

Article

Compensating Data Shortages in Manufacturing with Monotonicity Knowledge

Martin von Kurnatowski ¹, Jochen Schmid ^{1,*}, Patrick Link ², Rebekka Zache ², Lukas Morand ³,
Torsten Kraft ³, Ingo Schmidt ³, Jan Schwientek ¹ and Anke Stoll ²

¹ Fraunhofer Institute for Industrial Mathematics ITWM, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany; martin.von.kurnatowski@itwm.fraunhofer.de (M.v.K.); jan.schwientek@itwm.fraunhofer.de (J.S.)

² Fraunhofer Institute for Machine Tools and Forming Technology IWU, Reichenhainer Straße 88, 09126 Chemnitz, Germany; patrick.link@iwu.fraunhofer.de (P.L.); rebekka.zache@iwu.fraunhofer.de (R.Z.); anke.stoll@iwu.fraunhofer.de (A.S.)

³ Fraunhofer Institute for Mechanics of Materials IWM, Wöhlerstraße 11, 79108 Freiburg, Germany; lukas.morand@iwm.fraunhofer.de (L.M.); torsten.kraft@iwm.fraunhofer.de (T.K.); ingo.schmidt@iwm.fraunhofer.de (I.S.)

* Correspondence: jochen.schmid@itwm.fraunhofer.de

Abstract: Systematic decision making in engineering requires appropriate models. In this article, we introduce a regression method for enhancing the predictive power of a model by exploiting expert knowledge in the form of shape constraints, or more specifically, monotonicity constraints. Incorporating such information is particularly useful when the available datasets are small or do not cover the entire input space, as is often the case in manufacturing applications. We set up the regression subject to the considered monotonicity constraints as a semi-infinite optimization problem, and propose an adaptive solution algorithm. The method is applicable in multiple dimensions and can be extended to more general shape constraints. It was tested and validated on two real-world manufacturing processes, namely, laser glass bending and press hardening of sheet metal. It was found that the resulting models both complied well with the expert's monotonicity knowledge and predicted the training data accurately. The suggested approach led to lower root-mean-squared errors than comparative methods from the literature for the sparse datasets considered in this work.

Keywords: monotonic regression; manufacturing; informed machine learning; expert knowledge; semi-infinite optimization; shape constraints



Citation: Kurnatowski, M.v.; Schmid, J.; Link, P.; Zache, R.; Morand, L.; Kraft, T.; Schmidt, I.; Schwientek, J.; Stoll, A. Compensating Data Shortages in Manufacturing with Monotonicity Knowledge. *Algorithms* **2021**, *14*, 345. <https://doi.org/10.3390/a14120345>

Academic Editors: El-ghazali Talbi, Bernabe Dorransoro, Lionel Amodeo, Vincenzo Cutello and Mario Pavone

Received: 19 October 2021

Accepted: 25 November 2021

Published: 27 November 2021

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



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Systematic decision making in manufacturing—such as finding optimal parameter settings for a manufacturing process—requires appropriate models for that process. In particular, such models have to be sufficiently accurate, and at the same time, sufficiently quick at evaluating. In principle, for many industrial processes, precise simulation models based on detailed physical modeling can be built. Yet, these so-called white-box models are typically too slow to be of any practical use in exploring the process parameter space and in eventually finding optimal process parameters (online and offline). In this respect, machine learning models can be very useful surrogates with short runtimes.

Conventional machine learning models are purely data-based (so-called black-box models). Accordingly, the predictive power of such models is generally bad if the underlying training data $\mathcal{D} = \{(x_l, t_l) : l \in \{1, \dots, N\}\}$ are insufficient. Unfortunately, such data insufficiencies occur quite often, and they can come in one of the following forms: On the one hand, the available datasets can be too small and have too little variance in the input data points x_1, \dots, x_N . This problem frequently occurs in manufacturing [1] because varying the process parameters beyond well-tested operating windows is usually costly. On the other hand, the output data t_1, \dots, t_N can be too noisy.

Aside from potentially insufficient data, however, one often also has additional knowledge about the relation between the input variables and the responses to be learned. Such extra knowledge about the considered process is referred to as expert knowledge in the following. In [2], the interactions of users with software was tracked to capture their expert knowledge in a general form as training data for a classification problem. In [3], expert knowledge was used in the form of a specific algebraic relation between input and output to solve a parameter estimation problem with artificial neural networks. Such informed machine learning [4] techniques beneficially combine expert knowledge and data to build hybrid or gray-box models [5–12], which predict the responses more accurately than purely data-based models. In other words, by using informed machine learning techniques, one can compensate data insufficiencies with expert knowledge.

An important and common type of expert knowledge is prior information about the monotonicity behavior of the unknown functional relationship $x \mapsto y(x)$ to be learned. A large variety of concrete application examples with monotonicity knowledge can be found in ([13], Section 4.1) and ([14], Section 1), for instance. The present article exclusively deals with regression under such monotonicity requirements. For classification under monotonicity constraints, see, e.g., [14,15]. Along with convexity constraints, monotonicity constraints are probably the most intensively studied shape constraints [16] in the literature, and correspondingly, there exist plenty of different approaches to incorporate monotonicity knowledge in a machine learning model. See [17] for an extensive overview. Very roughly, these approaches can be categorized according to when the monotonicity knowledge is taken into account: in or only after the training phase. In the terminology of [4], this corresponds to the distinction between knowledge integration in the learning algorithm or in the final hypothesis.

A lot of methods—especially from the mathematical statistics literature, such as [18–24]—incorporate monotonicity knowledge only after training. These articles start with a purely data-based initial model, which in general does not satisfy the monotonicity requirements, and then monotonize this initial model according to a suitable monotonization procedure, such as projection [18–20,24], rearrangement [22,23,25] or tilting [21]. Among other things, it is shown in the mentioned articles that, in spite of noise in the output data, the arising monotonized models are close to the true relationship for sufficiently large training datasets. Summarizing, these articles show that for large datasets, noise in the output data can be compensated by monotonization to a certain extent.

In contrast to that, in some works, such as [13,17,26–29], monotonicity knowledge was incorporated already in training. In these articles, the monotonicity requirements were added as constraints—either hard [17,26,28,29] or soft [13,26]—to the data-based optimization of the model parameters. In [13,28], probabilistic monotonicity notions are used. In [26–29], support vector regressors in the linear-programming or the more standard quadratic-programming form, Gaussian process regressors or neural network models were considered, and the monotonicity of these models was enforced by constraints on the model derivatives at predefined sampling points [26,28,29] or on the model increments between predefined pairs of sampling points [27].

A disadvantage of the projection- and rearrangement-based approaches [22–24] from the point of view of manufacturing applications is that these methods are tailored to large datasets. Another disadvantage of these approaches is that the resulting models typically exhibit distinctive kinks, which are almost always unphysical. Additionally, the models resulting from the multidimensional rearrangement method by [23] are not guaranteed to be monotonic when trained on small datasets. A drawback of the tilting approach from [21] is that it is formulated and validated only for one-dimensional input spaces (intervals in \mathbb{R}). Accordingly, naively extending the non-adaptive discretization scheme from [21] to higher dimensions would result in long computation times. A downside of the in-training methods from [26,28,29] is that the sampling points at which the monotonicity constraints are imposed have to be chosen in advance (even though they need not coincide with the training data points).

With the method proposed in the present article, we address the aforementioned issues and shortcomings. In Section 2, our methodology for monotonic regression using semi-infinite optimization is introduced. It incorporates the monotonicity knowledge during training. Specifically, polynomial regression models are assumed for the input–output relationships to be learned. Since there is no after-training monotonicization step in the method, our models are smooth, and in particular, do not exhibit kinks. Also, due to the employed adaptive discretization scheme, the method is computationally efficient also in higher dimensions. To our knowledge, such an adaptive scheme has not been applied to solve monotonic regression problems before, especially not in situations with sparse data. In Section 4, the method is validated by means of two applications to real-world processes which are both introduced in Section 3, namely, laser glass bending and press hardening of sheet metal. It turns out that the adaptive semi-infinite optimization approach to monotonic regression is better suited for the considered applications with their small datasets and the resulting models are more accurate than those obtained with the comparative approaches from the literature.

2. Semi-Infinite Optimization Approach to Monotonic Regression

In this section, our semi-infinite optimization approach to monotonic regression is introduced. It will be referred to as the SIAMOR method later on for brevity.

2.1. Semi-Infinite Optimization Formulation of Monotonic Regression

In our approach to monotonic regression, polynomial models

$$x \mapsto \hat{y}_w(x) = \sum_{|\alpha| \leq m} w_\alpha x^\alpha \in \mathbb{R} \tag{1}$$

are used for all input–output relationships $x \mapsto y(x)$ to be learned. In the above relation (1), the sum extends over all d -dimensional multi-indices ([30], Section 1) $\alpha = (\alpha_1, \dots, \alpha_d) \in \mathbb{N}_0^d$ with degree $|\alpha| := \alpha_1 + \dots + \alpha_d$ less than or equal to some total degree $m \in \mathbb{N}$. The terms $x^\alpha := x_1^{\alpha_1} \dots x_d^{\alpha_d}$ are the monomials in d variables of degrees less than or equal to m , and w_α are the corresponding model parameters to be tuned by regression. Since there are exactly

$$N_m = \sum_{k=0}^m \binom{k+d-1}{d-1} = \binom{m+d}{m} \tag{2}$$

d -dimensional monomials of degrees less than or equal to m , the polynomial regression model (1) can be equivalently written as

$$\hat{y}_w(x) = \sum_{i=1}^{N_m} w_i \phi_i(x) = w^\top \boldsymbol{\phi}(x), \tag{3}$$

where the basis functions $\phi_1, \dots, \phi_{N_m}$ constitute any enumeration of the d -dimensional monomials of degrees less than or equal to m , while $w := (w_1, \dots, w_{N_m})^\top$ and $\boldsymbol{\phi}(x) := (\phi_1(x), \dots, \phi_{N_m}(x))^\top$.

Standard polynomial regression without regularization ([31], Section 2.1) is about solving the unconstrained optimization problem:

$$\min_{w \in W} \frac{1}{2} \sum_{l=1}^N (\hat{y}_w(x_l) - t_l)^2, \tag{4}$$

or in other words, about optimally adapting the model parameters $w_i \in [-r, r]$ of the polynomial model (3) to the available dataset $\mathcal{D} = \{(x_l, t_l) : l \in \{1, \dots, N\}\}$ containing

N points. In the above relation, the monomial coefficients are allowed to vary in the compact hyperbox

$$W = \{\mathbf{w} \in \mathbb{R}^{N_m} : -r \leq w_i \leq r \text{ for all } i \in \{1, \dots, N_m\}\} \tag{5}$$

with some large but finite $r > 0$. Since W is compact and non-empty, and since the mean-squared error objective function of (4) is continuous, the standard polynomial regression problem (4) for any given dataset \mathcal{D} has a solution \mathbf{w} (which is unique if, for instance, an ℓ^2 -regularization term is added).

In general, however, the resulting model $\mathbf{x} \mapsto \hat{y}_{\mathbf{w}}(\mathbf{x})$ does not necessarily exhibit the monotonicity behavior an expert expects for the underlying true physical relationship $\mathbf{x} \mapsto y(\mathbf{x})$. In order to enforce the expected monotonicity behavior, the following constraints on the signs of the partial derivatives $\partial_{x_j} \hat{y}_{\mathbf{w}}(\mathbf{x})$ are added to the unconstrained standard regression problem (4):

$$\sigma_j \cdot \partial_{x_j} \hat{y}_{\mathbf{w}}(\mathbf{x}) \geq 0 \quad \text{for all } j \in J \text{ and } \mathbf{x} \in X. \tag{6}$$

The numbers $\sigma_j \in \{-1, 0, 1\}$ indicate the expected monotonicity behavior for each coordinate direction $j \in \{1, \dots, d\}$:

- $\sigma_j = 1$ and $\sigma_j = -1$ indicate that $\mathbf{x} \mapsto y(\mathbf{x})$ is expected to be, respectively, monotonically increasing or decreasing in the j th coordinate direction;
- $\sigma_j = 0$ indicates that one has no monotonicity knowledge in the j th coordinate direction.

Also, $J := \{j \in \{1, \dots, d\} : \sigma_j \neq 0\}$ is the set of all directions for which a monotonicity constraint is imposed, and the vector

$$\boldsymbol{\sigma} := (\sigma_1, \dots, \sigma_d)$$

is referred to as the monotonicity signature of the relationship $\mathbf{x} \mapsto y(\mathbf{x})$. Finally, $X \subset \mathbb{R}^d$ is the (continuous) subset of the input space on which the polynomial model (1) is supposed to be a reasonable prediction for $\mathbf{x} \mapsto y(\mathbf{x})$. In this work, X was chosen to be identical with the range covered by the input training data points $\mathbf{x}_1, \dots, \mathbf{x}_N$. I.e., X is the compact hyperbox

$$X = [a_1, b_1] \times \dots \times [a_d, b_d] \tag{7}$$

with $a_j := \min_{l=1, \dots, N} x_{l,j}$ and $b_j := \max_{l=1, \dots, N} x_{l,j}$ and with $x_{l,j}$ denoting the j th component of the l th input data point \mathbf{x}_l . Writing

$$f(\mathbf{w}) := \frac{1}{2} \sum_{l=1}^N (\hat{y}_{\mathbf{w}}(\mathbf{x}_l) - t_l)^2 \quad \text{and} \quad g_j(\mathbf{w}, \mathbf{x}) := \sigma_j \cdot \partial_{x_j} \hat{y}_{\mathbf{w}}(\mathbf{x}) \tag{8}$$

for brevity, our monotonic regression problem (4)–(6) takes the neat and simple form

$$\min_{\mathbf{w} \in W} f(\mathbf{w}) \quad \text{s.t.} \quad g_j(\mathbf{w}, \mathbf{x}) \geq 0 \quad \text{for all } j \in J \text{ and } \mathbf{x} \in X. \tag{9}$$

Since the input set X is continuous and hence contains infinitely many points \mathbf{x} , the monotonic regression problem (9) features infinitely many inequality constraints. Consequently, (9) is a semi-infinite optimization problem [32–36] (or more precisely, a standard semi-infinite optimization problem, as opposed to a generalized one). It is well-known that any semi-infinite problem—and, in particular, the monotonic regression problem (9)—can be equivalently rewritten as a bi-level optimization problem [35,37,38], namely,

$$\min_{\mathbf{w} \in W} f(\mathbf{w}) \quad \text{s.t.} \quad \min_{\mathbf{x} \in X} g_j(\mathbf{w}, \mathbf{x}) \geq 0 \quad \text{for all } j \in J. \tag{10}$$

Commonly, the minimization subproblems in the constraints of (10) are called lower-level problems of (9).

2.2. Adaptive Solution Strategy

Since the feasible set of (9) is compact (by the finiteness of the parameter r in (5)) and non-empty (it contains $w := 0 \in \mathbb{R}^{N_m}$), the monotonic regression problem (9) has a solution by virtue of the Weierstraß extreme-value theorem. In order to compute such a solution of (9), a variant of the adaptive, iterative discretization algorithm by [39] is used. In a nutshell, the idea is the following: the infinite index set X of the original regression problem (9) is iteratively replaced by discretizations, that is, finite subsets $X^k \subset X$, and these discretizations are adaptively refined from iteration to iteration. In that manner, in every iteration k one obtains the ordinary (finite) optimization problem

$$\min_{w \in W} f(w) \quad \text{s.t.} \quad g_j(w, x) \geq 0 \quad \text{for all } j \in J \text{ and } x \in X^k \tag{11}$$

featuring only finitely many inequality constraints. As usual, we refer to (11) as the k th discretized problem. In each iteration k , two steps are performed, namely, an optimization step and an adaptive refinement step. In the optimization step, a solution w^k of the k th discretized problem (11) is computed. In the refinement step, for each direction $j \in J$, a point $x^{k,j} \in X$ is computed at which the j th monotonicity constraint at $w = w^k$ is violated most. In more precise terms, for every $j \in J$, an approximate solution $x^{k,j}$ of the global optimization problem

$$\min_{x \in X} g_j(w^k, x) \tag{12}$$

is computed. All the points $x^{k,j}$ for which a monotonicity violation occurs are then added to the current discretization X^k in order to obtain the new discretization X^{k+1} . If no more monotonicity violations occur, the iteration is stopped. As usual, (12) is referred to as the (k, j) th lower-level problem in the following.

With regard to the practical implementation of the above solution strategy, it is important to observe that the discretized problems (11) are standard convex quadratic programs [40]. Indeed, by inserting (3) into (8) and using the design matrix Φ with entries $\Phi_{li} := \phi_i(x_l)$, one obtains

$$f(w) = \frac{1}{2} \|\Phi w - t\|_2^2 = \frac{1}{2} w^\top \Phi^\top \Phi w - t^\top \Phi w + \frac{1}{2} t^\top t.$$

Consequently, the objective function of (11) is indeed quadratic and convex with respect to w . It is not strictly convex, though, in the sparse-data case $N < N_m$ considered in this paper. (Indeed, the kernel of the matrix $\Phi^\top \Phi \in \mathbb{R}^{N_m \times N_m}$ is equal to the kernel of $\Phi \in \mathbb{R}^{N \times N_m}$, and an $N \times N_m$ matrix with $N < N_m$ has a non-trivial kernel, of course.) Also, in view of

$$g_j(w, x) = \sigma_j \cdot \partial_{x_j} \hat{y}_w(x) = \sigma_j \cdot w^\top (\partial_{x_j} \phi(x)), \tag{13}$$

the constraints of (11) are indeed linear with respect to w .

With regard to the practical implementation, it is also important to observe that the objective functions $x \mapsto g_j(w^k, x)$ of the lower-level problems (12) are non-convex polynomials and therefore in general have several local minima. Consequently, (12) needs to be solved numerically with a global optimization solver.

2.3. Algorithm and Implementation Details

In the following, our adaptive discretization algorithm is described in detail. As has already been pointed out above, it is a variant of the general algorithm developed

by ([39], Section 2), and it is explained after Algorithm 1 how our variant differs from its prototype [39].

Algorithm 1 Adaptive discretization algorithm for monotonic regression.

Choose a coarse (but non-empty) rectangular grid X^0 in X . Set $k = 0$ and iterate over k .

1. Solve the k th discretized problem (11) to obtain optimal model parameters $w^k \in W$.
2. Solve the (k, j) th lower-level problem (12) approximately for every $j \in J$ to find approximate global minimizers $x^{k,j} \in X$. Add those of the points $x^{k,j}$ for which substantial monotonicity violations occur, i.e., for which

$$g_j(w^k, x^{k,j}) < -\varepsilon_j,$$

to the current discretization X^k and go to Step 1 with $k = k + 1$. If for none of the points $x^{k,j}$ substantial monotonicity violations occur, go to Step 3.

3. Check for monotonicity violations on a fixed, fine rectangular reference grid $X_{\text{ref}} \subset X$. If there are no such violations, that is, if

$$g_j(w^k, x) \geq -\varepsilon_j$$

for all $j \in J$ and $x \in X_{\text{ref}}$, then terminate. Else, for every direction j with violations, add the reference grid point $x_{\text{ref}}^{k,j}$ with the largest violation to X^k and go to Step 1 with $k = k + 1$.

In contrast to [39], the algorithm above does not require exact solutions of the (non-convex) lower-level problems. Indeed, Step 2 of Algorithm 1 only requires finding an approximate solution. Also, slight constraint violations are tolerated (Step 2 and 3), and a feasibility check on a reference grid (Step 3) is performed before termination. Without the feasibility check on the reference grid, it could happen that the algorithm—because of the merely approximate solutions of the lower-level problems—terminates at models \hat{y}_{w^k} which do not satisfy the imposed monotonicity constraints sufficiently well. Another difference to the algorithm from [39] is that there are several lower-level problems in each iteration in this work and not just one, because monotonicity is enforced in multiple coordinate directions in general.

In our specific applications, the parameters inherent in Algorithm 1 were chosen as follows. The degrees m of the polynomial models in this work were chosen as the largest possible values that did not result in an overfit, because increasing m enhances the model's accuracy in general. In this respect, the number of model parameters was allowed to exceed the number of data points ($N_m \geq N$), since the constraints represent additional information supplementing the data. As for the parameter r in (5), one only has to make sure that it is so large that the resulting box constraints in the discretized problems (11) actually do not restrain the solutions w^k (Step 1 of Algorithm 1). In other words, r should be so large that relaxing or even dropping the pertaining box constraints does not improve the minimizer computed for (11) anymore. In the specific applications considered here, $r = 10^5$ turned out to meet this requirement. As for the tolerances ε_j (Steps 2 and 3 of Algorithm 1), a monotonicity violation of 1% of the ranges covered by the in- and output training data was allowed for:

$$\varepsilon_j = 0.01 \frac{\max_{l=1, \dots, N} t_l - \min_{l=1, \dots, N} t_l}{\max_{l=1, \dots, N} x_{l,j} - \min_{l=1, \dots, N} x_{l,j}}. \quad (14)$$

And finally, concerning the reference grid X_{ref} in the finalization step (Step 3 of Algorithm 1), twenty values per input dimension, equidistantly distributed from the lower to the upper bound along each direction, were used.

Algorithm 1 was implemented in Python and the package `sklearn` was used for the numerical representation of the models. Since the discretized problems (11) are standard

convex quadratic programs, a solver tailored to that specific problem class was used, namely, quadprog [41]. It can solve quadratic programs with hundreds of variables and thousands of constraints in just a few seconds because it efficiently exploits the simple structure of the problem. Since, on the other hand, the lower-level problems (12) are global optimization problems with possibly several local minima, a suitable global optimization solver was required. We chose the solver `scipy.optimize.shgo` [42], which employs a simplicial homology strategy, and which, in our applications, turned out to be a good compromise between speed and reliability. For the problems considered in this article, `shgo`'s internal local optimization was configured to occur in every iteration, to multi-start from a Sobol set of $100 \cdot d$ points and to be executed using the algorithm L-BFGS-B with analytical gradients.

3. Applications in Manufacturing

In this section, two real-world manufacturing applications are described to which our monotonic regression algorithm was applied.

3.1. Laser Glass Bending

A first application example is laser glass bending. In the industrial standard process of glass bending [43], a flat glass specimen is placed in a furnace with furnace temperature T_f , and then the heated specimen is bent at a designated edge driven by gravity. As an additional feature, a laser can be added to the industrial standard process in order to specifically heat up the critical region of the flat glass around the bending edge, and thus to speed up the process and achieve smaller bending radii [44,45]. The laser can generally scribe in arbitrary directions. In the process considered here, however, the laser path is restricted to three straight lines parallel to the bending edge. While the middle line defines the bending edge, the two outer lines are at a fixed distance $\Delta_1/2 = 5.75$ mm in each direction to it. The laser spot moves along this path in multiple cycles with the number of cycles denoted by n_c . The scribing speed and the power of the laser are kept constant. A mechanical stop below the bending edge guarantees that the bending angle does not exceed 90° . An illustration of the laser glass bending process is shown in Figure 1.

The goal of the glass bending process considered here is to obtain bent glass parts with a prescribed bending angle. In order to achieve such a pre-defined bending angle, the process operator has to find suitable combinations of the two process parameters T_f and n_c , which is usually done based on experience. A more systematic approach, however, is to set up an appropriate model of the bending angle $y := \beta$ as a function of the process variables

$$x := (x_1, x_2) := (T_f, n_c) \in X, \quad (15)$$

where $X \subset \mathbb{R}^2$ is the rectangular set with the bounds specified in Table 1. In particular, such a model should allow sufficiently precise real-time predictions in order to support the process operator in quickly searching the parameter space X for optimal process parameter settings. Since SIAMOR models are polynomial by construction, they perfectly satisfy this real-time requirement. In contrast, a repeated evaluation of finite-element simulation models of the glass-bending process is too time-consuming to be of any practical use in quickly exploring the parameter space.

As generating experimental training data from the real process is cumbersome, a two-dimensional finite-element model was set up to generate data numerically. The simulation of the process was based on a coupled thermo-mechanical problem with finite deformation. Since the CO_2 laser used in the process operates in the opaque wavelength spectrum of glass, the heat supply was modeled as a surface flux into the deforming sheet. In this two-dimensional setting, the heat was assumed to be deposited instantaneously along the thickness direction and also instantaneously on all three laser lines. Radiation effects were ignored, and heat conduction inside the glass was described by the classical Fourier law with the heat conductivity obtained experimentally via laser flash analysis. In view of the relevant relaxation and process time scales for the applied temperature range, the

mechanical behavior of the glass was described by a simple Maxwell-type visco-elastic law. The deformation due to gravity is heavily affected by the pronounced temperature dependence of the viscosity above the glass transition, which was described in our models using the Williams–Landel–Ferry approximation [46]. The generation of the simulated data was conducted using the commercial finite-element code Abaqus[®]. It was used to create a training dataset comprising 25 data points sampled on a 2D rectangular grid. The values used for the two degrees of freedom (five for T_f and five for n_c) were placed equidistantly from the lower to the upper bounds given in Table 1.

Within these ranges and for the laser configuration described above, process experts expect the following monotonicity behavior: the bending angle $y = \beta$ should increase monotonically with increasing glass temperature in the critical region, and thus with increasing T_f and n_c . In other words, the monotonicity signature σ of the bending angle y as a function of the inputs x from (15) is expected to be

$$\sigma = (\sigma_1, \sigma_2) = (1, 1). \tag{16}$$

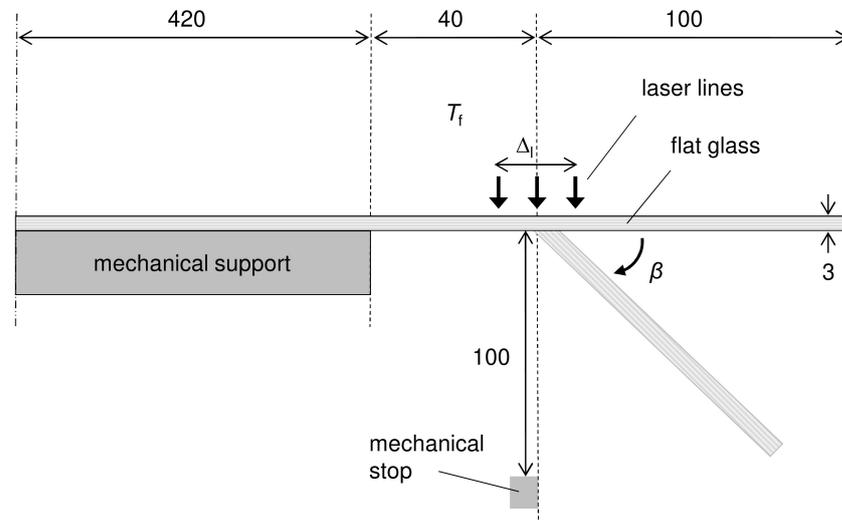


Figure 1. Side view of the laser glass bending process. Symbols: T_f —furnace temperature, Δ_l —distance between the left- and right-most laser line, β —bending angle. Lengths are given in mm.

Table 1. Ranges for the process variables of laser glass bending.

Variable	Min	Max	Phys. Unit
T_f	480	560	°C
n_c	40	50	—

3.2. Forming and Press Hardening of Sheet Metal

Another application example is press hardening [47]. In the experimental setup considered here, a blank is placed in a chamber furnace with a furnace temperature T_f above 900 °C. After heating the blank, an industrial robot transports it with handling time t_h into the cooled forming tool. In the following, the extra handling time $\Delta t_h = t_h - 10$ s is used instead, with 10 s being the minimum time the robot takes to move the blank from the furnace to the press. The final combined forming and quenching step allows for the variation of the press force F_p and the quenching time t_q . Afterwards, the formed part is transferred by the industrial robot to a deposition table for further cooling. An illustration of the process chain is shown in Figure 2.

The goal of the press hardening process considered in this work is to obtain a formed metal part with a prescribed hardness level, where the hardness is measured in units of the Vickers hardness number (unit symbol HV). In order to achieve such a pre-defined hardness, the process operator has to find suitable combinations of the four process parameters

$T_f, \Delta t_h, F_p, t_q$. And for that purpose, in turn, an appropriate model is needed for the hardness y of the formed part (at distinguished measurement points on the surface of the part) as a function of the process variables

$$x := (x_1, \dots, x_4) := (T_f, \Delta t_h, F_p, t_q) \in X, \tag{17}$$

where $X \subset \mathbb{R}^4$ is the hypercuboid set with the bounds specified in Table 2. In particular, such a model has to allow sufficiently accurate real-time predictions in order to help the process operator in quickly searching the parameter space X for optimal process parameter settings. Since SIAMOR models are polynomial by construction, they perfectly satisfy this real-time requirement. In contrast, due to the four-dimensional parameter space, already a single evaluation of a representative finite-element simulation model of the press-hardening process is prohibitively time-consuming to be of any practical use in quickly exploring the parameter space.

As in the case of glass bending, experiments for the press hardening process are expensive because they usually require manual adjustments, which tend to be time-consuming. Additionally, the local hardness measurements at the chosen measurement points on the surface of the quenched part are time-consuming as well. This is why the training data base we used is rather small. It contains 60 points resulting from a design of experiments with the four process variables $T_f, \Delta t_h, F_p$ and t_q ranging between the bounds in Table 2, along with the corresponding hardness values at six local measurement points (referred to as MP1, . . . , MP6 in the following).

In order to compensate this data shortage, expert knowledge is brought into play. An expert for press hardening expects the hardness to decrease monotonically with Δt_h and to increase monotonically with T_f as well as with t_q . In other words, the monotonicity signature σ of the hardness y (at any given measurement point) as a function of the inputs x from (17) is expected to be

$$\sigma = (\sigma_1, \dots, \sigma_4) = (1, -1, 0, 1). \tag{18}$$

In fact, a press hardening expert expects even a bit more, namely that the hardness should grow in a sigmoid-like manner with T_f and that it grows concavely towards saturation with increasing t_q . All these requirements result from qualitative physical considerations and are supported by empirical experience.

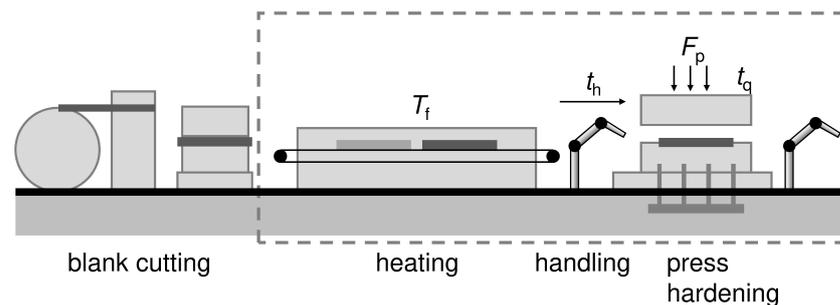


Figure 2. Side view of the press hardening process [47] indicating the considered process steps. Symbols: T_f —furnace temperature, t_h —handling time, F_p —press force, t_q —quenching time.

Table 2. Ranges for the process variables of press hardening.

Variable	Min	Max	Phys. Unit
T_f	871	933	°C
Δt_h	0	4	s
F_p	1750	2250	kN
t_q	2	6	s

4. Results and Discussion

In this section, we describe the results of our semi-infinite adaptive optimization approach to monotonic regression (SIAMOR) in the industrial processes described in Section 3, and compare them to the results of other approaches to incorporating monotonicity knowledge, which are well-known from the literature.

4.1. Informed Machine Learning Models for Laser Glass Bending

To begin with, the SIAMOR method was validated on a 1D subset of the data for laser glass bending, namely, the subset of all data points for which $n_c = 50$. This means that, out of the 25 data points, five points remained for training. First of all, ordinary unconstrained regression techniques were tried (see Figure 3a). A polynomial model of degree $m = 3$ (solid line) and a Gaussian process regressor [31] (GPR, dashed line) did not comply with the monotonicity knowledge at high T_f . A radial basis function (RBF) kernel was used for the GPR. This non-parametric model is always a reasonable choice for simulated data because it accurately reproduces the data themselves if the noise-level parameter is kept small. For all GPR models in this work, that parameter was set to 10^{-5} . Next, the polynomial model was regularized in a ridge regression (dotted line), where the squared ℓ^2 -norm $\lambda \|w\|_2^2$ with a regularization weight λ was added to the objective function in (4). $\lambda = 0.003$ was chosen, which was roughly the minimum necessary value to achieve monotonicity. However, the resulting model does not predict the data very well. Thus, all three models from Figure 3a were unsatisfactory.

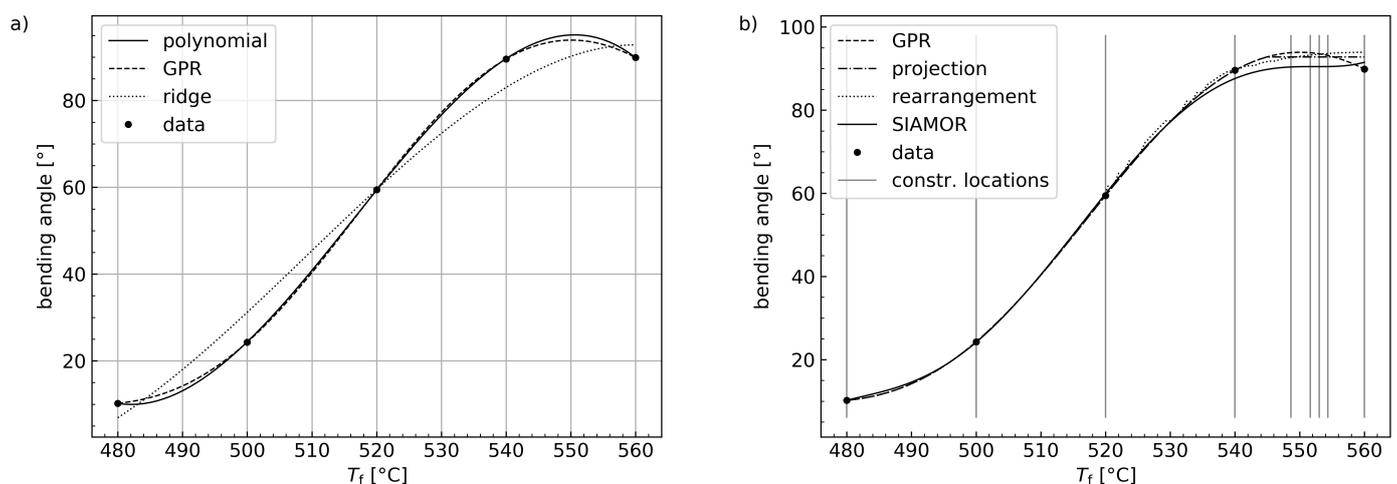


Figure 3. 1D regression for laser glass bending ($n_c = 50$). (a) Unconstrained regression. Solid: polynomial model ($m = 3$), dashed: Gaussian process regression (GPR) with RBF kernel (noise level 10^{-5}), dotted: polynomial ridge regression ($m = 3$, $\lambda = 0.003$). (b) Monotonic regression, with the solid line resulting from the SIAMOR method (see Sections 2.1–2.3) with degree $m = 5$. The projection [24] (dash-dotted) and rearrangement [22] (dotted) methods were fed with the dashed GPR curve as a non-monotonic reference predictor.

As a next step, the monotonicity requirement with respect to T_f was brought to bear, and monotonic regression with the SIAMOR method ($m = 5$) was used (see Figure 3b) and compared to the rearrangement [22] and to the monotonic projection [24] of the Gaussian process regressor from Figure 3a. As mentioned before, both comparative methods are based on a non-monotonic pre-trained reference predictor. This makes them fundamentally different to the SIAMOR method, which imposes the monotonicity already in the training phase. The projection was calculated as described in the Appendix A with $|G| = 80$ grid points. For the rearrangement method, the R package `monreg` was invoked from Python using the package `cpy2`. The degree m of the polynomial ansatz (1) used in the SIAMOR method was chosen as described in Section 2.3. For the specific case considered here, the curve started to vary unreasonably (albeit still monotonically) between the data points for

$m \geq 6$, and therefore $m = 5$ was chosen. The SIAMOR algorithm was initialized with five equidistant constraint locations in X^0 , and it converged in iteration 5 with a total number of nine constraints. The locations of the constraints are marked in Figure 3b by the gray, vertical lines. The adaptive algorithm automatically places the non-initial constraints in the non-monotonic region at high T_f . In terms of the root-mean-squared error

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{l=1}^N (\hat{y}(x_l) - t_l)^2} \quad (19)$$

on the training data, the SIAMOR model fits the data best; see Table 3. Another advantage of the SIAMOR model is that it is continuously differentiable, whereas the rearrangement and projection models exhibit (slight) unphysical kinks, which are typical for these methods [22]. And finally, the rearrangement and the projection models, for temperatures higher than 540 °C, both predict bending angles larger than 90°, which is unphysical due to the mechanical stop used in the glass bending process. In contrast, the predictions of the SIAMOR model do not (significantly) exceed 90°.

After these calculations on a 1D subset, the full 2D dataset of the considered laser glass bending process with its 25 data points was used. The results are shown in Figure 4. Again, part (a) of the figure displays an unconstrained Gaussian process regressor for comparison. The RBF kernel contained one length scale parameter per input dimension, and sklearn correctly adjusted these hyperparameters using L-BFGS-B. I.e., the employed length scales maximize the log-likelihood function of the model. Nevertheless, the model is unsatisfactory because it exhibits a bump in the rear right corner of the plot, contradicting the monotonicity knowledge.

Figure 4b shows the 2D monotonic projection of the GPR with the monotonicity requirements (16) with respect to T_f and n_c . It was calculated according to the Appendix A on a rectangular grid G consisting of 40^2 points (40 values per input dimension). The resulting model looks generally reasonable, and in particular, satisfies the monotonicity specifications, but it exhibits kinks and plateaus. The most conspicuous kink starts at about $T_f = 546$ °C, $n_c = 50$ and proceeds towards the front right. The rearrangement method by [23] was not used for comparison here because for small datasets in $d > 1$, it does not guarantee monotonicity.

Table 3. Root-mean-squared deviations (RMSE) of the monotonic regression models from the training data for laser glass bending (1D).

Monotonic Regression Type	RMSE [°]
projection [24]	1.3822
rearrangement [22]	1.8432
SIAMOR	1.1598

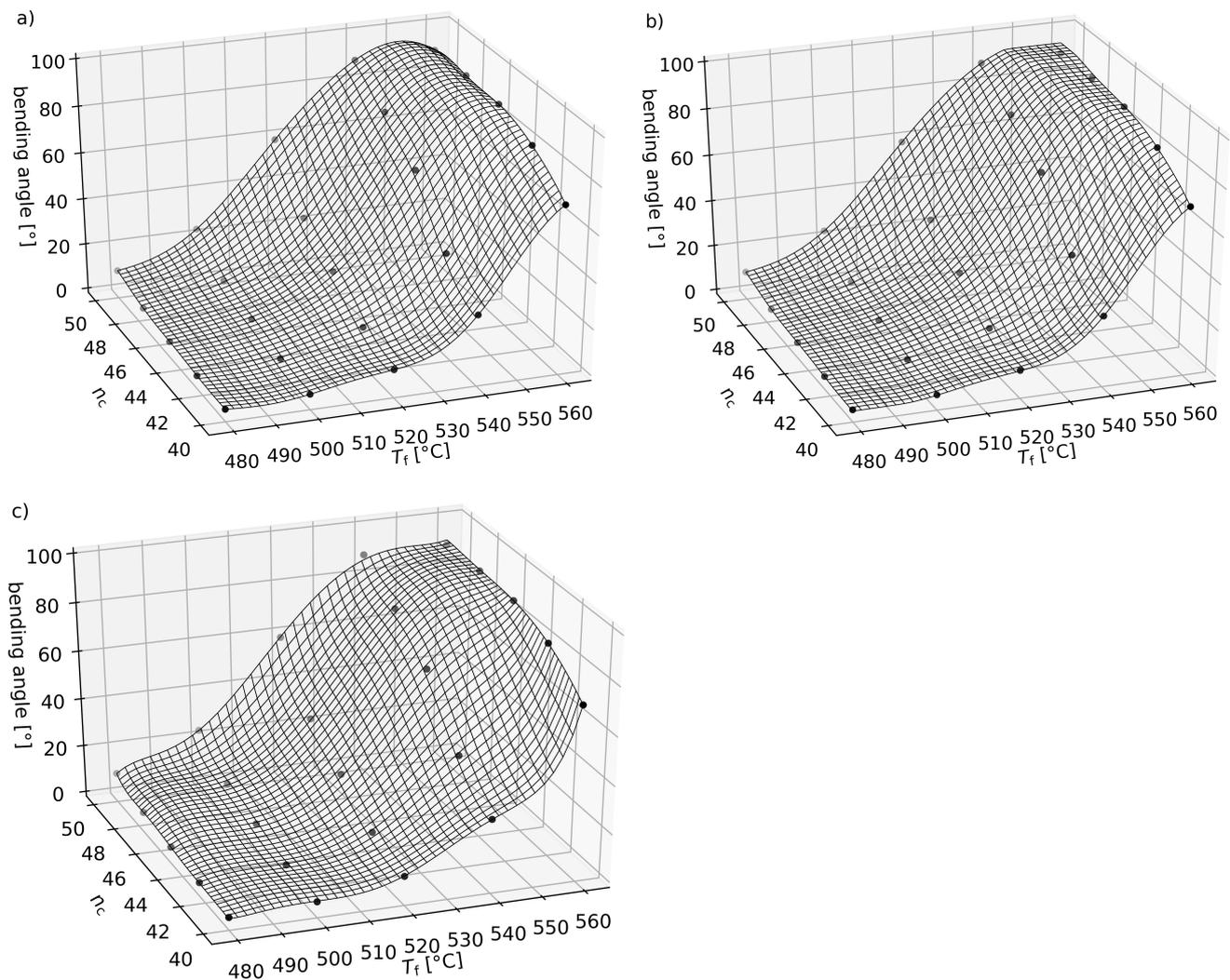


Figure 4. 2D regression for laser glass bending, where the markers represent the employed training data. (a) Gaussian process regression (non-monotonic) with a multi-length-scale RBF kernel (noise level 10^{-5}); (b) projection [24] of GPR; (c) monotonic regression of a polynomial model ($m = 7$) using the SIAMOR method (see Sections 2.1–2.3).

Figure 4c displays the corresponding response surface of a polynomial model of the form (1) with degree $m = 7$ trained with SIAMOR. For $m = 7$ there are $N_m = 36$ model parameters. The discretization X^0 was initialized with a rectangular grid using five equidistant values per dimension. The algorithm converged in iteration 11 with 69 final constraints. The resulting model is smoother than the one in Figure 4b, and it predicts the training data more accurately. Indeed, the corresponding RMSE values are 1.2518° for projection and 0.6607° for SIAMOR.

4.2. Informed Machine Learning Models for Forming and Press Hardening

As in the glass bending case, the SIAMOR method was first validated on a 1D subset of the data for the press hardening process. Namely, only those data points with $F_p = 2250$ kN, $\Delta t_h = 4$ s and $t_q = 2$ s were considered. These specifications are met by six data points, and these were used to train the models shown in Figure 5. The data are not monotonic due to experimental noise. However, they reflect the expected sigmoid-like behavior mentioned in Section 3.2, and this extends to the monotonized models. An unconstrained polynomial with $m = 3$ was chosen as the reference model to be monotonized for the comparative methods from the literature. Degrees lower than that resulted in larger deviations from the data and degrees higher than that resulted in overfitting. Thus, out of all models of the form (1), the hyperparameter choice $m = 3$ yielded the lowest RMSE values for

projection and rearrangement. For the monotonic regression with SIAMOR, $m = 6$ and five equidistant initial constraint locations in X^0 were chosen. It converged in iteration 8 with a total of 12 monotonicity constraints. In terms of the root-mean-squared error, the SIAMOR model predicts the training data more accurately, as can be seen in Table 4. The reason is that the rearrangement- and projection-based models are dragged away from the data by the underlying reference model, especially at high T_f .

Table 4. Root-mean-squared deviations (RMSE) of the monotonic regression models from the training data for forming and press hardening of sheet metal (1D).

Monotonic Regression Type	RMSE [HV]
projection [24]	5.0893
rearrangement [22]	4.8346
SIAMOR	3.3583

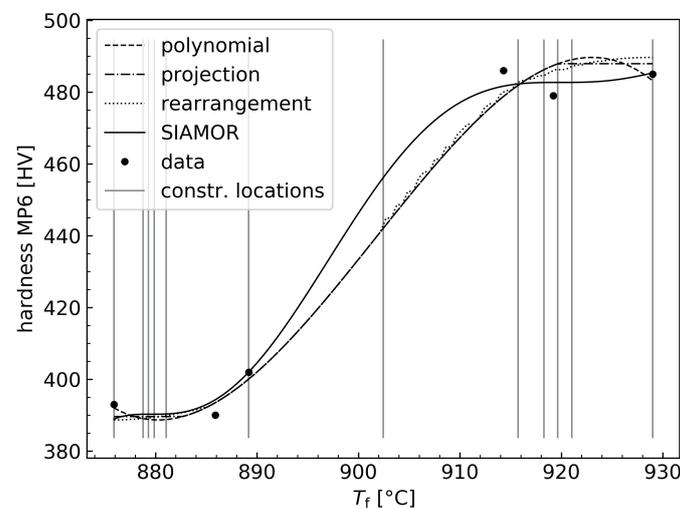


Figure 5. 1D regression for forming and press hardening of sheet metal ($F_p = 2250$ kN, $\Delta t_h = 4$ s, $t_q = 2$ s). Dashed: (non-monotonic) polynomial of degree $m = 3$ as the reference model, dash-dotted: projection [24], dotted: rearrangement [22], solid: SIAMOR (see Sections 2.1–2.3) with degree $m = 6$. The projection and rearrangement methods were fed with the dashed polynomial curve as non-monotonic reference predictor.

After these 1D considerations, the SIAMOR method was validated on the full 4D dataset of the press hardening process. Polynomial models with degree $m = 3$ were used for unconstrained regression and monotonic projection, and polynomials with $m = 6$ were used for the SIAMOR method. The resulting models are visualized in the surface plots in Figure 6. The unconstrained model from Figure 6a clearly shows non-monotonic predictions with respect to Δt_h . Furthermore, the hardness slightly decreases with the furnace temperature at T_f close to 930 °C, which is not the behavior expected by the process expert either. Figure 6b shows the monotonic projection of the unconstrained model. It was computed according to the Appendix A on a grid G consisting of 40^4 points. The monotonic projection exhibits the kinks that are characteristic of that method, and it yields an RMSE of 28.84 HV on the entire dataset.

With an overall RMSE of 10.14 HV, the model resulting from SIAMOR is more accurate for this application. A corresponding response surface is displayed in Figure 6c. In keeping with (18), monotonicity was required with respect to T_f (increasing), Δt_h (decreasing) and t_q (increasing). As $m = 6$, there are $N_m = 210$ model parameters and the discretization X^0 was initialized with a grid using four equidistant values per dimension. The algorithm converged in iteration 246 with 1372 final constraints. Our first try was with only two monotonicity requirements (namely, with respect to T_f and t_q). We observed, however,

that the final number of iterations decreased when the third monotonicity requirement was added. Thus, the monotonicity requirements in each direction promoted each other numerically within the algorithm and for the used data. This reduction in the number of iterations was not accompanied by a decrease in total calculation time because more lower-level problems have to be solved when there are more monotonicity directions.

With SIAMOR, monotonicity was achieved in all three input dimensions where it was required. See, e.g., Figure 6c, which is the monotonic counterpart of Figure 6a. A comparison of Figure 6a–c clearly shows how incorporating monotonicity expert knowledge helps compensate data shortages. Indeed, taking no monotonicity constraints into account at all (Figure 6a), we obtained an unexpected hardness minimum with respect to Δt_h at $\Delta t_h \approx 2.5$ s and small T_f . This also resulted in unnecessarily low predictions of the monotonic projection for small T_f and $\Delta t_h \gtrsim 1.5$ s in Figure 6b. The SIAMOR model (Figure 6c), by contrast, predicted more reasonable hardness values in this range without needing additional data, because it integrated the available monotonicity knowledge in the training phase.

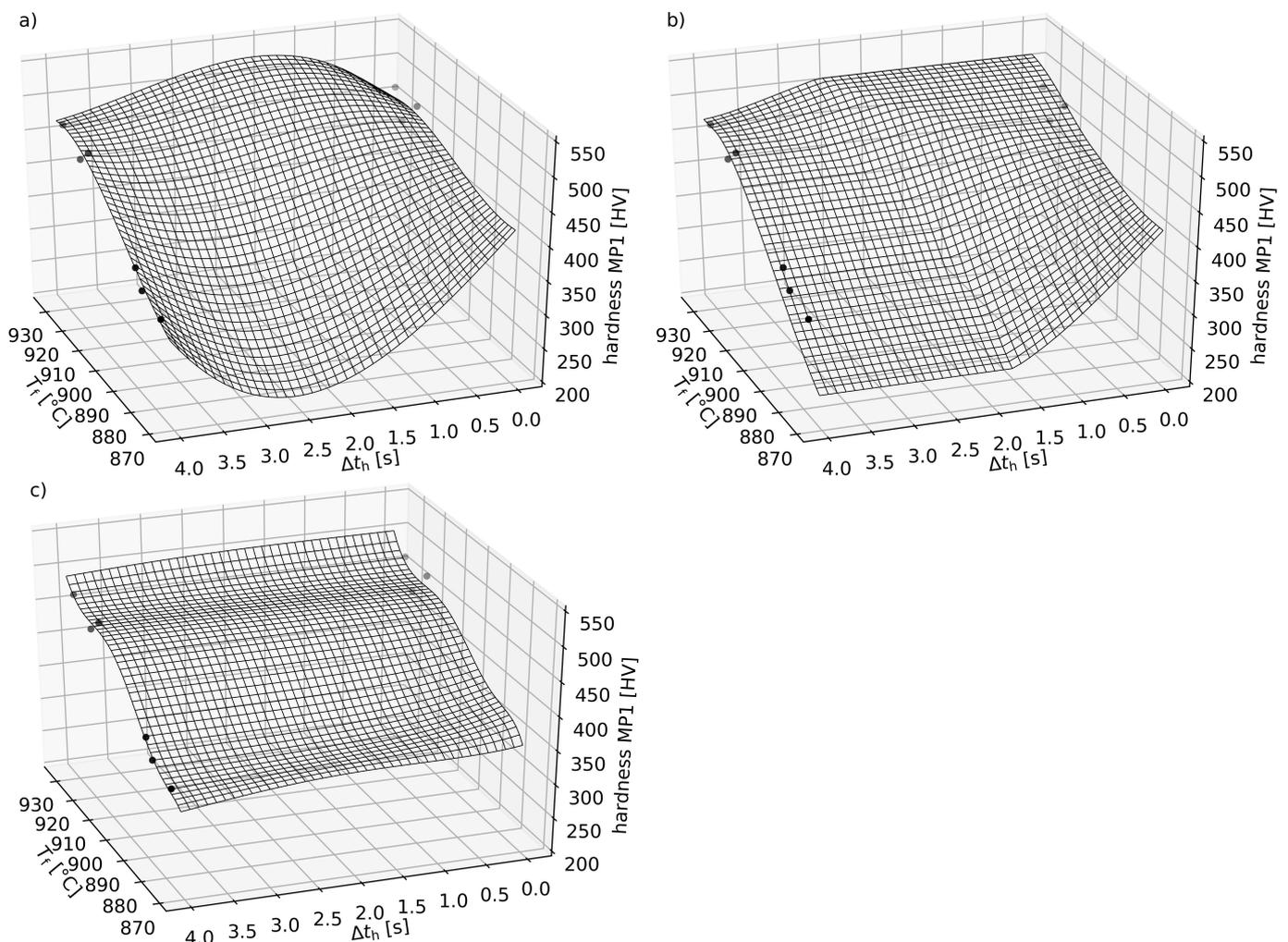


Figure 6. 4D regression for forming and press hardening of sheet metal using polynomial models ($F_p = 2250$ kN, $t_q = 2$ s). The markers represent those training points matching the specification of the corresponding plane in the input space. (a) Unconstrained $m = 3$, (b) projection [24] of unconstrained $m = 3$, (c) SIAMOR $m = 6$.

For the SIAMOR plots in Figure 7, Δt_h was reduced to 0 s. This figure shows that monotonicity is also achieved with respect to t_q . Without having explicitly demanded it, the hardness y shows the expected concave growth towards saturation with respect to t_q in Figure 7a. An additional increase in F_p leads to Figure 7b, where the sign of the second

derivative of y with respect to the quenching time t_q changes along the T_f -axis. I.e., the model changes its convexity properties in this direction and increases convexly instead of concavely with t_q at high T_f , $\Delta t_h = 0$ s and $F_p = 2250$ kN. This contradicts the process expert's expectations. A possible way out is to measure additional data (e.g., in the rear left corner of Figure 7b), which is elaborate and costly, however. Another possible way out is to add the concavity requirement $\partial_{x_4}^2 \hat{y}_w(\mathbf{x}) \leq 0$ for all $\mathbf{x} \in X$ with respect to the $x_4 = t_q$ direction to the monotonicity constraints (6) used exclusively so far. In order to solve the resulting constrained regression problem, one can use the same adaptive semi-infinite solution strategy, which was already used for the monotonicity constraints alone.

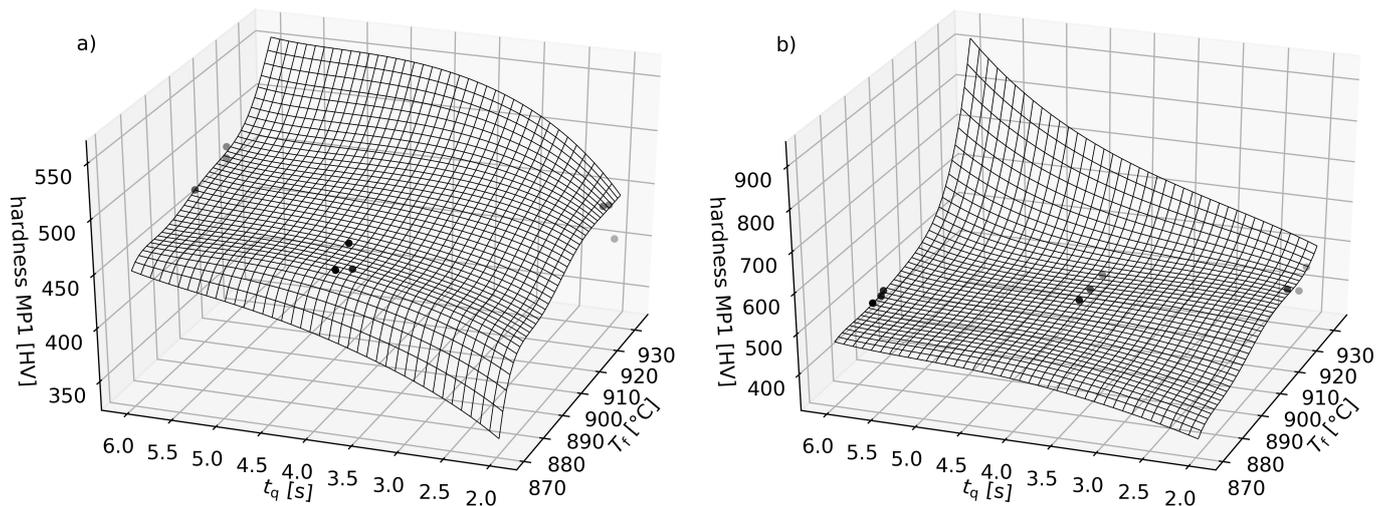


Figure 7. Response surfaces of 4D monotonic regression with SIAMOR ($m = 6$) for forming and press hardening of sheet metal ($\Delta t_h = 0$ s). The markers represent those training points matching the specifications of the corresponding planes in the input space. (a) $F_p = 1750$ kN, (b) $F_p = 2250$ kN.

5. Conclusions and Outlook

In this article, a proof of concept was conducted for the method of semi-infinite optimization with an adaptive discretization scheme to solve monotonic regression problems (SIAMOR). The method generates continuously differentiable models, and its use in multiple dimensions is straightforward. Polynomial models were used, but the method is not restricted to this type of model, even though it is numerically favorable because polynomial models lead to convex quadratic discretized problems. The monotonic regression technique was validated by means of two real-world applications from manufacturing. It resulted in predictions that complied very well with expert knowledge and that compensated for the lack of data to a certain extent. At least for the small datasets considered here, the resulting models predicted the training data more accurately than models based on the well-known projection or rearrangement methods from the literature.

While the present article is confined to regression under monotonicity constraints, semi-infinite optimization can also be exploited to treat other types of shape constraints such as concavity constraints, for instance. In fact, the shape constraints can be quite arbitrary, in principle. Additionally, this is only one of several aspects in the field of potential research on the method opened up by this work. Others are the testing of SIAMOR in combination with different model types, datasets or industrial processes. When using Gaussian process regressors instead of the polynomial models employed here, one can try out and compare various kernel types. Additionally, the SIAMOR method can be extended to locally varying monotonicity requirements (i.e., $\sigma_j = \sigma_j(\mathbf{x})$).

Another possible direction of future research is to systematically investigate how to speed up the solution of the global lower-level problems. When more complex models or shape constraints are used, this will be particularly important. The solution of multiple lower-level problems and the final feasibility test on the reference grid can be

parallelized to reduce the calculation time, for example. A rigorous investigation of the convergence properties and the asymptotic properties of the SIAMOR method and its possible generalizations is left to future research as well.

Author Contributions: Conceptualization, M.v.K., J.S. (Jochen Schmid), P.L. and A.S.; methodology, M.v.K., J.S. (Jochen Schmid) and J.S. (Jan Schwientek); software, M.v.K. and J.S. (Jochen Schmid); validation, M.v.K., J.S. (Jochen Schmid), P.L., R.Z., L.M., J.S. (Jan Schwientek) and A.S.; formal analysis, M.v.K., J.S. (Jochen Schmid) and J.S. (Jan Schwientek); investigation, M.v.K., J.S. (Jochen Schmid), P.L., R.Z., L.M. and A.S.; resources, P.L., L.M., I.S., T.K. and A.S.; data curation, P.L., L.M., I.S., T.K. and A.S.; writing—original draft preparation, M.v.K., J.S. (Jochen Schmid), P.L., R.Z. and I.S.; writing—review and editing, M.v.K., J.S. (Jochen Schmid), P.L., R.Z., L.M., I.S. and A.S.; visualization, M.v.K., P.L., R.Z. and L.M.; supervision, A.S. and T.K.; project administration, A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fraunhofer Society within the lighthouse project “Machine Learning for Production” (ML4P).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not available.

Acknowledgments: We would like to thank Michael Bortz and Raoul Heese for valuable discussions about integrating expert knowledge into machine learning models. We also gratefully acknowledge funding from the Fraunhofer Society within the lighthouse project “Machine Learning for Production” (ML4P). And finally, we would like to thank the anonymous reviewers for their valuable comments.

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

Abbreviations

The following abbreviations are used in this manuscript:

SIAMOR	Semi-infinite optimization approach to monotonic regression
GPR	Gaussian process regression
RBF	Radial basis function
RMSE	Root mean squared error

Appendix A. Computing Monotonic Projections

In order to validate our semi-infinite optimization approach to monotonic regression, it is compared, among other things, to the projection-based monotonicization approach by [24]. As has been pointed out in Section 1, the projection method starts out from a purely data-based initial model \hat{y}^0 (a Gaussian process regressor in the case of [24]) and then replaces this initial model by the monotonic projection \hat{y} of \hat{y}^0 . I.e., $\hat{y} : X \rightarrow \mathbb{R}$ is the monotonic square-integrable function with monotonicity signature σ that is closest to \hat{y}^0 in the L^2 -norm.

In order to numerically compute this monotonic projection \hat{y} , the original procedure proposed in [24] is not used here, though. Instead, the conceptually and computationally simpler methodology from [48] is employed. In this methodology, the input space X is discretized with a fine rectangular grid G . Then, the corresponding discrete monotonic projection $(\hat{y}(x))_{x \in G}$, that is, the solution of the constrained optimization problem

$$\min_{z \in \mathbb{R}^G} \sum_{x \in G} (z(x) - \hat{y}^0(x))^2 \quad \text{s.t.} \quad \sigma_j \cdot (z(x + h_j e_j) - z(x)) \geq 0 \quad \text{for all } j \in J \quad (\text{A1})$$

and all $x \in G$ for which $x + h_j e_j \in G$

is computed. In the above relation, \mathbb{R}^G is the $|G|$ -dimensional vector space of all \mathbb{R} -valued functions $z = (z(x))_{x \in G}$ defined on the discrete set G , $h_j > 0$ indicates the distance of adjacent grid points in the j th coordinate direction, and $e_j \in \mathbb{R}^d$ is the j th canonical unit

vector. It is shown in [48] that the extension of $(\hat{y}(x))_{x \in G}$ to a grid-constant function on the whole of X is a good approximation of the monotonic projection \hat{y} , if only the grid is fine enough and the initial model \hat{y}^0 is continuous, for instance. In contrast to [24], these approximation results from [48] also feature rates of convergence.

Since both the objective function and the constraints of (A1) are convex with respect to z , the problem (A1) is a convex program. We used cvxopt [49] to solve these problems because it offers a sparse matrix type to represent the large coefficient and constraint matrices for $d > 1$. Alternatively, the discrete monotonic projection problems can also be solved using any of the more sophisticated computational methods from ([50], Section 2.3), ([51], Section 4.1), or [52–56]. However, for the number of input dimensions considered here, our direct computational method is sufficient.

References

- Weichert, D.; Link, P.; Stoll, A.; Rüping, S.; Ihlenfeldt, S.; Wrobel, S. A review of machine learning for the optimization of production processes. *Int. J. Adv. Manuf. Technol.* **2019**, *104*, 1889–1902. [CrossRef]
- MacInnes, J.; Santosa, S.; Wright, W. Visual classification: Expert knowledge guides machine learning. *IEEE Comput. Graph. Appl.* **2010**, *30*, 8–14. [CrossRef]
- Heese, R.; Walczak, M.; Morand, L.; Helm, D.; Bortz, M. The Good, the Bad and the Ugly: Augmenting a Black-Box Model with Expert Knowledge. In *Artificial Neural Networks and Machine Learning—ICANN 2019: Workshop and Special Sessions; Lecture Notes in Computer Science*; Tetko, I.V., Kůrková, V., Karpov, P., Theis, F., Eds.; Springer International Publishing: Cham, Switzerland, 2019; Volume 11731, pp. 391–395. [CrossRef]
- Rueden, L.V.; Mayer, S.; Beckh, K.; Georgiev, B.; Giesselbach, S.; Heese, R.; Kirsch, B.; Pfrommer, J.; Pick, A.; Ramamurthy, R.; et al. Informed Machine Learning—A Taxonomy and Survey of Integrating Knowledge into Learning Systems. *IEEE Trans. Knowl. Data Eng.* **2021**. [CrossRef]
- Johansen, T.A. Identification of non-linear systems using empirical data and prior knowledge—An optimization approach. *Automatica* **1996**, *32*, 337–356. [CrossRef]
- Mangasarian, O.L.; Wild, E.W. Nonlinear knowledge in kernel approximation. *IEEE Trans. Neural Netw.* **2007**, *18*, 300–306. [CrossRef]
- Mangasarian, O.L.; Wild, E.W. Nonlinear knowledge-based classification. *IEEE Trans. Neural Netw.* **2008**, *19*, 1826–1832. [CrossRef] [PubMed]
- Cozad, A.; Sahinidis, N.V.; Miller, D.C. A combined first-principles and data-driven approach to model building. *Comput. Chem. Eng.* **2015**, *73*, 116–127. [CrossRef]
- Wilson, Z.T.; Sahinidis, N.V. The ALAMO approach to machine learning. *Comput. Chem. Eng.* **2017**, *106*, 785–795. [CrossRef]
- Wilson, Z.T.; Sahinidis, N.V. Automated learning of chemical reaction networks. *Comput. Chem. Eng.* **2019**, *127*, 88–98. [CrossRef]
- Asprion, N.; Böttcher, R.; Pack, R.; Stavrou, M.E.; Höller, J.; Schwientek, J.; Bortz, M. Gray-Box Modeling for the Optimization of Chemical Processes. *Chem. Ing. Tech.* **2019**, *91*, 305–313. [CrossRef]
- Heese, R.; Nies, J.; Bortz, M. Some Aspects of Combining Data and Models in Process Engineering. *Chem. Ing. Tech.* **2020**, *92*, 856–866. [CrossRef]
- Altendorf, E.E.; Restificar, A.C.; Dietterich, T.G. Learning from Sparse Data by Exploiting Monotonicity Constraints. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence, UAI'05, Edinburgh, UK, 26–29 July 2005*; AUAI Press: Arlington, VA, USA, 2005; pp. 18–26.
- Kotłowski, W.; Słowiński, R. Rule learning with monotonicity constraints. In *Proceedings of the 26th Annual International Conference on Machine Learning—ICML'09, Montreal Quebec Canada, 14–18 June 2009*; Danyluk, A., Bottou, L., Littman, M., Eds.; ACM Press: New York, NY, USA, 2009; pp. 1–8.
- Lauer, F.; Bloch, G. Incorporating prior knowledge in support vector machines for classification: A review. *Neurocomputing* **2008**, *71*, 1578–1594. [CrossRef]
- Groeneboom, P.; Jongbloed, G. *Nonparametric Estimation under Shape Constraints*; Cambridge University Press: Cambridge, UK, 2014; doi:10.1017/CBO9781139020893. [CrossRef]
- Gupta, M.; Cotter, A.; Pfeifer, J.; Voevodski, K.; Canini, K.; Mangylov, A.; Moczydlowski, W.; van Esbroeck, A. Monotonic Calibrated Interpolated Look-Up Tables. *J. Mach. Learn. Res. (JMLR)* **2016**, *17*, 1–47.
- Mukerjee, H. Monotone Nonparametric Regression. *Ann. Stat.* **1988**, *16*, 741–750. [CrossRef]
- Mammen, E. Estimating a smooth monotone regression function. *Ann. Stat.* **1991**, *19*, 724–740. [CrossRef]
- Mammen, E.; Marron, J.S.; Turlach, B.A.; Wand, M.P. A General Projection Framework for Constrained Smoothing. *Stat. Sci.* **2001**, *16*, 232–248. [CrossRef]
- Hall, P.; Huang, L.S. Nonparametric kernel regression subject to monotonicity constraints. *Ann. Stat.* **2001**, *29*, 624–647. [CrossRef]
- Detle, H.; Neumeyer, N.; Pilz, K.F. A simple nonparametric estimator of a strictly monotone regression function. *Bernoulli* **2006**, *12*, 469–490. [CrossRef]

23. Dette, H.; Scheder, R. Strictly monotone and smooth nonparametric regression for two or more variables. *Can. J. Stat.* **2006**, *34*, 535–561. [[CrossRef](#)]
24. Lin, L.; Dunson, D.B. Bayesian monotone regression using Gaussian process projection. *Biometrika* **2014**, *101*, 303–317. [[CrossRef](#)]
25. Chernozhukov, V.; Fernandez-Val, I.; Galichon, A. Improving point and interval estimators of monotone functions by rearrangement. *Biometrika* **2009**, *96*, 559–575. [[CrossRef](#)]
26. Lauer, F.; Bloch, G. Incorporating prior knowledge in support vector regression. *Mach. Learn.* **2007**, *70*, 89–118. [[CrossRef](#)]
27. Chuang, H.C.; Chen, C.C.; Li, S.T. Incorporating monotonic domain knowledge in support vector learning for data mining regression problems. *Neural Comput. Appl.* **2020**, *32*, 11791–11805. [[CrossRef](#)]
28. Riihimäki, J.; Vehtari, A. Gaussian processes with monotonicity information. *Proc. Mach. Learn. Res.* **2010**, *9*, 645–652.
29. Neumann, K.; Rolf, M.; Steil, J.J. Reliable integration of continuous constraints into extreme learning machines. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2013**, *21*, 35–50. [[CrossRef](#)]
30. Friedlander, F.G.; Joshi, M.S. *Introduction to the Theory of Distributions*, 2nd ed.; Cambridge University Press: Cambridge, UK, 1998.
31. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; Adaptive Computation and Machine Learning; MIT: Cambridge, MA, USA; London, UK, 2006.
32. Hettich, R.; Zencke, P. *Numerische Methoden der Approximation und Semi-Infiniten Optimierung*; Teubner Studienbücher: Mathematik; Teubner: Stuttgart, Germany, 1982.
33. Polak, E. *Optimization: Algorithms and Consistent Approximations*; Applied Mathematical Sciences; Springer: New York, NY, USA; London, UK, 1997; Volume 124.
34. Reemtsen, R.; Rückmann, J.J. *Semi-Infinite Programming*; Nonconvex Optimization and Its Applications; Kluwer Academic: Boston, MA, USA; London, UK, 1998; Volume 25.
35. Stein, O. *Bi-Level Strategies in Semi-Infinite Programming*; Nonconvex Optimization and Its Applications; Kluwer Academic: Boston, MA, USA; London, UK, 2003; Volume 71.
36. Stein, O. How to solve a semi-infinite optimization problem. *Eur. J. Oper. Res.* **2012**, *223*, 312–320. [[CrossRef](#)]
37. Shimizu, K.; Ishizuka, Y.; Bard, J.F. *Nondifferentiable and Two-Level Mathematical Programming*; Kluwer Academic Publishers: Boston, MA, USA; London, UK, 1997.
38. Dempe, S.; Kalashnikov, V.; Pérez-Valdés, G.A.; Kalashnykova, N. *Bilevel Programming Problems*; Springer: Berlin/Heidelberg, Germany, 2015; doi:10.1007/978-3-662-45827-3. [[CrossRef](#)]
39. Blankenship, J.W.; Falk, J.E. Infinitely constrained optimization problems. *J. Optim. Theory Appl.* **1976**, *19*, 261–281. [[CrossRef](#)]
40. Nocedal, J.; Wright, S.J. *Numerical Optimization*, 2nd ed.; Springer Series in Operations Research; Springer: New York, NY, USA, 2006.
41. Goldfarb, D.; Idnani, A. A numerically stable dual method for solving strictly convex quadratic programs. *Math. Program.* **1983**, *27*, 1–33. [[CrossRef](#)]
42. Endres, S.C.; Sandrock, C.; Focke, W.W. A simplicial homology algorithm for Lipschitz optimisation. *J. Glob. Optim.* **2018**, *72*, 181–217. [[CrossRef](#)]
43. Neugebauer, J. Applications for curved glass in buildings. *J. Facade Des. Eng.* **2014**, *2*, 67–83. [[CrossRef](#)]
44. Rist, T.; Gremmelspacher, M.; Baab, A. Feasibility of bent glasses with small bending radii. *CE/Papers* **2018**, *2*, 183–189. [[CrossRef](#)]
45. Rist, T.; Gremmelspacher, M.; Baab, A. Innovative Glass Bending Technology for Manufacturing Expressive Shaped Glasses with Sharp Curves. *Glass Performance Days*. 2019; pp. 34–35. Available online: <https://www.glassonweb.com/article/innovative-glass-bending-technology-manufacturing-expressive-shaped-glasses-with-sharp> (accessed on 26 November 2021)
46. Williams, M.L.; Landel, R.F.; Ferry, J.D. The Temperature Dependence of Relaxation Mechanisms in Amorphous Polymers and Other Glass-forming Liquids. *J. Am. Chem. Soc.* **1955**, *77*, 3701–3707. [[CrossRef](#)]
47. Neugebauer, R.; Schieck, E.; Polster, S.; Mosel, A.; Rautenstrauch, A.; Schönherr, J.; Pierschel, N. Press hardening—An innovative and challenging technology. *Arch. Civ. Mech. Eng.* **2012**, *12*, 113–118. [[CrossRef](#)]
48. Schmid, J. Approximation, characterization, and continuity of multivariate monotonic regression functions. *Anal. Appl.* **2021**. [[CrossRef](#)]
49. Andersen, M.; Dahl, J.; Liu, Z.; Vandenberghe, L. Interior-point methods for large-scale cone programming. In *Optimization for Machine Learning*; Sra, S., Nowozin, S., Wright, S.J., Eds.; MIT Press: Cambridge, MA, USA, 2011; pp. 55–83.
50. Barlow, R.E. *Statistical Inference under Order Restrictions: The Theory and Application of Isotonic Regression*; Wiley Series in Probability and Mathematical Statistics; Wiley: London, UK; New York, NY, USA, 1972; Volume 8.
51. Robertson, T.; Wright, F.T.; Dykstra, R. *Statistical Inference under Inequality Constraints*; Wiley Series in Probability and Mathematical Statistics. Probability and Mathematical Statistics Section; Wiley: Chichester, UK; New York, NY, USA, 1988.
52. Qian, S.; Eddy, W.F. An Algorithm for Isotonic Regression on Ordered Rectangular Grids. *J. Comput. Graph. Stat.* **1996**, *5*, 225–235. [[CrossRef](#)]
53. Spouge, J.; Wan, H.; Wilbur, W.J. Least Squares Isotonic Regression in Two Dimensions. *J. Optim. Theory Appl.* **2003**, *117*, 585–605. [[CrossRef](#)]
54. Stout, Q.F. Isotonic Regression via Partitioning. *Algorithmica* **2013**, *66*, 93–112. [[CrossRef](#)]
55. Stout, Q.F. Isotonic Regression for Multiple Independent Variables. *Algorithmica* **2015**, *71*, 450–470. [[CrossRef](#)]
56. Kyng, R.; Rao, A.; Sachdeva, S. Fast, Provable Algorithms for Isotonic Regression in all l_p -norms. In *Advances in Neural Information Processing Systems 28*; Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA 2015; pp. 2719–2727.