Shape Constraints in Symbolic Regression using Penalized Least Squares*

Viktor Martinek[1][0000−0001−6215−4783], Julia Reuter[2][0000−0002−7023−7965], Ophelia Frotscher[3][0000−0002−6915−1988], Sanaz Mostaghim[2][0000−0002−9917−5227], Markus Richter[3][0000−0001−8120−5646], and Roland Herzog[1][0000−0003−2164−6575]

[1] Interdisciplinary Center for Scientific Computing, Heidelberg University, 69120 Heidelberg, Germany viktor.martinek@iwr.uni-heidelberg.de, roland.herzog@iwr.uni-heidelberg.de

[2] Otto von Guericke University, Faculty of Computer Science, Chair of Computational Intelligence, 39106 Magdeburg, Germany julia.reuter@ovgu.de, sanaz.mostaghim@ovgu.de

[3] Faculty of Mechanical Engineering, University of Technology Chemnitz, Applied Thermodynamics, 09107 Chemnitz, Germany
ophelia.frotscher@mb.tu-chemnitz.de, m.richter@mb.tu-chemnitz.de

Abstract. We study the addition of shape constraints (SC) and their consideration during the parameter identification step of symbolic regression (SR). SC serve as a means to introduce prior knowledge about the shape of the otherwise unknown model function into SR. Unlike previous works that have explored SC in SR, we propose minimizing SC violations during parameter identification using gradient-based numerical optimization. We test three algorithm variants to evaluate their performance in identifying three symbolic expressions from synthetically generated data sets. This paper examines two benchmark scenarios: one with varying noise levels and another with reduced amounts of training data. The results indicate that incorporating SC into the expression search is particularly beneficial when data is scarce. Compared to using SC only in the selection process, our approach of minimizing violations during parameter identification shows a statistically significant benefit in some of our test cases, without being significantly worse in any instance.

Keywords: Symbolic regression · Shape constraints · Constrained least squares · Thermodynamics · Equations of state

1 Introduction

Symbolic regression (SR) is a supervised machine learning approach to discover underlying mathematical expressions from data. The search space of potential expressions is very large, and the problem is NP-hard, as proven by [21]. However, there are ways to incorporate prior knowledge beyond the data into the

* This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – HE 6077/14-1 and RI 2482/10-1 – within the Priority Programme "SPP 2331: Machine Learning in Chemical Engineering".

expression search to guide the search more efficiently. One way to do this is by using shape constraints (SC).

SC impose restrictions on the behavior (or shape) of a function. A relatively common example of a SC is monotonicity with respect to some variable. SC can be used to incorporate domain-specific knowledge or enforce desired behavior, which can be advantageous in many scenarios. For example, SC can help reduce the required number of data, or help bridge regions where no data is available. In addition, SC can mitigate the effect of noisy data, including data with outliers. Depending on how SC are implemented into the expression search, they can either provide additional guidance or even reduce the search space. These benefits are already shown in several studies, including [10,5,6,7,2,11,17,18,13]. Altogether, the expressions obtained have greater utility and reliability when they obey the domain-specific SC.

Our proposed approach differs from previous work in two points. First, we take into account the amount of SC violations already during the parameter identification step. To the best of our knowledge, this is only done in one other publication (see [11]). Second, this parameter identification step is carried out using second-order optimization and algorithmic differentiation. This allows us to find a set of parameters such that the resulting model conforms to the SC, while also fitting the data. As in other works, the SC violations are implemented as additional objectives which are considered in the multi-objective selection process of the genetic algorithm.

In the following section, we introduce the problem formulation of SR and extend it to include SC. We describe our approach to the parameter identification, which also minimizes the SC violations. We then explain how we incorporate the SC into the evolutionary algorithm. In Section 3, related works are briefly described and the differences to the present work are outlined. In Section 4, the experiments are detailed, and their results presented. Section 5 offers a summary and discussion of the results. Finally, we draw a conclusion in Section 6.

2 Shape Constraints in Symbolic Regression

In SR, the goal is to find a mathematical expression m, along with values for its parameters p, such that we

$$\text{Minimize} \quad \frac{1}{N}\sum_{i=1}^{N} f\left(\frac{y_i - m(X_i, p)}{y_i}\right) \quad \text{w.r.t. } (m, p). \tag{2.1}$$

The expression m comes from an admissible class of expressions ensuring its syntactic correctness. The parameter p is from a parameter space that generally depends on m. Above, the vector-valued X and the scalar-valued y are the independent and dependent variables of the model, and (X_i, y_i) denote the pairs of data in the data set, $i = 1, \dots, N$. The function f is known as loss function and typical examples include $f(r) = r^2$, where r may be the residual resulting from $y_i - m(X_i, p)$. This example leads to (2.1) representing the mean squared relative error.

Currently, genetic programming algorithms perform best among wide-ranging SR tasks (see [12]). Summarized briefly, the population-based genetic programming algorithms combine candidate expressions (crossover), and insert slight variations into candidate expressions (mutation). In this way, new candidate expressions are generated. Better candidate models replace less favorable ones in the population, which is known as the selection process.

As shorter and intuitively simpler expressions are preferred, problem (2.1) is augmented by an additional objective measuring the complexity, e. g., length, of the expression m in most SR approaches. Many approaches consider additional objectives, such as age (see [20]), or the coefficient of determination R^2.

In any case, (2.1) becomes a multi-objective problem. Depending on the algorithm, these competing objectives are either addressed through the concepts of Pareto dominance, or aggregated to a yield singular fitness value by means of a weighted sum.

In this work, we use thermodynamics-informed symbolic regression (TiSR) (see [14]), an SR implementation based on NSGA-II (see [4]). In contrast to NSGA-II, TiSR uses both Pareto dominance and fitness-based selection criteria to ensure sufficient diversity in the population. TiSR is available at https://github.com/scoop-group/TiSR.

SC are expressed by augmenting problem (2.1) using inequality and/or equality constraints. The problem then becomes

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{N}\sum_{i=1}^{N} f\left(\frac{y_i - m(X_i, p)}{y_i}\right) \quad \text{w.r.t. } (m, p) \\ \text{subject to} \quad & g(m(X, p)) \le 0 \quad \text{for all } X \in \Omega, \\ \text{and} \quad & h(m(X, p)) = 0 \quad \text{for all } X \in \Omega. \end{aligned} \tag{2.2}$$

The set Ω defines the range for the independent variable X on which the SC is relevant. Therefore, in general, there are infinitely many SC, that can be taken into account in two different ways.

In pessimistic approaches, interval arithmetic is used to compute an enclosure of the range of the SC function's values over $X \in \Omega$. This offers algorithms the possibility to certify a model's feasibility w.r.t. all SC. However, such approaches may deem feasible models infeasible, as observed and discussed by Haider et al. (see [6]). By contrast, optimistic approaches evaluate and enforce the SC only in a finite number of predetermined locations. While this may lead to models incorrectly classified as feasible, the evaluation of SC as well as their derivatives w.r.t. p is straightforward.

In this study, we employ an optimistic approach. The SC evaluation points $X^1, \dots, X^C$ are distinct from the data points $X_1, \dots, X_N$. In order to accommodate different types of SC, the SC functions g and h can be vector-valued. This allows several SC to be considered simultaneously, e. g., monotonicity and bounds on the function values. For example, to realize a monotonicity SC, $g(m(X, p))$ could calculate the first derivatives of m on all points X with the paramters p.

The parameters p of candidate model m can be identified using the evolutionary mechanisms only, as exemplified by Koza in the early literature on genetic programming for SR (see [9]). Using this approach, no separate parameter identification step with a fixed model is conducted, and only mutation and selection are used to identify the model along with its parameters $m(p)$. However, most contemporary SR algorithms treat the parameter identification as a subordinate problem, employing gradient-based fitting for each candidate model before the loss in (2.1) is determined. Kommenda et al. have shown the benefits associated with this approach (see [8]).

Using $f(r) = r^2$, the parameter identification subproblem, i. e., (2.2) with the model m fixed, becomes a constrained least-squares problem. Despite the relevance of this problem class in many industrial and academic fields, we were unable to find an appropriate open-source implementation of a constrained least-squares problem solver.

We therefore employ a penalty approach. Given a model m with parameters p, we measure the total violation of the SC in terms of the ℓ_2-squared penalty, i. e.,

$$\sum_k \sum_{\ell=1}^{C} h_k(m(X^\ell, p))^2 \quad \text{and} \quad \sum_k \sum_{\ell=1}^{C} \max\bigl(0, g_k(m(X^\ell, p))\bigr)^2, \tag{2.3}$$

where k are the different SC and C are the points they are evaluated on. These values represent the extent to which the SC are violated at the points of evaluation. Notice that the SC evaluation points $X^1, \ldots, X^C$ are the same for each SC in (2.3). This is merely for the sake of notational convenience. In our implementation, we allow the SC evaluation points to be different for each component h_k and g_k of the SC functions.

During the parameter identification step in our algorithm, we minimize a weighted sum of the loss and the SC violation (2.3). In order to keep the computational expense limited, we work with a fixed penalty parameter. We use the Newton implementation of the Julia library Optim.jl (see [16]) for this purpose. First- and second-order derivatives of the loss as well as the SC are provided through the algorithmic differentiation package ForwardDiff.jl (see [19]). This is an essential distinction from [11], where Monte-Carlo-based multi-objective local search is used for parameter fitting. Gradient-based approaches, where available, often outperform non-deterministic approaches (see [15]).

With the parameter values optimized, there are at least two distinct ways for an evolutionary algorithm to take into account the amount of SC violation encountered in a particular model. In so-called hard-constrained approaches, candidate models are removed from the population in case they are infeasible. By contrast, soft-constrained approaches incorporate SC violations through one or several additional objectives. Hard constraints may lead to a loss of diversity in the population, which is essential for evolutionary algorithms, while soft SC allow to improve upon the candidate in later generations.

Haider et al. compared hard- and soft-constrained approaches and found no statistically significant differences between them (see [5]). We therefore choose to implement a soft-constrained approach, where the two terms in (2.3) are added to form a single additional objective.

3 Related Work

In an early work, [13] introduce asymptotic SC into SR, controlling the function behavior at zero and infinity. They utilize a neural network to guide their Monte-Carlo tree search-based method and to produce expressions with the desired asymptotic behavior. Although effective through restriction of the search space, it is not directly applicable to other types of SC.

[1,2] use a satisfiability modulo theories (SMT). The SMT solver guarantees that solutions are feasible, resembling a pessimistic approach. The types of SC considered include bounds on function values and partial derivatives, as well as symmetries and periodicity. Their approach, namely "Counterexample-Driven Genetic Programming", also maintains a training set with counter examples, which is continuously extended during the expression search. However, SMT solvers may be computationally prohibitive and do not support all types of SC.

Closest to our approach in the present work is [11]. The SC are evaluated on a finite number of sample points. Similar to our approach, the SC are part of the parameter identification subproblem. Contrary, however, they use a multi-objective local search procedure based on Monte-Carlo methods for this problem. This avoids computing derivatives with respect to the parameters, but gradient-based approaches often outperform non-deterministic approaches (see [15]).

[10] introduce a new approach to shape-constrained SR. They use interval arithmetic to check SC satisfaction in a pessimistic manner. Their study is limited to box constraints on the function values and its partial derivatives. Two approaches of incorporating the SC into the expression search are studied. One uses hard constraints, while the other maintains separate populations for feasible and infeasible candidates. In constrast to our work, the SC violations are not part of the parameter identification subproblem.

[5] employs soft constraints which allows the SC violation to be reduced over multiple generations. The authors utilize three different evolutionary algorithms as a basis and study various ways of treating and incorporating SC into the expression search. The benefits of SC, e. g., for noisy data, are clearly shown. Interestingly, the authors conclude that in all their experiments, there are no statistically significant differences between the core algorithms nor among the approaches to incorporate the SC.

[6] compare optimistic and pessimistic SC evaluation. The authors conclude that optimistic approaches improve the extrapolation behavior of models, while pessimistic approaches are better suited for higher levels of noise.

4 Experiments

We conduct experiments on three different benchmark problems. In each case, a ground truth model is known and used to generate synthetic data. The goal of our experiments is to study the utility in using SC to reconstruct the ground truth model. In contrast to previous works, we do not emphasize the effects of SC on the composition of the final Pareto frontier (hall of fame).

In what follows, we first detail how we set up our proposed method and two baseline methods we compare our method to. This is followed by a description of the three benchmark problems.

4.1 Description of Algorithmic Variants

All variants are implemented in TiSR. (1) As a first baseline (`base`), we use TiSR without SC. (2) The second baseline (`obj`) takes SC into account partially. As in `base`, it solves the parameter identification problem without considering the SC. It then evaluates the SC violation (2.3) and introduces the combined amount of violation as a separate, additional objective. This approach resembles the one of Haider et al. (see [5]). (3) The third variant (`minim_obj`) is the one proposed in this work. It alters the parameter identification step. To lower the computational expense, it first identifies the parameters them without considering the SC, like both baseline methods. If certain criteria are met (mentioned below), the parameter identification is continued using the penalty-based constrained least squares. The SC violations after the parameter identification are used for the selection, like in `obj`. The key differences are summarized in Figure 4.1.

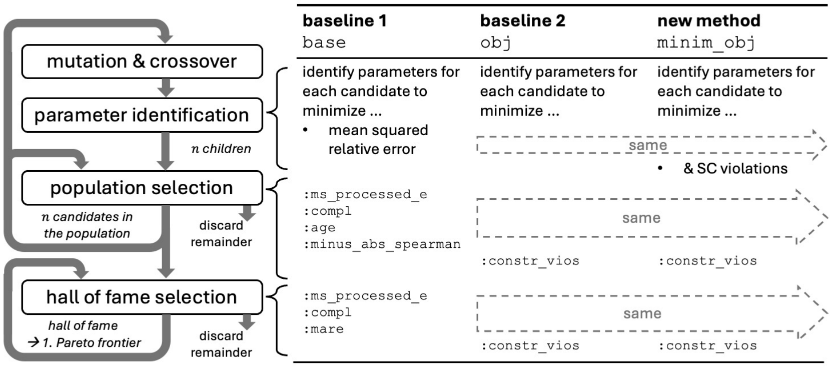

Fig. 4.1: Simplified illustration of NSGA-II (left) and overview of the key differences between the three algorithm variants with regard to parameter identification, population selection, and hall of fame selection (right).

Most of TiSR's default parameters are used in all three variants. To minimize the mean squared relative error, as shown in (2.1), the `fit_weights` parameter is set to $1/y_i$ for each residual value r_i, where y_i is the dependent variable for datum i. The maximum complexity, i. e., the number of operators and operands, is set to 5 above the reference ground truth expression. The default selection objectives of TiSR `[:ms_processed_e,:compl,:age]` are the mean squared error, complexity of the expression, and age of the expression (number of generations it has been present in the population). For the purpose of the present study, we extend these objectives by `:minus_abs_spear` and `:constr_vios`. Here

`:minus_abs_spear` is the negative of the absolute of the Spearman correlation, which is added as a supporting objective as defined in [23]. The `:constr_vios` objective contains the sum of the SC violations as in (2.3), which is always zero for the `base` variant of the algorithm. The default hall of fame objectives `[:ms_processed_e,:compl,:mare]` are extended by `:constr_vios`. The `:mare` objective represents the mean absolute relative error (MARE), which is calculated as in (2.1), but with loss function $f(r) = |r|$ in place of $f(r) = r^2$. Further parameters of TiSR that deviate from their defaults are summarized in Table 4.1.

Table 4.1: Overview and description of some of TiSR's parameters set for all experiments deviating from their default values.

Parameter	**Value**	**Description**
`t_lim`	`60*10`	The total runtime is set to ten minutes.
`pop_size`	`500`	The population size, i. e., the number of candidate models in the population, is set to 500.
`pow_abs_param`	`true`	The power operator (`^`) can only have parameters in its exponent. Variables or expressions are not allowed as exponents.
`always_drastic_simplify`	`1e-7`	Candidate models containing parameters that are closer to one or zero than 10^{-7} are simplified by rounding the respective parameters to one or zero.

All three algorithmic variants are allowed to use up to 30 iterations for each parameter identification subproblem. In fact, the `base` and `obj` variants use a Levenberg-Marquardt method for this purpose. The third variant `minim_obj` allots a budget of up to 20 iterations without SC penalization. Provided that the MARE is smaller than a pre-selected boundary, an additional maximum of 10 iterations with SC penalization follow. This boundary is set 5 % higher than the set noise level of the data. The SC penalty coefficient is fixed at 1.

4.2 Verification of Results

To check whether the ground truth is reconstructed, a separate verification data set is used. This data set is sampled on a wider range than the original data set, and using 2 to 5 times as much data. Approximately every 5 s, all candidate solutions in the hall of fame are passed to a callback function. Those with a MARE of at most 5 % higher than the noise level of the data are fitted to the verification data set. If the MARE with respect to the verification data set is lower than 0.0001 %, the experiment is terminated with success. We recall that only expressions with a complexity of up to 5 above the reference ground truth expression can be accepted. The expression search does not gain any information from the verification data set or the callback overall.

4.3 Description of the Benchmark Problems

We report on three different problems, each using a different ground truth expression. In all cases, no part of the expression nor its parameters are initially known to TiSR. Every problem has two independent and one dependent variable.

Gaussian Distribution The first problem (`gaussian`) is the density of a univariate Gaussian distribution with the mean fixed at 0:

$$y = \frac{\exp\left(-\frac{\theta^2}{2\sigma^2}\right)}{\sqrt{2\pi}\,\sigma}. \tag{4.1}$$

The independent variables σ and θ denote the standard deviation and the probability density, respectively. A graph illustrating Equation (4.1) is shown in Figure 4.2 on the left. We choose this as a benchmark problem, as there are many SC possible. The ground truth expression has a complexity of 11.

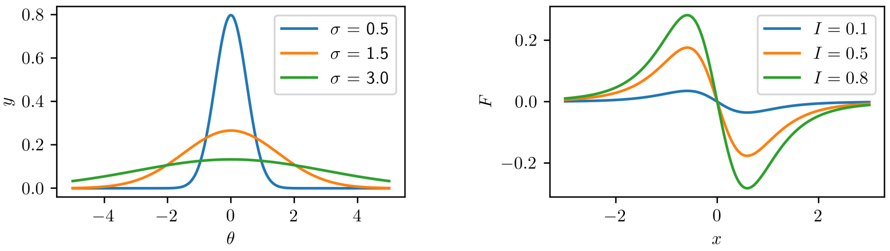

Fig. 4.2: Plot of the `gaussian` expression for varying standard deviation σ and random variable values θ (left) and the `magman` expression for varying distances x and varying current I (right).

For all variables, ranges are defined and within those, 100 points uniformly, randomly sampled. The ranges for the fitting data set are $[-5, 5]$ for θ, and $[0.5, 3]$ for σ. For the verification data set, 500 points are sampled within the ranges of $[-10, 10]$ and $[0.5, 5]$ for θ and σ, respectively. The function set for this expression is set to `[+,-,*,/,^,exp,pow2,sqrt]`, where `pow2` is a unary function raising to the power of 2.

We define SC on the function values and the first derivatives of the function. The SC on the function value are summarized in Table 4.2a, while the SC on the first derivatives are listed in Table 4.2b. For each of those, five points are sampled for each variable individually. Then five (θ, σ) pairs are randomly drawn from the total of the 25 potential pairs. These SC evaluation points remain the same during each run but may differ between runs.

Table 4.2: Collection of tables detailing the treatment of SC for `gaussian` and the `magman` expression.

(a) Treatment of the function value SC for the `gaussian` benchmark problem.

θ sample parameters	**σ sample parameters**	**SC**
logarithmically in $[-100, -0.01]$	uniformly in $[0.5, 6]$	$f \geq 0$
logarithmically in $[0.01, 100]$	uniformly in $[0.5, 6]$	$f \geq 0$

(b) Treatment of the monotonicity SC for the `gaussian` benchmark problem.

θ sample parameters	**σ sample parameters**	**SC**
logarithmically in $[-0.01, -100]$	uniformly in $[0.5, 6]$	$\partial f / \partial \theta \geq 0$
logarithmically in $[0.01, 100]$	uniformly in $[0.5, 6]$	$\partial f / \partial \theta \leq 0$
$\{0\}$	uniformly in $[0.5, 6]$	$\partial f / \partial \sigma \leq 0$

(c) Treatment of the function value SC for the `magman` benchmark problem.

x sample parameters	**I sample parameters**	**SC**
logarithmically in $[-1000, -0.1]$	uniformly in $[0.1, 0.8]$	$f \geq 0$
logarithmically in $[0.1, 1000]$	uniformly in $[0.1, 0.8]$	$f \leq 0$

(d) Treatment of the monotonicity SC for the `magman` benchmark problem, where r_1 and r_2 are the roots of the first derivative of the `magman` expression.

x sample parameters	**I sample parameters**	**SC**
logarithmically in $[-10, r_1]$	uniformly in $[0.1, 0.8]$	$\partial f / \partial x \geq 0$
uniformly in $[r_1, r_2]$	uniformly in $[0.1, 0.8]$	$\partial f / \partial x \leq 0$
logarithmically in $[r_2, 10]$	uniformly in $[0.1, 0.8]$	$\partial f / \partial x \geq 0$
logarithmically in $[-50, -0.1]$	uniformly in $[0.1, 0.8]$	$\partial f / \partial I \geq 0$
logarithmically in $[0.1, 50]$	uniformly in $[0.1, 0.8]$	$\partial f / \partial I \leq 0$

Magnetic Manipulator Force The second expression we use in our experiments is also used by [11] for their study of SR with SC. It describes the force exerted by an electromagnet on an iron ball moving along a rail in a magnetic manipulator system (`magman`). The expression is

$$F = \alpha \cdot \frac{x \cdot I}{(x^2 + \beta)^3}, \tag{4.2}$$

where the independent variables x and I denote the distance and the current, and the parameters α and β are system specific parameters. For (α, β), we adopt values $(5.25, 1.75)$ identified by [3]. The resulting function (4.2) is shown in Figure 4.2. The ground truth expression has a complexity of 11.

For the distance x, the variable range for the fitting data set is $[-3, 3]$, while we take $[-6, 6]$ for the verification data set. The ranges for the current I are $[0.1, 0.8]$ and $[0.1, 1.6]$ for the fitting and the verification data set, respectively. The function set is composed of `[+,-,*,/,^,pow2,pow3]`, where `pow3` is a unary function raising to the power of 3.

As in the case of the `gaussian` problem, function value and monotonicity SC are defined for `magman`, which are summarized in Table 4.2d. The selection of SC evaluation points is the same as described for the `gaussian` problem.

Van der Waals Force The third used expression is the Van der Waals thermodynamic equation of state (`vanderwaals`). It models the pressure p of a fluid as a function of the molar volume v and the temperature T as independent variables:

$$p = \frac{R \cdot T}{v - b} - \frac{a}{v^2}. \tag{4.3}$$

Here $R \approx 8.3145$ is the universal gas constant, and a and b are fluid-specific parameters. We use data for methanol, where the parameters (a, b) are $(0.9649, 6.702 \cdot 10^{-5})$ [22]. Figure 4.3 shows the function (4.3). The ground truth expression has a complexity of 12.

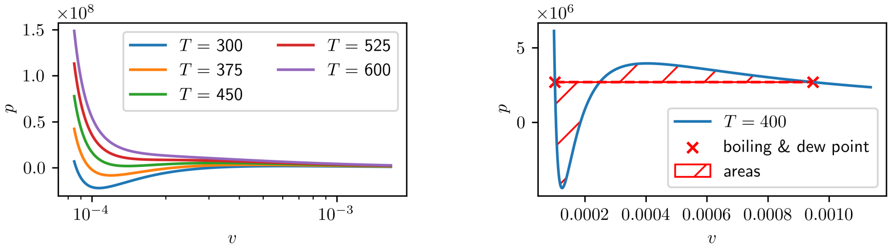

Fig. 4.3: Plot of the `vanderwaals` expression, where the pressure p is shown for varying temperatures T and specific volumes v (left), as well as an illustrative example to aid understanding of the Maxwell criterion at $T = 400$. The boiling and dew points are shown and connected with a dashed red line (right).

We choose the `vanderwaals` problem since it involves a relatively complex SC, allowing us to demonstrate the capabilities and potential of our approach to incorporate SC. The SC for this problem originates from the Maxwell criterion, as it is commonly enforced in thermodynamic equations of state to enable the calculation of the vapor-liquid phase-boundary. For states (T, v) inside the two-phase region, the pressure calculated by such equations is disregarded. Instead, the pressure along isotherms is constant throughout the two-phase region. We briefly summarize the essence to understand the SC without going into the thermodynamic detail. In Figure 4.3, the integral of the function along the isotherm (blue line) shifted by the entry points to the two-phase region (dashed red line) must be zero in-between the entry points (red cross).

We generate three equidistant, regular grids for the `vanderwaals` expression, one in the vapor, one in the liquid, and one in the supercritical phase. To obtain a more realistic grid, the variable ranges are defined to T and p rather than T and v, as is common in experimental practice. The variable ranges are $T = [450, 500]$, $p = [0.05 \cdot 10^6, 2 \cdot 10^6]$ for the gas phase, $T = [300, 400]$, $p = [6 \cdot 10^6, 7 \cdot 10^6]$ for the liquid phase, and $T = [550, 600]$, $p = [10 \cdot 10^6, 11 \cdot 10^6]$ for the supercritical phase. Instead of random sampling, the data are sampled equidistantly on regular grids in each of the three ranges. Additionally, four phase transitions points are added to the fitting data set. These points are used in the Maxwell criterion SC, and thus assumed to be known. Their values and usage in the SC are shown below. There are 147 and 300 points in the fitting and verification data set, respectively.

The Maxwell criterion is incorporated as an equality constraint. The exact phase transition states are specified inside the SC, marking the numerical integration bounds and the pressure-shifts for two isotherms. To reduce the computationally expensive numerical integration, the SC has two parts. Before assessing the Maxwell criterion in the second part, the deviations from the phase-transition states are assessed and also treated as equality constraints. Only for MARE to the transition states below 1 %, the integration is conducted, the Maxwell criterion checked, and the violation added to the one described above. To ensure a descent in the transition from the first to the second part of the SC, a dummy penalty of 1000 is added to the MARE, while the results of the integration are divided by 1000. This fix does not render the transition smooth, but ensures a descent in this case, and is worth the computational expense saved. For the isotherms $T_1 = 300$ and $T_2 = 400$, the pressure offsets are $p_1 \approx 595\,000$ and $p_2 \approx 2.70 \cdot 10^6$. The specific volumes at the two-phase transitions, which mark the integration bounds, are $v_{1,\mathrm{boiling}} \approx 8.61 \cdot 10^{-5}$ and $v_{1,\mathrm{dew}} \approx 0.003\,85$ for T_1, and $v_{2,\mathrm{boiling}} \approx 0.000\,102$ and $v_{2,\mathrm{dew}} = 0.000\,947$ for T_2.

4.4 Increasing Noise Levels

We conduct two studies using the three algorithmic variants and the three benchmark problems. In the first, the resilience to noise is studied by applying varying noise levels on the dependent component of each data set. We use normally distributed random noise scaled by the respective target to model relative noise. Each algorithm-noise level combination is run for 31 times for statistical comparison. The results are shown in Figure 4.4.

For 10 % and 30 % noise, all variants are successful between 28 and 31 times out of 31 runs. Also, for the `magman` expression search at 35 % noise level, all variants find the expression in all runs. At 35 % noise for the `gaussian` expression, `base` finds the expression only 14 times, while `obj` and `minim_obj` find it 15 and 17 times, respectively. The two-proportion z-test with a confidence level of 5 % is used to determine whether there are statistically significant differences for the three algorithms. For this study, the three algorithmic variants do not exhibit statistically significant differences.

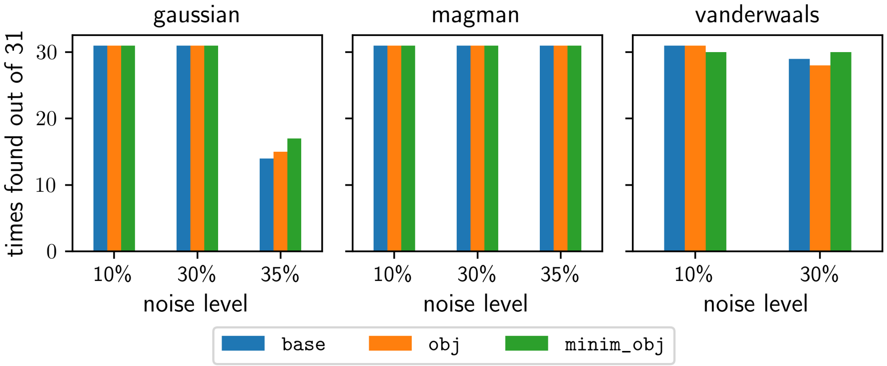

Fig. 4.4: Times out of 31 that each of the three algorithmic variants (`base`, `obj`, `minim_obj`) find each of the three ground truth expressions (`gaussian`, `magman`, `vanderwaals`) for noise levels of 10 %, 30 %, and 35 % noise levels.

4.5 Reducing Data

In the first part of the second study, we successively reduce the data available for fitting at 10 % noise level. We filter the data sets, by removing the data closest to the normalized center of the data, thus creating a “data hole”. First, data is normalized by dividing each datum x_{ij} by the maximum datum of its column j:

$$x_{ij}^{\text{norm}} = \frac{x_{ij}}{\max(x_j)} \tag{4.4}$$

Thereafter, the normalized center μ_j for each column j is determined using

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}^{\text{norm}}. \tag{4.5}$$

The data with the highest Euclidean distance to the normalized center, calculated by

$$d_i^2 = \sum_{j=1}^{m} (x_{ij}^{\text{norm}} - \mu_j)^2, \tag{4.6}$$

is removed last. Each experiment is repeated ten times. The study is conducted for the `gaussian` and the `magman` expression. The results are shown in Figure 4.5.

First, the significant results for the `gaussian` benchmark problem (see upper plot of Figure 4.5) are discussed. Assuming a significance level of 5 %, for 14 points, `minim_obj` is statistically significantly better than `base`, but not statistically significantly better than `obj`. Also, `obj` is not statistically significantly better than `base` in this case. At twelve points, both `obj` and `minim_obj` are statistically significantly better than `base`. For eight points, `minim_obj` is statistically significantly better than the other two variants, while `obj` is not significantly better than `base`. For six points, the only statistically significant result is

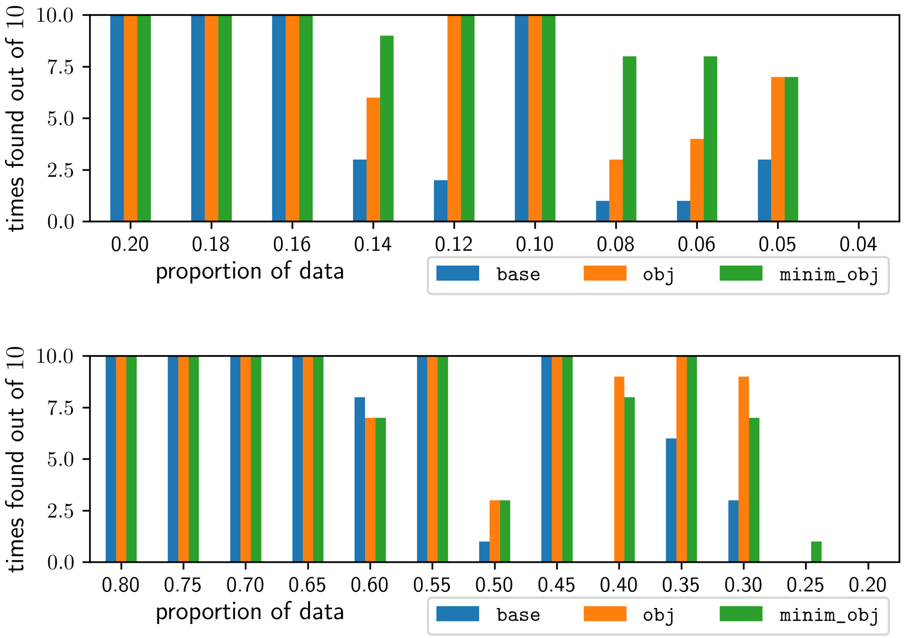

Fig. 4.5: Times out of ten that each algorithmic variant finds the `gaussian` (upper plot) and the `magman` (lower plot) expressions at 10 % noise level for different proportions of data out of 100 data points.

that `minim_obj` is better than `base`. At five and four points, none of the results are statistically significant.

The bottom plot of Figure 4.5 shows the same experiments for the `magman` expression. For 40 and 35 points, `obj` and `minim_obj` are significantly better than `base`, while not being statistically significantly different. At 30 points, `obj` is significantly better than `base`, while `minim_obj` is not. However, `obj` is not statistically significantly better than `minim_obj`.

For the `vanderwaals` expression, the data are reduced in a different manner. As mentioned above, the initial `vanderwaals` data set consists of three regular grids in three phases, i. e., the gas, the liquid, and the supercritical phase, as well as four points at the vapor and dew line, where no noise is added. Here, only the data in the liquid phase and the additional points are kept. Ten experiments are conducted with each, 0 % and 10 % noise levels. The results are shown in Figure 4.6. None of the results show statistically significant differences.

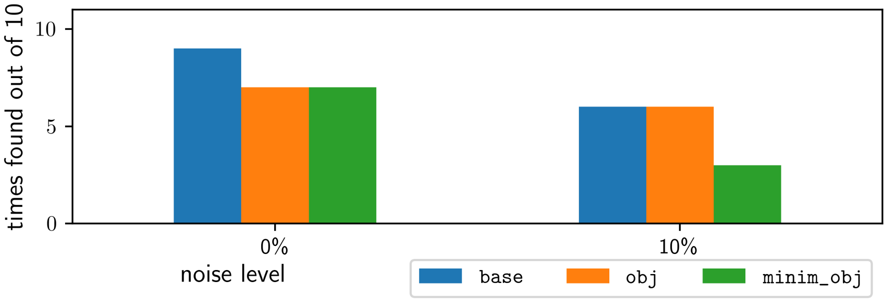

Fig. 4.6: Times out of ten that each algorithmic variant finds the `vanderwaals` expression using only data from the liquid phase and four phase transition points at 0 % and 10 % noise levels.

5 Summary and Discussion

In Section 4, we compare three algorithmic variants that differ in the way they incorporate used-defined SC (1) The first baseline variant (`base`) does not consider SC at all. (2) The second baseline variant (`obj`) likewise uses unconstrained least squares for the parameter identification subproblem and subsequently uses the SC violation for the purpose of selection. This resembles the algorithm proposed by Haider et al. (see [5]). (3) The third, proposed variant (`minim_obj`) already incorporates the SC violation into the parameter optimization step. Our implementation of all three variants are based on SR library TiSR.

Several experiments are conducted to test the resilience to noise and to a reduction of the amount of data. The data is synthetically generated from ground truth expressions. The experimental setup, including the verification step, allows us to evaluate the differences between the three algorithmic variants with respect to their capability of recovering the ground truth models exactly.

With regard to the noise level, our experiments indicate that the algorithmic variants that utilize SC (`obj` and `minim_obj`), do not show significant benefits compared to `base`. With regard to the amount of data, however, the two variants `obj` and `minim_obj` do exhibit statistically significant benefits compared to `base`. Among the two, our proposed approach `minim_obj` outperforms `obj` in a few cases with statistical significance.

Admittedly, we expected our proposed approach to stand out more clearly. This might be due to several reasons. For one, for all variants (including `base`), TiSR employs regularization by adding a squared ℓ_1-norm of the parameter vector, which makes it more robust against increasing noise levels and overfitting.

Furthermore, the data is synthetically generated using ground truth expressions with normally distributed noise and no outliers. We conjecture that for problems with systematic deviations or outliers in the data, minimizing SC violations during parameter identification might exhibit more significant benefits.

Finally, in situations where data is limited to a relatively small region, but additional knowledge is available concerning some global properties, SC are expected to be beneficial to achieve the proper extrapolation behavior. This situation, however, is not reflected in our current choice of example problems.

6 Conclusion

In this work, we propose a new way of incorporating SC into SR. Our approach uniquely accounts for SC violations during the parameter optimization phase and leverages second-order optimization along with algorithmic differentiation to enhance performance. For this purpose, we evaluate the SC violations in a number of sample points and accumulate them into a scalar value. This value is used both as a penalty during parameter optimization as well as a separate objective during the selection process. To manage the computational expense, we propose to include the penalty term in to the parameter optimization step only for models that demonstrate good fit quality already in the absence of SC. This preferred variant of our algorithm is referred to as `minim_obj`. In our experiments, this approach is statistically significantly better in a few of our test cases compared to two baseline variants, while it is never significantly worse.

Potential future work may study the benefits of this approach for empirical data in the absence of a ground truth model.

References

1. Błądek, I., Krawiec, K.: Solving symbolic regression problems with formal constraints. In: Proceedings of the Genetic and Evolutionary Computation Conference. GECCO '19, ACM (2019). https://doi.org/10.1145/3321707.3321743
2. Błądek, I., Krawiec, K.: Counterexample-driven genetic programming for symbolic regression with formal constraints. IEEE Transactions on Evolutionary Computation **27**(5), 1327–1339 (2023). https://doi.org/10.1109/tevc.2022.3205286
3. Damsteeg, J.W., Nageshrao, S.P., Babuska, R.: Model-based real-time control of a magnetic manipulator system. In: 2017 IEEE 56th Annual Conference on Decision and Control (CDC). IEEE (2017). https://doi.org/10.1109/cdc.2017.8264140
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (2002). https://doi.org/10.1109/4235.996017
5. Haider, C., Olivetti de França, F., Burlacu, B., Kronberger, G.: Using shape constraints for improving symbolic regression models. arXiv: 2107.09458 (2021)
6. Haider, C., Olivetti de França, F., Kronberger, G., Burlacu, B.: Comparing optimistic and pessimistic constraint evaluation in shape-constrained symbolic regression. In: Proceedings of the Genetic and Evolutionary Computation Conference. GECCO '22, ACM (2022). https://doi.org/10.1145/3512290.3528714
7. Haider, C., Kronberger, G.: Shape-constrained symbolic regression with NSGA-III, pp. 164–172. Springer Nature Switzerland (2022). https://doi.org/10.1007/978-3-031-25312-6_19

8. Kommenda, M.: Local Optimization and Complexity Control for Symbolic Regression. Ph.D. thesis, Johannes Kepler University Linz (2018), URN: urn:nbn:at:at-ubl:1-21036
9. Koza, J.R.: Genetic programming as a means for programming computers by natural selection. Statistics and Computing **4**(2) (1994). https://doi.org/10.1007/bf00175355
10. Kronberger, G., Olivetti de França, F., Burlacu, B., Haider, C., Kommenda, M.: Shape-constrained symbolic regression—improving extrapolation with prior knowledge. Evolutionary Computation **30**(1), 75–98 (2022). https://doi.org/10.1162/evco_a_00294
11. Kubalík, J., Derner, E., Babuška, R.: Symbolic regression driven by training data and prior knowledge. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference. GECCO '20, ACM (2020). https://doi.org/10.1145/3377930.3390152
12. La Cava, W., Orzechowski, P., Burlacu, B., Olivetti de França, F., Virgolin, M., Jin, Y., Kommenda, M., Moore, J.H.: Contemporary symbolic regression methods and their relative performance. arXiv: 2107.14351 (2021)
13. Li, L., Fan, M., Singh, R., Riley, P.: Neural-guided symbolic regression with asymptotic constraints. arXiv: 1901.07714 (2019)
14. Martinek, V., Frotscher, O., Richter, M., Herzog, R.: Introducing thermodynamics-informed symbolic regression – a tool for thermodynamic equations of state development. arXiv: 2309.02805 (2023)
15. Martins, J.R.R.A., Ning, A.: Engineering Design Optimization. Cambridge University Press (2021). https://doi.org/10.1017/9781108980647
16. Mogensen, P.K., Riseth, A.N.: Optim: A mathematical optimization package for Julia. Journal of Open Source Software **3**(24), 615 (2018). https://doi.org/10.21105/joss.00615
17. Piringer, D., Wagner, S., Haider, C., Fohler, A., Silber, S., Affenzeller, M.: Improving the flexibility of shape-constrained symbolic regression with extended constraints, pp. 155–163. Springer Nature Switzerland (2022). https://doi.org/10.1007/978-3-031-25312-6_18
18. Reinbold, P.A.K., Kageorge, L.M., Schatz, M.F., Grigoriev, R.O.: Robust learning from noisy, incomplete, high-dimensional experimental data via physically constrained symbolic regression. Nature Communications **12**(1) (2021). https://doi.org/10.1038/s41467-021-23479-0
19. Revels, J., Lubin, M., Papamarkou, T.: Forward-mode automatic differentiation in Julia. arXiv: 1607.07892 (2016)
20. Schmidt, M., Lipson, H.: Age-fitness Pareto optimization. In: Riolo, R., McConaghy, T., Vladislavleva, E. (eds.) Genetic Programming Theory and Practice VIII, pp. 129–146. Springer New York (2010). https://doi.org/10.1007/978-1-4419-7747-2_8
21. Virgolin, M., Pissis, S.P.: Symbolic regression is NP-hard. arXiv: 2207.01018 (2022)
22. Weast, R., Company, C.R.: Handbook of Chemistry and Physics: A Ready-reference Book of Chemical and Physical Data. CRC handbook series, The Company (1972)
23. Zille, H., Evrard, F., Reuter, J., Mostaghim, S., van Wachem, B.: Assessment of multi-objective and coevolutionary genetic programming for predicting the Stokes flow around a sphere. In: 14th International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control. EUROGEN 2021, Institute of Structural Analysis and Antiseismic Research, National Technical University of Athens (2021). https://doi.org/10.7712/140121.7959.18341