https://doi.org/10.1007/s42979-022-01094-0

ORIGINAL RESEARCH

# Evolutionary Symbolic Regression from a Probabilistic Perspective

Chi Gong[1,2] · Jordan Bryan[1] · Alex Furcoiu[1] · Qichang Su[1] · Rainer Grobe[1]

Received: 5 November 2021 / Accepted: 14 March 2022

**Abstract**
We examine the genetic evolution-based algorithm for symbolic regression from a probabilistic dynamical perspective. This approach permits us to follow the evolution of the search candidate functions from generation to generation as they improve their fitness and finally converge to the best function that matches a given data set. In particular, we use this statistical framework to explore the optimal external parameters that govern a special mutation operator, which can systematically improve the numerical value of constants contained in each candidate formula of the search space. We then apply symbolic regression to the chaotic logistic map and the Lorenz system.

**Keywords** Symbolic regression · Constant mutation operator

## Introduction

Symbolic regression (SR) is an iterative stochastic algorithm to search the space of mathematical expressions with the goal to identify the one that best describes a given experimental or simulated data set. It has been playing a small part in the general trend to include modern machine learning techniques and artificial intelligence [1] into scientific applications. To name a few recent examples, SR has been applied in weather data analysis [2, 3], pharmaceutical systems [4], astronomy [5], Newtonian mechanics [6], fundamental physics equations [7–9], solar electricity [10], fluid dynamics [11], discrete field theories [12], and in quantum field theory [13].

Within the applications in physics, SR has demonstrated its power in recovering algebraic and differential equations of known physical laws in the recent past. The great promise of such a new tool would be to offer nontrivial expressions, solutions, or differential equations for those situations that are not easily obtainable by traditional theoretical analysis. This powerful tool would no doubt assist us to extend the range of investigations in many nonlinear and non-perturbative physics problems. The general task of predicting the general dynamics from time series has been one of the goals of reservoir computing, which is a prominent machine learning technique [14, 15]. It has been applied successfully to accurate short-term prediction and attractor reconstruction of chaotic dynamical systems from time series data [16]. While these machine-learning based predictions are highly valuable, they do not provide analytical models, which could make further interpretations and mathematical analysis possible. We will illustrate below for the logistic map and the Lorenz system that SR can also be used to predict the chaotic dynamics from the time iterates associated within the regular domain.

Symbolic regression randomly combines an initial set of provided mathematical operators, functions, variables and constant parameters to uncover the relationships represented by the data [17–20]. It requires a wide choice for its many genetic programing related computational features, such as the number of individua per generation, the type of selection criteria, the evolutionary strategies, the mutation and crossover probabilities and the fitness criteria, to name just

✉ Qichang Su
qcsu@ilstu.edu

Chi Gong
cgong1@ilstu.edu

Jordan Bryan
jnbrya1@ilstu.edu

Alex Furcoiu
aafurco@ilstu.edu

Rainer Grobe
grobe@ilstu.edu

1 Intense Laser Physics Theory Unit, Department of Physics, Illinois State University, Normal, IL 61790-4560, USA

2 State Key Laboratory for GeoMechanics and Deep Underground Engineering, China University of Mining and Technology, Beijing 100083, China

Published online: 04 April 2022

a few. Due to this complexity, the numerical implementation of SR has to be viewed as a black-box algorithm. While in our opinion the SR has the enormous potential to become a valuable tool in the physical sciences, the determination of its optimal numerical working conditions requires often tedious and trial-and-error-based procedures, which depend sensitively on the details of the particular application. It would therefore be quite helpful to establish first steps towards a statistical foundation that could be used to examine the efficiency of these evolutionary stochastic algorithms.

A major challenge in SR is not only to discover the correct functional relationship among the independent variables, but also to determine the optimum numerical values of the parameters (constants) that can best fit the given data set. In its traditional implementation using genetic programming, state variables and parameters are being treated on an identical footing [21, 22]. In contrast to genetic algorithms, where the structure of the individuals is homogeneous and their format is fixed [23], genetic programming-based SR has to encompass the discrete sets of different functional forms as well as the infinite continuous spaces of the constants. If all constants in the target function (to be uncovered) happen to be just integers, SR can identify even complicated multi-variable dependences rather efficiently. However, more generally, once the data set requires parameters with non-integer values, SR has enourmous difficulties as numerical values are assigned randomly to constants (free parameters) for the first generation of the pool of candidate expressions. During the evolution, the code usually cannot directly update these assignments. The only way of obtaining improved numerical values is by using operational combinations of these fixed values, such as "mul(2.0,add(1.1, 1.8))", to represent the new "parameter" with value 5.8. The usual implementation of the evolutionary code does not allow for these types of simplifications to construct new leaves. This means that the complexity of expression trees can grow unnecessarily over time (bloat) and potentially good search candidates are eliminated despite their low errors due to parsimony constraints.

The goal of the present work is two-fold. First, we aim to introduce a statistical theoretical framework for SR that can provide us with better intuition and some first guidance about how to improve the efficiency of the algorithm. Second, using this probabilistic framework, we introduce a new special purpose mutation operator that can supplement the traditional mutation and crossover operators. In contrast to the latter operators that are usually employed in the exploratory phase to increase the diversity of the search space, this particular operator is aimed at accelerating the convergence by solely improving the numerical values of the parameters.

The article is organized as follows. In "Evolution of the fitness probability density based on deterministic tournament selection", the probabilistic concepts of the fitness density and function classes are introduced. "An illustration of the probabilistic theory of the symbolic regression for M = 3 function classes" provides a concrete example of the SR evolution, where each generational step can be followed from a probabilistic perspective. "In the constant-mutation operator (CMO)", the constant-mutation operator (CMO) is introduced. "Impact of the adaptive CMO in symbolic regression with mutations and cross-overs", we illustrate the power of including the adaptive CMO in SR. In "SR-guided reconstruction of iterative maps or differential equations from chaotic time series", an application of SR to the logistic map and the Lorenz system is examined. Finally, in "Summary and outlook into future challenges" we present a summary and outlook into future challenges.

## Evolution of the Fitness Probability Density Based on Deterministic Tournament Selection

As in most genetic programs, the evolution from generation to generation is based on iterated sequences of two consecutive actions. The first one is the selection of the new mating pool from the first generation of search candidates, which is then followed by the application of various types of mutation and crossover operations. There are basically four different (and commonly used) selection schemes, including proportionate, ranking, steady state and tournament selection [24–27].

We will focus here on the tournament selection, where $n_T$ candidates are randomly chosen from the entire population of $N_{\text{pop}}$ individua. There are deterministic as well as probabalistic versions of the tournament rules. In the latter, the ranked individuals are assigned a survival probability that is related to their fitness. In this work, we focus on the deterministic version, where among these $n_T$ individuals, only the single winner (the one with the best fitness) is chosen to join the mating pool. These tournaments are then repeated as often as desired (usually until the new mating pool with $N_{\text{pop}}$ members is filled). Due to the consecutive nature of the selection and evolutionary action, we can consider them separately. In this section, we focus entirely on the selection procedure and therefore neglect any consecutive mutation/crossover operations such that the selected mating pool becomes identical here to the next generation.

In "Parse-tree representation of expressions and function classes" below, we will introduce the underlying statistical framework for the tournament selection, by proposing new concepts of the total fitness probability density $\rho_{\text{tot}}(f, t)$ as well as the density $\rho_m(f, t)$ and the proportion $P_m(t)$ of each function class.

## Parse-Tree Representation of Expressions and Function Classes

The Hilbert space of the candidate functions (primitive set) in the zeroth generation is encoded here by an LISP-style flattened inverted syntax tree representation. We assume the initiation method is provided by the so-called "full" pattern resulting in sets of fully symmetric parse-trees of depth $D$ and corresponding length $L = 2^{D+1} - 1$ (defined by the number of nodes) with the number of terminals given by $2^D$. As the main ideas of this work can be easily generalized to multi-variable target functions, we can focus here on those input data that were generated by a target function $g_{\text{target}}(x)$, which depends only on one independent variable $x$. This means that the terminals (leaves) of each parse-tree represent either this variable $x$ or a constant numerical value of a parameter, which we denote by $c$. Let us further assume that these two choices are picked randomly with equal probability of 50%. The initial random values for all constants $c$ follow a chosen parameter probability density $p_{\text{para}}(c)$ with normalization $\int_{-\infty}^{\infty} \mathrm{d}c\, p_{\text{para}}(c) = 1$.

Unfortunately, there is a huge undesirable redundancy of different tree structures that lead to the same functional mathematical dependence on $x$. For example, while the function space defined by the two arithmetic operations "addition" and "multiplication" (with arity = 2) with depth = 2 (and resulting length $L = 7$) permits 128 different tree structures, it represents only $M = 8$ different types of mathematical functional relationships. We associate those functions that differ only by their particular values of the constants with the same function class, which we label by the subscript $m = 1, 2, \ldots, M$. Therefore, the total number $M$ of different function classes can be directly associated with the diversity of the population in this generation.

## The Fitness Probability Density, Proportions and Parameter Distributions

To have a concrete realization for the fitness for each function class, let us assume the argument $x$ of the target function $g_{\text{target}}(x)$, from which the input data are generated, is in the range $a \leq x \leq b$. For example, we can associate the error between $g_{\text{target}}(x)$ and the search candidate $g_m(x)$ with its fitness $f_m$, i.e., $f_m \equiv \int_a^b \mathrm{d}x\, |g_{\text{target}}(x) - g_m(x)|$. For each individual $g_m(x)$ in a class m, this fitness can be evaluated as a function of the constants $(c_1, c_2, \ldots, c_n) \equiv \mathbf{c}$ as $f_m = f_m(c_1, c_2, \ldots, c_n) \equiv f_m(\mathbf{c})$. As the initial probability density for the constants $\rho_{\text{para}}(\mathbf{c})$ for the zeroth generation is known (=chosen), it can be used to compute the initial *probability distribution of the fitness* $\rho_m(f)$ for each function class, given as $\rho_m(f) \equiv \int \mathrm{d}^n c\, \delta[f - f_m(\mathbf{c})] \rho_{\text{para}}(c_1) \ldots \rho_{\text{para}}(c_n)$. As the distribution of each of the constants changes during the evolution, so does the characteristic fitness density $\rho_m(f)$. For example, it follows that $\int_{f1}^{f2} \mathrm{d}f' \rho_m(f')$ is therefore the total probability that a representative function $g_m(x)$ has a fitness value inside the interval $(f_1, f_2)$ [28, 29]. The concept of the fitness density $\rho_m(f)$ should not be confused with the well-known fitness landscape, which is used in evolutionary optimization [28–31], which in our context resembles more $f_m(\mathbf{c})$ for each class.

In each generation (indexed by the generational integer time $t = 0, 1, 2, \ldots$), the proportions $P_m$ for each class of functions is defined by the ratio of the number of functions $N_m(t)$ belonging to class $m$ and the total number of functions in the population, denoted by $N_{\text{pop}}$. We denote these proportion (fractions) by $P_m(t) \equiv N_m(t)/N_{\text{pop}}$ and they naturally fulfill $\sum_{m=1}^{M} P_m(t) = 1$. For function spaces associated with parse-tree representations of small depth, the corresponding initial proportions $P_m(t = 0)$ can be determined directly from the possible structures of the tree. This means that we can determine the total *fitness probability density* for each generation $\rho_{\text{tot}}(f, t)$. It follows naturally as

$$\rho_{\text{tot}}(f, t) \equiv \sum_{m=1}^{M} P_m(t) \rho_m(f, t) \tag{1}$$

This is the central quantity of symbolic regression. The ultimate goal of the evolutionary algorithm [27] is to iteratively change the type of function classes, their proportions and their fitness densities for each generation, such that the density $\rho_{\text{tot}}(f, t)$ can be maximized for small arguments $f$.

## Probabilistic Theory for the Tournament Selection Scheme

In order to study the dynamics of the three types of key characteristics $\rho_{\text{tot}}(f, t)$, $P_m(t)$ and $\rho_m(f, t)$ from generation to generation, we have to examine first the tournament selection from a probabilistic perspective. As mentioned above, here out of the group of all $N_{\text{pop}}$ individuals, $n_T$ are randomly chosen to participate in each tournament. The tournament winner in this subset competition is the one with the minimum fitness among a set of $n_T$ alternatives and it is then selected to move on to the next generation. In other words, $\rho_{\text{tot}}(f, t + 1)$ is precisely the distribution of the minima (=winners) of groups of $n_T$ samples randomly drawn from the generation associated with $\rho_{\text{tot}}(f, t)$. This parameter $n_T$ naturally controls the selection pressure, we note that even for the least pressure, i.e., for the least elitist scheme with $n_T = 1$, there is for each tree a chance of $\exp(-n_T) = 36.8\%$ to not be selected, even if its fitness value happens to be the best in that entire generation. Due to this "omission" mechanism, this average fitness (and also the fitness of the best individuum) does not necessarily decrease monotonically from generation to generation. In the opposite (most elitist) scheme, $n_T = N_{\text{pop}}$, the new generation contains $N_{\text{pop}}$ identical copies of that individuum which had the least fitness of the prior generation.

The probability density $\rho_{\text{tot}}(f, t+1)$ can be computed from $\rho_{\text{tot}}(f, t)$ of the prior generation based on the following probabilistic arguments: Given the original distribution of fitnesses $\rho_{\text{tot}}(f, t)$, the probability that a randomly picked individuum has a fitness that is larger than f is given by the integral $\int_f^\infty \mathrm{d}f' \ \rho_{\text{tot}}(f', t) = 1 - \int_0^f \mathrm{d}f' \ \rho_{\text{tot}}(f', t)$. Likewise, if $n_T$ individual are chosen from the same original group, then the probability that all of the associated $n_T$ fitnesses are larger than f is given by $\Pr(n_T) \equiv [1 - \int_0^f \mathrm{d}f' \ \rho_{\text{tot}}(f', t)]^{n_T}$, where the simple product form reflects the fact that these consecutive random picks are *uncorrelated* with each other. This expression for $\Pr(n_T)$ has to be identical to the probability that a randomly picked fitness among the new group of tournament winners with density $\rho_{\text{tot}}(f', t+1)$ takes a value larger than $f$. The latter would be calculated as $\Pr(n_T) = \int_f^\infty \mathrm{d}f' \ \rho_{\text{tot}}(f', t)$. As these two expressions for $\Pr(n_T)$ have to be identical to each other for each value of f, we can obtain $\rho_{\text{tot}}(f, t+1)$ as $-\mathrm{d}\Pr(n_T)/\mathrm{d}f$. Applying this derivative with respect to $f$ to the original product form of $\Pr(n_T)$, we obtain

$$\rho_{\text{tot}}(f, t+1) = \rho_{\text{tot}}(f, t) n_T \left[ 1 - \int_0^f \mathrm{d}f' \ \rho_{\text{tot}}(f', t) \right]^{n_T - 1} \tag{2}$$

This means that the orginal fitness density $\rho_{\text{tot}}(f, t)$ gets improved by multiplying it with a collective shape-changing function, given by $S(f, t) \equiv n_T[1 - \sum_{m=1}^M P_m(t) \int_0^f \mathrm{d}f' \rho_m(f', t)]^{n_T - 1}$. This is apparently the key operator for the tournament-based selection as it describes the highly nonlinear impact of this selection scheme on the fitness density. Using the relationship $\rho_{\text{tot}}(f, t) = \sum_{m=1}^M P_m(t) \rho_m(f, t)$ from Eq. (2), the new density can naturally be written in terms of the individual original fitness densities for each class as

$$\rho_{\text{tot}}(f, t+1) = \sum_{m=1}^M P_m(t) S(f, t) \rho_m(f, t) \tag{3}$$

If we integrate both sides of this equation over all fitness values $f$, we obtain $\int_0^\infty \mathrm{d}f' \rho_{\text{tot}}(f', t+1)$ and therefore $\sum_{m=1}^M P_m(t) \int_0^\infty \mathrm{d}f \ S(f, t) \rho_m(f, t) \equiv \sum_{m=1}^M P_m(t+1)$. This equation suggests that the new fraction $P_m(t+1)$ of the individuals of class $m$ after the tournament can be obtained from the integral

$$P_m(t+1) = P_m(t) \int_0^\infty \mathrm{d}f \ S(f, t) \rho_m(f, t) \tag{4}$$

As the shape-changing function S(f,t) itself is a highly nonlinear function of the $P_m(t)$ as well as the $\rho_m(f, t)$, the iterative map from the set of the proportions $P_m(t)$ to their new values $P_m(t+1)$ for the next generation turns out to be rather nontrivial.

Using the new proportions $P_m(t+1)$, the new density at time $t+1$ can be expressed as $\rho_{\text{tot}}(f, t+1) = \sum_{m=1}^M P_m(t+1) \rho_m(f, t+1)$, which effectively defines the new individual densities $\rho_m(f, t+1)$. As we can rewrite the total density as $\rho_{\text{tot}}(f, t+1) = \sum_{m=1}^M P_m(t) \rho_m(f, t) S(f, t)$ [see Eq. (3)], this suggests that the new density for class m is given by the multiplication $\rho_m(f, t+1) = P_m(t) S(f, t) \rho_m(f, t) / P_m(t+1)$. If we insert the normalization factor $\int_0^\infty \mathrm{d}f \ S(f, t) \ \rho_m(f, t)$ from Eq. (4), we obtain the iterative map

$$\rho_m(f, t+1) = S(f, t) \rho_m(f, t) / \int_0^\infty \mathrm{d}f' \ S(f', t) \ \rho_m(f', t) \tag{5}$$

There are two particular limits of the general equation (4), for which more simplified expressions for the new proportions of each class $P_m(t+1)$ can be obtained. In the first limit, where the densities $\rho_m(f, t)$ are so narrow as a function of f that they do not overlap with each other, it is possible to derive a much simpler and direct iterative scheme to compute the time evolution of the proportions $P_m(t+1)$ directly from the set of all $P_m(t)$. In Appendix A, we derive that these equations take the form

$$P_m(t+1) = \left[ 1 - \sum_{m'=1}^{m-1} P_{m'}(t) \right]^{n_T} - \left[ 1 - \sum_{m'=1}^{m} P_{m'}(t) \right]^{n_T} \tag{6}$$

As these equations assume that the associate average fitnesses $f_m$, defined as $f_m \equiv \int_0^\infty \mathrm{d}f \, f \ \rho_m(f)$, can be ordered, i.e., $f_m < f_{m+1}$, the $P_m(t+1)$ do not depend on $\rho_m(f)$ nor $f_m$. In Appendix A, we also provide a simple numerical example of time evolution. For example, we show there that the proportion associated with the best fit individua ($m = 1$) approaches $P_1(t) \rightarrow 1$ in times as $P_1 = 1 - [1 - P_1(0)]^{t \ n_T}$. If we solve $P_1(t_{1/2}) = 0.5$ for time $t_{1/2}$, we find that after a number of generations given by (the integer part of) $t_{1/2} \equiv -n_T^{-1} \ln 2 / \ln[1 - P_1(0)]$, the fraction $P_1$ becomes the dominant function class as it exceeds 50%.

This solution for $t_{1/2}$ allows us to suggest that the total CPU time (required for the best proportion to reach 50%) is actually *independent* of our choice of $n_T$. As each single tournament requires the evaluation of $n_T$ fitnesses, and we require a total of $N_{\text{pop}}$ tournaments to select all $N_{\text{pop}}$ members of the next generation, the CPU-time for one generation is proportional to $n_T\, N_{\text{pop}}$. This means that the total CPU-time for a total number of $t_{1/2}$ generations (given by $t_{1/2}\, n_T\, N_{\text{pop}}$) does not depend on our choice for $n_T$ at all. This is interesting, as the evaluation of the fitnesses are the typical computational bottleneck in SR.

Let us now discuss the second (opposite) limit of Eq. (4), where the overlap of all fitness densities is assumed to be maximum such that we can assume that the $\rho_m(f,t)$ are independent of m. This means that the factor $\int_0^\infty \mathrm{d}f\; S(f,t)\; \rho_m(f,t)$, which we abbreviate as $\varpi(t)$, is identical for each class and, according to Eq. (4) leads to $P_m(t+1) = P_m(t)\varpi(t)$. In order to preserve the normalization $1 \equiv \sum_{m=1}^{M} P_m(t+1)\varpi(t)$, we find that the proportions of each class do not change in time, $P_m(t) = P_m(t=0)$. For a general evolution, the dynamics of the $P_m(t)$ will be somewhere between these two limiting cases. We compare it to the evolution of a concrete example from symbolic regression in the next section.

## An Illustration of the Probabilistic Theory of the Symbolic Regression for $M = 3$ Function Classes

Without any loss of generality, we assume here that the input data were derived from a target function that is simply given by the constant $g_{\text{target}}(x) = 0$ in the range $0 \le x \le 1$. To keep this example as transparent and illustrative as possible, let us define the primitive set of initial candidate functions by those that can be represented by a syntax tree of depth $D = 1$ and corresponding tree length $L = 3$ with the sole arithmetic operation "add". This means that the two input terminals (leaves) represent either the independent variable $x$ or a constant $c$. This space can represent only $M = 3$ different types of mathematical functional relationships. They are given by the three sets $g_1(x) = c_1 + c_2$ (denoted by class $m = 1$), $g_2(x) = x + c_1$ (class $m = 2$) and $g_3(x) = 2x$ (class $m = 3$). While this is obviously an extremely simple toy model, it actually contains many aspects of a real SR algorithm.

Let us assume that the computer algorithm is set up in such a way that the numerical values of all constants for the members of the 0th generation are chosen to be equally distributed inside the interval $0 \le c \le 1$. This means that $p_{\text{para}}(c)$ is just a product of uniform distributions for each constant, each centered at 0.5 and with a standard deviation 0.288. Using $\rho \equiv \int_{-\infty}^{\infty} \mathrm{d}^n c\; \delta[f' - f(\mathbf{c})] p_{\text{para}}(\mathbf{c})$, we obtain for the specific fitness densities for each class

$$\rho_1(f, t=0) = fU(f, 1-f) + (2-f)U(f, 1-f) \tag{7}$$

$$\rho_2(f, t=0) = U(f-0.5, 1.5-f) \tag{8}$$

$$\rho_3(f, t=0) = \delta(f-1) \tag{9}$$

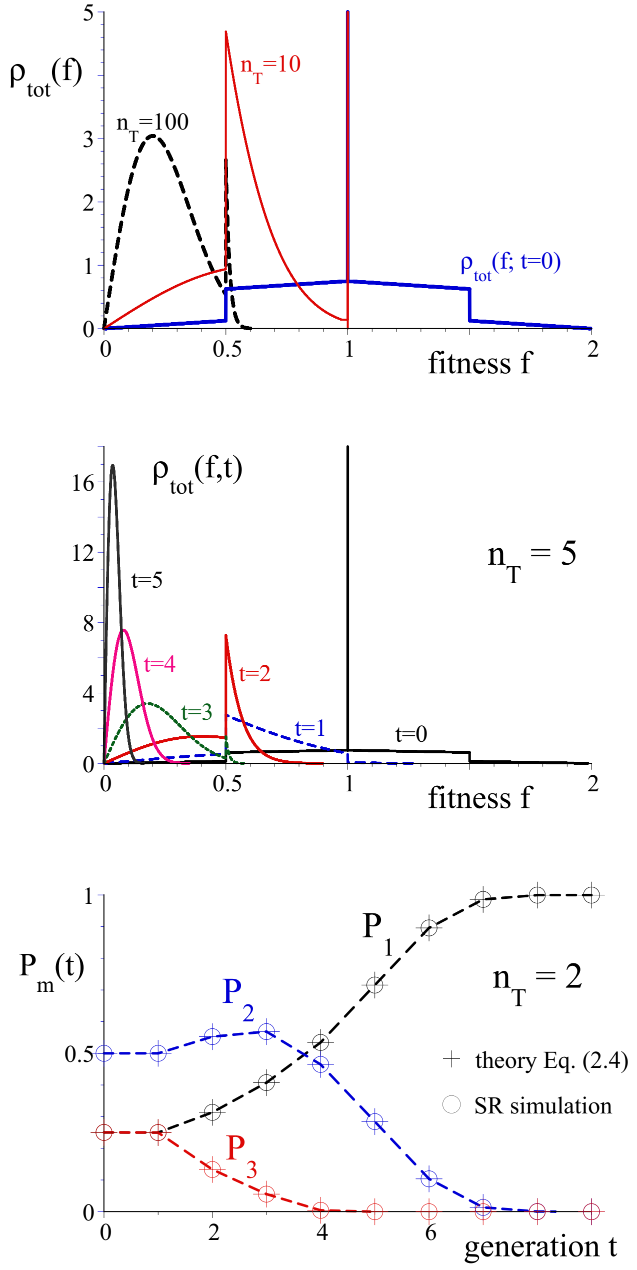

**Fig. 1** (**a**, top) Change of the fitness probability density $\rho_{\text{tot}}(f)$ (blue) after a tournament selection of sizes $n_T = 10$ and 100. They take the average values $f = 1.0, 0.55$ and 0.25. (**b**, middle) The evolution of the fitness probability density $\rho_{\text{tot}}(f,t)$ for the first five generations for the tournament size $n_T = 5$. (**c**, bottom) The time-dependence of the three proportions $P_1(t)$, $P_2(t)$ and $P_3(t)$ for tournament sizes $n_T = 2$ for a simulation with a population of 20,000 together with the analytical predictions of the probabilistic theory of Eq. (4)

where $U()$ denotes the generalized unit step function defined as $U(a, b) = 1$ if both $a \geq 0$ and $b \geq 0$ and $U(a, b) = 0$ otherwise. Here $\delta(f - 1)$ denotes the Dirac delta function. All three densities happen to have the same average fitness value $f_m = \langle f \rangle = \int_0^\infty \mathrm{d}f\, f\, \rho_m(f, t = 0) = 1$. While all three densities vanish for $f = 0$, the probability to find a candidate function for which $\int_0^{f_{\max}} \mathrm{d}f\, \rho_m(f, t = 0) > 0$ for any arbitrarily small (but nonzero) $f_{\max}$ does not vanish for class one, i.e., $\rho_1(f, t = 0)$. This means that the set of class $m = 1$ candidate functions contains the target function $g_{\text{target}}(x) = 0$. Obviously, $g_1(x) = c_1 + c_2$ is equal to the target, if both $c_1$ and $c_2$ approach zero.

In Fig. 1a, we graph the modification of the original fitness probability $\rho_{\text{tot}}(f, t = 0) = \sum_{m=1}^{3} P_m(t = 0)\rho_m(f, t = 0)$ due to two tournaments of sizes $n_T = 10$ and 100 as predicted from the numerical evaluation of Eq. (2). We see that the non-linear character of the tournament selection manifests itself in the formation of rather unusually shaped densities. Their precise shape of the new densities can hardly be guessed, but we see the expected overall shift to smaller fitness values. We also see that the density $\rho_{\text{tot}}(f, t = 0)$, which due to the original uniform distributions of the constants had discontinuities at $f = 0.5$, 1 and 1.5, develops into a smoother structure with increasing $n_T$.

In Fig. 1b, we display the dyamical evolution of the fitness density $\rho_{\text{tot}}(f, t)$ for the first five generations for a fixed tournament size $n_T = 5$. Overall, the trend with increasing time is very similar to the data shown in Fig. 1a as it reflects how the three fitness densities are deformed. In addition to the deformation, also the weight factors of each function $P_m(t)$ class are changing.

In Fig. 1c, we monitor the (generational) time evolution of the three proportions from there initial values $P_1(t) = 0.25$, $P_2(t) = 0.5$ and $P_3(t) = 0.25$ for the same tournament size as in Fig. 1b. We find that (independent of the size $n_T$) the evolution in each case approaches $P_1(t \to \infty) = 1$. However, the evolution is non-monotonic, for example, we see that $P_2(t \to \infty)$ grows first, before it decays to zero. As there are no approximations, the analytical data based on the numerical solutions to the iterative set agree perfectly with the fractions of the actual SR simulation with sufficient large population ($N_{\text{pop}} > 20,000$) as shown by the open circles. In summary, qualitatively, the dynamics of the proportions in this SR simulation are similar to the corresponding predictions of the non-overlapping fitness model introduced in Appendix A. This agreement gives some credence to this simpler model, even though the original fitness densities of Eqs. (7–9) did have a significant overlap.

## The Constant-Mutation Operator (CMO)

### Operational Definition and Its Impact on the Fitness Density

In contrast to most genetic operators (such as mutations or two- or multi-parent crossovers) that are created to mainly increase the diversity and permit us to explore new areas, the CMO operator introduced below is a special purpose operator that exploits a known area and aims at increasing the convergence to a minimum fitness within a given class of individuals. To overcome the principal bottleneck of most SR algorithms, which is the slow convergence with regard to reproducing the (non-integer) constants contained by the target function, we have supplemented the usual mutation and crossover operations by a new one that does not change the function class, but improves on the numerical value(s) of the constant(s) contained in an individual. If the CMO is being selected to act on the respective tournament winner, it first identifies the associated constant parameters of this expression. If it does not contain any, it acts as the unit-operator and keeps this individuum unchanged. If the expression contains numerous constants, it randomly picks out of this expression any constant $c$. The fitness value of this expression $f(c)$ is then compared to $f(g)$, where $g$ denotes an alternative random guess chosen from a given probability density $G(g)$. If $f(c) \leq f(g)$, then original value of the parameter is maintained and CMO would act as the unit operator. If, however, $f(g) \leq f(c)$, then the (better) numerical value $g$ will be inserted into the individuum and forwarded to the pool of search candidates for next generation. The CMO is therefore a very specialized version of the point mutation operator, which acts on any node independent of whether it contains a constant, an operation or a variable.

As during the early generational time frame (of mainly exploratory searches) most of the fitness densities for each function class have typically widely distributed parameter values, the particular numerical choice for $g$ is not so relevant. However, as later generation are comprised of mutated former tournaments winners, we can assume that the fitness density $\rho_m(f)$ of each class has improved already that the associated value of the parameter $c$ is already close to one that can minimize the fitness. This means that the choice of $g$ should take this knowledge of $c$ into account. Therefore $g$ should be correlated to $c$ suggesting that the more general $G = G(g, c)$ is more efficient. This also means that it becomes a conditional probability density, which also depends on the value for $c$, such that $\int G(g, c) = 1$ for any $c$. Furthermore, we will show below, that adapting the shape of $G(g, c)$ not only to $c$, but permitting it to change from generation to generation, i.e., $G = G(g, c, t)$, can further accelerate the rate of convergence.

To establish which functional forms of $G(g, c, t)$ are a priori most advantageous, we need to establish first a theoretical framework for how the action of the CMO modifies the fitness probability densities. Unfortunately, this action cannot be formally expressed as $\rho_m(f, t+1) \equiv \mathrm{CMO}[\rho_m(f, t)]$, as it is actually not a unique one-to-one mapping. However, for the iteration map $p_{\text{para}}(c, t+1) \equiv \mathrm{CMO}[p_{\text{para}}(c, t)]$ an explicit and unique operational scheme can be found based on the corresponding joint (and fully correlated) probability density $p_{\text{para}}(c, t)G(g, c, t)$. It is given by

$$\begin{aligned} p_{\text{para}}(c, t+1) &= \mathrm{CMO}[p_{\text{para}}(c, t)] \\ &= \int\int \mathrm{d}c'\mathrm{d}g'\delta[c - \mathrm{ArgMin}_{(c'\in R, g'\in R)} \\ &\quad (f(c'), f(g'))] \\ &\quad p_{\text{para}}(c', t)G(g', c', t) \\ &= \int\int_D \mathrm{d}c'\mathrm{d}g'\delta[c - c']p_{\text{para}}(c', t)G(g', c', t) \\ &\quad + \int\int_E \mathrm{d}c'\mathrm{d}g'\delta[c - g']p_{\text{para}}(c', t)G(g, c', t) \end{aligned} \tag{10}$$

where we denote with $D$ the domains in two-dimensional integration $(c', g')$ space, where the fitness satisfies $f(c') \le f(g')$, meaning that here we have $\mathrm{ArgMin}_{(c'\in R, g'\in R)}(f(c'), f(g')) = c'$. Similarly, the complementary area $E$ is defined by $f(g') \le f(c')$, such that $\mathrm{ArgMin}_{(c'\in R, g'\in R)}(f(c'), f(g')) = g'$.

As we can safely assume that after several generations the distribution of values of the parameter $c$ is already close to the one that can minimize the fitness density, we can assume that in this region the fitness is a monotonically increasing function of $|c - c_o|$. Here $c_o$ is the specific parameter value that minimizes the fitness. This monotonicity permits us to identify the integration spaces $D$ and $E$ and to perform the integration along the $c'$ or $g'$ axis. We obtain

$$\begin{aligned} &p_{\text{para}}(c, t+1) \\ &= \left[ \int_{-\infty}^{\mathrm{Min}(2c_o - c, c)} + \int_{\mathrm{Max}(2c_o - c, c)}^{\infty} \right] \mathrm{d}c' p_{\text{para}}(c', t)G(c, c') \\ &\quad + p_{\text{para}}(c, t)\left[ 1 - \int_{\mathrm{Min}(2c_o - c, c)}^{\mathrm{Max}(2c_o - c, c)} \mathrm{d}g' G(g', c) \right] \end{aligned} \tag{11}$$

where we use $\int_{-\infty}^{\infty} \mathrm{d}g' G(g', c, t) = 1$. These expressions permit for an illustrative interpretation. For example, the prefactor [1−...] to $p_{\text{para}}(c, t)$ is the total probability that a randomly picked value for $g'$ is further away from $c_o$ than the value $c$,

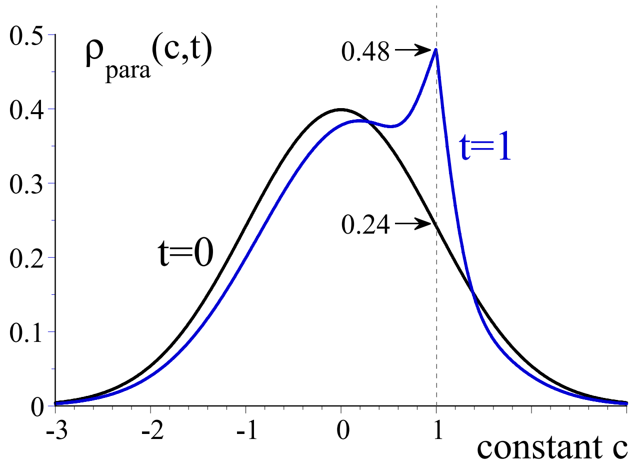

**Fig. 2** Illustration of the probability density doubling feature of the CMO operation for the fitness minimizing parameter $c = c_o$. We used $p_{\text{para}}(c, t) = (2\pi)^{-1/2}\mathrm{Exp}(-c^2/2)$ and $G(g, c) = (2\pi)^{-1/2}\sigma^{-1}\mathrm{Exp}[-(g - c)^2/(2\sigma^2)]$ with $\sigma = 0.5$. We see how the particular value of the density at the optimal parameter (chosen here to be $c_o = 1$) is doubled from $p_{\text{para}}(c_o, t) \approx 0.24$ to $p_{\text{para}}(c_o, t+1) \approx 0.48$

meaning that $f(c) < f(g')$. In other words, in this case, a proposed new value $g$ cannot improve the fitness originally associated with value $c$, distributed according to $p_{\text{para}}(c, t)$. The same interpretation applies to the first term, where the associated contribution of the weighted density $G(c, c')$ to $p_{\text{para}}(c, t+1)$ is provided.

While $p_{\text{para}}(c, t)$ does not necessarily take its maximum at $c = c_o$, the action of the CMO operation on $p_{\text{para}}(c, t)$ has an interesting "probability density doubling" property, when evaluated at the optimal parameter value $c = c_o$. This can be easily seen, if we evaluate Eq. (11) for the specific value $c = c_o$. It simplifies to $p_{\text{para}}(c_o, t+1) = \int_{-\infty}^{\infty} \mathrm{d}c' p_{\text{para}}(c', t)G(c_o, c') + p_{\text{para}}(c_o, t)$ . If we assume that $G(c_o, c')$ is sufficiently narrowly centered around $c' = c_o$, we can factor $p_{\text{para}}(c_o, t)$ out of the integral and using $\int_{-\infty}^{\infty} \mathrm{d}g' G(g', c) = 1$ we obtain $p_{\text{para}}(c_o, t+1) = 2p_{\text{para}}(c_o, t)$. This interesting feature is illustrated in Fig. 2, where we show the change of a initial Gaussian parameter distribution $\rho_{\text{para}}(c) = (2\pi)^{-1/2}\mathrm{Exp}(-c^2/2)$ under the CMO operation based on the conditional density $G(g, c) = (2\pi)^{-1/2}\mathrm{Exp}[-(g - c)^2/(2\sigma^2)]$.

While to have a fully analytical access to the way the action of the CMO modifies the distribution of the parameter from $p_{\text{para}}(c, t)$ to $p_{\text{para}}(c, t+1)$ is beneficial, what is more relevant is how the associated fitness density $\rho(f, t)$ is improved to $\rho(f, t+1)$. To convert the probability distribution $p_{\text{para}}$ back to the associated density of the fitness, i.e.,$\rho(f) = \int \mathrm{d}c\ \delta[f - f(c)]p_{\text{para}}(c)$ can be easily obtained in general, if we know how the fitness value $f$ is related to the parameter $c$. With the ultimate goal in mind that a highly desirable density should be narrowly localized close to the minimum value of $f$, there are obviously numerous single values that could be defined to measure and quantitatively compare the quality of several fitness densities. A good example could be either the mode of this distribution or the average fitness value defined as $\langle f \rangle \equiv \int_0^\infty \mathrm{d}f'\, f' \rho(f')$. The computation of the mean value can be quite conveniently and directly obtained from $p_{\text{para}}(c)$. If we exchange the integration order and perform the integration of the Dirac delta function, then $\langle f \rangle \equiv \int_0^\infty \mathrm{d}f'\, f' \int_{-\infty}^\infty \mathrm{d}c\, \delta[f' - f(c)] p_{\text{para}}(c) = \int_{-\infty}^\infty \mathrm{d}c f(c) p_{\text{para}}(c)$. If we apply the same procedure to the CMO modified density $p_{\text{para}}(c, t+1)$, we can perform the delta-function integration over $c$ first and then over $f'$ and obtain

$$
\begin{aligned}
\langle f \rangle (t+1) &\equiv \int_0^\infty \mathrm{d}f'\, f' \rho(f', t+1) \\
&= \int_0^\infty \mathrm{d}f'\, f' \int_0^\infty \mathrm{d}z\, \delta(f' - f(c)) p_{\text{para}}(c, t+1) \\
&= \int_{-\infty}^\infty \mathrm{d}c' \int_{-\infty}^\infty \mathrm{d}g'\, f(\text{ArgMin}[f(c'), f(g')]) \\
&\quad p_{\text{para}}(c', t) G(g, c', t)
\end{aligned} \tag{12}
$$

As $\langle f \rangle (t)$ cannot be uniquely tracked back to $p_{\text{para}}(c, t)$, it is unfortunately not possible to construct a direct iteration scheme that permits us to relate $\langle f \rangle (t)$ directly to $\langle f \rangle (t+1)$. This means that if two different $p_{\text{para}}(c, t)$ happen to have the same mean value $\langle f \rangle (t)$, their CMO-improved value $\langle f \rangle (t+1)$ can be nevertheless different in general. To examine how the conditional probability $G(g', c', t)$ can be chosen to lower the average fitness $\langle f \rangle$ in a most efficient way, we need to examine a concrete example, which we present in the next section.

## Variance Matching to Optimize the Reduction of the Average Fitness

While it is obviously advantageous to have the conditional probability $G(g, c)$ centered around the original value $c$, which is distributed according to $p_{\text{para}}(c, t)$, it is not clear how the optimum width of $G(g, c)$ should be chosen to maximize the reduction of the average fitness from $p_{\text{para}}(c, t)$ to $p_{\text{para}}(c, t+1)$. To examine this question numerically, we have considered first for simplicity the uniform probability $G(g, c) \equiv B^{-1} \theta[g - (c - B/2)] \theta[(B/2 + c) - g]$ of width $B$ and a similarly simple uniform distribution $p_{\text{para}}(c, t) \equiv \theta(c + 0.5) \theta(c - 0.5)$, where $\theta$ denotes the Heaviside unit-step function. The fitness is modeled here as $f(c) = (c - c_o)^2$.

There are two special opposite cases for the choice of the width $B$ that are of interest. In both extreme limits for a very narrow distribution, $B \to 0$, as well as for $B \to \infty$, the transformation from $p_{\text{para}}(c, t)$ to $p_{\text{para}}(c, t+1)$ becomes the unit-operator. In other words, we have $p_{\text{para}}(c, t+1) = p_{\text{para}}(c, t)$ and therefore $\rho(f, t+1) = \rho(f, t)$. As a result, there is no improvement of the fitness under the CMO operation. As neither limit ($B \to 0$ nor $B \to \infty$) improves the fitness, there must be an optimum width (denoted by $B_{\text{opt}}$), for which the CMO transformation leads to a maximum reduction of the fitness value.

To find this optimum width for a given value of $c_o$, we have calculated $p_{\text{para}}(c, t+1)$ according to Eq. (10) above for a wide range of widths $B$. We have then recorded the width $B_{\text{opt}}$ that minimized the final average fitness $\langle f \rangle \equiv \int_0^\infty \mathrm{d}f'\, f' \rho(f, t+1)$, where $\rho(f, t+1) = \int \mathrm{d}c' \delta[f - c'^2] p_{\text{para}}(c', t+1)$. In Fig. 3, we have graphed the optimum width $B_{\text{opt}}$ as a function of $c_o$.

Here the distribution of the original constant $c$ $p_{\text{para}}(c, t)$ was uniform with a width of 1 and centered around 0. The conditional probability for CMO was also uniform, but was centered around $g = c$ with a variable width of $B$.

The data in Fig. 2 suggest two important conclusions. First, the optimimum width $B_{\text{opt}}$ associated with the CMO operation takes its lowest value if the original parameter distribution $p_{\text{para}}(c, t)$ is centered around the optimal parameter $c_o$, which happens if $c_o$ is equal to zero. In this particular case, the optimum width $B_{\text{opt}}$ is 1, which matches precisely the width of the original distribution $p_{\text{para}}(c, t)$. The fact that the associated variance of $p_{\text{para}}(c, t)$, i.e., $\sigma^2 = \int_{-\infty}^\infty \mathrm{d}c (c - c_o)^2 p_{\text{para}}(c, t)$, which is equal to $\sigma = 0.288$,

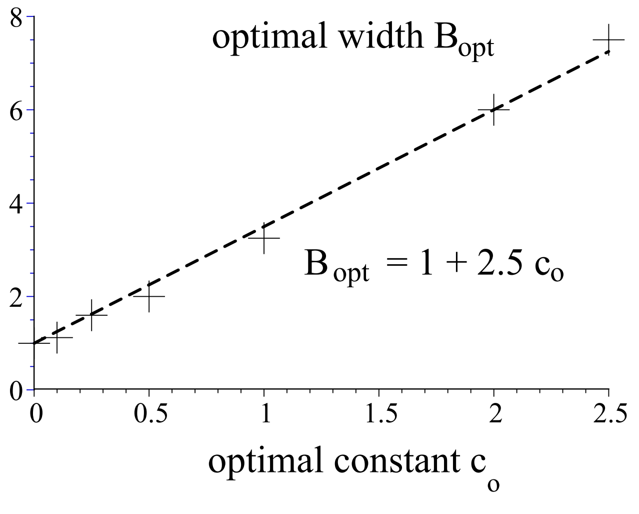

**Fig. 3** The optimum width $B$ of the conditional probability $G(g, c) = B^{-1} \theta[g - (c - B/2)] \theta[(B/2 + c) - g]$ to minimize the average $\langle f \rangle (t+1) = \int_{-\infty}^\infty \mathrm{d}c\, p_{\text{para}}(c, t+1)$. Here the fitness is modelled as $f(c) = (c - c_o)^2$ and the original density is $p_{\text{para}}(c) = \theta(c + 0.5) \theta(0.5 - c)$. For comparison, the dashed line is $B_{\text{opt}} = 1 + 2.5|c_o|$

is precisely matched by the corresponding variance of the conditional probabililty, i.e., $\int_{-\infty}^{\infty} \mathrm{d}g(g-g_0)^2 G(g,c)$, is not coincidental and valid for more general forms of the densities $G(g, c)$. For example, for a Gaussian choice, i.e., $G(g,c) = (2\pi)^{-1/2}\sigma_g^{-1}\mathrm{Exp}[-(g-c)/(2\sigma_g^2)]$, we find for $c_o = 0$ that the optimum value of the variance is $\sigma_g = 0.31$, which differs only by 7% from the associated variance 0.288 of $p_{\mathrm{para}}(c,t)$. This finding suggests that the action of the CMO can be optimal, if the variance of $G(g, c)$ is chosen to match that of $p_{\mathrm{para}}(c,t)$.

The second conclusion from Fig. 3 is that the required optimal value of $B_{\mathrm{opt}}$ increases with the distance between the center of $p_{\mathrm{para}}(c,t)$ and $c_o$. For the uniformly distributed $G(g, c)$ examined in this figure, the linear relationship $B_{\mathrm{opt}} = 1 + 2.5c_o$ approximates this increase very well. However, as the probability density doubling mechanism for $c = c_o$ discussed suggests, after a few generations and applications of the CMO, the density $p_{\mathrm{para}}(c,t)$ will naturally become centered around $c_o$, such that the variance matching of $p_{\mathrm{para}}(c,t)$ and $G(g, c)$ promises the best performance of the CMO is general.

Finally, we have to examine how the width (variance) of $p_{\mathrm{para}}(c,t)$ is reduced under consecutive actions of the CMO operation based on Eq. (11). To have another concrete example, we have used simple Gaussian distributions $p_{\mathrm{para}}(c,t=0) = (2\pi)^{-1/2}\sigma(0)^{-1}\mathrm{Exp}[-c^2/(2\sigma(0)^2)]$ and $G(g,c) = (2\pi)^{-1/2}\sigma_g^{-1}\ \mathrm{Exp}[-(g-c)/(2\sigma_g^2)]$, and computed $p_{\mathrm{para}}(c,t)$ after consecutive applications of the CMO for $t = 1, 2, \ldots, 7$. We then computed the new variances $\sigma^2(t) \equiv \int_{-\infty}^{\infty} \mathrm{d}c\ c^2 p_{\mathrm{para}}(c,t)$. While the functional forms $p_{\mathrm{para}}(c,0)$, $p_{\mathrm{para}}(c,1)$, ..., are quite different from each other, the variances $\sigma^2(t)$ are reduced by an indentical factor $\zeta$ that is independent of $t$, i.e., $\sigma(t+1)/\sigma(t) = \zeta$. This time independent factor $\zeta$ is a decreasing function of the ratio $\sigma(0)/\sigma_g$. For example, for $\sigma_g = \sigma_0$ we find $\zeta \approx 0.85$, for $\sigma_g = \sigma(0)/2$ we find $\zeta \approx 0.88$, or for $\sigma_g = \sigma(0)/3$ we find $\zeta \approx 0.91$. This means that the reduction of the width for consecutive CMO operation is simply given by the power law $\sigma(t) = \sigma(0)\zeta^t$. In Sect. 5 we will see the impact of this scaling with regard to the fitness reduction in a full SR simulation.

## The Scaling of the Variance Reduction of $p_{\mathrm{para}}(c,t)$ with the Tournament Size $n_T$

As for an efficient application of the CMO operation the value of the variance $\sigma^2$ of the parameter density $p_{\mathrm{para}}(c)$ plays an important role, we discuss here how it is reduced due to the tournament selection. To have a simple model, we assume that $p_{\mathrm{para}}(c)$ is symmetric around $c = c_o = 0$, which minimizes the associated fitness. Following a similar statistical argument based on the total probability from Sect. 2.3, one can show that $p_{\mathrm{para}}(c,t)$ changes to the distribution of tournament winners $p_{\mathrm{para}}(c,t+1)$ according to

$$p_{\mathrm{para}}(c,t+1) = n_T \rho(c,t) \left[ 1 - 2\int_0^{|c|} \mathrm{d}c' \rho(c',t) \right]^{n_T - 1} \tag{13}$$

To take a concrete example, if $\rho(c',t) = (2\pi)^{-1/2}\sigma^{-1}\mathrm{Exp}[-c^2/(2\sigma^2)]$, then we can use Eq. (13) to compute numerically the new variance $\sigma^2(t+1) \equiv \int_{-\infty}^{\infty} \mathrm{d}c\ c^2 p_{\mathrm{para}}(c,t+1)$. In the range $1 \le n_T \le 50$ the fitted expression given by $\sigma(t+1) \approx \sigma(t)1.65/(0.65 + n_T)$ matches the true tournament size dependence of $\sigma(t+1)$ with an error of less than 6%. In other words, we can assume that for large $n_T$ the application of the tournament reduces the width by a factor that is inversely proportional to $n_T$. To gain some confidence into the universality of this scaling of the width, we have repeated the calculation with a uniform density $\rho(c,t) = \theta(c+0.5)\theta(0.5-c)$. For this distribution, a similar relationship $\sigma(t+1) \approx \sigma(t)2.4/(1.4 + n_T)$ matches the true dependence of $\sigma(t+1)$ with an error of less than 1%.

As the CPU time increases linearly with $n_T$ due to the required ranking for each tournament, one can address the question if the application of two consecutive tournaments (with a small size $n_{T,2}$ each) is more effective in the reducing the width s than the application of a single tournament (with a larger $n_{T,1}$). If we assume the general dependence $\sigma(t+1, n_T) \approx \sigma(t)(\xi + 1)/(\xi + n_T)$ suggested above and equate the corresponding two widths $\sigma(t+1, n_{T,1}) = \sigma(t+2, n_{T,2})$, the same starting width $\sigma(t)$ cancels out and we obtain $n_{T,1} = (\xi + n_{T,2})^2/(\xi + 1) - \xi$. For example, if we assume $\xi = 1$ and $n_{T,2} = 5$, then only for $n_{T,1} \ge 17$ we obtain $\sigma(t+1, n_{T,1}) \le \sigma(t+2, n_{T,2})$. As here the required $n_{T,1}$ is much more than twice of $n_{T,1}$, two consecutive tournament selections (with $n_{T,2}$) are clearly much more CPU time efficient in reducing the fitness than a single selection.

## Impact of the Adaptive CMO in Symbolic Regression with Mutations and Cross-overs

After the probabilistic analysis of the prior sections, where the statistical features of the tournament selection and the CMO were examined in isolation, we will now provide an example of a practical situation. Here numerous evolution operations can act on the tournament winners, such as the traditional crossovers, and subtree-, point- and hoist-mutations as well as simple reproductions with the specified probabilities. In Table 1, we summarize the main characteristics of these SR evolutions. To keep the numerical example as

**Table 1** Parameters used in the symbolic regression leading to the eight curves in Fig. 4. For the point mutation operation, each node or leaf had a 20% replacement probability. For the plus-selection, an offspring only advances if its fitness is less than its parent, otherwise the parent advances

| | |
|---|---|
| Evolution characteristics | |
| Selection scheme | Tournament |
| Tournament style | Deterministic |
| Tournament size $n_T$ | 20 |
| Selection type | Plus |
| Operation set | $\{+, *\}$ |
| Terminal set | $\{c, x\}$ |
| Uniform constant range | [0, 1] |
| Initial length of trees L | 3 |
| Population size $N_{\text{pop}}$ | 5000 |
| Number of generations | 31 |
| Criterion for fitness | rmse |
| Probabilities of evolutionary operations | |
| Two-parent crossover | 50% |
| Sub-tree mutation | 10% |
| Hoist mutation | 10% |
| Point mutation, red curve | 10% |
| Point mutation, other curves | 0% |
| CMO, red curve | 0% |
| CMO, other curves | 20% |
| Reproduction, red curve | 10% |
| Reproduction, other curves | 0% |

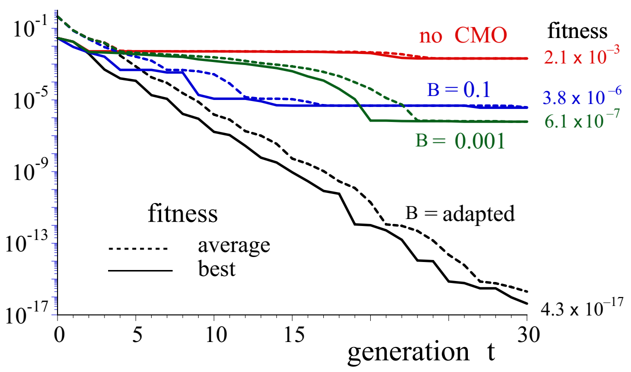

**Fig. 4** The fitness $f = \{\sum_{n=1}^{20}[y_n - g(x_n)]^2\}^{1/2}$ associated with the best individuum (and the average) per generation as a function of the generational time $t$. The conditional probability for the CMO was $G(g,c) = B^{-1}\theta[g - (c - B/2)]\theta[(B/2 + c) - g]$

simple and reproducable as possible, we have used the target function $g_{\text{target}}(x) = 0.5 + 0.1x^2$, which was sampled at 20 positions $x_n = (n-1)/20$ for $n = 1, 2, \ldots, 20$. So the input data is the set of 20 points $x_n, y_n$.

To have a general case where the initial generation does not contain the function class of the target function $g_{\text{target}}(x)$, we purposely restricted the initial generation to trees of small length $L = 3$. This population contains only the $M = 4$ different function classes $c_1$, $c_1 + x$, $c_1 x$ and $x^2$. Among these four classes, the first class with $g_1(x) = 0.53087$ with an (rmse) fitness of 0.0284 is the best fit. In order to add the relevant function class $c_1 + c_2 x^2$ to the population (which has the minimum tree length $L = 7$), the multiple applications of the mutation and crossover operations over several generations are required.

In Fig. 4, we compare the evolutionary improvement of the fitness associated with the best individum of all $N_{\text{pop}} = 5000$ search functions (and also the average fitness) over 30 generations for three groups of strategies. The red graph relies solely on a mixture of traditional evolution operators as indicated in Table 1. For the next group, represented by the blue and green graphs, the CMO operator with a fixed width $B = 0.1$ and 0.001 and uniform conditional probability density $G(g,c) = B^{-1}\theta[g - (c - B/2)]\theta[(B/2 + c) - g]$, was included with a probability of 20% in the evolution. The black graph is based on a variable width $B$, where a fixed $B = 0.01$ (to form the generations $1 \le t \le 5$) was followed by $B(t) = 10^{-0.5t}$ for $6 \le t$.

Let us now discuss the success rate of these three evolution strategies separately. The consecutive actions of the traditional evolution operators manage to decrease the fitness of the best individuum from its initial value $f = 0.0284$ for $t = 0$ down to $f = 2.1 \times 10^{-3}$ for $t = 30$. We note that this final expression was actually represented by a complicated tree of large length, containing combinations of many different constants. As we pointed out in the introduction, the traditional operations cannot directly update these numerical assignments to constants. Therefore, the only way of obtaining improved numerical values is using operational combinations of multiple constants. While these operations can reduce the optimum fitness by less than one order in magnitude during the first two generations, during the following 22 generations (up to $t = 24$) the improvements are miniscule.

This problem is significantly improved when the CMO operation (with a reasonable width $B$) is permitted to participate in the evolution. For example, for $B = 0.1$, the fitness is steadily reduced during the first 15 generations, after which this choice of $B$ is too large to effectively further reduce the fitness. In contrast, for a width a hundred times smaller $B = 0.001$, the fitness decreases at a much smaller rate (up to $t \approx 15$), as $B$ is too small compared to the average width of the parameter distribution during those generations. However, once this average width becomes comparable to the actual fitness width of the search function, CMO can act much more effectively and the fitness begins to reduce more efficiently. After 20 generations, however, average width has become less than $B$, and the CMO loses its ability to finetune the corresponding parameters in the optimum search function.

The black graph presents the best of both worlds as it continues to shrink the width of the CMO operation with the average parameter width. Here the final fitness is less that $4.3 \times 10^{-17}$ corresponding to the optimum $g(x) = 0.5 + 0.1x^2$, where each of the two constants is correct for all of its significant (16) digits. Furthermore, the nearly straight line for the logarithmic axis shows that the reduction of the fitness is nearly perfectly exponential, i.e., $f_{\text{opt}}(t) = 0.01 \times e^{-1.1t}$. Quite interestingly, the average fitness of the entire population follows a similar trend, i.e., $\langle f \rangle(t) = 0.2 \times e^{-1.1t}$.

In summary, we note that the CMO with adaptive variance matching can improve the final optimum fitness after 30 generations from $f_{\text{opt}} = 2.1 \times 10^{-3}$ to $f_{\text{opt}} = 4.3 \times 10^{-17}$, which is a highly impressive improvement of about 13 orders in magnitude. Equivalently, this corresponds to a change in accuracy of the parameters $c_1$ and $c_2$ from 3 to 16 significant digits.

## SR-Guided Reconstruction of Iterative Maps or Differential Equations from Chaotic Time Series

### The Logistic Map with Noise

Since its popularization by May in 1976 [32], the logistic map $x_{n+1} = rx_n(1 - x_n)$ has become the archetypical prototype of a nonlinear recurrence relation of degree 2. Due to its non-linear dissipative and non-invertible structure, all iterates converge to either regular $n$-cycles or to chaotic limit cycles, which depend on the control parameter $r$. In 1999, Tan has demonstrated [33] that some state-of-the-art machine learning technique based on recurrent parametrization networks can correctly predict the chaotic dynamics $(3.4 < r < 4)$ from the non-chaotic training time series of the stable regular regime $(0 < r < 3.4)$. We use this system here as a concrete example to show that SR can also recover the governing discrete map from its iterates $x_n$, requiring much less computational effort.

We have computed the first five iterates $x_n(r_j)$ $(n = 0, 1,..., 4)$ with the initial value $x_0 = 0.9$ for the two control values $r_1 = 1$ and $r_2 = 2$. In this regular domain all iterates approach a single fix point given by $x_{n=\infty} = 0$ (for $0 \le r < 1$) and $x_{n=\infty} = (r-1)/r$ (for $1 \le r < 3$). The SR algorithm was then fed a list of eight triples $Y_{\text{input}} = \{x_{n+1}(r_j), x_n(r_j), r_j\}$ as training sets with the goal to discover the most accurate and simplest expression that can relate $x_{n+1}(r_j)$ to $x_n(r_j)$ and $r_j$. Using an initial population of $N_{\text{pop}} = 40,000$ trial expressions, the algorithm predicted correctly the logistic map $x_{n+1} = rx_n(1 - x_n)$ after only a few seconds of CPU time.

To examine the robustness with regard to numerical inaccuracies, we have superimposed a 1% multiplicative noise to each of the eight iterates, by multiplying each of the exact iterates with 1.01 Rand, where Rand was a uniformly distributed random number between −1 and 1. For these noisy data, the SR algorithm predicted $x_{n+1} = rx_n(0.992 - x_n)$, which is still remarkably close to the functional form of the logistic map, albeit with a slightly different parameter. We have then repeated the same simulation, but this time we used the first 20 iterates, instead of only 5. For this larger training set, the noise was less relevant and the SR algorithm was able to fully recover the exact logistic map.

These simulations nicely reveal that the transient dynamics for the regular domain contains sufficient information to reconstruct the entire map even from noisy data. Once the map is recovered, it can then be applied to predict also the chaotic domain correctly. For completeness, we have also repeated the simulations for two sets ($r = 3.8$ and $r = 4$) in the chaotic regime, and again a small training set of only four iterates was required to identify the correct map, while noisy data required consistently a larger training set.

### Reconstructing the Set of Lorenz Differential Equations from Noisy Chaotic Data

Finally, we illustrate how SR can be used to re-construct the multiple sets of differential equations from chaotic time series in *several* coupled variables. As an example, we examine here the famous Lorenz equations [34], $dx/dt = \sigma(y - x)$, $dy/dt = x(\rho - z) - y$ and $dz/dt = xy - \beta z$. Using standard numerical solutions techniques (such as Runge–Kutta fourth-order schemes with self-adapting step size), these equations can be solved numerically in the chaotic regime ($\sigma = 10$, $\beta = 8/3$ and $\rho = 28$) for the initial conditions $x(0) = y(0) = z(0) = 10$. Alternatively, one can also perform numerical convergence tests by comparisons with the corresponding solutions obtained by black-box techniques, such as the ones provided by Mathematica. The three time-dependent solutions $\{x(t), y(t), z(t)\}$ and their derivatives $\{dx/dt, dy/dt, dz/dt\}$ were then discretized at times $t_n = n$ (with $n = 0, 1, 2,...,49$) and the resulting six 50-dimensional lists were used as the training input for the SR evolution. We used $N_{\text{pop}} = 40,000$ initial random analytical expressions, the standard binary operations $(+, -, *, /)$, and the probabilities similar to those in Table 1. After only 10 generations SR reproduced correctly each of the three Lorenz equations. This rapid convergence is actually not so surprising as the functional dependence of the three derivatives on $x$, $y$ and $z$ is actually not too complicated, despite having fully chaotic solutions. The differential equations do not contain any special functions, beyond simple products and additions, and therefore are recoverable by SR.

We should remark that while the nonlinear coupling of the three differential equations makes analytical solutions extremely difficult (if not impossible) to obtain, the SR approach is fortunately not affected by this complication. The reason for this advantage is the fact that the SR can construct each differential equation completely independently of the other two, i.e., the differential equations can be determined *sequentially*.

To examine the robustness of the genetic algorithm with regard to inaccuracies, we have repeated the schemes with noised data. The key question of interest is whether these perturbations will maintain the basic structure of the predicted differential equations [with possibly slightly corrected values of the parameters ($\sigma$, $\beta$ and $\rho$)] or whether SR predicts that entirely different functional forms are required for the differential equations, in order to minimize the errors with respect to the given data.

To examine this, we have added to each solution at each moment in time a uniformly distributed random number in the range $[-0.1, 0.1]$. In the time interval $0 < t < 50$, the set of three solutions $x(t_n)$, $y(t_n)$, $z(t_n)$ is highly oscillatory with about 90 local maxima und minima between $\pm 17$ for $x$, $\pm 22$ for $y$ and $10 < z < 43$ for $z(t)$, so therefore these perturbations can be significant on a relative scale. We found that the equations for $x$ and $y$ were predicted to be identical to the original ones, suggesting a remarkable robustness of the SR scheme. However, we note that one can "manipulate" a little bit the final prediction by choosing a relatively large parsimony coefficient (large penalty for complicated expressions), which then favored the original equations, which are simpler in nature.

While reducing a coupled set of differential equations to a single equation is not always possible in general, the set of Lorenz equations happens to be equivalent to just a single autonomous differential equation of third-order for $x(t)$. However, as one might have expected, this new equation for $\mathrm{d}^3x/\mathrm{d}t^3$ is rather complicated and a highly non-linear function of $x$, $\mathrm{d}x/\mathrm{d}t$, $\mathrm{d}^2x/\mathrm{d}t^2$, $\alpha$, $\beta$ and $\rho$ such that the SR algorithm was not able to re-construct its functional form from the data $x(t_n)$ within a reasonable CPU time. This suggests that it might be algorithmically more efficient for the genetic algorithm to construct the easier but coupled sets of differential equations (of lower order) rather than attempting to predict a single equation of a much more complicated structure. This is especially true due to the advantageous sequential nature (mentioned above) of recovering sets of equations.

We should finish here with a technical comment concerning the numerical platform used in our calculations. While the computational realization of genetic algorithms can generally be implemented in several programming languages, we found that an object-oriented language like Python provided us with the largest flexibility. Our results can be easily reproduced on any computational platform. For a small number of generations and small population sizes $N_{\text{pop}}$, most of the simulations can be performed on a simple lab top computer with CPU times of less than a few hours. For sophisticated simulations that can benefit from multi-threading, we had access to twentyfour computing nodes, where each node had two Intel® Xeon Gold 6248R 3 GHz CPUs, with 24 threads each. However, to improve on the most efficient method to implement symbolic regression algorithms into multi-processor and multi-threading environments remains an important challenge for future work.

## Summary and Outlook into Future Challenges

The purpose of this work was two-fold, to accompany the symbolic regression algorithm with a theoretical framework that permits us to obtain some first insight into the formation of optimal solutions under tournament selection. To alleviate the problem with finding the optimum values of free parameters, we have introduced a new constant-mutation operator that can significantly improve the rate of convergence. Using the statistical framework, we have examined the most efficient working conditions for this operator based on an adaptive variance matching. For a simple illustrative example, we have shown that the CMO can decrease the fitness of the optimum search function by 13 orders in magnitude.

Obviously, there are many questions that can be addressed in future studies. For example, the concrete example in "Impact of the adaptive CMO in symbolic regression with mutations and cross-overs" suggested an exponential reduction of the fitness of the best individuum (as well as of the average fitness) of the entire population. It would be interesting to determine how the corresponding rate depends on the many numerical parameters that characterize the evolution. Also, the CMO operation was employed for a (time-dependent) width $B(t)$, which was chosen independent of the particular value the constant takes. Our preliminary studies have shown that if this width is chosen as a (time-dependent) fraction of the absolute value of this constant, the convergence can possibly be accelerated even more. However, more systematic numerical as well as statistical theoretical studies are required to establish the universal efficiency of this particular chosen for the width for the $\sigma$.

In all of our studies, the probabilities for all mutation, cross-over and CMO operations to be applied to the respective tournament winners were held constant in time. As some serve to increase the diversity (in the early exploratory phases) and others serve more to enhance the convergence of a function class (later finetuning or convergence phases), experimenting with time dependent probabilities could be advantageous. The theoretical part of these studies might also employ the statistical framework based on the concepts of fitness densities and proportions, which is outlined in this work.

## Appendix

### (A) $P_m(t)$ for a model system of M classes

To get some first insight into the time scales of the temporal evolution of the proportions $P_m(t)$ of each class under the tournament selection, we present in this appendix an oversimplified system of M classes, where the corresponding initial fitness densities $\rho_m(f, t=0)$ are so narrow that they do not overlap with each other. This permits us to derive a universal iteration scheme, where the proportions $P_m(t)$ can be computed directly from the set of $P_m(t=0)$ without specifying the shape of $\rho_m(f, t=0)$.

In general, the new set of proportions $P_m(t+1)$ after the application of all $N_{\text{pop}}$ tournaments can be obtained via the expression

$$P_m(t+1) = \int_0^\infty \mathrm{d}f P_m(t)\rho_m(f,t)n_T \left[1 - \sum_{m'=1}^{M} P_{m'}(t)\int_0^f \mathrm{d}f'\rho_{m'}(f',t)\right]^{n_T-1} \tag{14}$$

The assumption of non-overlapping densities means that we can assign each class a unique mean fitness value, defined as $\int_0^\infty \mathrm{d}f'\, f'\rho_m(f',t) \equiv f_m$. This permits us to order the class labels such that their associated mean fitness increases with increasing label m, i.e., $f_m < f_{m+1}$. If we further assume that each $\rho_m(f,t)$ is basically non-zero only in the interval $[f_m - \Delta f/2, f_m + \Delta f/2]$, then the integration range of first integral $\int_0^\infty \mathrm{d}f$ of Eq. (14) can be approximated by $\int_{f_m-\Delta f/2}^{f_m+\Delta f/2} \mathrm{d}f$. This means that the largest upper integration value $f$ of the second integral $\int_0^f \mathrm{d}f'$ is at most $f = f_m + \Delta f/2$. As a result, some of the integrals in $S(f,t) = n_T[1 - \sum_{m'=1}^M P_{m'}(t)\int_0^f \mathrm{d}f'\rho_{m'}(f',t)]^{n_T-1}$ can be partially evaluated and therefore simplify significantly. The densities $\rho_{m'}(f',t)$ with a fitness lower than $f_m$ are integrated over their entire extent and we can use $\int_{f_m-\Delta f/2}^{f_m+\Delta f/2} \mathrm{d}f\rho_{m'}(f',t) = 1$. As a result, we obtain $P_{m'}(t)\int_0^f \mathrm{d}f'\rho_{m'}(f',t) = \sum_{m'=1}^{m-1} P_{m'}(t) + \int_0^f \mathrm{d}f'\rho_{m'}(f,t)$. This permits us to represent the entire integrand in $S(f, t)$ as a total derivative and we obtain

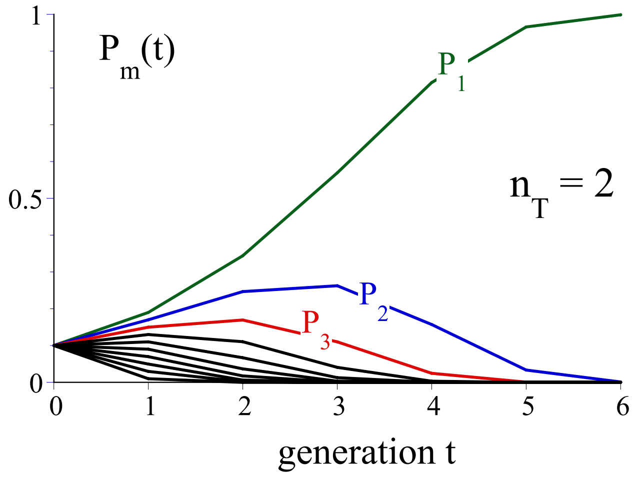

**Fig. 5** Evolution of the proportions $P_m(t)$ for $M = 10$ classes for the first five generations according to the model given by Eq. (16). The initial proportion were chosen $P_m(t=0) = 1/M$. We have only labeled the three proportions with the lowest three fitnesses

$$\begin{aligned}
P_m(t+1) &= \int_{f_m-\Delta f/2}^{f_m-\Delta f/2} \mathrm{d}f\, P_m(t)\rho_m(f,t)n_T \left[1 - \sum_{m'=1}^{M} P_{m'}(t) - \int_0^f \mathrm{d}f' P_m(t)\rho_m(f',t)\right]^{n_T-1} \\
&= \int_{f_m-\Delta f/2}^{f_m+\Delta f/2} \mathrm{d}f(-)\mathrm{d}/\mathrm{d}f \left[1 - \sum_{m'=1}^{M} P_{m'}(t) - \int_0^f \mathrm{d}f' P_m(t)\rho_m(f',t)\right]^{n_T} \\
&= (-)\left[1 - \sum_{m'=1}^{m-1} P_{m'}(t) - \int_0^f \mathrm{d}f' P_m(t)\rho_m(f',t)\right]^{n_T} \Bigg|_{f_m-\Delta f/2}^{f_m+\Delta f/2} \\
&= \left[1 - \sum_{m'=1}^{m-1} P_{m'}(t)\right]^{n_T} - \left[1 - \sum_{m'=1}^{m} P_{m'}(t)\right]^{n_T}
\end{aligned} \tag{15}$$

If we expand this set of equations, we obtain the sequences of mutually coupled iterations

$$\begin{aligned}
P_1(t+1) &= 1 - [1 - P_1(t)]^{n_T} \\
P_2(t+1) &= [1 - P_1(t)]^{n_T} - [1 - P_1(t) - P_2(t)]^{n_T} \\
P_3(t+1) &= [1 - P_1(t) - P_2(t)]^{n_T} - [1 - P_1(t) - P_2(t) - P_3(t)]^{n_T} \\
&\dots \\
P_M(t+1) &= [1 - P_1(t) - P_2(t) - P_3(t) - \dots - P_{M-1}(t)]^{n_T}
\end{aligned} \tag{16}$$

This means that we have obtained an iteration scheme to calculate the class populations $P_m(t+1)$ of the next generation solely from the set of $P_m(t)$, which have a lesser (or equal fitness). One can easily convince oneself that the norm is preserved by this set of maps. i.e., $\sum_{m=1}^{M} P_m(t+1) = \sum_{m=1}^{M} P_m(t) = 1$.

As an interesting side-note, we remark that despite the nonlinear feature of these iterative maps, for the class with the lowest fitness $m = 1$, we have the simpler iterative scheme $P_1(t+1) = 1 - [1 - P_1(t)]^{n_T}$, which converges consistently to $P_1(t \to \infty) \to 1$. If we introduce the complementary proportion $Q_1(t+1) \equiv 1 - P_1(t+1)$, we have $1 - P_1(t+1) = [1 - P_1(t)]^{n_T}$ such that $Q_1(t+1) = Q_1(t)^{n_T}$. This has the solution $Q_1(t) = Q_1(0)^{tn_T}$ such that we have $P_1(t) = 1 - [1 - P_1(0)]^{tn_T}$, so $P_1(t)$ grows monotonically on the time scale proportional to $n_T^{-1}$ and independent of the proportions $P_m$ of the other classes, as one might expect. While the decay is monotonic, its time scale depends not only on $n_T$, but also very sensitively on its initial value $P_1(0)$. If $P_1(0) \ll 1$, then for short times $P_1(t)$ grows linearly in time with a slope proportional to $n_T P_1(0)$, i.e., $P_1(t) = n_T P_1(0) t$.

On the opposite side, if m matches the total number of classes, i.e., m = M, then the iteration scheme for the class with the largest fitness $f_M$ simplifies to

$$P_M(t+1) = \left[1 - \sum_{m'=1}^{M-1} P_{m'}(t)\right]^{n_T} = [1 - \{1 - P_M(t)\}]^{n_T} = P_M(t)^{n_T} \tag{17}$$

This permits us to find the complete time evolution for $t = 1, 2, \dots$, as $P_M(t) = P_M(0)^{tn_T}$ following a universal monotonic exponential decay with decay time proportional to $n_T^{-1}$.

The time evolution of all the other proportions $P_M(t)$ for $m \neq 1$ and $m \neq M$ can be non-momotonic. As an example, in Fig. 5 we show the evolution of $M = 10$ classes and $P_m(t=0) = 1/M$ for the first five generations with a tournament size $n_T = 2$. We see that the low-fitness proportions, $P_m(t)$ (for $m = 1, ..., 5$) increase first and then decay, except $P_1(t)$, which approaches monotonically 1.

**Acknowledgements** We would like to thank Profs. B.K. Clark, X. Fang, Z.L. Li, Y.J. Li and G.H. Rutherford, and G. Jacob, Z. Smozhanyk, and T. Walsh for many helpful discussions and suggestions. This work has been supported by the NSF. C.G. would like to thank ILP for the nice hospitality during his visit to Illinois State University and acknowledges the China Scholarship Council program for his PhD research. We also acknowledge access to the HPC cluster provided by Illinois State University.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

## References

1. Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge: MIT Press; 2016.
2. Sarachik ES, Cane MA. The El-Nino southern oscillation phenomena. Cambridge, UK: Cambridge University Press; 2010.
3. Vladislavleva E, Friedrich T, Neumann F, Wagner M. Predicting the energy output of wind farms based on weather data: important variables and their correlation. Renew Energy. 2013;50:236.
4. Fitzsimmons J, Moscato P. Symbolic regression modelling of drug responses. In: First IEEE Conference on Artificial Intelligence for Industries; 2018.
5. Graham MJ, Djorgovski SG, Mahabal A, Donalek C, Drake A, Longo G. Data challenges of time domain astronomy. Distr Parallel Databases. 2012;30(5):371.
6. Schmidt M, Lipson H. Distilling free-form natural laws from experimental data. Science. 2009;324(5923):81.
7. Udrescu SM, Tegmark M. The Feynman database for symbolic regression. https://space.mit.edu/home/tegmark/aifeynman.html; 2020
8. Udrescu SM, Tegmark M. AI Feynman: a physics-inspired method for symbolic regression. Sci Adv. 2019;6(16):2631.
9. Durasevic M, Domagoj J, Scoczynski Ribeiro Martins M, Stjepan Picek P, Wagner M. Fitness landscape analysis of dimensionally-aware genetic programming featuring Feynman equations. arXiv:2004.12762v1 [cs.NE]; 2020.
10. Quade M, Abel M, Shafi K, Niven RK, Noack BR. Prediction of dynamical systems by symbolic regression. Phys Rev E. 2016;94:012214.
11. Gautier N, Aider JL, Duriez T, Noack B, Segond M, Abel M. Closed-loop separation control using machine learning. J Fluid Mech. 2015;770:442.
12. Qin H. Machine learning and serving of discrete field theories - when artificial intelligence meets the discrete universe. arXiv:1910.10147; 2019.
13. Gong C, Su Q, Grobe R. Machine learning techniques in the examination of the electron-positron pair creation process. J Opt Soc Am B. 2021;38:3582–91.
14. Zimmermann RS, Parlitz U. Observing spatio-temporal dynamics of excitable media using reservoir computing. Chaos. 2018;28:043118.
15. Tanaka G, Yamane T, HšŠroux JB, Nakane R, Kanazawa N, Takeda S, Numata H, Nakano D, Hirose A. Recent advances in physical reservoir computing: a review. Neural Netw. 2019;115:100.
16. Lu Z, Hunt BR, Ott E. Attractor reconstruction by machine learning. Chaos. 2018;28:061104.
17. Symbolic regression is a relatively young research field and there are no extensive reviews for direct applications in physics yet. Two interesting early articles are [17,18].
18. Vladislavleva K. Model-based problem solving through symbolic regression via Pareto genetic programming. PhD thesis, Tilburg University; 2008.
19. Minnebo W, Stijven S. Empowering knowledge computing with variable selection. M Sc thesis: University of Antwerp; 2011.
20. Bruneton JP, Cazenille L, Douin A, Reverdy V. Exploration and exploitation in symbolic regression using quality-diversity and evolutionary strategies algorithms. arXiv:1906.03959v1 [cs.NE]; 2019.
21. Koza JR. Genetic programming: on the programming of computers by means of natural selection. Cambridge: MIT Press; 1992.
22. Koza JR. Genetic programming. Cambridge: MIT Press; 1998.
23. Lambora A, Gupta K, Chopra K. Genetic algorithm—a literature review. In: International conference on machine learning, big data, cloud and parallel computing (COMITCon); 2019, p 380.

24. Miller B, Goldberg D. Genetic algorithms, tournament selection and the effects of noise. Complex Syst. 1995;9:193.
25. Blickle T, Thiele L. A comparison of selection schemes used in evolutionary algorithms. Evol Comput. 1996;4:361.
26. Goldberg D, Deb K. A comparative analysis of selection schemes used in genetic algorithms. Found Genet Algor. 1991;1:69.
27. Holland JH. Adaptation in natural and artificial systems. Cambridge: MIT Press; 1975.
28. Gavrilets S. Fitness landscapes and the origin of species. Princeton: Princeton University Press; 2004.
29. McCandlish DM. Visualizing fitness landscapes. Evolution. 2011;65:1544.
30. Wright S. The roles of mutation, inbreeding, crossbreeding, and selection in evolution. Proc Six Int Congr Genet. 1932;1:355.
31. Richter H, Engelbrecht A. Recent advances in the theory and application of fitness landscapes. Heidelberg: Springer; 2014.
32. May R. Simple mathematical models with very complicated dynamic. Nature. 1976;261:459.
33. Tan JPL. Simulated extrapolated dynamics with parametrization networks. arXiv:1902.03440v1 [nlin.CD]; 2019.
34. Lorenz EN. Deterministic nonperiodic flow. J Atmos Sci. 1963;20(2):130.