# ONLINE ROBUST PORTFOLIO RISK MANAGEMENT USING TOTAL LEAST-SQUARES AND PARALLEL SPLITTING ALGORITHMS

*Konstantinos Slavakis*[1], *Geert Leus*[2], *and Georgios B. Giannakis*[1]

[1]University of Minnesota
Digital Technology Center, Minneapolis, USA
Emails: slavakis@dtc.umn.edu, georgios@umn.edu

[2]Delft University of Technology
Dept. of Microelectronics, Delft, The Netherlands
Email: g.j.t.leus@tudelft.nl

## ABSTRACT

The present paper introduces a novel online asset allocation strategy which accounts for the sensitivity of Markowitz-inspired portfolios to low-quality estimates of the mean and the correlation matrix of stock returns. The proposed methodology builds upon the total least-squares (TLS) criterion regularized with sparsity attributes, and the ability to incorporate additional convex constraints on the portfolio vector. To solve such an optimization task, the present paper draws from the rich family of splitting algorithms to construct a novel online splitting algorithm with computational complexity that scales linearly with the number of unknowns. Real-world financial data are utilized to demonstrate the potential of the proposed technique.

***Index Terms—*** Markowitz portfolio, total least-squares, sparsity, splitting algorithms, proximal mapping, projection.

## 1. INTRODUCTION

Risk-aware asset allocation has been placed at the epicenter of financial engineering since the landmark paper of H. Markowitz [1], which basically advocates that an investor cannot achieve stock returns exceeding the risk-free scenario without carrying some risk.

If $\boldsymbol{r}$ denotes the $L \times 1$ random vector of *stock returns,* and $\boldsymbol{w}$ the $L \times 1$ *portfolio* or *asset allocation* vector, then risk in [1] is defined as the variance of the portfolio return $\boldsymbol{r}^\top \boldsymbol{w}$, where $\top$ stands for transposition. Hence, if $\boldsymbol{\rho} := \mathsf{E}[\boldsymbol{r}]$ is the expected return, the *mean-variance* framework [1] is defined as

$$\begin{aligned} &\min_{\boldsymbol{w}\in\mathbb{R}^L} \mathsf{E}\big[\big(\boldsymbol{r}^\top \boldsymbol{w} - \eta\big)^2\big] \\ &\text{s.t.} \quad \boldsymbol{w}^\top \boldsymbol{\rho} = \eta, \quad \boldsymbol{w}^\top \mathbf{1}_L = 1, \end{aligned} \tag{1}$$

where $\eta$ is the expected portfolio return, and $\mathbf{1}_L$ the $L \times 1$ vector of ones, introduced in order to fix the total capital equal to a constant (here 1). Task (1) is equivalent to minimizing over $\boldsymbol{w}$ the $\boldsymbol{w}^\top \boldsymbol{\Sigma} \boldsymbol{w}$ under the constraints of (1), where $\boldsymbol{\Sigma} := \mathsf{E}[\boldsymbol{r}\boldsymbol{r}^\top]$ stands for the correlation matrix of the stock returns.

Extensive experimentation has showed that many Markowitz-inspired portfolio construction algorithms cannot markedly outperform the *naive* asset allocation strategy of $\boldsymbol{w}_{\text{naive}} := \mathbf{1}_L/L$ [2,3]. The reason behind this is attributed to deteriorating effects caused by inaccuracies in sample estimates of $\boldsymbol{\rho}$ and $\boldsymbol{\Sigma}$ [2]. Moreover, the mean-variance strategy [1] appears to be oversensitive: small changes in returns or correlations can effect dramatic changes in the output of the optimization procedure [3]. Robust optimization methodologies based on $\min$-$\max$ and second-order cone arguments have also been considered to cope with the said limitations [4]. On the other hand, [3] has demonstrated that regularizing the mean-variance task using the (weighted) $\ell_1$-norm of $\boldsymbol{w}$ can yield stable asset allocation solutions that outperform the naive diversification strategy. Once the data are gathered, [3] relies on the linearly constrained *least angle regression (LARS)* scheme [5] to compute the "optimal" portfolio vector, under the expected return and fixed capital constraints.

This study follows the successful (weighted) $\ell_1$-norm regularization approach of [3], but in a novel *online learning* context. This is the case where an abundance of data flows into a "learning machine", so that there is a pressing need for sequential optimization which incorporates every newly arrived datum in a computationally inexpensive and time-efficient way. This is in contrast to the batch approach of [3]; there, LARS starts from scratch every time a new datum enters the system. Such a batch setting does not fit the needs of modern asset allocation. Indeed, most of the established exchanges, including NYSE, Nasdaq, and the Tokyo Stock Exchange, have nowadays fully or partially adopted electronic order-driven platforms, which process large amounts of bid and ask quotes in very short time intervals: the execution of market orders has dropped to less than 1ms in 2010, and the frequency of quotes can reach $10^5$ times a day [6].

In this online-learning context, the present paper introduces a novel approach to address the limitations of Markowitz-based methods, caused by errors in the estimates of $(\boldsymbol{\rho}, \boldsymbol{\Sigma})$. Regarding estimation of $\boldsymbol{\Sigma}$, and motivated by [7, 8], a *total least-squares (TLS)* approach is followed to model explicitly inaccuracies in the measurements of the stock returns, and allow for an optimization technique that accounts for perturbations in $\boldsymbol{\Sigma}$. As far as $\boldsymbol{\rho}$ is concerned, the linear constraint on the expected return is relaxed towards a closed convex set, which is reminiscent of the celebrated *support vector regression (SVR)* formalism [9]; namely, a closed hyperslab (see Section 3.1). Besides the standard constraints on the mean-variance setting, the novel framework can accommodate any other closed convex constraint on the portfolio vector $\boldsymbol{w}$.

To solve the resultant $\ell_1$-norm regularized, multi-constrained, online minimization task, a TLS approach is developed along with an alternating optimization technique, which draws from the rich family of *splitting algorithms* [10–13]. This yields an online risk-aware algorithm with a computational complexity that scales linearly in the number of unknowns, i.e., it is of order $\mathcal{O}(L)$. The potential of the proposed methodology is demonstrated through numerical tests, which suggest that the novel online learning technique outperforms the naive asset allocation benchmark on a set of real-world financial data [14].

### 1.1. Related work

The classical mean-variance framework (1) has been equipped with sparsity-promoting (weighted) $\ell_1$-norm regularization in [3]. Every time a new datum enters the system, LARS is employed to compute the asset allocation. The sparsity-aware TLS approach was introduced in [8]. There, in order to solve a sub-task of the alternating minimization strategy, the celebrated LASSO [15] was utilized. Motivated by [8], another batch technique can be found in [16], where a matching pursuit [17] methodology is adopted instead of the LASSO one. The principal effort on online TLS has been devoted, mainly, to its fast and efficient implementations [18–20]. To combat the limitations of the Markowitz-inspired portfolios, regarding the sensitivity on the estimation of $\boldsymbol{\Sigma}$, random matrix theory has been utilized in [21] in order to produce reliable estimates for $\boldsymbol{\Sigma}$.

## 2. PROBLEM STATEMENT

Let $\mathbb{N}$, $\mathbb{N}_*$, and $\mathbb{R}$ denote the set of all non-negative integers, positive integers, and real numbers, respectively. Given integers $j_1, j_2$, with $j_1 \leq j_2$, define also $\overline{j_1, j_2} := \{j_1, j_1+1, \ldots, j_2\}$.

Considering first a fixed time instant $n \in \mathbb{N}$, the mean-variance framework (1) assumes the model $\boldsymbol{r}_n^\top \boldsymbol{w} = \eta + v_n$, where the zero-mean noise $v_n$ captures unmodeled errors. Taking this model a step forward, for a given $\mathbb{N}_* \ni q < L$, the matrix $\boldsymbol{R}_n := [\boldsymbol{r}_n, \boldsymbol{r}_{n-1}, \ldots, \boldsymbol{r}_{n-q+1}] \in \mathbb{R}^{L\times q}$ collects the $q$ most recent returns. This formulation of the regression matrix follows the *affine projection algorithm (APA)* [22, 23] rationale. With $\boldsymbol{v}_n$ denoting the corresponding noise vector, the previous model reduces to $\boldsymbol{R}_n^\top \boldsymbol{w} = \eta \boldsymbol{1}_q + \boldsymbol{v}_n$, where $\boldsymbol{1}_q$ is the $q \times 1$ vector of all ones.

Motivated by the discussion in Section 1 on the deteriorating effects of erroneous $\boldsymbol{\Sigma}$ estimates, consider broadening the previous model to account for errors in $\boldsymbol{\Sigma}$ as in $(\boldsymbol{R}_n + \boldsymbol{E})^\top \boldsymbol{w} = \eta \boldsymbol{1}_q + \boldsymbol{v}_n$, for $\boldsymbol{E} \in \mathbb{R}^{L\times q}$. The asset allocation task, per time instant $n$, now becomes

$$\min_{\substack{(\boldsymbol{w},\boldsymbol{E})\in \\ \mathbb{R}^L\times\mathbb{R}^{L\times q}}} \frac{1}{2}\|(\boldsymbol{R}_n+\boldsymbol{E})^\top \boldsymbol{w} - \eta_n \boldsymbol{1}_q\|^2 + \sum_{i=1}^{L} s_i |w_i| + \frac{\lambda_E}{2}\|\boldsymbol{E}\|_{\mathrm{F}}{}^2$$
$$\text{s.t.} \begin{cases} \boldsymbol{w}^\top \boldsymbol{1}_L = 1, & |\boldsymbol{w}^\top \hat{\boldsymbol{\rho}}_n - \eta_n| \leq \epsilon, \\ \boldsymbol{w}^\top \boldsymbol{h} \geq c, & \text{etc}, \end{cases} \tag{2}$$

where it is worth stressing that $\eta_n$ is time-varying. The $\lambda_E$ is a user-defined positive parameter, and $\|\cdot\|_{\mathrm{F}}$ stands for the Frobenius norm of a matrix. The weighted $\ell_1$-norm regularization term is motivated by [3], and the positive coefficient $s_i$ is used to weigh the transaction cost, whenever an investor places capital $w_i$ for the $i$-th asset, $\forall i \in \overline{1, L}$. The $\epsilon > 0$ enabled constraint is introduced here in order to accommodate the possibly erroneous sample estimates $\hat{\boldsymbol{\rho}}_n$ of $\boldsymbol{\rho}$. Here, the sample estimate $\hat{\boldsymbol{\rho}}_n$ is defined as

$$\hat{\boldsymbol{\rho}}_n := \frac{1}{M}\sum_{m=n-M+1}^{n} \boldsymbol{r}_m, \tag{3}$$

where $M \in \mathbb{N}_*$. For $c > 0$, the inequality $\boldsymbol{w}^\top \boldsymbol{h} \geq c$ is interpreted as a tax-related constraint; for example, if one defines $\boldsymbol{h} := [1, 1, 0, \ldots, 0]^\top$, where the 1s identify the position of charity-related stocks, the investment of a total capital $\boldsymbol{w}^\top \boldsymbol{h} \geq c$ could trigger tax deductions.

## 3. SOLVING FOR ONLINE TLS PORTFOLIOS

### 3.1. Preliminaries

First, a few concepts are needed for the rest of the discussion.

**Definition 1** (Hyperplane, closed hyperslab, and projection)**.** Given a nonempty, closed, convex subset $C$ of $\mathbb{R}^L$, the *(metric) projection mapping onto* $C$ is defined as the operator $P_C$ which associates to a $\boldsymbol{w} \in \mathbb{R}^L$ the (unique) point of $C$ that solves the following minimization task: $P_C(\boldsymbol{w}) := \arg\min_{\boldsymbol{u}\in C}\|\boldsymbol{w}-\boldsymbol{u}\|$. Define also the *(metric) distance function to* $C$ as the function $d(\boldsymbol{w}, C) := \|\boldsymbol{w} - P_C(\boldsymbol{w})\|$, $\forall \boldsymbol{w} \in \mathbb{R}^L$. Clearly, $C = \arg\min_{\boldsymbol{u}\in\mathbb{R}^L} d(\boldsymbol{u}, C)$.

A few examples of closed convex sets are in order. Given a nonzero *normal* vector $\boldsymbol{x} \in \mathbb{R}^L$, and an $\alpha \in \mathbb{R}$, a *hyperplane* is defined as $\Pi := \{\boldsymbol{w} \in \mathbb{R}^L : \boldsymbol{x}^\top \boldsymbol{w} = \alpha\}$. For all $\boldsymbol{w} \in \mathbb{R}$,

$$P_\Pi(\boldsymbol{w}) = \boldsymbol{w} + \frac{\alpha - \boldsymbol{x}^\top \boldsymbol{w}}{\|\boldsymbol{x}\|^2}\boldsymbol{x}. \tag{4}$$

Given also an $\epsilon > 0$, a *closed hyperslab* is defined as $S := \{\boldsymbol{w} \in \mathbb{R}^L : |\boldsymbol{x}^\top \boldsymbol{w} - \alpha| \leq \epsilon\}$. For all $\boldsymbol{w} \in \mathbb{R}$,

$$P_S(\boldsymbol{w}) = \begin{cases} \boldsymbol{w} + \frac{\alpha+\epsilon-\boldsymbol{x}^\top\boldsymbol{w}}{\|\boldsymbol{x}\|^2}\boldsymbol{x}, & \text{if } \boldsymbol{w}^\top\boldsymbol{x} - \alpha > \epsilon, \\ \boldsymbol{w} + \frac{\alpha-\epsilon-\boldsymbol{x}^\top\boldsymbol{w}}{\|\boldsymbol{x}\|^2}\boldsymbol{x}, & \text{if } \boldsymbol{w}^\top\boldsymbol{x} - \alpha < -\epsilon, \\ \boldsymbol{w}, & \text{otherwise}. \end{cases} \tag{5}$$

The previous functional analytic tools have already shown their rich potential in online learning tasks [24–26]. Next is a concept that plays a key role in characterizing solutions of minimization tasks.

**Definition 2** (Fixed point set)**.** Given a mapping $T : \mathbb{R}^L \to \mathbb{R}^L$, its fixed point set is defined as the set $\mathrm{Fix}(T) := \{\boldsymbol{u} \in \mathbb{R}^L : T(\boldsymbol{u}) = \boldsymbol{u}\}$. For example, in the case of a closed convex set $C$, it can be readily verified that $\mathrm{Fix}(P_C) = C$.

The constraints in (2) are closed convex sets. More specifically, $C_0 := \{\boldsymbol{w} \in \mathbb{R}^L : \boldsymbol{w}^\top \boldsymbol{1}_L = 1\}$ is a hyperplane, and $C_1 := \{\boldsymbol{w} \in \mathbb{R}^L : |\boldsymbol{w}^\top \hat{\boldsymbol{\rho}}_n - \eta_n| \leq \epsilon\}$ is a closed hyperslab. Furthermore, the set $\{\boldsymbol{w} \in \mathbb{R}^L : \boldsymbol{w}^\top \boldsymbol{h} \geq c\}$ is also a closed convex set; namely, a closed halfspace [13]. Due to space limitations, this study is not meant to be exhaustive. For this reason, it is generally assumed that $\boldsymbol{w}$ satisfies a finite number of closed convex constraints, $\{C_k\}_{k=0}^K$, $K \in \mathbb{N}_*$; that is, $\boldsymbol{w} \in \bigcap_{k=0}^K C_k$.

However, there is no guarantee that the user-defined constraints $\{C_k\}_{k=0}^K$ are consistent, i.e., $\bigcap_{k=0}^K C_k \neq \emptyset$. To overcome inconsistencies, and thus ensure feasibility, (2) is replaced by

$$\min_{\substack{(\boldsymbol{w},\boldsymbol{E})\in \\ C_0\times\mathbb{R}^{L\times q}}} \Theta_n(\boldsymbol{w}, \boldsymbol{E}) := \min_{\substack{(\boldsymbol{w},\boldsymbol{E})\in \\ C_0\times\mathbb{R}^{L\times q}}} \frac{1}{2}\|(\boldsymbol{R}_n+\boldsymbol{E})^\top \boldsymbol{w} - \eta_n \boldsymbol{1}_q\|^2$$
$$+ \frac{1}{2}\sum_{k=1}^{K} \xi_k d^2(\boldsymbol{w}, C_k) + \sum_{i=1}^{L} s_i|w_i| + \frac{\lambda_E}{2}\|\boldsymbol{E}\|_{\mathrm{F}}{}^2, \tag{6}$$

where $\xi_k \in \mathbb{R}$ is a user-defined positive coefficient, which controls the contribution of the constraint $C_k$ to the previous loss function, $\forall k \in \overline{1, K}$. Notice that $C_0$ is kept implicit as a *hard* constraint. This is due to the requirement that the total available capital has to stay fixed to the value of 1.

The loss function $\Theta_n$ in (6) is non-convex due to the product between the variables $\boldsymbol{w}$ and $\boldsymbol{E}$. However, if one of $\boldsymbol{w}$ or $\boldsymbol{E}$ is fixed, $\Theta_n$ becomes convex with respect to the other variable. This observation will be used in the sequel to devise an alternating optimization strategy: first, given $\boldsymbol{E}$, one solves for $\boldsymbol{w}$, and after a value for $\boldsymbol{w}$ has been obtained, minimization is carried with respect to $\boldsymbol{E}$.

### 3.2. Solving for $\boldsymbol{w}$

Supposing $\boldsymbol{E}$ is fixed, define $\boldsymbol{X}_n := [\boldsymbol{x}_{n1}, \ldots, \boldsymbol{x}_{nq}] := \boldsymbol{R}_n + \boldsymbol{E}$, and verify that for $\boldsymbol{x}_{ni} \neq \mathbf{0}$, the least-squares term in (6) can be written as

$$\frac{1}{2}\|\boldsymbol{X}_n^\top \boldsymbol{w} - \eta_n \mathbf{1}_q\|^2 = \sum_{i=1}^{q} \frac{1}{2}\|\boldsymbol{x}_{ni}\|^2 \frac{\left(\boldsymbol{x}_{ni}^\top \boldsymbol{w} - \eta_n\right)^2}{\|\boldsymbol{x}_{ni}\|^2} = \sum_{i=1}^{q} \frac{1}{2}\|\boldsymbol{x}_{ni}\|^2 d^2(\boldsymbol{w}, \Pi_i),$$

where $\Pi_i$ denotes the hyperplane: $\Pi_i := \{\boldsymbol{u} \in \mathbb{R}^L : \boldsymbol{x}_{ni}^\top \boldsymbol{u} = \eta_n\}$, $i \in \overline{1, q}$. Hence, (6) can be recast as

$$\min_{\boldsymbol{w} \in C_0} \frac{1}{2} \sum_{i=1}^{I} \xi_i d^2(\boldsymbol{w}, C_i) + \sum_{i=1}^{L} s_i |w_i|, \tag{7}$$

where the closed convex sets have been re-ordered to obtain compact expressions, i.e., $I := q + K$, $C_i := \Pi_i$, $\forall i \in \overline{1, q}$, while for $i \in \overline{q+1, I}$, the set $C_i$ is $C_{i-q}$, defined in Section 3.1. Moreover, $\xi_i := \|\boldsymbol{x}_{ni}\|^2$, $\forall i \in \overline{1, q}$, and $\{\xi_i\}_{i=q+1}^{I}$ coincide with the $\{\xi_{i-q}\}_{i=q+1}^{I}$ of Section 3.1.

Formulation (7) can be viewed as a special case of the following optimization task: Given a set of convex, differentiable functions $\{f_i^{(n)}\}_{i=1}^{I}$, whose gradients $\{\nabla f_i^{(n)}\}_{i=1}^{I}$ are Lipschitz continuous [13], along with a set of convex, subdifferentiable functions $\{g_j\}_{j=1}^{J}$, and a set of user-defined positive weights $\{\xi_i\}_{i=1}^{I}$, $\{\lambda_j\}_{j=1}^{J}$, then (7) can be expressed as

$$\min_{\boldsymbol{w} \in \mathbb{R}^L} \iota_{C_0}(\boldsymbol{w}) + \sum_{i=1}^{I} \xi_i f_i^{(n)}(\boldsymbol{w}) + \sum_{j=1}^{J} \lambda_j g_j(\boldsymbol{w}), \tag{8}$$

where $\iota_{C_0}(\cdot)$ stands for the *indicator function of* $C_0$, i.e., the convex function defined as $\iota_{C_0}(\boldsymbol{w}) := 0$, if $\boldsymbol{w} \in C_0$, and $\iota_{C_0}(\boldsymbol{w}) := +\infty$, if $\boldsymbol{w} \notin C_0$. The time index in $f_i^{(n)}$ has been explicitly inserted to stress that this is a time-varying function. The task (7) is indeed a special case of (8); this can be readily verified if $f_i^{(n)} := \frac{1}{2}d^2(\cdot, C_i)$, $\forall i \in \overline{1, I}$, and $J := 1$, with $g_1(\boldsymbol{w}) := \sum_{i=1}^{L} s_i |w_i|$, $\forall \boldsymbol{w} \in \mathbb{R}^L$, and $\lambda_1 = 1$. Recall also that the gradient of $\frac{1}{2}d^2(\cdot, C_i)$ is Lipschitz continuous [13]. It is worth noticing here that the generality of (8) invites extensions of the TLS objective functions in (6) to costs like the $\varepsilon$-insensitive loss, and other functions used extensively in robust statistics [27].

A means of solving (8) could be a time-varying generalization of the celebrated *Projected Subgradient Method (PSM)* [28–30], due to the dependence of the task on $n$, and the subdifferentiability of $\{g_j\}_{j=1}^{J}$. However, in order to establish convergence, a careful choice of the parameters, which control the step along the descent direction, should be made. Usually, they need to be squared summable, but with an unbounded sum. In general, such a strategy on the step-parameters hinders the overall speed of convergence [31], which is undesirable in real-time applications, as the present online asset allocation task. For this reason, *splitting algorithms* [10, 12, 13] are utilized here. By this framework, instead of viewing the loss function in (8) as a whole object, effort is split into minimizing, *in parallel* or *concurrently*, each of the constituent functions comprising the whole loss. For example, the task of minimizing $\iota_{C_0}$ is straightforward; by the definition of $\iota_{C_0}$, the projection mapping $P_{C_0}$ accomplishes this task. For $f_i^{(n)} := \frac{1}{2}d^2(\cdot, C_i)$, the projection mappings $P_{C_i}$ also minimize $f_i^{(n)}$ in a single step. In order to achieve this also for the subdifferentiable functions $\{g_j\}_{j=1}^{J}$, a regularization approach will be adopted as detailed next.

Given a set of positive numbers $\{\gamma_j\}_{j=1}^{J}$, a *regularized* version of (8) is

$$\min_{\boldsymbol{w} \in \mathbb{R}^L} \iota_{C_0}(\boldsymbol{w}) + \sum_{i=1}^{I} \xi_i f_i^{(n)}(\boldsymbol{w}) + \sum_{j=1}^{J} \lambda_j e_{\gamma_j g_j}(\boldsymbol{w}), \tag{9}$$

where $e_{\gamma_j g_j}$ stands for the *Moreau envelope* of $\gamma_j g_j$.

**Definition 3** (Moreau envelopes and proximal mappings [13, 32, 33]). Given a positive number $\gamma$, and a convex function $g : \mathbb{R}^L \to \mathbb{R}$, the *Moreau envelope of* $\gamma g$ [32] is defined as

$$e_{\gamma g}(\boldsymbol{w}) := \min_{\boldsymbol{u} \in \mathbb{R}^L} \gamma g(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{u}\|^2. \tag{10}$$

For example, it is well-known [34] that $e_{|\cdot|}$, where $|\cdot|$ is the absolute function, is a scaled version of the celebrated Huber function, used extensively in robust statistics [27].

Proximal mapping is the one associating every $\boldsymbol{w}$ to the (unique) minimizer of (10), i.e., $\forall \boldsymbol{w} \in \mathbb{R}^L$,

$$\mathrm{Prox}_{\gamma g}(\boldsymbol{w}) := \arg \min_{\boldsymbol{u} \in \mathbb{R}^L} \gamma g(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{u}\|^2.$$

Further, $\mathrm{Fix}(\mathrm{Prox}_{\gamma g}) = \arg\min_{\boldsymbol{w} \in \mathbb{R}^L} g(\boldsymbol{w})$, for any $\gamma > 0$ [13].

The following fact suggests that the regularization offered by (9) to the original task (8) is meaningful.

**Fact 1** ([33]). The minimizers of $g_j$ coincide with those of $e_{\gamma_j g_j}$, i.e., $\arg\min_{\boldsymbol{w} \in \mathbb{R}^L} g_j(\boldsymbol{w}) = \arg\min_{\boldsymbol{w} \in \mathbb{R}^L} e_{\gamma_j g_j}(\boldsymbol{w})$, $\forall \gamma_j > 0$.

The next fact suggests that the proximal mapping with respect to the weighted $\ell_1$-norm yields a well-known form.

**Fact 2** ([13]). Given the function $g(\boldsymbol{w}) := \sum_{i=1}^{L} s_i |w_i|$, $\boldsymbol{w} \in \mathbb{R}^L$, and a positive coefficient $\gamma > 0$, then, if $\boldsymbol{y} := \mathrm{Prox}_{\gamma g}(\boldsymbol{w})$, for some $\boldsymbol{w} \in \mathbb{R}^L$, the $i$-th component of $\boldsymbol{y}$ is given by

$$y_i = \begin{cases} w_i - \gamma s_i, & \text{if } w_i > \gamma s_i, \\ w_i + \gamma s_i, & \text{if } w_i < -\gamma s_i, \\ 0, & \text{if } |w_i| \leq \gamma s_i, \end{cases} \quad \forall i \in \overline{1, L},$$

i.e., $\mathrm{Prox}_{\gamma g}$ is nothing but the *soft-thresholding* operator [35].

**Proposition 1** (Characterization of the minimizers). Define the convex function $\varphi_n := \iota_{C_0} + \sum_{i=1}^{I} \xi_i f_i^{(n)} + \sum_{j=1}^{J} \lambda_j e_{\gamma_j g_j}$. Then for any $\mu \in (0, 2)$, it holds that

$$\arg \min_{\boldsymbol{w} \in \mathbb{R}^L} \varphi_n(\boldsymbol{w}) = \mathrm{Fix}\big(P_{C_0}\big(\boldsymbol{I}_L + \mu(T_n - \boldsymbol{I}_L)\big)\big),$$

where $\boldsymbol{I}_L$ is the $L \times L$ identity matrix, and $T_n : \mathbb{R}^L \to \mathbb{R}^L$ is defined as

$$T_n := \sum_{i=1}^{I} \theta_i P_{C_i} + \sum_{j=1}^{J} \theta_{j+I} \,\mathrm{Prox}_{\gamma_j g_j}, \tag{11}$$

and the coefficients $\{\theta_i\}_{i=1}^{I+J}$ are given by

$$\theta_i := \begin{cases} \beta \xi_i, & \text{if } i \in \overline{1, I}, \\ \beta \frac{\lambda_i}{\gamma_i}, & \text{if } i \in \overline{I+1, I+J}, \end{cases} \tag{12}$$

with $\beta := \left(\sum_{i=1}^{I} \xi_i + \sum_{j=1}^{J} \frac{\lambda_i}{\gamma_i}\right)^{-1}$. Notice that the weights $\{\theta_i\}_{i=1}^{I+J}$ are constructed in such a way that $\sum_{i=1}^{I+J} \theta_i = 1$.

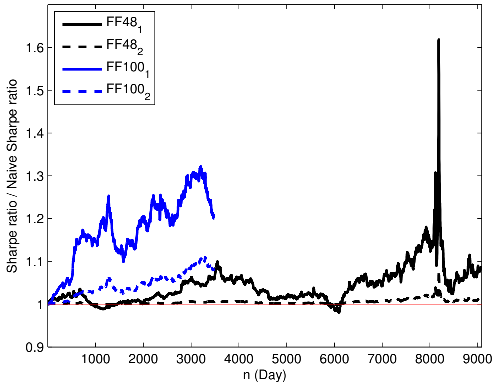


**Fig. 1**. The normalized Sharpe ratio of Algorithm 1, with respect to the naive Sharpe ratio, versus the daily time-horizon. Although the two employed data sets extend over different time periods, they have been aligned to a common starting point for ease in visualization.

*Proof.* Omitted due to space limitations. □

Notice that each term of $T_n$ minimizes a corresponding convex function in $\varphi - \iota_{C_0}$. These combined in parallel minimizing efforts constitute $T_n$. Based on the generic Krasnosel'skiĭ-Mann recursion [36,37], it can be shown that an infinitely often repetition of the mapping $P_{C_0}\big(\boldsymbol{I}_L + \mu(T_n - \boldsymbol{I}_L)\big)$ converges to a minimizer of $\varphi_n$ [11]. This is welcome in a batch setting. However, in an online scenario, such a repetition is unrealistic per time instant $n$. For this reason, in this study, $T_n$ is applied once per $n$ in order to update the vector $\boldsymbol{w}$. Note that this is the first time that the mapping $T_n$ is applied to an online setting, and thus to the online TLS framework. Due to space limitations, performance analysis, which was developed for the general model of (9), will be reported in a future work.

### 3.3. Solving for $\boldsymbol{E}$

Given $\boldsymbol{w}$, the minimization task with respect to $\boldsymbol{E}$ becomes

$$\min_{\boldsymbol{E}\in\mathbb{R}^{L\times q}} \frac{1}{2}\|(\boldsymbol{R}_n + \boldsymbol{E})^\top \boldsymbol{w} - \eta_n \mathbf{1}_q\|^2 + \frac{\lambda_E}{2}\|\boldsymbol{E}\|_{\mathrm{F}}^2.$$

Upon differentiating with respect to $\boldsymbol{E}$, the minimizer becomes

$$\boldsymbol{E} = \left[\frac{\eta_n - \boldsymbol{r}_{n1}^\top \boldsymbol{w}}{\lambda_E + \|\boldsymbol{w}\|^2}, \ldots, \frac{\eta_n - \boldsymbol{r}_{nq}^\top \boldsymbol{w}}{\lambda_E + \|\boldsymbol{w}\|^2}\right] \otimes \boldsymbol{w}, \tag{13}$$

where $\otimes$ stands for the Kronecker product. Algorithm 1 summarizes the basic steps of the previous process.

**Algorithm 1** Online total least-squares by parallel splitting

**for** $n = 1, 2, \ldots,$ **do**
 Available are the current estimates $(\boldsymbol{w}_n, \boldsymbol{E}_n)$, the matrix $\boldsymbol{R}_n$, the sample mean $\hat{\boldsymbol{\rho}}_n$, and the portfolio return $\eta_n$.
 Compute the weights $\{\theta_i\}_{i=1}^{I+J}$ as in (12).
 Calculate the projection mappings $\{P_{C_i}\}_{i=1}^{I}$ as in (4) and (5).
 Calculate the proximal mappings $\{\mathrm{Prox}_{\gamma_j g_j}\}_{j=1}^{J}$ as in Fact 2.
 Generate the mapping $T_n$ in (11), and compute $T_n(\boldsymbol{w}_n)$.
 Define $\boldsymbol{w}_{n+1} \coloneqq P_{C_0}\big(\boldsymbol{w}_n + \mu\big(T(\boldsymbol{w}_n) - \boldsymbol{w}_n\big)\big)$, where $C_0$ is the hyperplane that keeps the total capital equal to 1, and $P_{C_0}$ is computed as in (4).
 Having available $\boldsymbol{w}_{n+1}$, find $\boldsymbol{E}_{n+1}$ by (13).
**Ensure:** The updated estimates $(\boldsymbol{w}_{n+1}, \boldsymbol{E}_{n+1})$.
**end for**

It can be readily verified that the computational complexity of Algorithm 1 scales linearly in the number of unknowns. Specifically, it is of order $\mathcal{O}(qL)$.

**Proposition 2.** Regarding $\Theta_n$ in (6), it holds that $\forall n \in \mathbb{N}$,

$$\Theta_n(\boldsymbol{w}_{n+1}, \boldsymbol{E}_{n+1}) \leq \Theta_n(\boldsymbol{w}_{n+1}, \boldsymbol{E}_n) \leq \Theta_n(\boldsymbol{w}_n, \boldsymbol{E}_n).$$

*Proof.* Due to space limitations, the proof of this claim, as well as the full discussion on the analysis of Algorithm 1 are deferred to the journal version of this paper. □

## 4. NUMERICAL TESTS

Algorithm 1 has been validated on a well-known set of real-world data, taken from the Fama-French (FF) data library [14]. Specifically, tests are carried with '48_Industry_Portfolios_daily.txt' and '100_Portfolios_10x10_Daily.txt'. The first data set is a collection of daily returns for the period of 07/01/1969–09/28/2012, where $L = 48$. The second data set refers to a collection of daily returns for the period of 07/03/1978–06/30/1999, where $L = 100$.

For simplicity, Algorithm 1 has been employed only for the case where the constraints regarding the fixed capital amount, and the closed hyperslab are present. Validation follows the lines of [3]. Specifically, $\eta_n$ is calculated as a moving average process, with uniform weights, onto the naive portfolio returns over the most recent historic data, which extend over a period of a year. The same time period is used also for the value of $M$ in (3). Moreover, every time instant $n$, every portfolio vector $\boldsymbol{w}_n$ is tested against data that were not part of the training phase, meaning data that lie ahead of the time instant $n$. To obtain valid statistical results, portfolio returns were obtained over a future time span of 5 years. Given those portfolio returns, the mean and standard deviation were found so that their ratio formed the so called *Sharpe ratio*. Clearly, the larger the Sharpe ratio, the less risk for a given portfolio vector. The vertical axis of Fig. 1 depicts the Sharpe ratio of the proposed methodology, normalized with respect to the naive asset allocation strategy, where $\boldsymbol{w}_{\text{naive}} \coloneqq \mathbf{1}_L/L$, which according to the discussion in Section 1 is the benchmark [2,3].

Regarding the employed parameters for FF48$_1$, $\mu = 1$, $\epsilon = 0.5$, $\gamma = 10^{-3}$, $\lambda_E = 1$, and $s_i = 10^{-1}$, $\forall i \in \overline{1,L}$. For FF48$_2$, only the value of $s_i$s are changed to 1, in order to impose a more intense $\ell_1$-norm regularization on the total cost. As it can be seen from Fig. 1, this leads to more conservative estimates. For FF100$_1$, $\mu = 1$, $\epsilon = 0.5$, $\gamma = 10^{-3}$, $\lambda_E = 1$, and $s_i = 10^{-2}$, $\forall i \in \overline{1,L}$. Similar to the previous data set, a more conservative approach was followed for FF100$_2$, by changing only the value of $s_i$s to $10^{-1}$.

Fig. 1 speaks for the potential of the proposed methodology, since for almost the entire time period, and for both utilized data sets, Algorithm 1 outperforms the naive asset allocation strategy. The starting point for Algorithm 1 was set equal to $\boldsymbol{w}_{\text{naive}}$. However, Algorithm 1 demonstrated similar behavior for any starting point used. More experimental results will be reported in the journal version of this paper.

## 5. REFERENCES


[1] H. Markowitz, “Portfolio selection,” *Journal of Finance*, vol. 7, no. 1, pp. 77–91, Mar. 1952.

[2] V. DeMiguel, L. Garlappi, and R. Uppal, “Optimal versus naive diversification: How inefficient is the $1/N$ portfolio strategy?” *Review of Financial Studies*, vol. 22, no. 5, pp. 1915–1953, 2009.

[3] J. Brodie, I. Daubechies, C. De Mol, D. Giannone, and I. Loris, “Sparse and stable Markowitz portfolios,” in *Working Paper Series*. European Central Bank, Sept. 2008, no. 936.

[4] B. Scherer, “Can robust portfolio optimization help to build better portfolios?” *J. Asset Management*, vol. 7, no. 6, pp. 374–387, 2007.

[5] B. Efron, T. Hastie, I. M. Johnstone, and R. Tibshirani, “Least-angle regression,” *Annals Statist.*, vol. 32, no. 2, pp. 407–499, 2004.

[6] R. Cont, “Statistical modeling of high-frequency financial data: Facts, models, and challenges,” *IEEE Signal Process. Mag.*, vol. 28, no. 5, pp. 16–25, Sept. 2011.

[7] G. Golub and C. Van Loan, “An analysis of the total least-squares problem,” *SIAM J. Numer. Anal.*, vol. 17, pp. 883–893, 1980.

[8] H. Zhu, G. Leus, and G. B. Giannakis, “Sparsity-cognizant total least-squares for perturbed compressive sampling,” *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2002–2016, May 2011.

[9] B. Schölkopf and A. J. Smola, *Learning with Kernels*. Cambridge, MA: MIT Press, 2001.

[10] P.-L. Lions and B. Mercier, “Splitting algorithms for the sum of two nonlinear operators,” *SIAM J. Numer. Anal.*, vol. 16, pp. 964–979, 1979.

[11] P. L. Combettes, “Solving monotone inclusions via compositions of nonexpansive averaged operators,” *Optimization*, vol. 53, no. 5–6, pp. 475–504, Dec. 2004.

[12] P. L. Combettes and J.-C. Pesquet, “Proximal splitting methods in signal processing,” in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer-Verlag, 2011.

[13] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. New York: Springer, 2011.

[14] K. R. French, “Fama-French data library,” http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html.

[15] R. Tibshirani, “Regression shrinkage and selection via the LASSO,” *J. Royal. Statist. Soc. B.*, vol. 58, no. 1, pp. 267–288, 1996.

[16] T. Huang, Y. Liu, H. Meng, and X. Wang, “Adaptive matching pursuit with constrained total least-squares,” *EURASIP J. Advances Signal Process.*, vol. 2012, no. 76, 2012.

[17] S. Mallat and S. Zhang, “Matching pursuit in a time-frequency dictionary,” *IEEE Trans. Signal Process.*, vol. 41, pp. 3397–3415, 1993.

[18] S. van Huffel and P. Lemmerling, *Total Least-Squares and Error-in-Variables Modeling: Analysis, Algorithms, and Applications*. Boston, MA: Kluwer, 2002.

[19] C. E. Davila, “An efficient recursive total least-squares algorithm for FIR adaptive filtering,” *IEEE Trans. Signal Process.*, vol. 42, no. 2, pp. 268–280, Feb. 1994.

[20] D.-X. Chang, D.-Z. Feng, W.-X. Zheng, and L. Li, “A fast recursive total least-squares algorithm for adaptive IIR filtering,” *IEEE Trans. Signal Proc.*, vol. 53, no. 3, pp. 957–965, Mar. 2005.

[21] F. Rubio, X. Mestre, and D. P. Palomar, “Performance analysis and optimal selection of large minimum variance portfolios under estimation risk,” *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 4, pp. 337–350, Aug. 2012.

[22] T. Hinamoto and S. Maekawa, “Extended theory of learning identification,” *Trans. IEE Japan*, vol. 95-C, no. 10, pp. 227–234, 1975, in Japanese.

[23] K. Ozeki and T. Umeda, “An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties,” *IEICE Trans.*, vol. 67-A, no. 5, pp. 126–132, 1984, in Japanese.

[24] S. Theodoridis, K. Slavakis, and I. Yamada, “Adaptive learning in a world of projections: a unifying framework for linear and nonlinear classification and regression tasks,” *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 97–123, Jan. 2011.

[25] Y. Kopsinis, K. Slavakis, and S. Theodoridis, “Online sparse system identification and signal reconstruction using projections onto weighted $\ell_1$-balls,” *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 936–952, Mar. 2011.

[26] K. Slavakis and I. Yamada, “The adaptive projected subgradient method constrained by families of quasi-nonexpansive mappings and its application to online learning,” *SIAM J. Optim.*, vol. 23, no. 1, pp. 126–152, 2013.

[27] P. J. Huber, *Robust Statistics*. Wiley, 1981.

[28] B. T. Polyak, “A general method for solving extremal problems,” *Dokl. Akad. Nauk. SSSR*, vol. 174, no. 1, pp. 33–36, 1967.

[29] Y. I. Alber, A. N. Iusem, and M. V. Solodov, “On the projected subgradient method for nonsmooth convex optimization in a Hilbert space,” *Math. Programming*, vol. 81, pp. 23–35, 1998.

[30] P.-E. Maingé, “Strong convergence of projected subgradient methods for nonsmooth and nonstrictly convex minimization,” *Set-Valued Analysis*, vol. 16, pp. 899–912, 2008.

[31] B. T. Polyak, “Minimization of unsmooth functionals,” *Zh. uȳchisl. Mat. mat. Fiz.*, vol. 9, no. 3, pp. 509–521, 1969.

[32] J.-J. Moreau, “Fonctions convexes duales et points proximaux dans un espace Hilbertien,” *Acad. Sci. Paris Sér. A Math.*, vol. 255, pp. 2897–2899, 1962.

[33] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*. Berlin: Springer-Verlag, 2004.

[34] C. Michelot and M. L. Bougeard, “Duality results and proximal solutions of the Huber M-estimator problem,” *Appl. Math. Optim.*, vol. 30, pp. 203–221, 1994.

[35] A. Antoniadis and J. Fan, “Regularization of wavelet approximations,” *J. Amer. Statist. Assoc.*, vol. 96, pp. 939–967, 2001.

[36] M. A. Krasnosel'skiĭ, “Two remarks on the method of successive approximations,” *Uspehi Mat. Nauk*, vol. 10, pp. 123–127, 1955.

[37] W. Mann, “Mean value methods in iteration,” *Proc. Amer. Math. Soc.*, vol. 4, pp. 506–510, 1953.