



A Genetic Programming Based Heuristic to Simplify Rugged Landscapes Exploration

Gloria Pietropoli ¹, Giuliamaria Menara ¹, Mauro Castelli ^{2*}¹ Dipartimento di Matematica e Geoscienze, Università degli Studi di Trieste, H2bis Building, Via Alfonso Valerio 12/1, 34127 Trieste, Italy.² NOVA Information Management School (NOVA IMS), Universidade NOVA de Lisboa, Campus de Campolide, 1070-312 Lisboa, Portugal.

Abstract

Some optimization problems are difficult to solve due to a considerable number of local optima, which may result in premature convergence of the optimization process. To address this problem, we propose a novel heuristic method for constructing a smooth surrogate model of the original function. The surrogate function is easier to optimize but maintains a fundamental property of the original rugged fitness landscape: the location of the global optimum. To create such a surrogate model, we consider a linear genetic programming approach coupled with a self-tuning fitness function. More specifically, to evaluate the fitness of the produced surrogate functions, we employ Fuzzy Self-Tuning Particle Swarm Optimization, a setting-free version of particle swarm optimization. To assess the performance of the proposed method, we considered a set of benchmark functions characterized by high noise and ruggedness. Moreover, the method is evaluated over different problems' dimensionalities. The proposed approach reveals its suitability for performing the proposed task. In particular, experimental results confirm its capability to find the global argminum for all the considered benchmark problems and all the domain dimensions taken into account, thus providing an innovative and promising strategy for dealing with challenging optimization problems.

Keywords:

Genetic Programming;
Particle Swarm Optimization;
Surrogate Models;
Fitness Landscapes.

Article History:

Received:	15	March	2023
Revised:	02	June	2023
Accepted:	17	June	2023
Available online:	12	July	2023

1- Introduction

An optimization problem consists of maximizing (or minimizing) a fitness function relative to a fixed search space [1]. Some optimization problems are characterized by features that make it difficult for traditional statical optimization methods and search heuristics to effectively explore the search space. For instance, the presence of a considerable number of local optima might cause premature convergence of the search method and a consequent failure to discover the global optimum. Thus, while search heuristics have been used to address problems across different domains, their performance is tightly connected with the underlying problem's fitness landscape.

The fitness landscape is a theoretical concept that allows analyzing the relation between the candidate solutions of a problem and the corresponding fitness values. A formal definition of fitness landscape was proposed by Stadler in 2002 [2], who defined a fitness landscape as a triple (X, N, f) in which X is the set of candidate solutions, N is a neighborhood function, and f is the fitness function, a mapping from the set of the candidate solutions to the target variable space.

The analysis of the fitness landscape was fundamental to provide an intuitive understanding of the complexity of optimization problems [3], explaining evolutionary algorithm behavior [4, 5], and design benchmark problems [6]. In particular, to evaluate the difficulty of a given optimization problem, the number of local optima is the first characteristic

* CONTACT: mcastelli@novaims.unl.pt

DOI: <http://dx.doi.org/10.28991/ESJ-2023-07-04-01>

© 2023 by the authors. Licensee ESJ, Italy. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<https://creativecommons.org/licenses/by/4.0/>).

of the fitness landscape that one may consider. However, only considering the number of local optima is neither sufficient nor necessary for fully defining the complexity of the search process, and existing research has demonstrated that the number of local optima is not a proxy for determining the difficulty of an optimization problem [7]. Another method that can be used to assess the difficulty of an optimization problem is ruggedness. Given an optimization problem, its fitness landscape is considered rugged if it shows a large number of "*up-hills and down-hills*" fitness values [8]. In general, for an optimization problem, it is easy to fall into local optima and difficult to obtain a global optimum owing to the ruggedness of its fitness landscape [9].

Therefore, framing the search process necessary to solve an optimization problem in the context of a fitness landscape search, it is often necessary to explore a rugged multidimensional surface, which consistently turns the search for the global optimum into a challenging, time-consuming problem. This irregularity prevents the adoption of simple optimization techniques such as gradient descent and hill-climbing, as they cannot successfully deal with such rugged landscapes. Due to the importance of this problem, researchers have investigated the performance of evolutionary search algorithms that are gradient-free optimizers.

To reduce the burden associated with the exploration of a rugged landscape, the use of surrogate models (SMs) has been proposed [10]. An SM consists of defining an approximate fitness function whose evaluation is typically less expensive than its original counterpart. SMs are often used to reduce the cost of expensive objective functions by acting as approximation models that mimic the behavior of the original function as closely as possible while being fast surrogates for time-consuming computer simulations. In a few words, an SM is an easily measurable mathematical model that approximates an expensive objective function as precisely as possible.

To build a surrogate model, researchers relied on several techniques: artificial neural networks [11], genetic programming [12], or a combination of the techniques [13]. Surrogate models proved their suitability as a method to reduce the time necessary for exploring the search space of an optimization problem and were employed in several applicative domains in which the evaluation of the fitness function is particularly time-consuming. While surrogates have demonstrated their effectiveness, especially when combined with evolutionary computation algorithms [14], there is a relevant drawback to consider. The construction of surrogate models is usually based on techniques that, similarly to evolutionary algorithms, are characterized by several hyperparameters. Unfortunately, surrogate models can provide a satisfactory approximation of the original function only when the hyperparameters are correctly determined. As is common in machine learning, choosing the ideal values for different hyperparameters is a complex task that requires experience and significant effort in terms of time. Following the line of research that exploits the use of surrogate models to approximate rugged functions, this paper proposes the use of an SM-like method based on genetic programming and particle swarm optimization.

In particular, to build this SM-like method, our idea is to retain the appeal of genetic programming (GP) [15] algorithms, as they can handle challenging problems with high-quality designs while enhancing computational efficiency. GP is one of the most prominent evolutionary-based techniques, with the ability to evolve programs to solve specific problems, that belongs to the wider class of optimization techniques called population-based methods [16]. GP employs a variable model representation that makes no assumptions about the classes of models that will be relevant to the problem at hand and places the fewest possible constraints on the space of models that are allowed to be explored throughout the evolutionary process.

Combining what we have discussed so far, the main purpose of this research is to find the global optimum of rugged landscapes by evolving, through a GP-based routine, a valid surrogate function to approximate our problem adequately. To achieve this objective, we implemented a novel method called the GP-FST-PSO Surrogate Model that generates smooth surrogate approximations of the original landscapes, thus making their exploration (i.e., searching for the global optimum) straightforward. The main idea presented in this work relies on a combination of a GP-based algorithm coupled with a self-tuning fitness function. More specifically, to evaluate the fitness of the produced surrogate functions, we employ Fuzzy Self-Tuning Particle Swarm Optimization (FST-PSO) [17], a setting-free version of particle swarm optimization (PSO). FST-PSO was selected because, being a self-tuning algorithm, it does not require any hyperparameters, which facilitates the subsequent analysis of GP-FST-PSO. To estimate GP-FST-PSO efficiency, we considered a set of benchmark functions that are commonly used to test the proficiency of various optimization methods, as they are characterized by high noise and ruggedness. The proposed approach reveals its suitability for performing the proposed task. In particular, experimental results confirm its capability to find the global argminum for all the considered benchmark problems and all the domain dimensions taken into account, thus providing an innovative and promising strategy for dealing with challenging optimization problems.

The paper is structured as follows: Section 2 provides an overview of surrogate models and, in particular, of evolutionary-based techniques for implementing surrogate models. Thereafter, Section 3 reviews genetic programming (Section 3-1) and Fuzzy Self-Tuning Particle Swarm Optimization (Section 3-2). Section 3-3 introduces the proposed technique for performing surrogate modeling by combining the two methods reviewed above. The benchmark functions and the experimental settings are described in Section 4, and the results of the experimental campaign are presented in Section 5. Finally, Section 6 summarizes the main contributions of the paper and provides directions for further research.

2- Literature Review

The field of optimization has grown rapidly during the past few decades, and many new theoretical, algorithmic, and computational contributions have been proposed to solve various problems [18–20]. Recent developments in the field of optimization methods can mainly be divided into deterministic and heuristic approaches. Deterministic methods are based on a solid mathematical formulation and are commonly used to address simple optimization problems where the computational effort grows only polynomially with the dimensionality of the problem [21]. However, if the problem is NP-hard, the same effort grows exponentially, and even small problems can become unsolvable using these methods as they usually get trapped in local minima [21]. On the other hand, heuristic approaches are based on search strategies that incorporate some form of randomness, which increases their robustness [22]. As a result, such algorithms are very effective in handling hard or ill-conditioned optimization problems [23].

As in this study we define a new heuristic SM-like approach by coupling genetic programming and particle swarm optimization, the remaining part of this section is organized as follows: Section 2-1 reviews existing works in which GP and PSO have been combined to address specific optimization tasks, and Section 2-2 summarizes important concepts and contributions concerning the use of GA for the definition of valuable surrogate models.

2-1- Genetic Programming Coupled with Particle Swarm Optimization

GP and PSO are two computational-based techniques, both taking inspiration from the progression of biological life. GP is a global optimization algorithm that simulates the principles of the Darwinian theory of evolution, while PSO can mimic cooperation between individuals in the same group using swarm intelligence and the exchange of experiences from generation to generation. Because of the peculiarities of GP and PSO, it is possible to effectively combine the global search characteristic of GP and the local search capability of PSO to avoid premature convergence and improve the quality of solutions [24].

Several works that have already exploited the integration of GP and PSO are worth mentioning: First, Poli et al. [25], where the use of GP aims to extend PSO models by including biology-inspired strategies and extending the physics of the particles by evolving, through the use of GP, optimal force generating equations for controlling the particles in a PSO. Feng et al. [24] propose a new method for the simultaneous establishment of a visco-elastic rock material model structure and the related parameters based on a hybrid genetic programming system with an improved particle swarm optimization algorithm. More specifically, GP explores the model's structure, and a modified version of PSO identifies the parameters in the provisional model. Besides, Kanemasa & Aiyoshi [26] adopt PSO as a heuristic optimization method, and they augment PSO by using GP as a meta-algorithm to solve the learning problem of automatically generating tuning rules for the parameters in the PSO algorithm.

Despite the satisfactory results achieved on specific problems, the performance of all the methods based on a combination of GP and PSO is highly-sensitive to the choice of the hyperparameters [27]. The method proposed in this study overcomes this issue by combining GP with a hyperparameter-free version of the PSO algorithm. We expect that the resulting algorithm will produce robust results over different benchmarks because its performance will not be affected by the manual choice of the hyperparameters. Moreover, the definition of an algorithm based on the hyperparameters-free version of PSO removes the practitioners' burden of choosing the suitable values of the hyperparameters based on the task at hand.

2-2- Surrogate Modeling

To overcome the difficulty arising from finding the global optimum in rugged fitness landscapes (i.e., those with a lot of local optima), a possibility is the use of a surrogate model (SM) [28]. The broad consensus on surrogate-based evolutionary frameworks is that the algorithm efficiency can be improved by replacing, as much as possible, costly objective functions with surrogates that are considerably less expensive to build and compute. In this manner, the overall computational burden of the evolutionary search can be significantly reduced, since the effort required to build and use the surrogates is much lower than in the traditional approach that directly couples the evolutionary algorithm with the costly objective functions [29, 30].

Constructing an SM has proven to be very efficient, and SMs are employed in many scientific fields in which evaluating the original fitness function would represent a major obstacle. For instance, SMs have been applied to rotor blade design and optimization [31], high-speed civil transport [32], airfoil shape optimization [33], and diffuser shape optimization [34].

Concerning fitness landscape analysis, the early approaches focused on building global surrogates (i.e., surrogates defined on the whole original domain) [30] with the scope of modeling the entire fitness landscape. Nonetheless, because of the curse of dimensionality [35], several contributions started to tackle the problem using local surrogate models (i.e., surrogates defined on a restriction of the original domain) [36-38] or a combination of global and local surrogate models [13, 39].

Lian et al. [40] proposed an enhancement for standard GAs using a local surrogate search to expedite their convergence. The model uses a GA to generate a population of individuals and rank them with a real function. Then, a gradient-based local search is performed on the SM to find new, promising solutions. Both GA and local search are alternately used under a trust-region framework until the optimum is found.

SMs have also been exploited in the context of GP. For example, in Kattan & Galvan [41], the authors propose an SM based on genetic programming and radial basis function networks (RBFN) called GP-RBFN Surrogate. In particular, they use GP as an optimization engine to evolve both the structure of a radial basis function (RBF) and its parameters to obtain an upgrade of the standard RBFN surrogate model. Specifically, GP receives an RBF as a primitive and tries to evolve a new RBF expression with improved width parameters, then the evolved RBFN is used as a surrogate model to improve the GP search.

Despite the existence of several studies [42, 43], the scientific literature lacks scalable algorithms able to build surrogate models for complex problems and present robust performance across different benchmarks [44]. This is mainly due to the use of techniques characterized by a high number of hyperparameters that cannot scale well with high-dimensional search space. To the best of our knowledge, this is the first study in which GP was coupled with a hyperparameters-free PSO to build surrogate models for complex functions (i.e., characterized by a rugged fitness landscape). Our objective is to combine the advantages of GP and PSO to construct a robust algorithm that, relying on a PSO variant without parameters, can achieve satisfactory performance (i.e., surrogate models able to represent salient features of the original problems) across different benchmark problems.

3- Research Methodology

This section introduces the tools that will be exploited for the development of our method: First, genetic programming is presented in Section 3.1; subsequently, Fuzzy Self-Tuning Particle Swarm Optimization, a variant of another bio-inspired algorithm, is described in Section 3.2. Afterward, taking advantage of these methods, a new surrogate modeling technique for dealing with challenging optimization problems is introduced in Section 3.3.

3-1- Genetic Programming

Genetic algorithms (GAs), introduced by Holland (1992) [45], are a technique for solving both constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution. The main idea at the base of these methods is to encode and iteratively improve a population of individuals that encode candidate solutions (of fixed-length) for a given problem. At each iteration, called a generation, candidate solutions (called parents) are selected and mutated in order to generate the components (called offsprings) of the subsequent generation.

Genetic programming (GP) overcomes the main limitation of GAs concerning the fixed length of the solutions by providing a method for finding a computer program of unspecified size and shape to solve (or approximately solve) a problem [15]. It starts with an initial population of unfit randomly generated programs (i.e., individuals), and each of them is evaluated in terms of how well it performs through the introduction of a specific objective function (called the fitness function), whose nature varies according to the task of the problem under consideration. Some individuals in the population will turn out to be fitter than others, and the Darwinian principle of survival of the fittest (elitism) and genetic operators are used to create a new (and improved) offspring population starting from the current population of programs. The process is repeated for a given number of iterations, and, at each step, it will produce populations that exhibit an improvement in the average fitness, which corresponds automatically to an improvement in the quality of the solution. The two genetic operators involved in the GP evolutionary process are crossover and mutation. The crossover operator induces the mixing of genetic material from two selected parents to produce two new offsprings: once the parents are chosen, the trees that encode the solutions that they represent are cut at some random position, and the subtrees generated through this cut are swapped between them, leading to the creation of two new individuals. Mutation is a genetic operator which is applied to a single individual: a random number in the interval $[0, 1]$ is generated with uniform probability and compared to a predetermined mutation rate. If the random number is greater than the mutation rate, no mutation is applied to the individual under consideration; otherwise, the program will be modified. If this is the case, a random point in the tree is chosen, and subsequently the subtree below this point is replaced by a new, randomly generated subtree.

3-2- Fuzzy Self-Tuning Particle Swarm Optimization (FST-PSO)

Particle swarm optimization is a bio-inspired algorithm that performs the task of searching for an optimal solution over a fixed solution space [46]. In this algorithm, P particles (i.e., candidate solutions) are placed in a bounded D -dimensional search space, and all the particles cooperate to recognize and converge to the (global) optimal solution of the optimization problem. In the standard formulation of PSO, the movements of the particles are guided by their own best-known position over the search space (cognitive factor, $c_{cog} \in \mathbb{R}^+$) as well as the entire swarm's best-known position (social factor, $c_{soc} \in \mathbb{R}^+$): when better positions are discovered, these are subsequently used to guide the subsequent movements of the swarm. This process is repeated for a predetermined number of epochs, eventually leading the swarm (as a whole) to move closer to the optimum of the fitness function.

Due to their relevance, the coefficients c_{cog} and c_{soc} together with the values of the maximum and minimum velocity $\vec{v}_{max}, \vec{v}_{min} \in \mathbb{R}^D$ of particles along each dimension of the search space, which are other important hyperparameters, are usually carefully selected according to the characteristics of the specific problem under consideration. As it is not possible to determine such values analytically, their selection commonly involves extensive trials.

To overcome this issue, a variant of the classic PSO framework, named Fuzzy Self-Tuning PSO (FST-PSO) that does not require any user settings has been introduced in Nobile et al. [17]. Indeed, by adding a Fuzzy Rule-Based System (FRBS) to each particle, FST-PSO permits a during-the-optimization and independent adjustment of each particle according to its performance and its distance from the position of the current global best particle. FRBSs are models based on the concepts of fuzzy sets [47] (also known as uncertainty sets and they allow coping with uncertainty, imprecision, and non-linearity by making a model more flexible to change. The FST-PSO algorithm can be summarized in four steps (an in-depth description, as well as the pseudo-code of the method, can be found in Nobile et al. [17]).

The starting point consists in randomly placing all the particles in the search space. Afterward, as in standard PSO, it proceeds by evaluating the fitness function for all the particles and updating the information about the current global and local bests. At this point, FST-PSO exploits FRBS to perform multiple Sugeno inferences, proposed by Takagi and Sugeno in 1985 [48] for developing a systematic approach to generating fuzzy rules from a given input-output dataset. A typical fuzzy rule in a Sugeno fuzzy model has the form:

if x is A and y is B then $z = f(x, y)$,

where A and B are fuzzy sets, and $f(x, y)$ is usually a polynomial in the input variables x and y . In the case of FST-PSO, the output polynomial depends on each particle's performance with respect to the previous iteration and its distance from the current global best in the swarm. Finally, the velocity and position of all particles are updated, and the algorithm repeats until the exhaustion of the fitness evaluations' budget and, as the result of the optimization, the best solution found by the swarm is returned.

3-3-GP-FST-PSO Surrogate Model

Consider a D -dimensional search space $S \subseteq \mathbb{R}^D$ and a function $f : \mathbb{R}^D \rightarrow \mathbb{R}$. A minimization (maximization) problem consists in finding the optimal $\bar{x} \in \mathbb{R}^D$ such that $f(\bar{x}) \leq f(x)$ ($f(\bar{x}) \geq f(x)$) $\forall x \in \mathbb{R}^D \setminus \{\bar{x}\}$. Therefore, the function f associates a correspondent measure to a candidate solution in the search space, inducing a hyper-surface Λ , called a fitness landscape, representing the quality of all the possible solutions in the search space. Especially when dealing with applications related to real-life problems, this landscape appears irregular and noisy and is also often characterized by the existence of a considerable number of local optima, which can lead the optimizer to premature convergence.

To deal with this kind of scenario, we combine the two techniques mentioned above, i.e. GP (3.1) and FST-PSO (3.2). The goal of the proposed method, named the GP-FST-PSO Surrogate Model (which in turn stands for Genetic Programming Fuzzy Self-Tuning Particle Swarm Optimization Surrogate Model), is the definition of a smooth function, easier to optimize with respect to the original one, that still maintains a fundamental property of the original rugged fitness landscape: the location of the argminimum.

To accomplish the creation of such a surrogate model, we consider a genetic programming approach: the surrogate function will be created by starting from an initial random population of functions and then applying genetic operators (mutation and crossover) to this population for a predetermined number of generations. The FST-PSO algorithm enters the picture for the definition of the loss function L . The loss function that evaluates the quality of a solution must implicitly reward solutions characterized by the same argminimum as the original fitness landscape, as the aim of an optimization problem consists in finding the coordinates where the function is minimized. Therefore, the loss utilized in this paper is computed as follows: when a surrogate function is created via the GP routine, its argminimum is computed through the FST-PSO algorithm. Such computation does not involve a considerable computational effort, on the contrary, finding the argminimum of GP-generated functions is a computationally easy task. Indeed, the population consists of smooth functions, as the only operators used for their definitions are mathematical elementary operators (summation, subtraction, multiplication, and division) and two other typical GP operators (SWAP and DUP) that, in any case, do not lead to any noise in the construction of a function. Once the argminimum is computed, it is passed as the argument of the benchmark noisy function and the result is used as a measure for defining the loss of the GP-FST-PSO Surrogate Modeling method. The loss L , therefore, can be defined as follow:

$$L = f(FST-PSO(\tilde{f})) \quad (1)$$

where \tilde{f} is a function generated through GP and f is the original benchmark function. This quantity is minimized with respect to all the individuals of the current generation, and the best result leads to the best individual of the generation under consideration. This choice ensures that the functions whose argminimum is near to the one of the original function are rewarded and survive through epochs. Finally, when the argminimum of the surrogate function reaches the

argminimum of the original noisy benchmark function, the loss reaches its minimum as the original benchmark function cannot return any lower values. The use of an evolutionary method, such as GP, prevents the search from falling into local minima and getting stuck there, as it allows big jumps in the solution space. In other words, at each step, new areas of this space can be explored and discovered.

As the goal of the proposed method consists not only in finding the correct location in the search space where the argminimum lies but also in approximating the fitness landscape's shape, the aforementioned loss considers an additional term. This term consists of the root mean squared error (RMSE) computed between the surrogate function and the original benchmark function over a set of points uniformly sampled from the search space. The RMSE is added to the aforementioned loss L and the whole quantity is minimized. Thus, both the correctness of the argminimum location and the fact that the surrogate function consistently represents the original one are considered. The final loss can be outlined as follows:

$$L = f(FST - PSO(\tilde{f})) + RMSE(f, \tilde{f}) \quad (2)$$

The algorithm that summarizes the aforementioned proposed optimization method is presented in Figure 1.

```

1: function GP-FST-PSO( $f$ , search_space, pop_sz, n_gen)
2:   sampled_points  $\leftarrow$  sample( $f$ , search_space)
3:   pop  $\leftarrow$  random_pop(pop_sz)
4:   for  $1 \leq n \leq n\_gen$  do
5:     offsp  $\leftarrow$  tournament(pop, FST-PSO)            $\triangleright$  crossover
6:     pop  $\leftarrow$  mutation(offsp)                    $\triangleright$  mutation
7:     pop  $\leftarrow$  min $_{\tilde{f} \in \text{offsp}}$  FST-PSO( $\tilde{f}$ )       $\triangleright$  elitism
8:   end for
9:   best  $\leftarrow$  min $_{\tilde{f} \in \text{pop}}$  FST-PSO( $\tilde{f}$ )
10:  return best
11: end function

```

Figure 1. GP-FST-PSO Surrogate Model pseudocode, where f is the benchmark function that must be optimized, search space is the search space domain relative to f , pop_sz is the size of the population, and n_gen is the number of generations considered.

The different steps of the proposed algorithm are also displayed in Figure 2. As one can see, the FST-PSO algorithm is employed in the evaluation of the individuals' fitness.

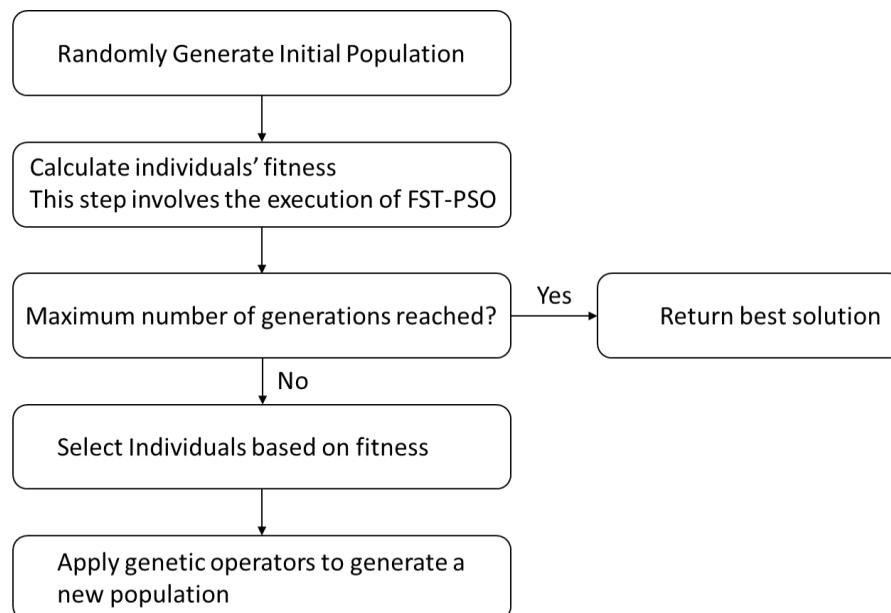


Figure 2. Flowchart of the proposed algorithm. The FST-PSO algorithm enters the picture for the definition of the fitness function

4- Experimental Settings

This section describes the benchmark functions used to validate the method (Sec. 4.1) and then presents the experimental setting (Sec. 4.2) needed to make the experiments completely reproducible. The code, for complete reproducibility of the proposed experiments, is available at ANONYMIZED LINK.

4-1-Benchmark Functions

The set of functions used to assess the proposed method is listed in Table 1. Here we summarize the main characteristics of the functions, such as their closed-form expression, their domain, and the value of their global minimum, together with the correspondent argminum. These functions (all widely used as benchmarks problems in the optimization framework [49, 50]) belong to a subset of the CEC 2005 test suite. They were selected because their structural characteristics of noise and ruggedness make them excellent candidates for studying the effectiveness of the proposed approach. The considered benchmark functions (difficult to optimize, non-convex, and multimodal) are Ackley, Alpine, Griewank, Xin-She Yang 2, Rastrigin, and Rosenbrock; all of them are characterized by various challenges and intrinsic difficulty. For example, Ackley has the global optimum located in a very small basin [51] and Griewank is characterized by a number of local minima that increases exponentially. In Rosenbrock, the global minimum is inside a long, narrow, parabolic-shaped flat valley: to find the valley is trivial, but to converge to the global minimum is difficult [51]. Moreover, all but the Rosenbrock function contain at least one trigonometric term, which provided a good way of imitating noise by introducing several local minima in the landscape.

Table 1. Closed expressions of the considered benchmark functions together with the domain, the location of the

Ackley	$f_{Ack}(\vec{x}) = 20 + e - 20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{d=1}^D x_d^2}\right) - \exp\left(\frac{1}{D}\sum_{d=1}^D \cos(2\pi x_d)\right)$
	Search Space: $[-30, +30]^D$
	Minimum: $f_{Ack}(\vec{0}) = 0$
Alpine	$f_{Alp}(\vec{x}) = \sum_{d=1}^D x_d \sin(x_d) + 0.1x_d $
	Search Space: $[-10, +10]^D$
	Minimum: $f_{Alp}(\vec{0}) = 0$
Griewank	$f_{Gri}(\vec{x}) = \frac{1}{4000} \sum_{d=1}^D x_d^2 - \prod_{d=1}^D \cos\left(\frac{x_d}{\sqrt{d}}\right) + 1$
	Search Space: $[-600, +600]^D$
	Minimum: $f_{Gri}(\vec{0}) = 0$
Rastrigin	$f_{Ras}(\vec{x}) = 10D + \sum_{d=1}^D (x_d^2 - 10\cos(2\pi x_d))$
	Search Space: $[-5.12, +5.12]^D$
	Minimum: $f_{Ras}(\vec{0}) = 0$
Rosenbrock	$f_{Ros}(\vec{x}) = \sum_{d=1}^{D-1} [10(x_d^2 - x_{d+1})^2 + (x_d - 1)^2]$
	Search Space: $[-5, +10]^D$
	Minimum: $f_{Ros}(\vec{1}) = 0$
Xin-She Yang N.2	$f_{Xin}(\vec{x}) = \sum_{d=1}^D x_d [\exp(\sum_{d=1}^D \sin(x_d^2))]^{-1}$
	Search Space: $[-2\pi, +2\pi]^D$
	Minimum: $f_{Xin}(\vec{0}) = 0$

4-2-Experimental Study

To make the experiments fully reproducible, this section describes the experimental settings (also reported in Table 2). For each benchmark function, 30 runs were performed to obtain statistically robust results. The initial population consists of 50 randomly generated individuals for problem dimensionality equal to 2 or 3, while it is equal to 100 for problem dimensionality equal to 4 (a larger number of individuals is chosen for higher dimensionality as the search for the minimum becomes more difficult when the dimensionality of the problem increases). Each initial individual is a random program of length 10, generated with the ramped half and half technique. Only the functions belonging to the function set introduced in Table 2 are considered, while the constants are selected from different intervals defined according to the domain and codomain boundaries of the benchmarks.

Table 2. Experimental settings. In the function set, DUP duplicates the element of a stack, SWAP swaps two elements at-top of the stack

PARAMETER	VALUE
FUNCTIONS SET	+,-,*,//, DUP, SWAP
NUMBER OF RUNS	30
FIRST GEN. INDIVIDUALS LENGTH	10
POPULATION SIZE ($D = 2, 3$)	50
POPULATION SIZE ($D = 4$)	100
NUMBER OF GENERATIONS	100
MUTATION RATE	0.2
SELECTION METHOD	Tournament of size 4
ELITISM	Best individuals survive

The linear GP exploited for evolving surrogate models is based on a stack architecture (see [52] for a detailed description). The algorithm adopted in this work considers an elitist strategy: in each generation, we maintain the individual with the best fitness. Concerning selection, we rely on a tournament operator. The mutation operator replaces, with probability $pm = 0.2$, an instruction (function or constant) with another random function or constant.

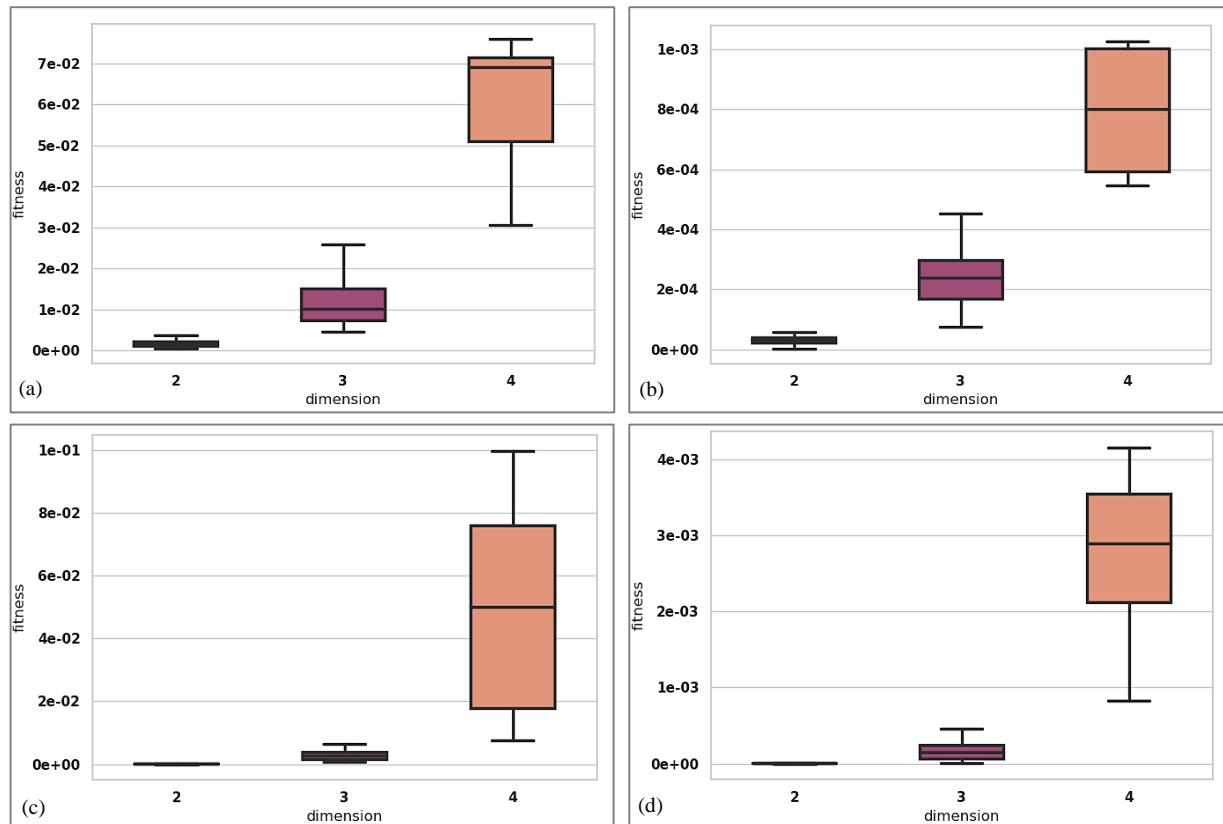
To assess the validity of the proposed method across the different benchmark functions, the fitness employed considers the value computed passing the argminum (computed using FST-PSO) of the surrogate function into the original benchmark function. If the benchmark function produces an output corresponding to its minimum (reported in Table 1), the surrogate model shares the same argminum of the original landscape. In this case, the surrogate model can provide a simpler option for solving the initial optimization task, free of its local optima and, consequently, of its irregularities and ruggedness.

5- Results

As stated in Section 3.3, the goal of this study is to assess the suitability of GP-FST-PSO for optimizing challenging landscapes. For each benchmark function described in Section 4.1, a statistical analysis of the fitness achieved by our method (computed over the 30 independent runs considered) is reported in Figure 3 for dimensions $D = 2$, $D = 3$, and $D = 4$. For the sake of clarity, the median value of the fitness is also reported, for all the considered problems' dimensionalities, in Table 3. Moreover, for $D = 2$, which allows the results to be displayed, some of the benchmark functions are displayed together with the correspondent GP-FST-PSO generated surrogate model (Figure 4). This visual representation allows the reader to better understand how the proposed method approximates the considered (noisy) landscapes.

Table 3. Median of the fitness values computed over the 30 runs performed for the considered benchmark functions in dimensions 2, 3, and 4, respectively

Ackley			Alpine		
2	3	4	2	3	4
1.57E-03	1.01E-02	6.91E-02	3.14E-05	2.37E-04	7.99E-04
Griewank			Xin-She Yang N.2		
2	3	4	2	3	4
5.18E-05	2.61E-03	5.00E-02	1.47E-04	1.33E-03	2.72E-01
Rastrig			Rosenbrock		
2	3	4	2	3	4
1.65E-06	1.47E-04	2.89E-03	1.49E-07	6.32E-04	4.48E-02



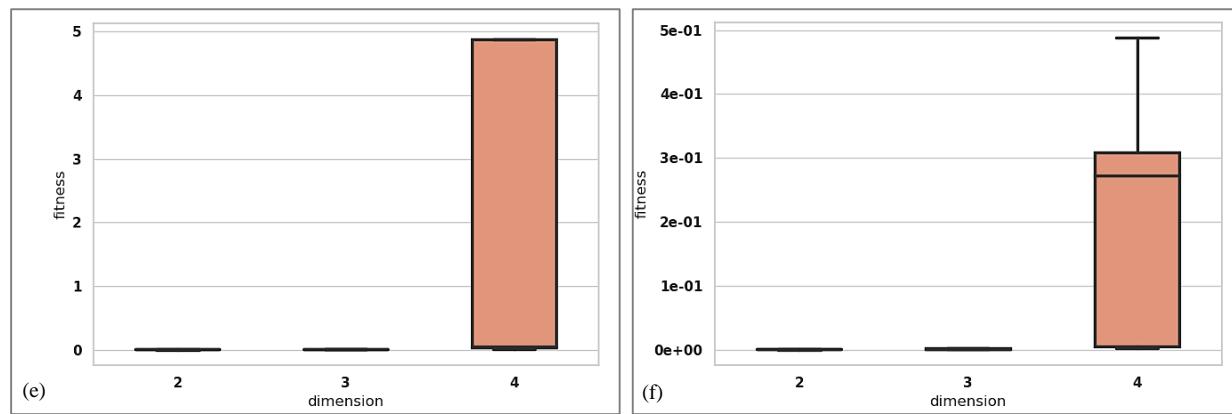
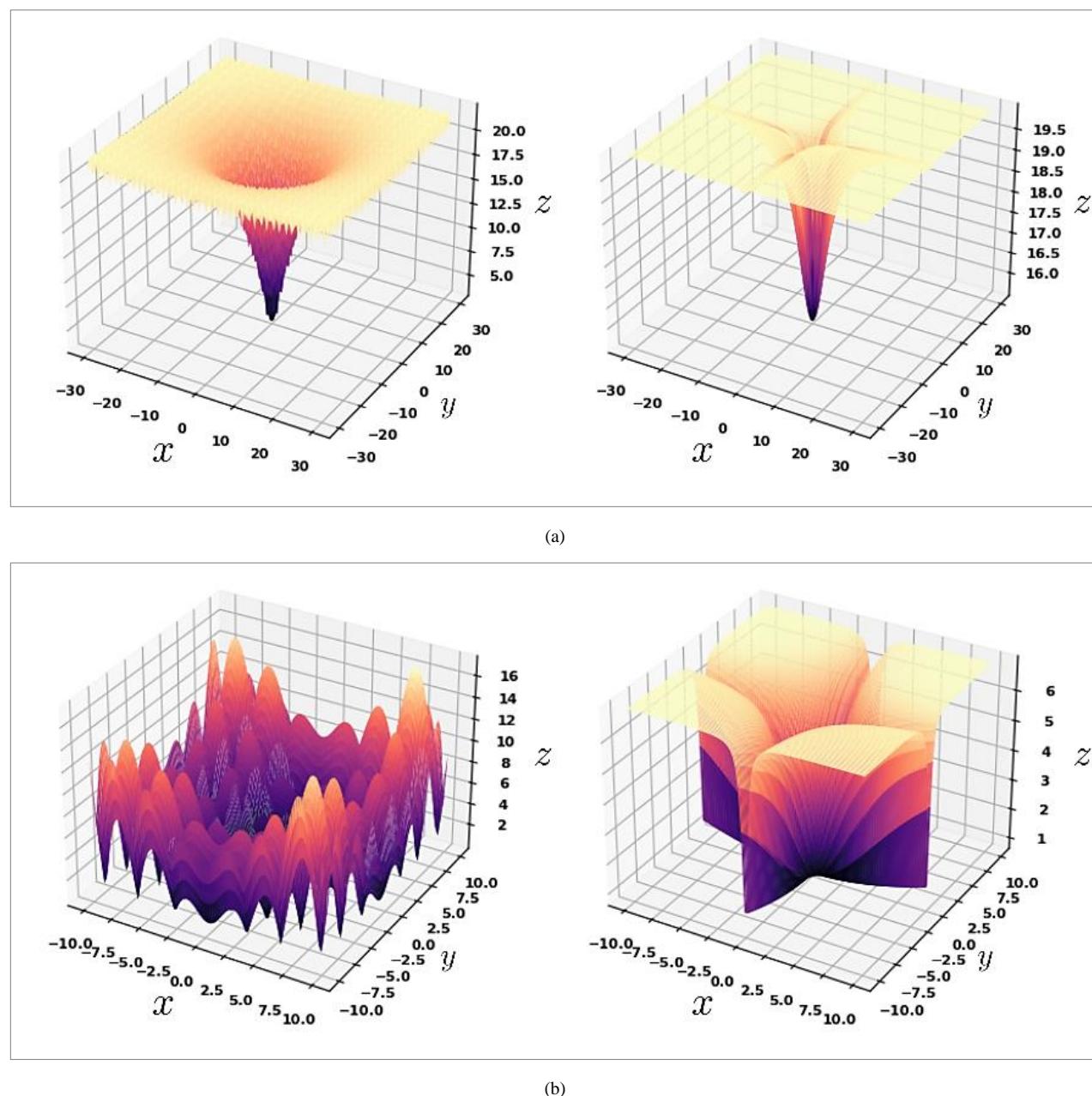
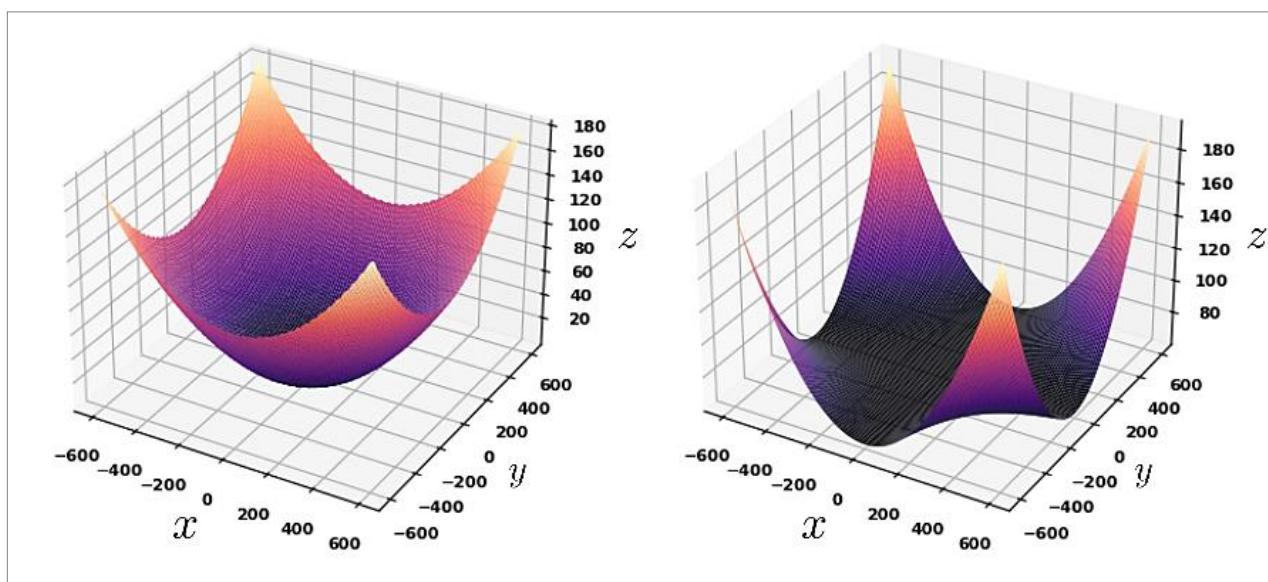
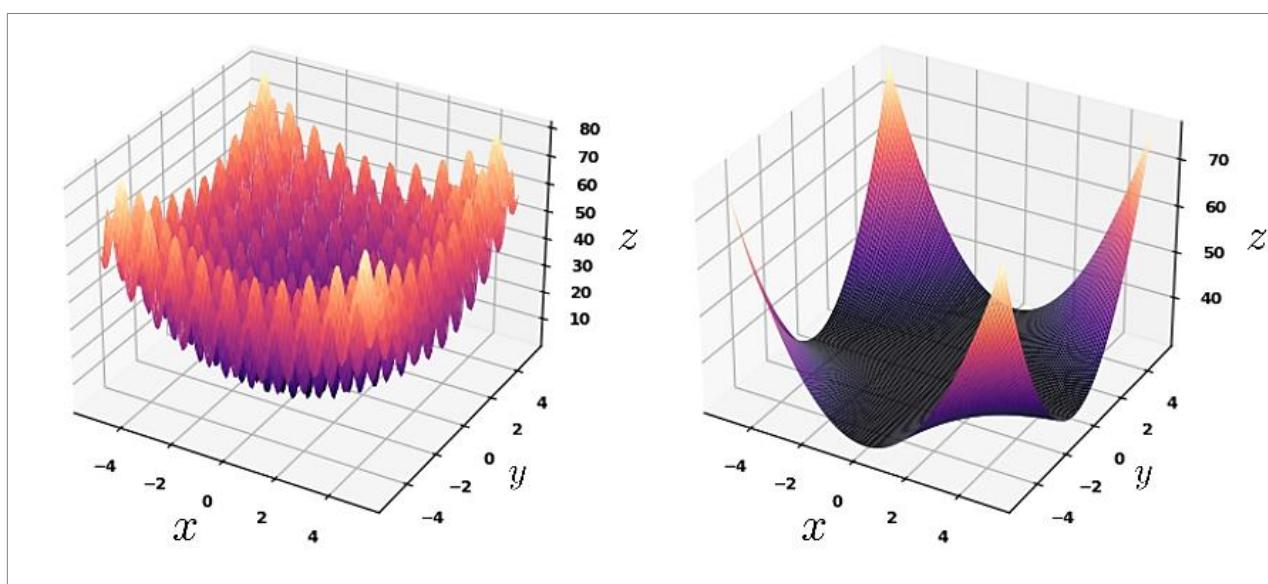


Figure 3. Boxplots of fitness distributions over the 30 independent runs performed for the considered benchmark functions considering $D = 2$, $D = 3$, and $D=4$. The black line inside the box represents the median. (a) Ackley, (b) Alpine, (c) Griewank, (d) Rastrigin, (e) Rosenbrock, (f) Xin-She Yang N.2.

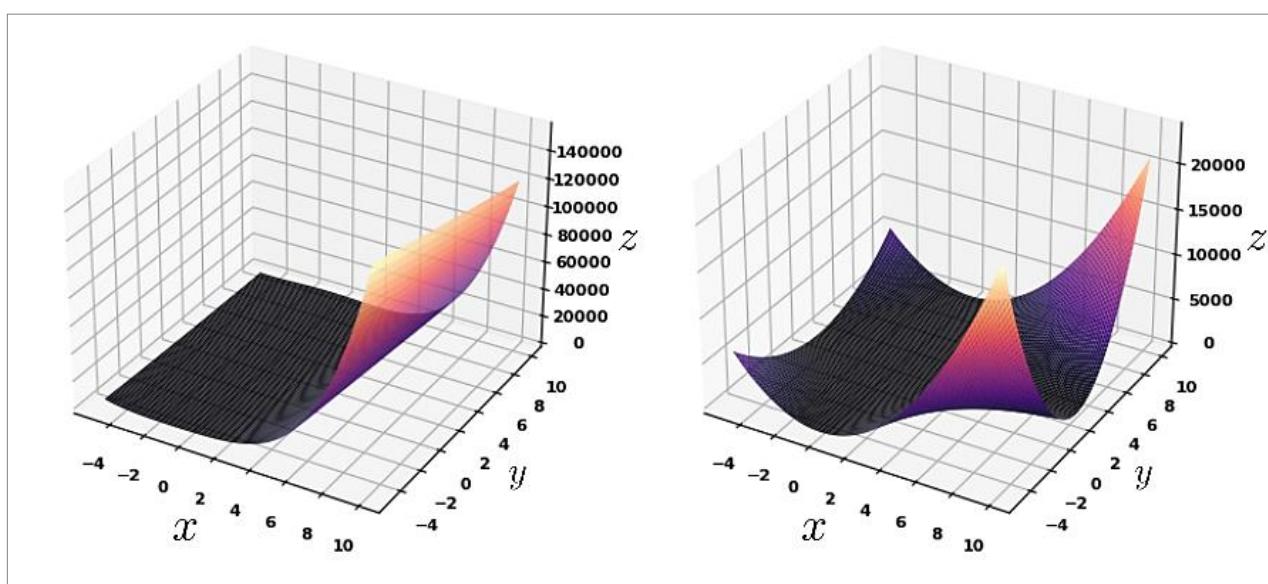




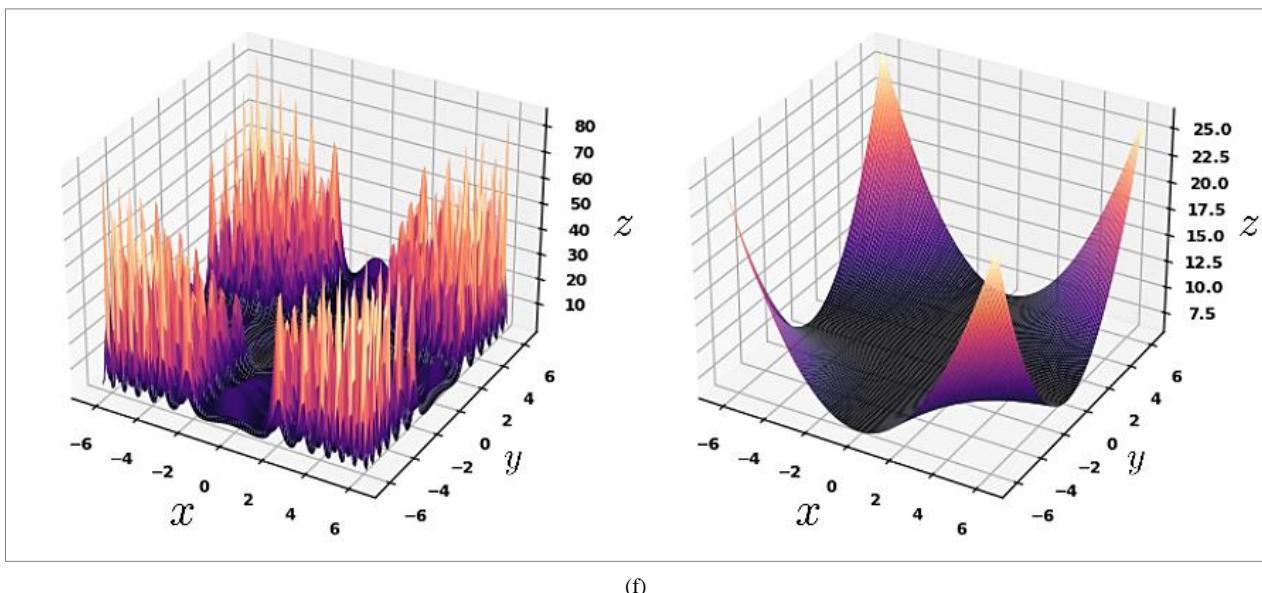
(c)



(d)



(e)



(f)

Figure 4. Example of $D = 2$ surrogate models of the benchmark functions considered, created by the GP-FST-PSO Surrogate Model. The original fitness landscape (on the left) is compared with the surrogate model (on the right). Benchmark functions considered are: (a) Ackley, (b) Alpine, (c) Griewank, (d) Rastrig, (e) Rosenbrock, and (f) Xin-She Yang N.2.

Focusing on the Ackley benchmark function, the fitness achieved by GP-FST-PSO is displayed via boxplots for dimension $D = 2$, $D = 3$, and, lastly, for dimension $D = 4$ in Figure 3-a. The fitness obtained by minimizing the surrogate function (for each of the considered problems' dimensionalities) suggests that the method leads to an excellent approximation of the global optimum of the original function (which is equal to 0). Moreover, experimental results show that the argminimum of the surrogate function corresponds to the minimum of the original fitness landscape, ensuring that the method does not get stuck in local optima. For dimension $D = 2$, it is also possible to observe the approximating function generated by our model: while the shapes of the original Ackley function and the surrogate model are very similar, the latter is devoid of the huge quantity of local optima that characterized Ackley.

Concerning the Alpine function, experimental results corroborate the suitability of the proposed method. Actually, the global minimum is achieved for all the considered dimensions (Figure 3-b), together with the correct argminimum that minimizes the considered function. Furthermore, these results are achieved after a few generations of the search process, thus suggesting that the proposed method can quickly determine a satisfactory approximation of the real function.

For the Griewank function, conclusions similar to those in the Alpine function case can be drawn (see Figure 3-c). In this situation as well, the proposed method correctly approximates the minimum and argminimum for $D = 2$, $D = 3$, and $D = 4$. Additionally, its convergence to a correct surrogate function is achieved after only a few generations. Figure 4-c provides an example of a surrogate function. As one can see, despite the similarity in terms of shape between the real and the generated function, the latter does not present the large number of local minima characterizing the Griewank function.

As for the Rastrig function, the method achieves excellent results (summarized in Figure 3-d) up to dimension $D = 4$, and similar observations as the ones discussed for the Alpine and Griewank functions can be drawn.

The proposed technique also proves itself successful when dealing with functions in which the minima are not located in $(\vec{0})$, as in the Rosenbrock function (results presented in Figure 3-e). In particular, the method correctly generates a surrogate function able to mimic both the argminimum location and the argminimum value, except for $D = 4$, where the method still gets close to the real argminimum location. It is indeed essential to stress that, even if the high fitness value seems to suggest that the method did not converge, this value is just a consequence of the very sharp curvature of the Rosenbrock function around the global optimum, and the obtained minimum is close to the real one. Concerning the Xin-She Yang N.2 function, the method is capable of providing a valid surrogate function for all the considered dimensions, as shown in Figure 3-f. Additionally, as shown in Figure 4-f, the surrogate model can extrapolate the relevant visual features of the benchmark, such as the location of the different peaks.

All in all, the experimental results demonstrate the suitability of the proposed method in correctly approximating rugged and multimodal fitness landscapes. In particular, for the majority of the benchmarks and dimensionality values considered, the proposed method is able to map the argmin of the surrogate model into the argmin of the original function. The visual analysis of the results (reported in Figure 4 for $D = 2$) complements the numerical findings: as one can see, GP-FST-PSO produces fitness landscapes that resemble the ones of the original functions but with smoother surfaces.

In other words, GP-FST-PSO achieves approximate functions that show the main properties (i.e., argmin and minimum value) of the original functions but without the presence of all the local optima characterizing the original counterparts. As a consequence, the exploration of these simplified landscapes through a heuristic algorithm would require less time and would be more effective than the exploration of the fitness landscapes of the original functions. Moreover, the proposed algorithm does not require a tedious and time-consuming hyperparameters tuning phase. Thanks to the specific setting-free PSO implementation, it is possible to execute the proposed GP-FST-PSO algorithm relying on the usage of the same GP parameters independently from the specific benchmark under consideration. This is one of the main contributions of this work facing the scientific literature, in which the use of SMs is usually accompanied by the introduction of several hyperparameters.

Last but not least, experimental results suggest that while the performance of the algorithm is robust across the considered dimensionality values, increasing the dimensionality of the problem results in a higher variance of the results and, in general, we can observe that for $D = 4$, the proposed technique reaches the poorest values of fitness. Still, the approximation obtained can be considered extremely good at least for three of the considered benchmarks, with an error smaller than 10^{-3} . Moreover, the argmin corresponds to the one of the original functions in the large majority of the considered benchmarks and problems' dimensionality. Concerning the number of local optima characterizing the original functions, it seems that the number of local optima is not a proxy for determining the difficulty of an optimization problem, as also discussed in the existing scientific literature [7]. As a consequence, an algorithm designed to simplify the original landscape through surrogate models should provide a good approximation of the original rugged function independently from the number of local optima characterizing the problem under analysis.

6- Conclusion

This paper investigates the possibility of solving challenging optimization problems by evolving, via GP, surrogate models that, despite being computationally easier to optimize, preserve the same location for the global optimum. To evaluate the fitness of the surrogate models generated during the evolutionary process, we exploited another evolutionary technique, i.e., FST-PSO, to assess the quality of the global minimum of the individual (the aforementioned surrogate function). The closer the surrogate global minimum gets to the real global minimum of the original noisy benchmark function, the more are the chances for the individual to survive to the next generation.

The proposed method, called the GP-FST-PSO Surrogate Model, has been tested over different functions, all of which are widely used as benchmark problems in the optimization framework. The method was also tested among different dimensions of the domain space to assert its validity in higher dimensions as well. The experimental results show that, in each of the considered benchmark problems, the GP-FST-PSO Surrogate Model achieves a satisfactory outcome. Overall, GP-FST-PSO was able to generate surrogate models that mimic the shape of the original function and, most importantly, that are characterized by the same location for the global argminimum. Excellent results are achieved for all the considered dimensions and benchmarks, thus strengthening the suitability of the proposed method for simplifying the exploration of rugged landscapes.

This work represents the first attempt to create a GP-based surrogate model exploiting the FST-PSO technique to evaluate fitness. Considering the promising results obtained in this first analysis, this work paves the way for multiple possible future developments focused on improving the benefits provided by this combination. Among the different possibilities, one could rely on a more advanced GP technique for the evolution of the population instead of classic linear GP. Further, from a more theoretical perspective, one could investigate the behavior of the GP-FST-PSO Surrogate Model when applied to a less regular function (e.g., continuous or L^1) than the ones considered in the present work (all belonging to the C^k class, with $k \geq 1$). Finally, while the majority of the works relying on SMs considers single-objective optimization, real-world problems are characterized, most of the time, by multiple objectives. Thus, we plan to extend this work to cover multi-objective optimization problems.

7- Declarations

7-1-Author Contributions

Conceptualization, G.P., G.M. and M.C.; methodology, G.P., G.M. and M.C.; software, G.P. and G.M.; validation, G.P., G.M. and M.C.; formal analysis, G.P. and G.M.; investigation, G.P., G.M. and M.C.; resources, M.C.; writing—original draft preparation, G.P. and G.M.; writing—review and editing, G.P., G.M. and M.C.; visualization, G.P.; supervision, M.C.; project administration, M.C.; funding acquisition, M.C. All authors have read and agreed to the published version of the manuscript.

7-2-Data Availability Statement

Data sharing is not applicable to this article.

7-3- Funding

This work was funded by national funds through the FCT - Foundation for Science and Technology, I.P., within the scope of the project UIDB/04152/2020 - Centro de Investigação em Gestão de Informação (MagIC)/NOVA IMS.

7-4- Institutional Review Board Statement

Not applicable.

7-5- Informed Consent Statement

Not applicable.

7-6- Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

8- References

- [1] Peres, F., & Castelli, M. (2021). Combinatorial optimization problems and metaheuristics: Review, challenges, design, and development. *Applied Sciences* (Switzerland), 11(14), 6449. doi:10.3390/app11146449.
- [2] Stadler, P.F. (2002). Fitness landscapes. *Biological Evolution and Statistical Physics*. Lecture Notes in Physics, 585, Springer, Berlin, Germany. doi:10.1007/3-540-45692-9_10.
- [3] Merz, P. (2004). Advanced fitness landscape analysis and the performance of memetic algorithms. *Evolutionary Computation*, 12(3), 303–325. doi:10.1162/1063656041774956.
- [4] Richter, H. (2013). Dynamic Fitness Landscape Analysis. *Evolutionary Computation for Dynamic Optimization Problems. Studies in Computational Intelligence*, 490. Springer, Berlin, Germany. doi:10.1007/978-3-642-38416-5_11.
- [5] Tan, Z., Li, K., & Wang, Y. (2021). Differential evolution with adaptive mutation strategy based on fitness landscape analysis. *Information Sciences*, 549, 142–163. doi:10.1016/j.ins.2020.11.023.
- [6] Vanneschi, L., Castelli, M., & Manzoni, L. (2011). The K landscapes. Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation. doi:10.1145/2001576.2001773.
- [7] Horn, J., & Goldberg, D. E. (1995). Genetic Algorithm Difficulty and the Modality of Fitness Landscapes. *Foundations of genetic algorithms*, Elsevier, Amsterdam, Netherlands. doi:10.1016/b978-1-55860-356-1.50016-9.
- [8] Zou, F., Chen, D., Liu, H., Cao, S., Ji, X., & Zhang, Y. (2022). A survey of fitness landscape analysis for optimization. *Neurocomputing*, 503, 129–139. doi:10.1016/j.neucom.2022.06.084.
- [9] Li, W., Li, S., Chen, Z., Zhong, L., & Ouyang, C. (2019). Self-feedback differential evolution adapting to fitness landscape characteristics. *Soft Computing*, 23(4), 1151–1163. doi:10.1007/s00500-017-2833-y.
- [10] Jin, Y. (2011). Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2), 61–70. doi:10.1016/j.swevo.2011.05.001.
- [11] Eason, J., & Cremaschi, S. (2014). Adaptive sequential sampling for surrogate model generation with artificial neural networks. *Computers and Chemical Engineering*, 68, 220–232. doi:10.1016/j.compchemeng.2014.05.021.
- [12] Lew, T. L., Spencer, A. B., Scarpa, F., Worden, K., Rutherford, A., & Hemez, F. (2006). Identification of response surface models using genetic programming. *Mechanical Systems and Signal Processing*, 20(8), 1819–1831. doi:10.1016/j.ymssp.2005.12.003.
- [13] Zhou, Z., Ong, Y. S., Nair, P. B., Keane, A. J., & Lum, K. Y. (2007). Combining global and local surrogate models to accelerate evolutionary optimization. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 37(1), 66–76. doi:10.1109/TSMCC.2005.855506.
- [14] Manzoni, L., Papetti, D. M., Cazzaniga, P., Spolaor, S., Mauri, G., Besozzi, D., & Nobile, M. S. (2020). Surfing on fitness landscapes: A boost on optimization by Fourier surrogate modeling. *Entropy*, 22(3), 285. doi:10.3390/e22030285.
- [15] Koza, J. R. (1994). Genetic programming as a means for programming computers by natural selection. *Statistics and Computing*, 4(2), 87–112. doi:10.1007/BF00175355.
- [16] Talbi, E. G. (2009). *Metaheuristics: From Design to Implementation*. Metaheuristics: From Design to Implementation. John Wiley & Sons, Hoboken, United States. doi:10.1002/9780470496916.

- [17] Nobile, M. S., Cazzaniga, P., Besozzi, D., Colombo, R., Mauri, G., & Pasi, G. (2018). Fuzzy Self-Tuning PSO: A settings-free algorithm for global optimization. *Swarm and Evolutionary Computation*, 39, 70–85. doi:10.1016/j.swevo.2017.09.001.
- [18] Vanneschi, L., Castelli, M., & Silva, S. (2014). A survey of semantic methods in genetic programming. *Genetic Programming and Evolvable Machines*, 15(2), 195–214. doi:10.1007/s10710-013-9210-0.
- [19] Pan, J. S., Hu, P., Snášel, V., & Chu, S. C. (2022). A survey on binary metaheuristic algorithms and their engineering applications. *Artificial Intelligence Review*, 1–67. doi:10.1007/s10462-022-10328-9.
- [20] Lagaros, N. D., Kournoutsos, M., Kallioras, N. A., & Nordas, A. N. (2023). Constraint handling techniques for metaheuristics: a state-of-the-art review and new variants. *Optimization and Engineering*, 1–48. doi:10.1007/s11081-022-09782-9.
- [21] Lin, M. H., Tsai, J. F., & Yu, C. S. (2012). A review of deterministic optimization methods in engineering and management. *Mathematical Problems in Engineering*, 2012. doi:10.1155/2012/756023.
- [22] Sörensen, K., Sevaux, M., & Glover, F. (2018). A History of Metaheuristics. *Handbook of Heuristics*. Springer, Cham, Switzerland. doi:10.1007/978-3-319-07124-4_4.
- [23] Papila, N., Shyy, W., Griffin, L., & Dorney, D. J. (2002). Shape optimization of supersonic turbines using global approximation methods. *Journal of Propulsion and Power*, 18(3), 509–518. doi:10.2514/2.5991.
- [24] Feng, X. T., Chen, B. R., Yang, C., Zhou, H., & Ding, X. (2006). Identification of visco-elastic models for rocks using genetic programming coupled with the modified particle swarm optimization algorithm. *International Journal of Rock Mechanics and Mining Sciences*, 43(5), 789–801. doi:10.1016/j.ijrmms.2005.12.010.
- [25] Poli, R., Langdon, W.B., Holland, O. (2005). Extending Particle Swarm Optimisation via Genetic Programming. *Genetic Programming. EuroGP 2005, Lecture Notes in Computer Science*, 3447, Springer, Berlin, Germany. doi:10.1007/978-3-540-31989-4_26.
- [26] Kanemasa, M., & Aiyoshi, E. (2014). Algorithm tuners for PSO methods and genetic programming techniques for learning tuning rules. *IEEJ Transactions on Electrical and Electronic Engineering*, 9(4), 407–414. doi:10.1002/tee.21986.
- [27] Cavazzuti, M. (2017). Deterministic Optimization. In *Stochastic Process Optimization using Aspen Plus®*. CRC Press, Boca Raton, United States. doi:10.1201/9781315155739-2.
- [28] Bhosekar, A., & Ierapetritou, M. (2018). Advances in surrogate based modeling, feasibility analysis, and optimization: A review. *Computers and Chemical Engineering*, 108, 250–267. doi:10.1016/j.compchemeng.2017.09.017.
- [29] Emmerich, M., Giotis, A., Özdemir, M., Bäck, T., Giannakoglou, K. (2002). Metamodel—Assisted Evolution Strategies. *Parallel Problem Solving from Nature — PPSN VII, PPSN 2002, Lecture Notes in Computer Science*, 2439. Springer, Berlin, Germany. doi:10.1007/3-540-45712-7_35.
- [30] Ulmer, H., Streichert, F., & Zell, A. (n.d.). Evolution strategies assisted by Gaussian processes with improved preselection criterion. *The 2003 Congress on Evolutionary Computation*, 2003. CEC'03. doi:10.1109/cec.2003.1299643.
- [31] Booker, A.J., Dennis, J.E., Frank, P.D., Serafini, D.B., & Torczon, V. (1998). Optimization Using Surrogate Objectives on a Helicopter Test Example. *Computational Methods for Optimal Design and Control. Progress in Systems and Control Theory*, 24. Birkhäuser, Boston, United States. doi:10.1007/978-1-4612-1780-0_3.
- [32] Knill, D. L., Giunta, A. A., Baker, C. A., Grossman, B., Mason, W. H., Haftka, R. T., & Watson, L. T. (1999). Response surface models combining linear and euler aerodynamics for supersonic transport design. *Journal of Aircraft*, 36(1), 75–86. doi:10.2514/2.2415.
- [33] Rai, M., Madavan, N., & Huber, F. (2000). Improving the unsteady aerodynamic performance of transonic turbines using neural networks. *38th Aerospace Sciences Meeting and Exhibit*. doi:10.2514/6.2000-169.
- [34] Madsen, J. I., Shyy, W., & Haftka, R. T. (2000). Response surface techniques for diffuser shape optimization. *AIAA Journal*, 38, 1512–1518. doi:10.2514/3.14576.
- [35] Donoho, D. L. (2000). High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture*, 1(2000), 32.
- [36] Giannakoglou, K. C. (2002). Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. *Progress in Aerospace Sciences*, 38(1), 43–76. doi:10.1016/S0376-0421(01)00019-7.
- [37] Ong, Y. S., Nair, P. B., & Keane, A. J. (2003). Evolutionary optimization of computationally expensive problems via surrogate modeling. *AIAA Journal*, 41(4), 687–696. doi:10.2514/2.1999.
- [38] Tong, H., Huang, C., Minku, L. L., & Yao, X. (2021). Surrogate models in evolutionary single-objective optimization: A new taxonomy and experimental study. *Information Sciences*, 562, 414–437. doi:10.1016/j.ins.2021.03.002.

- [39] Zhou, Z., Ong, Y. S., Lim, M. H., & Lee, B. S. (2007). Memetic algorithm using multi-surrogates for computationally expensive optimization problems. *Soft Computing*, 11(10), 957–971. doi:10.1007/s00500-006-0145-8.
- [40] Lian, Y., Liou, M. S., & Oyama, A. (2004). An enhanced evolutionary algorithm with a surrogate model. Proceedings of genetic and evolutionary computation conference, 26-30 June, 2004, Seattle, United States.
- [41] Kattan, A., & Galvan, E. (2012). Evolving radial basis function networks via GP for estimating fitness values using surrogate models. 2012 IEEE Congress on Evolutionary Computation. doi:10.1109/cec.2012.6256108.
- [42] Liu, Q., Wu, X., Lin, Q., Ji, J., & Wong, K. C. (2021). A novel surrogate-assisted evolutionary algorithm with an uncertainty grouping based infill criterion. *Swarm and Evolutionary Computation*, 60, 100787. doi:10.1016/j.swevo.2020.100787.
- [43] Wu, M., Wang, L., Xu, J., Hu, P., & Xu, P. (2022). Adaptive surrogate-assisted multi-objective evolutionary algorithm using an efficient infill technique. *Swarm and Evolutionary Computation*, 75, 101170. doi:10.1016/j.swevo.2022.101170.
- [44] Tang, J., Wang, H., & Xiong, L. (2023). Surrogate-assisted multi-objective optimization via knee-oriented Pareto front estimation. *Swarm and Evolutionary Computation*, 77, 101252. doi:10.1016/j.swevo.2023.101252.
- [45] Holland, J. H. (1992). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT Press, Cambridge, United States. doi:10.7551/mitpress/1090.001.0001.
- [46] Kennedy, J., & Eberhart, R. (n.d.). Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks. doi:10.1109/icnn.1995.488968.
- [47] Zadeh, L. A., Klir, G. J., & Yuan, B. (1996). Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems. Advances in Fuzzy Systems — Applications and Theory, World Scientific Publishing, Singapore. doi:10.1142/2895.
- [48] Takagi, T., & Sugeno, M. (1985). Fuzzy Identification of Systems and Its Applications to Modeling and Control. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-15(1), 116–132. doi:10.1109/TSMC.1985.6313399.
- [49] Jamil, M., & Yang, X. S. (2013). A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2), 150–194. doi:10.1504/IJMMNO.2013.055204.
- [50] Liu, J., & Bailey, J. (2019). AI 2019: Advances in Artificial Intelligence. Lecture Notes in Computer Science, 2-5 December, 2015, Adelaide, Australia. doi:10.1007/978-3-030-35288-2.
- [51] Plevris, V., & Solorzano, G. (2022). A Collection of 30 Multidimensional Functions for Global Optimization Benchmarking. *Data*, 7(4), 46. doi:10.3390/data7040046.
- [52] Perkis, T. (1994). Stack-based genetic programming. Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence. doi:10.1109/icec.1994.350025.