2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

A weighted linearization approach to gradient descent optimization

Damiano Rotondo[1] , Mayank Shekhar Jha[2]

Abstract— **The weighted linearization is a generalization of the first-order Taylor approximation where the computation of the Jacobian matrices at the point of interest is replaced by the computation of the integral of the Jacobian matrix function by a weighting function that expresses how much different parts of the domain should be taken into account during the linearization. This paper blends the weighted linearization with the existing gradient descent (GD) method to develop a novel optimization technique named *weighted gradient descent* (WGD). The WGD is shown to outperform the GD in terms of mean absolute error, given an appropriate tuning of the WGD hyper-parameters, when applied to various nonlinear functions that are multi-modal in nature, thus exhibiting several optima.**

I. Introduction

Nonlinear functions are of special interest to various domains of engineering and technology since most systems of practical interest exhibit nonlinearities, including multi-scale spatio-temporal phenomena [1], dead-zone [2], multi-stability [3], to name a few. Such systems call for incorporation of nonlinearities within a given approach, which may be a non-trivial task for tractable design, and non-applicable in general sense for complex large scaled systems. Nonlinear programming problems in management science and operations research [4] with inherent model nonlinearities present challenges in that searching for the global optimum within an acceptable computational time is generally difficult.

On the contrary, approaches for linear systems are very well established in the literature with scalable design, analysis and generalised closed form solutions [5]. These approaches can be applied to nonlinear systems by using effective linear approximations that are equivalent to the nonlinearities in a neighborhood of the operating point of interest. These linear approximations are often obtained by finite difference approaches in form of piecewise or first-order methods to solve nonlinear optimization problems [6]. Further functional expansion methods such as Magnus expansions [7], Chen-Fleiss expansion [8], etc. have been developed for formal linear differential equations. Although they are useful in obtaining explicit expressions of the solutions and can be adapted to address ordinary nonlinear differential equations on smooth manifolds [9], they remain limited in face of multi-modal functions with several local/global minima [10]. In this context, the most common linearization method, i.e. expansion in Taylor's series around the operating point, is a popular approach and remains quite effective for approximating nonlinearities as long as the deviation of the state variables from the operating point is minor [11].

On the other hand, gradient descent (GD) based approaches have proven to be extremely effective in various fields where the central problem is cast as an optimization problem calling for minimization (or maximization) of a cost function with respect to a given set of parameters. Most of the GD based approaches require the objective function to be differentiable, call for computation of first order partial derivatives and remain relatively efficient in face of nonlinear functions as well as non-smooth problems. For example, [12] used GD based linearization under constraints for nonlinear model predictive control; [13] presented a GD based approach for non-convex non-smooth problems; [14] presented a stochastic GD based approach for nonlinear ill-posed problems; [15] presented approaches for multi-point generalization of the gradient descent iteration for local optimization for non smooth problems. GD based approaches are of central importance in various fields of engineering and technology, including machine learning [16] and modern deep learning [17], [18], system identification of discrete-time systems [19] and continuous-time systems [20], [21], reinforcement learning for optimal control learning [22] in model-free as well as model-based settings [23], etc. Although GD based approaches are effective in general sense, they suffer from slower convergence rate issues in face of noisy objective functions and strong non-linearity.

As such, it becomes imperative to develop effective GD based approaches that lead to faster convergence to global/local optima. In this context, the weighted linearization technique was recently proposed in [24], wherein computation of the Jacobian matrices at the state trajectory of interest is replaced by the multiple integral over the state and input spaces with the corresponding Jacobian matrix functions multiplied by a weighting function. It was shown that the standard Taylor linearization can be recovered as a particular case of the proposed weighted linearization. Moreover, in [25] the weighted linearization technique has been incorporated in the equations of the extended Kalman filter to obtain a new version of the Kalman filter, referred to as weighted Kalman filter, which was shown to to exhibit better convergence properties and less average estimation error than the extended Kalman filter.

This paper blends the benefits of the recently proposed weighted linearization technique [24] with the existing GD method, to develop a novel *weighted gradient descent* (WGD) approach that generalises the existing standard GD

[1] Department of Electrical and Computer Engineering (IDE), University of Stavanger, Stavanger, Norway. E-mail: damiano.rotondo@uis.no
[2] CRAN, UMR 7039, CNRS, Université de Lorraine, 54506 Vandoeuvre-lès-Nancy Cedex, France. E-mail: mayank-shekhar.jha@univ-lorraine.fr

method. The paper shows that the WGD outperforms the GD in terms of mean absolute error, given an appropriate tuning of the WGD hyper-parameters, when applied to various nonlinear functions that are multi-modal in nature with several local/global minima.

The present section is followed by background details on the weighted linearization in Section 2 and the novel WGD approach along with corresponding pseudo-algorithm in Section 3. Section 4 presents the simulation study, followed by Section 5 which discusses the observed performance, and finally Section 6 which draws the main conclusions.

II. Weighted Linearization

The most common approach to obtain a linear approximation of a nonlinear function is to truncate the Taylor series so that only the zeroth and first order terms are kept. Let $f : \mathbb{R} \to \mathbb{R}$ and $\bar{x} \in \mathbb{R}$, then such an approximation is obtained as follows:

$$f(x) \approx f(\bar{x}) + \frac{df}{dx}(\bar{x})\,(x - \bar{x}) \tag{1}$$

In the multi-variable case, i.e., when $f : \mathbb{R}^n \to \mathbb{R}$, the derivative is replaced by the gradient:

$$\nabla f(x) \triangleq [\partial f(x)/\partial x_1, \partial f(x)/\partial x_2, \ldots, \partial f(x)/\partial x_n]^T \tag{2}$$

so that (1) becomes:

$$f(x) \approx f(\bar{x}) + \nabla f(\bar{x})(x - \bar{x}) \tag{3}$$

for a given $\bar{x} \in \mathbb{R}^n$. It is well-known that a geometric interpretation for (1) and (3) is that they describe the line and hyperplane, respectively, which are tangent to $f(x)$ at $x = \bar{x}$. It is in such sense that the linear approximation is considered to be valid only near $\bar{x}$, where *near* depends on the general behaviour of the nonlinear function about $\bar{x}$.

In [24], an alternative way to approximate the function $f(x)$ has been proposed and named *weighted linearization*, described hereafter in the multi-variable case. Let us consider a function $\rho : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, which will be referred to as *weighting function*, satisfying the following condition:

$$\int_{\mathbb{R}^n} \rho(x) d^n x = 1 \tag{4}$$

Then, we define as the *linear approximation of* $f(x)$ *weighted through* ρ the following function:

$$f_\rho(x) = f(\bar{x}) + \left(\int_{\mathbb{R}^n} \rho(x) \nabla f(x) d^n x \right) (x - \bar{x}) \tag{5}$$

Note that (5) is a generalization of (3), which can be recovered through a specific choice of the weighting function. In fact, by choosing:

$$\rho(x) = \delta(x - \bar{x}) \tag{6}$$

where $\delta(\cdot)$ denotes the multi-variable Dirac delta function, i.e., the measure defined in $\mathbb{R}^n$ such that:

$$\int_{\mathbb{R}^n} \delta(x) f(x) d^n x = f(0) \tag{7}$$

for every compactly supported continuous function f, then one obtains from (5):

$$\begin{aligned} f_\delta(x) &= f(\bar{x}) + \left(\int_{\mathbb{R}^n} \delta(x - \bar{x}) \nabla f(x) d^n x \right) (x - \bar{x}) \\ &= f(\bar{x}) + \nabla f(\bar{x})(x - \bar{x}) \end{aligned} \tag{8}$$

In rough words, a geometric interpretation of (5) is that it describes the hyperplane passing through $(\bar{x}, f(\bar{x}))$ but in general not tangent to $f(x)$ in that point, which captures the overall trend of the nonlinear function *about* $\bar{x}$, where the precise interpretation of *about* is expressed by the weighting function. As a matter of example, let us consider the function:

$$f(x) = x + x^3 \tag{9}$$

which can be approximated at the point $\bar{x} = 2$ via standard linearization as follows:

$$\begin{aligned} f(x) &\approx f(2) + \frac{df}{dx}(2)(x - 2) = 10 + 13(x - 2) \\ &= 13x - 16 \end{aligned} \tag{10}$$

On the other hand, if we choose the weighting function as a Gaussian function centred at $\bar{x}$ with RMS width σ, the weighted linearization becomes described by:

$$f(x) \approx f(2) + \int_{-\infty}^{+\infty} \exp\left(-\frac{(x-2)^2}{2\sigma^2} \right) (1 + 3x^2) dx \frac{x - 2}{\sigma\sqrt{2\pi}} \tag{11}$$

For example, using $\sigma = 1$, we obtain $f(x) \approx 16x - 22$, whereas with $\sigma = 2$, we obtain $f(x) \approx 25x - 40$.

The graphical representation of $f(x)$ and the obtained linear approximations are shown in Fig. 1, where the fact that the slope of $f(x)$ tends to grow faster to the right of $\bar{x}$ than it does to the left of $\bar{x}$ leads to the increased slope of the linearized function when the parameter σ is increased.

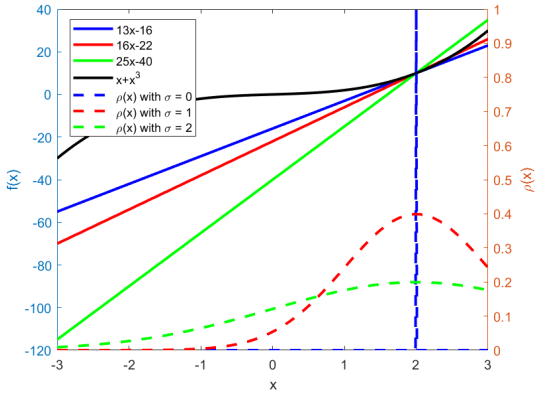


Fig. 1: $f(x) = x + x^3$ with its weighted linearization (Gaussian $\rho(x)$ centred at $\bar{x}$ with RMS width σ).

III. Weighted Gradient Descent

The main contribution of this paper is to adapt the gradient descent optimization algorithm to work under a computation of the gradient based on weighted linearization. In this way,

an alternative version of this popular first-order iterative optimization algorithm is obtained, referred to in the following as *weighted gradient descent* (WGD).

In the proposed WGD, with the goal of finding the minimum of a multi-variable function $f : \mathbb{R}^n \to \mathbb{R}$, one starts with a guess x_0 and a weighting function $\rho_0(x)$, and considers the sequence $x_0, x_1, x_2, \ldots$ such that:

$$x_{m+1} = x_m - \gamma_m \int_{\mathbb{R}^n} \rho_n(x) \nabla f(x) d^n x, \quad m \geq 0 \qquad (12)$$

where both the learning rate γ and the weighting function $\rho(x)$ are allowed to change at every iteration.

A basic intuition for this alternative algorithm, which is expressed in a pseudo-code form in Algorithm 1, can be provided by using the common analogy of a person trying to get down a mountain based on the observed steepness. In the standard GD algorithm, that person would use only the slope at its current position to decide in which direction to move further, a logic that would lead to wrongly deciding not to move further as soon as a plateau of any size, even infinitesimal, is reached. On the other hand, in the WGD, that person would look around and would use the slope at different points, weighted through the weighting function, to decide in which direction to move further.

By an appropriate choice of the weighting function, the weighted gradient algorithm can attenuate undesirable effects due to nonlinearities that can deteriorate the performance of gradient-based algorithms. As a matter of example, we show in Fig. 2 the case of the multi-modal scalar function $f(x) = x^2 + 1000 \sin(x)$. This function exhibits a derivative which bounces between positive and negative values (blue line in the figure, denoted as $\sigma = 0$), which may lead a gradient-based algorithm to getting stuck in oscillations around local minima. If we consider instead the derivative weighted through a Gaussian function with RMS width σ, the above-mentioned issues get attenuated (red and yellow lines in Fig. 2, corresponding to $\sigma = 1$ and $\sigma = 2$, respectively), thus facilitating the convergence of a gradient-based algorithm towards a global minimum.

IV. Results

In order to assess the performance of the WGD against the standard GD, we have considered different two-variable functions used for testing optimization algorithms, comprising bowl-shaped, plate-shaped, valley-shaped and multi-modal functions, which are described in detail below. For each function, once denoted the two variables as x and y, we have used 1000 initial conditions (x_0, y_0) distributed within a domain $\mathcal{X}_0 \times \mathcal{Y}_0$, with $\mathcal{X}_0$ and $\mathcal{Y}_0$ intervals defined below for each function.

We proceed to evaluating the performance of standard gradient descent versus weighted gradient descent for different values of the learning rate γ, which is kept constant between samples for simplicity. The termination conditions have been set to $\nabla_{\min} = 10^{-6}$, $m_{\max} = 10^4$ and $\Delta_{\min} = 10^{-6}$. At each sample, the weighting function is chosen as a square

Algorithm 1 Weighted gradient descent

Input: function to be optimised $f(x)$
Output: value of x where $f(x)$ is - hopefully - minimum
Choose an initial guess x_0, an initial weighting function $\rho_0(x)$, and an initial learning rate γ_0
Set current best value $x_{\text{best}} = x_0$
Set termination parameters:
    current iteration $m = 0$
    current gradient norm $\nabla_f = \infty$
    current step size $\Delta_x = \infty$
Set termination conditions:
    maximum number of iterations $m_{\max}$
    minimum gradient norm $\nabla_{\min}$
    minimum step size $\Delta_{\min}$
while $m < m_{\max}$ AND $\nabla_f > \nabla_{\min}$ AND $\Delta_x > \Delta_{\min}$ **do**
  Compute:
    $g_m = \int_{\mathbb{R}^n} \rho_m(x) \nabla f(x) d^n x$
    $x_{m+1} = x_m - \gamma_m g_m$
  Update termination parameters:
    $\nabla_f = \|g_m\|$
    $\Delta_x = x_{m+1} - x_m$
    $m = m + 1$
  Choose a new weighting function $\rho_m(x)$ and a new learning rate γ_m
  if $f(x_m) < f(x_{\text{best}})$ **then**
    $x_{\text{best}} = x_m$
  end if
end while
return x_{best}

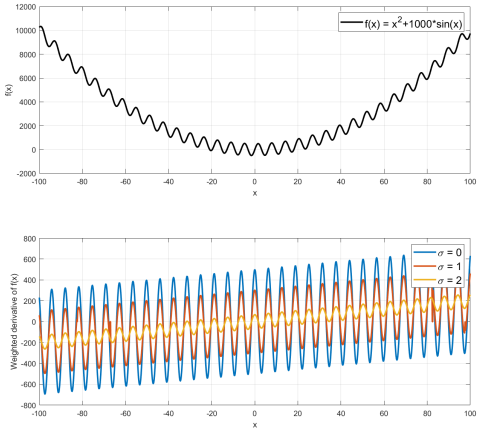


Fig. 2: Handling multi-modal functions via a weighted derivative.

distribution centred around the current (x_m, y_m), as follows:

$$\rho_m(x, y) = \begin{cases} \frac{1}{4b^2} & \text{if } (x, y) \in \mathbb{B} \\ 0 & \text{otherwise} \end{cases} \qquad (13)$$

with $\mathbb{B} = [x_m - b, x_m + b] \times [y_m - b, y_m + b]$, where $2b$ is the side of the square. Note that (13) tends to the multi-variable Dirac delta function centred at (x_m, y_m) when $b \to 0$.

For each function $f_i(x, y)$, learning rate γ, parameter of the weighting function b and initial condition (x_0, y_0), we run the gradient descent algorithm and compute the mean absolute error (MAE) of the best explored value of the

function under consideration. The MAEs are collected in contour plots, which show the performance obtained with different values of the learning rate γ and the hyperparameter b. The standard gradient descent is obtained when $b = 0$, so the benchmark for comparing GD versus WGD is given by the shades of gray taken in the lowest portion of the plots. In rough words, a value of b that corresponds to a darker shade of gray than the one appearing closer to the γ-axis indicates that the WGD behaves better than GD for those values of γ and b, whereas a lighter shade of gray signals a worsening in the performance. Finally, a red color is used to indicate values of γ and b that produce a NaN result, i.e. divergence of the algorithm.

The employed functions are described in the following subsections.

A. Bohachevsky function

The Bohachevsky function is bowl-shaped and described by the following expression:

$$f_1(x, y) = x^2 + 2y^2 - 0.3\cos(3\pi x) - 0.4\cos(4\pi y) + 0.7$$

which has a global minimum $f(x^*, y^*) = 0$, at $(x^*, y^*) = (0, 0)$. The domain for the initial conditions has been chosen as $(x_0, y_0) \in [-100, 100] \times [-100, 100]$. The weighted gradient has been computed as:

$$\nabla f_1(x, y, b) = \begin{bmatrix} 2x + \frac{0.15}{b}\left[\cos\left(3\pi(x - b)\right) - \cos\left(3\pi(x + b)\right)\right] \\ 4y + \frac{0.2}{b}\left[\cos\left(4\pi(y - b)\right) - \cos\left(4\pi(y + b)\right)\right] \end{bmatrix}$$

B. Zakharov function

The Zakharov function is plate-shaped and is described by the following expression:

$$f_2(x, y) = x^2 + y^2 + (0.5x + y)^2 + (0.5x + y)^4 \qquad (14)$$

which has a global minimum $f_2(x^*, y^*) = 0$, at $(x^*, y^*) = (0, 0)$. The domain for the initial conditions has been chosen as $(x_0, y_0) \in [-5, 10] \times [-5, 10]$. The weighted gradient is computed as:

$$\nabla f_2(x, y, b) = \begin{bmatrix} \left(2.5 + 1.3b^2\right)x + \left(1 + 2.5b^2\right)y \cdots \\ \qquad \cdots + 0.3x^3 + 1.5x^2y + 3xy^2 + 2y^3 \\ \left(1 + 2.5b^2\right)x + \left(4 + 5b^2\right)y \cdots \\ \qquad \cdots + 0.5x^3 + 3x^2y + 6xy^2 + 4y^3 \end{bmatrix}$$

C. Dixon-Price function

The Dixon-Price function is valley-shaped and described by the following expression:

$$f_3(x, y) = (x - 1)^2 + 2(2y^2 - x)^2 \qquad (15)$$

which has a global minimum $f_3(x^*, y^*) = 0$, at $(x^*, y^*) = (1, \sqrt{2}/2)$. The domain for the initial conditions has been chosen as $(x_0, y_0) \in [-10, 10] \times [-10, 10]$. The weighted gradient is computed as:

$$\nabla f_3(x, y, b) = \begin{bmatrix} 6x - 8y^2 - 2 - \frac{8}{3}b^2 \\ 32b^2y - 16xy + 32y^3 \end{bmatrix}$$

D. Rosenbrock function

The Rosenbrock function is valley-shaped and described by the following expression:

$$f_4(x, y) = 100\left(y - x^2\right)^2 + (x - 1)^2 \qquad (16)$$

which has a global minimum $f_4(x^*, y^*) = 0$ at $(x^*, y^*) = (1, 1)$, which is difficult to converge to using gradient-based optimization algorithms [26]. The domain for the initial condition has been chosen as $(x_0, y_0) \in [-5, 10] \times [-5, 10]$. The weighted gradient is computed as:

$$\nabla f_4(x, y, b) = \begin{bmatrix} (400b^2 + 2)x + 400x^3 - 400xy - 2 \\ 200\left(y - x^2 - \frac{b^2}{3}\right) \end{bmatrix}$$

E. Beale function

The Beale function is multimodal and described by the following expression:

$$\begin{aligned} f_5(x, y) = &(1.5 - x + xy)^2 + \left(2.3 - x + xy^2\right)^2 \\ &+ \left(2.625 - x + xy^3\right)^2 \end{aligned} \qquad (17)$$

which has a global minimum $f_5(x^*, y^*) = 0$ at $(x^*, y^*) = (3, 0.5)$. The domain for the initial condition has been chosen as $(x_0, y_0) \in [-4.5, 4.5] \times [-4.5, 4.5]$. The weighted gradient is computed as:

$$\nabla f_5(x, y, b) = \begin{bmatrix} \frac{2b^6x}{7} + 6b^4xy^2 + \frac{2b^4x}{5} \cdots \\ \cdots + 10b^2xy^4 + 4b^2xy^2 \cdots \\ \cdots - 4b^2xy - \frac{2b^2x}{3} + 5.3b^2y \cdots \\ \cdots + 1.5b^2 + 2xy^6 + 2xy^4 \cdots \\ \cdots - 4xy^3 - 2xy^2 - 4xy + 6x \cdots \\ \cdots + 5.3y^3 + 4.5y^2 + 3y - 12.8 \\ \cdots 2b^6y + 6b^4x^2y + \frac{20b^4y^3}{3} + \frac{4b^4y}{3} \cdots \\ \cdots - \frac{2b^4}{3} + 20b^2x^2y^3 + 4b^2x^2y \cdots \\ \cdots - 2b^2x^2 + 5.3b^2x + 2b^2y^5 + \frac{4b^2y^3}{3} \cdots \\ \cdots - 2b^2y^2 - \frac{2b^2y}{3} - \frac{2b^2}{3} \cdots \\ \cdots + 6x^2y^5 + 4x^2y^3 - 6x^2y^2 \cdots \\ \cdots - 2x^2y - 2x^2 + 15.8xy^2 + 9xy + 3x \end{bmatrix}$$

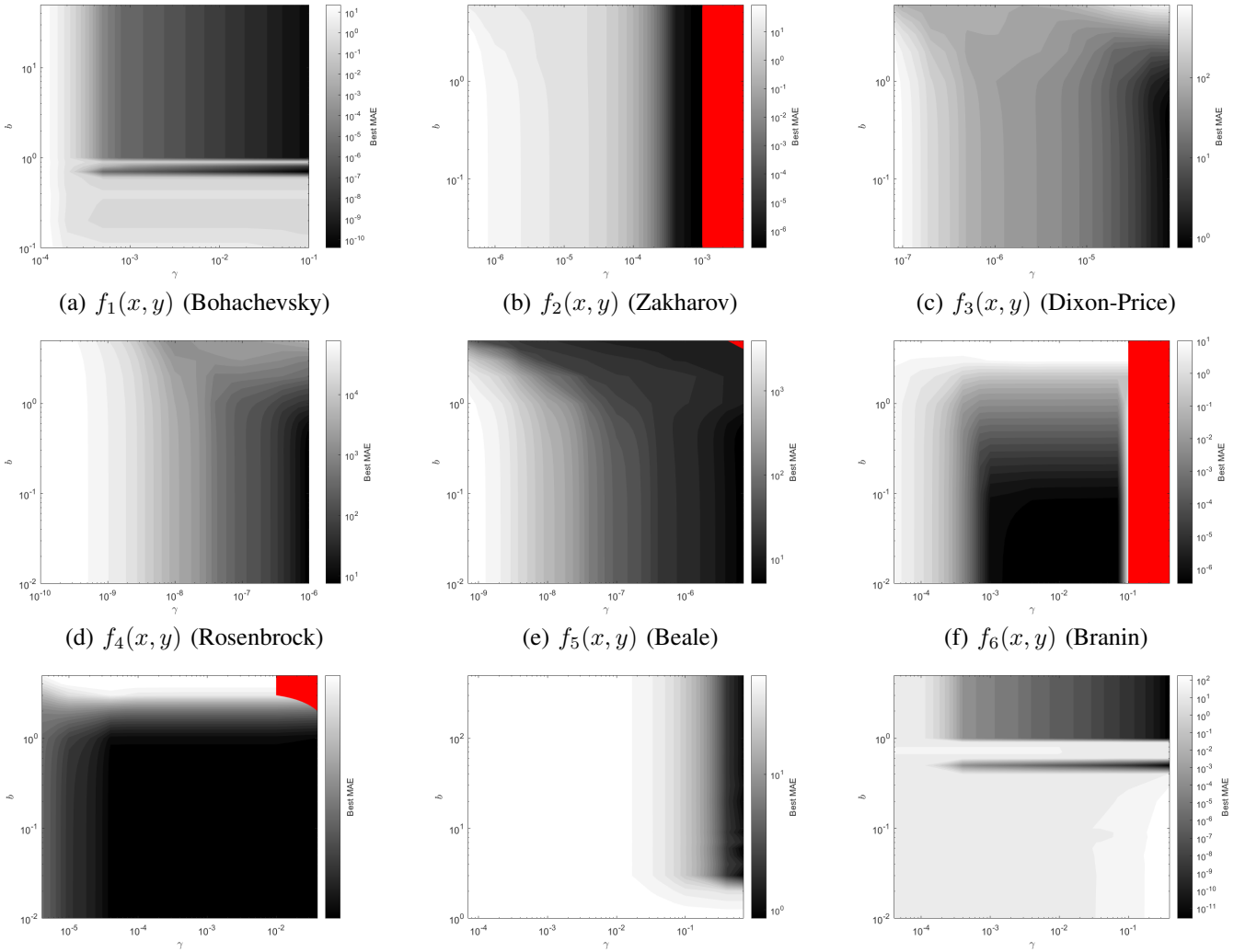
F. Branin function

The Branin function has three global minima and is described by the following expression:

$$f_6(x, y) = \left(y - \frac{5.1x^2}{4\pi^2} + \frac{5x}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x) + 9.6 \qquad (18)$$

with three global minima $f_6(x^*, y^*) = 0$ at $(x^*, y^*) = (-\pi, 12.3)$, $(x^*, y^*) = (\pi, 2.3)$ and $(x^*, y^*) = (9.4, 2.5)$. The domain for the initial conditions has been chosen as $(x_0, y_0) \in [-5, 10] \times [0, 15]$. The weighted gradient is computed as:

$$\nabla f_6(x, y, b) = \begin{bmatrix} 0.2(b^2x + x^3) - \frac{51}{4\pi^3}\left(b^2 + 3x^2\right) \cdots \\ \cdots + \frac{5\sin(b)\sin(x)}{b}\left(\frac{1}{4\pi} - 2\right) - 0.52xy \cdots \\ \cdots + \frac{10}{\pi}\left(\frac{5x}{\pi} + y - 6\right) + 3.1x \\ 2\left(\frac{5x}{\pi} + y - 6\right) - \frac{17}{20\pi^2}\left(b^2 + 3x^2\right) \end{bmatrix}$$



(a) $f_1(x, y)$ (Bohachevsky) (b) $f_2(x, y)$ (Zakharov) (c) $f_3(x, y)$ (Dixon-Price)

(d) $f_4(x, y)$ (Rosenbrock) (e) $f_5(x, y)$ (Beale) (f) $f_6(x, y)$ (Branin)

(g) $f_7(x, y)$ (Styblinski-Tang) (h) $f_8(x, y)$ (Griewank) (i) $f_9(x, y)$ (Rastrigin)

Fig. 3: MAE of the best explored value

G. Styblinski-Tang function

The Styblinski-Tang function is described by the following expression:

$$f_7(x, y) = \frac{1}{2}\left(x^4 + y^4 - 16x^2 - 16y^2 + 5x + 5y\right) + 78.3 \quad (19)$$

which has the global minimum $f(x^*, y^*) = 0$ at $(x^*, y^*) = (-2.9, -2.9)$. The domain for the initial conditions has been chosen as $(x_0, y_0) \in [-5, 5] \times [-5, 5]$. The weighted gradient is computed as:

$$\nabla f_7(x, y, b) = \begin{bmatrix} 2b^2x + 2x^3 - 16x + 2.5 \\ 2b^2y + 2y^3 - 16y + 2.5 \end{bmatrix}$$

H. Griewank function

The Griewank function has many regularly distributed local minima. It is described by the following expression:

$$f_8(x, y) = \frac{x^2 + y^2}{4000} - \cos(x)\cos\left(\frac{y}{\sqrt{2}}\right) + 1 \quad (20)$$

which has the global minimum $f_8(x^*, y^*) = 0$ at $(x^*, y^*) = (0, 0)$. The domain for the initial condition has been chosen as $(x_0, y_0) \in [-600, 600] \times [-600, 600]$. The weighted gradient is computed as:

$$\nabla f_8(x, y, b) = \begin{bmatrix} \frac{b^2x + 2000\sqrt{2}\sin(b)\sin(\frac{b}{\sqrt{2}})\sin(x)\cos(\frac{y}{\sqrt{2}})}{2000b^2} \\ \frac{b^2y + 2000\sin(b)\sin(\frac{b}{\sqrt{2}})\cos(x)\sin(\frac{y}{\sqrt{2}})}{2000b^2} \end{bmatrix}$$

I. Rastrigin

The Rastrigin function is highly multimodal and has several regularly distributed local minima. It is described by the following expression:

$$f_9(x, y) = 20 + x^2 + y^2 - 10\cos(2\pi x) - 10\cos(2\pi y) \quad (21)$$

which has the global minimum $f_9(x^*, y^*) = 0$ at $(x^*, y^*) = (0, 0)$. The domain for the initial conditions has been chosen as $(x_0, y_0) \in [-5.12, 5.12] \times [-5.12, 5.12]$. The weighted gradient is computed as:

$$\nabla f_{14}(x, y, b) = \begin{bmatrix} 2x + \frac{10\sin(2\pi x)\sin(2\pi b)}{b} \\ 2y + \frac{10\sin(2\pi y)\sin(2\pi b)}{b} \end{bmatrix}$$

V. Discussion

The functions chosen in this study exhibit diverse attributes such as bowl and valley shapes, and include mono-modal and multi-modal functions with one or multiple distributed minima (global/local). Overall, with the exception of Branin function (Fig. 3f) and Styblinski-Tang function (Fig. 3g), the proposed WGD outperforms the standard GD if the learning rate and weighting function length (span) are suitably chosen. For instance, in face of appropriately chosen length $b \in [0, 0.9]$ and with learning rate remaining not too high or too low, the WGD leads to an MAE that is less or as good as that obtained via standard GD approach. As such, weighting function parameter(s) and learning rate constitute the so called hyper-parameter set and need to be fine tuned to the objective function at hand. Indeed, presence of weighting function such as a square enlarges the scope of the optimizer by presenting a set of weighted possibilities at each step, and leads to an assessment of the data (objective function topology) at a finer granularity. However, if the size of such a function is very large (for example, $b > 1$) then, WGD is not necessarily able to perform well plausibly due to presence of several nonlinear typologies (for example, valleys) within the weighting function range leading to a relatively much coarse assessment. For instance, Rosenbrock function which is difficult to converge with standard GD, WGD performs generally much better with less learning rate value and value of weighting function parameter that is not very large (Fig. 3d).

It is worth noting that in face of highly multi-modal functions with several local/global minima, WGD is able to demonstrate a better performance overall over a large spectrum values of λ, b.

VI. Conclusions

We have introduced a novel gradient-based optimization approach based on the weighted linearization. The proposed weighted gradient descent is able to attenuate undesirable effects due to nonlinearities that typically deteriorate the performance of gradient-based algorithms. Different two-variable functions comprising bowl-shaped, plate-shaped, valley-shaped and multi-modal functions have been used to assess the performance of the proposed approach. The results have shown that in many cases the weighted gradient descent outperforms the standard gradient descent if the hyper-parameters are properly tuned.

The discussion in this work is limited to a specific choice of the weighting function, i.e., a square distribution centred around the current point. Future work will be devoted to a deeper study of how to perform a good selection of the weighting function, possibly taking into account the local features of the nonlinearities. Additional paths for future research concern how the weighted gradient descent algorithm can be modified to incorporate stochastic elements and to explore its potential for applications in machine learning.

References

[1] M. Vidyasagar. *Nonlinear systems analysis*. SIAM, 2002.

[2] J. Zhou, C. Wen, and Y. Zhang. Adaptive output control of nonlinear systems with uncertain dead-zone nonlinearity. *IEEE Transactions on Automatic Control*, 51(3):504–511, 2006.

[3] H. Yang, B. Jiang, and V. Cocquempot. A survey of results and perspectives on stabilization of switched nonlinear systems with unstable modes. *Nonlinear Analysis: Hybrid Systems*, 13:45–60, 2014.

[4] D. Bertsimas, J. Dunn, and Y. Wang. Near-optimal nonlinear regression trees. *Operations Research Letters*, 49(2):201–206, 2021.

[5] T. Kailath. *Linear systems*, volume 156. Prentice-Hall Englewood Cliffs, NJ, 1980.

[6] R. Misener and C. A. Floudas. Global optimization of mixed-integer quadratically-constrained quadratic programs (miqcqp) through piecewise-linear and edge-concave relaxations. *Mathematical Programming*, 136(1):155–182, 2012.

[7] F. Casas and A. Iserles. Explicit magnus expansions for nonlinear equations. *Journal of Physics A: Mathematical and General*, 39(19):5445, 2006.

[8] Y. Yao, B. Yang, F. He, Y. Qiao, and D. Cheng. Attitude control of missile via fliess expansion. *IEEE Transactions on Control Systems Technology*, 16(5):959–970, 2008.

[9] H.J. Sussmann. Lie brackets and local controllability: a sufficient condition for scalar-input systems. *SIAM Journal on Control and Optimization*, 21(5):686–713, 1983.

[10] K. Beauchard, J. Le Borgne, and F. Marbach. On expansions for nonlinear systems error estimates and convergence issues. *Comptes Rendus. Mathématique*, 361(G1):97–189, 2023.

[11] W.H. Foy. Position-location solutions by Taylor-series estimation. *IEEE transactions on aerospace and electronic systems*, (2):187–194, 1976.

[12] G. Torrisi, S. Grammatico, R.S. Smith, and M. Morari. A projected gradient and constraint linearization method for nonlinear model predictive control. *SIAM Journal on Control and Optimization*, 56(3):1968–1999, 2018.

[13] M. Nikolova and P. Tan. Alternating structure-adapted proximal gradient descent for nonconvex nonsmooth block-regularized problems. *SIAM Journal on Optimization*, 29(3):2053–2078, 2019.

[14] B. Jin, Z. Zhou, and J. Zou. On the convergence of stochastic gradient descent for nonlinear ill-posed problems. *SIAM Journal on Optimization*, 30(2):1421–1450, 2020.

[15] X.Y. Han and A.S. Lewis. Survey descent: A multipoint generalization of gradient descent for nonsmooth optimization. *SIAM Journal on Optimization*, 33(1):36–62, 2023.

[16] D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] H. Schaeffer and S.G. McCalla. Extending the step-size restriction for gradient descent to avoid strict saddle points. *SIAM Journal on Mathematics of Data Science*, 2(4):1181–1197, 2020.

[18] D. Soydaner. A comparison of optimization algorithms for deep learning. *International Journal of Pattern Recognition and Artificial Intelligence*, 34(13):2052013, 2020.

[19] J.-N. Juang. *Applied system identification*. Prentice-Hall, Inc., 1994.

[20] J. Sirignano and K. Spiliopoulos. Stochastic gradient descent in continuous time. *SIAM Journal on Financial Mathematics*, 8(1):933–961, 2017.

[21] H. Garnier, L. Wang, and P.C. Young. *Direct identification of continuous-time models from sampled data: Issues, basic solutions and relevance*. Springer, 2008.

[22] R.S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

[23] G.C. Andrei, M.S. Jha, and D. Theillol. Complementary reward function based learning enhancement for deep reinforcement learning. In *European Workshop on Advanced Control and Diagnosis*, pages 237–247. Springer, 2022.

[24] D. Rotondo. Weighted linearization of nonlinear systems. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 69(7):3239–3243, 2021.

[25] D. Rotondo. The weighted kalman filter. *IFAC World Congress*, 2023.

[26] V. Picheny, T. Wagner, and D. Ginsbourger. A benchmark of kriging-based infill criteria for noisy optimization. *Structural and multidisciplinary optimization*, 48:607–626, 2013.