

# Shaping and Dilating the Fitness Landscape for Parameter Estimation in Stochastic Biochemical Models

Marco S. Nobile <sup>1,2,3</sup> , Daniele M. Papetti <sup>4</sup> , Simone Spolaor <sup>4,5</sup> , Paolo Cazzaniga <sup>6</sup> , Luca Manzoni <sup>7\*</sup> 

<sup>1</sup> Department of Environmental Sciences, Informatics and Statistics – Ca' Foscari University of Venice, Italy

<sup>2</sup> Department of Industrial Engineering & Innovation Sciences – Eindhoven University of Technology, Eindhoven, The Netherlands

<sup>3</sup> Bicocca Bioinformatics, Biostatistics and Bioimaging Research Center (B4), Italy

<sup>4</sup> Department of Informatics, Systems and Communication – University of Milano-Bicocca, Milan, Italy

<sup>5</sup> Microsystems, Department of Mechanical Engineering – Eindhoven University of Technology, Eindhoven, The Netherlands

<sup>6</sup> Department of Human and Social Sciences – University of Bergamo, Italy

<sup>7</sup> Department of Mathematics and Geosciences, University of Trieste, Italy

\* Correspondence: lmanzoni@units.it

**Featured Application:** Potential applications of this work regard the parameter estimation and the resulting analyses of complex biochemical systems characterized by a stochastic behavior, which allows for elucidating the unknown underlying mechanisms.

**Abstract:** Parameter Estimation (PE) of biochemical reactions is one of the most challenging tasks in Systems Biology, given the pivotal role of these kinetic constants in driving the behavior of biochemical systems. PE is a non-convex, multi-modal, and non-separable optimization problem, whose fitness landscape is unknown; moreover, the quantities of the biochemical species appearing in the system can be low, making biological noise a non-negligible phenomenon and mandating the use of stochastic simulation. Finally, the values of the kinetic parameters typically follow a log-uniform distribution, thus the optimal solutions are situated in the lowest orders of magnitude of the search space. In this work, we further elaborate on a novel approach to address the PE problem based on a combination of adaptive swarm intelligence and Dilation Functions (DFs). DFs require prior knowledge of the characteristics of the fitness landscape; therefore, we leverage an alternative solution to evolve optimal DFs. On top of this approach, we introduce surrogate Fourier modeling to simplify the PE, by producing a smoother version of the fitness landscape that excludes the high frequency components of the fitness function. Our results show that the PE exploiting evolved DFs has performance comparable to the PE run with a custom DF. Moreover, surrogate Fourier modeling allows for improving the convergence speed. Finally, we discuss some open problems related to the scalability of our methodology.

**Keywords:** parameter estimation; fitness landscape manipulation; stochastic simulation; biochemical models; Fourier surrogate modeling; dilation functions.

**Citation:** Nobile, M.S.; Papetti, D.M.; Spolaor, S.; Cazzaniga P., Manzoni, L. Shaping and dilating the fitness landscape for parameter estimation in stochastic biochemical models. *Appl. Sci.* **2022**, *1*, 0. <https://doi.org/>

Received:

Accepted:

Published:

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Copyright:** © 2022 by the authors. Submitted to *Appl. Sci.* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The Parameter Estimation (PE) problem consists in estimating a reliable kinetic parameterization of a biochemical model, and represents one of the most complex but fundamental tasks in Systems Biology, since kinetic constants are necessary to simulate the dynamics of the system under investigation. The inference of a good parameterization thus allows for validating the model, and formulating predictions on the emergent behavior of the system in perturbed conditions [1].

Similarly to other real-world optimization problems, PE is typically characterized by many local optima that impede reaching the global optimum, due to premature convergence of any optimization meta-heuristics. Moreover, when dealing with the PE of

stochastic biochemical models, the multidimensional fitness landscape is rugged, making the optimization process even harder and time consuming [2–4].

Many works proposed solutions focused on the simplification of the optimization task by exploiting specific transformation functions to re-map the original candidate solutions into a modified search space, or by using “smoothed” surrogate fitness landscapes [5]. The manipulation of the fitness landscape can be realized by exploiting the so-called Dilation Functions (DFs) [6], which can “compress” and “dilate” localized areas of the search space, or other approaches such as the Shrinking Space Technique [7–9], which alters the search space to concentrate the search on specific areas, and the Space Transformation Search, [10], which transforms the original search space and simultaneously evaluates the solutions in both the original and transformed spaces. The generation of surrogate models has been shown to be particularly advantageous when algorithms that generally require a huge number of fitness evaluations to converge to the optimal solution—e.g., evolutionary computation, swarm intelligence—are used to tackle the optimization problem. As a matter of fact, the surrogate model is based on an approximated fitness function that can be evaluated with a reduced computational effort [11]. Multi-modal, rugged, and noisy fitness landscapes—as those typically associated with the PE problem of stochastic biochemical models—can be efficiently tackled with a surrogate modeling method called surF [12], which exploits the discrete Fourier transform. While surF is applied here on the landscape induced by a *stochastic* model, it can clearly be applied to a deterministic landscape, as done, for example, in [13] and [12]. A peculiarity of surF is that the number of low frequency spectral coefficients used for the inverse Fourier transform can be set to adjust the smoothness of the created surrogate model.

In [5], DFs and surF were coupled with FST-PSO [14], a settings-free version of Particle Swarm Optimization (PSO) [15], to investigate the PE problem of biochemical systems. The transformation of the candidate solutions operated by the DFs together with the surrogate model obtained with surF allowed for improving the optimization process carried out with FST-PSO, which altogether provided good quality results for the PE problem while reducing the number of fitness evaluations required.

In this paper, we extend the work presented in [5], by extensively investigating the impact of (1) DFs [6] and (2) the combination of DFs and surF [12] on the performance and the results of the PE of stochastic biochemical models. Moreover, we tackle a limitation of these approaches that typically require specific knowledge about the features of the fitness landscape of the optimization problem under investigation. To this aim, we included in the PE problem an additional parameter to be estimated, which encodes the number of low frequency Fourier coefficients retained to build the surrogate with surF, and we automatically derive an optimal DF utilizing a separate optimization process. Our results highlight the advantages of using these approaches when dealing with the PE problem and reveal some limitations regarding the optimization of the DFs.

The paper is structured as follows. In Section 2 we introduce the formalism used to define reaction-based models of biochemical systems, the algorithm employed for the stochastic simulations of the dynamics, the meta-heuristics exploited for the optimization process, the concept of DF, and the method used to create surrogates of the fitness landscapes. In Section 3 we present the results related to the investigation of the effect of using DF to solve the PE problem, and the performance of DFs when coupled with surF. Section 4 provides some final remarks and directions for future works.

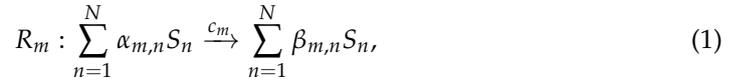
## 2. Materials and Methods

### 2.1. Reaction-Based Modeling and Stochastic Simulation Algorithm

Considering the stochastic formulation of chemical kinetics [16], we can define a Reaction-Based Model (RBM) by specifying two disjoint sets:

- the set  $\mathcal{S} = \{S_1, \dots, S_N\}$  of molecular species;
- the set  $\mathcal{R} = \{R_1, \dots, R_M\}$  of the biochemical reactions describing all interactions among the species in  $\mathcal{S}$ .

In RBMs, a biochemical reaction is defined as:



where  $\alpha_{m,n}, \beta_{m,n} \in \mathbb{N}$  indicate the stoichiometric coefficients associated with the  $n$ -th reactant and with the  $n$ -th product of the  $m$ -th reaction, respectively. The value  $c_m \in \mathbb{R}^+$ , associated to reaction  $R_m$ , is named stochastic (kinetic) constant and encompasses all its chemical-physical properties.

Although deterministic simulations of mechanistic RBMs allow for reproducing the dynamic behavior of biochemical systems [17], only stochastic simulations of RBMs can reproduce the emerging effects due to biological noise, which plays an important role when the molecular species occur in low amounts within the system [18].

The Stochastic Simulation Algorithm (SSA) [16] is a seminal procedure that allows for generating an exact realization of the temporal evolution of a RBM. SSA requires that (i) all reactions involving the chemical species must take place in a single volume, characterized by constant physical conditions (e.g., pressure, temperature) throughout the simulation; (ii) the molecules must be uniformly distributed within the volume, which is then considered well-stirred; (iii) the (discrete) quantity of each molecular species is denoted as an integer number  $x_n \in \mathbb{N}$ , for each  $n = 1, \dots, N$ .

The state of the system at time  $t$  is denoted in SSA by the vector  $\mathbf{x} = \mathbf{x}(t) \equiv (x_1(t), \dots, x_N(t))$ , and its content is updated according to the execution of reactions. To be more precise, at each step of the simulation, SSA identifies the reaction that will take place in the time interval  $[t, t + \tau)$  according to the probability of each reaction to occur in the infinitesimal time step  $[t, t + dt)$ , which is proportional to the so-called propensity function of each reaction. The propensity functions are calculated as  $a_m(\mathbf{x}) = c_m d_m(\mathbf{x})$ , where  $d_m(\mathbf{x})$  correspond to the number of distinct combinations of the reactant molecules involved in reaction  $R_m$ , in the current state of the system  $\mathbf{x}$ .

Successively, SSA calculates the waiting time  $\tau$  before the reaction is executed, as follows:

$$\tau = \frac{1}{a_0(\mathbf{x})} \ln\left(\frac{1}{\zeta_1}\right),$$

where  $\zeta_1$  is a random number sampled from the uniform distribution in  $(0, 1)$ , and  $a_0(\mathbf{x}) = \sum_{m=1}^M a_m(\mathbf{x})$ .  $m$  denotes the index of the reaction to fire, and corresponds to the smallest integer in the interval  $[1, M]$  such that

$$\sum_{m'=1}^m a_{m'}(\mathbf{x}) > \zeta_2 a_0(\mathbf{x}),$$

where  $\zeta_2$  is a random number sampled in from the uniform distribution in  $[0, 1]$ . The interested reader is referred to [16] for more details about SSA.

It is worth noting that the vector  $\mathbf{c} = (c_1, \dots, c_M)$  of stochastic constants governs the behavior of the system, since it affects the calculation of the propensity functions of the reactions. Unfortunately, the values in  $\mathbf{c}$  are hard (or even impossible) to measure with specific lab experiments; nonetheless, such fundamental parameters can be estimated by exploiting Computational Intelligence methods [19].

In this work, the PE problem for the inference of  $\mathbf{c}$  is tackled by assuming that the sets  $\mathcal{S}$  and  $\mathcal{R}$ , and of the vector  $\mathbf{x}(0)$  denoting the molecular amounts of the species present in the RBM at time  $t = 0$ , are known. We also assume that experimental data (e.g., the amounts of a subset of molecular species measured at some time points  $\mathbf{t} = \{t_1, \dots, t_K\}$ ) is available.

The PE consists in identifying the vector  $\mathbf{c}$  of stochastic constants that allow for obtaining a simulated dynamics that resembles the behavior of the system as observed in

the experimental data. Formally, the PE problem can be re-stated as the minimization of the following fitness function:

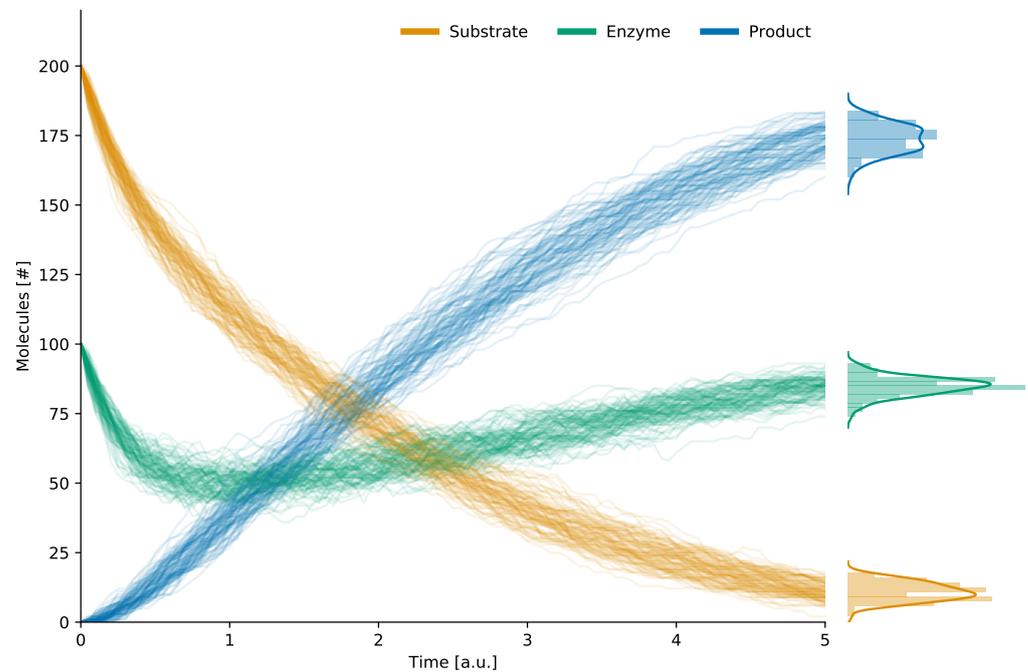
$$f(\mathbf{c}) = \sum_{k=1}^K \sqrt{\sum_{n=1}^N (X_n^c(t_k) - O_n(t_k))^2}, \quad (2)$$

where  $X_n^c(t_k)$  is the simulated amount of the species  $S_n$  at time  $t_k$ , obtained with the putative parameterization  $\mathbf{c}$ , and  $O_n(t_k)$  is the experimental (target) amount of  $S_n$  measured at time  $t_k$ .

Note that, due to the stochasticity of SSA simulations, two or more evaluations of the fitness of the same parameterization  $\mathbf{c}$ , considering the same initial state  $\mathbf{x}(0)$ , generally lead to different values. In Figure 1 we show the result of 100 independent SSA simulations of the model of Michaelis-Menten (MM) enzyme kinetics [20], which will be used as a test case in this paper. The MM model consists of the following reactions:

- $R_1 : S + E \xrightarrow{c_1=10^{-2}} ES;$
- $R_2 : ES \xrightarrow{c_2=10^{-1}} S + E;$
- $R_3 : ES \xrightarrow{c_3=1} E + P.$

Figure 1 highlights that different SSA runs of the MM model—executed by using the stochastic constants associated with the reactions, and the initial condition given in Table 1—generate simulation outcomes that are quantitatively different from each other (due to stochastic noise) but in agreement from a qualitative point of view, as denoted by the distributions of the final amount of the three species  $S$  (substrate),  $E$  (enzyme), and  $P$  (product).



**Figure 1.** Example of stochastic simulation: 100 independent SSA runs of the MM model—all starting from the same initial state ( $S = 200, E = 100, ES = 0, P = 0$ ) and using the same parameterization—lead to quantitatively different trajectories and a final distributions of the chemical species.

The noise due to stochastic simulations affects the fitness evaluations and thus hampers the identification of the global optimum of the PE problem: any hill climbing or gradient-based approach would be misled by this issue, and generally be unable to converge to the

**Table 1.** Initial state of the MM model.

Molecular species	Amount
$S$ (substrate)	200
$E$ (enzyme)	100
$ES$ (enzyme-substrate complex)	0
$P$ (product)	0

global optimum. This is the reason why Computational Intelligence meta-heuristics should be definitely preferred in this context [19]. In this paper, we exploit the swarm intelligence method named FST-PSO [14] that is briefly described in the next section.

### 2.2. Fuzzy Self-Tuning Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a widespread meta-heuristics for global optimization [15]. PSO is based on a population (i.e., the swarm) of candidate solutions (i.e., the particles) that move inside a bounded  $D$ -dimensional search space. The particles cooperate with the aim of identifying the global best solution to the problem under investigation, and the strategy exploited during the optimization balances the global exploration and the local exploitation capabilities of the particles. In particular, the convergence process is driven by two specific settings called cognitive attractor  $C_{\text{cog}} \in \mathbb{R}^+$  and the social attractor  $C_{\text{soc}} \in \mathbb{R}^+$  of the particles. The performance of PSO is also influenced by the so-called inertia factor  $\omega \in \mathbb{R}^+$ , which is used to prevent a chaotic movement of particles, and by the maximum and minimum velocity ( $\vec{v}_{\text{max}}, \vec{v}_{\text{min}} \in \mathbb{R}^D$ ) allowed along each dimension of the search space, which can drastically affect the quality of the optimal solutions found. It is worth noting that the settings of this meta-heuristic must be accurately adjusted considering the characteristics of the optimization problem, a fine tuning process that usually requires many trials, since these values cannot be determined analytically.

To overcome this limitation, Fuzzy Self-Tuning PSO [14], which is an improved version of PSO that does not require any user settings, has been recently introduced. FST-PSO dynamically adjusts the settings of each particle, independently from the rest of the swarm, thanks to a Fuzzy Rule Based System (FRBS). The FRBS evaluates the performances of the particle and its distance from the position of the current global best particle to (possibly) modify its settings during the optimization process.

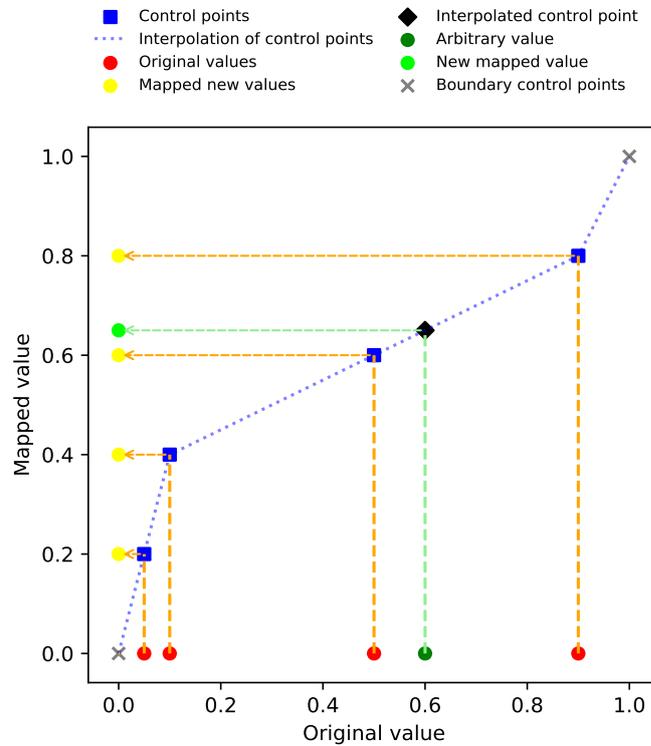
It has been shown in the literature that FST-PSO can outperform the classic PSO (and many competitor algorithms) in several benchmark [14] and real-world problems [21–24].

### 2.3. Dilation Functions

Dilation Functions (DFs) are reversible transformations of a  $D$ -dimensional real space with the aim of “expanding” and “compressing” the regions of the search space in order to ease the optimization process and improve the performances obtained by any optimization method (e.g., PSO). Formally, DFs are defined as mappings in the unitary interval  $\mathcal{W} : [0, 1] \rightarrow [0, 1]$ ; these mappings can be re-scaled to any arbitrary interval by means of linear transformations. Such functions are applied individually to each dimension of the search space, and it is also possible to apply different DFs to different dimensions. In this work we employed DFs generated by two different methods: a) DFs based on control points [6]; b) DFs automatically determined through evolutionary strategies [25].

A control point  $\chi$  is a couple such that  $\chi \in (0, 1) \times (0, 1)$ . A DF is defined by a set of  $Q + 2$  control points  $\chi = \{(0, 0), \chi_0, \dots, \chi_{Q-1}, (1, 1)\}$ , where the control points  $\chi_i$  (blue square dots in Figure 2),  $i = 0, \dots, Q - 1$ , are in ascending order (i.e.,  $\chi_{i-1} \leq \chi_i \leq \chi_{i+1}$ ) and the points  $(0, 0)$ ,  $(1, 1)$  are called boundary control points (which are represented by gray crosses in Figure 2). The control points map their values into the respective new values according to their definition (such values are represented in Figure 2 by red and yellow points respectively). In order to map any given value of the unitary interval, a linear interpolation between the control points  $\in \chi$  is required. In such way, the original

value is applied to the linear function that interpolates the control points and the point is mapped into the “dilated” value (green points in Figure 2). In Figure 3, an example of



**Figure 2.** Mapping of parameters with control points.

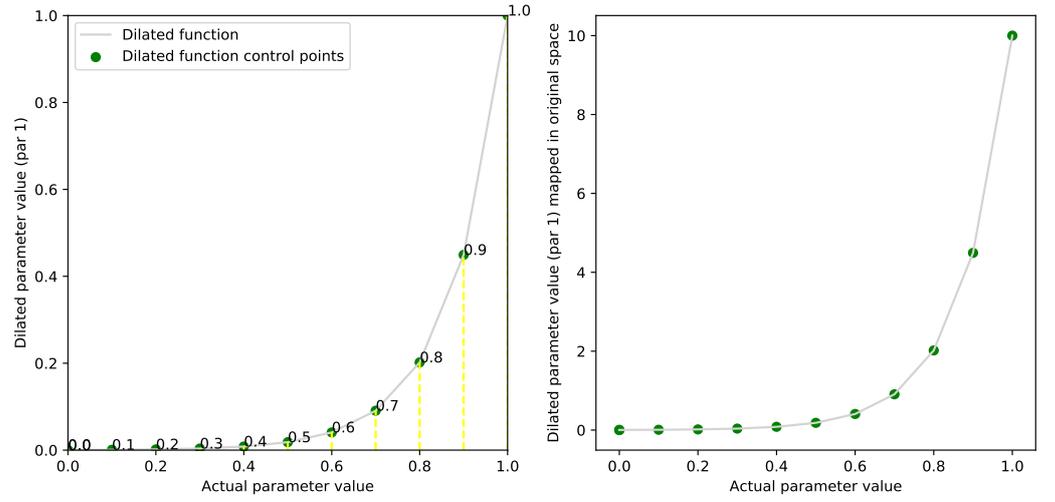
DF based on control points is provided; in particular, an anti-log function is represented. The selection of this DF is determined by the log-uniform distribution followed by the stochastic parameters in biochemical models (*i.e.*, they present a uniform distribution when a logarithmic scale is considered). Due to this consideration, in [26], we proved that it is possible to improve the performances of PE using Particle Swarm Optimization (PSO) by means of a particle initialization according to a log-uniform distribution. The anti-log DF can be used to initialize the particles in regions of the search space characterized by fitness values of lower orders of magnitude. In general, the update rules of the positions of the particles performed in a linear fashion prevent the exploration of such regions. The use of the anti-log DF during the optimization process – and for the initialization of the particles – can overcome this problem by allowing the exploration of the search space in a logarithmic fashion by dilating the regions in the lowest orders of magnitudes. The DF is defined by the control points determined according to Equation 3:

$$\chi_i = (i/Q, \exp(-8(1 - i/Q))), \quad (3)$$

for  $i = 1, \dots, Q$ , with  $Q = 9$ .

#### 2.4. Evolving Dilation Functions

Recently, Papetti *et al.* [25] showed that *ad-hoc* DFs for a fitness landscape can be evolved autonomously by means of an evolutionary algorithm. The DFs are defined by a composition of Basis Functions (BFs), which are bijective and monotonic increasing functions that map the unitary interval into the unitary interval. In this work we used two classes of BFs: i) a family of linear transformations (reported in Equation 4); ii) the *folding*



**Figure 3.** Effect of the anti-log DF. Left: control points encoding the DF. Right: effect of the mapping of a particle's component ( $x$  axis) in the original search space of the PE problem ( $y$  axis).

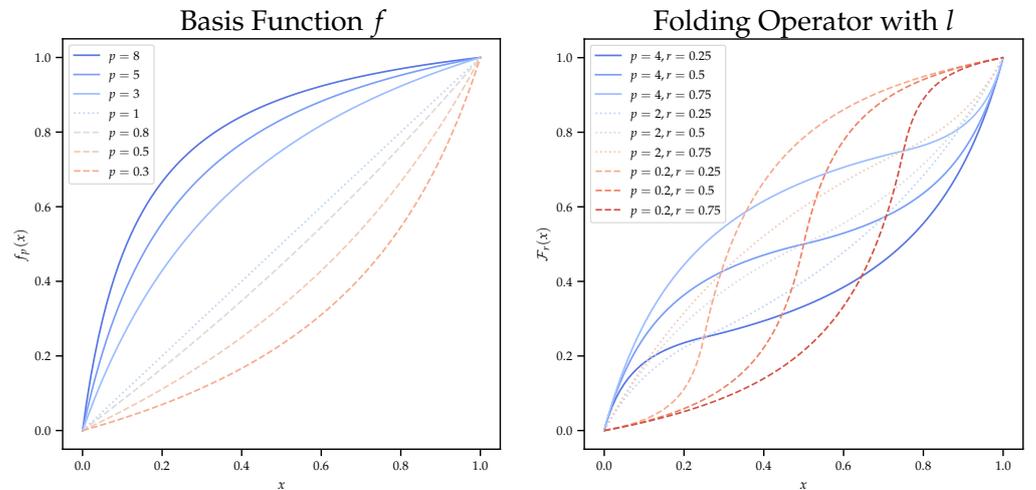
operators (Equation 5). The linear transformation class is characterized by the following function:

$$l_p(x) = \frac{px}{(p-1)x+1}, \quad (4)$$

The parameter  $p$  ( $p \in [0, \infty)$ ) in the first class of BFs determines the magnitude of the distortion applied to the search space with values further from 1 (see Figure 4 on the left). The *folding operator* is based on a BF  $q_p$ , its inverse  $q_p^{-1}$ , and a point  $r \in [0, 1]$  which is the center where the points are moved away or toward (examples of *folding operators* with different values of  $r$  and  $p$  are showed in Figure 4 on the right).

$$\mathcal{F}_r(q_p) = \begin{cases} rq_p\left(\frac{x}{r}\right) & x \leq r \\ (1-r)q_p^{-1}\left(\frac{x}{1-r} - \frac{r}{1-r}\right) + r & x > r \end{cases} \quad (5)$$

In this work,  $q_p = l_p$  and  $r \in 1/4, 1/2, 3/4$ . We also introduced the identity (*i.e.*,  $l_p(x) = x$ ) function as a BF to let the evolutionary process to use fewer transformations than the maximum number allowed, and reducing the complexity of the DF. The five BFs used in



**Figure 4.** Left: different examples of the linear transformation  $l_p$  with different values of  $p$ . Right: examples of *folding operators* with varying values of  $p$  and  $r$ .

this work to compose the DFs are reported in Table 2.

**Table 2.** Basis functions used to define DFs.

ID	Name	Semantics
0	$I_p$	Identity
1	$l_p$	Linear transformation
2	$\mathcal{F}_{\frac{1}{4}}(l_p)$	Folding operators
3	$\mathcal{F}_{\frac{1}{2}}(l_p)$	
4	$\mathcal{F}_{\frac{3}{4}}(l_p)$	

To effectively evolve a DF composed of an arbitrary number of BFs, a two-layered algorithm was developed by Papetti *et al.* in [25]. This approach leverages the computation of a fitness function  $\tilde{f}$  that is based on the computation of the original fitness function  $f$  – or on one of its surrogates  $\tilde{f}, \hat{f}$  – in various dilated points. In particular, an operative definition of the fitness function  $\tilde{f}$  is the following:  $I$  points are randomly sampled from the dilated search space according to a uniform distribution; for each dilated point, the respective fitness value  $f$  is extracted;  $\tilde{f}$  is the average of the fitness values of the 20% best points w.r.t. the respective fitness values. The outer layer is a  $(\mu + \lambda)$ -Evolutionary Strategy [27] with the aim to find the optimal individual representing the structure of the composition of the BFs; *i.e.*, which BFs and in which order are composed. The evolved individuals are strings of length  $\xi$  – with  $\xi > 1$  – belonging to the alphabet  $\{0, 1, 2, 3, 4\}^\xi$ , where the integers are unique identifiers of the BFs considered for this work (see Table 2). The parameters of the BFs of each candidate DF are optimized by the inner layer by means of FST-PSO (please refer to the Sub-section 2.2 for the details) based on the computation of the fitness  $\tilde{f}$ . As soon as FST-PSO terminates, the best fitness value found  $\tilde{g}$ , together with the parameters listed in the particle, are provided to the outer layer and  $\hat{g}$  is the actual fitness value of the candidate DF that will drive the evolution process toward optimal DFs. When the overall evolutionary process ends, the best DF  $\mathcal{W}^*$  – that is the individual with the best value of  $\tilde{f}$  – is returned. The number of sampled points  $I$  depends on the dimensions of the search space  $D$ , and it can automatically inferred through a heuristic  $I = \lceil 5 + 10\sqrt{D} \rceil$ .

### 2.5. Surrogate Fourier Modeling with *surF*

Periodic functions, as well as functions defined only on a limited domain, can be represented via a weighted sum of frequencies by computing their Fourier transform. Representing a function via frequencies allows for performing some interesting manipulations. For example, removing higher frequencies has a smoothing effect on the signal. This effect is being employed by the recently introduced *Fitness Landscape Surrogate Modeling with Fourier Filtering (surF)* technique [12]. There, a surrogate model of the fitness landscape is produced by “smoothing out” the original landscape with the intention of removing local optima where the optimization might stop.

Let  $D$  be the number of dimensions of the search space and let the search space itself be the hypercube  $[\ell, u]^D \subset \mathbb{R}^D$ . Let  $f: [\ell, u]^D \rightarrow \mathbb{R}$  be the objective function, *i.e.*,  $f$  represents the fitness landscape. Since  $f$  is usually not known in an analytical form, we work with *samples* of  $f$  and, consequently, we will employ a discrete version of the Fourier transform, the *Discrete Cosine Transform (DCT)* [28]. It is worth noting that the time and space complexity of *surF* mainly depends on the choice of the method used to compute the Fourier transform; thus, different approaches would give different space and time complexity bounds.

**Discrete Cosine Transform.** To recall the DCT, let us start with the simpler case of a single dimension, *i.e.*,  $D = 1$ . Hence, we have a function  $f: [\ell, u] \rightarrow \mathbb{R}$  and  $\rho \in \mathbb{N}$  equally-spaced points  $z_0, \dots, z_{\rho-1}$  in the interval  $[\ell, u]$  where  $f$  is sample. That is, we know

$f(z_0), \dots, f(z_{\rho-1})$ . The DFT is used to represent each of these points as a weighted sum of frequencies:

$$f(z_k) = \sum_{j=0}^{\rho-1} \psi_j e^{2\pi i \frac{jk}{\rho}} \quad \text{for } 0 \leq k < \rho.$$

Notice how the entire set of  $\rho$  points is entirely determined by the coefficients  $\psi_0, \dots, \psi_{\rho-1}$  representing the amplitudes of the different frequencies. That is, given  $\psi_0, \dots, \psi_{\rho-1}$  it is possible to recover the values  $f(z_0), \dots, f(z_{\rho-1})$  via the *inverse DFT*.

Working with frequencies, however, has some advantages: since each  $\psi_j$  is associated with a different frequency, we can manipulate it. One of the simplest modifications can be performed by zeroing out all but the first  $\gamma \in \{0, \dots, \rho - 1\}$  coefficients. The effect is that all the  $\rho - \gamma$  higher frequencies are removed. When computing the inverse DFT on the coefficients  $\psi_0, \dots, \psi_{\gamma-1}, 0, \dots, 0$  we obtain  $\rho$  points  $y_0, \dots, y_{\rho-1}$  as a “smoothed” version of  $f(z_0), \dots, f(z_{\rho-1})$ . While we have only obtained  $\rho$  points, we can define a surrogate of  $f$  as a function  $\tilde{f}: [\ell, u] \rightarrow \mathbb{R}$  performing a linear interpolation among the points  $y_0, \dots, y_{\rho-1}$ . Since  $\tilde{f}$  is a smoothed version of  $f$ , it might have a reduced amount of local optima, thus helping the optimization process.

The procedure can be extended in any number  $D$  of dimensions. Instead of having  $\rho$  points sampled from  $f$  there will be  $\rho^D$  points sampled from  $f$  in an equally-spaced grid in  $[\ell, u]^D$ . Consequently,  $\rho^D$  coefficients are obtained via DFT and of those, only  $\gamma^D$  are preserved (i.e., all  $\psi_{i_1, \dots, i_D}$  with  $i_j < \gamma$  for all  $1 \leq j \leq D$ ). Hence, the fitness landscape reconstructed by surF has a tunable level of non-zero high-frequency components, which can be controlled by means of the  $\gamma$  hyper-parameter.

**Reducing the number of samples.** The major drawback of the previous approach is that sampling  $\rho^D$  points from  $f$  can be extremely expensive, since the number of points grows exponentially with the number of dimensions. Hence, a way to reduce the number of sampling operations of  $f$  is employed by surF. The main idea is to use  $\sigma \ll \rho^D$  points of  $f$  to generate a first surrogate of  $\hat{f}$  which is then used for the sampling and the remaining of the surF procedure.

To construct  $\hat{f}$ ,  $\sigma$  points  $\vec{z}_0, \dots, \vec{z}_{\sigma-1}$  are randomly selected in  $[\ell, u]^D$ . The selection procedure can be, for example, a uniform selection in the search space. To evaluate  $\hat{f}$  on a point  $\vec{z} \in [\ell, u]^D$  the following procedure is employed:

- If  $\vec{z}$  is in the convex hull defined by the points  $\vec{z}_0, \dots, \vec{z}_{\sigma-1}$ , then  $\hat{f}(\vec{z})$  is obtained by a linear interpolation;
- otherwise, a linear interpolation is not possible and  $\hat{f}(\vec{z})$  is defined as  $\hat{f}(\vec{z}')$  where  $\vec{z}'$  is the point among  $\vec{z}_0, \dots, \vec{z}_{\sigma-1}$  nearest to  $\vec{z}$ .

Once  $\hat{f}$  is constructed, we can sample  $\rho^D$  points from it and proceed with the previously define construction of a smooth surrogate (of  $\hat{f}$  in this case).

**Parameters of surF.** The surF algorithm requires the following settings:

- $\sigma$ , which is the number of samples from  $f$  used to build  $\hat{f}$ ;
- $\rho$ , which is the “density” of samples from  $\hat{f}$  to obtain the  $\rho^D$  points used to calculate the DFT;
- $\gamma$ , which controls the number of low frequencies preserved.

Concerning the first settings, in this work we assume  $\sigma = 100$ , generated using quasi random sampling using Sobol sequences [29], one of the five methods natively supported by surF along with pseudo-random generators, chaotic sequences, quantum random numbers and entropic point packing [13]. Quasi-random sampling is surF’s default, because it was shown to be the most effective approach when a few samples can be collected [13].

The  $\rho$  value should be selected carefully in order to not exceed the resources of the machine. In this work, we will investigate the PE of a model with 3 missing parameters using  $\rho = 100$ , leading to the calculation of  $1 \times 10^6$  interpolated values.

Finally, the  $\gamma$  parameter is particularly sensitive: as a matter of fact, it has a relevant impact on the optimization, because a  $\gamma$  value too small can make the reconstruction of the original global minimum impossible. Since the shape and characteristics of the fitness landscape are unknown, the optimal  $\gamma$  cannot be determined using analytical approaches. One possible solution for this problem is trial and error: repeat the optimization with an increasing number of  $\gamma$  values, observing which setting leads to the best solution. Alternatively, we can co-evolve the optimal  $\gamma$  by extending the candidate solutions' representation, as described in the next section.

### 3. Results

In this section, we will investigate the impact on the PE performances of both DFs and surF. In all tests that follow, we will estimate the stochastic parameters of the MM model by using the settings-free algorithm FST-PSO, running for 100 iterations. The  $\gamma$  parameter of surF is co-evolved with the rest of the parameters; (the range of possible values was set to  $[2, 10]$ ). As baseline, we will exploit the variant of FST-PSO with fuzzy rules for minimum velocity disengaged, denoted by FST-PSO<sub>no v<sub>min</sub></sub>, which was shown to be the most effective choice for this specific problem [22]. In the case of problems modified using a DF or a combination of DF and surF, we will exploit the standard version of FST-PSO. All the analyses were implemented in the Python programming language, exploiting the FST-PSO [14] and surF [13] libraries.

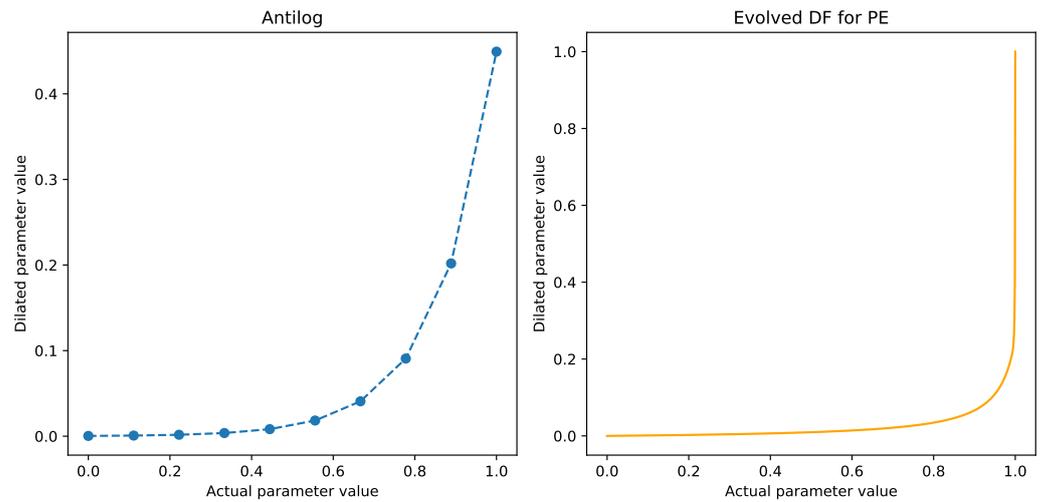
#### 3.1. Effect of DFs on the PE Problem

As a first group of tests, we compared the performance of the PE using the analytical anti-log DF and an evolved DF. In the latter case, we automatically evolved DFs composed of up to 5 BFs, by means of the two-layered algorithm presented in Section 2.3. The population of the evolutionary algorithm was formed by 10 individuals, and we evolved the population for 11 generations. Each process to determine the best parameterization of the BFs was performed with 10 particles for 10 iterations. The number of sampled points  $I$  to compute  $\hat{f}$  was determined with the heuristic proposed in Section 2.3, which corresponds to  $I = 23$  in the following experiments. The overall budget to evolve the DF with such settings is 253 000 fitness evaluations.

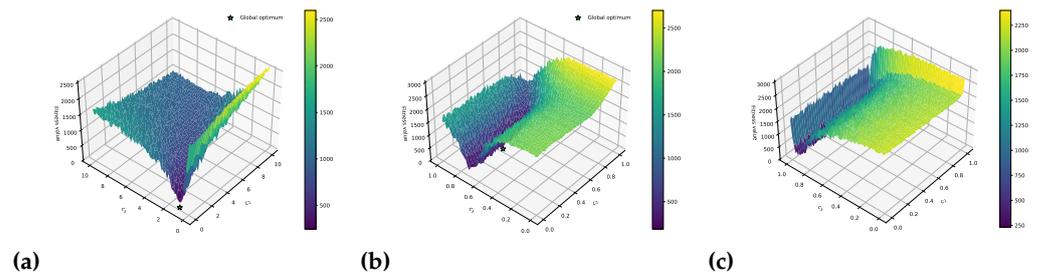
The actual DFs used here—i.e., the anti-log and the evolved one—are shown in Figure 5. It is worth noting that the optimal DF evolved by our approach for the PE of the MM model, despite being “stronger” than the analytical DF, applies a similar dilation to the fitness landscape, i.e., the region corresponding to the lowest orders of magnitude is expanded. To be more specific, the optimal DF obtained for the PE is a composition of 3 BFs (2 out of 5 BFs were identity functions that can be ignored), as follows:  $l_{0.09}(\mathcal{F}_{\frac{1}{2}}(l_{4.65}(l_{0.02}(x_d))))$ , for all  $d = 1, \dots, D$ .

Figure 6 shows the fitness landscapes for the PE of the MM model (restricted to the parameters  $c_2$  and  $c_3$  in order to represent the 3D landscape) corresponding to: the original problem (left); the landscape dilated with the analytical anti-log function (center); the landscape dilated using the evolved DF (right). The effect of these DFs is to shift away the global optimum from the lowest orders of magnitude. The shift exerted by the evolved function seems to be stronger, revealing a larger region characterized by sub-optimal fitness values (yellow points), even though the intensity also depends on the magnitude of the original parameters.

We compared the performance of FST-PSO, executed considering the three aforementioned fitness landscapes, to estimate the parameters of the MM model. In each case, we performed 30 independent runs using the default functioning settings that correspond to 14 individuals and 100 iterations. In the first case (i.e., the original fitness landscape), the



**Figure 5.** Graphical representation of the DF used in the PE problem. Left: anti-log function analytically designed to expand the fitness landscape in the lowest orders of magnitude. Right: evolved DF that substantially perform the same dilation of the anti-log DF.



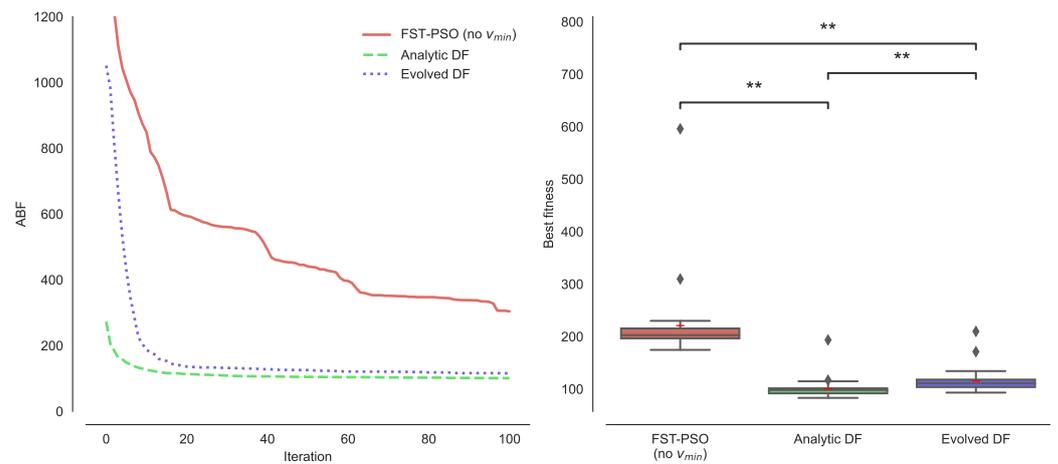
**Figure 6.** Effect of DFs on the fitness landscape of the MM model. (a) Original fitness landscape. (b) Fitness landscape diluted by means of the analytical anti-log DF. (c) Fitness landscape diluted by means of the optimal evolved DF.

fuzzy reasoning that governs the  $v_{min}$  of FST-PSO is disabled, because it is known to affect the performance in the original formulation of the PE problem [22].

Figure 7 shows the convergence plot (left) and the distributions of the best fitness values found over 30 runs of each methodology (right). These results show that the analytic DF (green dashed line) is the best option for the PE problem: the Average Best Fitness (ABF) is indeed much lower with respect to the original landscape (coral solid line) and slightly lower than in the case of the evolved DF (blue dotted line). Although the best individual in the first iteration of the optimization with the evolved DF is, on average, worse than in the case of the analytical anti-log DF, all runs based on the evolved DF are characterized by a very fast convergence: as a matter of fact, after 20 iterations the average quality of the identified solutions is comparable (Figure 7, left), as also confirmed by the boxplots (Figure 7, right). The distributions reported by means of the boxplot were compared by leveraging the Mann-Whitney U rank test. The  $p$ -values confirmed that the differences are statistically significant. It is worth noting that the original landscape induced by the PE problem is difficult to explore, even with a low number of dimensions (3, in these tests), as evidenced by the outliers corresponding to FST-PSO runs ended with very high fitness values.

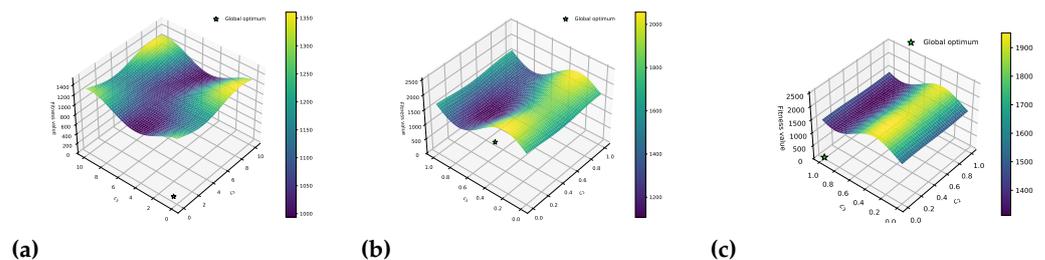
### 3.2. Combining DFs and Fourier surrogate modeling

As a second group of tests, we applied the new surF algorithm to all the previous fitness landscapes, in order to optimize a filtered and smoothed version of the optimization problem. Figure 8 shows the effect of surF using  $\gamma = 2$ : in the case of the original



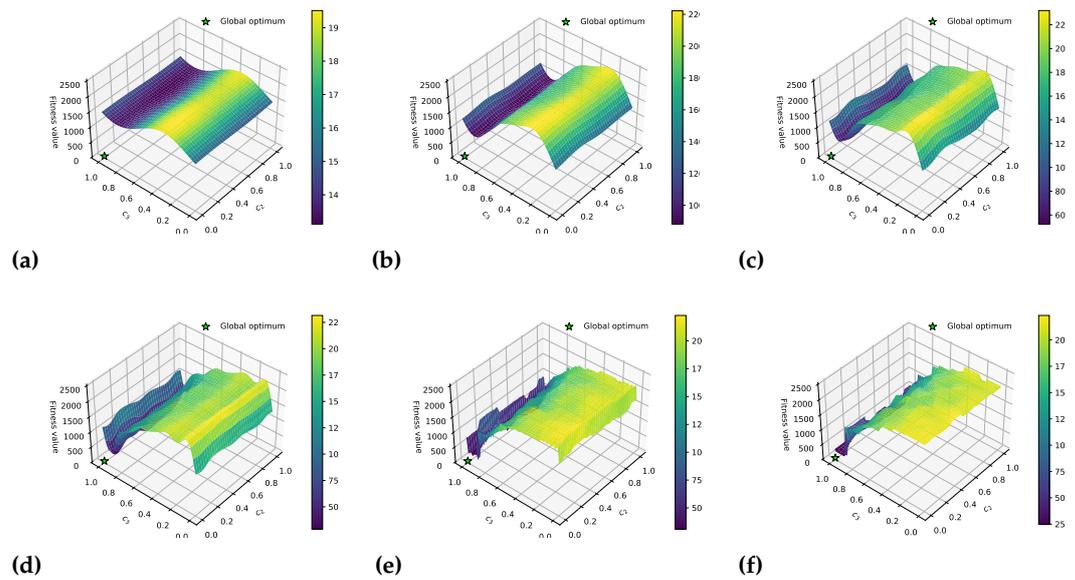
**Figure 7.** Left: convergence plot of the PE of the MM model obtained with FST-PSO (no  $v_{min}$ ) (coral solid line), FST-PSO using the analytic DF (green dashed line), FST-PSO using the evolved DF (blue dotted line). The lines correspond to the ABE calculated over 30 runs. Right: boxplots representing the distributions of the best solutions found at the end of each run, for each methodology; red dashes denote the mean of the distributions, while diamonds denote the outliers. The asterisks denote the  $p$ -values obtained by comparing the distributions by means of the Mann-Whitney U tests (\*  $p$ -value  $\leq 0.5$ , \*\*  $p$ -value  $\leq 0.0001$ ).

landscape, the low number of Fourier coefficients prevents the reconstruction of the high frequency optimum, a circumstance already discussed in [5,12]; in the case of the landscape dilated with the anti-log DF, the smoothed landscape is denoted by a global optimum corresponding to the actual target parameterization; finally, in the case of surF applied to the fitness landscape dilated using the evolved DF, the problem of the missing high frequency components again prevents a correct reconstruction of the landscape, as in the case of the original problem.



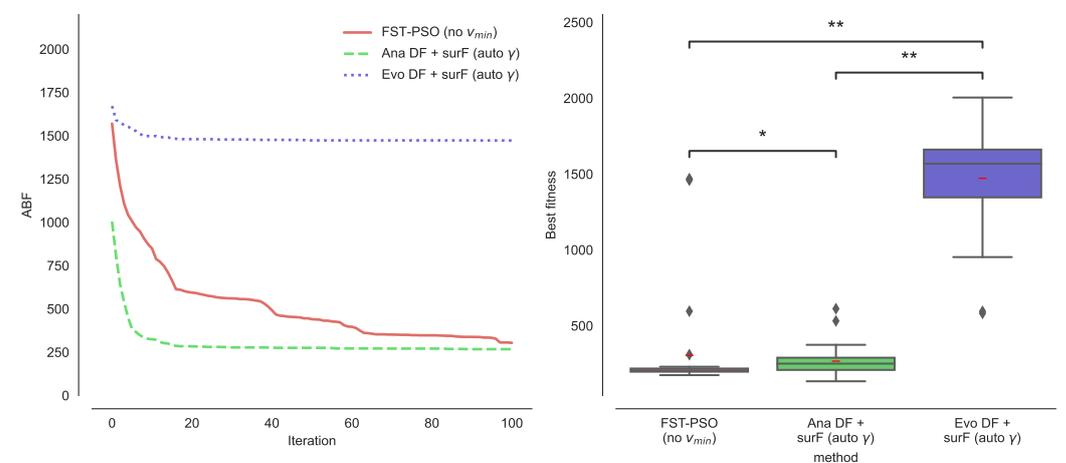
**Figure 8.** Effect of the surrogate modeling on the fitness landscape of the MM model. (a) Surrogate model of the original fitness landscape. (b) Surrogate model of the fitness landscape dilated by means of the anti-log DF. (c) Surrogate model of the fitness landscape dilated by means of the evolved DF.

To further investigate the effect of the parameter  $\gamma$  on the fitness landscape, we created different surrogate models starting from the fitness landscape obtained with the evolved DF (Figures 9). The plots highlight that even in the case of  $\gamma = 100$ , the global optimum cannot be properly reconstructed, as the excessively strong dilation caused by the evolved DF is detrimental to the optimization results when coupled to Fourier surrogate modeling. Figure 10 confirms this insight by showing that FST-PSO cannot converge in the case of the surrogate model created with surF, starting from the fitness landscape dilated with the evolved DF (blue dotted line); on the contrary, FST-PSO using the surrogate model created with surF starting from the fitness landscape dilated with the analytical anti-log DF, is capable of reaching the global optimum. In any case, it is worth mentioning that thanks to the strategy of co-evolving the  $\gamma$  value with the rest of the candidate solution, the



**Figure 9.** Surrogate models of the fitness landscape for the PE of the MM model dilated using the evolved DF, with  $\gamma = 2$  (a),  $\gamma = 3$  (b),  $\gamma = 5$  (c),  $\gamma = 10$  (d),  $\gamma = 50$  (e),  $\gamma = 100$  (f).

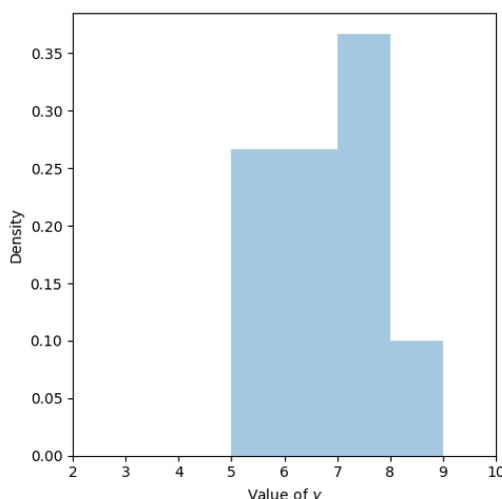
methodology no longer requires domain knowledge to achieve accurate solutions for the PE problem of biochemical systems.



**Figure 10.** Left: convergence plot of the PE of the MM model obtained with FST-PSO (no  $v_{min}$ ) (coral solid line), FST-PSO using the analytic DF + surF (green dashed line), FST-PSO using the evolved DF + surF (blue dotted line). The lines correspond to the ABF calculated over 30 runs. Right: boxplots representing the distribution of the best solutions found at the end of each run; red dashes denote the mean of the distributions, while diamonds denote outliers. The asterisks denote the  $p$ -values obtained by comparing the distributions by means of the Mann-Whitney U tests (\*  $p$ -value  $\leq 0.5$ , \*\*  $p$ -value  $\leq 0.0001$ ).

#### 4. Discussion

In this work, we investigated the impact of two methods for the fitness landscape manipulation, namely surF [12] and DFs [6], on the PE problem of stochastic biochemical models. Following the results obtained in [5], we addressed a common limitation of these two methods, that is, the need for expert knowledge regarding the characteristics of the fitness landscape. This was achieved by modifying the surF algorithm in order to co-evolve



**Figure 11.** Distribution of the optimal  $\gamma$  values identified by our approach.

the optimal value of the  $\gamma$  hyper-parameter, and by performing a separate optimization process to evolve an optimal DF for the problem at hand [25].

Our results show that the PE employing the evolved DF has a performance that is comparable to the one employing the analytical DF, further corroborating their application for the PE of biochemical models and in contexts where there is no knowledge about the fitness landscape [25]. The co-evolution of the  $\gamma$  hyper-parameter in surF also proved to be an effective strategy, by improving the convergence speed and the final results with respect to a PE not employing surF surrogates, notably with one less hyper-parameter to be manually set by the user when compared to the original surF algorithm [12]. Figure 11 shows the distribution of the optimal  $\gamma$  values automatically identified by our approach. As can be observed from the plot, the mode of the distribution is  $\gamma = 7$  and, overall, the optimization of this parameter led always to values between 5 and 8. This confirms the adequacy of the search space for the parameter  $\gamma$  (i.e., [2, 10]) employed in our tests.

However, these improvements were observed only on fitness landscapes that were dilated with the analytical DF exploiting expert knowledge. In our tests, the application of surF prevented the convergence to optimal solutions on landscapes dilated with the evolved DF, suggesting that not all DFs are beneficial to the construction of surrogates by means of surF, even when a high value of  $\gamma$  is employed. As a future extension of this work, we plan to further investigate this issue by evolving DFs that take into account the transformations of the landscape applied by surF. Moreover, we plan to improve the evolution of DFs by identifying different dilations for each dimension of the considered PE problem.

Another limitation that affects surF regards the algorithm used to generate the surrogate models, which is characterized by a high time and space complexity. As already discussed in [13], surF relies on DFT which can calculate the multi-dimensional spectra of the fitness landscape from equispaced samples. That is, surF interpolates the samples over a multi-dimensional regular lattice, an operation characterized by exponential complexity. In practical terms, surF has reasonable time and space complexity up to 5 or 6 variables, but the computation time required makes it unsuitable for more than 4 variables. Thus, we plan to investigate different approaches to compute the Fourier transform using, for example, sparse samples. This will greatly improve the performances of surF, in terms of computation time. We will also explore alternative strategies, not involving the computation of a Fourier transform, in order to generate surrogate models with a less-than-exponential complexity, thus allowing the synergistic application of DFs and surF to high dimensional optimization problems, such as the PE of large RBMs with several missing parameters [30].

**Author Contributions:** Conceptualization, M.S.N., D.M.P. and L.M.; methodology, M.S.N., D.M.P. and L.M.; software, M.S.N., D.M.P. and L.M.; formal analysis, M.S.N., D.M.P., S.S., P.C. and L.M.; investigation, M.S.N. and D.M.P.; data curation, M.S.N.; writing—original draft preparation, M.S.N., D.M.P., S.S., P.C. and L.M.; writing—review and editing, M.S.N., D.M.P., S.S., P.C. and L.M.; visualization, M.S.N. and D.M.P.; supervision, M.S.N. and L.M.; project administration, L.M.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Not applicable.

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

## Abbreviations

The following abbreviations are used in this manuscript:

ABF	Average Best Fitness	
BF	Basis Function	
DCT	Discrete Cosine Transform	
DF	Dilation Function	
E	Enzyme	
ES	Enzyme-Substrate complex	
FRBS	Fuzzy Rule Based System	
FST-PSO	Fuzzy Self-Tuning Particle Swarm Optimization	
MM	Michaelis-Menten	408
P	Product	
PE	Parameter Estimation	
PSO	Particle Swarm Optimization	
RBM	Reaction-Based Model	
S	Substrate	
SSA	Stochastic Simulation Algorithm	
surF	Fitness Landscape Surrogate Modeling with Fourier Filtering	

**Mathematical notation**

The following mathematical notation is used in this manuscript:

$\alpha_{m,n}$	stoichiometric coefficients associated to the $n$ -th reactant	409
$\beta_{m,n}$	stoichiometric coefficients associated to the $m$ -th reaction	410
$\mathbf{c}$	vector of stochastic constants	411
$c_m$	stochastic (kinetic) constant	
$C_{\text{cog}}$	cognitive attractor of FST-PSO	
$C_{\text{soc}}$	social attractor of FST-PSO	
$D$	number of dimensions of the search space	
$d_m$	number of distinct combinations of the reactant molecules	
$f$	original fitness function	
$\tilde{f}$	dilated fitness function	
$\bar{f}, \hat{f}$	surrogate fitness functions	
$\mathcal{F}_r(q_p)$	folding operator	
$\gamma$	number of the lower frequencies to not be zeroed	
$I$	number of sampled points to compute the dilated landscape	
$[\ell, u]^D$	lower and upper bounds of the search space	
$l_p$	linear Basis Function	
$O_n(t_k)$	the experimental (target) amount of $S_n$ measured at time $t_k$	
$p$	parameter of the linear Basis Function $l_p$	
$\psi_\rho$	coefficient representing the amplitude of $\rho$ -th frequency	
$r$	parameter of the folding operator	
$\mathcal{R}$	set of biochemical reactions	
$R_m$	$m$ -th biochemical reaction	
$\rho$	number of equally spaced points to build the surrogate function	412
$S$	set of molecular species	
$S_i$	$i$ -th molecular specie	
$\sigma$	number of samples used to construct the surrogate	
$t$	time of the system	
$\mathbf{t}$	vector of time points	
$t_k$	$k$ -th time point	
$\tau$	waiting time	
$\vec{v}_{\text{max}}$	maximum velocity of the FST-PSO particles	
$\vec{v}_{\text{min}}$	minimum velocity of the FST-PSO particles	
$\omega$	inertia factor of FST-PSO	
$X_n^c(t_k)$	simulated amount of the species $S_n$ at time $t_k$	
$\mathbf{x}(t)$	vector representing the state of the system at time $t$	
$x_n$	amount of the $n$ -th molecular specie	
$\chi$	control point	
$\chi_Q$	$Q$ -th control point	
$\xi$	length of the individuals representing the DFs	
$\chi$	vector of control points	
$y_\rho$	fitness value of the $\rho$ -th point of the surrogate	
$z_\rho$	$\rho$ -th point of the search space to build the surrogate function	
$\zeta_1$	random number sampled from an uniform distribution	
$\zeta_2$	random number sampled from an uniform distribution	

**References**

1. Munsky, B.; Tuzman, K.T.; Fey, D.; Dobrzynski, M.; Kholodenko, B.N.; Olson, S.; Huang, J.; Fox, Z.; Singh, A.; Grima, R.; et al. *Quantitative Biology: Theory, Computational Methods, and Models*; The MIT Press, 2018. 413-415
2. Nobile, M.S.; Besozzi, D.; Cazzaniga, P.; Mauri, G.; Pescini, D. A GPU-based multi-swarm PSO method for parameter estimation in stochastic biological systems exploiting discrete-time target series. In *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*; Giacobini, M.; Vanneschi, L.; Bush, W., Eds.; Springer, 2012; Vol. 7246, LNCS, pp. 74–85. 416-418

3. Nobile, M.S.; Besozzi, D.; Cazzaniga, P.; Mauri, G.; Pescini, D. Estimating reaction constants in stochastic biological systems with a multi-swarm PSO running on GPUs. In Proceedings of the Proceedings of the 14th Annual Conference companion on Genetic and Evolutionary Computation (ACM 2012). ACM, 2012, pp. 1421–1422. 419–421
4. Daigle, B.J.; Roh, M.K.; Petzold, L.R.; Niemi, J. Accelerated maximum likelihood parameter estimation for stochastic biochemical systems. *BMC Bioinformatics* **2012**, *13*, 68. 422–423
5. Nobile, M.S.; Cazzaniga, P.; Spolaor, S.; Besozzi, D.; Manzoni, L. Fourier Surrogate Models of Dilated Fitness Landscapes in Systems Biology: or how we learned to torture optimization problems until they confess. In Proceedings of the Proceedings of the 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB). IEEE, 2020, pp. 1–8. 424–426
6. Nobile, M.S.; Cazzaniga, P.; Ashlock, D.A. Dilation Functions in Global Optimization. In Proceedings of the Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC 2019). IEEE, 2019, pp. 2300–2307. 427–428
7. Chunming, F.; Yadong, X.; Jiang, C.; Xu, H.; Huang, Z. Improved differential evolution with shrinking space technique for constrained optimization. *Chinese Journal of Mechanical Engineering* **2017**, *30*, 553–565. 429–430
8. Wang, Y.; Cai, Z.; Zhou, Y. Accelerating adaptive trade-off model using shrinking space technique for constrained evolutionary optimization. *International Journal for Numerical Methods in Engineering* **2009**, *77*, 1501–1534. 431–432
9. Aguirre, A.H.; Rionda, S.B.; Coello Coello, C.A.; Lizárraga, G.L.; Montes, E.M. Handling constraints using multiobjective optimization concepts. *International Journal for Numerical Methods in Engineering* **2004**, *59*, 1989–2017. 433–434
10. Wang, H.; Wu, Z.; Liu, Y.; Wang, J.; Jiang, D.; Chen, L. Space transformation search: a new evolutionary technique. In Proceedings of the Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation (ACM 2009). ACM, 2009, pp. 537–544. 435–437
11. Bhosekar, A.; Ierapetritou, M. Advances in surrogate based modeling, feasibility analysis, and optimization: A review. *Computers & Chemical Engineering* **2018**, *108*, 250–267. 438–439
12. Manzoni, L.; Papetti, D.M.; Cazzaniga, P.; Spolaor, S.; Mauri, G.; Besozzi, D.; Nobile, M.S. Surfing on fitness landscapes: a boost on optimization by Fourier surrogate modeling. *Entropy* **2020**, *22*, 285. 440–441
13. Nobile, M.S.; Spolaor, S.; Cazzaniga, P.; Papetti, D.M.; Besozzi, D.; Ashlock, D.A.; Manzoni, L. Which random is the best random? A study on sampling methods in Fourier surrogate modeling. In Proceedings of the Proceeding of the 2020 IEEE World Congress on Evolutionary Computation (CEC 2020). IEEE, 2020. 442–444
14. Nobile, M.S.; Cazzaniga, P.; Besozzi, D.; Colombo, R.; Mauri, G.; Pasi, G. Fuzzy Self-Tuning PSO: A settings-free algorithm for global optimization. *Swarm and Evolutionary Computation* **2018**, *39*, 70–85. 445–446
15. Poli, R.; Kennedy, J.; Blackwell, T. Particle swarm optimization. *Swarm Intelligence* **2007**, *1*, 33–57. 447
16. Gillespie, D.T. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry* **1977**, *81*, 2340–2361. 448
17. Cazzaniga, P.; Damiani, C.; Besozzi, D.; Colombo, R.; Nobile, M.S.; Gaglio, D.; Pescini, D.; Molinari, S.; Mauri, G.; Alberghina, L.; et al. Computational strategies for a system-level understanding of metabolism. *Metabolites* **2014**, *4*, 1034–1087. 449–450
18. Elowitz, M.B.; Levine, A.J.; Siggia, E.D.; Swain, P.S. Stochastic gene expression in a single cell. *Science* **2002**, *297*, 1183–1186. 451
19. Nobile, M.S.; Tangherloni, A.; Rundo, L.; Spolaor, S.; Besozzi, D.; Mauri, G.; Cazzaniga, P. Computational Intelligence for Parameter Estimation of Biochemical Systems. In Proceedings of the Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC 2018). IEEE, 2018, pp. 1–8. 452–454
20. Nelson, D.; Cox, M. *Lehninger Principles of Biochemistry*; W. H. Freeman Company, 2004. 455
21. Empereur-Mot, C.; Pesce, L.; Doni, G.; Bochicchio, D.; Capelli, R.; Perego, C.; Pavan, G.M. Swarm-CG: Automatic Parametrization of Bonded Terms in MARTINI-Based Coarse-Grained Models of Simple to Complex Molecules via Fuzzy Self-Tuning Particle Swarm Optimization. *ACS Omega* **2020**. 456–458
22. Tangherloni, A.; Spolaor, S.; Cazzaniga, P.; Besozzi, D.; Rundo, L.; Mauri, G.; Nobile, M.S. Biochemical parameter estimation vs. benchmark functions: A comparative study of optimization performance and representation design. *Applied Soft Computing* **2019**, *81*, 105494. 459–461
23. SoltaniMoghadam, S.; Tatar, M.; Komeazi, A. An improved 1-D crustal velocity model for the Central Alborz (Iran) using Particle Swarm Optimization algorithm. *Physics of the Earth and Planetary Interiors* **2019**, *292*, 87–99. 462–463
24. Fuchs, C.; Spolaor, S.; Nobile, M.S.; Kaymak, U. A Swarm Intelligence Approach to Avoid Local Optima in Fuzzy C-Means Clustering. In Proceedings of the Proceedings of the 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2019). IEEE, 2019, pp. 1–6. 464–466
25. Papetti, D.M.; Ashlock, D.A.; Cazzaniga, P.; Besozzi, D.; Nobile, M.S. If You Can't Beat It, Squash It: Simplify Global Optimization by Evolving Dilation Functions. In Proceedings of the Proceedings of the 2021 IEEE Congress on Evolutionary Computation (CEC 2021), 2021. 467–469
26. Cazzaniga, P.; Nobile, M.S.; Besozzi, D. The impact of particles initialization in PSO: parameter estimation as a case in point. In Proceedings of the Proceedings of the 2015 IEEE International Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2015). IEEE, 2015, pp. 1–8. 470–472
27. Beyer, H.G.; Schwefel, H.P. Evolution strategies—a comprehensive introduction. *Natural Computing* **2002**, *1*, 3–52. 473
28. Cooley, J.W.; Tukey, J.W. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation* **1965**, *19*, 297–301. 474–475
29. Sobol', I.M. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki* **1967**, *7*, 784–802. 476–477

- 
30. Spolaor, S.; Gribaudo, M.; Iacono, M.; Kadavy, T.; Oplatková, Z.K.; Mauri, G.; Pillana, S.; Senkerik, R.; Stojanovic, N.; Turunen, E.; et al. Towards Human Cell Simulation. In *High-Performance Modelling and Simulation for Big Data Applications*; Springer, 2019; Vol. 11400, *Lecture Notes in Computer Science*, pp. 221–249. 478  
479  
480