

Article

A Systematic Method for Generating Discrete Multidimensional Thomae-like Functions

Andrei Pătrăușanu *, Adrian Florea, Mihai Neghină, Radu Chiș and Alina Dicoiu

Department of Computer Science and Electrical Engineering, Lucian Blaga University of Sibiu, 4 Emil Cioran Str., 550025 Sibiu, Romania; adrian.florea@ulbsibiu.ro (A.F.); mihai.neghina@ulbsibiu.ro (M.N.); radu.chis@ulbsibiu.ro (R.C.); alina.dicoiu@ulbsibiu.ro (A.D.)

* Correspondence: andrei.patrasanu@ulbsibiu.ro

Abstract

Generating multidimensional rugged landscapes is very important for testing and benchmarking, considering the increasing number of multi-objective optimization problems and the pervasive use of computational systems. The diverse nature of such problems has led to a variety of algorithmic solutions, designed to create more scalable and problem-independent algorithms. This paper introduces a method for constructing a class of discrete multidimensional Thomae-like functions, generalizing the classical Thomae function. While the original function is continuous at irrational points and discontinuous at rational ones, we extend this approach to higher-dimensional discrete spaces. We present the properties of three known analytic generalizations and analyze five proposed discrete cases according to the relationship between the numbers of input and output values. The most important properties of such functions are discussed, including their periodicity, number of local maxima, autocorrelation, and output variations under small input changes. The last property has significant importance for simulating and testing in various application areas. This work unifies and extends existing concepts into a systematic method for generating discrete multidimensional functions, opening new directions for both theoretical investigations and practical applications of algorithms in engineering.

Keywords: Thomae; function; discrete; multidimensional; engineering

1. Introduction

At present, the study of complex multi-objective functions plays a central role in the development, testing, and benchmarking of state-of-the-art multi-objective optimization algorithms. This is amplified by the growing number of optimization problems arising from diverse fields. In medical health areas, categories of optimization problems include proper dosing in chemotherapy [1] and optimization of waiting time and medical services [2]. Other examples include digital twin systems [3,4] for geotechnic research or management of parking lots. Additional examples can also be found in engineering fields, including energy consumption [5] and construction site management [6]. Such diverse real-world optimization problems are highly complex, where the constraints of many objectives that are usually conflicting must be considered.

To address the growth of optimization problems and their corresponding parameters, design variables, and objectives, various multi-objective algorithmic solutions have been proposed, including non-dominated sorting-based algorithms such as the Non-

Academic Editor: George Angelos Papadopoulos

Received: 12 June 2026

Revised: 16 July 2026

Accepted: 22 July 2026

Published: 25 July 2026

Copyright: © 2026 by the author.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Dominated Sorting Algorithm II (NSGA-II) [7], Non-Dominated Sorting Algorithm III (NSGA-III) [8], swarm-based Hyperplane-Assisted Multi-objective Particle Swarm Optimization (Hyperplane-Assisted MOPSO) [9], and Red Deer Algorithm (RDA) [10]. Each algorithm is designed based on particular principles, which offer distinct strengths that can guide the search towards the optimal solutions. However, no algorithms are superior to others across all problem types, as stated by the No Free Lunch theorem.

Consequently, there is a need to generate representative landscapes that encapsulate features such as multidimensionality, discontinuity, and sensitivity to input perturbations. These characteristics are well represented in real-world optimization problems. In particular, rugged multidimensional landscapes are essential to evaluate the robustness and resilience of algorithms against contemporary optimization systems. Therefore, generating benchmark and objective functions that emulate complex real-world optimization scenarios is crucial.

A classic example of such a function is the Thomae function, also known as the popcorn or raindrop function. As originally defined in 1875 by Karl Johannes Thomae [11], it is a pathological function, discontinuous at every rational number, but continuous at every irrational one (Equation (1)). It was built on the foundation of the well-known Dirichlet function, a nowhere continuous function. Due to its pathological nature, the Thomae function has been widely adopted and used in real-world analyses, as an example of a function that defies continuity properties. It exhibits high sensitivity to small input perturbations, making it ideal for generating challenging landscapes for optimization algorithms.

The Thomae function has well-known analytic generalization but, considering the need for its digital representation in computer systems, a scalable discrete generalization scheme is needed. This study proposes a method for generating multidimensional landscapes as multivalued functions like the original Thomae function, for which small changes in the input can lead to dramatic changes in the output. The proposed method is intended to be exploited in future research as an adversarial fitness function for evolutionary algorithms, whose paradigm is based on asymptotic convergence over successive generations. At later stages, minor changes in the design variables are likely to result in minor changes in the objective space. Therefore, the proposed framework can be further used and developed as a benchmark to test the strengths and limitations of multi-objective optimization algorithms in the context of various optimization problems. It could also constitute an initial step toward constructing a broader set of adversarial benchmarks, aimed at demonstrating the limitations of current state-of-the-art algorithms.

This work is part of an ongoing PhD thesis entitled “*The development of FADSE framework with new evolutionary algorithms for solving modelling, optimization, and simulation multi-objective optimization problems in various engineering domains*”. The Framework for Advanced Design Space Exploration (FADSE) [12] is an open-source tool that is developed and extended during the current thesis, which facilitates the research and testing of evolutionary algorithms. It incorporates classic state-of-the-art algorithms, problems, quality indicators and newly introduced methodologies. As a continuation of the current work, one of the next objectives for the FADSE is the integration of Thomae-like functions into its optimization problem library. This library will be extended to support a flexible and straightforward architecture for implementing, analyzing and benchmarking the proposed Thomae-like function. Although the mathematical formulations of these functions are independent of their implementation in the FADSE, the framework’s open-source object-oriented structure is designed to facilitate extensibility, enabling researchers to easily incorporate further proposed families of Thomae-like functions as well as other classes of problems.

This work started with the following two research questions: “What benchmarks are used in multi-objective evolutionary algorithms and frameworks and why?” and “Could Thomae-like functions be used for testing and evaluating multi-objective evolutionary algorithms and frameworks?” Some of the well-known benchmark suites are DTLZ, ZDT [13,14], WFG [15], and CEC [16]. The ZDT sets are typically used for two-objective tests, while DTLZ is suited for multi-objective tests. WFG is considered a more up-to-date benchmark than DTLZ/ZDT for multi/many-objective optimizations. These benchmark suites are computationally simple but exhibit various barriers such as convexity, concavity, discontinuity, multiple fronts (testing the capacity of an algorithm to not be stuck in a local optimum), and convergence to the global optimum. The CEC suite focuses on metrics for evaluating the diversity (spread), scalability, robustness, and convergence of algorithms. These benchmarks exhibit only smooth variations in terms of correlation between the input and output of the function. Currently, Thomae-like functions are not part of these state-of-the-art benchmarks. They could be used as adversarial functions for evolutionary algorithms to test their resilience against large output variations produced by minor input changes. The majority of benchmarks are based on continuity assumptions, while Thomae-like functions can test an algorithm’s behavior in response to certain spikes (as shown in later sections) in the objective space. Furthermore, gradient-based algorithms face significant challenges from a Thomae-like function, as such a function is nowhere differentiable and, hence, the gradient cannot be computed. The impact of such a function over a typical Pareto front would be interesting to study. It might result in a fragmented front, consisting of isolated points, instead of a smooth curve. Moreover, classical quality indicators such as hypervolume may not capture the irregular behavior introduced by such a function, making the implementation of updated metrics and indicators necessary.

The remainder of the paper is structured as follows: Section 2 presents similar related work from the literature, while the definition of the Thomae function is presented in Section 3. Section 4 highlights the well-known continuous generalizations of the Thomae function, Section 5 proposes and discusses five cases of discrete generalizations according to the number of input and output values, and Section 6 presents a discussion of five aspects; namely, the number of maxima, number and statistics of jumps, autocorrelation, numerical integral, and spectrum. The Section 7 concludes the paper by highlighting directions for future work.

2. Related Work

The construction of rugged and complex landscapes has been studied in several research domains, including fractal analysis, procedural modeling, and evolutionary computing benchmarking. One approach is based on the Weierstrass function [17], which is continuous everywhere but differentiable nowhere. Its properties, including differentiability, have been thoroughly analyzed in [18]. Due to its nature, the Weierstrass function has been widely adopted and used in signal processing (irregular signals) and fractal geometry. Properties such as periodicity and self-similar behavior make this function suitable for studying scalable multidimensional landscapes. However, it is still a continuous function, meaning that small perturbations of the input produce bounded variations in the output.

There are some well-known fractal-based approaches, such as the Mandelbrot set [19], which exhibits a self-similar structure and is used to model natural phenomena. Other approaches include fractal interpolation functions, such as that described in [20], where the synergy of the Lagrange basis and Fourier transform may also generate oscillatory functions, with applications in various domains, including approximation theory. As a consequence of their design, such models preserve continuity and limit the analysis of great discontinuities. Another example is the Katsura [21] function, which is a continuous

but nowhere differentiable fractal-like function. This function has been adopted in benchmark suites such as the COCO/BBOB [22], which has 24 noise-free, real-parameter, single-objective benchmark functions for testing and analyzing optimization algorithms.

Another important category of methods is based on noise analyses, particularly Perlin noise [23], which was originally designed for computer graphics technology. The smooth landscapes generated using this approach have been adopted and used in terrain generation, simulation systems, and real-world environment modeling. A focused analysis of this approach has been presented in [24], highlighting its applications and properties. Besides their complexity, the functions used in noise analyses remain inherently continuous and smooth.

Related surveys, such as [25], have reviewed existing benchmarks according to their strengths, applicability, and limitations. New benchmark suites are proposed in [26,27], introducing 30 so-called multidimensional ‘hard benchmark’ functions for global optimization. These functions are also intended to challenge optimization algorithms emphasizing high-dimensionality landscapes. In [28], the authors presented a recent review of 315 benchmarks and test functions for optimization algorithms and machine learning. Moreover, recent approaches, such as those described in [29,30], use genetic programming principles to develop new hard benchmarks, indicating that the existing paradigm of benchmarking may not be challenging enough for modern state-of-the-art algorithms.

Standardized sets of test problems and benchmarks, such as the ZDT and DTLZ suites in [13,14], are still widely used in multi-objective optimization. The Pareto fronts provided in these suites are typically used to compare the convergence, diversity, and coverage of various algorithms. Despite this, the existing functions do not capture high discrete variations.

Closer to the present work are other studies focusing on the Thomae function. In [31], the authors discussed modifications of the Thomae function’s differentiability, and their investigation revealed structural changes that influence its regularity. Classic theoretical concerns regarding the integration of highly irregular functions have been discussed in [32]. Other works, such as [33–35], have used the properties of Thomae-like functions to build complex numeric systems such as the pendulum wave, wherein the analytical solution is shown to be a subset of the Thomae function.

Despite the multitude and richness of test functions and benchmarks, most frameworks remain continuous or exhibit controlled irregularity. Consequently, small changes in the input lead to controlled or bounded variations in the output, a property that can be exploited by optimization algorithms. Yet, discrete output with great variations produced by small input changes are considerably less explored. This work addresses this research gap by proposing a framework for generating and testing discrete Thomae-like functions and their feasibility as synthetic benchmarks in the FADSE framework.

3. The Thomae Function

Also known as the popcorn or raindrop function, the Thomae function is defined as

$$f: \mathbb{R} \rightarrow [0,1], \quad f(x) = \begin{cases} 1, & x = 0 \\ \frac{1}{q}, & x = \frac{p}{q}, \\ 0, & x \in \mathbb{R} \setminus \mathbb{Q} \end{cases} \quad \text{with } p \in \mathbb{Z}^*, q \in \mathbb{N} \text{ and coprime} \quad (1)$$

and exhibits the following well-known properties.

- A. Periodicity: It is periodic with a period of 1.
- B. It is discontinuous at every rational number, but continuous at every irrational number.

The Lipschitz/Hölder condition is not satisfied:

$$(\forall) C, \alpha > 0, (\exists) x \in \left(0, \left(\frac{1}{C}\right)^{1/\alpha}\right) \cap \mathbb{R} \setminus \mathbb{Q} \text{ so that } C \cdot |x - 0|^\alpha < |f(x) - f(0)| = 1$$

- C. As demonstrated in [31], it is continuous but not differentiable at irrationals. Along the sequences of irrationals, the difference quotient tends to zero, but suitable sequences of rational approximations with bounded denominators prevent the existence of the derivative. Following its discontinuity at every rational point, the function is nowhere differentiable.
- D. The set of non-zero producing inputs/spikes ($\mathbb{Q}$) is countably infinite and dense. Since each spike is a local maximum, the set of local extrema is also countably infinite and dense, conveying ruggedness in a topological sense.
- E. It can arbitrarily generate low non-zero values within any given interval, due to the dense nature of the rationals:

$$(\forall) x_0 \in \mathbb{R}, (\forall) \delta > 0, (\forall) \varepsilon > 0, (\exists) \xi \in (x_0, x_0 + \delta) \cap \mathbb{Q}, \text{ so that } f(\xi) < \varepsilon$$

- F. Within any interval, the function can be shifted so that the autocorrelation integral can be computed and drops to 0:

$$(\forall) \delta > 0, (\exists) \xi \in \mathbb{R} \setminus \mathbb{Q}, 0 < \xi < \delta, \text{ so that:}$$

$$(\forall) x \in \mathbb{R} \setminus \mathbb{Q} \Rightarrow f(x) = 0 \Rightarrow f(x) \cdot f(x + \xi) = 0$$

$$(\forall) x \in \mathbb{Q}, (\forall) \xi \in \mathbb{R} \setminus \mathbb{Q} \Rightarrow x + \xi \in \mathbb{R} \setminus \mathbb{Q} \Rightarrow f(x + \xi) = 0 \Rightarrow f(x) \cdot f(x + \xi) = 0$$

$$\int f(x) \cdot f(x + \xi) dx = \int 0 dx = 0$$

4. Known Analytic Generalizations

4.1. Generalizations of Type $f: \mathbb{R} \rightarrow \mathbb{R}$

The Thomae function is itself known as a modified Dirichlet function, obtained by changing the unitary values at rational positions. The first obvious generalization is thus replacing those values with other decreasing functions, as shown in Equation (2). Another generalization would be to replace $\mathbb{Q}$ with another dense countable set, such as a set of algebraic numbers $\mathbb{A}$, as represented in Equation (3). To preserve continuity over $\mathbb{R} \setminus \mathbb{A}$, $g(x)$ would also need to have certain limit properties not mentioned in Equation (3). When extending the notion of a function to the special distributions used in signal processing, the generalization could include Dirac distributions $\delta(x)$, as shown in Equation (4).

$$f(x) = \begin{cases} g(q), & x \in \mathbb{Q} \\ 0, & x \in \mathbb{R} \setminus \mathbb{Q} \end{cases} \quad (2)$$

where $\lim_{q \rightarrow \infty} g(q) = 0$

$$f(x) = \begin{cases} g(x), & x \in \mathbb{A} \\ 0, & x \in \mathbb{R} \setminus \mathbb{A} \end{cases} \quad (3)$$

$$f(x) = \sum_{x_0 \in \mathbb{A}} g(x_0) \cdot \delta(x - x_0) \quad (4)$$

4.2. Generalizations of Type $f: \mathbb{R}^n \rightarrow \mathbb{R}$

Considering functions with multiple input values and one output value, any aggregation of the inputs can be conceivably used for transformation into a 1D function. The examples in Equations (5)–(7) show the use of least common multiple (lcm) on the denom-

inators of the rational inputs and the use of common aggregation techniques such as the sum and product of inputs.

$$f(x_1, \dots, x_n) = \begin{cases} \frac{1}{q}, & x_k = \frac{p_k}{q_k} \in \mathbb{Q} \text{ and } q = \text{lcm}(q_1, \dots, q_n) \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

$$f(x_1, \dots, x_n) = \begin{cases} \frac{1}{q}, & \sum_k x_k = \frac{p}{q} \in \mathbb{Q} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

$$f(x_1, \dots, x_n) = \begin{cases} \frac{1}{q}, & \prod_k x_k = \frac{p}{q} \in \mathbb{Q} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

4.3. Generalizations of Type $f: \mathbb{R} \rightarrow \mathbb{R}^m$

To generate multiple relatively independent outputs from one input, the input would need to be split into m virtual inputs, each applied to a Thomae-like function $f: \mathbb{R} \rightarrow \mathbb{R}$.

For the case of $m = 2$, the simplest method is to expand the input value into its binary representation, separating the odd and even digits (including decimals). For the general case, the binary digits are separated according to their position *modulo* m .

More visually appealing is the use of space-filling curves, such as the Hilbert curve represented in two and three dimensions in Figure 1. Considering the path of length 1, the m virtual values would be the coordinates of the input in the higher-dimensional space. While also a binary operation in disguise, the use of space-filling curves has the advantage of preserving locality in the sense that close input values generate closely positioned vectors of m values.

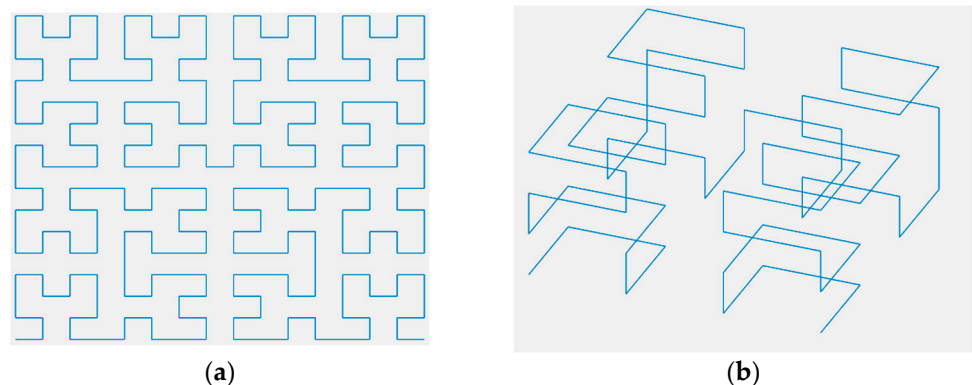


Figure 1. Examples of Hilbert curves: (a) 2D and (b) 3D.

5. Discrete Thomae-like Functions

The need for digital versions of Thomae-like functions (discrete in both the definition and value domains) arises from the pervasiveness of computers in science and engineering. While not all properties can be preserved due to the inherent limitations of discrete functions, the most important properties will be present in the versions discussed in this section, including periodicity, an absolute maximum and many local maxima in each period, large jumps at many coordinates, a general baseline of low values (not quite 0 since computers do not operate with true irrationals), and a sharply decreasing autocorrelation. The irregular behavior of Thomae-like functions resembles that of certain optimization problems, such as power distribution in electric networks. In electric networks, a minor perturbation—such as a perturbation to a single electric line due to a fallen tree—could massively damage the functionality of the rest of the network, as other lines risk being overloaded. If the critical capacity limit is exceeded by only 1%, the consequences could

be catastrophic, and the system's state may rapidly change from stable to large-scale blackout.

For conciseness, the names and definitions of the variables in the discrete functions are presented below:

$x_k, k = 1..n$, input values, n , number of inputs

$o_k, k = 1..m$, output values, m , number of outputs

$S_k, k = 1..m$, additive bias/seed

$M_k, k = 1..m$, multiplicative bias, the average number of global maxima per unit interval

$Q_k, k = 1..m$, maximum allowed denominator

The proposed family of discrete functions $f_d(x)$ without additive or multiplicative bias ($S = 0, M = 1$), as described in Equation (8), approximate the continuous Thomae function $f_c(x)$ in the sense that, when restricting the definition domain to the rational numbers, the discrete functions converge uniformly to the Thomae function when the maximum allowed denominator Q tends to infinity:

$$f_c: \mathbb{Q} \rightarrow \mathbb{Q}, f_c(x) = \frac{1}{q}, \text{ for } x = \frac{p}{q},$$

$$f_d: \mathbb{Q} \rightarrow \mathbb{Q}, f_d(x) = \frac{1}{q}, \text{ where } q = \text{index} \left(\min_{k=1..Q} |k \cdot x - [k \cdot x]| \right) \quad (8)$$

$$\text{then } \lim_{Q \rightarrow \infty} f_d(x) = f_c(x)$$

The proposed discrete construction is closely related to bounded-denominator Diophantine approximation since, as shown in Equation (9), each sample is associated with a nearby rational whose denominator determines the assigned amplitude. Moreover, although Farey sequences are not explicitly used, the bounded-denominator rational candidates may coincide with the fractions appearing in a Farey sequence of order Q .

After transitioning to the discrete version of the function, some of the properties presented in Section 3 are altered:

- A. Periodicity: For suitable values of F_s , its periodicity remains unchanged, with a period of 1. For irrational values of F_s , the function becomes pseudo-periodic, because an irrational value will not preserve the exact same period in a rigorous mathematical sense. Nevertheless, the periodic behavior of the function remains unchanged, within certain approximation.
- B. The continuity is inherently lost during discretization.
- C. Differentiability is inherently lost during discretization.
- D. For rational F_s , all inputs produce non-zero outputs. A detailed analysis is outlined in Section 6.
- E. The output values are constrained by the maximum denominator Q .
- F. Like the analytic Thomae function, the autocorrelation drops quickly, especially noticeable for higher F_s . A detailed analysis is presented in Section 6.

Table 1 presents a summary of the properties' status after the discretization process.

Table 1. Status of Thomae properties after discretization.

Prop.	Status	Discrete Version
	Periodicity	preserved *
	Continuity	lost
	Differentiability	lost
	No. of maxima	affected
	Autocorrelation	affected

* For suitable sampling frequency F_s .

Sections 5.1–5.5 discuss all possible cases according to the relationship between the number of inputs and the number of outputs.

5.1. Base Case $n = m = 1$

The base case for the proposed discrete Thomae-like functions attempts to compute the closest fractional approximation of the input with a denominator lower than or equal to Q_1 . The output can be computed as shown in Equation (9):

$$q = \text{index} \left(\min_{k=1..Q_1} \left| k \cdot (x_1 - S_1) - \frac{[k \cdot M_1 \cdot (x_1 - S_1)]}{M_1} \right| \right), \quad o_1 = \frac{1}{q} \quad (9)$$

where $[\blacksquare]$ represents rounding to the closest integer. If there are multiple denominators with the same minimum value, the one corresponding to the minimum index value is selected. It should be noted that selecting the smallest, largest, or first denominator will produce different landscapes. Figure 2 shows the output of the function with default parameters $S_1 = 0$, $M_1 = 1$, and $Q_1 = 10^3$ when the input is varied from 0 to 1 with a step of 10^{-5} , as well as a zoom around $x_1 = 0.5$. The effects of the parameters could be summarized as follows:

- The maximum denominator Q_1 affects the output values. Larger Q_1 values allow for lower output values, being closer to the analytic Thomae function.
- The average number of global maxima per unit interval M_1 controls periodicity by stretching or shrinking the base function. M_1 needs not be an integer, but it may change the functions' outputs beyond stretching and shrinking.

The seed S_1 introduces an additive bias, shifting the periodic global maximum by the same amount. Seeds equal to multiples of one period would have no effect, while negative seeds or seeds larger than one period could be reduced to positive values that are lower than the period without adverse effects. Similarly to M_1 , non-integer values of S_1 may produce slight differences in the output, but this will not cause a major change to the desired core (behavior) of such families of functions. For suitable values for M_1 and S_1 ($M_1 = 3$, $S_1 = 0.5$), the interpretation of stretching and shifting with preservation of the number of maxima is mathematically valid.

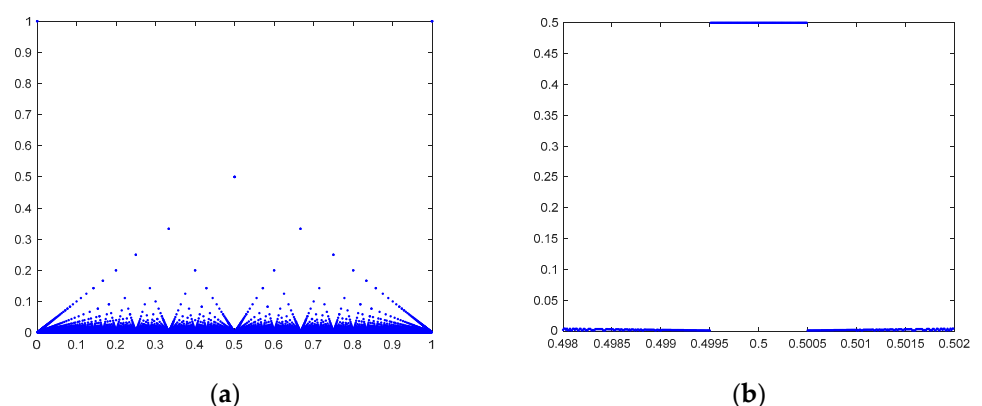


Figure 2. Discrete Thomae-like function ($m = n = 1$) for various input value intervals: (a) $[0, 1]$ and (b) $[0.498, 0.502]$.

Figure 3 shows the effects of multiplicative and additive bias.

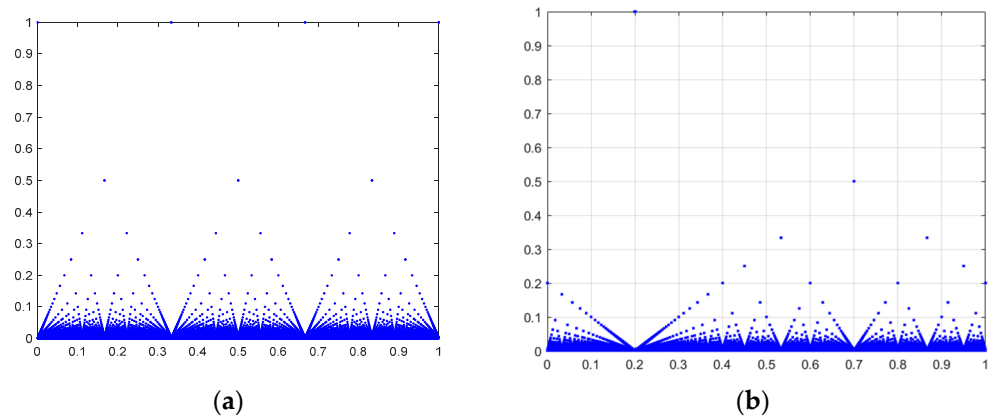


Figure 3. Discrete Thomae-like function ($m = n = 1$) for various bias parameters: (a) $S_1 = 0$ and $M_1 = 3$; (b) $S_1 = 0.2$ and $M_1 = 1$.

5.2. Iterated Case $n = m = 1$

In the base case, a lot of values around any maximum would generate the same output, which is a limitation introduced by Q . A more refined version could compute the output as in Equation (10):

$$\begin{aligned} q_1 &= \text{index} \left(\min_{k=1..Q_1} \left| k \cdot (x_1 - S_1) - \frac{[k \cdot M_1 \cdot (x_1 - S_1)]}{M_1} \right| \right), \\ q_2 &= \text{index} \left(\min_{k=1..Q_1} \left| k \cdot Q_1 \cdot (x_1 - S_1) - \frac{[k \cdot Q_1 \cdot M_1 \cdot (x_1 - S_1)]}{M_1} \right| \right), \\ o_1 &= \frac{2}{q_1 + q_2} \end{aligned} \quad (10)$$

By multiplying with Q_1 , the computation evaluates the “next” decimals of the input. This procedure could be extended to any number of iterations, but even two iterations are sufficient to ensure that the input is extremely precise for generating the local maxima. Figure 4 shows the output of the function with two iterations, where the input varies from 0 to 1 with a step of 10^{-5} , as well as a zoom around $x_1 = 0.5$, for the default parameters. Similar real-world optimization problems can be observed in fields such as meteorology, where extremely small errors can lead to significantly different forecasts. For example, a measurement error of only 10^{-4} might lead to a totally different weather prediction for the next 7 days. This research area is typically described using non-linear systems of differential equations. This could also be considered equivalent to a large variation in the fitness of an individual after slightly mutating a gene using a genetic algorithm.

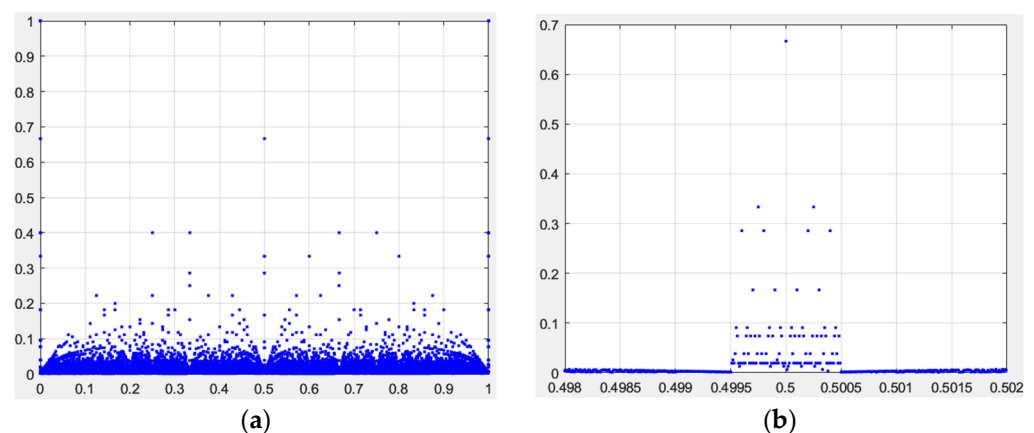


Figure 4. Refined Thomae-like function ($m = n = 1$) for various input value intervals: (a) $[0, 1]$ and (b) $[0.498, 0.502]$.

Although not producing exactly the same output values, the refined case with i iterations could be assimilated to the base case with a higher maximum allowed denominator; that is, $Q_{iterated} = Q_{base}^i$, especially for Q values expressed as powers of 10.

To simplify the explanation for the following cases, the procedure of generating one output o_k from one input x_k will be denoted as H_k , as shown in Equation (11). The index distinction of H_k is necessary to allow for different parameters along different dimensions.

$$o_k = H_k(x_k) \quad (11)$$

5.3. Extended Case $n = m > 1$

In this case, each input generates an output with Equation (11) independently of the other inputs. However, the parameters for each output dimension may differ (e.g., $S_k \neq S_l, M_k \neq M_l, Q_k \neq Q_l$), so that two identical values x_k and x_l may generate different outputs along different dimensions due to these different parameters. Figure 5 shows all possible output combinations for a discrete Thomae-like function with parameters $m = 2, Q_1 = Q_2 = 10^2$. The precise input values needed to reach any point in the output space are determined by the bias parameters S and M .

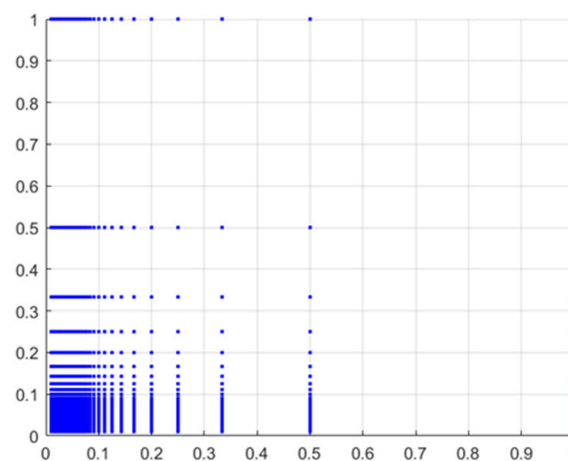


Figure 5. Values of the Thomae-like function ($m = n = 2$) in the output space.

More generally, the inputs x_k can be combined into virtual inputs j_k via linear or non-linear methods, as described in Section 5.4, before computing the outputs.

5.4. Asymmetric Case $n > m$

If there are more inputs than outputs, the inputs must be combined to create a number of virtual inputs $j_1 \dots j_m$ that matches the number of outputs, followed by the generation of the outputs from the virtual inputs, similarly to Equation (11). Linear combinations can be described in a matrix form, with the use of an $m \times n$ matrix of coefficients G :

$$\begin{bmatrix} j_1 \\ \vdots \\ j_m \end{bmatrix} = G_{m \times n} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad o_k = H_k(j_k)$$

Table 2 shows several examples of the matrices and explicit linear combinations of inputs into virtual inputs. Matrices that allow any input x_i to influence only one output j_k , such as those of sequential or cyclic partitioning, would essentially split the multidimensional optimization problem into m trivial one-dimensional problems.

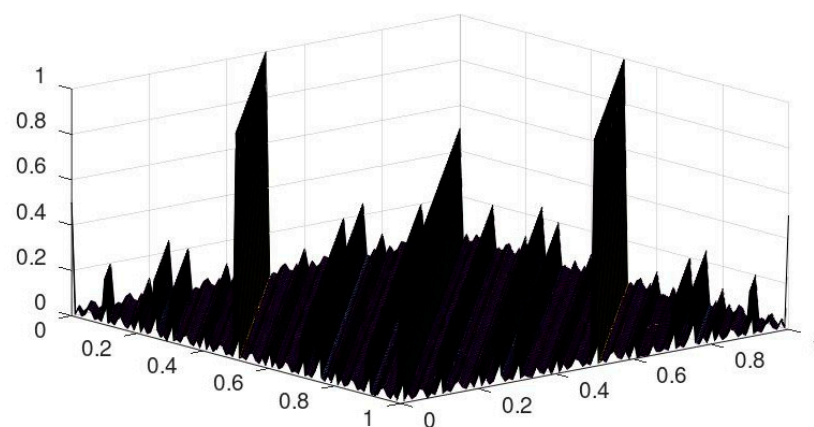
Table 2. Examples of linear combinations of 7 inputs $\{x_i\}$ into 3 virtual inputs $\{j_k\}$.

Matrix of Coefficients G	Explicit Linear Combinations
Balanced sequential partitioning $G = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \end{bmatrix}$	$j_1 = \frac{x_1 + x_2}{2}, \quad j_3 = \frac{x_6 + x_7}{2}$ $j_2 = \frac{x_3 + x_4 + x_5}{3}$
Cyclic partitioning $G = \begin{bmatrix} 1/3 & 0 & 0 & 1/3 & 0 & 0 & 1/3 \\ 0 & 1/2 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 1/2 & 0 \end{bmatrix}$	$j_1 = \frac{x_1 + x_4 + x_7}{3}, \quad j_3 = \frac{x_3 + x_6}{2}$ $j_2 = \frac{x_2 + x_5}{2}$
Minimally overlapping linear combinations $G = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$	$j_1 = x_1 + x_2 + x_3, \quad j_3 = x_5 + x_6 + x_7$ $j_2 = x_3 + x_4 + x_5$
General case $G = \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} & g_{15} & g_{16} & g_{17} \\ g_{21} & g_{22} & g_{23} & g_{24} & g_{25} & g_{26} & g_{27} \\ g_{31} & g_{32} & g_{33} & g_{34} & g_{35} & g_{36} & g_{37} \end{bmatrix}$	$j_k = \sum_{i=1}^7 g_{ki} \cdot x_i$

Non-linear methods for combining the inputs x_k into virtual inputs j_k with the help of aggregation functions g_k may also be chosen to generate discrete Thomae-like functions.

$$j_k = g_k(x_1, \dots, x_n), \quad o_k = H_k(j_k)$$

Other methods of combining the inputs x_k into virtual inputs j_k may be chosen depending on the particularities of the problem at hand that requires discrete Thomae-like functions. The generalizations presented in Table 2 have been chosen for the current work. Other approaches and mappings are admissible, in the sense that they will provide a discrete Thomae-like function for the corresponding case. Figure 6 shows the output for a discrete Thomae-like function with $m = 1, n = 2, S_1 = 0.25, M_1 = 2, Q_1 = 10^4$.

**Figure 6.** Asymmetric Thomae-like function ($n = 2, m = 1$) with multiplicative bias $S_1 = 0.25, M_1 = 2, Q = 10^4$.

5.5. Asymmetric Case $n < m$

If there are more outputs than inputs, the inputs need to be split to create a number of virtual inputs $j_1 \dots j_m$ that matches the number of outputs. One method is to compute the number of outputs that need to be generated by any input, as uniformly as possible, and to compute the virtual inputs as coordinates of space-filling curves. The example below considers two inputs and three outputs.

(j_1, j_2) are the coordinates of x_1 on the 2D Hilbert curve, and $j_3 = x_2$.

$$o_k = H_k(j_k) \text{ for } k = 1 \dots 3$$

While it is not perfect, a uniform distribution of input values x would generate a relatively uniform multi-dimensional distribution of coordinates at the fractional position of x on Hilbert curves. Averaging may change the distribution of the inputs, while finite-order Hilbert mappings introduce quantization, multiplicity, and boundary constraints. Apart from the examples presented in this work, including the ones in Section 4.3, different methods could be considered. When constructing benchmarks in future work, specific underlying constraints may emerge from application requirements.

6. Analysis of Discrete Thomae-like Functions

The analysis presented in this section focuses on the base case, with additive bias $S_1 = 0$ and multiplicative bias $M_1 = 1$. The additive bias simply shifts the function, without affecting any of the properties discussed, while the multiplicative bias shrinks or extends the period of the function, with straightforward and intuitive consequences. The iterative case is not discussed here, because it could be broadly understood as a version of the base case with a different maximum allowed denominator. The extended case generates outputs by evaluating the inputs along each dimension separately; thus, all properties that hold for the base case would also hold for any dimension of the extended case. The asymmetric cases are just versions of the extended case with virtual inputs. As long as the virtual inputs generated cover the input space reasonably well and uniformly, which happens for the presented versions, the properties of the extended case hold true.

Another consideration is the representation of the inputs. If an input can only be represented with 3 decimal places, then a signal with a sampling frequency of 1 kHz would be enough for analyzing the output of the discrete Thomae-like function, given that the input limitation affects the output properties such as the number of maxima or jumps. The analysis detailed in this paper considers input signals with a sampling frequency of up to 10 MHz (7 decimal places).

6.1. Number of Maxima

Similarly to the analytic Thomae function, the discrete versions have one global maximum per period. Regarding local maxima, for the analytic function, every rational could be seen as a local maximum but, since the rationales are countably infinite and the irrationals are uncountably infinite, the percentage of local maxima would still be 0. The discrete versions have a multitude of local maxima, depending on the parameter Q and the limitations in representation for the inputs. Table 3 shows the number of local maxima when varying Q and F_s (sampling frequency) in powers of 10. For $Q \geq F_s$, the percentage of local maxima is about 40% of the number of values that can be represented. The distribution of global minima is a separate property. For $Q = F_s$, the percentage of absolute minima is exactly 40% of the number of values, as expected from the Euler totient function $\phi(Q) = Q \prod_{p|Q} (1 - \frac{1}{p})$ for powers of 10 in the sequence $[\frac{1}{Q}, \frac{2}{Q}, \dots, \frac{Q}{Q}]$, since each number coprime with Q generates a global minimum of value $1/Q$.

$$\frac{\phi(10^p)}{10^p} = \frac{2^{p-1}(2-1)5^{p-1}(5-1)}{10^p} = 0.4$$

We illustrate the computation of the number of local maxima in Table 3 for $F_s = 100$ and $Q = 10$. Since $F_s = 100$, the input values are quantized to two decimal places, yielding the set of possible values $\{0.00, 0.01, \dots, 0.99, 1.00\}$. Because $Q = 10$, the denominator (b) is restricted to the set $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Consider the input value $x = 0.27$ as an example. For each admissible denominator (b), the product $x * b$ is computed and its dis-

tance to the nearest integer is evaluated. The denominator associated with the minimum distance is then selected. The corresponding calculations are as follows:

$0.27 * 1 = 0.27$, the distance to the nearest integer is 0.73;
 $0.27 * 2 = 0.54$, the distance to the nearest integer is 0.46;
 $0.27 * 3 = 0.81$, the distance to the nearest integer is 0.19;
 $0.27 * 4 = 1.08$, distance to the nearest integer is 0.08;
 $0.27 * 5 = 1.35$, the distance to the nearest integer is 0.35;
 $0.27 * 6 = 1.69$, the distance to the nearest integer is 0.38;
 $0.27 * 7 = 1.89$, the distance to the nearest integer is 0.11;
 $0.27 * 8 = 2.16$, the distance to the nearest integer is 0.16;
 $0.27 * 9 = 2.43$, the distance to the nearest integer is 0.43;
 $0.27 * 10 = 2.70$, the distance to the nearest integer is 0.30.

In this example, the minimum distance is obtained for $b = 4$. This process is repeated for all possible input values, producing the following sequence of selected denominators:

1	1	1	1	1	1	1	1	1	1	10	9	8	8	7
7	6	6	6	5	5	5	9	9	4	4	4	4	7	7
10	3	3	3	3	3	3	8	8	5	5	5	7	7	9
9	2	2	2	2	2	2	2	2	2	9	9	7	7	5
5	5	8	8	3	3	3	3	3	3	10	7	7	4	4
4	4	9	9	5	5	5	6	6	6	7	7	8	8	9
10	1	1	1	1	1	1	1	1	1	1				

Each denominator generates an output equal to its reciprocal $\frac{1}{\text{denominator}}$. The local maxima are highlighted in the previous sequence. Consequently, the local maxima of the output sequence correspond to the local minima of the denominator sequence. A local maximum is identified whenever a denominator value is smaller than its immediate left and right neighbors. For flat regions (plateaus, e.g., 1 1 1 1 1 1 1 1 1 1), only one representative maximum is counted (e.g., ... 10 3 3 3 3 3 3 8 ...). Furthermore, the first plateau of values (1 1 1 1 1 1 1 1 1 1) and the last plateau of values (1 1 1 1 1 1 1 1 1 1) are considered to be identical due to the periodicity of the function. Therefore, for $F_s = 100$ and $Q = 10$, the sequence contains 10 distinct local maxima. Finally, the percentage of local maxima is calculated as $\frac{\text{number_of_maxima}}{F_s} * 100$.

Table 3. Number and percentage of local maxima over a period.

$Q \backslash F_s$	10^1	10^2	10^3	10^4	10^5	10^6	10^7
10^1	4 40%	4 40%	4 40%	4 40%	4 40%	4 40%	4 40%
10^2	10 10%	40 40%	40 40%	40 40%	40 40%	40 40%	40 40%
10^3	10 1%	301 30.1%	402 40.2%	394 39.4%	394 39.4%	394 39.4%	394 39.4%
10^4	10 0.1%	1013 10.13%	3326 33.26%	4092 40.92%	3904 39.04%	3892 38.92%	3891 38.91%
10^5	10 0.01%	1013 1.013%	30,119 30.119%	33,117 33.117%	40,870 40.870%	39,311 39.311%	39,123 39.123%
10^6	10 0.001%	1013 0.101%	101,414 10.141%	330,066 33.007%	331,255 33.126%	407,179 40.718%	394,125 39.413%
10^7	10 0.0001%	1013 0.010%	101,414 1.014%	3,011,460 30.115%	3,328,977 33.290%	3,308,901 33.089%	4,084,152 40.841%

6.2. Number and Statistics of Jumps

A jump appears when two consecutive values of the input space (limited in this study by the number of decimal places in the representation of the inputs) generate different output values. For the analytic Thomae function, it is meaningless to talk about jumps, since the input space is the set of real numbers. However, for the discrete Thomae-like versions presented in this work, the analysis of jumps could numerically express the perceived ruggedness of the functions.

Table 4 shows the number of jumps for each configuration. For the considered powers of 10 and $\geq F_s$, each input generates a jump, since even and odd numerators for the fractional part of the input determine different coprime denominators. The number of jumps is defined as the number of transitions between consecutive plateaus of constant denominator values. Considering the previous example of $F_s = 100$ and $Q = 10$, there are 32 jumps highlighted in the following sequence: 1 1 1 1 1 1 1 1 1 1 j 10 j 9 j 8 8 j 77 j 6 6 6 6 j 5 5 5 j 9 9 j 4 4 4 4 j 7 7 j 10 j 3 3 3 3 3 3 j 8 8 j 5 5 5 j 7 7 j 9 9 j 2 2 2 2 2 2 2 2 j 9 9 j 7 7 j 5 5 5 j 8 8 j 3 3 3 3 3 3 j 10 j 7 7 j 4 4 4 4 j 9 9 j 5 5 5 j 6 6 6 j 7 7 j 88 j 9 j 10 j 1 1 1 1 1 1 1 1 1 1.

The output values are generated by these denominators as $\frac{1}{\text{denominator}}$. These values are used to compute the means and standard deviations shown in Tables 5 and 6.

Table 4. Number of jumps over a period.

$F_s \backslash Q$	10^1	10^2	10^3	10^4	10^5	10^6	10^7
10^1	10	10	10	10	10	10	10
10^2	32	100	100	100	100	100	100
10^3	32	892	1000	1000	1000	1000	1000
10^4	32	3046	9898	10,000	10,000	10,000	10,000
10^5	32	3046	89,024	99,898	100,000	100,000	100,000
10^6	32	3046	304,236	988,278	999,898	1,000,000	1,000,000
10^7	32	3046	304,236	8,887,448	9,988,324	9,999,898	10,000,000

Tables 5 and 6 show the means and standard deviations of the absolute values of the jumps, respectively. As the maximum allowed denominator grows and the number of jumps becomes larger, the absolute values and standard deviation of the jumps become lower because the jumps are increasingly smaller, occurring most often between large denominator values.

Table 5. Mean absolute values of jumps.

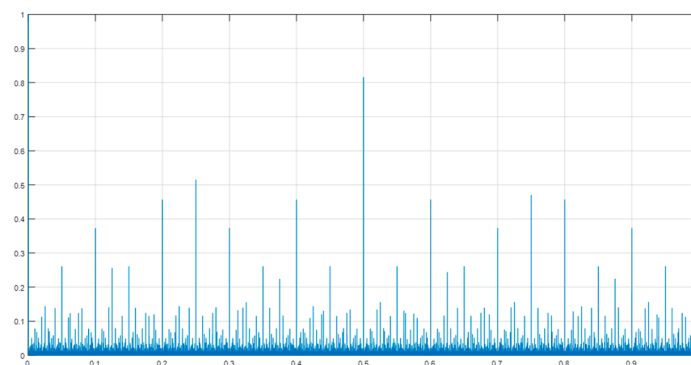
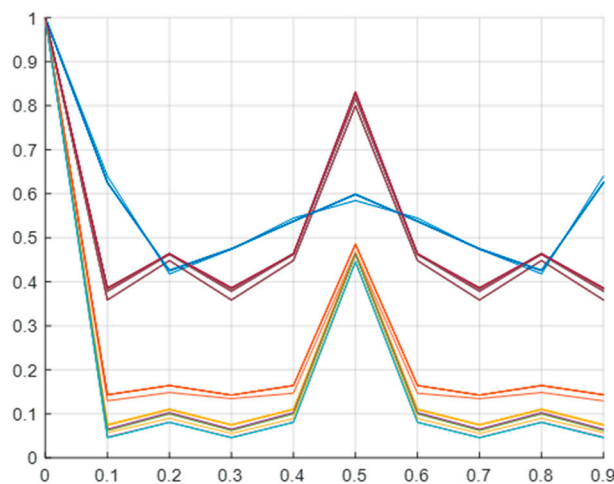
$F_s \backslash Q$	10^1	10^2	10^3	10^4	10^5	10^6	10^7
10^1	0.3	0.3	0.3	0.3	0.3	0.3	0.3
10^2	$4.74 \cdot 10^{-2}$	$7.60 \cdot 10^{-2}$	$7.63 \cdot 10^{-2}$	$7.63 \cdot 10^{-2}$	$7.63 \cdot 10^{-2}$	$7.63 \cdot 10^{-2}$	$7.63 \cdot 10^{-2}$
10^3	$4.74 \cdot 10^{-3}$	$2.80 \cdot 10^{-2}$	$1.39 \cdot 10^{-2}$	$1.39 \cdot 10^{-2}$	$1.39 \cdot 10^{-2}$	$1.39 \cdot 10^{-2}$	$1.39 \cdot 10^{-2}$
10^4	$4.74 \cdot 10^{-4}$	$4.70 \cdot 10^{-3}$	$6.91 \cdot 10^{-3}$	$2.12 \cdot 10^{-3}$	$2.11 \cdot 10^{-3}$	$2.11 \cdot 10^{-3}$	$2.11 \cdot 10^{-3}$
10^5	$4.74 \cdot 10^{-5}$	$4.70 \cdot 10^{-4}$	$2.78 \cdot 10^{-3}$	$1.18 \cdot 10^{-3}$	$2.99 \cdot 10^{-4}$	$2.99 \cdot 10^{-4}$	$2.99 \cdot 10^{-4}$
10^6	$4.74 \cdot 10^{-6}$	$4.70 \cdot 10^{-5}$	$4.70 \cdot 10^{-4}$	$7.13 \cdot 10^{-4}$	$1.71 \cdot 10^{-4}$	$4.04 \cdot 10^{-5}$	$4.04 \cdot 10^{-5}$
10^7	$4.74 \cdot 10^{-7}$	$4.70 \cdot 10^{-6}$	$4.70 \cdot 10^{-5}$	$2.78 \cdot 10^{-4}$	$1.22 \cdot 10^{-4}$	$2.27 \cdot 10^{-5}$	$5.20 \cdot 10^{-6}$

Table 6. Standard deviations of the absolute values of jumps.

$\begin{matrix} Q \\ F_s \end{matrix}$	10^1	10^2	10^3	10^4	10^5	10^6	10^7
10^1	0.3266	0.3266	0.3266	0.3266	0.3266	0.3266	0.3266
10^2	$1.42 \cdot 10^{-1}$	$1.57 \cdot 10^{-1}$	$1.57 \cdot 10^{-1}$	$1.57 \cdot 10^{-1}$	$1.57 \cdot 10^{-1}$	$1.57 \cdot 10^{-1}$	$1.57 \cdot 10^{-1}$
10^3	$4.69 \cdot 10^{-2}$	$6.15 \cdot 10^{-2}$	$5.66 \cdot 10^{-2}$	$5.66 \cdot 10^{-2}$	$5.66 \cdot 10^{-2}$	$5.66 \cdot 10^{-2}$	$5.66 \cdot 10^{-2}$
10^4	$1.49 \cdot 10^{-2}$	$2.18 \cdot 10^{-2}$	$2.29 \cdot 10^{-2}$	$1.86 \cdot 10^{-2}$	$1.86 \cdot 10^{-2}$	$1.86 \cdot 10^{-2}$	$1.86 \cdot 10^{-2}$
10^5	$4.72 \cdot 10^{-3}$	$7.05 \cdot 10^{-3}$	$8.11 \cdot 10^{-3}$	$7.71 \cdot 10^{-3}$	$5.95 \cdot 10^{-3}$	$5.95 \cdot 10^{-3}$	$5.95 \cdot 10^{-3}$
10^6	$1.49 \cdot 10^{-3}$	$2.23 \cdot 10^{-3}$	$2.75 \cdot 10^{-3}$	$2.84 \cdot 10^{-3}$	$2.48 \cdot 10^{-3}$	$1.89 \cdot 10^{-3}$	$1.89 \cdot 10^{-3}$
10^7	$4.71 \cdot 10^{-4}$	$7.07 \cdot 10^{-4}$	$8.81 \cdot 10^{-4}$	$9.68 \cdot 10^{-4}$	$9.38 \cdot 10^{-4}$	$7.87 \cdot 10^{-4}$	$5.99 \cdot 10^{-4}$

6.3. Autocorrelation

Like the analytic Thomae function, the autocorrelation drops very quickly, especially noticeable for higher F_s values, as shown in Figure 7. Focusing on the autocorrelation displacements that can be computed for all sampling frequencies, Figure 8 shows the general trend, while ignoring the (mostly) lower autocorrelation values that are in-between. The red group, with the highest autocorrelation value above 0.8 at displacement 0.5, corresponds to the cases with $Q \geq F_s$. The blue group, with the fewest inflections and an autocorrelation value around 0.6 at displacement 0.5, corresponds to the cases with $F_s = 1$. The yellow group, with an autocorrelation value below 0.5 at displacement 0.5, corresponds to the remaining cases with $Q < F_s$.

**Figure 7.** Autocorrelation for the Thomae-like function for $Q = 10^7$ and $F_s = 10^7$.**Figure 8.** Autocorrelation trends for Thomae-like functions.

6.4. Numerical Integral

The analytic function is Riemann integrable with a value of 0, but numerical integration could be performed on the discrete versions as a measure of the spikes' overall influence. Table 7 shows the values obtained with the trapezoidal rule. As expected, larger values of Q provide a lower baseline of the function and, thus, a lower integral value, indicating a reduced influence of large local maxima such as 0.5.

Table 7. Numerical integral (trapezoidal rule) over a period.

$Q \backslash F_s$	10^1	10^2	10^3	10^4	10^5	10^6	10^7
10^1	0.270	0.270	0.270	0.270	0.270	0.270	0.270
10^2	0.371	$5.20 \cdot 10^{-2}$	$5.15 \cdot 10^{-2}$	$5.15 \cdot 10^{-2}$	$5.15 \cdot 10^{-2}$	$5.15 \cdot 10^{-2}$	$5.15 \cdot 10^{-2}$
10^3	0.363	$6.30 \cdot 10^{-3}$	$8.38 \cdot 10^{-3}$	$8.30 \cdot 10^{-3}$	$8.30 \cdot 10^{-3}$	$8.30 \cdot 10^{-3}$	$8.30 \cdot 10^{-3}$
10^4	0.363	$6.53 \cdot 10^{-3}$	$8.79 \cdot 10^{-3}$	$1.20 \cdot 10^{-3}$	$1.19 \cdot 10^{-3}$	$1.18 \cdot 10^{-3}$	$1.18 \cdot 10^{-3}$
10^5	0.363	$6.51 \cdot 10^{-3}$	$9.26 \cdot 10^{-3}$	$1.14 \cdot 10^{-3}$	$1.64 \cdot 10^{-4}$	$1.62 \cdot 10^{-4}$	$1.64 \cdot 10^{-4}$
10^6	0.363	$6.51 \cdot 10^{-3}$	$9.32 \cdot 10^{-3}$	$1.20 \cdot 10^{-3}$	$1.42 \cdot 10^{-4}$	$2.17 \cdot 10^{-5}$	$2.15 \cdot 10^{-5}$
10^7	0.363	$6.51 \cdot 10^{-3}$	$9.32 \cdot 10^{-3}$	$1.21 \cdot 10^{-3}$	$1.47 \cdot 10^{-4}$	$1.70 \cdot 10^{-5}$	$2.75 \cdot 10^{-6}$

6.5. Spectrum

The analytic function does not have an insightful Fourier transform within the interval $[0, 1]$, because the function is zero almost everywhere. By contrast, the discrete versions can have a useful discrete Fourier transform by preserving information. For $Q < F_s$, many close input values have the same output (as shown in Figure 2b), which generates a strong continuous component, with larger discrepancies between Q and F_s creating larger magnitudes in the continuous component. For $Q \geq F_s$, especially in the case of larger values of Q , as the baseline becomes lower and is covered with all sorts of numbers close to 0 (as indicated by the lower mean and standard deviation of absolute jumps), the spectrum becomes more uniform, signaling similarities with random white noise. Figures 9 and 10 show examples of the two phenomena discussed.

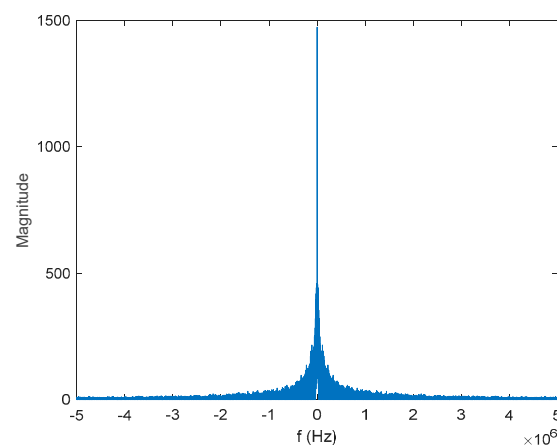


Figure 9. Magnitude spectrum for the Thomae-like function for $Q = 10^5$ and $F_s = 10^7$.

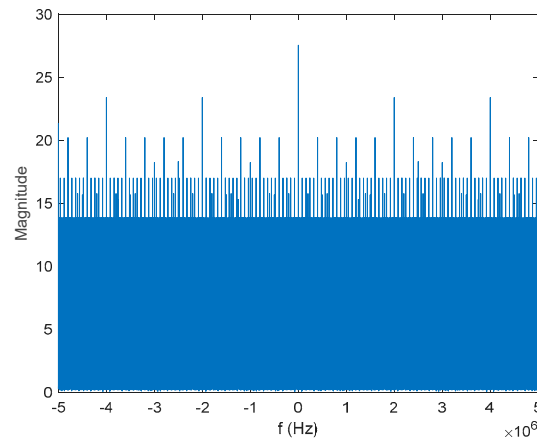


Figure 10. Magnitude spectrum for the Thomae-like function for $Q = 10^7$ and $F_s = 10^7$.

7. Conclusions and Perspectives

7.1. Preliminary Results

The mathematical model presented in this paper is intended to represent a baseline for testing the convergence of evolutionary algorithms, whose implementation assumes that a small change in the input would produce a small variation in the output. Discrete Thomae-like functions contradict this assumption, generating large variations in the output under small changes in the input. This fact makes such functions relevant candidates for benchmarking against current state-of-the-art algorithms by targeting atypical types of optimization problems.

One well-studied real-world optimization problem that aligns with Thomae-like concept is the High-Performance Gearbox Design [36,37]. As demonstrated in previous research [38], this problem exhibits the “hunting tooth” physical phenomenon. When designing high-speed gearboxes (such as those used in electric vehicle powertrains, wind turbines, or aerospace actuators), engineers must choose a gear ratio. If the gear ratio is simplified to a fraction with a small denominator (e.g., $\frac{2}{1}$, $\frac{3}{2}$, $\frac{4}{3}$), the exact same teeth on the mating gears will smash into each other repeatedly after just a few revolutions. If there is even a microscopic manufacturing flaw on one tooth, it will repeatedly hit the same spot on the opposing gear, causing loud acoustic noise and severe vibration spikes. These aspects resemble a Thomae-like behavior. If the gear ratio is an irrational number or a fraction with an extremely large prime denominator, the tooth mesh exhibits a shifting pattern. Any single tooth will evenly distribute its contact across all other teeth; thus, vibrations cancel and the wear is perfectly uniform.

As a multi-objective mathematical formulation, this problem could be regarded as a reduction gearbox optimization problem where engineers need to optimize the exact gear ratio x . The decision space is defined as $\Omega = [2.5, 4.5]$, offering a feasible engineering window to choose a gear ratio x . Objective 1 (continuous) is to maximize efficiency. The motor has an ideal operating rotation per minute (RPM) curve, meaning that there is a mathematically perfect continuous ratio (e.g., 3.72) that maximizes the battery range and acceleration. The farther one moves from this sweet spot, the worse the vehicle’s efficiency. Objective 1 is described in Equation (12):

$$\min f_1(x) = (x - 3.72)^2 \quad (12)$$

Objective 2 is to minimize vibro-acoustic harshness and wear (Thomae-like behavior). The vibration amplitude $V(x)$ of the gearbox is a baseline level of a structural noise V_0 , creating a massive spike whenever the ratio matches a simple rational fraction because

of the hunting tooth resonance. The smaller the denominator, the more catastrophic the vibration spike. Objective 2 is described in Equation (13):

$$\min f_2(x) = V_0 + \alpha * T(x) \quad (13)$$

where α is the amplification factor of the resonance and $T(x)$ is the Thomae function, injecting severe spikes at every rational gear ratio $\frac{p}{q}$, based on the severity of its denominator q . The algorithms used may find a gear ratio that yields very good efficiency (f_1), but this gear ratio could also generate acoustic noise and fail prematurely (f_2), thus highlighting the conflicting nature of the objectives.

The design of radio-frequency integrated circuits, particularly fractional-N phase-locked loop (PLL) synthesizers, provides another real-world optimization problem which exhibits Thomae-like behavior. These circuits generate precise carrier frequencies for 5G, Wi-Fi, or Bluetooth devices. A well-known phenomenon in fractional-N PLLs is “fractional spurs” [39,40]. To synthesize the desired frequency, the PLL multiplies the baseline frequency by a fractional value x . For rational values of x with a small denominator, the internal digital divider tends to follow the same rounding patterns, producing a discrete component known as fractional spurs. For irrational values, the rounding error becomes completely chaotic, and its spectral energy spreads over a broad frequency range, appearing as noise.

For mobile phones, the multi-objective optimization problem consists of engineers determining an optimal value for the fractional ratio x . The objectives could be formulated as follows. Objective 1 is to maximize the connection speed, as described in Equation (14):

$$\min f_1(x) = |x - x_{target}| \quad (14)$$

A mobile phone would ideally switch to the desired frequency as soon as possible. The closer x is to the ideal value x_{target} of the channel, the quicker the circuit locks onto the target. The second objective (Thomae-like behavior), described in Equation (15), is to minimize interferences (spurs).

$$\min f_2(x) = Z_{noise} + \beta * T(x) \quad (15)$$

where Z_{noise} is the baseline noise value, $T(x)$ is the Thomae function, and $\beta > 0$ represents the sensitivity of the synthesizer to fractional spurs. $T(x)$ captures the observation that fractional division ratios corresponding to rational numbers with small denominators generate the strongest periodic error sequences and, consequently, the largest spurious spectral components. As the denominator increases, the spur level decreases, while ratios that are not well approximated by low-denominator rationales exhibit predominantly noise-like behavior.

As mentioned above, future research will include runs, simulations, and analyses of the behavior of state-of-the-art algorithms, including NSGA-II, NSGA-III, and MOPSO, on Thomae-like functions (optimization problems). The preliminary results and comparisons of these algorithms have been presented and discussed in our previous works [41–44].

7.2. Further Perspectives

This paper presents a systematic mathematically designed model of generating discrete Thomae-like functions. Starting with the analytic cases presented in Section 3, we present five discrete generalizations and discuss the behavior of these functions and the preservation of their properties. Our analysis focuses on the base case ($n = m = 1$), with additive bias $S_1 = 0$ and multiplicative bias $M_1 = 1$. The additive bias shifts the function without interfering with any of the discussed properties, while the multiplicative bias shrinks or extends the period of the function. The iterative case could be regarded as an-

other version of the base case with a different value for Q . Similarly, the extended and asymmetric cases preserve the properties of the base case; therefore, the properties of the base case remain valid for any dimensionality.

Tables 3–7 present the statistics of the discrete functions according to the percentage of local maxima, the number and distribution of jumps, and numerical integration. Similarly to the analytic version, autocorrelation drops very quickly, as shown in Figure 7, particularly for higher F_s values. The numeric integration and spectral aspects are also discussed, providing an analysis of the influence generated by the relationship between Q and F_s .

To facilitate reproducibility of the experiments presented in this work, which we conducted in Matlab R2022b, we specify the following considerations. The grid consists of F_s samples. A plateau of values is considered only once (the first plateau encountered), as shown in the example in Section 6. Specifically, the starting plateau is the same as the last one due to periodicity. For autocorrelation, we used the `xcorr()` function. For the discrete Fourier transform, we used the `fft()` function from the core Matlab library. Regarding floating point computations, the default limit of 10^{-15} in Matlab was used. As mentioned above, our experiments did not exceed the value of 10^{-7} . Therefore, although the rounding errors were not negligible, our experiments were not affected. Moreover, the main properties presented in this paper were also not affected.

Future developments may include examining other values of Q , which will result in different types of properties. Especially interesting are prime values of Q , because all the lower natural numbers are coprime, or highly composite numbers, with very few lower coprime numbers. The proposed generalization can be combined with other known generalizations to produce various useful results, as mentioned in the Related Work section. Moreover, the analysis shows that the current output values are sparser at the top and denser at the bottom. Other definitions may distribute the outputs more uniformly, constituting a basis for further analysis.

Moreover, considering the diverse range of optimization problems, some of which are presented in this work, the need to identify and simulate Thomae-like functions on computing machines has become imperative. For example, a minor miscalculation of weather forecast data might lead to a major misprediction for the estimated time interval. A fallen tree that affects a single local electric power line could also damage the rest of the network. The proposed mathematical model could be regarded as an extreme case of these real-world natural disaster problems. Chaotic systems exhibit such behaviors in an extreme form. Furthermore, this work could be a pioneer in generating a set of adversarial fitness functions for multi-objective optimization algorithms, designed to identify their weaknesses. Such developments will be included in the FADSE framework for testing state-of-the-art algorithms and establishing a public benchmark test suite. Since the core library of the FADSE is built upon and extends the open-source evolutionary algorithm library jMetal 6.6, incorporating new Thomae-like functions into the FADSE will primarily involve extending its optimization problem library. The following sample code constitutes an example of problem configuration as an input for the FADSE:

```

<design_space>
  <problem_config name = "PLL" type = "Thoame-function"> ...
    <parameter name = "maximumTimeOfASimulation" value = "140"/>
    ...
  </problem_config>
  <parameters> ...
    <parameter name = "x" description = "gear ratio" type = "double" min = "2.5"
max = "4.5"/>
    ...
  </parameters>
  ...
  <metaheuristic name = "NSGAI" config_path = "nsgai.properties"/>
</design_space>

```

As shown in Figure 11, the class hierarchy of optimization problems in the FADSE enables its extensibility with new problems.

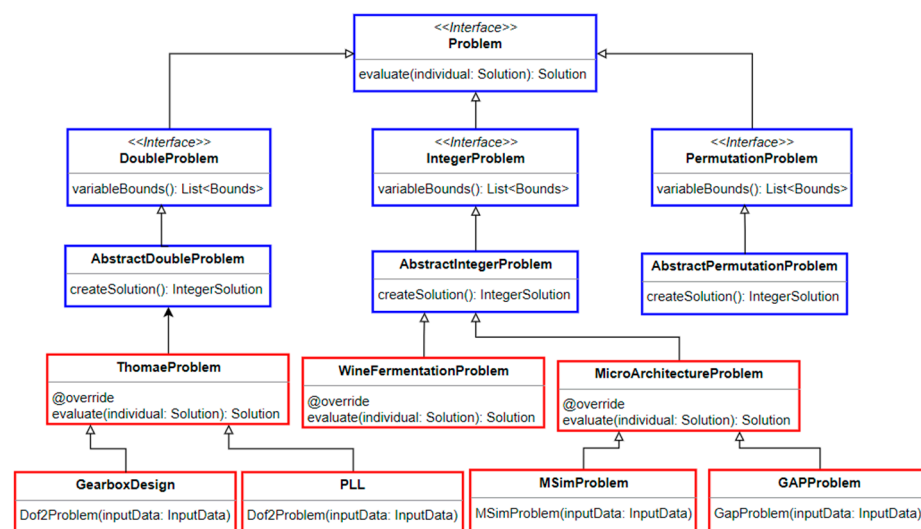


Figure 11. Hierarchy of optimization problems in FADSE.

As the FADSE is an open-source framework, it will provide researchers with a straightforward and flexible platform for testing, evaluating, and further extending the library with various families of Thomae-like functions. Features that facilitate, for example, comparisons between a Thomae-like, function-based benchmark and current DTLZ, CEC, and ZDT benchmarks will also be included in future releases of the FADSE. The current development architecture of the FADSE is shown in Figure 12.

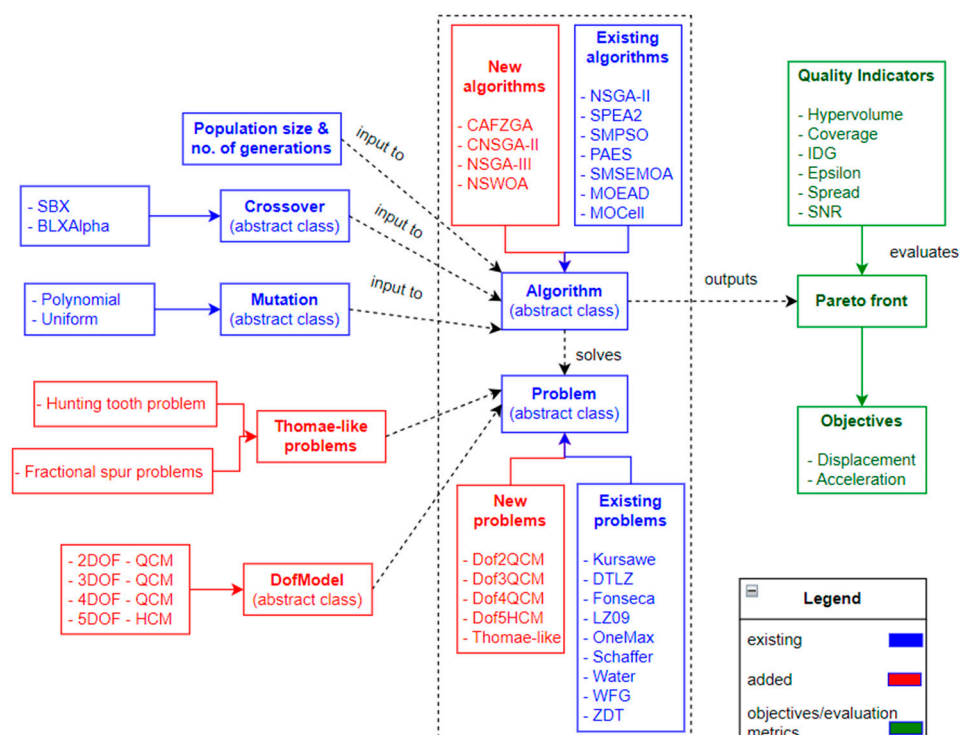


Figure 12. FADSE development infrastructure [45].

The features of the FADSE include an easily extensible object-oriented design, the generation of automatic visual graphs, and a custom-designed user interface. Moreover, the development of adapted quality indicators might be necessary, since classical metrics such as hypervolume may not adequately capture the irregular behavior of Thomae-like functions. To facilitate research in this direction, we aim to include and provide the necessary tools and algorithms in the FADSE for validating adapted quality indicators. The existing class hierarchy in the FADSE offers straightforward extensibility for any performance metrics, enabling researchers to easily implement, extend, and analyze new metrics and various families of Thomae-like functions under different optimization scenarios.

Author Contributions: Conceptualization, A.P., R.C. and A.D.; Methodology, A.P.; Software, A.P. and A.D.; Validation, A.P., M.N. and R.C.; Formal analysis, M.N. and R.C.; Investigation, A.P. and M.N.; Resources, A.P.; Data curation, A.F. and A.D.; Writing—original draft, A.P., A.F. and M.N.; Writing—review & editing, A.P., A.F., M.N. and A.D.; Visualization, A.F. and M.N.; Supervision, A.F., R.C. and A.D.; Project administration, A.F. and A.D.; Funding acquisition, A.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

References

1. Abdulrashid, I.; Delen, D.; Usman, B.; Uzochukwu, M.I.; Ahmed, I. A multi-objective optimization framework for determining optimal chemotherapy dosing and treatment duration. *Healthc. Anal.* **2024**, *5*, 100335. <https://doi.org/10.1016/j.health.2024.100335>.

2. Baesler, F.; Cornejo, O.; Obreque, C.; Forcael, E.; Carrasco, R. A Multi-Objective Simulation–Optimization Framework for Emergency Department Efficiency Using RSM and Goal Programming. *Systems* **2025**, *13*, 912. <https://doi.org/10.3390/systems13100912>.
3. Sahu, D.; Sinha, P.; Prakash, S.; Yang, T.; Rathore, R.; Wang, L. A multi objective optimization framework for smart parking using digital twin pareto front MDP and PSO for smart cities. *Sci. Rep.* **2025**, *15*, 7783. <https://doi.org/10.1038/s41598-025-91565-0>. PMID: 40044776.
4. Dastpak, P.; Mohammadi, E.; Javankhoshdel, S.; Korfiatis, G.; Sousa, R. A framework to optimize a designed geotechnical system probabilistically using MLP-ANN and ELECTERE decision making—A nailed wall study. *Geod. AI* **2025**, *2*, 100014. <https://doi.org/10.1016/j.geoai.2025.100014>.
5. Zhu, X.; Ji, Y. A digital twin-based multi-objective optimization method for technical schemes in process industry. *Int. J. Comput. Integr. Manuf.* **2022**, *36*, 443–468. <https://doi.org/10.1080/0951192X.2022.2126013>.
6. Kim, H.-J.; Kim, M.-J.; Jin, Y.-B. An Integrated Multi-Objective Optimization Framework for Environmental Performance: Sunlight, View, and Privacy in a High-Density Residential Complex in Seoul. *Sustainability* **2025**, *17*, 7490. <https://doi.org/10.3390/su17167490>.
7. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. <https://doi.org/10.1109/4235.996017>.
8. Deb, K.; Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. Evol. Comput.* **2014**, *18*, 577–601.
9. Song, Q.; Liu, Y.; Zhang, X.; Zhang, Y. Hyperplane-Assisted Multi-objective Particle Swarm Optimization with Twofold Proportional Assignment Strategy. *Int. J. Comput. Intell. Syst.* **2024**, *17*, 306. <https://doi.org/10.1007/s44196-024-00702-6>.
10. Fathollahi-Fard, A.M.; Hajiaghaei-Keshteli, M. Red Deer Algorithm (RDA): A new optimization algorithm inspired by Red Deers' mating. In *International Conference on Industrial Engineering*; IEEE: New York, NY, USA, 2016; Volume 12, pp. 331–342.
11. Thomae, J. *Einleitung in die Theorie der Bestimmten Integrale (Introduction to the Theory of Definite Integrals)*; Verlag von Louis Nebert: Halle an der Saale, Germany, 1875.
12. Calborean, H.; Vintan, L. An automatic design space exploration framework for multicore architecture optimizations. In *Proceedings of the 9th RoEduNet IEEE International Conference, Sibiu, Romania, 24–26 June 2010*; pp. 202–207.
13. Deb, K.T. Scalable Test Problems for Evolutionary Multiobjective Optimization. In *Evolutionary Multiobjective Optimization*; Abraham, A., Jain, L., Goldberg, R., Eds.; Advanced Information and Knowledge Processing; Springer: London, UK, 2005. https://doi.org/10.1007/1-84628-137-7_6.
14. Zitzler, E.; Deb, K.; Thiele, L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evol. Comput.* **2000**, *8*, 173–195. <https://doi.org/10.1162/10635600568202>.
15. Huband, S.; Hingston, P.; Barone, L.; While, L. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Trans. Evol. Comput.* **2006**, *10*, 477–506.
16. Liang, J.; Qu, B.; Suganthan, P. *Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization*; Nanyang Technological University: Singapore, 2013.
17. Weierstraß, K. Über Continuirliche Functionen Eines Reellen Arguments, die für Keinen Werth des Letzteren Einen Bestimmten Differentialquotienten Besitzen. In *Ausgewählte Kapitel aus der Funktionenlehre*; Siegmund-Schultze, R., Ed.; Teubner-Archiv zur Mathematik; Springer: Vienna, Austria, 1988; Volume 9. https://doi.org/10.1007/978-3-7091-9533-8_5.
18. Hardy, G.H. Weierstrass's Non-Differentiable Function. *Trans. Am. Math. Soc.* **1916**, *17*, 301–325. <https://doi.org/10.2307/1989005>.
19. Mandelbrot, B.B. *The Fractal Geometry of Nature*; W.H. Freeman: New York, NY, USA, 1982.
20. Navascués, M. Fractal Polynomial Interpolation. *Z. Anal. Anwendungen* **2005**, *24*, 401–418. <https://doi.org/10.4171/ZAA/1248>.
21. Katsuura, H. Continuous nowhere-differentiable functions—An application of contraction mappings. *Am. Math. Mon.* **1991**, *98*, 411–416.
22. Finck, S.; Hansen, N.; Ros, R.; Auger, A. *Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions*; RR-6829; INRIA: Le Chesnay-Rocquencourt, France, 2009. Available online: <https://inria.hal.science/inria-00362633v2/document> (accessed on 05 July 2026).
23. Wu, Z.; Liu, J. Perlin noise and its improvements: A literature review. *Appl. Comput. Eng.* **2024**, *77*, 278–287. <https://doi.org/10.54254/2755-2721/77/20240437>.
24. Lagae, A.; Lefebvre, S.; Cook, R.; DeRose, T.; Drettakis, G.; Ebert, D.; Lewis, J.; Perlin, K.; Zwicker, M. A Survey of Procedural Noise Functions. *Comput. Graph. Forum* **2010**, *29*, 2579–2600. <https://doi.org/10.1111/j.1467-8659.2010.01827.x>.

25. Jamil, M.; Yang, X.-S. A literature survey of benchmark functions for global optimization problems. *Int. J. Math. Model. Numer. Optim.* **2013**, *4*, 150–194. <https://doi.org/10.1504/IJMMNO.2013.055204>.
26. Plevris, V.; Solorzano, G. A Collection of 30 Multidimensional Functions for Global Optimization Benchmarking. *Data* **2022**, *7*, 46. <https://doi.org/10.3390/data7040046>.
27. Layeb, A. New Hard Benchmark Functions for Global Optimization. *arXiv* **2022**, arXiv:2202.04606. <https://doi.org/10.48550/arXiv.2202.04606>.
28. Naser, M.Z.; Khaled al-Bashiti, M.; Teymori Gharah Tapeh, A.; Eslamlou, A.D.; Naser, A.; Kodur, V.; Hawileeh, R.; Abdalla, J.; Khodadadi, N.; Gandomi, A.H. A Review of 315 Benchmark and Test Functions for Machine Learning Optimization Algorithms and Metaheuristics with Mathematical and Visual Descriptions. *arXiv* **2024**, arXiv:2406.09581. <https://doi.org/10.48550/arXiv.2406.09581>.
29. He, Y.; Aranha, C. Evolving Benchmark Functions to Compare Evolutionary Algorithms via Genetic Programming. In Proceedings of the 2024 IEEE Congress on Evolutionary Computation (CEC), Yokohama, Japan, 30 June–5 July 2024; pp. 1–8. <https://doi.org/10.1109/CEC60901.2024.10611801>.
30. Niewenhuis, D.; van den Berg, D. Making Hard(Er) Benchmark Test Functions. In *ECTA 2022*; SciTePress: Valletta, Malta, 2022. <https://doi.org/10.5220/0011405300003332>.
31. Beanland, K.; Roberts, J.; Stevenson, C. Modifications of Thomae’s Function and Differentiability. *Am. Math. Mon.* **2009**, *116*, 531–535.
32. Lebesgue, H. Sur l’intégration des fonctions discontinues. *Ann. Sci. l’Éc. Norm. Supér.* **1910**, *3*, 361–450.
33. Copar, S.; Weis, C. Gap widths in the fractional parts of roots. *Int. J. Number Theory* **2025**, *21*, 2281–2304. <https://doi.org/10.1142/S1793042125501118>.
34. Feng, X.; Lu, C.; Schulte, J.; Shan, Z.; Liu, G. Pendulum waves: A model of Thomae’s function. *Am. J. Phys.* **2023**, *91*, 964–969. <https://doi.org/10.1119/5.0131936>.
35. Bortolatto, R.B. On an Example of du Bois-Reymond, Dyadic Rationals, and Thomae’s Function. *Math. Mag.* **2026**, *99*, 73–79. <https://doi.org/10.1080/0025570X.2024.2369492>.
36. Marafona, J.D.M.; Carneiro, G.N.; Marques, P.M.T.; Martins, R.C.; António, C.C.; Seabra, J.H.O. Gear design optimization: Stiffness versus dynamics. *Mech. Mach. Theory* **2024**, *191*, 105503. <https://doi.org/10.1016/j.mechmachtheory.2023.105503>.
37. Younes, E.B.; Changenet, C.; Bruyère, J.; Rigaud, E.; Perret-Liaudet, J. Multi-objective optimization of gear unit design to improve efficiency and transmission error. *Mech. Mach. Theory* **2022**, *167*, 104499. <https://doi.org/10.1016/j.mechmachtheory.2021.104499>.
38. Bechhoefer, E.; Janke, C. Hunting Tooth Gear Condition Indicator: For Epicycle Gear Fault Detection. In *2024 International Conference on Control, Automation and Diagnosis (ICCAD)*; IEEE: Paris, France, 2024; pp. 1–6. <https://doi.org/10.1109/ICCAD60883.2024.10553742>.
39. Kennedy, M.; Mazzaro, V.; Mai, D. Nonlinearity-Induced Spurs in Fractional-N Frequency Synthesizers. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *69*, 2617–2622. <https://doi.org/10.1109/TCSII.2022.3169224>.
40. Mai, D.; Dahlan, A.; Kennedy, M. MASH DDSM-Induced spurs in a fractional-N frequency synthesizer. In *2019 26th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2019*; IEEE: New York, NY, USA, 2019; pp. 13–16. <https://doi.org/10.1109/ICECS46596.2019.8965008>.
41. Jahr, R.; Calborean, H.; Vintan, L.; Ungerer, T. Finding near-perfect parameters for hardware and code optimizations with automatic multi-objective design space explorations. *Concurr. Comput. Pract. Exp.* **2015**, *27*, 2196–2214.
42. Pătrăușanu, A.; Florea, A.; Neghină, M.; Chiș, R.; Dicoiu, A. Fourier Apparent Front: A Method of Generating Apparent Fronts Without a Template. In *Technological Innovation to Tackle Societal Challenges. DoCEIS 2026*; Camarinha-Matos, L.M., Ferrada, F., Eds.; IFIP Advances in Information and Communication Technology; Springer: Cham, Switzerland, 2027; Volume 790. https://doi.org/10.1007/978-3-032-28939-1_7.
43. Pătrăușanu, A.; Florea, A.; Neghină, M.; Chiș, R.; Dicoiu, A. Car suspension system design optimization through bio-inspired evolutionary algorithms. In Proceedings of the 28th International Conference on System Theory, Control and Computing (IC-STCC), Sinaia, Romania, 10–12 October 2024. <https://doi.org/10.1109/ICSTCC62912.2024.10744669>.
44. Florea, A.; Cofaru, I.; Pătrăușanu, A.; Cofaru, N.; Fiore, U. Superposition of populations in multi-objective evolutionary optimization of car suspensions. *Eng. Appl. Artif. Intell.* **2023**, *126*, 107026. <https://doi.org/10.1016/j.engappai.2023.107026>.
45. Pătrăușanu, A.; Florea, A.; Neghină, M.; Dicoiu, A.; Chiș, R. A Systematic Review of Multi-Objective Evolutionary Algorithms Optimization Frameworks. *Processes* **2024**, *12*, 869. <https://doi.org/10.3390/pr12050869>.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.