

Article

# On the Adaptive Penalty Parameter Selection in ADMM

Serena Crisci <sup>1,†</sup> , Valentina De Simone <sup>1,\*,†</sup>  and Marco Viola <sup>2,†</sup> 

<sup>1</sup> Department of Mathematics and Physics, University of Campania “Luigi Vanvitelli”, Viale Abramo Lincoln, 5, 81100 Caserta, Italy; serena.crisci@unicampania.it

<sup>2</sup> School of Mathematics and Statistics, University College Dublin, Belfield, D04 V1W8 Dublin, Ireland; marco.viola@ucd.ie

\* Correspondence: valentina.desimone@unicampania.it

† These authors contributed equally to this work.

**Abstract:** Many data analysis problems can be modeled as a constrained optimization problem characterized by nonsmooth functionals, often because of the presence of  $\ell_1$ -regularization terms. One of the most effective ways to solve such problems is through the Alternate Direction Method of Multipliers (ADMM), which has been proved to have good theoretical convergence properties even if the arising subproblems are solved inexactly. Nevertheless, experience shows that the choice of the parameter  $\tau$  penalizing the constraint violation in the Augmented Lagrangian underlying ADMM affects the method’s performance. To this end, strategies for the adaptive selection of such parameter have been analyzed in the literature and are still of great interest. In this paper, starting from an adaptive spectral strategy recently proposed in the literature, we investigate the use of different strategies based on Barzilai–Borwein-like stepsize rules. We test the effectiveness of the proposed strategies in the solution of real-life consensus logistic regression and portfolio optimization problems.

**Keywords:** ADMM; adaptive parameter selection; Barzilai–Borwein rules

## 1. Introduction

The alternating direction method of multipliers [1] (ADMM) has been recognized as a simple but powerful algorithm to solve optimization problems of the form

$$\begin{aligned} \min_{u \in \mathbb{R}^n, v \in \mathbb{R}^m} & H(u) + G(v) \\ \text{subject to} & Eu + Fv = d \end{aligned} \quad (1)$$

where  $H: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  and  $G: \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$  are closed, proper, and convex functions,  $E \in \mathbb{R}^{p \times n}$ ,  $F \in \mathbb{R}^{p \times m}$ ,  $d \in \mathbb{R}^p$ . ADMM splits the problem into smaller pieces, each of which is then easier to handle, blending the benefits of dual decomposition and augmented Lagrangian methods [1]. Starting from an initialization  $(u^0, v^0, \zeta^0)$  and  $\tau > 0$ , at each iteration, ADMM updates the primal and dual variables as

$$u^{k+1} = \underset{u}{\operatorname{argmin}} H(u) + \frac{\tau}{2} \left\| d - Eu - Fv^k + \frac{\zeta^k}{\tau} \right\|^2 \quad (2)$$

$$v^{k+1} = \underset{v}{\operatorname{argmin}} G(v) + \frac{\tau}{2} \left\| d - Eu^{k+1} - Fv + \frac{\zeta^k}{\tau} \right\|^2 \quad (3)$$

$$\zeta^{k+1} = \zeta^k + \tau (d - Eu^{k+1} - Fv^{k+1}). \quad (4)$$

ADMM is guaranteed to converge under mild assumptions for any fixed value of the penalty parameter  $\tau$ , even if the subproblems are solved inexactly [2]. Despite this, it is well-known that the choice of the parameter  $\tau$  is problem-dependent and can affect the practical performance of the algorithm, yielding a highly inefficient method when it



**Citation:** Crisci, S.; De Simone, V.; Viola, M. On the Adaptive Penalty Parameter Selection in ADMM.

*Algorithms* **2023**, *16*, 264. <https://doi.org/10.3390/a16060264>

Academic Editor: Sorin-Mihai Grad

Received: 4 April 2023

Revised: 17 May 2023

Accepted: 19 May 2023

Published: 25 May 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

is not properly selected. Some work has been done to develop suitable techniques for tuning the values of  $\tau$  at each iteration, with the aim of speeding up the convergence in practical applications [3–6]. In [3], the authors proposed an adaptive strategy based on primal and dual residuals, with the idea that  $\tau^k$  forces both to have a similar magnitude. Such a scheme is not guaranteed to converge; nevertheless, the standard convergence theory (fixed values of  $\tau$ ) still applies if one assumes that  $\tau^k$  becomes fixed after a finite number of iterations. A more general and reliable approach has been introduced in [7], where the authors proposed a strategy for automatic selection of the penalty parameter in ADMM, borrowing ideas from spectral stepsize selection strategies in gradient-based methods for unconstrained optimization [8–11]. In particular, starting from the analysis of the dual unconstrained formulation of problem (1), in [7] an optimal penalty parameter at each iteration is defined as the reciprocal of the geometric mean of Barzilai–Borwein-like spectral stepsizes [8,10], corresponding to a gradient step of the Fenchel conjugates of the functions  $H$  and  $G$ , respectively.

Relying on the procedure suggested in [7], this paper aims at investigating the practical efficiency of ADMM, employing adaptive selections of the penalty parameter based on different spectral stepsize rules. Indeed, spectral analysis of Barzilai–Borwein (BB) rules (and their variants) has shown how different choices can influence the practical acceleration of gradient-based methods for both constrained and unconstrained smooth optimization, due to the intrinsic different abilities of such stepsizes of capturing the spectral properties of the problem. Indeed, for strictly convex quadratic problems, the Barzilai–Borwein updating rules correspond to the inverses of the Rayleigh quotients of the Hessian, thus providing suitable approximations of the inverses of its eigenvalues. Such an ability has been exploited within ad hoc steplength selection strategies to obtain practical accelerations of gradient-based methods; moreover, this property is preserved in the case of general non-quadratic minimization problems [9,11–13]. In this view, we combine the adaptive ADMM scheme with state-of-the-art stepsizes to compute reliable approximations of the penalty parameter  $\tau^k$  at each iteration. The resulting variants of the ADMM scheme thus obtained are compared on two real-life applications in the frameworks of binary classification on distributed architectures and portfolio selection.

The paper is organized as follows. The adaptive ADMM algorithm is described in Section 2. Numerical experiments are reported in Section 3. Finally, some conclusions are drawn in Section 4.

## 2. Adaptive Penalty Parameter Selection in ADMM Method

In this section, we describe the strategy for automatic selection of the penalty parameter in ADMM according to the procedure proposed in [7], in which the authors introduced an adaptive selection of  $\tau$  based on the spectral properties of the Douglas–Rachford (DR) splitting method applied to the dual problem of (1).

Given a closed convex (proper) function  $f$  defined on  $\mathbb{R}^n$ , the Fenchel conjugate of  $f$  is the closed convex (proper) function  $f^*$  defined by

$$f^*(x^*) = \sup_x \{x^\top x^* - f(x)\} = -\inf_x \{f(x) - x^\top x^*\},$$

(see [14]). The dual problem of (1) is given by

$$\max_{\zeta \in \mathbb{R}^p} \zeta^\top d - H^*(E^\top \zeta) - G^*(F^\top \zeta), \quad (5)$$

where  $H^*$  and  $G^*$  denote, respectively, the Fenchel conjugate of  $H$  and  $G$ . Problem (5) can be equivalently rewritten as

$$\min_{\zeta \in \mathbb{R}^p} H^*(E^\top \zeta) - \zeta^\top d + G^*(F^\top \zeta). \quad (6)$$

It can be proved that solving (1) by ADMM is equivalent to solving the dual problem (6) by means of the DR method [15], which, in turn, is equivalent to applying the DR scheme to

$$0 \in (E^\top \partial H^*(E^\top \zeta) - d) + F^\top \partial G^*(F^\top \zeta).$$

In this way, two sequences  $(\bar{\zeta}^k)_k$  and  $(\zeta^k)_k$  are generated such that

$$0 \in \frac{\bar{\zeta}^{k+1} - \zeta^k}{\tau^k} + (E^\top \partial H^*(E^\top \bar{\zeta}^{k+1}) - d) + F^\top \partial G^*(F^\top \zeta^k), \tag{7}$$

$$0 \in \frac{\zeta^{k+1} - \zeta^k}{\tau^k} + (E^\top \partial H^*(E^\top \bar{\zeta}^{k+1}) - d) + F^\top \partial G^*(F^\top \zeta^{k+1}). \tag{8}$$

Then, Proposition 1 in [7] proves that the choice of the parameter  $\tau^k$  that guarantees the minimal residual of  $(E^\top \partial H^*(E^\top \zeta^{k+1}) - d) + F^\top \partial G^*(F^\top \zeta^{k+1})$  in DR steps is given by

$$\tau^k = \sqrt{\alpha_k \beta_k} \tag{9}$$

where  $\alpha_k, \beta_k > 0$  are BB stepsizes arising by imposing the following quasi-Newton conditions:

$$\alpha_k = \operatorname{argmin}_{\alpha \in \mathbb{R}} \left\| \alpha^{-1} (\bar{\zeta}^k - \bar{\zeta}^{k-1}) - (\phi^k - \phi^{k-1}) \right\|, \tag{10}$$

$$\beta_k = \operatorname{argmin}_{\beta \in \mathbb{R}} \left\| \beta^{-1} (\zeta^k - \zeta^{k-1}) - (\psi^k - \psi^{k-1}) \right\|, \tag{11}$$

where  $\phi^k \in E^\top \partial H^*(E^\top \bar{\zeta}^k) - d$ ,  $\phi^{k-1} \in E^\top \partial H^*(E^\top \bar{\zeta}^{k-1}) - d$ ,  $\psi^k \in F^\top \partial G^*(F^\top \zeta^k)$ , and  $\psi^{k-1} \in F^\top \partial G^*(F^\top \zeta^{k-1})$ . Note that  $1/\alpha_k$  and  $1/\beta_k$  can be interpreted as spectral gradient stepsizes of type BB1 for  $H^*(E^\top \bar{\zeta}^k) - (\bar{\zeta}^k)^\top d$  and  $G^*(F^\top \zeta^k)$ , respectively.

Based on the equivalence between DR and ADMM, the optimal DR stepsize  $\tau^k$  defined in (9) corresponds to the optimal penalty parameter for the ADMM scheme. Moreover, to compute practical estimates of these optimal parameters for ADMM, the dual problem is not required to be supplied, thanks to the theoretical link between primal and dual variables. Indeed, the optimality condition for Problem (2) prescribes

$$0 \in \partial H(u^{k+1}) - E^\top \left( \zeta^k + \tau^k (d - Eu^{k+1} - Fv^k) \right),$$

which is equivalent to

$$E^\top \left( \zeta^k + \tau^k (d - Eu^{k+1} - Fv^k) \right) \in \partial H(u^{k+1}).$$

Recalling that for a closed proper convex function  $f$ ,  $x \in \partial f^*(x^*)$  if and only if  $x^* \in \partial f(x)$  (see [14], Corollary 23.5.1), from the previous relation, we obtain

$$u^{k+1} \in \partial H^* \left( E^\top \left( \zeta^k + \tau^k (d - Eu^{k+1} - Fv^k) \right) \right),$$

and, hence, it follows

$$Eu^{k+1} - d \in E \partial H^* \left( E^\top \bar{\zeta}^{k+1} \right) - d, \tag{12}$$

where  $\bar{\zeta}^{k+1} := \zeta^k + \tau^k (d - Eu^{k+1} - Fv^k)$ . Similarly, from the optimality condition for the subproblem (3), one can obtain

$$Fv^{k+1} \in F \partial G^*(F^\top \zeta^{k+1}). \tag{13}$$

From (12) and (13), we have

$$\bar{\zeta}^{k+1} \in \zeta^k - \tau^k \left( (E\partial H^*(E^\top \bar{\zeta}^{k+1}) - d) + F\partial G^*(F^\top \zeta^k) \right), \tag{14}$$

$$\zeta^{k+1} \in \zeta^k - \tau^k \left( (E\partial H^*(E^\top \bar{\zeta}^{k+1}) - d) + F\partial G^*(F^\top \zeta^{k+1}) \right). \tag{15}$$

Finally, one can define  $\Delta\bar{\zeta}^{k-1} = \bar{\zeta}^k - \bar{\zeta}^{k-1}$  and  $\Delta\bar{H}^{k-1} = \partial\bar{H}(\bar{\zeta}^k) - \partial\bar{H}(\bar{\zeta}^{k-1})$ , where  $\bar{H}(\bar{\zeta}) := H^*(E^\top \bar{\zeta}) - \bar{\zeta}^\top d$ , and the set subtraction is given by the Minkowski–Pontryagin difference [16,17]. In particular, a practical computation of an element of  $\Delta\bar{H}^{k-1}$  can be provided through quantities available at the current ADMM iteration by exploiting (12); thus, with a slight abuse of notation, we may write

$$\Delta\bar{H}^{k-1} = E(u^k - u^{k-1}).$$

Then, the two BB-based rules can be recovered as

$$\alpha_k^{BB1} = \operatorname{argmin}_{\alpha \in \mathbb{R}} \left\| \alpha^{-1} \Delta\bar{\zeta}^{k-1} - \Delta\bar{H}^{k-1} \right\| = \frac{\|\Delta\bar{\zeta}^{k-1}\|^2}{(\Delta\bar{\zeta}^{k-1})^\top \Delta\bar{H}^{k-1}}, \tag{16}$$

$$\alpha_k^{BB2} = \operatorname{argmin}_{\alpha \in \mathbb{R}} \left\| \Delta\bar{\zeta}^{k-1} - \alpha \Delta\bar{H}^{k-1} \right\| = \frac{(\Delta\bar{\zeta}^{k-1})^\top \Delta\bar{H}^{k-1}}{\|\Delta\bar{H}^{k-1}\|^2}. \tag{17}$$

With a similar argument, the curvature estimates of  $\bar{G}(\bar{\zeta}) := G^*(F^\top \bar{\zeta})$  are provided by the following stepsizes:

$$\beta_k^{BB1} = \operatorname{argmin}_{\beta \in \mathbb{R}} \left\| \beta^{-1} \Delta\bar{\zeta}^{k-1} - \Delta\bar{G}^{k-1} \right\| = \frac{\|\Delta\bar{\zeta}^{k-1}\|^2}{(\Delta\bar{\zeta}^{k-1})^\top \Delta\bar{G}^{k-1}}, \tag{18}$$

$$\beta_k^{BB2} = \operatorname{argmin}_{\beta \in \mathbb{R}} \left\| \Delta\bar{\zeta}^{k-1} - \beta \Delta\bar{G}^{k-1} \right\| = \frac{(\Delta\bar{\zeta}^{k-1})^\top \Delta\bar{G}^{k-1}}{\|\Delta\bar{G}^{k-1}\|^2}, \tag{19}$$

where  $\Delta\bar{\zeta}^{k-1} = \bar{\zeta}^k - \bar{\zeta}^{k-1}$  and  $\Delta\bar{G}^{k-1} = F(v^k - v^{k-1})$ . The previous quasi-Newton conditions express a local property of linearity of the dual subgradients with respect to the dual variables. The validity of this assumption can be checked during the iterative procedure to test the reliability of the spectral BB-based parameters. In particular, the stepsizes (16)–(19) can be considered reliable when, respectively, the ratios

$$\alpha_{\text{cor}} = \frac{(\Delta\bar{\zeta}^{k-1})^\top \Delta\bar{H}^{k-1}}{\|\Delta\bar{\zeta}^{k-1}\| \|\Delta\bar{H}^{k-1}\|} \quad \text{and} \quad \beta_{\text{cor}} = \frac{(\Delta\bar{\zeta}^{k-1})^\top \Delta\bar{G}^{k-1}}{\|\Delta\bar{\zeta}^{k-1}\| \|\Delta\bar{G}^{k-1}\|} \tag{20}$$

are bounded away from zero.

Then, as safeguarding condition, the update of the penalty parameter is realized in accordance with (9) when both the ratios in (20) are greater than a prefixed threshold  $\bar{\epsilon} \in (0, 1)$  expressing the required level of reliability provided by the estimates (10) and (11). If only one of the ratios satisfies the safeguarding condition, the corresponding stepsize is used to estimate  $\tau^k$ , which eventually is set equal to the last updated value when both stepsizes are considered inaccurate, i.e., when  $\alpha_{\text{cor}} \leq \bar{\epsilon}$  and  $\beta_{\text{cor}} \leq \bar{\epsilon}$ .

A general ADMM scheme with adaptive selection of the penalty parameter based on the described procedure is outlined in Algorithm 1. We remark that different versions of Algorithm 1 arise depending on the different rules selected for computing the spectral stepsizes  $\alpha_k$  and  $\beta_k$  in STEP 4. In particular, in the scheme originally proposed in [7],  $\alpha_k$  and  $\beta_k$  are provided by a generalization of the *adaptive steepest descent* (ASD) strategy introduced in [10], which performs a proper alternation of larger and smaller stepsizes. In the next section, we will compare this procedure with other updating rules based on both single and alternating BB-based strategies.

As final remark, we recall that a proof of the convergence of ADMM with variable penalty parameter was provided in [3] under suitable assumptions on the increase and decrease in the sequence  $\{\tau^k\}_k$  (see Section 4—Theorem 4.1 in [3]). Although this convergence analysis cannot straightforwardly be applied to our case, as observed in [7], this issue may be bypassed in practice by turning off the adaptivity after a finite number of steps.

---

**Algorithm 1** A general scheme for ADMM with adaptive penalty parameter selection

---

Initialize  $u^0, v^0, \zeta^0, \tau_0 > 0, \bar{\epsilon} \in (0, 1), \bar{n} \geq 1$

For  $k = 0, 1, 2, \dots$

STEP 1 Compute  $u^{k+1}$  by solving (2).

STEP 2 Compute  $v^{k+1}$  by solving (3).

STEP 3 Update  $\zeta^{k+1}$  by means of (4).

STEP 4 If  $\text{mod}(k, \bar{n}) = 1$  then

$$\bar{\zeta}^{k+1} := \zeta^k + \tau^k(d - Eu^{k+1} - Fv^k)$$

Compute spectral stepsizes  $\alpha_k, \beta_k$  according to (10) and (11)

Compute correlations  $\alpha_{\text{cor}}, \beta_{\text{cor}}$

if  $\alpha_{\text{cor}} > \epsilon_{\text{cor}}$  and  $\beta_{\text{cor}} > \epsilon_{\text{cor}}$  then

$$\tau^{k+1} = \sqrt{\alpha_k \beta_k}$$

elseif  $\alpha_{\text{cor}} > \bar{\epsilon}$  and  $\beta_{\text{cor}} \leq \bar{\epsilon}$

$$\tau^{k+1} = \alpha_k$$

elseif  $\alpha_{\text{cor}} \leq \bar{\epsilon}$  and  $\beta_{\text{cor}} > \bar{\epsilon}$

$$\tau^{k+1} = \beta_k$$

else

$$\tau^{k+1} = \tau^k$$

endif

else

$$\tau^{k+1} = \tau^k$$

endif

STEP 5 end for

---

### 3. Numerical Experiments

In this section we present the results of the numerical experiments performed to assess the performances of Algorithm 1 equipped with different choices for the update of  $\tau^k$  in STEP 4. We compare the alternating strategy used in [7] with BB1-like rules  $\alpha_k^{BB1}$  (16)–(18), BB2-like rules  $\alpha_k^{BB2}$  (17)–(19), and a different alternating strategy, based on a modified version of the  $ABB_{\text{min}}$  rule [18], introduced in [19]. The five algorithms compared in this section are the following:

- “Vanilla ADMM”, in which  $\tau^k$  is fixed to  $\tau_0$  throughout all iterations;
- “Adaptive ADMM” [7], in which  $\alpha_k$  and  $\beta_k$  are set to

$$\alpha_k = \begin{cases} \alpha_k^{BB2}, & \text{if } 2\alpha_k^{BB2} > \alpha_k^{BB1}, \\ \alpha_k^{BB1} - \frac{\alpha_k^{BB2}}{2}, & \text{otherwise,} \end{cases} \quad \beta_k = \begin{cases} \beta_k^{BB2}, & \text{if } 2\beta_k^{BB2} > \beta_k^{BB1}, \\ \beta_k^{BB1} - \frac{\beta_k^{BB2}}{2}, & \text{otherwise;} \end{cases}$$

- “Adaptive ADMM-BB1”, in which  $\alpha_k$  and  $\beta_k$  are set to  $\alpha_k^{BB1}$  and  $\beta_k^{BB1}$ , respectively;
- “Adaptive ADMM-BB2”, in which  $\alpha_k$  and  $\beta_k$  are set to  $\alpha_k^{BB2}$  and  $\beta_k^{BB2}$ , respectively;

- “Adaptive ADMM-ABB<sub>min</sub>”, in which  $\alpha_k$  and  $\beta_k$  are set to

$$\alpha_k^{ABB_{min}} = \begin{cases} \min\{\alpha_j^{BB2} : j = \max\{1, k - m_\alpha\}, \dots, k\}, & \text{if } \alpha_k^{BB2} < \delta_k \alpha_k^{BB1}, \\ \alpha_k^{BB1}, & \text{otherwise,} \end{cases}$$

$$\beta_k^{ABB_{min}} = \begin{cases} \min\{\beta_j^{BB2} : j = \max\{1, k - m_\alpha\}, \dots, k\}, & \text{if } \beta_k^{BB2} < \delta_k \beta_k^{BB1}, \\ \beta_k^{BB1}, & \text{otherwise,} \end{cases}$$

where  $m_\alpha = 2$ ,  $\delta_0 = 0.5$ , and  $\delta_k$  is updated as follows:

$$\delta_{k+1} = \begin{cases} \delta_k / 1.2, & \text{if } \alpha_k^{BB2} < \delta_k \alpha_k^{BB1} \text{ (or } \beta_k^{BB2} < \delta_k \beta_k^{BB1}), \\ \delta_k \cdot 1.2, & \text{otherwise.} \end{cases}$$

For all the algorithms, following [7], we set  $\epsilon_{cor} = 0.2$  and  $\bar{n} = 2$ . The methods stop when the relative residual is below a prefixed tolerance  $tol > 0$  within a maximum number of iterations, where the relative residual is defined by

$$\max \left\{ \frac{\|r_p^k\|_2}{\max\{\|Eu^k\|_2, \|Fv^k\|_2, \|d\|_2\}}, \frac{\|r_d^k\|_2}{\|E^T \zeta^k\|_2} \right\}, \tag{21}$$

with  $r_p^k = d - Eu^k - Fv^k$  and  $r_d^k = \tau^k E^T F(v^k - v^{k-1})$ . All experiments were performed using MATLAB. Recently, the PADPD algorithm [20] has been proposed as an algorithm analogous to ADMM, but it has fixed stepsize and is out of the scope of this paper to compare.

### 3.1. Consensus $\ell_1$ -Regularized Logistic Regression

As a first experiment, we test the proposed algorithms on the solution of consensus  $\ell_1$ -regularized logistic regression (see Section 4.2 in [1,7]). Let us consider a dataset consisting of  $M$  training pairs  $(D_i, y_i) \in \mathbb{R}^n \times \{0, 1\}$ . The aim is to build a linear classifier by minimizing a regularized logistic regression functional, exploiting a distributed computing architecture. One can do so by partitioning the original dataset into  $S$  subsets of size  $m_1, \dots, m_S$ , such that  $\sum_{s=1}^S m_s = M$ , and solving the optimization problem

$$\begin{aligned} \min_{x_1, \dots, x_S \in \mathbb{R}^n, z \in \mathbb{R}^n} & \sum_{s=1}^S \sum_{j=1}^{m_s} \log \left( 1 + e^{-y_{s,j} D_{s,j}^T x_s} \right) + \lambda \|z\|_1 \\ \text{s.t.} & x_s - z = 0, \quad \text{for } s = 1, \dots, S, \end{aligned} \tag{22}$$

where  $x_s$  is the local variable on the  $s$ -th computational node, which acts as a linear classifier of the  $s$ -th subset,  $(D_{s,j}, y_{s,j})$  represents the  $j$ -th training pair of the  $s$ -th subset, and  $z$  is the global variable.

We can reformulate problem (22) as

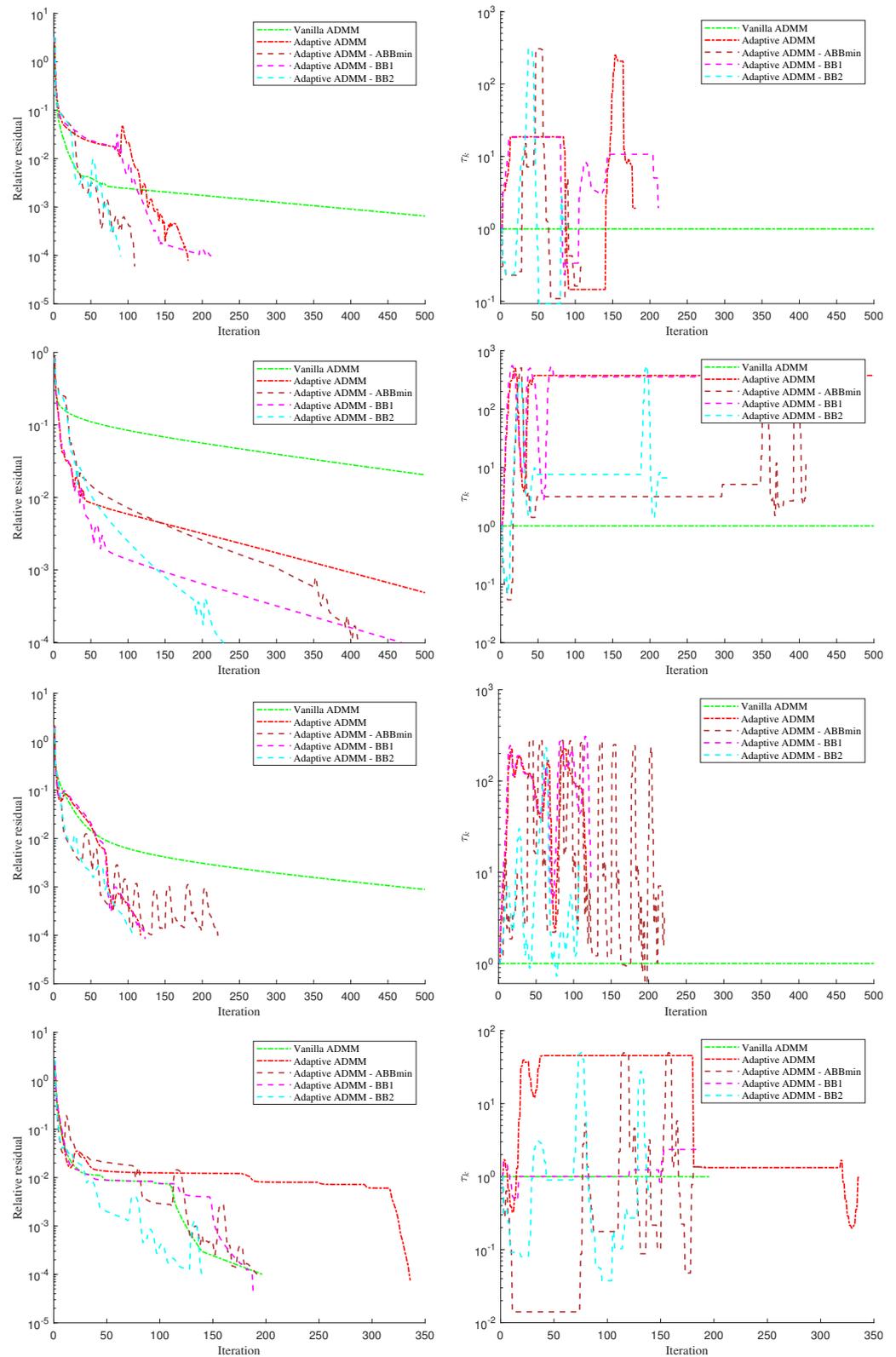
$$\begin{aligned} \min_{u,v} & H(u) + G(v) \\ \text{s.t.} & u - Fv = 0 \end{aligned} \tag{23}$$

where we set  $u = (x_1^\top, \dots, x_S^\top)^\top \in \mathbb{R}^{nS}$ ,  $v = z$ ,  $H(u) = \sum_{s=1}^S \sum_{j=1}^{m_s} \log \left( 1 + e^{-y_{s,j} D_{s,j}^T x_s} \right)$ ,  $G(v) = \lambda \|v\|_1$ , and  $F = (I_n, \dots, I_n)^\top$  (i.e., the stacking of  $S$  identity matrices of order  $n$ ). Scaling the dual variable, the augmented Lagrangian function associated with problem (23) is

$$\mathcal{L}_A = H(u) + G(v) + \frac{\tau}{2} \left\| u - Fv - \frac{\zeta}{\tau} \right\|^2 - \frac{\tau}{2} \left\| \frac{\zeta}{\tau} \right\|^2,$$

where  $\zeta = (\zeta_1^\top, \dots, \zeta_S^\top)^\top \in \mathbb{R}^{nS}$ ,  $\tau > 0$ . Starting from given estimates  $u^0, v^0, \zeta^0$ , and  $\tau^0$ , and observing that the minimization problem in  $u$  can be split into  $S$  independent





**Figure 1.** Relative residual (left) and  $\tau^k$  history (right) against number of iterations for the 4 consensus logistic regression instances. From top to bottom: a9a, cod-rna, ijcn1, phishing.

### 3.2. Portfolio Optimization

In the modern portfolio theory, an optimal portfolio selection strategy has to realize a trade-off between risk and return. Recently, l1-regularized Markowitz models have been

considered; the l1 penalty term is used to stabilize the solution process and to obtain sparse solutions, which allow one to reduce holding costs [21–24]. We focus on a multi-period investment strategy [23] that is either medium- or long-term; thus, it allows the periodic reallocation of wealth among the assets based on available information. The investment period is partitioned into  $m$  sub-periods, delimited by the rebalancing dates  $t_1, \dots, t_{m+1}$ , at which points decisions are taken. Let  $n$  be the number of assets and  $u_j \in \mathbb{R}^n$  the portfolio held at the rebalancing date  $t_j$ . The optimal portfolio is defined by the vector  $u = (u_1^\top, u_2^\top, \dots, u_m^\top)^\top \in \mathbb{R}^N$ , where  $N = m \cdot n$ . A separable form of the risk measure obtained by summing single period terms is considered

$$\rho(u) = \sum_{j=1}^m u_j^\top C_j u_j,$$

where  $C_j \in \mathbb{R}^{n \times n}$  is the covariance matrix, assumed to be positive definite, estimated at  $t^j$ . With this choice, the model satisfies the time consistency property.  $l_1$ -regularization has been used to promote sparsity in the solution. Moreover, the  $l_1$  penalty term either avoids or limits negative solutions; thus, it is equivalent to a penalty on short positions.

Let  $\xi_{\text{init}}$  and  $\xi_{\text{term}}$  be the initial wealth and the target expected wealth resulting from the overall investment, respectively, and let  $r_j$  be the expected return vector estimated at time  $j$ . The l1-regularized selection can be formulated as the following compact constrained optimization problem [23,24]:

$$\begin{aligned} \min_u \quad & \frac{1}{2} u^\top C u + \lambda \|u\|_1 \\ \text{s.t.} \quad & A u = b \end{aligned} \tag{27}$$

where  $C = \text{diag}(C_1, C_2, \dots, C_m) \in \mathbb{R}^{N \times N}$  is a  $m \times m$  diagonal block matrix,  $A$  is an  $m \times m$  lower bidiagonal block matrix, with blocks of dimension  $1 \times n$ , defined as

$$\begin{aligned} \text{diag}(A) &= (1_n^\top, 1_n^\top, \dots, 1_n^\top), \\ \text{subdiag}(A) &= (-(1_n + r_1)^\top, \dots, -(1_n + r_{m-1})^\top), \end{aligned}$$

and  $b = (\xi_{\text{init}}, 0, 0, \dots, \xi_{\text{term}})^\top \in \mathbb{R}^m$ . Methods based on Bregman iteration have proved to be efficient for the solution of Problem (27) as well [23–26]. Now we reformulate Problem (27) as

$$\begin{aligned} \min_{u,v} \quad & H(u) + G(v) \\ \text{s.t.} \quad & u - v = 0 \end{aligned} \tag{28}$$

where  $H(u) = \frac{1}{2} u^\top C u$  restricted to the set  $\{u \mid Au = b\}$  and  $G(v) = \lambda \|v\|_1$ . Scaling the dual variable, the augmented Lagrangian function associated with Problem (28) is

$$\mathcal{L}_A = H(u) + G(v) + \frac{\tau}{2} \left\| u - v - \frac{\xi}{\tau} \right\|^2 - \frac{\tau}{2} \left\| \frac{\xi}{\tau} \right\|^2,$$

where  $\xi, \tau > 0$ . Starting from given estimates  $u^0, v^0, \xi^0$ , and  $\tau^0$ , at each iteration  $k$ , adaptive ADMM updates the estimates as

$$u^{k+1} = \underset{Au=b}{\operatorname{argmin}} \frac{1}{2} u^\top C u + \frac{\tau^k}{2} \left\| u - v^k - \frac{\xi^k}{\tau^k} \right\|^2, \tag{29}$$

$$v^{k+1} = \underset{v}{\operatorname{argmin}} \lambda \|v\|_1 + \frac{\tau^k}{2} \left\| u^{k+1} - v - \frac{\xi^k}{\tau^k} \right\|^2, \tag{30}$$

$$\xi^{k+1} = \xi^k + \tau^k (u^{k+1} - v^{k+1}). \tag{31}$$

Given  $(u^k, v^k, \zeta^k)$ , the  $u$  update (29) is an equality-constrained problem that involves the solution of the related Karush–Kuhn–Tucker (KKT) system (see [1], Section 4.2.5), which results in a linear system with positive definite coefficient matrix. The minimization with respect to  $v$  can be carried out efficiently using the soft thresholding operator.

We test the effectiveness on the real market data. We show results obtained using the following datasets [26]:

- FF48 (Fama & French 48 Industry portfolios): Contain monthly returns of 48 from July 1926 to December 2015. We simulate investment strategies of length 10, 20, and 30 years, with annual rebalancing.
- NASDAQ100 (NASDAQ 100 stock Price Data): Contains monthly returns from November 2004 to April 2016. We simulate investment strategy of length 10, with annual rebalancing.
- Dow Jones (Dow Jones Industrial): Contains monthly returns from February 1990 to April 2016. We simulate investment strategy of length 10, with annual rebalancing.

We set for all the tests  $\lambda = 10^{-2}$  to enforce a sparse portfolio. We compared the 5 considered algorithms in terms of number of iterations needed to reach a tolerance  $10^{-5}$  on the relative residual, letting them run for a maximum of 3000 iterations. Moreover, we considered some financial performance measures expressing the goodness of the optimal portfolios, i.e.,

- density: the number of nonzero elements in the solution (percentage) that gives an estimation of holding cost;
- ratio: it estimates the risk reduction when the naive strategy (at each rebalancing date the total wealth is equally divided among the assets) is taken as the benchmark, defined as

$$ratio = \frac{u_{naive}^\top C u_{naive}}{u_*^\top C u_*},$$

where the numerator is the variance of the portfolio produced by the naive strategy and the denominator is the variance of the optimal portfolio.

Tables 2–6 report the results of the experiments performed on the 5 datasets with different choices for the value of  $\tau_0$  (namely, 0.1, 0.5, 1) for a total of 15 instances.

The five algorithms are overall able to obtain equivalent portfolios when converging to the desired tolerance. However, they behave quite differently in terms of iterations needed to converge. In general, the adaptive strategies allow a reduction in the computational complexity with respect to the Vanilla ADMM, which is unable to reach the desired tolerance in 9 out of 15 instances. Among the 4 adaptive strategies, “Adaptive ADMM-BB1” seems to be the most effective, being able to outperform all the others in 9 out of 15 instances, and being the second best in 4 out of 15, performing on average 20% of iterations more than the best in this case. Unlike what happened in the case of consensus logistic regression problems, here “Adaptive ADMM-BB2” and “Adaptive ADMM-ABB<sub>min</sub>” appear to perform poorly, suggesting that the use of too small of values for  $\tau^k$  may slow down the convergence.

**Table 2.** Performance for portfolio FF48 with 10-year simulation.

Method	Iterations	Density	Ratio
		$\tau^0 = 0.1$	
Vanilla ADMM	846	0.096	2.064
Adaptive ADMM [7]	321	0.096	2.064
Adaptive ADMM-BB1	190	0.096	2.064
Adaptive ADMM-BB2	458	0.096	2.064
Adaptive ADMM-ABB <sub>min</sub>	509	0.096	2.064
		$\tau^0 = 0.5$	
Vanilla ADMM	3000	—	—
Adaptive ADMM [7]	488	0.096	2.064
Adaptive ADMM-BB1	340	0.096	2.064

**Table 2.** *Cont.*

Method	Iterations	Density	Ratio
Adaptive ADMM-BB2	644	0.096	2.064
Adaptive ADMM-ABB <sub>min</sub>	665	0.096	2.064
$\tau^0 = 1$			
Vanilla ADMM	3000	—	—
Adaptive ADMM [7]	283	0.096	2.064
Adaptive ADMM-BB1	232	0.096	2.064
Adaptive ADMM-BB2	617	0.096	2.064
Adaptive ADMM-ABB <sub>min</sub>	618	0.096	2.064

**Table 3.** Performance for portfolio FF48 with 20-year simulation.

Method	Iterations	Density	Ratio
$\tau^0 = 0.1$			
Vanilla ADMM	871	0.111	2.430
Adaptive ADMM [7]	518	0.111	2.430
Adaptive ADMM-BB1	470	0.111	2.430
Adaptive ADMM-BB2	1698	0.111	2.430
Adaptive ADMM-ABB <sub>min</sub>	298	0.111	2.430
$\tau^0 = 0.5$			
Vanilla ADMM	3000	—	—
Adaptive ADMM [7]	454	0.112	2.430
Adaptive ADMM-BB1	450	0.111	2.430
Adaptive ADMM-BB2	1291	0.111	2.430
Adaptive ADMM-ABB <sub>min</sub>	1049	0.111	2.430
$\tau^0 = 1$			
Vanilla ADMM	3000	—	—
Adaptive ADMM [7]	454	0.111	2.430
Adaptive ADMM-BB1	361	0.111	2.430
Adaptive ADMM-BB2	1252	0.111	2.430
Adaptive ADMM-ABB <sub>min</sub>	775	0.111	2.430

**Table 4.** Performance for portfolio FF48 with 30-year simulation.

Method	Iterations	Density	Ratio
$\tau^0 = 0.1$			
Vanilla ADMM	1700	0.147	5.134
Adaptive ADMM [7]	263	0.147	5.134
Adaptive ADMM-BB1	331	0.147	5.134
Adaptive ADMM-BB2	1806	0.147	5.134
Adaptive ADMM-ABB <sub>min</sub>	281	0.147	5.134
$\tau^0 = 0.5$			
Vanilla ADMM	3000	—	—
Adaptive ADMM [7]	270	0.147	5.134
Adaptive ADMM-BB1	327	0.147	5.134
Adaptive ADMM-BB2	324	0.147	5.134
Adaptive ADMM-ABB <sub>min</sub>	275	0.146	5.134
$\tau^0 = 1$			
Vanilla ADMM	3000	—	—
Adaptive ADMM [7]	252	0.147	5.134
Adaptive ADMM-BB1	269	0.147	5.134
Adaptive ADMM-BB2	458	0.147	5.134
Adaptive ADMM-ABB <sub>min</sub>	595	0.146	5.134

**Table 5.** Performance for portfolio NASDAQ100 with 10-year simulation.

Method	Iterations	Density	Ratio
$\tau^0 = 0.1$			
Vanilla ADMM	1421	0.023	2.005
Adaptive ADMM [7]	1140	0.023	2.005
Adaptive ADMM-BB1	906	0.023	2.005
Adaptive ADMM-BB2	3000	—	—
Adaptive ADMM-ABB <sub>min</sub>	3000	—	—
$\tau^0 = 0.5$			
Vanilla ADMM	1984	0.023	2.005
Adaptive ADMM [7]	1368	0.023	2.005
Adaptive ADMM-BB1	947	0.023	2.005
Adaptive ADMM-BB2	3000	—	—
Adaptive ADMM-ABB <sub>min</sub>	3000	—	—
$\tau^0 = 1$			
Vanilla ADMM	3000	—	—
Adaptive ADMM [7]	1789	0.023	2.005
Adaptive ADMM-BB1	1307	0.023	2.005
Adaptive ADMM-BB2	3000	—	—
Adaptive ADMM-ABB <sub>min</sub>	3000	—	—

**Table 6.** Performance for portfolio Dow Jones with 10-year simulation.  $\epsilon_{\text{cor}} = 0.2$ .

Method	Iterations	Density	Ratio
$\tau^0 = 0.1$			
Vanilla ADMM	1211	0.065	1.370
Adaptive ADMM [7]	505	0.065	1.370
Adaptive ADMM-BB1	510	0.065	1.370
Adaptive ADMM-BB2	645	0.065	1.370
Adaptive ADMM-ABB <sub>min</sub>	1573	0.065	1.370
$\tau^0 = 0.5$			
Vanilla ADMM	3000	—	—
Adaptive ADMM [7]	997	0.065	1.370
Adaptive ADMM-BB1	691	0.065	1.370
Adaptive ADMM-BB2	1077	0.065	1.370
Adaptive ADMM-ABB <sub>min</sub>	1161	0.065	1.370
$\tau^0 = 1$			
Vanilla ADMM	3000	—	—
Adaptive ADMM [7]	511	0.065	1.370
Adaptive ADMM-BB1	678	0.065	1.370
Adaptive ADMM-BB2	910	0.065	1.370
Adaptive ADMM-ABB <sub>min</sub>	1061	0.065	1.370

#### 4. Conclusions

In this paper, we analyzed different strategies for the adaptive selection of the penalty parameter in ADMM. Exploiting the equivalence between ADMM and the Douglas–Rachford splitting method applied to the dual problem, as suggested in [7], optimal penalty parameters can be estimated at each iteration from spectral stepsizes of a gradient step applied to the Fenchel conjugates of the objective functions. To this end, we selected different spectral steplength strategies based on the Barzilai–Borwein rules, which have been proved to be very efficient in the context of smooth unconstrained optimization.

We compared the different adaptive strategies on the solution of problems coming from distributed machine learning and multiperiod portfolio optimization. The results show that, while adaptive versions of ADMM are usually more effective than the “vanilla” one (using a prefixed penalty parameter), different strategies might perform better on different problems. Moreover, in some cases, the proposed alternation rule might get stuck

with a fixed penalty parameter, leading to a slower convergence. Future work will deal with the analysis of an improved version of the adaptation strategies, aimed at solving the aforementioned issue and their analysis on wider problem classes.

**Author Contributions:** All authors have contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was partially supported by Istituto Nazionale di Alta Matematica-Gruppo Nazionale per il Calcolo Scientifico (INdAM-GNCS), by the Italian Ministry of University.

**Data Availability Statement:** Not applicable.

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

## References

1. Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; Eckstein, J. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn.* **2011**, *3*, 1–122. [[CrossRef](#)]
2. Eckstein, J.; Bertsekas, D.P. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.* **1992**, *55*, 293–318. [[CrossRef](#)]
3. He, B.S.; Yang, H.; Wang, S.L. Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. *J. Optim. Theory Appl.* **2000**, *106*, 337–356. [[CrossRef](#)]
4. Ghadimi, E.; Teixeira, A.; Shames, I.; Johansson, M. On the Optimal Step-size Selection for the Alternating Direction Method of Multipliers. *IFAC Proc. Vol.* **2012**, *45*, 139–144. [[CrossRef](#)]
5. Goldstein, T.; Li, M.; Yuan, X. Adaptive primal-dual splitting methods for statistical learning and image processing. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1–9.
6. Song, C.; Yoon, S.; Pavlovic, V. Fast ADMM algorithm for distributed optimization with adaptive penalty. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.
7. Xu, Z.; Figueiredo, M.; Goldstein, T. Adaptive ADMM with spectral penalty parameter selection. In Proceedings of the Artificial Intelligence and Statistics, PMLR, Lauderdale, FL, USA, 20–22 April 2017; pp. 718–727.
8. Barzilai, J.; Borwein, J.M. Two-point step size gradient methods. *IMA J. Numer. Anal.* **1988**, *8*, 141–148. [[CrossRef](#)]
9. Fletcher, R. On the Barzilai-Borwein method. In *Optimization and Control with Applications*; Qi, L., Teo, K., Yang, X., Pardalos, P.M., Hearn, D., Eds.; Applied Optimization; Springer: New York, NY, USA, 2005; Volume 96, pp. 235–256.
10. Zhou, B.; Gao, L.; Dai, Y.H. Gradient Methods with Adaptive Step-Sizes. *Comput. Optim. Appl.* **2006**, *35*, 69–86. [[CrossRef](#)]
11. Raydan, M. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM J. Optim.* **1997**, *7*, 26–33. [[CrossRef](#)]
12. Crisci, S.; Ruggiero, V.; Zanni, L. Steplength selection in gradient projection methods for box-constrained quadratic programs. *Appl. Math. Comput.* **2019**, *356*, 312–327. [[CrossRef](#)]
13. Crisci, S.; Kružík, J.; Pecha, M.; Horák, D. Comparison of active-set and gradient projection-based algorithms for box-constrained quadratic programming. *Soft Comput.* **2020**, *24*, 17761–17770. [[CrossRef](#)]
14. Rockafellar, R.T. *Convex Analysis*; Princeton University Press: Princeton, NJ, USA, 1997; Volume 11.
15. Esser, E. Applications of Lagrangian-based alternating direction methods and connections to split Bregman. *CAM Rep.* **2009**, *9*, 31.
16. Pontryagin, L.S. Linear differential games. *SIAM J. Control Optim.* **1974**, *12*, 262–267. [[CrossRef](#)]
17. Nurminski, E.; Uryasev, S. Yet Another Convex Sets Subtraction with Application in Nondifferentiable Optimization. *arXiv* **2018**, arXiv:1801.06946.
18. Frassoldati, G.; Zanni, L.; Zanghirati, G. New adaptive stepsize selections in gradient methods. *J. Ind. Manag. Optim.* **2008**, *4*, 299–312. [[CrossRef](#)]
19. Bonettini, S.; Zanella, R.; Zanni, L. A scaled gradient projection method for constrained image deblurring. *Inverse Probl.* **2009**, *25*. [[CrossRef](#)]
20. Shaho Alaviani, S.; Kelkar, A.G. Parallel Alternating Direction Primal-Dual (PADPD) Algorithm for Multi-Block Centralized Optimization. *J. Comput. Inf. Sci. Eng.* **2023**, *23*, 051010. [[CrossRef](#)]
21. Brodie, J.; Daubechies, I.; DeMol, C.; Giannone, D.; Loris, I. Sparse and stable Markowitz portfolios. *Proc. Natl. Acad. Sci. USA* **2009**, *30*, 12267–12272. [[CrossRef](#)]
22. Corsaro, S.; De Simone, V. Adaptive  $l_1$ -regularization for short-selling control in portfolio selection. *Comput. Optim. Appl.* **2019**, *72*, 457–478. [[CrossRef](#)]
23. Corsaro, S.; De Simone, V.; Marino, Z.; Perla, F.  $l_1$ -regularization for multi-period portfolio selection. *Ann. Oper. Res.* **2020**, *294*, 75–86. [[CrossRef](#)]
24. Corsaro, S.; De Simone, V.; Marino, Z.; Scognamiglio, S.  $l_1$ -Regularization in Portfolio Selection with Machine Learning. *Mathematics* **2022**, *10*, 540. [[CrossRef](#)]

25. Corsaro, S.; De Simone, V.; Marino, Z. Fused Lasso approach in portfolio selection. *Ann. Oper. Res.* **2021**, *299*, 47–59. [[CrossRef](#)]
26. Corsaro, S.; De Simone, V.; Marino, Z. Split Bregman iteration for multi-period mean variance portfolio optimization. *Appl. Math. Comput.* **2021**, *392*, 125715. [[CrossRef](#)] [[PubMed](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.