Trabalho apresentado no XLI CNMAC, Unicamp - Campinas - SP, 2022.

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics
Preprint

On the Delayed Weighted Gradient Method with Simultaneous Step-Size Search

Hugo Lara Urdaneta [1]
Federal University of Santa Catarina, Department of Control and Automation Engineering and Computing, Blumenau, Brazil
Rafael Aleixo [2]
Federal University of Santa Catarina, Department of Mathematics, Blumenau, Brazil

Abstract. In this article it is presented a two step first order algorithm, based on bidimensional minimization, to deal with convex quadratic optimization problems. Our analysis show linear convergence and A-orthogonality of the gradient iterates. Numerical experimentation show the effectiveness of our method.

Keywords. Gradient methods, convex quadratic optimization, Krylov subspace methods, DWGM.

1 Introduction

The gradient methods play a key role in optimization techniques. Cauchy developed the first methodology for unconstrained optimization, the well-known Steepest Descent Method [4]. Two features o the method are worth mentioning. Simplicity and low cost per iteration where only gradient information is required is the first one, while a very slow rate of convergence is the second one. These two conflicting characteristics motivate a great amount of work in the attempt of balancing them by using only gradient information at the same time that its convergence is accelerated. Low cost gradient methods that have widely been effective were proposed in literature (see for instance [7, 14] and references therein). The gradient methods for unconstrained minimization problem

$$\text{minimize}_{x\in\mathbb{R}^n} \quad f(x)$$

generate a sequence of solution approximations x_k satisfying $x_{k+1} = x_k - \alpha_k g_k$ where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable, $g_k = \nabla f(x_k)$ and $\alpha_k > 0$. The selection of the step length α_k depends on the chosen method. Among the choices we shall mention the classical SD (steepest descent) which was proposed by Cauchy to solve nonlinear systems of equations. In this case,

$$\alpha_k^{SD} = \text{argmin}_\alpha f(x_k - \alpha g_k). \tag{1}$$

Instead of minimizing the objective function, the minimum gradient step length (MG) aims to minimize the norm of the gradient at the next step

$$\alpha_k^{MG} = \text{argmin}_\alpha \|\nabla f(x_k - \alpha g_k)\|_2. \tag{2}$$

Assuming the objective function f to be a strictly convex quadratic function, that is, for a symmetric and positive definite (SPD) matrix $A \in \mathbb{R}^{n\times n}$, the unconstrained optimization problem becomes

$$\text{minimize}_{x\in\mathbb{R}^n} \quad f(x) = \frac{1}{2}x^T Ax - b^T x. \tag{3}$$

[1] hugo.lara.urdaneta@ufsc.br
[2] rafael.aleixo@ufsc.br

Under this assumption, simple calculations on (1) and (2) give

$$\alpha_k^{SD} = \frac{g_k^T g_k}{g_k^T A g_k} \quad \text{and} \quad \alpha_k^{MG} = \frac{g_k^T A g_k}{g_k^T A^2 g_k}.$$

In order to overcome the slow convergence of gradient methods with the former step lengths, several other choices were proposed in literature, for instance [3, 9–11].

Recently, Oviedo-Leon [13] proposed the Delayed Weighted Gradient Method (DWGM) with the same objective, to overcome the poor performance of the Gradient Methods. DWGM is a two-step gradient method that combines smoothing and delaying techniques to accelerate the convergence, while avoiding the well-known zigzagging behaviour of the gradient methods. Each of the two DWGM step sizes are calculated sequentially, so the first step-size information is necessary to calculate the second one. Andreani and Raydan [1] demonstrated several important properties of DWGM, including the finite termination of the method, in exact arithmetics. In short, DWGM can outperform the conjugate gradient method [1, 13] and, therefore, it is a candidate method for practical problems. In this article we develop a two-step gradient method where both step-sizes are simulteneously calculated as optimal solutions of a bidimensional optimization problem.

Now we briefly describe the DWGM algorithm [13], and establish some of its properties. We consider the strictly convex quadratic minimization problem (3). Since the gradient $g(x) \equiv \nabla f(x) = Ax - b$, then the unique global solution $A^{-1}b$ for the problem (3) also solves the linear system $Ax = b$. For large n, many low cost iterative methods have been proposed and analyzed. The so-called gradient type methods emerge as competitive choices since they show fast linear convergence (see [2, 5, 7, 12]).

From a starting point $x_0 \in \mathbb{R}^n$, consider $g_k = g(x_k)$. The well-known minimum gradient method (see [2]) is given by the iteration $x_{k+1} = x_k - \alpha_k^{MG} g_k$, where $\alpha_k^{MG} = g_k^T w_k / \|w_k\|_2^2$ and $w_k = Ag_k$. Here, the step-size is defined as $\alpha_k = \operatorname{argmin}_{\alpha>0} \|\nabla f(x_k - \alpha g_k)\|_2$. It is easy to check that $\alpha_k = \operatorname{argmin}_{\alpha>0} \|g_k - \alpha w_k\|_2$, which leads to the expression above. The minimum gradient norm method calculates the next iterate as the point alongside the current gradient at which the norm of the next gradient is minimized. As a two step gradient method, DWGM incorporates a delaying step defined as follows [13]: The first stage uses the ordinary minimum gradient point

$$y_k = x_k - \alpha_k^{MG} g_k.$$

Then, calculates the next iterate as

$$x_{k+1} = x_{k-1} + \beta_k (y_k - x_{k-1}),$$

where $\beta_k = g_{k-1}^T (g_{k-1} - r_k) / \|g_{k-1} - r_k\|_2^2$. The step size is defined by

$$\beta_k = \operatorname{argmin}_{\beta \in \mathbb{R}} \|\nabla f(x_{k-1} + \beta(y_k - x_{k-1}))\|_2.$$

It is straightforward to see that $\nabla f(x_{k-1} + \beta(y_k - x_{k-1})) = g_{k-1} - \beta(g_{k-1} - r_k)$, for $r_k = g_k - \alpha_k w_k$. This leads to $\beta_k = \operatorname{argmin}_{\beta \in \mathbb{R}} \|g_{k-1} - \beta(g_{k-1} - r_k)\|_2 = g_{k-1}^T (g_{k-1} - r_k) / \|g_{k-1} - r_k\|_2^2$. By merging the definition of y_k into x_{k+1} and simple manipulation, the next iterate can be rewritten as $x_{k+1} = (1 - \beta_k) x_{k-1} + \beta_k x_k - \beta_k \alpha_k g_k$.

Some of the properties that DWGM enjoys, established in [1, 13] include the non negativity of β_k for all k, the monotonic decreasing of $\{\|g_k\|_2\}$ as well as the q-linear convergence of $\{g_k\}$ to zero when k goes to infinity (which implies that $\{x_k\}$ converges to the unique global minimizer of f), and finite convergence by using A-orthogonality of the gradient vector at the current iteration with all previous gradient vectors.

The remainder of the article is organized as follows: In the next section we describe our two step-size gradient method, and analyze the convergence. Section 3 is devoted to numerical experimentation, and at the last section some concluding remarks are offered.

2 Another Two Step-Size Gradient Method

In this section we develop our algorithm defining step sizes by wielding optimization arguments. Consider the problem (3) and iterates x_{k-1}, x_k. Let us denote $g_k = Ax_k - b$, $w_k = Ag_k$, $y_k(\alpha) = x_k - \alpha g_k$ and $r_k(\alpha) = g_k - \alpha w_k$. To build our iteration, like DWGM, we define the delaying step by $x_{k+1}(\alpha, \beta) = x_{k-1} + \beta(y_k(\alpha) - x_{k-1})$. We choose algorithmic values for α and β in a way that $\theta(\alpha, \beta) := \|\nabla f(x_{k+1}(\alpha, \beta))\|^2$ is minimized. To this aim, observe that since

$$\nabla f(x_{k+1}(\alpha, \beta)) = g_{k-1} + \beta(r_k(\alpha) - g_{k-1}), \tag{4}$$

we have $\theta(\alpha, \beta) = \|g_{k-1} + \beta(r_k(\alpha) - g_{k-1})\|^2$, and by the first order optimality conditions, we obtain

$$\frac{\partial \theta}{\partial \alpha} = -2[g_{k-1} + \beta(r_k(\alpha) - g_{k-1})]^T w_k = 0, \tag{5}$$

$$\frac{\partial \theta}{\partial \beta} = 2[g_{k-1} + \beta(r_k(\alpha) - g_{k-1})]^T (r_k(\alpha) - g_{k-1}) = 0. \tag{6}$$

From the equation (5) we get $g_{k-1}^T w_k + \beta(g_k - g_{k-1})^T w_k - \alpha\beta w_k^T w_k = 0$ obtaining

$$\alpha\beta = \frac{g_{k-1}^T w_k}{w_k^T w_k} + \beta \frac{(g_k - g_{k-1})^T w_k}{w_k^T w_k}, \tag{7}$$

and so,

$$\alpha = \frac{g_{k-1}^T w_k}{\beta w_k^T w_k} + \frac{(g_k - g_{k-1})^T w_k}{w_k^T w_k}. \tag{8}$$

The equation (6) leads to

$$g_{k-1}^T(g_k - g_{k-1}) + \beta(g_k - g_{k-1})^T(g_k - g_{k-1}) - \alpha g_{k-1}^T w_k - 2\alpha\beta(g_k - g_{k-1})^T w_k + \alpha^2\beta w_k^T w_k = 0. \tag{9}$$

For simplicity, denote $b = g_{k-1}^T(g_k - g_{k-1})$, $c = g_{k-1}^T w_k$, $d = \|g_k - g_{k-1}\|_2^2$, $e = (g_k - g_{k-1})^T w_k$, and $f = \|w_k\|_2^2$, so equation (9) becomes

$$b - \alpha c + \beta d - 2\alpha\beta e + \alpha^2\beta f = 0. \tag{10}$$

The expressions (7) and (8) written in the notation above lead to $\alpha\beta = \frac{c}{f} + \beta\frac{e}{f}$ and $\alpha = \frac{c}{\beta f} + \frac{e}{f}$, and merging these expressions in (10) we obtain for $\beta \neq 0$

$$\left(b - \frac{ce}{f}\right) + \beta\left(d - \frac{e^2}{f}\right) = 0,$$

which implies that $\beta = \frac{ce - bf}{df - e^2}$. Now, by merging it in (8), we obtain $\alpha = \frac{cd - be}{ce - bf}$. Returning to the original notation, and using $p_k = g_k - g_{k-1}$ we have

$$\alpha_k = \frac{g_{k-1}^T w_k \|p_k\|_2^2 - g_{k-1}^T p_k p_k^T w_k}{g_{k-1}^T w_k p_k^T w_k - g_{k-1}^T p_k \|w_k\|_2^2} \qquad \text{and} \qquad \beta_k = \frac{g_{k-1}^T w_k p_k^T w_k - g_{k-1}^T p_k \|w_k\|_2^2}{\|p_k\|_2^2 \|w_k\|_2^2 - (p_k^T w_k)^2}.$$

We call the algorithm bidimensional delayed weighted gradient method or BiDWGM. We summarize it in Algorithm 1. Below we present the convergence analysis and some properties of BiWDGM.

Algorithm 1 BiDWGM

Require: $A \in \mathbb{R}^{n\times n}$ SPD, $x_0 \in \mathbb{R}^n$, x_{-1}, $g_{-1} = g(x_{-1})$, $\epsilon > 0$.

$w_{-1} = Ag_{-1}$; $x_0 = x_{-1} - \dfrac{g_{-1}^T w_{-1}}{\|w_{-1}\|_2} g_{-1}$; $g_0 = g(x_0)$;
$k = 0$;
while $\|g_k\|_2 > \epsilon$ **do**
  $w_k = Ag_k$; $p_k = g_k - g_{k-1}$;
  $b_k = g_{k-1}^T p_k$; $c_k = g_{k-1}^T w_k$; $d_k = p_k^T p_k$; $e_k = p_k^T w_k$; $f_k = w_k^T w_k$;
  $\alpha_k = \dfrac{c_k d_k - b_k e_k}{c_k e_k - b_k f_k}$; $\beta_k = \dfrac{c_k e_k - b_k f_k}{d_k f_k - e_k^2}$;
  $y_k = x_k - \alpha_k g_k$; $r_k = g_k - \alpha_k w_k$;
  $x_{k+1} = x_{k-1} + \beta_k(y_k - x_{k-1})$; $g_{k+1} = g_{k-1} + \beta_k(r_k - g_{k-1})$;
  $k = k + 1$;
end while

Lemma 2.1. *Let $\{x_k\}$ be a sequence generated by the Algorithm 1. Then $\{\|g_k\|\}$ is a monotonically decreasing sequence.*

Proof. First observe from (4) that $g_{k+1}(\alpha_k, \beta_k) = g_{k+1}$, $g_{k+1}(0,1) = g_k$ and $g_{k+1}(\alpha, 1) = r_k(\alpha)$. Then, the optimality of the pair (α_k, β_k) implies $\|g_{k+1}(\alpha_k, \beta_k)\| \leq \|g_{k+1}(\alpha, 1)\|$, for any α. In particular, for $\alpha_k^{MG} = \frac{g_k^T w_k}{w_k^T w_k}$ we obtain

$$\|g_{k+1}\| \leq \|r_k(\alpha_k^{MG})\|.$$

On the other hand, we can prove that

$$\|r_k(\alpha_k^{MG})\|^2 = \|g_k - \alpha_k^{MG} w_k\|^2 = \|g_k\|^2 - \frac{(g_k^T w_k)^2}{\|w_k\|^2} < \|g_k\|^2$$

because $g_k^T w_k > 0$. This leads to

$$\|g_{k+1}\| < \|g_k\| \tag{11}$$

that is, $\{\|g_k\|\}$ decreases monotonously. □

Lemma 2.2. *Let β_k the parameter defined in Algorithm 1. Then for each $k \in \mathbb{N}$,*

$$0 < \beta_k \leq \frac{1}{2}\left(1 + \frac{\|g_{k-1}\|^2}{\|g_{k-1} - r_k(\alpha_k)\|^2}\right)$$

Proof. We first see the non negativity of β_k. From (6) we have

$$g_{k-1}^T r_k(\alpha_k) = \|g_{k-1}\|^2 - \beta_k \|g_{k-1} - r_k(\alpha_k)\|^2. \tag{12}$$

It follows from Cauchy-Schwarz inequality and Lemma 2.1 and (11) that

$$g_{k-1}^T r_k(\alpha_k) \leq \|g_{k-1}\|\|r_k(\alpha_k)\| < \|g_{k-1}\|\|g_k\| < \|g_{k-1}\|^2.$$

In view of the last expression and (12) we obtain $\beta_k > 0\ \forall k \in \mathbb{N}$. Finally, by using the well-known inequality $u^T v \leq \frac{1}{2}(\|u\|^2 + \|v\|^2)$ we arrive at

$$\beta_k \leq \frac{1}{2}\left(1 + \frac{\|g_{k-1}\|^2}{\|g_{k-1} - r_k(\alpha_k)\|^2}\right)$$

which proves the lemma. □

DOI: 10.5540/03.2022.009.01.0288

Lemma 2.3. *In Algorithm 1, it follows for* $k = 0, 1, 2, \ldots$

1. $r_k(\alpha_k)^T A g_k = \left(1 - \frac{1}{\beta_k}\right) g_{k-1}^T A g_k$.
2. $r_k(\alpha_k)^T r_k(\alpha_k) - r_k(\alpha_k)^T g_k = -\alpha_k \left(1 - \frac{1}{\beta_k}\right) g_{k-1}^T A g_k$.
3. $g_{k+1}^T r_k(\alpha_k) = g_{k+1}^T g_{k-1}$.
4. $g_{k+1}^T A g_k = 0$.

Proof. 1. For steps 4 and 7 of Algorithm 1, and (8) we have

$$r_k(\alpha_k)^T A g_k = (g_k - \alpha_k w_k)^T w_k = g_k^T w_k - \left[\frac{g_{k-1}^T w_k}{\beta_k} + (g_k - g_{k-1})^T w_k\right] = \left(1 - \frac{1}{\beta_k}\right) g_{k-1}^T w_k.$$

2. Again by step 7 of the algorithm, and item 1, we get

$$r_k(\alpha_k)^T (r_k(\alpha_k) - g_k) = -\alpha_k r_k(\alpha_k)^T w_k = -\alpha_k \left(1 - \frac{1}{\beta_k}\right) g_{k-1}^T w_k.$$

3. By (6) and step 8 of the algorithm we obtain

$$\begin{aligned} g_{k+1}^T (r_k(\alpha_k) - g_{k-1}) &= [g_{k-1} + \beta_k (r_k(\alpha_k) - g_{k-1})]^T (r_k(\alpha_k) - g_{k-1}) \\ &= g_{k-1}^T (r_k(\alpha_k) - g_{k-1}) + \beta_k \|r_k(\alpha_k) - g_{k-1}\|^2 = 0. \end{aligned}$$

4. From item 1 and step 8 we obtain

$$\begin{aligned} g_{k+1}^T A g_k &= [g_{k-1} + \beta_k (r_k(\alpha_k) - g_{k-1})]^T w_k = (1 - \beta_k) g_{k-1}^T w_k + \beta_k r_k(\alpha_k)^T w_k \\ &= (1 - \beta_k) g_{k-1}^T w_k + \beta_k \left(1 - \frac{1}{\beta_k}\right) g_{k-1}^T w_k = 0. \end{aligned}$$

□

Theorem 2.1. *Let* $\{x_k\}$ *be a sequence generated by Algorithm 1, and* $\lambda_1, \lambda_2, \ldots, \lambda_n > 0$ *the eigenvalues of the matrix square root of* A *(i.e.* $A^{1/2}$*). Then the sequence* $\{g_k\}$ *converges to zero Q linearly with convergence factor* $\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}$.

Proof. From Lemma 2.1 we have that $\{\|g_k\|\}$ is monotonically decreasing, and bounded by zero. Then, $\{\|g_k\|\}$ is convergent. On the other hand, note that, from (8)

$$\|r_k(\alpha_k)\|^2 = \|g_k\|^2 - 2\alpha_k g_k^T w_k + \alpha_k^2 \|w_k\|^2 = \|g_k\|^2 - \frac{(g_k^T w_k)^2}{\|w_k\|^2}.$$

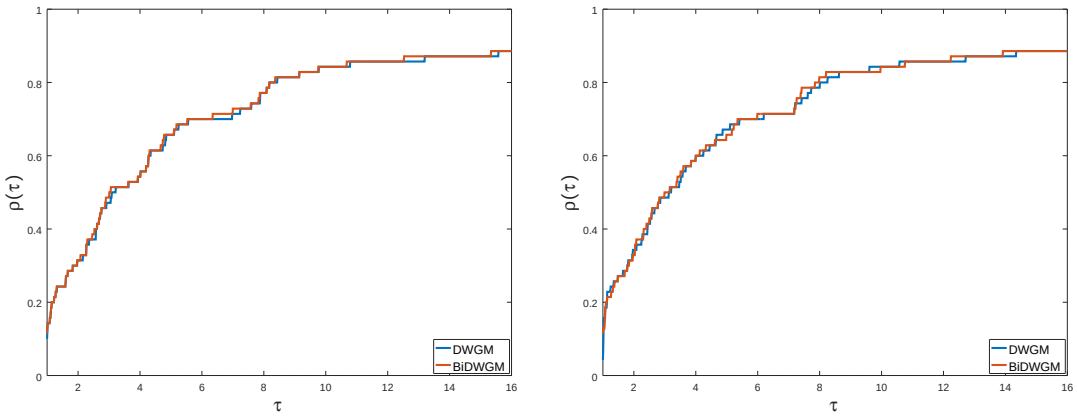
The last equality comes from the part 4 of the lemma above. We can write

$$\|r_k(\alpha_k)\|^2 = \|g_k\|^2 - \frac{(g_k^T w_k)}{\|w_k\|^2} \frac{(g_k^T w_k)}{\|g_k\|^2} \|g_k\|^2 = \left[1 - \frac{(g_k^T w_k)}{\|w_k\|^2} \frac{(g_k^T w_k)}{\|g_k\|^2}\right] \|g_k\|^2.$$

We denote $v_k = A^{-1/2} g_k$ and rewrite

$$\frac{(g_k^T w_k)}{\|w_k\|^2} \frac{(g_k^T w_k)}{\|g_k\|^2} = \frac{(v_k^T v_k)^2}{(v_k^T A v_k)(v_k^T A^{-1} v_k)}.$$

By using the Kantorovich inequality to this expression, and merging the result in the former we get $\|r_k(\alpha_k)\| \leq \left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}\right) \|g_k\|$. Now noting that, from (11) we have $\|g_{k+1}\| \leq \|r_k(\alpha_k)\|$, we immediately conclude that $\{g_k\}$ converges to zero Q-linearly with convergence factor $\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}$ and hence, since A is positive definite, we also conclude that $\{x_k\}$ tends to the unique minimizer of f when k goes to infinity. □



(a) Iteration. (b) CPU time.

Figure 1: Performance profiles.

Remark: From part 4 of Lemma 2.3, $g_{k-1}^T w_k = 0$. It means that,

$$\alpha_k = \frac{g_{k-1}^T w_k \|p_k\|_2^2 - g_{k-1}^T p_k p_k^T w_k}{g_{k-1}^T w_k p_k^T w_k - g_{k-1}^T p_k \|w_k\|_2^2} = \frac{-g_{k-1}^T p_k p_k^T w_k}{-g_{k-1}^T p_k \|w_k\|_2^2} = \frac{p_k^T w_k}{\|w_k\|_2^2} = \frac{g_k^T w_k}{\|w_k\|_2^2} = \alpha_k^{\mathrm{MG}}.$$

On the other hand, (6) directly implies,

$$\beta_k = \frac{g_{k-1}^T (g_{k-1} - r_k(\alpha))}{\|g_{k-1} - r_k(\alpha)\|_2^2} = \beta_k^{\mathrm{DWGM}}.$$

That is, for the convex quadratic minimization problem (3), BiDWGM and DWGM are mathematically equivalent.

3 Numerical Experiments

In this section we present some numerical experiments. The objective is to evaluate the numerical behaviour of the BiDWGM algorithm. All the experiments were performed on a intel(R) CORE(TM) i7-4770, CPU 3.40 GHz with 16 GB RAM. In order to verify the equivalence between DWGM and BiDWGM we propose an experiment with real data obtained from the SuiteSparse Matrix Collection [6]. We perform a comparison between DWGM and BiDWGM.

We obtained for our numerical tests seventy positive definite matrices from the SuiteSparse Matrix Collection. Then we solved the resulting seventy linear systems $Ax = b$ with the DWGM and BiDWGM algorithms with $b = [1, 1, \ldots, 1]^T$ and $x_0 = [0, 0, \ldots, 0]^T$. The stopping criterium used is $\|g_k\|_2 \leqslant 10^{-5}$. A comparison between the performance [8] of DWGM and BiDWGM is done on Figure 1. We observe that DWGM and BiDWGM have a similar performance, this is a simple consequence of the equivalence between the algorithms demonstrated above. Same behaviour is observed when we compare CPU times.

4 Conclusions

The delayed weighted gradient method is a two-step gradient method that promotes a convergence acceleration of the gradient method. In this work, we have derived, via a two-dimensional

optimization, a new two-step gradient method called BiDWGM. We also have proved some properties of the new algorithm, and its global convergence. In spite to be developed from different theoretical arguments, the methods BiDWGM and DWGM are equivalent. This claim is supported by theoretical arguments as well as by the numerical experiments. Such equivalence is true for convex quadratic optimization problems, but if applied to more general convex problems, both arguments yield different procedures, which will be the focus of our future work.

References

[1] R. Andreani and M. Raydan. "Properties of the delayed weighted gradient method". In: **Computational Optimization and Applications** 78 (2021), pp. 167–180. DOI: 10.1007/s10589-020-00232-9.

[2] R. De Asmundis et al. "An efficient gradient method using the Yuan steplength". In: **Computational Optimization and Applications** 59 (2014), pp. 541–563. DOI: 10.1007/s10589-014-9669-5.

[3] J. Barzilai and J. M. Borwein. "Two-point step size gradient methods". In: **IMA Journal of Numerical Analysis** 8.1 (1988), pp. 141–148. DOI: 10.1093/imanum/8.1.141.

[4] A. L. Cauchy. "Méthode générale pour la résolution des systemes d'équations simultanées". In: **Comptes Rendus Sci. Paris** 25 (1847), pp. 536–538.

[5] Y. H. Dai, Y. Huang, and X. W. Liu. "A family of spectral gradient methods for optimization". In: **Computational Optimization and Applications** 74 (2019), pp. 43–65. DOI: 10.1007/s10589-019-00107-8.

[6] T. A. Davis and Y. Hu. "The University of Florida Sparse Matrix Collection". In: **ACM Trans. Math. Softw.** 38.1 (2011). DOI: 10.1145/2049662.2049663.

[7] D. di Serafino et al. "On the steplength selection in gradient methods for unconstrained optimization". In: **Applied Mathematics and Computation** 318 (2018), pp. 176–195. DOI: 10.1016/j.amc.2017.07.037.

[8] E. D. Dolan and J. J. Moré. "Benchmarking optimization software with performance profiles". In: **Mathematical Programming** 91 (2002), pp. 201–2013. DOI: 10.1007/s101070100263.

[9] R. Fletcher. "A limited memory steepest descent method". In: **Mathematical Programming** 135 (2012), pp. 413–436. DOI: 10.1007/s10107-011-0479-6.

[10] G. Frassoldati, L. Zanni, and G. Zanghirati. "New adaptive stepsize selections in gradient methods". In: **Journal of Industrial & Management Optimization** 4.2 (2008), pp. 299–312. DOI: 10.3934/jimo.2008.4.299.

[11] A. Friedlander et al. "Gradient method with retards and generalizations". In: **SIAM Journal on Numerical Analysis** 36.1 (1999), pp. 275–289. DOI: 10.1137/S003614299427315X.

[12] Y. Huang et al. "Gradient methods exploiting spectral properties". In: **Optimization Methods and Software** 35.4 (2020), pp. 681–705. DOI: 10.1080/10556788.2020.1727476.

[13] H. F. Oviedo-Leon. "A delayed weighted gradient method for strictly convex quadratic minimization". In: **Computational Optimization and Applications** 74 (2019), pp. 729–746. DOI: 10.1007/s10589-019-00125-6.

[14] T. Serafini, G. Zanghirati, and L. Zanni. "Gradient projection methods for large quadratic programs and applications in training support vector machines". In: **Optimization Methods and Software** 20 (2005), pp. 353–378. DOI: 10.1080/10556780512331318182.

