Les Cahiers du GERAD ISSN: 0711–2440

Locally weighted regression models for surrogate-assisted design optimization

B. Talgorn, C. Audet,
S. Le Digabel, M. Kokkolaras

G–2016–113

November 2016

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Locally weighted regression models for surrogate-assisted design optimization

Bastien Talgorn [a]

Charles Audet [b]

Sébastien Le Digabel [b]

Michael Kokkolaras [a]

[a] *GERAD & Department of Mechanical Engineering, McGill University, Montréal (Québec) Canada*

[b] *GERAD & Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montréal (Québec) Canada*

bastien.talgorn@mail.mcgill.ca
charles.audet@gerad.ca
sebastien.le.digabel@gerad.ca
michael.kokkolaras@mcgill.ca

November 2016

Les Cahiers du GERAD

G–2016–113

Abstract: Locally weighted regression combines the advantages of polynomial regression and kernel smoothing. We present three ideas for appropriate and effective use of LOcally WEighted Scatterplot Smoothing (LOWESS) models for surrogate optimization. First, a method is proposed to reduce the computational cost of LOWESS models. Second, a local scaling coefficient is introduced to adapt LOWESS models to the density of neighboring points while retaining smoothness. Finally, an appropriate order error metric is used to select the optimal shape coefficient of the LOWESS model. Our surrogate-assisted optimization method relies on the the Mesh Adaptive Direct Search (MADS) algorithm in which LOWESS models are used to generate and rank promising candidates. The blackbox functions governing the optimization problem are then evaluated at these ranked candidates with an opportunistic strategy, thus minimizing CPU time. Extensive computational results are reported for three engineering design problems. These results demonstrate the effectiveness of the LOWESS models as well as the order error metric for surrogate-assisted optimization.

Keywords: Local regression, order error, surrogate models, derivative-free optimization, Mesh Adaptive Direct Search (MADS)

Acknowledgments: All authors acknowledge the partial support of FRQNT grant PR-182098; B. Talgorn and M. Kokkolaras are also grateful for the partial support of NSERC/Hydro-Quebec grant EGP2 498903-16; such support does not constitute an endorsement by the sponsors of the opinions expressed in this article. B. Talgorn would like to thank Stéphane Alarie of IREQ, Patricia Gillett-Kawamoto of GERAD and Sylvain Arreckx of GERAD for their invaluable insights and comments.

1 Introduction

We consider the optimization problem

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathcal{X}} & f(\mathbf{x}) \\ \text{subject to} & c_j(\mathbf{x}) \leq 0, \quad j = 1, 2, ..., m, \end{array} \tag{P}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the vector of design optimization variables and $\mathcal{X}$ is a subset of $\mathbb{R}^n$ possibly unbounded defined by some box constraints. At least one (but typically many) of the functions $\{f, c_1, \ldots, c_m\}$ is evaluated using computational procedures known as blackboxes. Functions evaluated using blackboxes may be nonsmooth and nonconvex, and their derivatives (if they exist) may be very hard to approximate in a computationally practical and dependable manner. Moreover, a blackbox evaluation is often expensive in terms of CPU time, may crash or fail to return a meaningful value.

1.1 Surrogate-assisted optimization

This work focuses on solving the blackbox optimization problem (P) using a surrogate-assisted approach. When the blackbox output is known at p different data points $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_p\} \subset \mathcal{X}$, it is possible to build surrogate models $\{\hat{f}, \hat{c}_1, \ldots, \hat{c}_m\}$ of the objective and the constraints to estimate their value for a design $\mathbf{x} \notin \mathbf{X}$. In this document, we use single and double hats to denote surrogate functions and cross-validation values, respectively.

We then generate a candidate solution to Problem (P) by solving the *surrogate* problem

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathcal{X}} & \hat{f}(\mathbf{x}) \\ \text{subject to} & \hat{c}_j(\mathbf{x}) \leq 0 \quad \forall j \in 1, 2, ..., m. \end{array} \tag{$\hat{P}$}$$

If surrogates provide an accurate representation of the blackbox output, then a minimizer of the surrogate problem ($\hat{P}$) will be a good candidate for the solution of the main problem (P). This leads to two questions: 1) how to quantify the accuracy of a surrogate model in the context of surrogate assisted optimization, and 2) which surrogate modeling technique to use? Section 3 proposes a metric to answer the first question, and Section 4 gives numerical evidence to answer the second question.

1.2 Locally weighted regression

In previous work, we investigated the suitability of Polynomial Response Surfaces (PRSs) [1, 33, 36] and Kernel Smoothing (KS) models [1, 22] to surrogate optimization [5]. We concluded that, while useful, these models have certain limitations: PRSs do not fit data of underlying multimodal functions well, even when a large number of data points is available. KS models have a tendency to undershoot because predictions are always bounded by observations.

In this work, we consider LOcally WEighted Scatterplot Smoothing (LOWESS) [9, 10, 11, 12] models, which constitute a combined generalization of PRSs and KS models. LOWESS builds local polynomial models that emphasize the data points in the vicinity of the point $\boldsymbol{\xi} \in \mathcal{X}$ where the blackbox functions f and $\{c_j\}_{j=1...m}$ are to be estimated by assigning larger weights to data points close to it than to points that are further away from it. Specifically, the weight of a data point $\mathbf{x} \in \mathbf{X}$ in a local regression model will be $\phi(\lambda\|\mathbf{x} - \boldsymbol{\xi}\|_2/d_q(\boldsymbol{\xi}))$, where ϕ is a kernel function, $\lambda > 0$ is a user-defined shape coefficient and $d_q(\boldsymbol{\xi})$ is a scaling distance that takes into account the density of data points in the neighborhood of $\boldsymbol{\xi}$.

Locally weighted regression was first used as early as 1931 for fitting time series [26]. In 1979, Cleveland [9] introduced locally weighted local regression for univariate data, and presented an iterative method for robust regression. This method was then made available as a FORTRAN code [10]. The equivalence between LOWESS models and kernel smoothing when the number of observations tends toward infinity was established in [32]. The method was then generalized to multivariate data and applied to data exploration and diagnostic checking [11, 12]. The detailed and asymptotic properties of these models are investigated in [37, 25, 17, 16], and an overview is given in [34]. The method was then implemented in the SAS/STAT software under the

name LOESS [13], applied to hydrological forecast in the Great Salt Lake [24] and to robotic control learning algorithms [39].

In preliminary studies, we observed that LOWESS models generate promising results in surrogate assisted optimization. However, their main drawback is that they require the solution of a linear system for each model prediction. In particular, if the local polynomial regression is quadratic, then one prediction in $\boldsymbol{\xi} \in \mathbb{R}^n$ requires solving a linear system of size $(n+1)(n+2)/2$, which has a computational cost of $\mathcal{O}(n^4)$. Therefore, LOWESS models are only practical for small-size design optimization problems, or when blackboxes are exceptionally expensive to exercise.

The first novel element of this work is a computationally efficient method for solving the aforementioned linear system: the computation cost is made independent from the number of blackbox outputs and is reduced by using information available from previous model predictions and by an educated choice between an iterative or direct solution method.

The second novel element of this work is the use of a statistical method to estimate the scaling distance $d_q(\boldsymbol{\xi})$. In the existing literature, $d_q(\boldsymbol{\xi})$ is the distance of the q^{th} closest data point to $\boldsymbol{\xi}$ [9, 11, 12]. This leads to $d_q(\boldsymbol{\xi})$ not being differentiable everywhere on $\mathcal{X}$, which is detrimental to the smoothness of the surrogate model (see $\partial\hat{y}/\partial\xi$ in Figure 2 of Section 3.1). We observe that the values $\{\|\mathbf{x}-\boldsymbol{\xi}\|_2^2\}_{\mathbf{x}\in\mathbf{X}}$ are well fitted by a Gamma distribution. The value $d_q(\boldsymbol{\xi})$ is then computed so that the expected number of data points within this distance from $\boldsymbol{\xi}$ is q. The resulting estimate of scaling distance $d_q(\xi)$ is differentiable everywhere, which allows for differentiable and reliable models $\hat{y}$.

The third contribution of this work is pertinent to the selection of the shape parameter $\lambda > 0$, which controls the decrease of the weights. It has the same value for all weights $w_i(\boldsymbol{\xi}), i = 1, ..., p$, all prediction points $\boldsymbol{\xi} \in \mathbf{X}$ and all blackbox output $\{f, c_1, \ldots, c_m\}$.

In previous work, we showed that order errors are relevant indicators of the accuracy of a surrogate model [5]. Order errors aim at quantifying how well the solution to the surrogate problem ($\hat{P}$) matches that of the the original problem (P). In [5] we used an error metric for each of the $m+1$ surrogate models corresponding to the $m+1$ outputs of the blackbox. This metric can be used to fine-tune the model for each blackbox output. This is not possible here since the parameter λ has the same value for each of the $m+1$ surrogate models. As a consequence, we need to define a metric that quantifies the quality of the multi-output model $\boldsymbol{\xi} \to \{\hat{f}(\boldsymbol{\xi}), \hat{c}_1(\boldsymbol{\xi}), \ldots, \hat{c}_m(\boldsymbol{\xi})\}$. Therefore, we propose a generalization of the order error to multi-output surrogates. This order error quantifies how frequently the surrogate model is able to correctly predict which of two points is best in terms of feasibility and optimality. Moreover, the order error used in this work relies on leave-one-out cross-validation, which enables a robust assessment of the predictive capability of the model outside the set $\mathbf{X}$. The global shape coefficient λ is selected to minimize this order error.

1.3 Outline of the paper

In Section 2, we present the precursors of the LOWESS method (polynomial regression and kernel smoothing), the LOWESS model itself, and a way efficiently fit the output of a blackbox. Section 3 presents the method for computing the weight of each data point: we introduce a statistical method to estimate the scaling distance $d_q(\boldsymbol{\xi})$, and a metric specifically designed to quantify the efficiency of a multi-output model in surrogate optimization. Section 4 shows computational results for three engineering design problems, and in Section 5 we draw conclusions and discuss possible directions of future work.

2 LOWESS models

As a convention, we denote $\boldsymbol{\xi} \in \mathcal{X} \subseteq \mathbb{R}^n$ the point of the design space where we want to predict the value of a function y, which can be either the objective function f or a constraint function c_j (with $j = 1...m$).

Locally Weighted Scatterplot Smoothing (LOWESS) models build a local polynomial regression at the point $\boldsymbol{\xi}$ where the value of a function y is to be estimated [3, 9, 10, 11, 12]. This local regression is denoted $\hat{y}_{\boldsymbol{\xi}}(\mathbf{x})$ and emphasizes data points that are close to $\boldsymbol{\xi}$. The value of the LOWESS model at $\boldsymbol{\xi}$ is then defined

as $\hat{y}(\boldsymbol{\xi}) = \hat{y}_{\boldsymbol{\xi}}(\boldsymbol{\xi})$. It is important to understand that the actual value of the local regression $\hat{y}_{\boldsymbol{\xi}}$ is only calculated at $\boldsymbol{\xi}$.

2.1 Precursors: Quadratic regression and Kernel smoothing

Consider the quadratic regression

$$\hat{y}(\boldsymbol{\xi}) = \mathbf{z}(\boldsymbol{\xi})^\top \boldsymbol{\alpha},$$

where $\boldsymbol{\alpha} \in \mathbb{R}^k$ is the vector of coefficients of the quadratic function, $k = (n+1)(n+2)/2$ is the number of basis functions, and the vector $\mathbf{z} : \mathbb{R}^n \to \mathbb{R}^k$ is defined as

$$\mathbf{z}(\mathbf{x}) = \Big[\underbrace{1}_{\substack{\text{constant}\\ \text{term}}} \quad \underbrace{x_1 \dots x_n}_{\substack{\text{linear}\\ \text{terms}}} \quad \underbrace{x_1^2 \dots x_n^2}_{\substack{\text{quadratic}\\ \text{terms}}} \quad \underbrace{x_1 x_2 \dots x_{n-1} x_n}_{\substack{\text{bilinear}\\ \text{terms}}} \Big]^\top. \tag{1}$$

The functions $\{z_j\}_{j=1,\dots,k}$ form a basis of the quadratic polynomials in $\mathbb{R}^n$. The matrix $\mathbf{Z}$ is determined by calculating $\mathbf{z}$ at each data point:

$$\mathbf{Z} = \Big[\mathbf{z}(\mathbf{x}_1) \ \dots \ \mathbf{z}(\mathbf{x}_p)\Big]^\top \in \mathbb{R}^{p \times k}.$$

The vector $\boldsymbol{\alpha}$ is computed such as to minimize

$$\|\mathbf{Z}\boldsymbol{\alpha} - \mathbf{y}\|_2^2 + \|\boldsymbol{\alpha}\|_{\mathbf{J}}^2,$$

where $\mathbf{y} = [y(\mathbf{x}_1) \ \dots \ y(\mathbf{x}_p)]^\top$, $\|\boldsymbol{\alpha}\|_{\mathbf{J}}^2 = \boldsymbol{\alpha}^\top \mathbf{J} \boldsymbol{\alpha}$, $\mathbf{J} \in \mathbb{R}^{k \times k}$ is a diagonal matrix such that $\mathbf{J}_{1,1} = 0$, $\mathbf{J}_{i,i} = r$ for $i = 2, 3, \dots, k$, and $r \geq 0$ is a regularization parameter. The Tikhonov regularization term $\|\boldsymbol{\alpha}\|_{\mathbf{J}}^2$ ensures uniqueness and existence of an optimal value for $\boldsymbol{\alpha}$ [35]. In this study, we use a default value of $r = 10^{-3}$. The value $\mathbf{J}_{1,1} = 0$ allows the regression to be scale-invariant, which means that for any two functions y and y' such that $y' = ay + b$, with $a \in \mathbb{R}$ and $b \in \mathbb{R}$, we have $\hat{y}' = a\hat{y} + b$. The optimal value for $\boldsymbol{\alpha}$ is found by solving the normal equations

$$(\mathbf{Z}^\top \mathbf{Z} + \mathbf{J})\boldsymbol{\alpha} = \mathbf{Z}^\top \mathbf{y}.$$

Kernel Smoothing models (KS) use a weighted sum of all the data points,

$$\hat{y}(\boldsymbol{\xi}) = \frac{\sum_{i=1}^p w_i^{KS}(\boldsymbol{\xi}) y_i}{\sum_{i=1}^p w_i^{KS}(\boldsymbol{\xi})},$$

where $w_i^{KS}(\boldsymbol{\xi})$ quantifies the importance of the observation $[\mathbf{x}_i, y(\mathbf{x}_i)]$ for a prediction in $\boldsymbol{\xi}$. The general rule is that $w_i^{KS}(\boldsymbol{\xi})$ decreases when the distance between $\boldsymbol{\xi}$ and $\mathbf{x}_i$ increases. KS models rely on a kernel function $\phi :^+ \mathbb{R} \to \mathbb{R}^+$, which must be non-increasing, and on a user-defined parameter λ^{KS}:

$$w_i^{KS}(\boldsymbol{\xi}) = \phi(\lambda^{KS} \|\boldsymbol{\xi} - \mathbf{x}_i\|_2).$$

In this study, we use the Gaussian kernel $\phi(d) = \exp(-\pi d^2)$. The advantage of KS models is that the computation is immediate. It does not require the solution of a linear system. However, KS models have a tendency to "undershoot" (low values will be predicted higher, high values will be predicted lower) since the prediction is always bounded by the observations. This also makes KS models unsuitable to extrapolation. However, despite their tendency to undershoot, KS models generally tend to respect the order (i.e. which of two points is best) or sign of the output [5].

2.2 Construction of the local polynomial regression

The local regression around $\boldsymbol{\xi}$ is

$$\hat{y}_{\boldsymbol{\xi}}(\mathbf{x}) = \mathbf{z}_{\boldsymbol{\xi}}^\top(\mathbf{x}) \boldsymbol{\alpha}_{\boldsymbol{\xi}}. \tag{2}$$

There is a certain similarity to polynomial regression, so we note two major differences. First, the coefficients of the polynomial are different for each value of $\boldsymbol{\xi}$. Indeed, the fundamental principle of LOWESS models is to build a different polynomial regression at each prediction point $\boldsymbol{\xi}$. The second difference is that the

basis functions are different for each $\boldsymbol{\xi}$, which will be useful later to reduce computation time. Moreover, to increase the robustness of the regression, we use a number q of basis functions that is strictly smaller than the number of data points p. Specifically, depending on the value of p, $\mathbf{z}_{\boldsymbol{\xi}}$ only uses the $q < p$ first terms of the vector $\mathbf{z}$ defined in Equation (1). Table 1 lists the values of q based on p, and indicates the resulting type of local regression.

Table 1: Set of basis functions based on the number of data points

Condition on p	q	Type of regression
$p \leq n+1$	N.A.	No model construction
$n+1 < p \leq 2n+1$	$n+1$	Linear regression
$2n+1 < p \leq \frac{(n+1)(n+2)}{2}$	$2n+1$	Quadratic regression without bilinear terms
$\frac{(n+1)(n+2)}{2} < p$	$\frac{(n+1)(n+2)}{2}$	Quadratic regression

This leads to the set of basis functions:

$$\mathbf{z}_{\boldsymbol{\xi}}(\mathbf{x}) = \begin{bmatrix} z_1(\mathbf{x}-\boldsymbol{\xi}) \\ \vdots \\ z_q(\mathbf{x}-\boldsymbol{\xi}) \end{bmatrix} \in \mathbb{R}^q,$$

where the functions $\{z_i\}_{i=1,\ldots,q}$ are defined in Equation (1). Note that $\mathbf{z}_{\boldsymbol{\xi}}(\boldsymbol{\xi}) = [1\ 0 \ldots 0]^\top$. We then define the design matrix $\mathbf{Z}_{\boldsymbol{\xi}}$

$$\mathbf{Z}_{\boldsymbol{\xi}} = \Big[\mathbf{z}_{\boldsymbol{\xi}}(\mathbf{x}_1)\ \ldots\ \mathbf{z}_{\boldsymbol{\xi}}(\mathbf{x}_p)\Big]^\top \in \mathbb{R}^{p\times q}$$

and compute the coefficients of the local regression $\boldsymbol{\alpha}_{\boldsymbol{\xi}} \in \mathbb{R}^q$ to minimize

$$\|\mathbf{Z}_{\boldsymbol{\xi}}\boldsymbol{\alpha}_{\boldsymbol{\xi}} - \mathbf{y}\|^2_{\mathbf{W}_{\boldsymbol{\xi}}} + \|\boldsymbol{\alpha}_{\boldsymbol{\xi}}\|^2_J, \tag{3}$$

where $\mathbf{W}_{\boldsymbol{\xi}}$ is a diagonal matrix such that $(\mathbf{W}_{\boldsymbol{\xi}})_{i,i} \propto w_i(\boldsymbol{\xi}) \geq 0$, with $\text{trace}(\mathbf{W}_{\boldsymbol{\xi}}) = 1$. The constraint on the trace of $\mathbf{W}_{\boldsymbol{\xi}}$ ensures that the ratio between the two terms of (3) remains the same for all $\boldsymbol{\xi}$. The details of the computation of $w_i(\boldsymbol{\xi})$ are described in Section 3. Using the normal equations, $\boldsymbol{\alpha}_{\boldsymbol{\xi}}$ can be obtained by solving the symmetric system:

$$\Big(\underbrace{\mathbf{Z}_{\boldsymbol{\xi}}^\top \mathbf{W}_{\boldsymbol{\xi}} \mathbf{Z}_{\boldsymbol{\xi}} + \mathbf{J}}_{\mathbf{A}_{\boldsymbol{\xi}}}\Big)\boldsymbol{\alpha}_{\boldsymbol{\xi}} = \mathbf{Z}_{\boldsymbol{\xi}}^\top \mathbf{W}_{\boldsymbol{\xi}} \mathbf{y}. \tag{4}$$

We observe that this system generalizes the two types of models described in Section 2.1. On the one hand, if there is only one constant basis function (i.e., $q = 1$, $\mathbf{z}_{\boldsymbol{\xi}}(\mathbf{x}) = 1$ and $\mathbf{Z}_{\boldsymbol{\xi}} = [1 \ldots 1]^\top$), we recover KS models. On the other hand, if the weights are chosen to be constant (i.e., $w_i(\boldsymbol{\xi}) = 1/p$, $\forall i = 1, ..., p$), we recover polynomial regression models.

2.3 Computing predictions for several outputs

In the context of surrogate constrained optimization problems, it is necessary to predict the function values of objective and constraints. That means that $m+1$ predictions must be made, which requires $m+1$ solutions of the linear system (4), or the explicit computation of the inverse of $\mathbf{A}_{\boldsymbol{\xi}}$. This section shows that the predictions at $\boldsymbol{\xi}$ for all blackbox outputs can be performed by solving a single linear system.

Equation (2) yields

$$\begin{aligned} \hat{y}(\boldsymbol{\xi}) = \hat{y}_{\boldsymbol{\xi}}(\boldsymbol{\xi}) &= \mathbf{z}_{\boldsymbol{\xi}}(\boldsymbol{\xi})^\top \boldsymbol{\alpha}_{\boldsymbol{\xi}} \\ &= \mathbf{e}_1^\top \boldsymbol{\alpha}_{\boldsymbol{\xi}} \ \text{ with } \mathbf{e}_1 = [1\,0 \ldots 0]^\top \in \mathbb{R}^q \\ &= \alpha_{\boldsymbol{\xi},1}, \end{aligned}$$

which means that we only need to compute the first component of $\boldsymbol{\alpha}_{\boldsymbol{\xi}}$ to make a prediction at $\boldsymbol{\xi}$. Then, we define $\mathbf{u}_{\boldsymbol{\xi}} \in \mathbb{R}^q$ as the first column (or the transpose of the first row) of $\mathbf{A}_{\boldsymbol{\xi}}^{-1}$. This vector $\mathbf{u}_{\boldsymbol{\xi}}$ can be calculated by solving the linear system

$$\mathbf{A}_{\boldsymbol{\xi}}\mathbf{u}_{\boldsymbol{\xi}} = \mathbf{e}_1. \tag{5}$$

Using Equation (4), we obtain

$$\hat{y}(\boldsymbol{\xi}) = \mathbf{u}_{\boldsymbol{\xi}}^\top \mathbf{Z}_{\boldsymbol{\xi}}^\top \mathbf{W}_{\boldsymbol{\xi}}\, \mathbf{y}.$$

By replacing the vector $\mathbf{y}$ by the matrix $\mathbf{Y} \in \mathbb{R}^{p\times(m+1)}$ in which each row stores the values of the blackbox outputs for one data point:

$$\mathbf{Y} = \begin{bmatrix} f(\mathbf{x}_1) & c_1(\mathbf{x}_1) & \dots & c_m(\mathbf{x}_1) \\ \vdots & \vdots & & \vdots \\ f(\mathbf{x}_p) & c_1(\mathbf{x}_p) & \dots & c_m(\mathbf{x}_p) \end{bmatrix}$$

we can make the predictions of all the blackbox outputs all at once:

$$\hat{\mathbf{y}}(\boldsymbol{\xi}) = \begin{bmatrix} \hat{f}(\boldsymbol{\xi}) & \hat{c}_1(\boldsymbol{\xi}) & \dots & \hat{c}_m(\boldsymbol{\xi}) \end{bmatrix} = \mathbf{u}_{\boldsymbol{\xi}}^\top \mathbf{Z}_{\boldsymbol{\xi}}^\top \mathbf{W}_{\boldsymbol{\xi}} \mathbf{Y}.$$

2.4 Solving the linear system directly or iteratively

Many different methods can be used to solve the linear system (5). The important question is whether it is more efficient to solve this problem directly or iteratively. Taking into account that $\mathbf{A}_{\boldsymbol{\xi}}$ is symmetric and positive-definite, we consider two methods: a direct method based on Cholesky decomposition and an iterative method based on conjugate gradients.

In the context of surrogate-assisted optimization, the model $\hat{\mathbf{y}}(\boldsymbol{\xi})$ is evaluated at a large number of points $\boldsymbol{\xi}$ that get closer and closer to each other as the solution of the surrogate problem ($\hat{P}$) unfolds. Let us denote the last point where the model $\hat{\mathbf{y}}(\boldsymbol{\xi})$ was evaluated at with $\boldsymbol{\xi}_0$. If the distance between $\boldsymbol{\xi}_0$ and $\boldsymbol{\xi}$ is small enough, then the value $\mathbf{u}_{\boldsymbol{\xi}_0}$ is assumed to be a good starting point for solving the linear system (5) with an iterative method.

Figure 1 depicts the computational effort required for solving a linear system with both methods. The top plot compares computation time depending on $\|\boldsymbol{\xi} - \boldsymbol{\xi}_0\|_2$ for $n \in \{2, 4, 8, 16\}$. The bottom plot compares computation time depending on the initial residual $\|\mathbf{A}_{\boldsymbol{\xi}}\mathbf{u}_{\boldsymbol{\xi}_0} - \mathbf{e}_1\|_2$. The tolerance on the residual for the iterative method is 10^{-9}. From this figure, we conclude that the iterative method is more efficient if $\|\mathbf{A}_{\boldsymbol{\xi}}\mathbf{u}_{\boldsymbol{\xi}_0} - \mathbf{e}_1\|_2 \leq 10^{-4}$ or if $n \leq 3$.

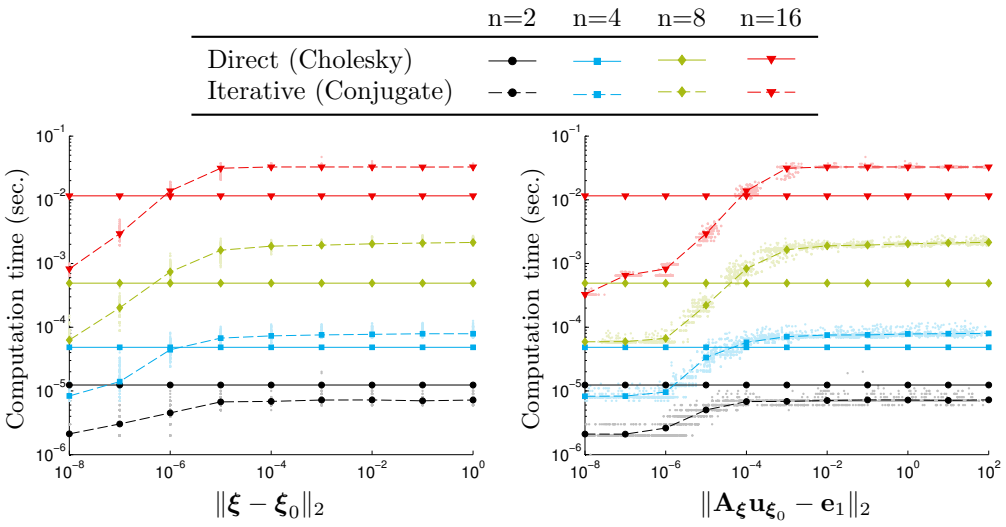

Figure 1: Comparison of computation time for the solution of the linear system with direct and iterative methods

3 Weights computation in LOWESS models

The weight $w_i(\boldsymbol{\xi})$ quantifies the relative importance of the data point $\mathbf{x}_i$ in the construction of the local regression at $\boldsymbol{\xi}$:

$$w_i(\boldsymbol{\xi}) = \phi\left(\lambda \frac{\|\boldsymbol{\xi} - \mathbf{x}_i\|_2}{d_q(\boldsymbol{\xi})}\right),$$

where $\phi(d) = e^{-\pi d^2}$ is the Gaussian kernel function, $\lambda > 0$ is a parameter that control the general shape of the model, and $d_q(\boldsymbol{\xi})$ is a local scaling coefficient that estimates the distance of the q^{th} closest data point to $\boldsymbol{\xi}$. References [9, 11] suggest the use of a kernel function with compact support (namely, the tri-cubic function $\phi(d) = (1-d^3)^3$), but we found that in our context this can lead to ill-posed systems for some values of λ and $d_q(\boldsymbol{\xi})$, hence our choice of the Gaussian kernel. In addition to this choice of ours, we introduce two novel ideas for the calculation of $d_q(\boldsymbol{\xi})$ and λ, described in the next two subsections.

3.1 Using the Gamma distribution to compute the scaling distance $d_q(\boldsymbol{\xi})$

The empirical method of computing $d_q(\boldsymbol{\xi})$ as the distance between $\boldsymbol{\xi}$ and the q^{th} closest data point to $\boldsymbol{\xi}$ leads to $d_q(\boldsymbol{\xi})$ and $\hat{y}$ not being differentiable everywhere (see Figure 2). We propose to compute $d_q(\boldsymbol{\xi})$ so that it satisfies

$$\mathbb{E}\left[\text{card}\Big\{\mathbf{x}_i : \mathbf{x}_i \in \mathbf{X}, \|\boldsymbol{\xi} - \mathbf{x}_i\|_2 \leq d_q(\boldsymbol{\xi})\Big\}\right] = q. \tag{6}$$

In other words, the expected number of data points within this distance from $\boldsymbol{\xi}$ is q.

Such statistical criteria requires the modeling of $\|\boldsymbol{\xi} - \mathbf{x}_i\|_2$ as a random variable. In the preliminary stages of this study, we investigated which distribution would fit these data well. The first idea was to use a noncentral chi (or chi-square) distribution but such distribution does not handle the case where the coordinates of the data points $\{\mathbf{x}_1, ..., \mathbf{x}_p\}$ are correlated and we are not aware of any closed-form distribution that is strictly defined to handle this case.

However, we observed that the values $\left\{\|\boldsymbol{\xi} - \mathbf{x}_i\|_2^2\right\}_{i=1,...,p}$ can be well fitted by a Gamma distribution, whose parameters are denoted $k_{\boldsymbol{\xi}}$ (shape parameter) and $\theta_{\boldsymbol{\xi}}$ (scale parameter). To estimate these parameters, we first compute the empirical mean and variance

$$\mu_{\boldsymbol{\xi}} = \frac{1}{p}\sum_{i=1}^{p} \|\boldsymbol{\xi} - \mathbf{x}_i\|_2^2 \quad \text{and} \quad \sigma_{\boldsymbol{\xi}}^2 = \frac{1}{p-1}\sum_{i=1}^{p} \left(\|\boldsymbol{\xi} - \mathbf{x}_i\|_2^2 - \mu_{\boldsymbol{\xi}}\right)^2.$$

Then, the parameters of the Gamma distribution are obtained from

$$k_{\boldsymbol{\xi}} = \frac{\mu_{\boldsymbol{\xi}}^2}{\sigma_{\boldsymbol{\xi}}^2} \quad \text{and} \quad \theta_{\boldsymbol{\xi}} = \frac{\sigma_{\boldsymbol{\xi}}^2}{\mu_{\boldsymbol{\xi}}}.$$

If $\mu_{\boldsymbol{\xi}} > 0$ and $\sigma_{\boldsymbol{\xi}} > 0$, then $k_{\boldsymbol{\xi}}$ and $\theta_{\boldsymbol{\xi}}$ are continuous and differentiable with respect to $\boldsymbol{\xi}$. The highly unlikely case where $\sigma_{\boldsymbol{\xi}} = 0$ occurs when all the data points are located at an equal distance from $\boldsymbol{\xi}$. In this case, we set $w_i(\boldsymbol{\xi}) = 1/p$ for all $i = 1, ..., p$. This behavior does not introduce discontinuities in $\mathbf{W}_{\boldsymbol{\xi}}$ when $0 < q < p$ and also handles the even more unlikely case where $\mu_{\boldsymbol{\xi}} = 0$ since $\mu_{\boldsymbol{\xi}} = 0$ implies that $\sigma_{\boldsymbol{\xi}} = 0$.

Note that there are more accurate methods to estimate the parameters of the Gamma distribution [40, 30, 8]. However, these methods have drawbacks. They might require the use of a logarithm, which is problematic when a distance is zero (e.g., for $\boldsymbol{\xi} \in \mathbf{X}$). They might be iterative and more computationally expensive. For these reasons, we prefer the method proposed above, which is simple and leads to very smooth values of $k_{\boldsymbol{\xi}}$, $\theta_{\boldsymbol{\xi}}$ and, as a result, $d_q(\boldsymbol{\xi})$. We believe that these benefits outweigh the disadvantages of the estimation bias.

Once these two parameters are obtained, Equation (6) leads to

$$d_q(\boldsymbol{\xi}) = \sqrt{g^{(-1)}\left(k_{\boldsymbol{\xi}}, \theta_{\boldsymbol{\xi}}; \frac{q}{p}\right)},$$

where $g^{(-1)}(k, \theta; .)$ is the inverse function of the cumulative density function of a Gamma distribution with parameters (k, θ).

Figure 2 compares two LOWESS models obtained using the empirical method and the Gamma distribution for the calculation of $d_q(\boldsymbol{\xi})$. For both models, the global shape parameter λ is set to 1. The plot at the top

of the figure depicts the data points and the two LOWESS models. The two models are very similar. The plot in the middle of the figure depicts the value of $d_q(\boldsymbol{\xi})$ as computed by the two methods. As expected, the empirical method leads to "sawtooth" behavior. The Gamma distribution yields a smooth function but tend to overestimate $d_q(\boldsymbol{\xi})$. The plot at the bottom of the figure depicts the derivative of the model (approximated by means of finite differences). The Gamma distribution yields much smoother values.

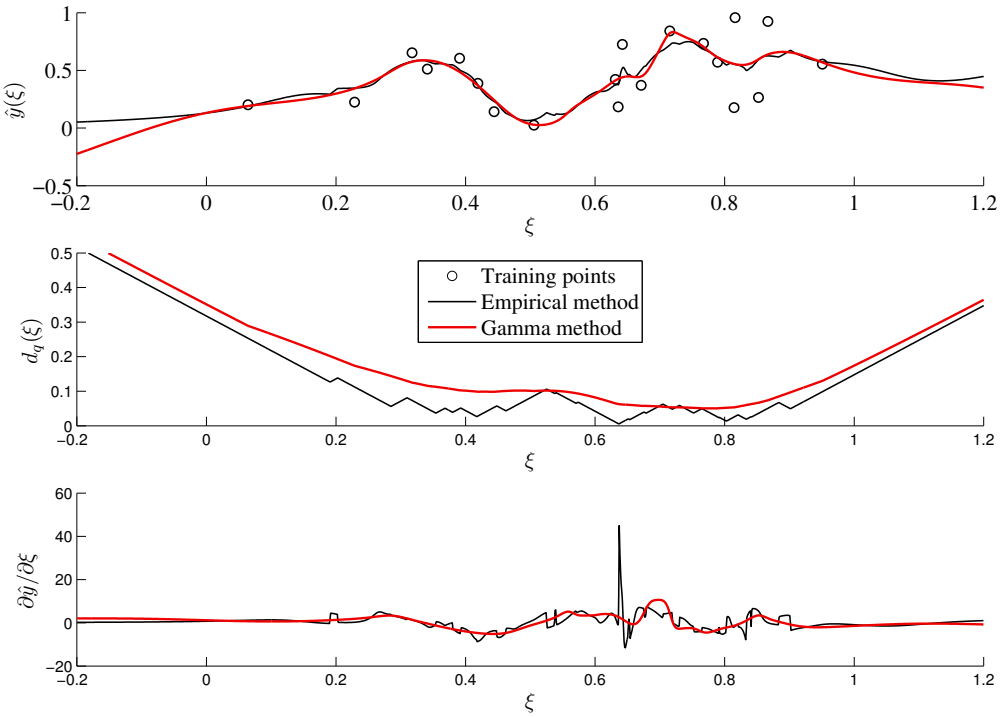

Figure 2: Comparison of LOWESS models obtained using empirical and Gamma distribution methods for the computation of $d_q(\boldsymbol{\xi})$

Figure 3 compares the value of $d_q(\boldsymbol{\xi})$ in the context of surrogate-assisted optimization. We display the values obtained during the 7 first iterations for the Lockwood optimization problem (see Section 4), which has $n = 6$ variables. The plot on the left of the figure depicts the values of $d_q(\boldsymbol{\xi})$ with both methods for each of the first 12,000 LOWESS model evaluations. It shows that there is an excellent correlation (99.9%) between the two methods. The plot on the right of the figure shows the evolution of $d_q(\boldsymbol{\xi})$ as the optimization unfolds, as well as the number of training points and basis functions.

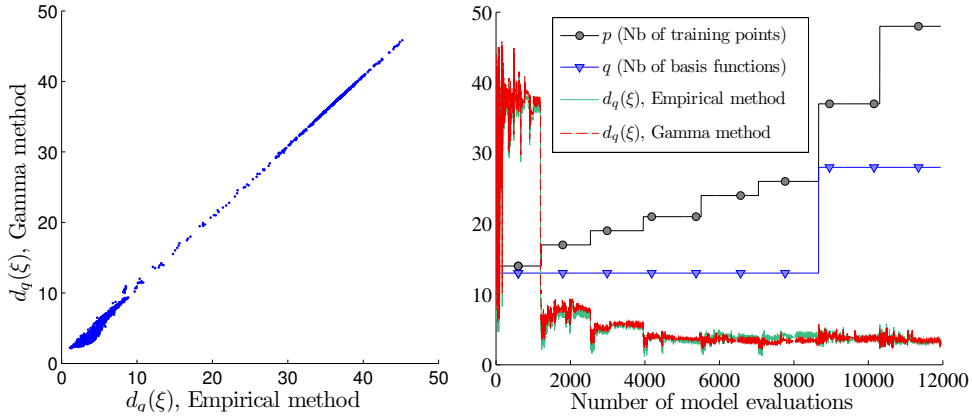

Figure 3: Comparison of the value of $d_q(\boldsymbol{\xi})$ obtained using empirical and Gamma distribution methods in the context of surrogate-assisted optimization

3.2 Computing λ using order error minimization

For multi-output models, it is impossible to compute different values of λ for each blackbox outputs. Therefore we compute the scalar λ that minimizes an error metric $\mathcal{M}(\lambda)$, which reflects the overall predictive capability of the multi-output model $\left[\hat{f}(\boldsymbol{\xi})\ \hat{c}_1(\boldsymbol{\xi})\ \ldots\ \hat{c}_m(\boldsymbol{\xi})\right]$. Specifically, we propose a novel approach that utilizes the order error metric introduced in [5], based on the fact that Problems (P) and ($\hat{P}$) have the same minimizer(s) when the following two conditions are satisfied:

$$f(\mathbf{x}) \leq f(\mathbf{x}') \Leftrightarrow \hat{f}(\mathbf{x}) \leq \hat{f}(\mathbf{x}'), \quad \forall \mathbf{x}, \mathbf{x}' \in \mathcal{X} \tag{7}$$

$$c_j(\mathbf{x}) \leq 0 \Leftrightarrow \hat{c}_j(\mathbf{x}) \leq 0, \quad \forall \mathbf{x} \in \mathcal{X}, \forall j = 1, 2, ..., m. \tag{8}$$

The metric proposed in [5] quantifies the violation of these conditions for each blackbox outputs. To adapt this metric to multi-output models, we use the aggregate constraint violation function [18]

$$h(\mathbf{x}) = \sum_{j=1}^{m} \max\{0, c_j(\mathbf{x})\}^2,$$

and define the order operators

$$\mathbf{x} \prec \mathbf{x}' \Leftrightarrow \begin{cases} h(\mathbf{x}) < h(\mathbf{x}') \\ \text{or} \\ h(\mathbf{x}) = h(\mathbf{x}') \text{ and } f(\mathbf{x}) < f(\mathbf{x}') \end{cases}$$

$$\mathbf{x} \preceq \mathbf{x}' \Leftrightarrow \texttt{not}(\mathbf{x}' \prec \mathbf{x})$$

which are transitive. In particular, a minimizer $\mathbf{x}^*$ of Problem (P), i.e. a feasible design with the best objective, or the least infeasible design with the best objective, is such that $\mathbf{x}^* \preceq \mathbf{x}$, $\forall \mathbf{x} \in \mathcal{X}$. Note that this operator differs from the *dominance* operator often used in multiobjective optimization.

By the same principle, we define the operator $\hat{\prec}$ by using $\hat{f}$ and $\hat{h} = \sum_{j=1}^{m} \max\{0, \hat{c}_j(\mathbf{x})\}^2$ instead of f and h. Conditions (7) and (8), which guarantee the equivalence between Problems (P) and ($\hat{P}$), can then be completely reformulated as:

$$\mathbf{x} \prec \mathbf{x}' \Leftrightarrow \mathbf{x} \hat{\prec} \mathbf{x}', \quad \forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}, \tag{9}$$

and we can define the aggregate order error (AOE) metric, which quantifies the violation of (9) on the data points:

$$\mathcal{M}_{AOE} = \frac{1}{p^2} \sum_{i=1}^{p} \sum_{j=1}^{p} \texttt{xor}\Big(\mathbf{x}_i \prec \mathbf{x}_j, \mathbf{x}_i \hat{\prec} \mathbf{x}_j\Big),$$

where `xor` is the exclusive `or` operator (i.e., $\texttt{xor}(A, B) = 1$ if the booleans A and B differ and 0 otherwise). This error metric is 0 if the model is able to correctly predict which of two points is best for any pair of data points in $\mathbf{X}$. However, to quantify the predictive capacity of the model outside of the data points $\mathbf{X}$, we introduce an error metric based on cross-validation [1, 5, 15, 38]. As above, and using the convention that a Leave-One-Out (LOO) cross-validation value $\hat{\hat{y}}(\mathbf{x}_i)$ is the value of a model built without using the data point $\mathbf{x}_i$, we obtain the functions $\hat{\hat{f}}$ and $\hat{\hat{h}}$, and the operator $\hat{\hat{\prec}}$. We then define the aggregated order error with cross-validation (AOECV) metric:

$$\mathcal{M}_{AOECV} = \frac{1}{p^2} \sum_{i=1}^{p} \sum_{j=1}^{p} \texttt{xor}\Big(\mathbf{x}_i \prec \mathbf{x}_j, \mathbf{x}_i \hat{\hat{\prec}} \mathbf{x}_j\Big).$$

The shape parameter λ is chosen to minimize $\mathcal{M}_{AOECV}(\lambda) + \log(\lambda)/p^3$. In the case where several values of λ lead to the same value of $\mathcal{M}_{AOECV}$ (because of the piecewise-constant nature of the metric), the second term will favor smaller values of λ, and thus smoother models. The term $1/p^3$ ensures that the smoothness term will always be an order of magnitude smaller than $\mathcal{M}_{AOECV}$.

4 Numerical examples

We test the efficiency of LOWESS models in solving three engineering design optimization problems: the *tension compression string design* (TCSD) problem defined in [19, 2, 6], the *Lockwood* problem [28, 27, 21, 23] and the *Solar 6* problem, which is the 6^{th} instance of the Solar benchmark collection [20]. Table 2 describes the main properties of these problems (all computations were performed on Intel Xeon X5675 (3.07GHz) processors, with 96Gb of RAM).

Table 2: Parameters of the test problems; there is no previously best known objective function value for the Solar 6 problem

Problem	TCSD	Lockwood	Solar 6
n	3	6	5
m	4	4	6
f^*	0.0126653	22 739.90 ($)	43.9554 (M$)
Best known objective	0.0126652 [19]	22 739.67 ($) [23]	43.9554 (M$)
Budget	$1000(n+1)$	$1000(n+1)$	$100(n+1)$
Mean execution time	0.068 sec	2.4 sec	267 sec

The TCSD problem aims at minimizing the weight of a tension compression spring under 4 mechanics constraints. The Lockwood problem minimizes the cost of the extraction of liquid pollutant from the ground in the Lockwood Solvent Ground Water Plume, in Montana. The Solar 6 problem considers the minimization of the cost of the thermal storage in a solar farm under constraints that ensure the feasibility of the design and the ability of the system to sustain a certain electrical power output during a nychthemeron[1]. We scaled the outputs of the blackbox used to evaluate the objective and constraint functions of the Solar 6 problem as listed in Table 3. The changes in the units of the objective and in the failure values of f, c_1 and c_6 (i.e., the values that are returned when the execution of the blackbox fails to return values) allow us to avoid perturbing the LOWESS model with extremely high values while still being able to take into account the information returned by the blackbox: if the computation failed, the design is not feasible and the objective is high. In addition, the values of the surrogate models of the binary constraints c_2, c_3, c_4, and c_5 are rounded in $\{0, 1\}$.

Table 3: Modification on the Solar 6 problem

	Original problem	Modified problem
Unit of the objective	$	M$
Failure value for f	10^{+20}	2.10^{+7}
Failure value for c_1 and c_6	10^{+20}	100

4.1 Surrogate-assisted optimization

We use the MADS algorithm [4] to solve Problem (P) using a surrogate-assisted optimization strategy. MADS is based on a search-and-poll paradigm [7]: the (optional) search step provides the chance to the design engineer to solve the problem in any heuristic or systematic manner. The poll consists of generating candidates in a neighborhood of the incumbent solution by means of positive spanning theory, and ensures the convergence properties of the algorithm. Our surrogate-assisted optimization strategy utilizes the search step to build a surrogate model of the blackbox and solve the Problem ($\hat{P}$) in order to generate a candidate that is then evaluated with the blackbox. We also use the surrogate model to sort the poll candidates in a computationally cost-effective manner. We then use the blackbox to evaluate the sorted candidates opportunistically, which means that the sequential evaluations process is aborted once a candidate is found to lead to an improved solution. This surrogate-assisted (as opposed to surrogate-based) approach improves the efficiency of MADS.

We compare four optimization strategies. The *Quadratic models* strategy uses a local quadratic model and a trust region in the search and the sorting of the poll candidates [14]. The other three strategies are based on the procedure described in [5] but the surrogate modeling technique differs. In the *Kernel Smoothing*

[1] A nychtemeron is a time period of 24 consecutive hours.

strategy, the surrogate model is the kernel smoothing method described in Section 2.1, with a Gaussian kernel where the shape coefficient λ^{KS} is optimized as described in Section 3.2. The *LOWESS Linear* and *LOWESS Quadratic* strategies are the LOWESS models described in this work where the local regression is linear and quadratic, respectively. The scaling distance $d_q(\boldsymbol{\xi})$ is computed using the Gamma distribution method, and the shape parameter λ is selected using the order error metric described in Section 3.2.

4.2 Results

For each problem , we generate $\rho_{\max} = 50$ starting points with Latin Hypercube sampling [29]. We then perform one optimization run with each starting point. For each optimization run ρ, we denote $f_{s,\rho,i}$ the objective function value of the best design found by solver[2] s after i blackbox evaluations (excluding evaluations of surrogates). We assign $f_{s,\rho,i} = +\infty$ if no feasible design is found. The best solution found among all runs of all solvers is denoted f^*. We then define the relative discrepancy between f^* and the best solution for optimization run ρ of solver s after i blackbox evaluations as

$$\delta_{s,\rho,i} = \frac{f_{s,\rho,i} - f^*}{f^*} \geq 0.$$

This definition is appropriate for all three considered problems because their objective function (weight or cost) represent a positive quantity. For each set of runs, we present the value of the median discrepancy after i blackbox evaluations. The use of the median is motivated by the wide ranges of magnitude that the discrepancy values can take; moreover, discrepancy can take the value of zero or infinity, which makes the use of the geometric or arithmetic means, respectively, impossible.

For a given tolerance $\tau \geq 0$, the *ratio of problems solved* for solver s after i blackbox evaluations is calculated as

$$r_{s,i}(\tau) = \frac{1}{\rho_{\max}} \text{card}\{i : \delta_{s,\rho,i} \leq \tau,\ i = 1, ..., \rho_{\max}\}.$$

The data profile represents the ratio of solved problems after i evaluations [31]. For each set of optimization runs, we present the data profiles for $\tau \in \{10^{-1}, 10^{-2}, 10^{-3}\}$. When inspecting the median discrepancy curve, smaller values indicate a better performance. On the contrary, for the data profiles, the larger the better. Figures 4, 5 and 6 depict median discrepancy curves and data profiles for the three problems. Figure 7 and 8 summarize median discrepancy and ratios of solved problems for exhausted blackbox evaluation budgets.

When only the number of blackbox evaluations is considered (and not the actual computation time), the LOWESS models outperform the quadratic and kernel smoothing models in nearly all cases.

When considering the performance depending on the computation time, the LOWESS methods require more time than Kernel Smoothing which itself requires more time than the quadratic model. This is particularly obvious in the TCSD problem, for which the blackbox evaluation time is short. In the Lockwood problem, the LOWESS methods lead to a significant improvement of convergence rate, which makes these methods efficient even though blackbox evaluation time is moderate. In the Solar 6 problem, the longer blackbox evaluation time makes LOWESS computation time insignificant. All solvers have a comparable performance, with the exception of the LOWESS models which have a better ratio of solved problem for small values of τ.

Finally, for all three problems and for each of the performance metrics, the best final score is always achieved by one of the LOWESS methods. The LOWESS methods improve the ratio of solved problems by up to 22%, compared to quadratic models.

[2] We use the term solver to denote the MADS algorithm with a search step that solves a surrogate optimization problem; the four different solvers represent the four different types of surrogate models used.

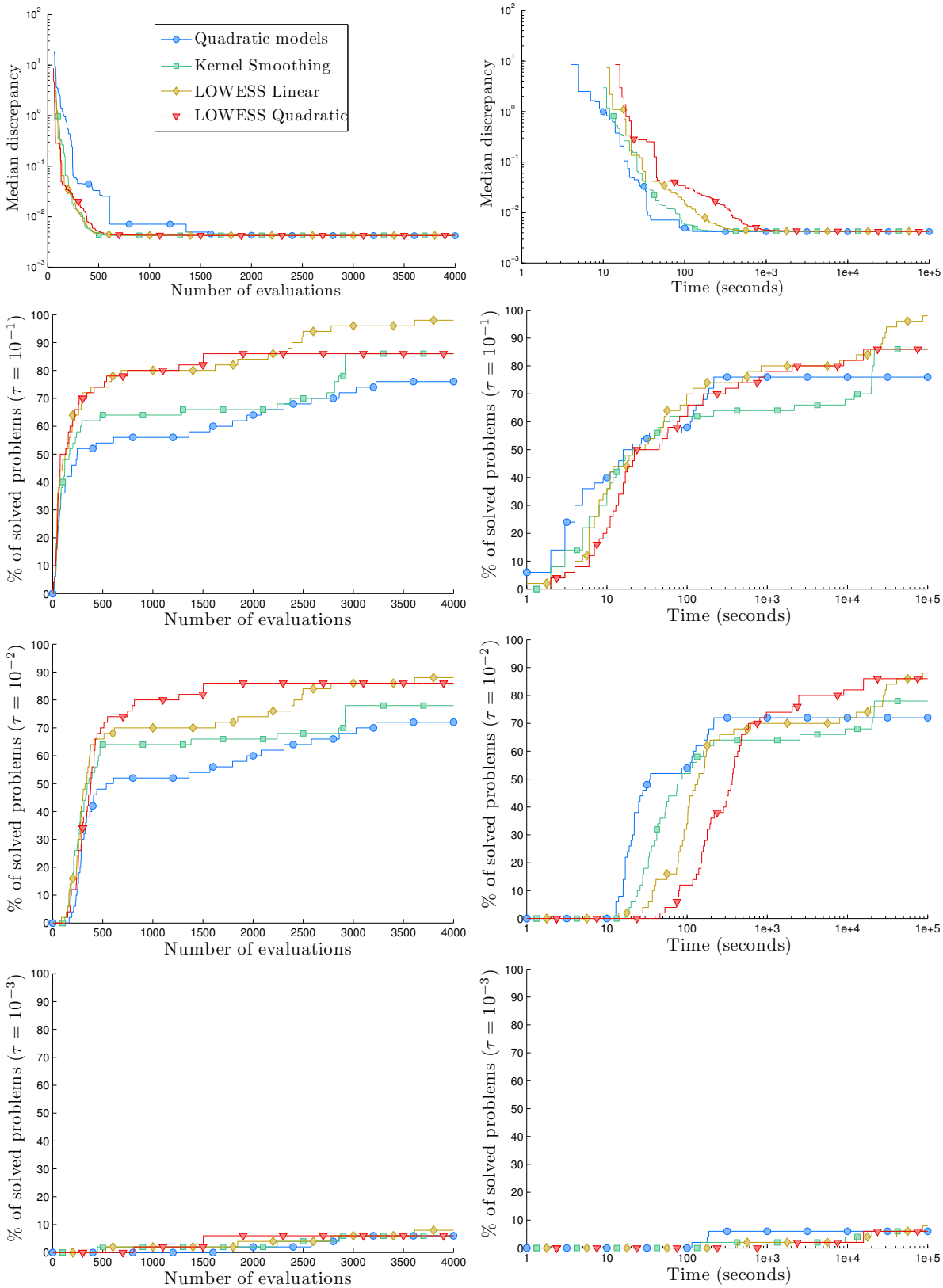

Figure 4: TCSD problem results; for the plots at the top row, smaller discrepancy indicates better performance; for the remainder of the plots, larger percentage indicates better performance

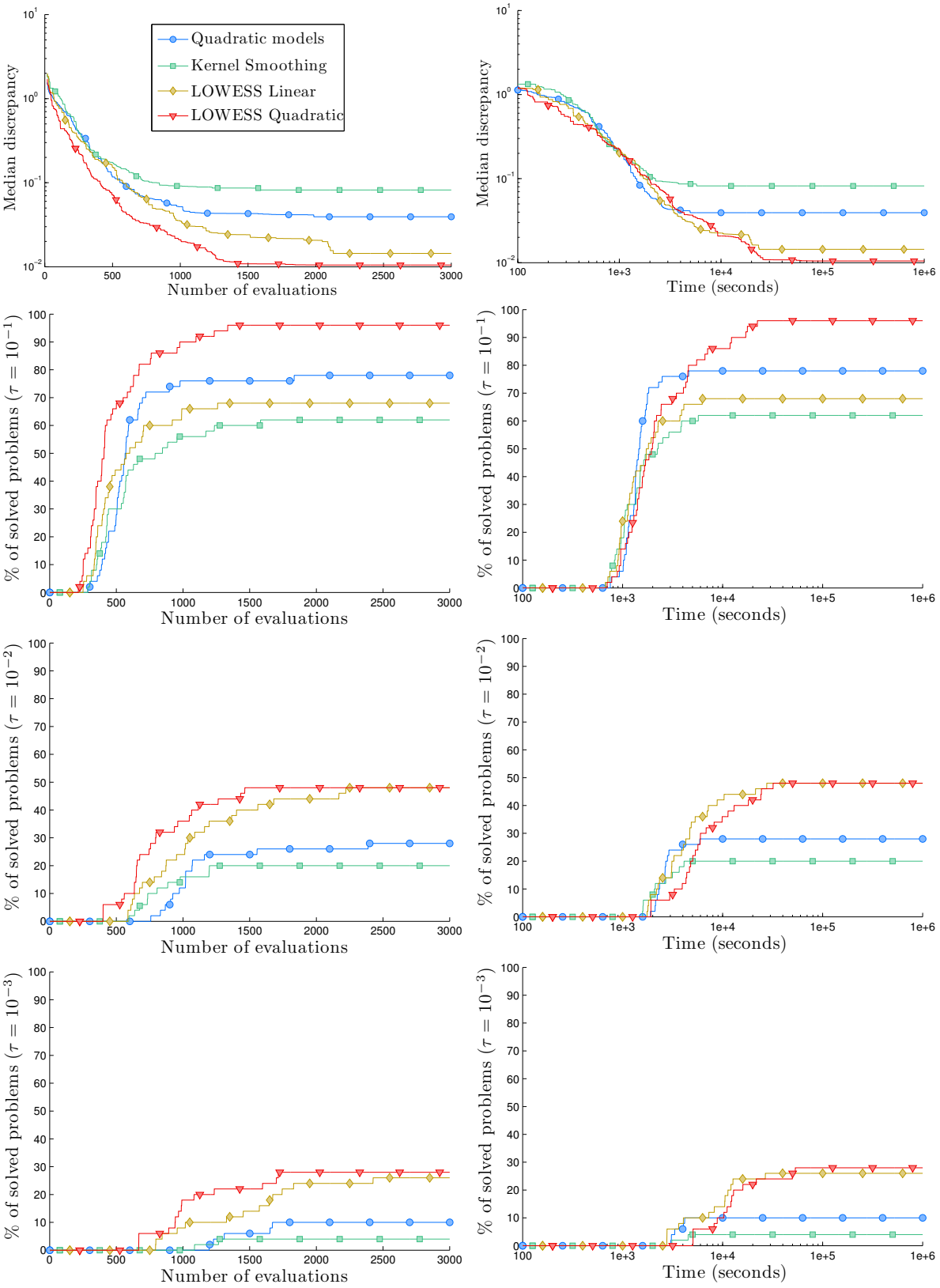

Figure 5: Lockwood problem results; for the plots at the top row, smaller discrepancy indicates better performance; for the remainder of the plots, larger percentage indicates better performance

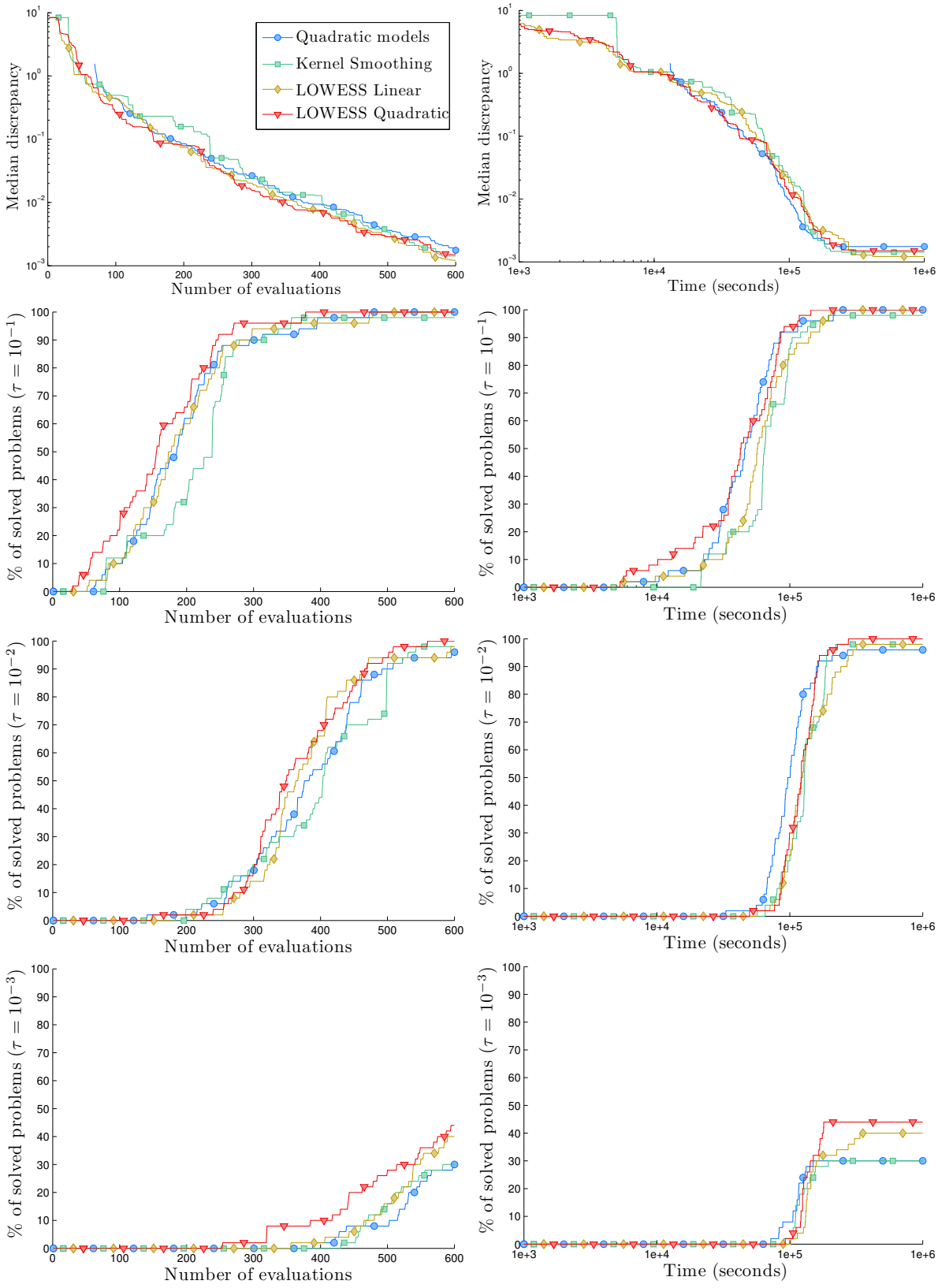

Figure 6: Solar 6 problem results; for the plots at the top row, smaller discrepancy indicates better performance; for the remainder of the plots, larger percentage indicates better performance

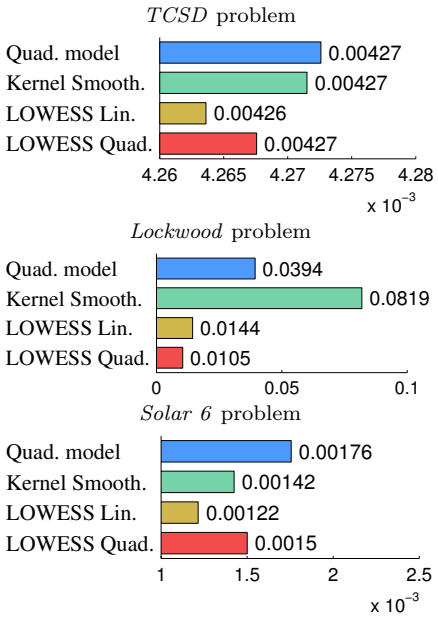

Figure 7: Median discrepancy at exhausted function evaluation budget; smaller values indicate better performance

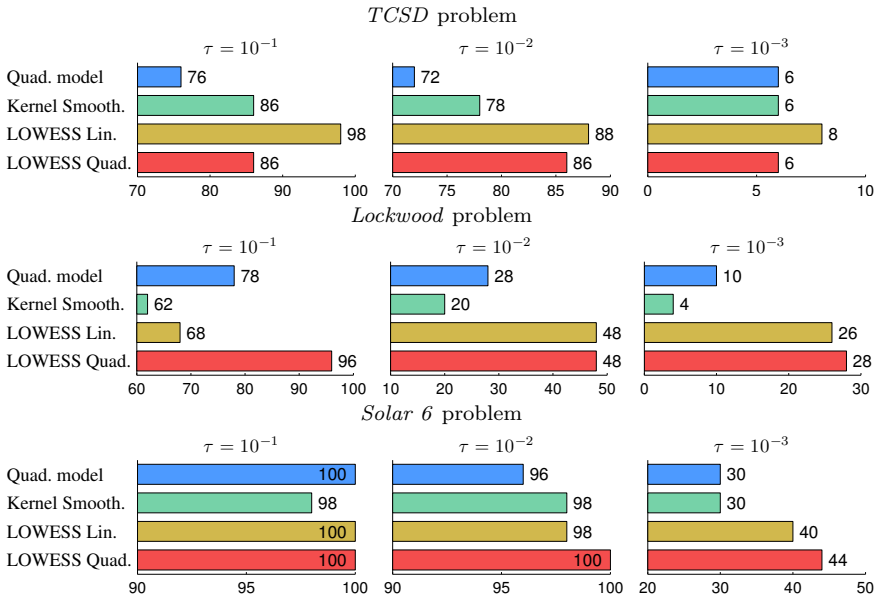

Figure 8: Percentage of solved problems at exhausted function evaluation budget; larger values indicate better performance

The LOWESS model with linear local regression is more time-efficient than with quadratic regression. It is also the best solver in many categories. Thus, it seems particularly relevant on inexpensive problems. This performance can be explained by the fact that it relies on a smaller number of basis functions, thus making it more robust. LOWESS with quadratic local regression performs better on the Lockwood problem and slightly better on the Solar 6 problem. An interesting strategy could be to alternate the use of these two models.

5 Concluding summary

LOWESS models are quite useful surrogates albeit computationally expensive to build. We have proposed ways to reduce the computation time necessary to build LOWESS models by choosing appropriate techniques

for solving linear systems. Using the Gamma distribution to compute the local scaling distance $d_q(\boldsymbol{\xi})$ allows to build smoother models with an equivalent predictive capacity. The use of an order error to optimize the global shape coefficient λ leads to models that are highly suitable to surrogate optimization. For inexpensive blackboxes, the use of linear LOWESS models is recommended whereas for expensive blackboxes, the performance of quadratic LOWESS models overcomes the long computation time required to build them.

Future work could consider using the order error to optimize the shape parameter and/or the regularization coefficient of other types of multi-output surrogate models (e.g., radial basis function (RBF) and Kriging models). Moreover, the multi-output order error proposed in this work can be generalized to multiobjective optimization. Finally, as the use of the Gamma distribution allows for the construction of differentiable models, it would be interesting to use the derivatives of the LOWESS models to solve the surrogate problem using efficient gradient-based algorithms.

References

[1] E. Acar and M. Rais-Rohani. Ensemble of metamodels with optimized weight factors. Structural and Multidisciplinary Optimization, 37(3):279–294, 2009.

[2] J. Arora. Introduction to Optimum Design. Elsevier Science, 2004.

[3] C.G. Atkeson, A.W. Moore, and S. Schaal. Locally weighted learning. Artificial Intelligence Review, pages 11–73, 1997.

[4] C. Audet and J.E. Dennis, Jr. Mesh adaptive direct search algorithms for constrained optimization. SIAM Journal on Optimization, 17(1):188–217, 2006.

[5] C. Audet, M. Kokkolaras, S. Le Digabel, and B. Talgorn. Order-based error for managing ensembles of surrogates in derivative-free optimization. Les Cahiers du Gerad, G-2016-36, 2016.

[6] A.D. Belegundu. A Study of Mathematical Programming Methods for Structural Optimization. University of Iowa, 1982.

[7] A.J. Booker, J.E. Dennis, Jr., P.D. Frank, D.B. Serafini, V. Torczon, and M.W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. Structural and Multidisciplinary Optimization, 17(1):1–13, 1999.

[8] S. C. Choi and R. Wette. Maximum Likelihood Estimation of the Parameters of the Gamma Distribution and Their Bias. Technometrics, 11(4):683–690, 1969.

[9] W.S. Cleveland. Robust locally weighted regression and smoothing scatterplots. Journal of the American Statistical Association, 74:829–836, 1979.

[10] W.S. Cleveland. LOWESS: A Program for Smoothing Scatterplots by Robust Locally Weighted Regression. The American Statistician, 35(1), 1981.

[11] W.S. Cleveland and S.J. Devlin. Locally weighted regression: An approach to regression analysis by local fitting. Journal of the American Statistical Association, 83:596–610, 1988.

[12] W.S. Cleveland, S.J. Devlin, and E. Grosse. Regression by local fitting: methods, properties, and computational algorithms. Journal of Econometrics, 37(1):87–114, 1988.

[13] R.A. Cohen. An Introduction to PROC LOESS for Local Regression. In Proceedings of the 24th SAS users group international conference, 1999. www.ats.ucla.edu/stat/SAS/library/.

[14] A.R. Conn and S. Le Digabel. Use of quadratic models with mesh-adaptive direct search for constrained black box optimization. Optimization Methods and Software, 28(1):139–158, 2013.

[15] B. Efron. Estimating the error rate of a prediction rule: Improvement on Cross-Validation. Journal of the American Statistical Association, 78(382):316–331, 1983.

[16] J. Fan. Local linear regression smoothers and their minimax efficiencies. The Annals of Statistics, 21:196–216, 1993.

[17] J. Fan and I. Gijbels. Variable bandwidth and local linear regression smoothers. The Annals of Statistics, 20:2008–2036, 1992.

[18] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. Mathematical Programming, Series A, 91:239–269, 2002.

[19] H. Garg. Solving structural engineering design optimization problems using an artificial bee colony algorithm. Journal of Industrial and Management Optimization, 10(3):777–794, 2014.

[20] M. Lemyre Garneau. Modelling of a solar thermal power plant for benchmarking blackbox optimization solvers. Master's thesis, École Polytechnique de Montréal, 2015. Available at https://publications.polymtl.ca/1996/.

[21] R.B. Gramacy and S. Le Digabel. The mesh adaptive direct search algorithm with treed Gaussian process surrogates. Pacific Journal of Optimization, 11(3):419–447, 2015.

[22] T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[23] A. Kannan and S.M. Wild. Benefits of deeper analysis in simulation-based groundwater optimization problems. In Proceedings of the XIX International Conference on Computational Methods in Water Resources (CMWR 2012), June 2012. Available at www.mcs.anl.gov/~wild/papers/2012/AKSW12.pdf.

[24] U. Lall, Y.I. Moon, H.H. Kwon, and K. Bosworth. Locally weighted polynomial regression: Parameter choice and application to forecasts of the Great Salt Lake. Water Resources Research, 42(5):n/a–n/a, 2006. W05422.

[25] C. Loader. Local regression and likelihood. New York: Springer-Verlag, 1999.

[26] F.R. Macaulay. The smoothing of time series. chapter Curve Fitting and Graduation, pages 31–42. National Bureau of Economic Research, 1931.

[27] L.S. Matott, K. Leung, and J. Sim. Application of MATLAB and Python optimizers to two case studies involving groundwater flow and contaminant transport modeling. Computers & Geosciences, 37(11):1894–1899, 2011.

[28] L.S. Matott, A.J. Rabideau, and J.R. Craig. Pump-and-treat optimization using analytic element method flow models. Advances in Water Resources, 29(5):760–775, 2006.

[29] M.D. McKay, R.J. Beckman, and W.J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics, 21(2):239–245, 1979.

[30] T.P. Minka. Estimating a gamma distribution, 2002. http://research.microsoft.com/en-us/um/people/minka/papers/minka-gamma.pdf.

[31] J.J. Moré and S.M. Wild. Benchmarking derivative-free optimization algorithms. SIAM Journal on Optimization, 20(1):172–191, 2009.

[32] H.-G. Müller. Weighted Local Regression and Kernel Methods for Nonparametric Curve Fitting. Journal of the American Statistical Association, 82(397):231–238, March 1987.

[33] J. Müller and R. Piché. Mixture surrogate models based on dempster-shafer theory for global optimization problems. Journal of Global Optimization, 51(1):79–104, 2011.

[34] M. Natrella. NIST/SEMATECH e-Handbook of Statistical Methods. NIST/SEMATECH, July 2010.

[35] M.J.L. Orr. Introduction to radial basis function networks, 1996.

[36] N.V. Queipo, R.T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P.K. Tucher. Surrogate-based analysis and optimization. Progress in Aerospace Sciences, 41(1):1–28, January 2005.

[37] D. Ruppert and M. P. Wand. Multivariate locally weighted least squares regression. The Annals of Statistics, 22:1346–1370, 1994.

[38] M. Stone. An asymptotic equivalence of choice of model by cross-validation and akaike's criterion. Journal of the Royal Statistical Society. Series B (Methodological), 39(1):44–47, 1977.

[39] J.-A. Ting, S. Vijayakumar, and S. Schaal. Locally Weighted Regression for Control, pages 613–624. Springer US, Boston, MA, 2010.

[40] D.S. Wilks. Maximum Likelihood Estimation for the Gamma Distribution Using Data Containing Zeros. Journal of Climate, 3(12):1495–1501, December 1990.