Order of Nonlinearity as a Complexity Measure for Models Generated by Symbolic Regression via Pareto Genetic Programming

Ekaterina J. Vladislavleva, *Member, IEEE*, Guido F. Smits, *Member, IEEE*, and Dick den Hertog

***Abstract*—This paper presents a novel approach to generate data-driven regression models that not only give reliable prediction of the observed data but also have smoother response surfaces and extra generalization capabilities with respect to extrapolation. These models are obtained as solutions of a genetic programming (GP) process, where selection is guided by a tradeoff between two competing objectives—numerical accuracy and the order of nonlinearity. The latter is a novel complexity measure that adopts the notion of the minimal degree of the best-fit polynomial, approximating an analytical function with a certain precision. Using nine regression problems, this paper presents and illustrates two different strategies for the use of the order of nonlinearity in symbolic regression via GP. The combination of optimization of the order of nonlinearity together with the numerical accuracy strongly outperforms "conventional" optimization of a size-related expressional complexity and the accuracy with respect to extrapolative capabilities of solutions on all nine test problems. In addition to exploiting the new complexity measure, this paper also introduces a novel heuristic of alternating several optimization objectives in a 2-D optimization framework. Alternating the objectives at each generation in such a way allows us to exploit the effectiveness of 2-D optimization when more than two objectives are of interest (in this paper, these are accuracy, expressional complexity, and the order of nonlinearity). Results of the experiments on all test problems suggest that alternating the order of nonlinearity of GP individuals with their structural complexity produces solutions that are both compact and have smoother response surfaces, and, hence, contributes to better interpretability and understanding.**

***Index Terms*—Complexity, evolutionary multiobjective optimization, extrapolation, genetic programming (GP), industrial data analysis, model selection.**

I. Introduction

FUNDAMENTAL model building is a time- and labor-consuming process in industrial engineering—particularly in polymer research. When time is crucial, empirical estimation of fundamental relationships in process variables is certainly preferable to discovering first-principle models and developing mathematical apparatus to operate with them. Outputs (e.g., system control variables) are usually difficult to monitor, can be measured only in a lab, or require expensive hardware for analysis. The measurements are taken within a limited range of operating conditions and are usually available offline. However, models built upon these measurements are expected to be used to control dynamic processes online.

Because industrial data is always corrupted by noise and is driven by a combination of measured and unmeasured process variables, models should not only accurately predict the observed output but also have some extra generalization capabilities. Examples of such capabilities are insensitivity to a certain amount of noise in the inputs or a capability to extrapolate the output outside the observed region. In addition, generated models should be interpretable, in order to provide additional understanding of the underlying process. The requirements for empirical models in industrial settings are defined as follows (see, also, [1]):

- capability for online reliable prediction of process outputs within the given range of operating conditions and *outside* this range;
- interpretability and the possibility of integrating information from first principles;
- low maintenance and development costs with no (or negligible) operator interference;
- robustness with respect to the variability in process inputs;
- the ability to detect novelties in data to attune itself toward changes in the process.

There is no single technique producing models that fulfil all of the requirements listed above. In the creation of an input–output model, the main emphasis is usually put on the goodness of fit in the given range of inputs. For real-life problems, however, the other requirements are at least equally valuable.

Several empirical modeling techniques are used for constructing input–output regression models including linear regression, nonlinear regression, [2], kriging, [3], radial basis functions, [4], neural networks, [5], support vector machines (SVMs), [6], [7], and genetic programming (GP), [8]. These techniques are used together to complement each other in modeling complex chemical processes. Neural networks, SVMs, and GP have advantages over classical statistical methods in the following cases (see [1]):

- where no or negligible *a priori* information is known about the process and no assumptions on models can be made;
- where modeling problems are multidimensional with either too much or too little data.

Symbolic regression via GP has incontestable advantages over neural networks and SVMs for problems where models

Manuscript received June 05, 2007; revised November 15, 2007, March 10, 2008, and April 21, 2008. First published September 26, 2008; current version published April 01, 2009. This work was supported by Tilburg University and Dow Benelux B.V.

E. J. Vladislavleva and D. den Hertog are with the Department of Econometrics and Operations Research, Tilburg University, Tilburg 5000 LE, The Netherlands (e-mail: katya@vanillamodeling.com; D.denHertog@uvt.nl).

G. F. Smits is with the Core R&D Department, Dow Benelux B.V., Terneuzen 4542 AA, The Netherlands (e-mail; GFSmits@dow.com).

Digital Object Identifier 10.1109/TEVC.2008.926486

have to be simple and interpretable analytic expressions and have a reasonable generalizing behavior [1], [9].

Symbolic regression via GP is a methodology to automatically generate symbolic models that describe functional relationships on given data. Inspired by principles of natural selection, symbolic regression via GP sets up an artificial evolution of models, optimizing some of them, evaluating how well they fit the observed data, and then using this information to decide which models to use as parents for the next generation [8], [10]–[12]. Such evolutionary search continues until a perfect solution is found or the allotted computation time is exceeded. If no *a priori* information is given about the relationship between inputs and outputs, the search space is a set of all possible symbolic models representing valid operations from the fixed set on the given input variables. Classical numerical optimization techniques applied in an empirical modeling framework can be less effective, if they make certain assumptions about the structure of input–output models *a priori* and, hence, limit the search space or make the search biased.

The benefits of symbolic regression via GP include the following.

- There are no prior assumptions on model structure.
- The final predicting model or model ensemble is chosen from a rich set of nonlinear empirical models that is generated automatically.
- Sensitivity analysis of the inputs and variable selection is implicitly performed with no extra cost, which reduces the dimensionality of the problem.
- No assumptions are made on independence of input variables.
- Insight can be provided due to the symbolic representation of models (e.g., in a form of low-order variable transformations that can be used as "natural" meta-variables).

Because the class of potential nonlinear solutions is broad, the main problem is to choose the model or a set of models of optimal complexity [1], [7], [13]. The model selection problem, together with the requirements for a reliable model, lead us to the following questions: How can we measure and control the complexity of models? How can we quickly assess their dissimilarities in interpretability, flexibility to fit the data, and ability to predict the outputs outside the training range?

This paper introduces a new complexity measure reflecting the features of the response surfaces related to symbolic models. The process of model selection under complexity control with this measure, called the order of nonlinearity, improves the extrapolation capabilities of solutions generated by symbolic regression.

In addition to the new complexity measure, a novel heuristic is also introduced, aimed at producing accurate solutions that are genotypically and phenotypically simple. The process of two-objective optimization of model error and expressional complexity alternated at each generation with the two-objective optimization of model error and the order of nonlinearity produces competitive solutions that have low expressional complexity and low orders of nonlinearity.

This paper is organized as follows. Section II briefly describes the concept of symbolic regression via GP and touches on various issues of the complexity control. Section III introduces a new complexity measure reflecting the order of nonlinearity of the evolved models. Section IV describes the results of testing the new measure empirically. Section V concludes by discussing the applicability of the order of nonlinearity in a GP framework.

II. Symbolic Regression via GP

Construction of an unknown function in a high-dimensional space from a finite number of samples bears the risk of overfitting. Care should be also taken to avoid the process noise. Therefore, from models approximating the noisy data, the ones that have minimal (optimal) complexity should be chosen. Okkam's razor principle states: "No more things should be presumed to exist than are absolutely necessary." Following this principle, we should limit and control the complexity of models we create and favor the simplest ones to take part in the evolution. The structural complexity is not the only measure to minimize, in order to produce simple and interpretable models. Also, the extrapolative capabilities of models are important, as is the presence of good local properties, which can be related to first principles.

New models for the next iteration are generated from the current population of models by means of crossover, mutation, and copying. Models gain the right to propagate their good features to successive generations. The current implementation of symbolic regression via GP uses an elite-based selection strategy to select good models and endow them with improved propagation rights. This strategy consists of determination and preservation of a representative set of high quality models, obtained by a given step of evolution [14]. This elite set is stored in an *archive* [1], [14]. It is built based upon the concept of dominance in a space of selected optimization objectives. Propagation rights are granted to all archive members irrespective of their relative numerical goodness of fit. Archive members generate offspring for the next evolutionary step[1]; and these new models are then used to optimize the archive. When the iteration process is terminated, the set of final GP solutions is determined by the archive at the last iteration step. An excellent description of the elite-based strategy for multiobjective evolutionary computation is given in [15] and [16].

The process of updating the archive at each generation employs the concept of Pareto optimality in a set of selected optimization objectives $\Theta = C_1, C_2, \ldots C_k$ (see [14], [15], and Fig. 1). We use the nondominated sorting in the objective space to create an archive of a constant size (see the example in Fig. 2).

The fitness measure used during the evolution is the normalized mean squared error (NMSE). It is bounded by 0 and 1 with a perfect fit corresponding to 1 [see (21)–(23)].

A. Complexity Control

In classical GP applications [8], [10], [11], survival of the fittest is the only criterion to find the optimal model. However, besides numerical closeness to the observed data, there are certainly more characteristics that reflect the quality of the generated models. In applications, the measured data is almost always

[1]In case of crossover, we always choose one parent from the archive, and the other from a population.

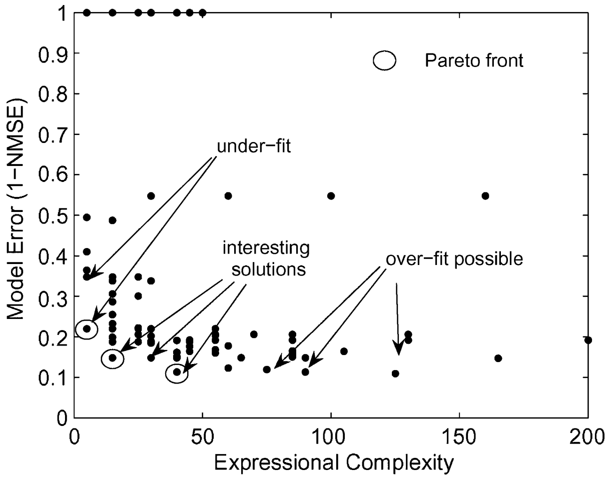

Fig. 1. Pareto front and overfitting. For the shown set of GP-individuals, plotted in the objective space of model expressional complexity versus model error $(1-\text{NMSE})$, the nondominated set consists of only three individuals. Points with low model error and very high expressional complexity are the first suspects of overspecialization. The best-of-run models are almost always at the bottom-right corner of the plot (i.e., are almost always overfitting). Points with too low expressional complexity may be too compact to describe the data, and, hence, have high errors. Note that we want to focus the search on the area around zero (low error and low complexity).

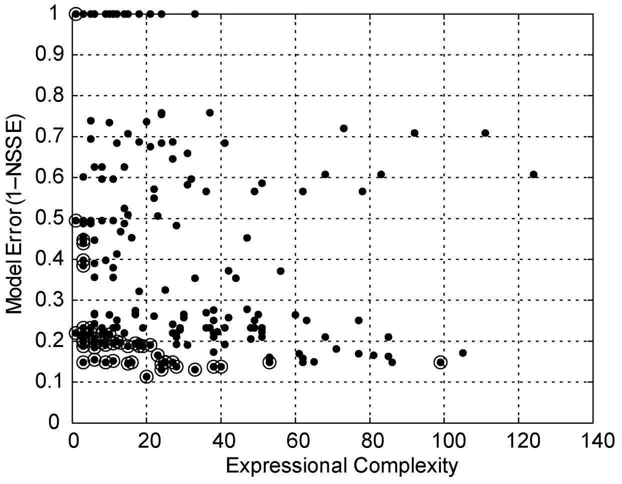

Fig. 2. Archive at the first generation. The figure illustrates the creation of an archive from the initial population. Dots represent 100 population individuals, which are plotted in the objective space of model expressional complexity and model error $(1-\text{NMSE})$ (all subtrees act as individual models to increase the effectiveness of the search). Fifty individual points from the plotted set, which are dominated by the least number of other individuals, are selected to form an archive (depicted as circles). These archive models will be granted the propagation rights to form the next generation.

with noise, so that perfectly accurate models are not the goal. We presume that "the" solution does not exist. Often models with high goodness of fit look so obscure that it becomes infeasible to convince process engineers to implement them for controlling real online processes. In these cases, simpler credible models with a lower level of fitness will be preferred over complex ones. Moreover, limiting the complexity of models may be vital in avoiding overfitting of data and also modeling the process noise. Too complex models are difficult to use, whereas too simple models may give poor prediction. For classical modeling techniques, model complexity is controlled by *a priori* knowledge of the process and the true underlying relationship [7]. In these cases, parsimony pressure is introduced in the fitness function, and a resulting composite fitness function is defined as a linear combination of prediction error and a complexity term (the latter is called a regularization coefficient).

Because penalizing models for high complexity is a natural way to control bloat (i.e., excessive growth in the size of GP individuals without improvements in fitness) [8], [17]–[26], GP researchers have quite extensively analyzed evolutions under parsimony pressure and their relationship to bloat [27]–[29]. The definition of the parsimony pressure as a linear term, added to the fitness function, causes GP to perform well on some problems [20], [27], [30], and less well on others [8], [18]. Soule and Foster showed [28] that the linear coefficients in a composite fitness function, relating numerical fitness and the structural complexity, can be used as a good indicator of the performance of a GP population. However, the search for a good combination of these coefficients requires some intuition and empirical testing.

We firmly believe that when no *a priori* information about the problem is known, the measure for parsimony pressure has to be optimized individually and simultaneously with numerical fitness. This fosters an intelligent tradeoff between model simplicity and model accuracy. Optimizing complexity and accuracy in a truly multiobjective way will exempt us from making risky assumptions about the exact relationship between complexity and accuracy, and will, therefore, not bias the search. With such bi-objective selection, the GP system is pushed to produce both accurate and simple individuals, from which the best ones form a Pareto front—a set of optimal tradeoffs in the 2-D performance space of complexity and accuracy.

For the complexity definition, one can think of two directions in determining the qualitative complexity of the GP model: *complexity of the model expression* (compactness of the genotype) and *behavior of the associated response surface* (smoothness of the phenotype).

Two-dimensional optimization with striving for compact and accurate GP solutions has been shown to systematically outperform the standard single-objective GP on a variety of regression and classification problems [1], [31], [32]. Similar results have been reported for genetic algorithm (GA) in [33] and [34]. This paper aims to explore the feasibility of producing smooth and accurate GP solutions that do not necessarily have "simple" structures, but have "simple" response surfaces, and, hence, generalize better. If producing smooth and accurate solutions proves feasible, this will indicate the possibility of combining the structural and behavioral complexity measures for creating both smooth and simple GP models, without a risk of either bloat or overfitting.

B. Structural Complexity of an Expression

Until now, almost all complexity measures considered by the GP community have addressed the structural complexity of an individual.

An individual is a regression model—a functional relationship between system inputs and outputs built on a limited set of observations (measurements or other models). In our case, it is represented as a valid mathematical expression based on a set of given basic functions, input variables, and random constants sampled from a certain range. Model expressions may vary in

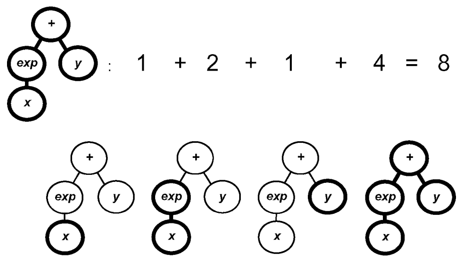

Fig. 3. Expressional complexity of a tree model is the total number of nodes in all subtree models.

size and, in this paper, are represented as tree structures. All subtrees in a tree representation of an expression are considered as separate models during the selection process. Thus, more search space is effectively covered by a population of a fixed number of models.

A set of basic operations contains functions that have one or two arguments. Typical representatives of the set are the standard arithmetical functions: addition, subtraction, multiplication, and division. They may also include power, trigonometric, logarithmic, exponential, and logical functions.

There is a variety of intuitive measures for determining the size of structures operated by a tree-based GP:

1) the number of nodes in a tree;
2) the number of layers in a tree;
3) path length, etc.[2]

We have been using a so-called expressional complexity (see [1]). This measure is determined by the sum of the number of nodes in all subtrees of a given tree. It favors the flatter trees (i.e., trees with fewer layers and, hence, with fewer nested functions) over deep unbalanced trees (in the case of an equal number of nodes). An example of such a case is shown in Fig. 3. The expressional complexity can be interpreted as a size of the model obtained by substituting all inner functions of the model by their function bodies. Keijzer and Foster call this complexity measure a visitation length, show that it is a close relative of the path length, and provide a thorough review of its mathematical properties [36].

The expressional complexity measure is successfully used throughout implementations in MatLab and *Mathematica* (DataModeler add-in). Optimizing the structural complexity of evolving models together with the goodness of fit produces compact solutions that are interpretable and reliable within the training range. After extrapolation, these solutions often demonstrate unwanted behavior, caused by overfitting.

III. Order of Nonlinearity of an Expression

A. Motivation

With respect to the quality of the function determined by the tree expression and the behavior of its response surface, there is a range of complexity measures:

1) the number of variables in the tree representation (the total number of variables present at the leaves is an indicator of model complexity; the number of unique variables reflects the dimensionality of the model);
2) the number of binary and unary functions present at inner nodes;
3) some componentwise nonlinearity of functions present at inner nodes (e.g., addition is less nonlinear, and, hence, simpler than exponentiation), etc.

Our main objective in measuring the nonlinearity of a model is to favor smooth and extrapolative behavior of the response surface and to discourage highly nonlinear behavior (which is unstable towards minor changes in inputs and is dangerous for extrapolation).

It would, moreover, be desirable to find a nonlinearity measure that agrees with an intuitive impression of the complexity of an elementary function. In other words, we want exponentiation to be more complex than taking a square, summation to be simpler than multiplication, and taking a square root and division to be very complex in the neighborhood of zero.

In [37], we introduced a complexity measure that reflects the order of nonlinearity of a model. It is a quantitative measure reflecting the nonlinear growth of the response function determined by a labeled tree. The definition is based on the minimum degree of a polynomial approximating the function on a certain interval with a certain precision.

The reasoning is simple: an obvious measure for the complexity of a multivariate polynomial is its degree. Any tabulated multivariate function can be associated with its unique *best-fit* approximating polynomial. The degree of this polynomial can be considered as a measure for the order of nonlinearity of the response surface of the original function. This order can be seen as a numeric value of the deviation of the response surface from a linear hyperplane.

To make this idea suitable for an evolutionary optimization framework, and in particular, for our implementation of symbolic regression, we had to make several simplifications in its realization.

The best-fit polynomials are difficult to find [38]. Even though we settle for a polynomial giving a *good* approximation of the function, instead of the best-fit polynomial, we still need a procedure that is computationally efficient. Because our goal is to compare the behavior of response surfaces of GP models at each step of the evolution, we strove to:

1) calculate the order of nonlinearity iteratively for a given model, starting from the terminals[3];
2) consider an *efficient* Chebyshev polynomial approximation of a function [39], [40], instead of a labor-intensive search for an optimal best-fit polynomial for a given precision;
3) determine the order of nonlinearity as the degree of the approximating polynomial only for univariate operators, and use another definition of nonlinearity for bivariate functions and compositions.

The first implementation described in [37] used a least squares polynomial approximation for complexity determination. Given a set of points $(x_1, y_1), \dots, (x_n, y_n)$ and a

[2]Iba [35] uses a minimal description length principle, which is not explicitly a complexity measure, but is related to a representation and can be considered as a complexity measure.

[3]Because we consider the subexpressions of a symbolic model as independent models, we want the order of nonlinearity to be easily computable for all subtrees as well as for the parent tree.

maximum order p, $(p < n)$, least squares fitting produces a polynomial $P_{LS} = \sum_{k=1}^{p} a_k x^k$ of degree p that minimizes the error $E_{LS} = \sum_{i=1}^{n} \left(y_i - \sum_k a_k x_i^k\right)^2$ with respect to the coefficients a_k. However, certain conditions for points $x_1, \dots, x_p$ must hold for a least squares polynomial to exist and to be unique. For high degrees, the problem of finding a unique polynomial often becomes ill-defined. For better treatment of steep response surfaces, we therefore made good use of a more stable and reliable framework—Chebyshev approximations (see, also, [41]).

Our current implementation for a given accuracy constructs a Chebyshev polynomial approximation of a function, and takes the degree of the resulting polynomial as a basis for the measure of nonlinearity of a univariate function. An exact definition of the order of nonlinearity appears in Section III-B.

Before giving the definition, we would first like to comment on "approximation of a given accuracy." It is said that $P(x)$ approximates a continuous function $f(x)$ on interval $[a,b]$ with accuracy ϵ, if

$$\max_{x \in [a,b]} |f(x) - P(x)| \leq \epsilon. \tag{1}$$

We change the above definition for error evaluation and consider a finite number of samples $x \in S \subset [a,b]$. The way in which we determine the test set S indeed affects the true quality of the approximation. If S has too few points, then condition (1) is superficial.

If the test set consists of too many points, then error estimation can require excessive computational resources. The choice of S is dictated by a tradeoff between the efficiency of computation and the desired accuracy. In the current implementation, the test set S consists of equidistant points whose number changes dynamically depending on the length of the interval $[a,b]$.

Further on, by a polynomial approximating a univariate function f on interval $[a,b]$ with a certain precision ϵ, we will denote an approximation in a class of Chebyshev polynomials; more precisely, a polynomial $P_f(x) = \sum_{i=0}^{n-1} c_i T_i(x; a, b)$ of a minimal order, such that

$$\max_{x \in S \subset [a,b]} |f(x) - P_f(x)| \leq \epsilon. \tag{2}$$

Here, $T_i(x; a, b) = T_i((2x-(b+a))/(b-a)), i = 1, \dots, n-1$, and $T_i(z)$ is an ith Chebyshev polynomial on $[-1,1]$.

For a univariate function given analytically, the degree of the approximating polynomial depends on the interval, in which the function is being approximated. This implies the necessity of including scaling into the definition of complexity and calculating the ranges for every inner node that corresponds to a univariate operation. In order to be able to treat bivariate functions as univariate (for polynomial approximations), we also need to estimate the ranges of inner nodes, corresponding to bivariate operations.

We would like to emphasize that the function ranges corresponding to inner nodes cannot be determined accurately from the ranges of terminals by using simple interval computations.

In general, the range evaluation of a function defined on a real interval should take into account the monotonicity and extrema of this function and may involve unwanted computation time.

There is a price to pay, however, to avoid these computations. Every subexpression of a symbolic model is explicitly evaluated to obtain the goodness of fit of predicted output to the original output. Therefore, the ranges of subexpressions can be estimated by simply taking the minimum and the maximum of the predicted outputs in the fitness evaluation routine. Such evaluation of ranges of all subfunctions of the given model can indeed introduce some inaccuracy if the extrema of subfunctions are not at the sampling points. Because this is the best we can do without doing extra calculations, it will be good enough for an efficient comparison of the nonlinearity of the GP models.

Once the ranges for all nodes in the tree-based symbolic model are found, then the order of nonlinearity of this model can be computed according to the following inductive definition.

B. Definition

Inductive Definition of the Order of Nonlinearity of a Symbolic Model: Let a tree structure represent a valid analytical model over a set of variables $\mathcal{V} = \{x_1, x_2, \dots, x_{\text{var}}\}$, and a set of constants $\mathcal{C} \subset \mathbf{R}$ with functions from a set $\Phi = \Phi_1 \cup \Phi_2$, where $\Phi_1 = \{\text{sqrt}(x), \ \ln(x), \ \exp(x), \ \exp(-x), \sin(x), \cos(x), x^{\text{const}}, \text{const}^x\}$, $\Phi_2 = \{x+y, x \cdot y, x/y, x^y\}$. Assuming that the precision ϵ is given, the complexity of the tree structure is calculated from the leaves to the root according to the following definition:

A) The complexity of a single node referring to a constant $\text{const} \in \mathcal{C}$ is zero

$$\text{comp}(\text{const}) = 0. \tag{3}$$

B) The complexity of a single node referring to a variable from $x_i \in \mathcal{V}$ is one

$$\text{comp}(x_i) = 1. \tag{4}$$

C) The complexity of an inner node referring to unary function $f \in \Phi_1$ is related to the complexity of the child node referring to a function, variable, or constant g, $g \in \Phi \bigcup \mathcal{V} \bigcup \mathcal{C}$, and the range of the child node $[a,b]$ by the following:

$$\text{comp}(f \circ g) = \text{comp}(g) \cdot n_f \tag{5}$$

where n_f is the minimal degree of P_f, a Chebyshev approximation of function $f(x)$, $x \in [a,b]$ with approximation error (2) at most ϵ.

NB: The complexity of an inner node referring to a unary function $f \in \Phi_1$, $f(x, \text{const}) = \text{const}^x \equiv e^{x \cdot \ln \text{const}}$, is related to the complexity of the child node referring to a function, or to a variable $g \in \Phi \bigcup \mathcal{V}$ and the range of the child node $[a,b]$ by the following:

$$\text{comp}(f \circ g) = \text{comp}(e^{g \ln \text{const}}) = \text{comp}(g) \cdot n_{\text{power}} \tag{6}$$

where n_{power} is the minimal degree of a Chebyshev approximation of function $\exp^{x \ln \text{const}}$, $x \in [a,b]$ with the approximation error (2) at most ϵ.

D) The complexity of an inner node referring to summation and subtraction $\{+,-\} \in \Phi_2$ is related to the complexities of child nodes referring to g_1, g_2 from $\Phi \bigcup \mathcal{V} \bigcup \mathcal{C}$ by the following:

$$\mathrm{comp}(g_1 + g_2) = \max\{\mathrm{comp}(g_1), \mathrm{comp}(g_2)\} \quad (7)$$
$$\mathrm{comp}(g_1 - g_2) = \max\{\mathrm{comp}(g_1), \mathrm{comp}(g_2)\}. \quad (8)$$

E) The complexity of an inner node referring to multiplication $\{*\} \in \Phi_2$ is related to the complexities of child nodes referring to g_1, g_2 from $\Phi \bigcup \mathcal{V} \bigcup \mathcal{C}$ by the following:

$$\mathrm{comp}(g_1 \cdot g_2) = \mathrm{comp}(g_1) + \mathrm{comp}(g_2). \quad (9)$$

F) The complexity of an inner node referring to division $\{/\} \in \Phi_2$ is related to the complexities of child nodes referring to g_1, g_2 from $\Phi \bigcup \mathcal{V} \bigcup \mathcal{C}$ with ranges $[a, b]$ and $[c, d]$ by the following:

$$\mathrm{comp}(g_1/g_2) = \mathrm{comp}(g_1) + \mathrm{comp}(g_2) \cdot n_{\mathrm{div}} \quad (10)$$

where n_{div} is a minimal degree of the Chebyshev approximation of a function $1/x$ on interval $x \in [c, d]$ with the approximation error (2) at most ϵ.

G) The complexity of the root node determines the complexity of the tree structure.

The inductive definition described above is an algorithm to calculate the order of nonlinearity.

In the current implementation, the maximum admissible degree of the Chebyshev approximation is limited to 100. If the precision of the approximation by a 100th-order polynomial still exceeds ϵ, then the complexity value is set to a predefined limit. This limit value is 10 000 for the current implementation. The value for precision ϵ is fixed to 0.0001. The fixed precision used for estimating the degree of Chebyshev approximation will imply higher degrees for wider domain ranges. This penalizes the GP system for constructing solutions with too much diversity in the ranges of the inner nodes.

The number of points at which the approximation error is evaluated is dynamic and depends on the length of the interval of approximation. Currently, we take $\max\{20 \cdot \lceil b - a \rceil, 500\}$ equidistant points on $[a, b]$. This number should vary, depending on the problem difficulty and the descriptiveness of the input data file.

An example of the nonlinearity calculation for a simple two-variable model for $\epsilon = 10^{-6}$ is given in Fig. 4.

C. Discussion

The definition of the order of nonlinearity implies that the complexity of a parent model is never less than the nonlinearity of any of its submodels. This definition allows us to implicitly take the complexity of the representation into account, and make the order of nonlinearity a characteristic of a genotype. Often this causes overestimation of the true order of nonlinearity of the simplified expression. We do this deliberately to push the system towards creating simplified expressions, and to penalize possible precision errors, caused by unnecessary scaling. For example, let trees T_1, T_2, and T_3 represent models x^2/x^2, x/x, and 1 (see Fig. 5). Despite the fact that the models have identical response surfaces, computation of values of T_1 may cause loss of precision. This is why the nonlinearity complexities of $T_1 - T_3$ strictly decrease: $\mathrm{comp}(T_1) = 12$, $\mathrm{comp}(T_2) = 6$, $\mathrm{comp}(T_3) = 0$.

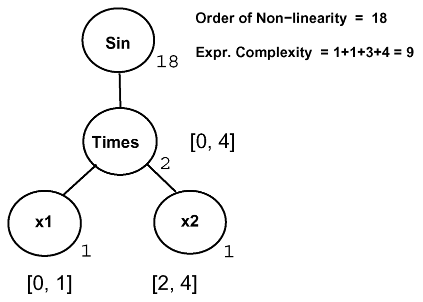

Fig. 4. Example of nonlinearity calculation for a two-variable model. If $x_1 \in [0, 1]$ and $x_2 \in [2, 4]$, then "$x_1 \times x_2$" takes values from the interval $[0, 4]$. Nonlinearities of the terminal nodes are one. The nonlinearity of the $\times$ node is $1 + 1 = 2$. Therefore, the nonlinearity of the Sin node is two times the degree of the Chebyshev approximation of function $\sin x$ on the interval $[0, 4]$. If the chosen approximation accuracy is 10^{-6}, then the order of nonlinearity of the root node is $2 \cdot 9 = 18$. The expressional complexity (visitation length) of the model is 9.

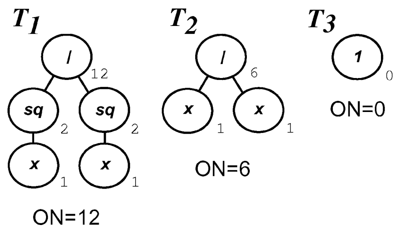

Fig. 5. Genotypic nature of the order of nonlinearity. The order of nonlinearity depends on the representation, because interval arithmetics is taken into account: trees T_1, T_2, and T_3 determine models x^2/x^2, x/x, 1, for $x \in [1, 2]$. Whereas the response surfaces of the models are identical, the orders of nonlinearity are different.

A situation may arise in which, during the calculation of the order of nonlinearity, we encounter a node corresponding to a function with a very small range (e.g., smaller than 10^{-15}). In this case, we do not evaluate the order of nonlinearity by rules A)–F), but assign the complexity of the child node to the node complexity. Although the function with a range less than 10^{-15} could be considered as a constant (in part because the accurate approximation of this function and the error evaluation become questionable due to roundoff errors), we do not make the order of nonlinearity zero, but include the nonlinearity of the child tree into the definition.

Because the order of nonlinearity is determined inductively for every node of the tree starting from the leaves, in most cases, an overestimation of the true minimal order of a polynomial approximating the function with given precision will be produced for unary functions.

For example, consider the function $f(x) = \sin(x)$ on $[0, 4]$. The minimal degree of the Chebyshev approximation of accuracy 10^{-6} is 9 and the order of nonlinearity of the model is also 9 (see Figs. 6 and 7).

Now consider function $f(x) = \sin(\exp(x))$ on $[0, 4]$. The Chebyshev approximation of degree 76 has accuracy 10^{-6}, but

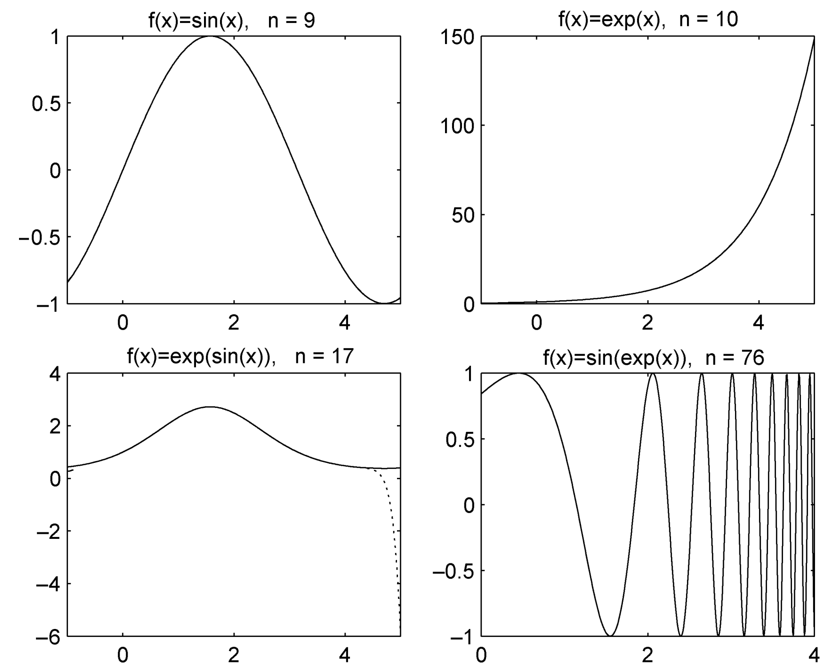

Fig. 6. Chebyshev approximation with precision 10^{-6} of functions $\sin(x)$, $\exp(x)$, $\exp(\sin(x))$, and $\sin(\exp(x))$ on interval $[0, 4]$ and their behavior by extrapolation. The plots represent the functions and their Chebyshev approximations on the interval $[0, 4]$ for 10^{-6} error bound. The degree n of the approximation is given in plot captions. The orders of nonlinearity computed for $x \in [0, 4]$ are shown in Fig. 7.

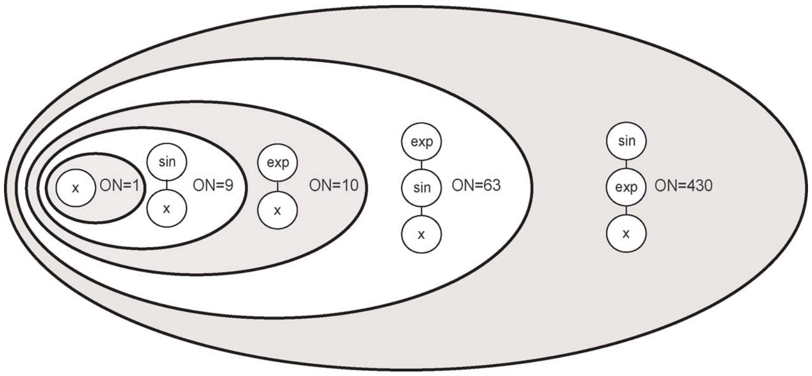

Fig. 7. Modularity over a set of symbolic models. Orders of nonlinearity of tree structures are computed for the domain $x \in [0, 4]$.

the order of nonlinearity of the tree, representing $\sin(\exp(x))$, is 430 (see Figs. 6 and 7).

Such overestimation of the true degree of a polynomial approximating a univariate function with a given precision is deliberate. The order of nonlinearity represents a comparative measure and also builds a modularity (structural separation) in a space of all possible symbolic models over a given set of variables.

Before turning to the empirical analysis of features of the defined complexity measure, we would like to make one more remark. Symbolic regression via GP is used for problems with multiple inputs (up to 1000 or more). The problem of constructing a polynomial approximation in $\mathbf{R}^k$ quickly becomes numerically intensive with $k \gg 2$. This explains why, in our inductive definition, we treat symbolic models as univariate functions. In general, formulas (7), (9), and (10) should not be considered as the rules for finding the true minimal degree of the Chebyshev polynomial.

IV. Numerical Results

A. Test Problems

Real-life applications operate with complex processes, where the true dependency between system inputs and outputs is usually unknown or very complex, and cannot be expressed in one equation. To demonstrate the order of nonlinearity control as a mechanism for preventing overfitting, we selected a suit of synthetic regression problems, which allowed us to generate reliable noise-free test data for inter- and extrapolation. The target equations for chosen problems are given as follows:

$$f_1(x_1, x_2) = \frac{e^{-(x_1-1)^2}}{1.2 + (x_2 - 2.5)^2} \tag{11}$$

$$f_2(x) = e^{-x} x^3 \cos x \sin x (\cos x \sin^2 x - 1) \tag{12}$$

$$f_3(x_1, x_2) = f_2(x_1)(x_2 - 5) \tag{13}$$

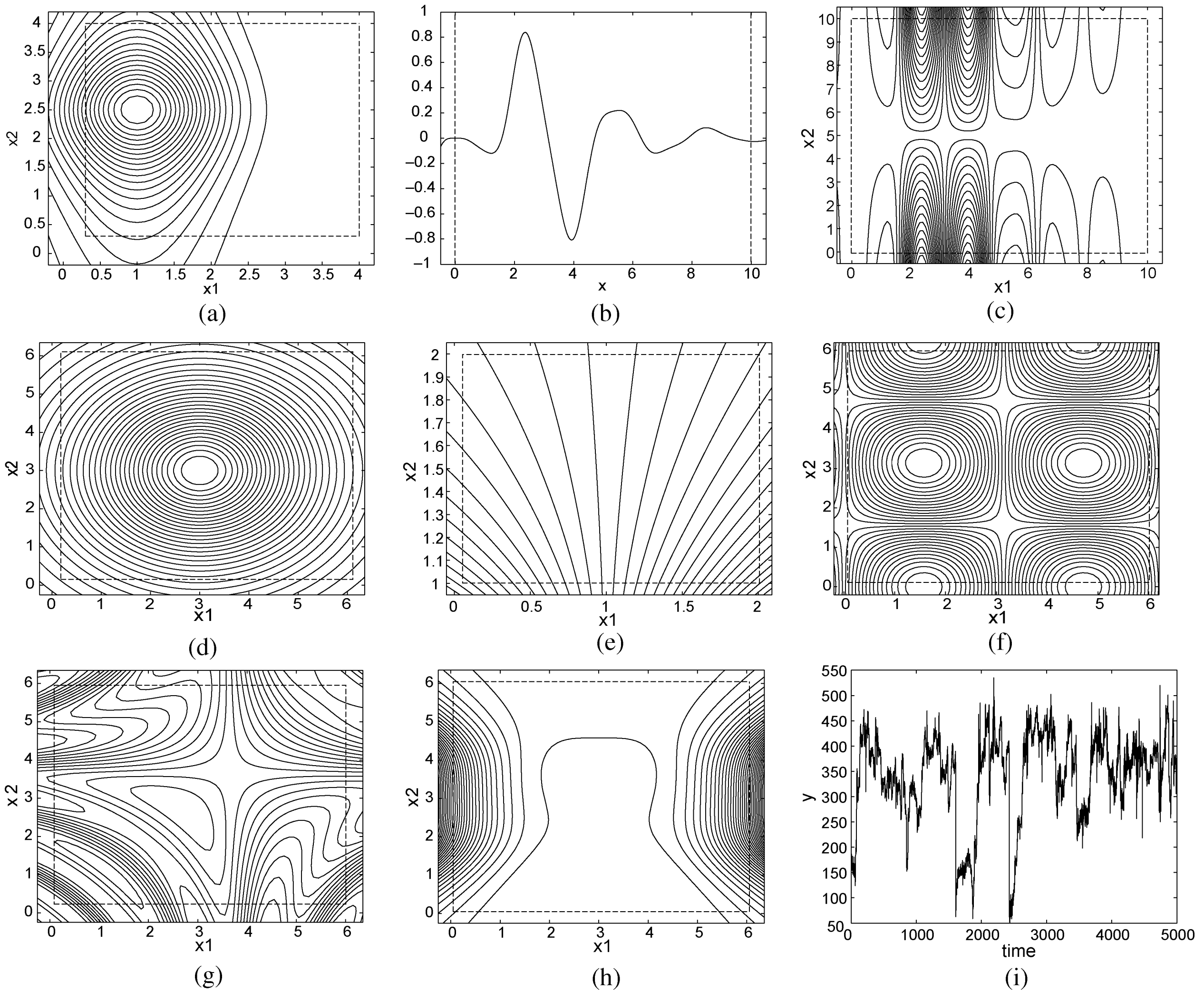

Fig. 8. Contour plots of the target functions (or their projections) in the test regions. The dashed lines represent the boundaries of the training regions. Note that none of the target response surfaces displays pathological behavior in the test region. We expect the same from the GP solutions with good extrapolation properties. (a) Kotanchek; (b) Salustowicz; (c) Salustowicz2D; (d) UBall5D, $x_3 = x_4 = x_5 = 0$; (e) RatPol3D, $x_3 = 2$; (f) SineCosine; (g) Ripple; (h) RatPol2D; and (i) Tower problem: 5000 response values.

$$f_4(x_1,x_2,x_3,x_4,x_5) = \frac{10}{5+\sum_{i=1}^{5}(x_i-3)^2} \tag{14}$$

$$f_5(x_1,x_2,x_3) = 30\frac{(x_1-1)(x_3-1)}{x_2^2(x_1-10)} \tag{15}$$

$$f_6(x_1,x_2) = 6\sin x_1 \cos x_2 \tag{16}$$

$$f_7(x_1,x_2) = (x_1-3)(x_2-3) + 2\sin((x_1-4)(x_2-4)) \tag{17}$$

$$f_8(x_1,x_2) = \frac{(x_1-3)^4+(x_2-3)^3-(x_2-3)}{(x_2-2)^4+10} \tag{18}$$

The choice of several target expressions was inspired by [42], which introduced 13 functions to analyze the performance of scaled GP. We selected the most difficult problems of that set, and still modified most of them to make the regression process more challenging for our ParetoGP system. The first equation defines the Kotanchek function first used in [1]. The second function originates from [43]; we call it the Salustowicz function. The third equation (13) is our 2-D version of the Salustowicz function, which we call Salustowicz2D. The function defined in (14) is our favorite problem. This 5-D equation, which we call the UBall5D[4] function, was inspired by a simpler 2-D problem from [42] and [44]. Our GP systems appears to have most difficulties in discovering the simple and harmonious input–output relationship of the UBall5D function. Target expressions for RatPol3D (15), SineCosine (16), and Ripple (17) problems are adopted from [44], with a linear transformation of variables: $x_i \mapsto x_i - 3$ and a few other modifications. The RatPol2D problem, defined by (18), represents another rational polynomial that is challenging for GP. The contour plots of nine target functions [or projections onto 2-D intervals for functions (14) and (15)] shown in Fig. 8 confirm the high nonlinearity of the underlying response surfaces.

The data for the ninth test problem come from an industrial problem on modeling gas chromatography measurements of the composition of a distillation tower. This Tower problem contains 5000 records and 23 potential input variables. The esti-

[4]Five-dimensional unwrapped ball.

mated parameter is propylene concentration at the top of the distillation tower. The samples are from a gas chromatograph and are taken every 15 min. The 23 potential inputs are temperatures, flows, and pressures related to the distillation tower. The actual sampling rate is 1 min, but 15-min averages of the inputs are used for model development to synchronize with the output measurement. The measurements (5000 for each variable) are not treated as time series, but simply used as samples for a regression model. The propylene concentration needs to modeled as a function of relevant inputs only. The range of the measured propylene concentration is very broad and covers most of the expected operating conditions in the distillation tower.

B. Experimental Setup

To compare the effects of optimizing different complexity measures in symbolic regression via GP, we devised experiments for three optimization schemes:

CASE I: Pareto optimization of the sum of squared errors and expressional complexity;

CASE II: Pareto optimization of the sum of squared errors and the order of nonlinearity;

CASE III: Pareto optimization of the sum of squared errors and expressional complexity, alternated with the Pareto optimization of the sum of squared errors and the order of nonlinearity at every generation.

Note that the optimization of the goodness of fit is present in all cases, because constructing accurate models is our first priority. Comparison of CASE I with CASE II will show that creating accurate and "structurally simple" (i.e., more compact) equations may still lead to highly nonlinear pathological predictions, compared with creating accurate equations of a low or reduced order of nonlinearity.

The presence of CASE III experiments is our attempt to blend optimization of the structural complexity and the nonlinearity with accuracy optimization in a "multiobjective" fashion. We are wary about using a composite objective function, which uses a linear combination of objectives of interest, due to its sensitivity to the particular linear coefficients. Instead of limiting the search by using a composite objective function, one should pursue a true multiobjective search and use a vector of objective functions, whose components are optimized individually and simultaneously. However, the multiobjective approach scales badly when the number of objectives increases. We propose a novel heuristic of overcoming the curse of dimensionality of the objective space by alternating two two-objective optimizations of CASE III.

Because the focus of this paper is nonlinearity control, the formalization and generalization of the approach of CASE III for a general multiobjective optimization problem would overburden the reader. Instead, we illustrate the idea using a particular case, in which one accuracy measure and two complexity measures are defined, where the priority is on the accuracy.

Three objectives need to be minimized during the model development cycle: prediction error of the model's phenotype (Error $=$ $1-$ NMSE), the expressional complexity of the model's genotype (complexity), and the order of nonlinearity of the model's genotype (nonlinearity). The order of importance of these objectives is as follows: error minimization is the primary objective and expressional complexity and nonlinearity are equally important secondary objectives. Taking this into account, the original multiobjective optimization problem can be substituted by a different one, according to the following scheme:

$$\min_{\text{all generations}}(\text{error, complexity, nonlinearity}) \tag{19}$$

$$\Downarrow$$

$$\begin{cases} \min\limits_{\text{odd generations}}(\text{error, complexity}) \\ \min\limits_{\text{even generations}}(\text{error, nonlinearity}). \end{cases} \tag{20}$$

Computational complexity of such substitution drops down from $O(n^3)$ to $O(n^2)$, where n is the total number of models in the population and the archive.[5]

The experiments of CASE III are formulated to combine the best properties of solutions of CASES I and II. Accurate models are thus expected to be both compact and "smooth" (i.e., generalize well), while not producing pathologies in the unseen areas of the input space.

The results of the experiments are compared with respect to the number of pathologies produced on test data with extrapolation, average order of nonlinearity and expressional complexity, and the area percentages under the convex hulls of archives plotted in expressional complexity versus model error and the order of nonlinearity versus model error spaces.

The detailed results of CASES 1–III follow immediately after descriptions of the settings for GP parameters and the choice of training and test data.

C. Data Sampling and GP Settings

As mentioned previously, we focused on the synthetic data in this paper so that we could generate a sufficient amount of reliable, outlier-free test samples in the regions outside the training regions. The details of sampling procedures used for generation of training and test data are given in Table I.

The Tower problem contains real-life data for which the true input–output relationship is unknown. To assess extrapolative capabilities of GP solutions for the Tower problem, we decided to select significant input variables at a preprocessing step, and then used only those to divide the data into training and test sets. The driving variables identified at initial screening using the fitness inheritance approach (see [9]) were x_1, x_4, x_6, x_{12}, and x_{23}. The 5000 data records corresponding to these inputs were scaled into the 5-D cube $[0,1]^5$. All records belonging to the interval $[0.02, 0.98]^5$ were selected into the training set, and the remaining records formed a test set for extrapolation.

In all experiments of this paper, one optimization measure is always the numerical fitness, determined as a NMSE between observed output vector y and the predicted output vector py

$$\text{NMSE}(y, py) = \frac{1 - \text{MSE}(\text{scale}(y), \text{scale}(py))}{1 + \text{MSE}(\text{scale}(y), \text{scale}(py))} \tag{21}$$

$$\text{MSE}(y, py) = \frac{1}{n}\sum_{i=1}^{n}(y_i - py_i)^2 \tag{22}$$

[5]We use the nondominated sorting algorithm to select a fixed number of least dominated models at the Pareto front to update the archive.

TABLE I
SAMPLING STRATEGY FOR TRAINING AND TEST DATA FOR NINE REGRESSION PROBLEMS. THE TABLE REPRESENTS THE SAMPLING STRATEGY, AND THE NUMBER OF POINTS WE USE FOR TRAINING AND TESTING GP SOLUTIONS. NOTATION $x = Rand(a, b)$ MEANS THAT THE x VARIABLE IS SAMPLED RANDOMLY FROM AN INTERVAL $[a, b]$. NOTATION $x_1 = (a_1 : c_1 : b_1)$, $x_2 = (a_2 : c_2 : b_2)$ DETERMINES A UNIFORM MESH WITH STEP LENGTH (c_1, c_2) ON AN INTERVAL $[a_1, b_1] \times [a_2, b_2]$

Problem name	Training Data	Test Data
Kotanchek Eq (11)	100 points x_1, x_2= Rand(0.3,4)	2026 points (x_1, x_2)=(-0.2:0.1:4.2)
Salutowicz Eq (12)	100 points x=(0.05:0.1:10)	221 points x=(-0.5:0.05:10.5)
Salutowicz2D Eq (13)	601 points x_1=(0.05:0.1:10) x_2=(0.05:2:10.05)	2554 points x_1=(-0.5:0.05:10.5) x_2=(-0.5:0.5:10.5)
UBall5D Eq (14)	1024 points x_i=Rand(0.05,6.05)	5000 points x_1=Rand(-0.25,6.35)
RatPol3D Eq (15)	300 points x_1, x_3=Rand(0.05,2) x_2=Rand(1,2)	2701 points x_1, x_3= =(-0.05:0.15:2.1) x_2=(0.95:0.1:2.05)
SineCosine Eq (16)	30 points x_1, x_2=Rand(0.1,5.9)	961 points x_1, x_2= =(-0.05:0.02:6.05)
Ripple Eq (17)	300 points x_1, x_2= =Rand(0.05,6.05)	1000 points x_1, x_3=Rand(-0.25,6.35)
RatPol2D Eq (18)	50 points x_1, x_2=Rand(0.05,6.05)	1157 points x_1, x_2=(-0.25:0.2:6.35)
Tower	3136 points all input values in $[0.02, 0.98]$	1864 points one of the inputs in $[0, 0.02) \bigcup (0.98, 1]$

$$\text{scale}(y) = \frac{y - \min y}{\max y - \min y}. \tag{23}$$

We conducted 50 independent GP runs for each approach and for each problem. All GP settings except for the optimization complexity (expressional, order of nonlinearity, or "both, but alternating") are the same for each test problem. The number of generations is fixed to 250 for all problems except SineCosine and UBall5D problems. These two had to be modeled over 500 generations in order to get an appropriate goodness of fit. Other parameter setting are given in Table II. Point mutation and balanced crossovers were used as genetic operators. When a crossover is performed, the crossover node in the first parent is selected randomly and uniformly. The level from which this node is sampled dictates the selection of a crossover node in the second parent—the levels of the nodes should be the same or similar. According to our experience, and to [45], balanced crossovers allow reduction in the risk of bloat and also overcome the crossover bias in expressional complexity, introduced by the standard uniform crossover (for the latter, see [36]).

TABLE II
GP PARAMETERS FOR NINE TEST PROBLEMS

Number of independent runs	50
Total number of generations	250 and (500 for SinCos and UBall5D)
Population size	100
Archive size	50
Population tournament size	7
Archive tournament size	5
Crossover rate	0.95
Mutation rate	0.05
Rate of mutation on terminals	0.3
Basic Function Set	$+, -, *, /, square$ $x^{real}, x + real, x \cdot real$
Kotanchek, SineCosine, Tower	Basic Set, e^x, e^{-x}
Salustowicz, Salustowicz2D, Ripple	Basic Set, e^x, e^{-x}, $\sin x$, $\cos x$

D. Results and Discussion

The solutions of each independent GP run are stored in an archive that contains 50 expressions. These 50 individuals lie at the Pareto front in "optimization complexity" versus "model fitness" objective space containing all individuals evaluated during the current GP run. For our purposes, all of these individuals are equally valuable GP solutions. The "customer," or the domain expert, will have to choose one of these solutions or, better, an ensemble of solutions that satisfies customer needs; see [46]. We, therefore, combined all archive solutions of independent runs in one ensemble at a postanalysis stage and analyzed the properties of the resulting set of $50 \times 50 = 2500$ solutions across different cases and different test problems.

In addition to analyzing the archive solutions, we studied the features of the best-of-the-run solutions, to conform with the standard GP practice, where the elite-preservation strategy based on the multiobjective model selection is not commonly used.

The detailed results appear in Table III. The second and third columns of this table contain the fraction of equations that showed pathological behavior on the test data. Column two is a percentage of equations producing infinite or undefined root mean squared error (RMSE). The latter is computed from the vectors of predicted values of the model $\mathbf{py}$ and the target values on the test data $\mathbf{ty}$ as

$$\text{RMSE}(\mathbf{ty}, \mathbf{py}) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(ty_i - py_i)^2} \tag{24}$$

where m is the number of test points.

Column three is a percentage of solutions for which the RMSE is infinite, undefined, or excessively large. The threshold for pathologically high error is chosen to be 100. It corresponds to the MSE equal to 10^4. Equations producing these large errors on test data are dangerously erroneous, and a high fraction of them in the set of solutions indicates the tendency to overfitting.

Column four of the table represents the percentage of equations that have the highest allowed order of nonlinearity, equal to 10 000. This value indicates highly nonlinear behavior of a model and the potential to have a pathology on unseen data. We

TABLE III

RESULTS OF THE THREE EXPERIMENTS ON SELECTED TEST PROBLEMS. CASE I CONSISTS OF OPTIMIZATION OF FITNESS AND EXPRESSIONAL COMPLEXITY; CASE II CONSISTS OF OPTIMIZATION OF FITNESS AND THE ORDER OF NONLINEARITY; AND CASE III CONSISTS OF OPTIMIZATION OF THE FITNESS AND ALTERNATED AT EACH GENERATION EXPRESSIONAL COMPLEXITY AND NONLINEARITY. FOR EACH EXPERIMENT, THE QUALITY OF ARCHIVE SOLUTIONS AND THE BEST-OF-RUN SOLUTIONS OVER 50 INDEPENDENT RUNS IS ASSESSED AGAINST TRAINING AND TEST DATA WITH EXTRAPOLATION. CASE II CONSISTENTLY OUTPERFORMS OTHER EXPERIMENTS WITH RESPECT TO THE RATE OF PATHOLOGIES ON TEST DATA OVER ALL TEST PROBLEMS

Problem	All final solutions over all runs (2500)					Best-of-the-run solutions over all runs (50)						
and	Pathologies on Test Data		Complexity			Model Error					Complexity	
Experiment	RMSE $= \infty$, %	RMSE ≥ 100, %	Order of Non-linearity $= 10^4$, %	$< 10^4$, mean	Express. mean	Training Data, RMSE median	IQR	Test Data, RMSE $= \infty$, %	≤ 100, median	IQR	Order of Non-lin. mean	Express. mean
Column Number	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)
Kotanchek												
CASE I	29%	30%	41%	509.1	106.2	0.052	0.02	52%	0.075	0.06	7050	291
CASE II	5%	7%	2%	327.9	243.3	0.055	0.03	18%	0.072	0.06	2476	350
CASE III	10%	10%	6%	383.7	114.9	0.052	0.02	42%	0.069	0.04	5239	319
Salustowicz												
CASE I	34%	36%	86%	12.2	111.9	0.216	0.07	24%	0.233	0.90	10000	298
CASE II	11%	28%	31%	38.9	194.6	0.218	0.08	28%	0.212	0.14	9604	351
CASE III	17%	19%	37%	18.7	132.2	0.212	0.10	34%	0.229	0.50	9804	337
Salustowicz2D												
CASE I	36%	47%	82%	29.0	96.8	0.938	0.20	50%	1.061	0.52	9802	276
CASE II	8%	20%	5%	134.4	226.1	0.832	0.54	24%	0.739	0.69	5095	320
CASE III	18%	23%	23%	227.7	99.7	0.932	0.31	50%	0.817	0.47	9040	293
UBall5D												
CASE I	43%	45%	92%	46.0	141.2	0.173	0.03	66%	0.277	0.86	10000	329
CASE II	16%	18%	4%	33.3	254.5	0.183	0.01	42%	0.637	1.16	4314	377
CASE III	12%	14%	10%	80.6	93.4	0.178	0.02	32%	1.024	1.60	7109	359
RatPol3D												
CASE I	15%	15%	22%	58.7	104.6	0.221	0.01	24%	1.037	0.05	3885	292
CASE II	5%	5%	1%	29.4	262.0	0.218	0.02	8%	1.029	0.03	887	357
CASE III	10%	10%	3%	31.4	122.7	0.218	0.02	22%	1.033	0.04	1838	317
SineCosine												
CASE I	12%	26%	78%	77.0	132.9	1.393	0.33	22%	17.869	28.89	9294	311
CASE II	8%	17%	8%	96.0	348.0	1.328	0.28	12%	3.469	7.58	5288	472
CASE III	12%	19%	21%	119.2	155.0	1.270	0.32	22%	7.599	26.16	9203	449
Ripple												
CASE I	26%	26%	30%	298.2	99.1	1.325	0.21	42%	1.457	1.29	5870	278
CASE II	6%	7%	2%	114.3	250.4	1.311	0.10	10%	1.467	0.52	2037	341
CASE III	7%	8%	6%	220.5	130.0	1.312	0.15	20%	1.486	0.48	3598	308
RatPol2D												
CASE I	52%	55%	42%	108.9	109.3	0.612	0.34	50%	3.881	8.34	6115	283
CASE II	14%	15%	2%	18.3	241.1	0.553	0.20	26%	2.063	1.78	1236	367
CASE III	23%	24%	4%	32.8	117.9	0.663	0.21	38%	3.179	3.34	3263	337
Tower												
CASE I	19%	19%	1%	38.2	84.8	30.3	1.38	42%	40.4	8.4	741	293
CASE II	10%	10%	0%	50.0	257.6	30.1	1.43	24%	40.7	7.8	462	349
CASE III	15%	15%	0%	67.1	112.5	30.1	1.37	32%	42.7	7.5	500	294

expect best-of-the-run solutions to have these high nonlinearity values, due to their inclination to overfitting. Because the rest of the solutions are expected to have lower nonlinearity, small percentages in column four are preferred.

Column five contains the mean order of nonlinearity of those solutions that have the nonlinearity below the threshold of 10 000. Low values in this column for solutions of CASE I for Salustowicz, Salustowicz2D, UBall5D, and SineCosine problems demonstrate the "all or nothing" phenomenon for the orders of nonlinearity of models generated by expressional complexity minimization. For example, for CASE I of the Salustowicz problem, we see that 86% of final solutions have the order of nonlinearity 10 000, and the *remaining* 14% have average nonlinearity equal to only 12.2. We illustrate such a situation for one GP run in Fig. 9.

Column six contains the average expressional complexity of 2500 solutions among independent runs. From columns two to six, we observe that CASE I produces, on average, more compact expressions than CASE II (see column six), although a greater fraction of these have pathologies on test data (see columns two and three). We performed pairwise statistical significance tests for solutions of CASES I–III, and concluded that CASE II outperforms CASE I with respect to the error on test data for all test problems (see Table IV, columns two and three). For significance tests, the solutions with infinite errors (or errors higher than 100) were assigned an error value of 100. ANOVA tests and Wilcoxon–Mann–Whitney rank sum tests were performed to compare the means and the medians of different sets of error values of the equal number of samples. Table IV represents the p-values for the 95% significance level. The mean and the median error on test data for the solutions of CASE II were significantly smaller than those of the solutions of CASE I (the maximum p-value is 0.0004 for the Wilcoxon test on the RatPol3D problem).

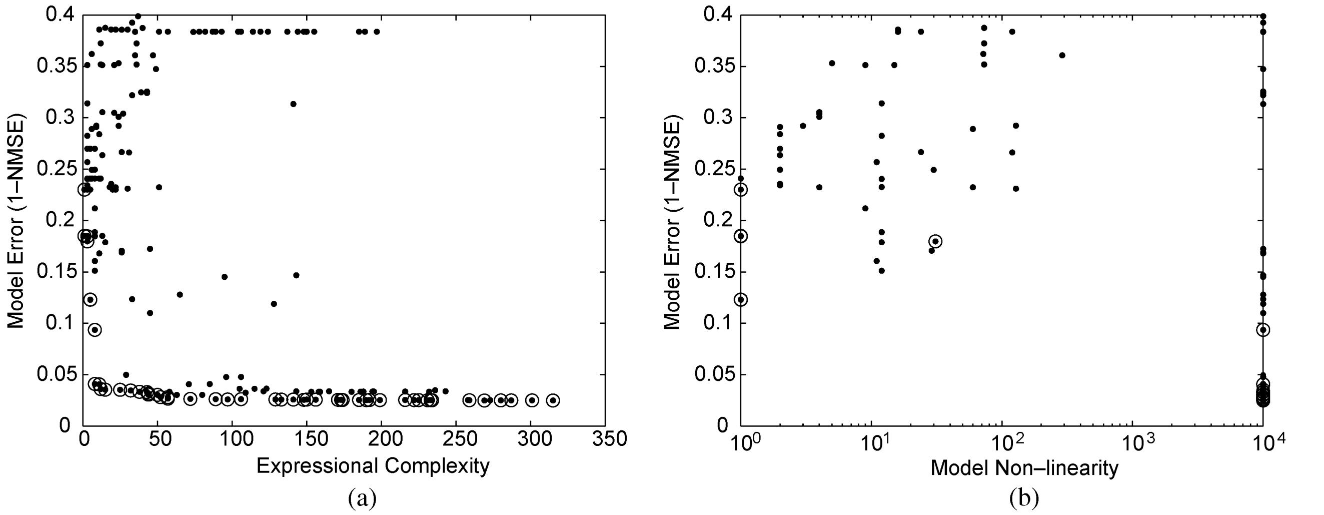

Fig. 9. Solutions of a GP run of CASE I experiment for the Salustowicz2D problem plotted in different objective spaces. In (a), solutions of a CASE I (circled dots) run and their subequations (black dots) are mapped to the original objective space—expressional complexity versus model error. We see a horizontal trend in solutions—most of the equations have similar errors, despite growth in expressional complexity. If we plot the same archive solutions in a different objective space of the order of nonlinearity versus model error (b), we observe a big imbalance in the distribution of the nonlinearity. All solutions with errors below 0.1 (45 out of 50) reach the nonlinearity threshold. Of the five remaining equations, four have the nonlinearity equal to one (two very similar errors, and therefore, appear as one circled dot in the plot); that is, they are estimated as linear in the original inputs. (a) Original objective space; (b) nonlinearity versus model error.

The smoothness of solutions of CASE II comes at the expense of higher expressional complexity. Solutions generated in CASE II consist, for the most part, of "simple" operators (such as addition, subtraction, and multiplication), but are bulky and sometimes difficult to interpret. This excessive growth in structure disappears when the CASE III experiment is used. Comparing the results of CASES II and III in Table III, we observe that the average expressional complexity of solutions can be reduced in CASE III (column six), however, with a side effect of an increased pathology rate (columns two and three). The significance tests show that CASE II significantly outperforms CASE III in the error on the test set on eight out of nine test problems: Kotanchek, UBall5D (only for the mean error), RatPol3D, RatPol2D, Tower, and SineCosine, and Ripple (only for the median error).

In a comparison of CASE III with CASE I, the tests show that the errors on the test data produced by solutions of CASE III are significantly smaller than those of CASE I in all test problems (see columns two and three of Table IV). This brings us to the first important conclusion: solutions obtained in CASE III with alternating expressional complexity and the order of nonlinearity, as well as solutions of CASE II with nonlinearity minimization, are significantly smoother than the ones of CASE I with minimization of expressional complexity. The fact that CASE III produces solutions competitive with CASE I (with respect to the error) and CASE II (with respect to the expressional complexity) is counterintuitive and rather surprising. The alternation of optimization complexities *at each generation* is a very crude heuristic aimed at producing both compact and smooth equations. The fact that it works motivates us to explore the scalability of this approach to cases in which a multitude of objectives needs to be satisfied.

The second part of Table III shows the results for the 50 best-of-the-run solutions for each case. The median and the interquartile range of the RMSE over 50 best-of-the-run solutions per experiment are given in columns seven and eight of Table III. The resulting values look similar for CASES I–III, and no clear trends can be observed with respect to the superiority of any one approach on the training data.

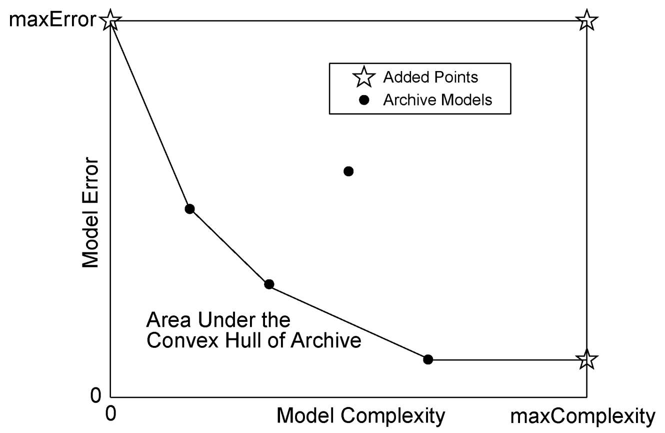

Fig. 10. Area under the convex hull of a GP archive.

The difference among solutions of CASES I–III becomes obvious when the best-of-the-run equations are evaluated on the test sets. Column nine of Table III represents the fraction of best-of-the-run equations that have a pathology on the test data, defined as an infinite RMSE. The trend is similar to the one revealed in columns two and three: CASE I has the highest rate of pathologies at extrapolation. The only exception in this rule is the Salustowicz problem, with 24%, 28%, and 34% pathological equations from the 50 best-of-the-run equations.

Columns 10 and 11 of Table III contain the median error and the interquartile range of a set of errors that are smaller than 100. For example, for the solutions of CASE I of the Kotanchek problem, we observe that 52% of best-of-the-run solutions (26 equations out of 50) have a pathology on the test data. If those

TABLE IV
SIGNIFICANCE OF CONCLUSIONS ABOUT THE RESULTS OF THE THREE EXPERIMENTS. WE PERFORMED ONE-WAY ANOVA TESTS AND WILCOXON–MANN–WHITNEY TESTS FOR ANALYZING DIFFERENCES IN THE MEANS AND THE MEDIANS OF ACCURACY VALUES OF SOLUTIONS OF CASES I–III. THE p-VALUES FOR ANOVA TESTS ARE OBTAINED FROM THE F-STATISTICS AT THE 95% CONFIDENCE LEVEL (ANOVA). THE p-VALUES FOR WILCOXON–MANN–WHITNEY TESTS ARE OBTAINED FROM THE Z-STATISTICS AT THE 95% CONFIDENCE LEVEL AND ARE DOUBLED FOR TWO-SIDED TESTS (WILCOXON). TO OBTAIN EQUAL SAMPLE SIZES, WE TRUNCATED ALL INFINITE AND EXCESSIVELY LARGE VALUES OF RMSE (THOSE WHERE RMSE $\geq=100$) ON THE TEST DATA AT 100

Problem and Hypothesis	Comparing the Error (RMSE) on TEST data of				Comparing the Average Area Percentage	
	All Solutions (2500)		Best-of-the-run Solutions (50)			
	ANOVA, p-value	Wilcoxon, p-value	ANOVA, p-value	Wilcoxon, p-value	ANOVA, p-value	Wilcoxon, p-value
Kotanchek						
CASE II outperforms CASE I	0	0	0.0003	0.0057	0.2924	0.2715
CASE II outperforms CASE III	9.03E-07	0.0000	0.008	0.1936	0.0504	0.0137
CASE III outperforms CASE I	0	6E-33	0.31	0.1962	0.3625	0.2539
Salustowicz						
CASE II outperforms CASE I	0	0	0.3312	0.9579	0	0
CASE III outperforms CASE II	0	0	0.5606	0.8017	0.2696	0.29
CASE III outperforms CASE I	0	0	0.6971	0.8664	0.0000	0.0000
Salustowicz2D						
CASE II outperforms CASE I	0	0	0.0184	0.0006	9E-21	9E-16
CASE II outperforms CASE III	0.0462	0.0055	0.0148	0.0095	0.0361	0.0919
CASE III outperforms CASE I	0	0	0.9295	0.5181	3E-12	4E-11
UBall5D						
CASE II outperforms CASE I	0	0	0.0328	0.1379	1E-14	5E-12
CASE II outperforms CASE III	0	0.8655	0.5058	0.9858	-	-
CASE III outperforms CASE II	-	-	-	-	0.0025	0.0009
CASE III outperforms CASE I	0	0	0.0047	0.0908	1E-12	9E-13
RatPol3D						
CASE II outperforms CASE I	0	0.0004	0.0287	0.0579	-	-
CASE I outperforms CASE II	-	-	-	-	0.0000	0.0000
CASE II outperforms CASE III	0	0	0.0503	0.0824	-	-
CASE III outperforms CASE II	-	-	-	-	3.E-07	1.E-07
CASE III outperforms CASE I	0	0	0.8117	0.8109	0.7237	0.3647
SineCosine						
CASE II outperforms CASE I	0	0	0.0006	0.0000	0.1982	0.1168
CASE II outperforms CASE III	0.1288	0	0.9963	0.2711	0.7107	0.8388
CASE III outperforms CASE I	0	0	0.9847	0.0003	0.0819	0.0682
Ripple						
CASE II outperforms CASE I	0	0	0.0006	0.0109	0.8438	0.39
CASE II outperforms CASE III	0.3902	0	0.1318	0.4248	-	-
CASE III outperforms CASE I	0	0	0.0501	0.0925	-	-
RatPol2D						
CASE II outperforms CASE I	0	0	0.0004	0.0000	0.9015	0.5884
CASE II outperforms CASE III	0	0	0.0305	0.0096	0.5995	0.8985
CASE III outperforms CASE I	0	0	0.1707	0.0728	0.6196	0.9149
Tower						
CASE II outperforms CASE I	0	0	0.08	0.1769	-	-
CASE I outperforms CASE II	-	-	-	-	0.0000	0.0000
CASE II outperforms CASE III	0	0	0.2921	0.1348	-	-
CASE III outperforms CASE II	-	-	-	-	0.0000	0.0000
CASE III outperforms CASE I	0	0	0.5648	0.9661	0.8767	0.7329

and also other equations producing errors higher than 100 are removed from the sample, then the median of the remaining 24 equations will be 0.075, and the interquartile range will be 0.06. This is an argument for using archives of equations and for being very cautious in using best-of-the-run solutions. If only best-of-the-run solutions are sought for, then all runs where the best equation produces a pathology are lost. This corresponds to an incredible waste of 52% *of the spent effort* in the example of CASE I solutions of the Kotanchek problem.

The significance tests for the errors of best-of-the-run equations are performed in a style similar to that used for all solutions. We first assign an error value of 100 to equations producing undefined and infinite values, as well as those exceeding 100. We then perform the pairwise ANOVA and Wilcoxon tests to determine the significance in the difference of the mean and median errors among 50 solutions of CASES I–III. The p-values of the tests are given in columns three and four of Table IV. We can observe that CASE II significantly outperforms CASE I with respect to best-of-the-run errors on test data on seven out of nine problems (for UBall5D only for the mean error).

CASE III significantly outperforms CASE I on best-of-the-run errors only for the mean error on the UBall5D problem. There is no significant difference in the error samples of CASE III and CASE I for the rest of the problems. The second important conclusion that we can make for CASE III is that it is *nowhere* significantly worse than CASE I—even on best-of-the-run solutions.

The average values of the order of nonlinearity and expressional complexity of the best-of-the-run equations are given in columns 12 and 13 of Table III. The conclusions on the complexity of best-of-the-run solutions are the same: 1) CASE I

TABLE V
EXAMPLES OF BEST-OF-THE-RUN SOLUTIONS FOR THE SALUSTOWICZ 2D PROBLEM

Experiment	Best-of-the-run Equations of five independent runs
CASE I	
1)	$-\sin\left[\sin\left[\frac{x_1}{\frac{1.7082+1.00066e^{-x_1}}{(-1+x_1)^2x_1^2-2.369x_2}-5.355e^{x_1}x_2+(2.369+5.355x_2+\sin[\sin[x_2]])^2}\right]\right]$
2)	$\frac{\cos\left[x_2+\left(x_2+\frac{\sin[x_1]}{\left(x_1+\sin\left[x_1+(6.7169+x_2)^2\right]\right)^2}\right)^2\right]}{\left(x_1+\frac{x_1^2}{\left(6.7169+x_1-1.x_1^2\right)^2}+\left(-6.7169-2.x_1+x_1^2\right)^2\right)^2}$
3)	$\left(-8+x_2^{3.5089}\right)\left(3.8823+7.4638x_1^2+\frac{\left(-15.4146+(-7.7073+x_1)x_1+\frac{8}{x_2}+x_2\right)^2}{\cos[64]+\frac{x_1}{-7.7073-\frac{0.0518828}{\frac{x_1}{-7.7073+\frac{1}{x_1^{1.5467}}}+\cos[64]}}}\right)$
4)	$5.33706-2.3856\mathrm{Sec}[4.4562-\cos[8.932x_2]]\sin[3.8411+2.0392x_1+\cos[x_2]]+(-4.4562+\cos[8.932x_2])\times$ $\cdots\times\left(-4.4562+\frac{\sin\left[\frac{\sin[e^{x_2}]}{-3.8411+x_1}\right]}{-3.8411+x_1}\right)$
5)	$\frac{x_1-x_2}{x_1^2-x_2}+x_2\cos\left[\frac{x_1\left(x_1+\frac{x_2}{1-x_1}+x_2\cos\left[\cos\left[\frac{2.4388}{x_1}\right]\right]\right)}{-x_2+\frac{\cos[x_1]}{x_1}}\right]$
CASE II	
1)	$x_1(-x_1^{1.185}+x_1(-8.8+12202.5x_1(-8.8422+(-8.8422+x_1)x_2))+\frac{1}{(4.3648-1.x_1)^2}\times$ $\cdots\times(9.4094-x_1)x_2^2(9.4094+3.3661x_1+x_2)\left(-8.8422+(9.4094-1.x_1)x_1\sin\left[x_1^2\right]\right)^2\sin[x_2])$
2)	$(x_1-x_2)\left(x_1-1.6784\left(-7.7516+x_1(-3.-7.7516x_2)+x_2+x_1^2x_2\right)\cos[7.7516-2x_1]\right)$
3)	$e^{-x_2-\cos[x_1]-\cos[x_2]}\cos[1.7922-2x_1]\cos[5.418-\cos[0.7922x_1]]$
4)	$\left(10.3834-x_1-\sin[x_1]^2\right)$ $\cdots\times\left(-4.1467+4.8212\left(-e^{-\sin[x_1]}+4.8212\left(-4.3834+e^{-2x_1}-e^{-x_2}+x_2\right)+\cos[x_1+x_2]\right)\sin[x_1-\sin[x_1]]\right)$
5)	$x_1(-2.2003-x_1-9.4325(0.466332-9.4325x_1)\times$ $\cdots\times\left(-x_1+(-7.8043+x_1)^2\left(x_1+(-9.9167+x_1)^2(22.7295-4.8807x_2)\right)\right)\cos[7.8043-x_1]\cos[x_1])$
CASE III	
1)	$-\frac{0.123944(-2.8733+x_1)(-4.9692+x_2)}{\left(x_1-0.035185\left(-2.8733+x_1^{1.7365}-\left(-\frac{7.1755}{x_1}+x_1\right)^4\right)\right)^2}$
2)	$\frac{1}{-3.+x_1}(-3.5211+x_2)(1.-1.\cos[3.0688-3.0688x_1])\times$ $\cdots\times\left(-2.+2.\cos[x_1]-1.\cos\left[2+\frac{6.1376}{-2.+x_2}\right]+\cos[0.6542-\cos[3.0688-3.0688x_1]]\right)$
3)	$5.8277(-7.7992+x_1)^2x_1^2(5.8277(-7.7992+x_1)(-7.7992+x_1(-5.2613+x_2))+$ $\cdots+x_1(-5.2613+x_2)(x_1+5.8277(-7.7992+x_1)\sin[7.7992-x_1])^2)\sin[x_1]$
4)	$-4.318-\frac{e^{\sin[\sin[5-x_1]]}(x_1+x_2)\left(x_1+2\left(-5+x_2^2\right)\right)\sin[2x_1]}{x_1x_2}+$ $\cdots+\sin\left[\frac{x_1}{-11.9479+x_2+x_2^2}\right]$
5)	$\frac{\sin[2x_1]}{((x_1+\cos[x_1]+(27.4092x_2+\cos[2x_1-x_2])(1.9861+\sin[x_1]))\left(x_1+(27.4092x_2+\cos[x_1-x_2])^2(7.6219+\sin[x_1])\right))}$

produces more compact expressions at the expense of high orders of nonlinearity; 2) CASE II produces solutions with lower nonlinearity, but higher expressional complexity; 3) CASE III produces lower orders of nonlinearity than CASE I does, and lower expressional complexity than CASE II.

E. Analysis of the Evolved Programs

Note that the differences in the expressional complexity of best-of-the-run equations for CASES I–III are not big. As an example, we give five best-of-the-run solutions for each case on the Salustowicz problem in Table V. All equations in Table V are simplified in Mathematica, so the solutions of CASE II appear to be short (because the linear operations on constants and input variables are already executed). The purpose of Table V is to provide the reader with a visual impression of the differences primarily between CASE I and CASE II solutions and to support our claim that shorter equations may be less convincing for an engineer than longer but less nonlinear equations. Formulas of solutions of CASE I in Table V illustrate that opting for shorter equations generates many nested functions that may make no physical sense.

F. Further Discussion: Areas Under Pareto Fronts

To conclude the analysis of performance of CASES I–III, we compared the average areas under the convex hulls of the archive at the last generation. For two optimization objectives, a good measure to assess the quality of the approach is the area under the Pareto front in the complexity versus error space. The smaller the area, the closer the knee of the Pareto front is to zero. Because CASES I–III exploit different complexity measures, we use the concept of the area under the Pareto front in

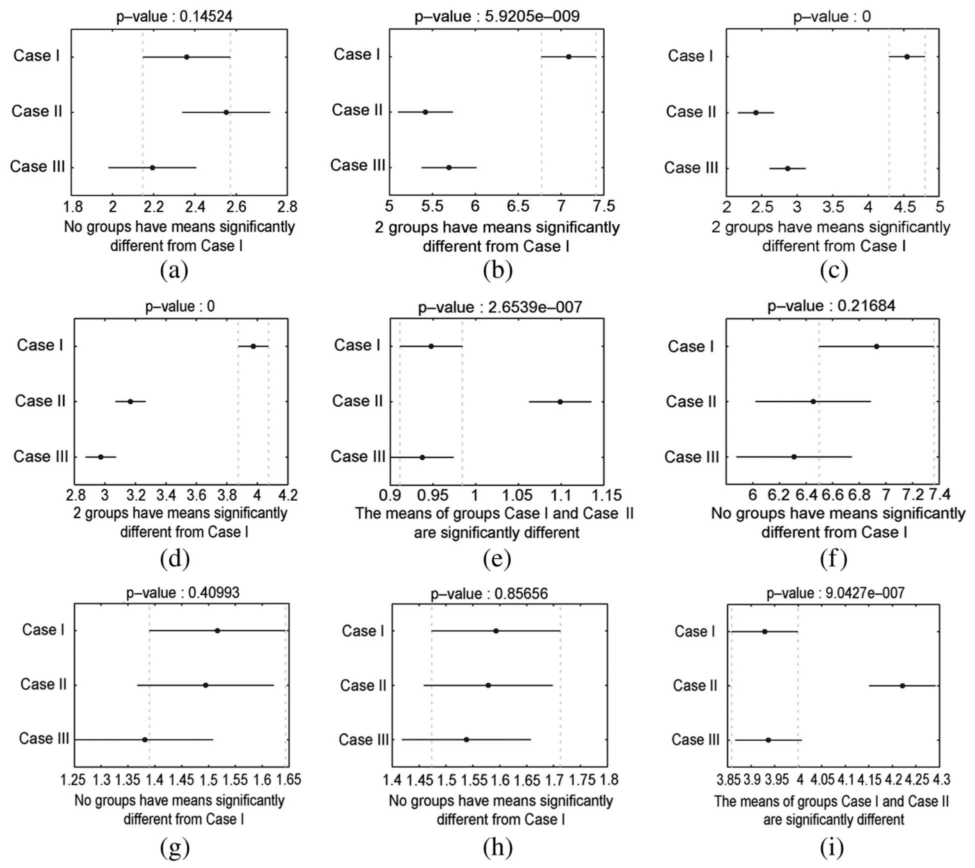

Fig. 11. Multiple comparison tests for average percentage of areas under the convex hulls of archive solutions for nine test problems. Plots represent the 95% confidence intervals of the two-sided tests for multiple comparison of the means of average area percentages over 50 independent runs. Smaller values of the average area percentage are preferred. Groups are significantly different if the confidence intervals do not overlap. The least p-value for the most significantly different groups is given in the plot title. The surprising observation is that CASE I (with expressional complexity minimization) does not outperform CASE III (with alternating complexities) on any test problem. (a) Kotanchek; (b) Salustowicz; (c) Salustowicz2D; (d) UBall5D; (e) RatPol3D; (f) SineCosine; (g) Ripple; (h) RatPol2D; and (i) Tower.

the following definition of the performance characteristic of a GP run.

1) The archive at the last generation is plotted in two objective spaces: expressional complexity versus model error and the order of nonlinearity versus model error.
2) In both objective spaces, the following points are added to the set of solutions: (0, maxError), (maxComplexity, maxError), and (maxComplexity, $\text{minError}_{\text{CurrentArchive}}$). These points are needed to determine the convex hull of the solutions in each objective space (see Fig. 10).
3) An area C of the convex hull of the resulting set of points is computed together with the following percentage:

$$\frac{\text{maxComplexity} \times \text{maxError} - C}{\text{maxComplexity} \times \text{maxError}\ 100\%.}$$

4) The final area percentage for the GP run is defined as an average of the two area percentages under the convex hull of the archive, computed in the expressional complexity versus error and in the order of nonlinearity versus error objective spaces.

We computed the average area percentages of two objective spaces for independent runs of each experiment, and performed the multiple comparison tests for significance differences in the mean values for CASES I–III (ANOVA tests at the 95% confidence level). The results of these comparisons are plotted in Fig. 11.

Columns five and six of Table IV report the p-values for the pairwise comparisons of the average area percentage with the ANOVA tests and the Wilcoxon tests. The conclusion from these statistical tests are as follows: CASE III is statistically better than CASE I on three out of nine problems (Salustowicz, Salustowicz2D, and UBall5D) and is not statistically different from CASE I on the rest of the problems. CASE III is better than CASE II on three problems (UBall5D, RatPol3D, and Tower) and is statistically the same for the rest of the problems. These results make CASE III the winner in comparison with producing the least average percentage area under the convex hull of the archive in two objective spaces.

V. Conclusion

This paper introduces a novel complexity measure for creating smoother individuals in symbolic regression via GP. We suggest computing the order of nonlinearity iteratively for *genotypes* of symbolic models according to a set of rules A)–G). The notion of the new measure is based on a degree of Chebyshev polynomial approximation of a certain accuracy.

We demonstrate the positive effects of controlling the order of nonlinearity on nine nonlinear test problems, and indicate that a similar gain in extrapolative capabilities of symbolic models is obtained on other problems from industrial applications (see [1], [9], and [46]).

One of the weaknesses of the order of nonlinearity is an overestimation of the true minimal degree of Chebyshev approximation of accuracy ϵ for unary functions and the approximate nature of the definition for functions of multiple arguments. Even for functions of two arguments, constructing a Chebyshev approximation is performed in terms of tensor products, and it represents a nontrivial computational procedure. For functions of more variables, it is difficult to construct the Chebyshev polynomial approximation of a given accuracy, thus making it difficult to compare the order of nonlinearity with the degree of such an approximation.

The presented order of nonlinearity applied as a second optimization objective in combination with numerical accuracy to symbolic regression via GP favors models with smoother response surfaces. On all nine test problems, these models show significantly better extrapolative capabilities over models generated with controlled expressional complexity.

We observed that models generated with minimization of the order of nonlinearity (CASE II experiments) are less compact than those generated via optimization of expressional complexity (CASE I experiments). To combine the benefits of creating compact expressions with smoother response surfaces, we have proposed a new hybrid approach to symbolic regression: Pareto optimization of the goodness of fit and expressional complexity, alternated with the Pareto optimization of the goodness of fit and the order of nonlinearity at every generation (CASE III experiments).

The vast majority of models obtained with the order of nonlinearity control (CASES II and III) exhibit "graceful degradation" by extrapolation. This corresponds to an intuitive expectation that smoother approximations mimic the original output longer when extrapolated outside the training range.

Models generated with nonlinearity control do not get singularities when extrapolated over reasonable distances. The fundamental question regarding the way in which to obtain a reliable prediction of the output on an extrapolated domain remains a subject for further research.

Acknowledgment

The authors would like to thank the anonymous reviewers for the extremely valuable comments and enlightening questions that have not only helped them improve the presentation of the results and the convincingness of their experiments, but in general, made their research on this subject more mature.

E. J. Vladislavleva would also like to thank Dr. S. van Eijndhoven from Eindhoven University of Technology for connecting her to Dow Benelux, Dr. A. Kordon from the Dow Chemical Company for the Tower data set, Dr. M. Kotanchek from Evolved Analytics LLC, Dr. R. Pell from the Dow Chemical Company, and Dr. A. Kordon for helpful comments on the earlier versions of this paper.

References

[1] G. Smits and M. Kotanchek, "Pareto-front exploitation in symbolic regression," in *Genetic Programming Theory and Practice II*, U.-M. O'Reilly, T. Yu, R. L. Riolo, and B. Worzel, Eds. Ann Arbor, MI: Springer-Verlag, May 13–15, 2004, ch. 17, pp. 283–299.

[2] , R. A. Johnson and D. W. Wichern, Eds., *Applied Multivariate Statistical Analysis*. Upper Saddle River, NJ: Prentice-Hall, 1988.

[3] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and analysis of computer experiments," *Statist. Sci.*, vol. 4, pp. 409–435, 1989.

[4] M. D. J. Powell, "Radial basis functions for multivariable interpolation: A review," in *Algorithms for Approximation*, ser. Inst. Math. and Its Appl. Conf. Oxford, U.K.: Clarendon Press, 1987, pp. 143–167.

[5] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan, 1994.

[6] V. Vapnik, "The support vector method," in *Proc. 7th Int. Conf. Artif. Neural Netw.*, London, U.K., 1997, pp. 263–271.

[7] V. Cherkassky and F. Mulier, *Learning From Data: Concepts, Theory, and Methods*. New York: Wiley, 1998.

[8] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press, 1992.

[9] G. Smits, A. Kordon, K. Vladislavleva, E. Jordaan, and M. Kotanchek, "Variable selection in industrial datasets using Pareto genetic programming," in *Genetic Programming Theory and Practice III*, T. Yu, R. L. Riolo, and B. Worzel, Eds. Ann Arbor, MI: Kluwer, May 12–14, 2005.

[10] W. B. Langdon and R. Poli, *Foundations of Genetic Programming*. Heidelberg, Germany: Springer-Verlag, 2002.

[11] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone, *Genetic Programming—An Introduction; On the Automatic Evolution of Computer Programs and Its Applications*. San Francisco, CA: Morgan Kaufmann, 1998.

[12] J. R. Koza, *Genetic Programming II: Automatic Discovery of Reusable Programs*. Cambridge, MA: MIT Press, May 1994.

[13] V. Cherkassky, "Model complexity control and statistical learning theory," *Natural Comput.: Int. J.*, vol. 1, no. 1, pp. 109–133, 2002.

[14] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evolut. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

[15] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolut. Comput.*, vol. 3, no. 1, pp. 1–16, 1995.

[16] M. Laumanns, L. Thiele, E. Zitzler, and K. Deb, "Archiving with guaranteed convergence and diversity in multi-objective optimization," in *Proc. Genetic Evolut. Comput. Conf.*, W. B. Langdon, E. Cantú-Paz, K. E. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. K. Burke, and N. Jonoska, Eds., New York, 2002, pp. 439–447.

[17] T. Blickle and L. Thiele, "Genetic programming and redundancy," in *Proc. Genetic Algorithms Within the Framework of Evolut. Comput.*, J. Hopf, Ed., Saarbrücken, Germany, 1994, pp. 33–38, Max-Planck-Institut für Informatik (MPI-I-94-241).

[18] P. Nordin and W. Banzhaf, "Complexity compression and evolution," in *Proc. 6th Int. Conf. Genetic Algorithms*, L. Eshelman, Ed., Pittsburgh, PA, Jul. 15–19, 1995, pp. 310–317.

[19] N. F. McPhee and J. D. Miller, "Accurate replication in genetic programming," in *Proc. 6th Int. Conf. Genetic Algorithms*, L. Eshelman, Ed., Pittsburgh, PA, Jul. 15–19, 1995, pp. 303–309, Morgan Kaufmann.

[20] T. Soule, J. A. Foster, and J. Dickinson, "Code growth in genetic programming," in *Proc. 1st Annu. Conf. Genetic Programm.*, J. R. Koza, D. E. Goldberg, D. B. Fogel, and R. L. Riolo, Eds., Jul. 28–31, 1996, pp. 215–223.

[21] W. B. Langdon and R. Poli, "Fitness causes bloat: Mutation," in *Proc. 1st Eur. Workshop Genetic Programm.*, W. Banzhaf, R. Poli, M. Schoenauer, and T. C. Fogarty, Eds., Paris, France, Apr. 14–15, 1998, vol. 1391, pp. 37–48.

[22] W. B. Langdon, T. Soule, R. Poli, and J. A. Foster, "The evolution of size and shape," in *Advances in Genetic Programming 3*, L. Spector, W. B. Langdon, U.-M. O'Reilly, and P. J. Angeline, Eds. Cambridge, MA: MIT Press, Jun. 1999, ch. 8, pp. 163–190.

[23] W. Banzhaf and W. B. Langdon, "Some considerations on the reason for bloat," *Genetic Programm. Evolvable Mach.*, vol. 3, no. 1, pp. 81–91, Mar. 2002.

[24] M. J. Streeter, "The root causes of code growth in genetic programming," in *Proc. Eur. Conf. Genetic Programm.*, C. Ryan, T. Soule, M. Keijzer, E. Tsang, R. Poli, and E. Costa, Eds., Essex, U.K., Apr. 14–16, 2003, vol. 2610, pp. 443–454.

[25] S. Gustafson, A. Ekart, E. Burke, and G. Kendall, "Problem difficulty and code growth in genetic programming," *Genetic Programm. Evolvable Mach.*, vol. 5, no. 3, pp. 271–290, Sep. 2004.

[26] R. Poli, W. B. Langdon, and S. Dignum, "On the limiting distribution of program sizes in tree-based genetic programming," in *Proc. 10th Eur. Conf. Genetic Programm.*, M. Ebner, M. O'Neill, A. Ekárt, L. Vanneschi, and A. I. Esparcia-Alcázar, Eds., Valencia, Spain, Apr. 11–13, 2007, vol. 4445, pp. 193–204.

[27] B.-T. Zhang and H. Mühlenbein, "Balancing accuracy and parsimony in genetic programming," *Evolut. Comput.*, vol. 3, no. 1, pp. 17–38, 1995.

[28] T. Soule and J. A. Foster, "Effects of code growth and parsimony pressure on populations in genetic programming," *Evolut. Comput.*, vol. 6, no. 4, pp. 293–309, Winter, 1998.

[29] T. Soule and R. B. Heckendorn, "An analysis of the causes of code growth in genetic programming," *Genetic Programm. Evolvable Mach.*, vol. 3, no. 3, pp. 283–309, Sep. 2002.

[30] T. Blickle, "Evolving compact solutions in genetic programming: A case study," in *Proc. Int. Conf. Evolut. Comput.*, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds., Berlin, Germany, Sept. 22–26, 1996, vol. 1141, pp. 564–573, Parallel Problem Solving From Nature IV.

[31] S. Bleuler, M. Brack, L. Thiele, and E. Zitzler, "Multiobjective genetic programming: Reducing bloat using SPEA2," in *Proc. Congr. Evolut. Comput.*, May 27–30, 2001, pp. 536–543.

[32] X. Llorà, D. E. Goldberg, I. Traus, and E. B. i Mansilla, "Accuracy, parsimony, and generality in evolutionary learning systems via multiobjective selection," in *Proc. Int. Workshop Learn. Classifier Syst.*, Granada, Spain, 2002, pp. 118–142.

[33] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolut. Comput.*, vol. 2, no. 3, pp. 221–248, 1994.

[34] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evolut. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[35] H. Iba, H. de Garis, and T. Sato, "Genetic programming using a minimum description length principle," in *Advances in Genetic Programming*, K. E. Kinnear, Jr., Ed. Cambridge, MA: MIT Press, 1994, ch. 12, pp. 265–284.

[36] M. Keijzer and J. Foster, "Crossover bias in genetic programming," in *Proc. 10th Eur. Conf. Genetic Programm.*, M. Ebner, M. O'Neill, A. Ekárt, L. Vanneschi, and A. I. Esparcia-Alcázar, Eds., Valencia, Spain, Apr. 11–13, 2007, vol. 4445, pp. 33–43.

[37] N. Garshina and C. Vladislavleva, "On development of a complexity measure for symbolic regression via genetic programming," in *Modeling Report for Dow Benelux B.V.* Eindhoven, The Netherlands: Technische Universiteit Eindhoven, 2004.

[38] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*. New York: Cambridge Univ. Press, 1992.

[39] T. J. Rivlin, *The Chebyshev Polynomials*. New York: Wiley, 1974.

[40] P. Tchebycheff, "Sur les questions de minima qui se rattachent à la représentation approximative des fonctions," *Mémoires Acad. Sci. St. Pétersbourg par divers savants*, vol. 7, p. 191, 1857.

[41] E. J. Vladislavleva, "Symbolic regression via genetic programming," in *Final Thesis for Dow Benelux B.V.* Eindhoven, The Netherlands: Technische Universiteit Eindhoven, 2005.

[42] M. Keijzer, "Improving symbolic regression with interval arithmetic and linear scaling," in *Proc. Eur. Conf. Genetic Programm.* , C. Ryan, T. Soule, M. Keijzer, E. Tsang, R. Poli, and E. Costa, Eds., Essex, UK, 2003, vol. 2610, pp. 70–82.

[43] R. Salustowicz and J. Schmidhuber, "Probabilistic incremental program evolution," *Evolut. Comput.*, vol. 5, no. 2, pp. 123–141, 1997.

[44] A. Topchy and W. F. Punch, "Faster genetic programming based on local gradient search of numeric leaf values," in *Proc. Genetic Evolut. Comput. Conf.*, L. Spector, E. D. Goodman, A. Wu, W. B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke, Eds., 2001, pp. 155–162.

[45] H. Xie, M. Zhang, and P. Andreae, "An analysis of depth of crossover points in tree-based genetic programming," in *Proc. IEEE Congr. Evolut. Comput.*, D. Srinivasan and L. Wang, Eds., Sept. 25–28, 2007, pp. 4561–4568.

[46] M. Kotanchek, E. Vladislavleva, and G. Smits, "Trustable symbolic regression models: Using ensembles, interval arithmetic and pareto fronts to develop robust and trust-aware models," in *Genetic Programming Theory and Practice V*, ser. Genetic and Evolutionary Computation, R. L. Riolo, T. Soule, and B. Worzel, Eds. Ann Arbor, MI: Springer-Verlag, May 11–13, 2007, vol. 6, ch. 12, pp. 203–222.

Ekaterina J. Vladislavleva (S'05–M'08) graduated in mathematical theory of intelligent systems from the Faculty of Mathematics and Mechanics, Moscow State University of Lomonossov, Russia, in 2000. She also holds a degree of Professional Doctorate in Engineering from Eindhoven University of Technology, Eindhoven, The Netherlands. Currently, she is working towards the Ph.D. degree at the Department of Econometrics and Operations Research, Tilburg University, The Netherlands, which she will receive in 2008, spending half of her research time at the Core R&D Department of Dow Benelux B.V.

Her current research interests are in industrial data analysis and data-driven modeling through evolutionary computation, particularly in the industrial scale symbolic regression via genetic programming. After receiving the Ph.D. degree, she will work as a Postdoctoral Fellow on synergetic coapplication of symbolic and numeric data-driven modeling in the Group of Computer Arithmetic and Numerical Techniques at the Department of Mathematics and Computer Science, Antwerp University, Belgium.

Guido F. Smits (M'98) received the M.S. degree in organic chemistry from University of Antwerp, Belgium, in 1980, the Ph.D. degree in mathematics and physical sciences from the University of Leiden, The Netherlands, in 1985, and the postmasters degrees in informatics from University of Leuven, Belgium, in 1986, in polymer physics and chemistry from University of Eindhoven, Netherlands, in 1987, and in knowledge technology from University of Ghent, Belgium, in 1989.

He is a Research Leader in the Modeling Group within the Engineering and Process Sciences Department, The Dow Chemical Company. His main area of interest and expertise is in industrial applications of computational intelligence techniques such as neural nets, genetic algorithms, genetic programming, support vector machines, and product design in general. He published more than 60 technical papers and currently holds 15 patents.

Dick den Hertog received the M.S. degree (*cum laude*) in applied mathematics and the Ph.D. degree (*cum laude*) in operations research from Delft University, Delft, The Netherlands, in 1989 and 1992, respectively.

He is a Professor of Operations Research at Tilburg University. His research interests cover various fields in linear and nonlinear optimization, e.g., simulation-based optimization and robust optimization. He is also active in applying the theory in real-life applications. He is Scientific Director of CentER and Vice-Dean of Research of the Faculty of Economics and Business Administration. From 1992 to 1999, he was a Consultant for optimization at CQM in Eindhoven.