05
	Sobol	1.74029e-12	3.27968e-06	8.02896e-06	7.9745e-05
F9	Uniform	-49.99679961	-49.974143	0.017657015	-49.91002395
	Sobol	-49.99918951	-49.97510998	0.017065942	-49.90639252
F10	Uniform	-209.7419641	-208.4276972	0.818360554	-205.0789349
	Sobol	-209.8337961	-208.5022905	0.812124732	-205.3469735
F11	Uniform	0	0	0	0
	Sobol	0	0	0	0
F12	Uniform	0	0	0	0
	Sobol	0	0	0	0
F13	Uniform	0	3.2232e-266	0	2.0915e-264
	Sobol	0	0	0	0
F14	Uniform	0	0	0	0
	Sobol	0	0	0	0
F15	Uniform	2.88863e-10	3.95297e-05	6.81962e-05	0.000539295
	Sobol	6.54265e-11	3.86996e-05	8.96985e-05	0.00078598
F16	Uniform	0.361200905	0.519630123	0.046317335	0.561227875
	Sobol	0.335928346	0.518470912	0.045585594	0.561218823
F17	Uniform	0.998003838	4.043044755	3.163046766	10.76318076
	Sobol	0.998003838	3.601954815	1.611273813	4.950491316
F18	Uniform	0.397887496	0.397923624	6.61164e-05	0.398687607
	Sobol	0.39788746	0.397912308	2.43179e-05	0.39804469
F19	Uniform	0	0	0	0
	Sobol	0	0	0	0
F20	Uniform	0	0	0	0
	Sobol	0	0	0	0
F21	Uniform	-12300.45544	-6938.720956	1587.407462	-4230.529436
	Sobol	-12543.04909	-9687.180084	1462.894813	-8282.324765
F22	Uniform	-1.801303277	-1.779556247	0.110360158	-1.213885896
	Sobol	-1.801303407	-1.787402461	0.088641985	-1.213928373
F23	Uniform	0	0	0	0
	Sobol	0	0	0	0
F24	Uniform	-1.031628452	-1.03162643	2.43465e-06	-1.031610792
	Sobol	-1.031628453	-1.031626501	2.19649e-06	-1.031613702
F25	Uniform	0.18	0.18	2.22045e-16	0.18
	Sobol	0.18	0.18	2.22045e-16	0.18
F26	Uniform	0	0	0	0
	Sobol	0	0	0	0
F27	Uniform	-186.7308888	-186.7051266	0.051282519	-186.3108865
	Sobol	-186.7309062	-186.7095935	0.029478442	-186.5577986
F28	Uniform	3.000002172	4.170346413	7.707063001	84.02092422
	Sobol	3.000000107	3.991951289	5.074167186	30.000341
F29	Uniform	0.000307638	0.000318621	1.32023e-05	0.000462732
	Sobol	0.000307569	0.000318333	1.03873e-05	0.000375863
F30	Uniform	-10.15295453	-10.13252176	0.017005832	-10.04382616
	Sobol	-10.15318733	-10.13287387	0.015353458	-10.05663323
F31	Uniform	-10.40237253	-10.3798188	0.018875398	-10.30688898
	Sobol	-10.40271789	-10.38176051	0.020988182	-10.19237174
F32	Uniform	-10.53589419	-10.51277832	0.018866881	-10.41872719
	Sobol	-10.53619789	-10.51418148	0.016738175	-10.44138049
F33	Uniform	0.006411844	20.0135309	39.66525575	39.66525575
	Sobol	0.005819206	13.34216806	17.52652296	172.0519372
F34	Uniform	0	0	0	0
	Sobol	0	0	0	0
F35	Uniform	-3.8627775	-3.859679781	0.006847554	-3.804867162
	Sobol	-3.862781547	-3.862620891	0.000238942	-3.860818898

Function	Initialization	Indicator			
		Best	Mean	Std.	Worst
F36	Uniform	-3.321943883	-3.279385233	0.065068461	-3.10532123
	Sobol	-3.321965501	-3.318085905	0.019773765	-3.156633643
F37	Uniform	0	0	0	0
	Sobol	0	0	0	0
F38	Uniform	4.44089e-16	4.44089e-16	0	4.44089e-16
	Sobol	4.44089e-16	4.44089e-16	0	4.44089e-16
F39	Uniform	0.011107381	0.027136237	0.008466955	0.058664404
	Sobol	5.71962e-05	0.026634453	0.008825925	0.054782729
F40	Uniform	-2.423112689	-1.153174657	0.301352462	-1.080726151
	Sobol	-2.423118765	-1.573883678	0.743770693	-0.907991324

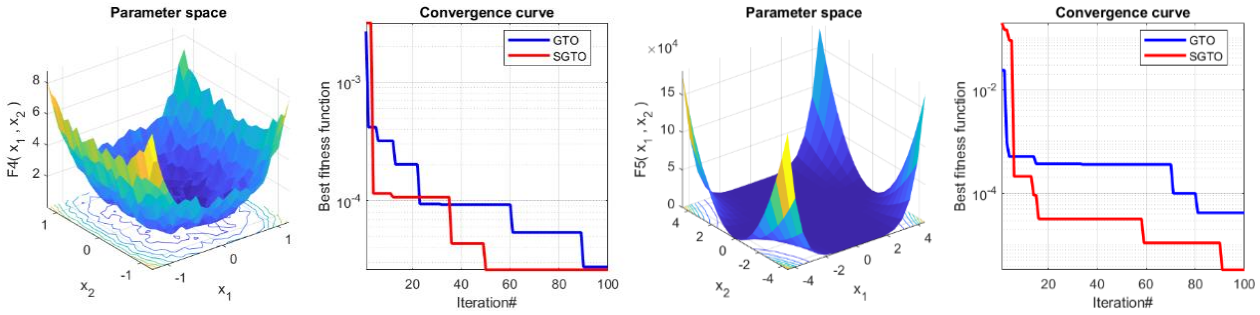
** Std: Standard deviation.*

Nonetheless, in 5 out of 40 test cases, SGTO exhibited slightly higher standard deviations compared to GTO. Additionally, despite its superior optimization performance, SGTO showed marginally higher computational demands in certain execution-time metrics. Specifically, SGTO demonstrated slightly longer execution times than GTO in all three measured execution-time metrics in 6 out of 40 test cases. However, these time differences were minimal, often amounting to mere fractions of a second, as depicted in Table 2. Taken together, these findings highlight a performance trade-off, where SGTO consistently provides better solution quality while incurring only minor increases in variability and computational costs. Despite these small compromises, the improved results suggest that the benefits of SGTO generally outweigh the computational overhead.

Table 2. Execution Time Results for 6 Benchmark Functions Where SGTO Exhibits Longer Runtime

Function	Best		Mean		Standard Deviation	
	GTO	SGTO	GTO	SGTO	GTO	SGTO
F14	0.90625	2.234375	1.066927083	2.90671875	0.22176295	0.473256746
F16	0.875	1.09375	0.969114583	1.256145833	0.179727886	0.199159175
F17	6.609375	6.734375	7.782291667	9.375520833	0.754003349	0.920652291
F18	0.375	0.40625	0.473333333	0.514010417	0.14912447	0.172908344
F25	0.359375	0.390625	0.454791667	0.469010417	0.12709375	0.144368339
F31	0.625	0.640625	0.84203125	0.850208333	0.191348035	0.219605371

The modest increase in SGTO's execution time can be traced primarily to the added computational overhead of quasi-random initialization and its downstream effects on each optimization run. SGTO replaces purely random sampling with Sobol sequences to seed its initial population. Generating these sequences requires extra arithmetic, such as matrix operations, bit-wise scrambling, and gray-code conversions, compared to simple pseudorandom draws [26], [27].



(a) **(b)**

Figure 5. Function Plot and Convergence Curve for Selected Benchmark Functions: (a) F4, (b) F5, (c) F6, (d) F8, (e) F9, (f) F16, (g) F17, (h) F18, (i) F21, (j) F22, (k) F24, (l) F28, (m) F30, (n) F35, (o) F39, and (p) F40

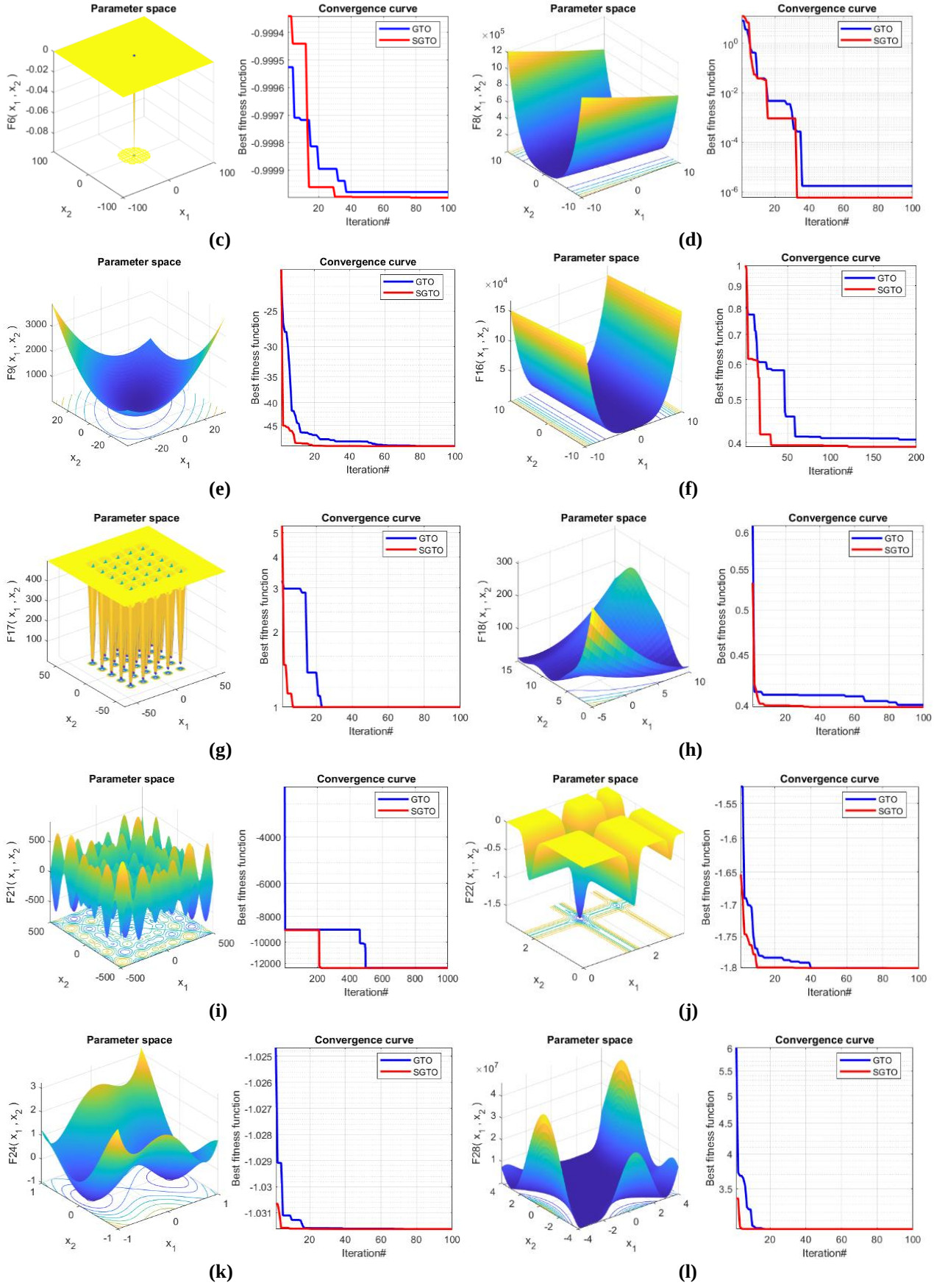


Figure 5. (*Continued.*) **Function Plot and Convergence Curve for Selected Benchmark Functions: (a) F4, (b) F5, (c) F6, (d) F8, (e) F9, (f) F16, (g) F17, (h) F18, (i) F21, (j) F22, (k) F24, (l) F28, (m) F30, (n) F35, (o) F39, and (p) F40**

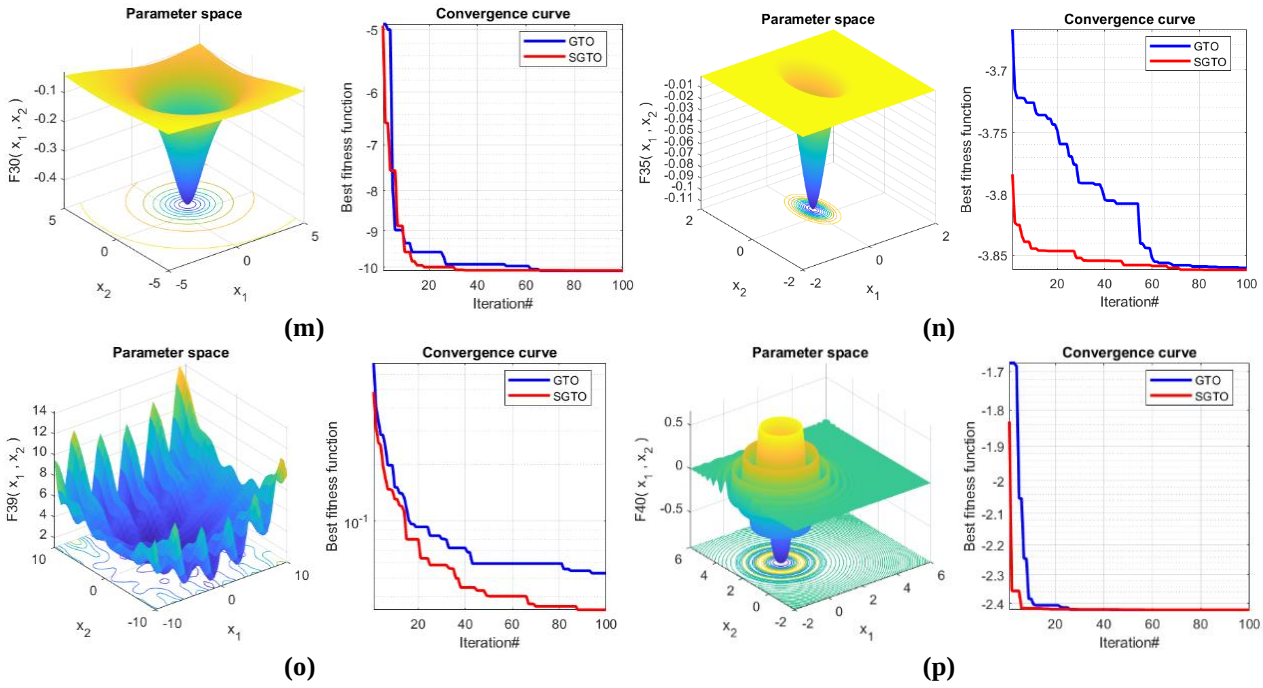


Figure 5. **(*Continued.*) Function Plot and Convergence Curve for Selected Benchmark Functions: (a) F4, (b) F5, (c) F6, (d) F8, (e) F9, (f) F16, (g) F17, (h) F18, (i) F21, (j) F22, (k) F24, (l) F28, (m) F30, (n) F35, (o) F39, and (p) F40**

Fig. 5 illustrates the convergence behavior of both SGTO and GTO across several benchmark functions, highlighting the optimization trajectories and the rate at which optimal solutions are approached. Notably, SGTO's convergence curve often begins with a lower value (approaching the optimal solution) or descends faster, which is attributable to the more equidistributed initialization of search agents facilitated by Sobol sequences. This improved initial spatial coverage enables SGTO to rapidly identify promising regions of the search space, thus accelerating the search process and reaching optimal or near-optimal solutions faster than GTO. As iterations progress, both algorithms exhibit steady convergence; however, SGTO's accelerated early-stage exploration and exploitation contribute to more efficient fitness improvement and a more robust convergence profile. These observations affirm the benefit of Sobol initialization in enhancing GTO's convergence behavior without altering its core search mechanisms.

3.2 Results on Engineering Design Problems

As a matter of fact, metaheuristic algorithms are not designed to solve constrained optimization problems directly [28]. Therefore, we used the straightforward death penalty technique to transform the original problems to their unconstrained form. This technique involves a complete removal of any infeasible solution from the population. The primary limitation of the straightforward death penalty technique is that it can significantly reduce the algorithm's ability to find the global optimal solution, particularly when the feasible region is small or difficult to reach. This is because the technique enforces strict exclusion of infeasible solutions, which can overly restrict the search space, preventing the algorithm from exploring potentially promising regions that are close to the constraint boundaries [29].

3.2.1 Cantilever Beam

This problem involves the weight optimization of a cantilever beam with a square cross-section. The beam is supported at the leftmost block, and there is a given vertical force at the free end, as shown in Fig. 6. The design variables are x_1, x_2, x_3, x_4, x_5, representing the widths of the square cross-section of each five beam segments. Each segments have a fixed and uniform thickness. The objective is to determine the optimal width of each beam segments that minimize the total weight of the beam, subject to stress constraints. The bound constraints are set as $0.01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100$. The mathematical formulation is given as follows.

Minimize:

$$f(X) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5). \tag{17}$$

Subject to:

$$g(X) = \frac{61}{{x_1}^3} + \frac{37}{{x_2}^3} + \frac{19}{{x_3}^3} + \frac{7}{{x_4}^3} + \frac{1}{{x_5}^3} - 1 \leq 0. \tag{18}$$

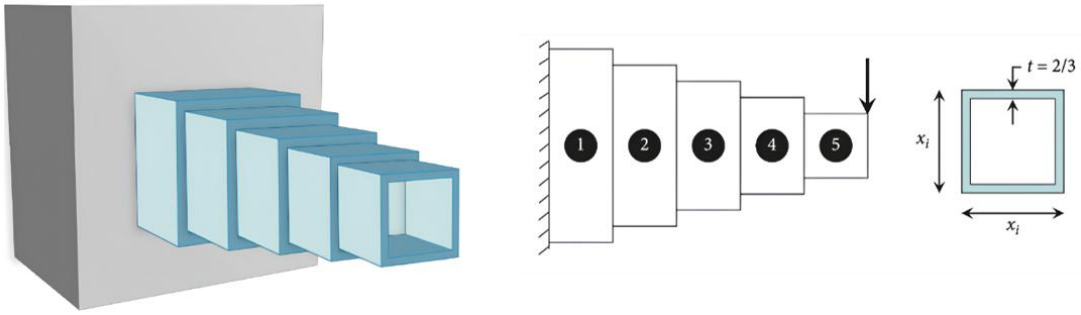


Figure 6. Cantilever Beam Design

Table 3 presents the five best solutions obtained by SGTO and GTO for this problem. It can be observed that SGTO offers better solutions compared to GTO. Furthermore, Table 4 compares the statistical results of SGTO and GTO, showing that SGTO consistently yields more reliable results based on the best, mean, standard deviation, and even the worst possible weight of the cantilever beam.

Table 3. Comparison of the Best Results of the Cantilever Beam Design

	$f(x)$	x_1	x_2	x_3	x_4	x_5
GTO	1.340164074	6.017553256	5.340219595	4.508685432	3.504463145	2.106066942
	1.340218176	5.939619531	5.368692254	4.505261136	3.486172194	2.178110264
	1.340310371	6.020577963	5.214085021	4.521381761	3.553881483	2.16940664
	1.340324582	6.007727763	5.295937455	4.422739076	3.55515644	2.197999868
	1.340382554	6.068366176	5.268713318	4.513413896	3.432460414	2.19753585
SGTO	**1.340083573**	**6.054008663**	**5.292576344**	**4.477170684**	**3.493057772**	**2.158884816**
	1.340090057	6.27784226	5.288057144	4.486191097	3.48077355	2.192996181
	1.340200147	6.070563763	5.237281837	4.524809644	3.507188152	2.137723065
	1.340323951	6.020890548	5.387370025	4.430588565	3.465515659	2.175185696
	1.340400678	5.995763649	5.347262759	4.556968567	3.456662285	2.124122836

Table 4. Comparison of Statistical Results of the Cantilever Beam Design

	Best	Mean	Std.	Worst
GTO	1.340164074	1.34198915	0.001047075	1.346851792
SGTO	1.340083573	1.341886592	0.001039628	1.345402422

3.2.2 Three-Bar Truss

This problem seeks to minimize the weight of a statically loaded three-bar truss under stress, buckling, and deflection constraints. The design variables are the cross-sectional areas x_1 and x_2. The objective is to minimize the total weight of the truss, which is directly proportional to the sum of the products of each bar's length and its corresponding cross-sectional area, as illustrated in Fig. 7. The bound constraints are set as $0 \leq x_1, x_2 \leq 1$. The length of the bar is set as $l = 100\ cm$, the applied vertical load is $P = 2\ kN/cm^2$, and the allowable axial stress is $\sigma = 2\ kN/cm^2$. The optimization model can be expressed as follows.

Minimize:

$$f(X) = 2\sqrt{2}x_1 + x_2 \times l. \tag{19}$$

Subject to:

$$g_1(X) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}{x_1}^2 + 2x_1x_2} P - \sigma \leq 0, \tag{20}$$

$$g_2(X) = \frac{x_2}{\sqrt{2}{x_1}^2 + 2x_1x_2} P - \sigma \leq 0, \tag{21}$$

$$g_3(X) = \frac{x_2}{\sqrt{2}x_2 + x_1} P - \sigma \leq 0. \tag{22}$$

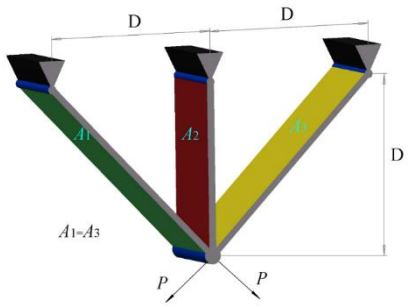


Figure 7. Three-Bar Truss Design

Table 5 presents the five best solutions obtained by SGTO and GTO for this problem, where SGTO consistently produced better solutions compared to GTO. Table 6 further compares the statistical results for both algorithms, showing that SGTO delivers more reliable results across all performance indicators. Notably, SGTO achieved a lower total weight for the truss design.

Table 5. Comparison of the Best Results of the Three-Bar Truss Design

	$f(x)$	x_1	x_2
GTO	263.8915812	0.788668938	0.408191288
	263.8915973	0.788449927	0.408786857
	263.8917638	0.788134957	0.409679163
	263.8917881	0.788681762	0.408165228
	263.8918346	0.788940584	0.407433272
SGTO	**263.8915222**	**0.788448514**	**0.408804094**
	263.8915399	0.788711975	0.408048165
	263.8915637	0.788345792	0.409090389
	263.8915748	0.788971889	0.407319253
	263.8916043	0.789011549	0.407205429

Table 6. Comparison of Statistical Results of the Three-Bar Truss Design

	Best	Mean	Std.	Worst
GTO	263.8915812	263.9541945	0.095367583	264.516143
SGTO	263.8915222	263.9076591	0.013671271	263.9381255

3.2.3 Gear Train Design

The gear train design problem is an unconstrained optimization. As shown in Fig. 8, the design variables are x_1, x_2, x_3, x_4, representing the number of teeth on every four gears, which must be integers (discrete). The objective is to minimize the squared error between a desired gear ratio and the actual gear ratio in a four-gear train. The bound constraints are set as $12 \leq x_1, x_2, x_3, x_4 \leq 60$. The mathematical formulation is as follows.

Minimize:

$$f(X) = \left(\frac{1}{6.931} - \frac{x_2 x_3}{x_1 x_4}\right)^2 \tag{23}$$

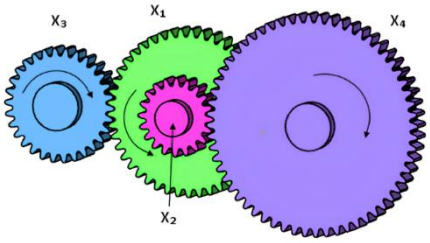


Figure 8. Gear Train Design

Table 7 shows the five best solutions obtained by SGTO and GTO for the gear train design problem, where SGTO demonstrates superior performance in finding optimal solutions. In Table 8, the statistical

comparison between the two algorithms reveals that SGTO consistently provides more reliable results across all evaluation metrics. The solutions offered by SGTO are remarkably different from GTO, allowing it to achieve further reduction in the error of gear ratio.

Table 7. Comparison of the Best Results of the Gear Train Design

	$f(x)$	x_1	x_2	x_3	x_4
GTO	1.72316e-16	31.13911389	12.00723064	12.3368186	32.97127337
	4.53724e-16	33.15643026	12.01598039	12.01756859	30.18585922
	6.89852e-16	30.59516067	12.02656554	12	32.69384127
	7.72546e-16	36.97451388	12.00117943	12	26.99594392
	9.15741e-16	36.01354871	12	12	27.7135641
SGTO	**2.27287e-17**	**33.11543814**	**12**	**19.37586573**	**48.66399625**
	6.09345e-17	39.67469654	12	13.71374923	28.74879932
	1.93234e-16	24.07521196	12	12.00152053	41.46133255
	4.88555e-16	40.90261779	12	16.54622811	33.64534467
	5.20659e-16	31.23611307	12	12	31.95224195

Table 8. Comparison of Statistical Results of the Gear Train Design

	Best	Mean	Std.	Worst
GTO	1.72316e-16	2.15526e-11	6.79982e-11	9.37141e-10
SGTO	2.27287e-17	9.52217e-12	2.74286e-11	3.37508e-10

However, while the results obtained are promising, they do not fully align with the nature of this problem, as the design variables in gear train design must be discrete rather than continuous. For instance, it would not make sense for the number of teeth of a gear to be fractional. To address this, each design variable is rounded to the nearest integer after every iteration, ensuring that the designs are feasible. Table 9 presents the results from these adjusted problems, showcasing the performance of SGTO and GTO when the design variables comply with the nature of the problem. This adjustment provides a more realistic comparison, taking into account the inherent constraints of the problem. Notably, SGTO continues to lead in terms of solution quality, which is also supported by statistical results shown in Table 10.

Table 9. Comparison of the Best Results of the Gear Train Design (Adjusted)

	$f(x)$	x_1	x_2	x_3	x_4
GTO	9.92158e-10	47	13	12	23
	9.92158e-10	23	12	13	47
	9.92158e-10	47	12	13	23
	9.92158e-10	47	12	26	46
	9.92158e-10	47	12	13	23
SGTO	**2.30782e-11**	**53**	**13**	**20**	**34**
	2.30782e-11	34	13	20	53
	9.93988e-11	57	13	31	49
	1.54505e-10	43	13	21	44
	9.92158e-10	47	13	12	23

Table 10. Comparison of Statistical Results of the Gear Train Design (Adjusted)

	Best	Mean	Std.	Worst
GTO	9.92158e-10	3.67909e-08	1.23241e-07	7.77863e-07
SGTO	2.30782e-11	1.48469e-08	4.52698e-08	7.77863e-07

3.2.4 Pressure Vessel Design

The purpose of this problem is to minimize the manufacturing costs of a cylindrical pressure vessel capped by hemispherical heads on both ends. The design variables are the thickness of the shell x_1, thickness of the head x_2, the inner radius x_3, and the length of the cylindrical section of the vessel x_4, depicted in Fig. 9. The objective is to minimize the total costs, which include welding, material, and forming, while also satisfying constraints related to the required internal volume, structural stress limits, and practical design bounds on the vessel dimensions. The bound constraints are set as $1 \times 0.0625 \leq x_1, x_2 \leq 99 \times 0.0625$ and $10 \leq x_3, x_4 \leq 200$. The optimization model of the problem is given as follows.

Minimize:

$$f(X) = 0.06224x_1x_3x_4 + 1.7781x_2{x_3}^2 + 3.1661{x_1}^2x_4 + 19.84{x_1}^2x_3. \quad (24)$$

Subject to:

$$g_1(X) = -x_1 + 0.0193x_3 \leq 0, \quad (25)$$

$$g_2(X) = -x_2 + 0.00954x_3 \leq 0, \quad (26)$$

$$g_3(X) = -\pi{x_3}^2x_4 - \frac{4}{3}\pi{x_3}^3 + 1296000 \leq 0, \quad (27)$$

$$g_4(X) = x_4 - 240 \leq 0. \quad (28)$$

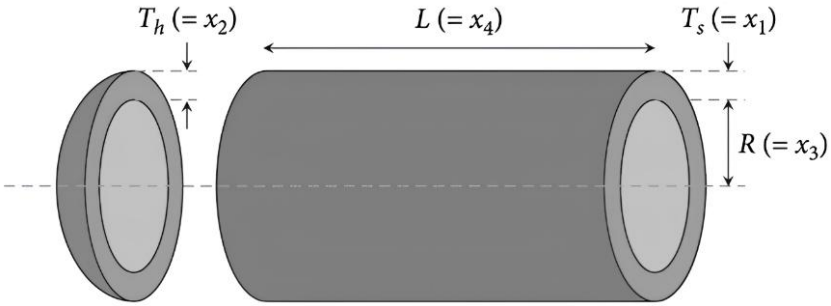


Figure 9. Pressure Vessel Design

Table 11 displays the five best solutions obtained by SGTO and GTO for the pressure vessel design problem. Similarly, SGTO exceeds GTO in terms of finding optimal solutions. Observe that the length of the cylindrical section (x_4) are remarkably different in each of the solutions SGTO provided, thus allowing SGTO to reach a lower objective value. The statistical results of both algorithms are shown in Table 12, which confirms that SGTO consistently produced a more reliable result in achieving a reduction of the pressure vessel manufacturing costs.

Table 11. Comparison of the Best Results of the Pressure Vessel Design

	$f(x)$	x_1	x_2	x_3	x_4
GTO	5913.017322	0.781356548	0.385077476	40.54152241	198.2777099
	5980.2784	0.79316822	0.393010242	41.09569262	192.4257825
	6001.28558	0.783797749	0.422008016	40.9449089	191.7374971
	6006.915316	0.786290719	0.421551683	40.89372303	193.1272677
	6017.331945	0.840533891	0.411123734	43.58954469	159.5219976
SGTO	**5898.752718**	**0.787042751**	**0.390855263**	**41.04821232**	**190.1232608**
	5928.354891	0.780282512	0.382311872	40.71082551	195.5657763
	5932.05526	0.780912511	0.393473847	40.79963814	194.4433666
	5941.481586	0.789391661	0.397240478	40.93157913	192.5303594
	5950.806141	0.797182002	0.387449719	41.17052399	188.7615467

Table 12. Comparison of Statistical Results of the Pressure Vessel Design

	Best	Mean	Std.	Worst
GTO	5913.017322	6806.528085	353.5090536	7441.603852
SGTO	5898.752718	6258.942484	119.1210598	6442.458823

3.2.5 Piston Lever Design

This problem focuses on minimizing the volume of a piston lever mechanism, which is critical for reducing material costs and improving system efficiency. The design variables are the piston height x_1, base width x_2, piston diameter x_3, and stroke length x_4, which is shown in Fig. 10. The objective is to minimize the volume enclosed by the piston, while also satisfying constraints related to the force balance, moment resistance, geometric design feasibility, and mechanical integrity. The bound constraints are set as $0.05 \leq x_1, x_2, x_3 \leq 500$ and $0.05 \leq x_4 \leq 120$. The lever operating angle is $\theta = 45°$, the applied load is $Q = 10000\ lbs$, the total lever length is $L = 240\ in$, the maximum allowable bending moment is $M_{max} = 1.8 \times 10^6\ lbs\ in$, and the oil pressure is $P = 1500\ psi$. The complete optimization model is given as follows.

Minimize:

$$f(X) = \frac{1}{4}\pi {x_3}^2 (L_2 - L_1). \tag{29}$$

Subject to:

$$g_1(X) = QLcos\theta - R \times F \leq 0, \tag{30}$$

$$g_2(x) = Q(L - x_4) - M_{max} \leq 0, \tag{31}$$

$$g_3(x) = 1.2(L_2 - L_1) - L_1 \leq 0, \tag{32}$$

$$g_4(x) = \frac{x_3}{2} - x_2 \leq 0, \tag{33}$$

where,

$$R = \frac{|-x_4(x_4 \sin\theta + x_1) + x_1(x_2 - x_4 \cos\theta)|}{\sqrt{(x_4 - x_2)^2 + {x_1}^2}}, \tag{34}$$

$$F = \frac{\pi P {x_3}^2}{4}, \tag{35}$$

$$L_1 = \sqrt{(x_4 - x_2)^2 + {x_1}^2}, \tag{36}$$

$$L_2 = \sqrt{(x_4 \sin\theta + x_1)^2 + (x_2 - x_4 \cos\theta)^2}. \tag{37}$$

The best obtained results for piston lever design are shown in Table 13. According to the findings, both SGTO and GTO agreed on setting the value of piston height (x_1), but mostly differs in the value of stroke length (x_4). This leads to SGTO being able to reach the lowest possible volume enclosed by the piston. Looking at the statistical results in Table 14, the variability of both algorithms is significant, which is expected due to the implementation of the straightforward death penalty technique. It is worth mentioning here that it is also possible that the number of initial populations is inadequate, considering the given number of design variables. However, this problem can be avoided by setting up a larger initial population.

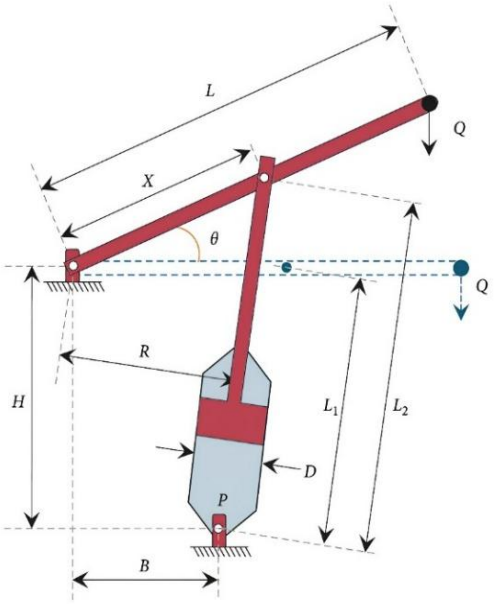


Figure 10. Piston Lever Design

Table 13. Comparison of the Best Results of the Piston Lever Design

	$f(x)$	x_1	x_2	x_3	x_4
GTO	8.921285544	0.05	2.088772581	4.157604232	115.8149417
	9.437447522	0.05	2.201104582	4.169782409	116.037304
	9.490468868	0.05	2.12913255	4.24740004	110.8052399
	9.578598853	0.05	2.139014616	4.257475882	110.5049002
	9.624589647	0.05	2.162282806	4.246069156	112.3216578

	$f(x)$	x_1	x_2	x_3	x_4
SGTO	**8.727356528**	**0.05**	**2.094263341**	**4.107733368**	**118.6226458**
	9.170561899	0.05	2.184519507	4.1260692	118.3761426
	9.258473179	0.05	2.217913681	4.11603762	120
	9.289918569	0.05	2.171350396	4.164422308	116.505379
	9.355160917	0.05	2.127282663	4.219436668	112.9556048

Table 14. Comparison of Statistical Results of the Piston Lever Design

	Best	Mean	Std.	Worst
GTO	8.921285544	152.6466053	242.0550117	1337.639627
SGTO	8.727356528	181.9788569	265.8432291	1162.233957

Considering the execution time, SGTO exhibits slightly higher computational demands in certain cases. While SGTO outperforms GTO in solution quality, the execution time for some of the problems is less efficient, indicating a trade-off between solution quality and computational cost. However, unlike in the first test group on benchmark functions, where SGTO was completely defeated on all terms for six functions listed in Table 2, SGTO was only inferior in either the mean or standard deviation of the computation time. In Table 15 below, the execution time for each of the five engineering design problems is presented, highlighting where SGTO's performance is compromised by longer computation times compared to GTO.

Table 15. Execution Time Results for Engineering Design Problems

Problem	Best		Mean		Standard Deviation	
	GTO	SGTO	GTO	SGTO	GTO	SGTO
A	0.828125	0.671875	1.507916667	0.75078125	0.421405011	0.119971276
B	3.953125	1.234375	**4.94203125**	4.037239583	**0.783599472**	2.736155212
C1	1.15625	1	1.75421875	1.23640625	0.351648288	0.348900567
C2	1.328125	1.046875	1.847864583	1.265572917	0.483685831	0.275654696
D	4.296875	1.4375	6.035104167	5.55984375	**1.900235505**	1.924024105
E	4.21875	4.21875	**5.142083333**	6.177239583	**0.720718867**	0.880936722

** C1: Gear Train Design, C2: Gear Train Design (Adjusted).*

3.3 Results on Epidemiological Model

In this section, we present the application of the proposed SGTO algorithm along with the GTO algorithm to estimate five parameters of the SEAR model in the context of COVID-19 transmission dynamics, as shown in Fig. 4 and Eqs. (12), (13), (14), (15). To approach this estimation task, we convert the system of differential equations into an optimization framework. The objective is to minimize the error in predicting the population dynamics over time, which is calculated by using Eq. (16). The model parameters are treated as design variables, adjusted iteratively to minimize the error, ensuring that the model best fits the observed data and accurately reflects the epidemic's progression.

For comparison and validation purposes, we directly adopt the parameter values obtained by the Genetic Algorithm (GA) as reported in the reference. This result serves as a benchmark for evaluating the performance of GTO and SGTO. By comparing the results obtained from GTO and SGTO with those from GA, we ensure that the parameters estimated by GTO and SGTO are consistent with previous findings, confirming the relevance and accuracy of these new optimization approaches.

Table 16. Comparison of the Best Result of the COVID-19 Model

Parameter	Genetic Algorithm	Giant Trevally Optimizer	
	Mutation Probability (0.125)	Uniform	Sobol
β	0.99102	0.46706746	0.34861697
ε	0.0083116	0.0023645444	0.0024009698
α	0.18169	0.0056193593	9.8876343e-05
γ	0.012122	0.0098240005	0.0099419019
Λ	552.2431	8004.7041	9117.2999

Parameter	Genetic Algorithm	Giant Trevally Optimizer	
	Mutation Probability (0.125)	Uniform	Sobol
$f(x)$	0.1317	0.062077	0.057693

Based on the results in Table 16, it can be observed that both GTO and SGTO effectively minimized the objective function. According to the findings, both GTO and SGTO discover remarkably different parameter values than those estimated by GA, which allows GTO and SGTO to achieve a significantly lower error. It is worth mentioning that the use of Eq. (16) as the objective function provided a clear and quantifiable measure of the model's predictive accuracy.

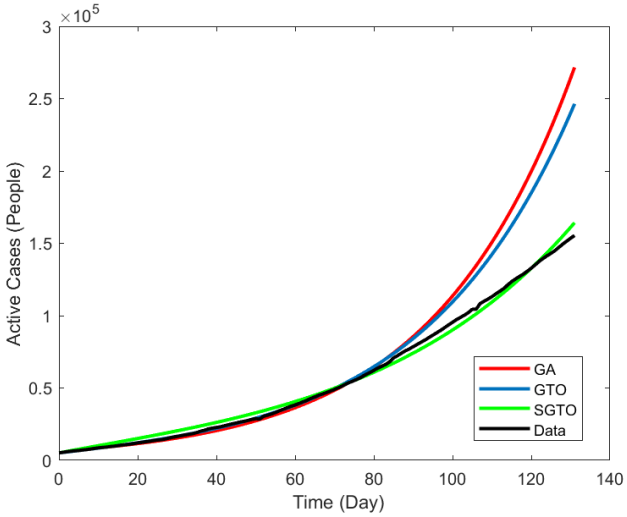


Figure 11. Comparison of Estimated and Actual Active Cases by GA, GTO, and SGTO for the COVID-19 Model

Furthermore, by taking the obtained parameter values to simulate the number of active cases, it is evident from Fig. 11 that the results generated by SGTO are closely aligned with the real data. In contrast, the result obtained by GTO is consistent with those from the GA, though they exhibit a closer fit to the real data compared to the GA estimates. This observation suggests a significant improvement in model performance following the optimization of parameters, highlighting the practicality of SGTO and GTO as valuable tools for calibrating epidemiological models and simulating disease transmission dynamics.

4. CONCLUSION

This paper proposes the Sobol-initialized Giant Trevally Optimizer (SGTO), which leverages low-discrepancy Sobol sequences during initialization to improve global exploration and mitigate premature convergence, addressing a key limitation of conventional metaheuristics. Experimental results consistently show that SGTO outperforms the original GTO in solution quality across diverse benchmarks and engineering problems, with only minor increases in computational time for certain test cases. Since GTO itself was previously validated against standard metaheuristics, the improved performance of SGTO suggests clear advantages in high-dimensional and complex search spaces. Furthermore, the successful application of SGTO in calibrating epidemiological models, such as those used for COVID-19 transmission dynamics, highlights its practical relevance and versatility in addressing real-world optimization challenges. While direct comparisons with other algorithms remain a task for future research, SGTO demonstrates strong real-world applicability and robustness.

Nevertheless, several research gaps remain open for future research. While this study has demonstrated the effectiveness of implementing the Sobol sequence in the initialization phase, further exploration into hybrid strategies that incorporate physics-inspired mechanisms or adaptive search behaviors could yield even more powerful algorithms. Additionally, extending the application of SGTO to combinatorial optimization problems, such as quadratic embedding and domination in graph theory, presents a compelling direction for future work. Finally, investigating alternative low-discrepancy sequences or randomized initialization

methods may further enhance the balance between exploration and exploitation, potentially leading to even more robust and efficient optimization algorithms.

Author Contributions

Ikhsan Rizqi Az-Zukruf As-Shidiq: Conceptualization, Validation, Writing – Original Draft, Writing – Review and Editing. E. Andry Dwi Kurniawan: Methodology, Software, Visualization. Kuntjoro Adji Sidarto: Supervision. All authors discussed the results and contributed to the final manuscript.

Funding Statement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Acknowledgment

The authors gratefully acknowledge the Department of Mathematics, Institut Teknologi Bandung, for providing the computational resources essential for this research. We also thank the reviewers for their valuable comments and suggestions, which contributed to the improvement of this manuscript. Additionally, we extend our appreciation to Aminatus Sa'adah for helpful discussions and support during the development of this work.

Declarations

The authors declare that there are no conflicts of interest in this study.

Declaration of Generative AI and AI-assisted Technologies

Generative AI tools (e.g., ChatGPT) were used solely for language refinement, including grammar, spelling, and clarity. The scientific content, analysis, interpretation, and conclusions were developed entirely by the authors. All final text was reviewed and approved by the authors.

REFERENCES

[1] M. Khadivi, T. Charter, M. Yaghoubi, M. Jalayer, M. Ahang, A. Shojaeinasab, and H. Najjaran, "DEEP REINFORCEMENT LEARNING FOR MACHINE SCHEDULING: METHODOLOGY, THE STATE-OF-THE-ART, AND FUTURE DIRECTIONS," Oct. 2023. doi: https://doi.org/10.1016/j.cie.2025.110856.

[2] G. B. Fotopoulos, P. Popovich, and N. H. Papadopoulos, "REVIEW NON-CONVEX OPTIMIZATION METHOD FOR MACHINE LEARNING," Oct. 2024. doi: 10.48550/arXiv.2410.02017.

[3] F. Jiang, Y. Zhou, J. Liu, and Y. Ma, "ON HIGH DIMENSIONAL POISSON MODELS WITH MEASUREMENT ERROR: HYPOTHESIS TESTING FOR NONLINEAR NONCONVEX OPTIMIZATION," Dec. 2022. doi: https://doi.org/10.1214/22-AOS2248.

[4] X.-S. Yang, *NATURE INSPIRED METAHEURISTIC ALGORITHMS*. U.K.: Luniver Press, 2010.

[5] A. Darwish, "BIO-INSPIRED COMPUTING: ALGORITHMS REVIEW, DEEP ANALYSIS, AND THE SCOPE OF APPLICATIONS," *Future Computing and Informatics Journal*, vol. 3, no. 2, pp. 231–246, Dec. 2018. doi: https://doi.org/10.1016/j.fcij.2018.06.001.

[6] D. Whitley, "A GENETIC ALGORITHM TUTORIAL," Department of Computer Science, Colorado State University, Fort Collins, CO, USA, 1993.

[7] R. Eberhart and J. Kennedy, "A NEW OPTIMIZER USING PARTICLE SWARM THEORY," Aug. 2002. doi: https://doi.org/10.1109/MHS.1995.494215

[8] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "TEACHING-LEARNING-BASED OPTIMIZATION: A NOVEL METHOD FOR CONSTRAINED MECHANICAL DESIGN OPTIMIZATION PROBLEMS," *CAD Computer Aided Design*, vol. 43, no. 3, pp. 303–315, Mar. 2011. doi: https://doi.org/10.1016/j.cad.2010.12.015.

[9] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A GRAVITATIONAL SEARCH ALGORITHM," *Inf Sci (N Y)*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009. doi: https://doi.org/10.1016/j.ins.2009.03.004.

[10] H. T. Sadeeq and A. M. Abdulazeez, "GIANT TREVALLY OPTIMIZER (GTO): A NOVEL METAHEURISTIC ALGORITHM FOR GLOBAL OPTIMIZATION AND CHALLENGING ENGINEERING PROBLEMS," *IEEE Access*, vol. 10, pp. 121615–121640, 2022. doi: https://doi.org/10.1109/ACCESS.2022.3223388.

[11] K. A. Sidarto and A. Kania, "FINDING ALL SOLUTIONS OF SYSTEMS OF NONLINEAR EQUATIONS USING SPIRAL DYNAMICS INSPIRED OPTIMIZATION WITH CLUSTERING," Sept. 2015. doi: https://doi.org/10.20965/jaciii.2015.p0697

[12] D. W. Sims, E. J. Southall, N. E. Humphries, G. C. Hays, C. J. A. Bradshaw, J. W. Pitchford, A. James, M. Z. Ahmed, A. S. Brierley, M. A. Hindell, D. Morritt, M. K. Musyl, D. Righton, E. L. C. Shepard, V. J. Wearmouth, R. P. Wilson, M. J. Witt

and J. D. Metcalfe., "SCALING LAWS OF MARINE PREDATOR SEARCH BEHAVIOUR," *Nature*, vol. 451, no. 7182, pp. 1098–1102, Feb. 2008. doi: https://doi.org/10.1038/nature06518.

[13] N. E. Humphries, N. Queiroz, J. R. M. Dyer, N. G. Pade, M. K. Musyl, K. M. Schaefer, D. W. Fuller, J. M. Brunnschweiler, T. K. Doyle, J. D. R. Houghton, G. C. Hays, C. S. Jones, L. R. Noble, V. J. Wearmouth, E. J. Southall and D. W. Sims., "ENVIRONMENTAL CONTEXT EXPLAINS LEVY AND BROWNIAN MOVEMENT PATTERNS OF MARINE PREDATORS," *Nature*, vol. 465, no. 7301, pp. 1066–1069, Jun. 2010. doi: https://doi.org/10.1038/nature09116.

[14] R.U. Seydel, *TOOLS FOR COMPUTATIONAL FINANCE*. Springer Berlin Heidelberg, 2009. doi: https://doi.org/10.1007/978-3-540-92929-1.

[15] S. Joe and F. Y. Kuo, "CONSTRUCTING SOBOL' SEQUENCES WITH BETTER TWO-DIMENSIONAL PROJECTIONS," *SIAM Journal on Scientific Computing*, vol. 30, no. 5, pp. 2635–2654, 2007. doi: https://doi.org/10.1137/070709359.

[16] Sobol', I. M., "UNIFORMLY DISTRIBUTED SEQUENCES WITH AN ADDITIONAL UNIFORM PROPERTY," *USSR Computational Mathematics and Mathematical Physics*, vol. 16, no. 5, pp. 236-242, 1976. doi: https://doi.org/10.1016/0041-5553(76)90154-3.

[17] M. Jamil and X. S. Yang, "A LITERATURE SURVEY OF BENCHMARK FUNCTIONS FOR GLOBAL OPTIMISATION PROBLEMS," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, no. 2, pp. 150–194, 2013. doi: https://doi.org/10.1504/IJMMNO.2013.055204.

[18] E. P. Adorio, "MVF-MULTIVARIATE TEST FUNCTIONS LIBRARY IN C FOR UNCONSTRAINED GLOBAL OPTIMIZATION." [Online]. Available: http://www.mat.univie.ac.at/

[19] W. Zhao, L. Wang, and S. Mirjalili, "ARTIFICIAL HUMMINGBIRD ALGORITHM: A NEW BIO-INSPIRED OPTIMIZER WITH ITS ENGINEERING APPLICATIONS," *Comput Methods Appl Mech Eng*, vol. 388, Jan. 2022. doi: https://doi.org/10.1016/j.cma.2021.114194.

[20] A. H. Gandomi, X. S. Yang, and A. H. Alavi, "ERRATUM: CUCKOO SEARCH ALGORITHM: A METAHEURISTIC APPROACH TO SOLVE STRUCTURAL OPTIMIZATION PROBLEMS," Apr. 2013. doi: https://doi.org/10.1007/s00366-012-0308-4.

[21] 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," Mar. 2014. doi: https://doi.org/10.4018/jdsst.2013040103.

[22] H. Bayzidi, S. Talatahari, M. Saraee, and C. P. Lamarche, "SOCIAL NETWORK SEARCH FOR SOLVING ENGINEERING OPTIMIZATION PROBLEMS," *Comput Intell Neurosci*, vol. 2021, Sept. 2021. doi: https://doi.org/10.1155/2021/8548639.

[23] R. Zheng, A. G. Hussien, H. M. Jia, L. Abualigah, S. Wang, and D. Wu, "AN IMPROVED WILD HORSE OPTIMIZER FOR SOLVING OPTIMIZATION PROBLEMS," *Mathematics*, vol. 10, no. 8, Apr. 2022. doi: https://doi.org/10.3390/math10081311.

[24] B. S. Yildiz, P. Mehta, N. Panagant, S. Mirjalili, and A. R. Yildiz, "A NOVEL CHAOTIC RUNGE KUTTA OPTIMIZATION ALGORITHM FOR SOLVING CONSTRAINED ENGINEERING PROBLEMS," *J Comput Des Eng*, vol. 9, no. 6, pp. 2452–2465, Dec. 2022. doi: https://doi.org/10.1093/jcde/qwac113.

[25] E. A. D. Kurniawan, F. Fatmawati, and A. Dianpermatasari, "MODEL MATEMATIKA SEAR DENGAN MEMPERHATIKAN FAKTOR MIGRASI TERINFEKSI UNTUK KASUS COVID-19 DI INDONESIA," *Limits: Journal of Mathematics and Its Applications*, vol. 18, no. 2, p. 142, Nov. 2021. doi: https://doi.org/10.12962/limits.v18i2.7774.

[26] P. Bratley and B. L. Fox, "Algorithm 659: IMPLEMENTING SOBOL'S QUASIRANDOM SEQUENCE GENERATOR," *ACM Transactions on Mathematical Software*, vol. 14, no.1, pp. 88-100, 1988. doi: https://doi.org/10.1145/42288.42289

[27] Antonov, I. A., and Saleev, V. M., "AN ECONOMIC METHOD OF COMPUTING LPT-SEQUENCES," *USSR Computational Mathematics and Mathematical Physics*, vol. 19, no. 1, pp. 252-256, 1979. doi: https://doi.org/10.1016/0041-5553(79)90085-5

[28] C. A. C. Coello, "THEORETICAL AND NUMERICAL CONSTRAINT-HANDLING TECHNIQUES USED WITH EVOLUTIONARY ALGORITHMS: A SURVEY OF THE STATE OF THE ART," *Comput. Methods Appl. Mech. Eng.* 191(11-12), 1245-1287, Jan. 2002. doi: https://doi.org/10.1016/S0045-7825(01)00323-1

[29] C. A. C. Coello, "A SURVEY OF CONSTRAINT HANDLING TECHNIQUES USED WITH EVOLUTIONARY ALGORITHMS," Laboratorio Nacional de Informática Avanzada, Veracruz, Mexico, 1999.