Journal of the Franklin Institute 361 (2024) 107267

Contents lists available at ScienceDirect

Journal of the Franklin Institute

journal homepage: www.elsevier.com/locate/fi

Projected gradient descent method for cardinality-constrained portfolio optimization

Xiao-Peng Li [a], Zhang-Lei Shi [b], Chi-Sing Leung [c,*], Hing Cheung So [c]

[a] State Key Laboratory of Radio Frequency Heterogeneous Integration (Shenzhen University), Shenzhen 518060, China
[b] College of Science, China University of Petroleum (East China), Qingdao, 266580, China
[c] Department of Electrical Engineering, City University of Hong Kong, Hong Kong Special Administrative Region of China

ARTICLE INFO

Keywords:
Sparse portfolio
Mean–variance model
ℓ_0-norm
Projected gradient descent
Non-negative constraint

ABSTRACT

Cardinality-constrained portfolio optimization aims at determining the investment weights on given assets using the historical data. This problem typically requires three constraints, namely, capital budget, long–only, and sparsity. The sparsity restraint allows investment managers to select a small number of stocks from the given assets. Most existing approaches exploit the penalty technique to handle the sparsity constraint. Therefore, they require tweaking the associated regularization parameter to obtain the desired cardinality level, which is time-consuming. This paper formulates the sparse portfolio design as a cardinality-constrained nonconvex optimization problem, where the sparsity constraint is modeled as a bounded ℓ_0-norm. The projected gradient descent (PGD) method is then utilized to deal with the resultant problem. Different from existing algorithms, the suggested approach, called ℓ_0-PGD, can explicitly control the cardinality level. In addition, its convergence is established. Specifically, the ℓ_0-PGD guarantees that the objective function value converges, and the variable sequences converges to a local minimum. To remedy the weaknesses of gradient descent, the momentum technique is exploited to enhance the performance of the ℓ_0-PGD, yielding ℓ_0-PMGD. Numerical results on four real-world datasets, viz. NASDAQ 100, S&P 500, Russell 1000, and Russell 2000 exhibit the superiority of the ℓ_0-PGD and ℓ_0-PMGD over existing algorithms in terms of mean return and Sharpe ratio.

1. Introduction

In the last few years, several algorithms [1–3] were developed for asset management. Portfolio optimization [4–7] is one of asset management goals. It aims to search for a weight vector of the given assets to construct an investment portfolio for the future investment. One popular manner is to exploit the data-driven technique to handle the protfolio optimization [8,9]. Besides, the model-based methods [10] are proposed, which utilizes the mean–variance theory. The benefits of the model-based approaches are demonstrated through case studies on representative dynamic systems [10]. By introducing a risk parameter, portfolio optimization is formulated as a constrained quadratic programming problem with **capital budget** constraint and/or **long-only** constraint [4,5]. The former means that the sum of portfolio weights should be equal to one, while the latter indicates that the weights are nonnegative. In addition, the long-only constraint is able to avoid short-selling in real-world investment. The short-selling may result in an extremely high risk and hence many conservative funds do not allow the short-selling. Since Markowitz's seminal work, various frameworks

* Corresponding author.
E-mail addresses: x.p.li@szu.edu.cn (X.-P. Li), zlshi@upc.edu.cn (Z.-L. Shi), eeleungc@cityu.edu.hk (C.-S. Leung), hcso@ee.cityu.edu.hk (H.C. So).

https://doi.org/10.1016/j.jfranklin.2024.107267
Received 12 May 2023; Received in revised form 20 August 2024; Accepted 12 September 2024
Available online 19 September 2024

have been suggested, such as the equally-weighted portfolio [11], stochastic portfolio [12], high-order portfolio [13], and sparse portfolio [14].

Despite the great success of the mean–variance model, it has been reported that parameter uncertainty and estimation error lead to poor and unstable investment results [15–20]. To this end, robust portfolio optimization model has been suggested. For example, [15] has proposed using shrinkage transformation to remove noisy correlations in the covariance matrix. Specifically, it adopts the shrinkage transformation and random matrix theory-based filter to generate two covariance matrices. Then, the improved covariance matrix is given as a linear combination of the two matrices. A regularized robust estimator [16] aims at providing both reliable estimation of the expected return vector and covariance matrix when the dataset is of small size. Moreover, it has been revealed that the long-only constraint is capable of shrinking the large covariance toward the average covariance [20]. Thereby, imposing the long-only constraint is one way to reduce the covariance matrix estimation error.

On the other hand, traditional Markowitz's formulations yield a dense portfolio, leading to difficulty in management and high transaction costs [6,14,21–24]. Therefore, modern portfolio optimization imposes a **sparsity constraint** on the weight vector, where the sparsity property is characterized using ℓ_0-norm. Compared with the traditional Markowitz's model, the optimization with sparsity regularization/constraint is able to select several important assets to construct a portfolio whose performance closely fits that of the market index. For example, the sparsity constraint allows the management of only 20 assets to effectively replicate Russell 1000 index, which significantly decreases the number of transaction operations and associated fees, particularly for portfolios that necessitate frequent rebalancing. Additionally, the portfolio is able to exclude illiquid stocks.

Though ℓ_0-norm accurately characterizes the sparsity property, it renders the optimization problem NP-hard [25] as it is nonconvex and discrete. Therefore, most existing works suggest replacing the ℓ_0-norm with its convex surrogate, i.e., ℓ_1-norm [26–28]. Nevertheless, when the ℓ_1-norm is adopted for sparsity control, the long-only constraint has to be excluded, resulting in potential short-selling. This is because the capital budget and long-only constraints make the ℓ_1-norm term to be a constant value of 1. In other words, the ℓ_1-norm is inapplicable to achieve sparse portfolios in this case. With the coexistence of long-only and budget constraints, the ℓ_p-norm with $0 < p < 1$ and other nonconvex functions are exploited to handle the sparse portfolio optimization [29–36]. For example, an algorithm based on the interior point approach [33] is proposed to seek for near-optimal sparse portfolios for the ℓ_p-norm regularized Markowitz model. In addition, the $\ell_{0.5}$-norm regularization is incorporated into the mean–variance model [29]. In [23], an efficient sequential method based on the successive convex optimization is derived for the sparse portfolio design.

However, the above-mentioned approaches are unable to directly specify the portfolio cardinality since the sparsity level relies on the associated regularization parameter. To solve this issue, Bourguignon et al. [37] propose formulating sparse portfolio optimization as a sparsity-constrained model and then handle the resultant problem via mixed-integer programming (MIP) [38]. Though the sparsity level can be explicitly controlled, it is computationally demanding. The reason is that solving MIP requires the combination of a continuous optimization procedure and an integer programming procedure. In [39], an alternating direction method of multipliers (ADMM) based algorithm is developed. However, it has a major drawback, that is, the long-only constraint is not considered. In addition, its convergence analysis is limited to the Lagrange value. In [40], a local relaxation algorithm is suggested. It solves a sequence of small, local quadratic programs by initially projecting asset returns onto a reduced metric space. This is followed by clustering within this space to identify sub-groups of assets that best emphasize a suitable measure of similarity among different assets. Besides, work [41] converts the required return as a constraint with the lowest target return and then formulates the portfolio design problem as a max–min optimization. Subsequently, the interval split method and the supergradient approach are adopted as the solver for then resultant optimization task.

On the other hand, the neurodynamic approach has been adopted for cardinality-constrained portfolio optimization [42–44]. In [42], the collaborative neurodynamic approach is exploited to handle portfolio optimization in a bilevel manner. At the lower level, multiple neurodynamic optimization models are utilized to compute Pareto-optimal solutions, while at the higher level, the particle swarm optimization algorithm is employed to optimize weights for diversifying the Pareto-optimal solutions. In [43], the expected return and investment risk are scalarized as a weighted Chebyshev function, while cardinality constraints are effectively represented by introduced binary variables as an upper bound. Subsequently, the cardinality-constrained portfolio selection is formulated as a mixed-integer optimization problem and then the resultant task is addressed using collaborative neurodynamic optimization. In [44], the cardinality-constrained portfolio selection is reformulated as a biconvex optimization problem with conditional value at risk. Then, a two-timescale duplex neurodynamic approach is proposed to handle the reformulated portfolio optimization problem.

This paper devises an algorithm for cardinality-constrained portfolio optimization based on the concept of projected gradient descent (PGD), where the sparsity constraint is formulated as a bounded ℓ_0-norm. Therefore, we are able to explicitly control the cardinality level via an upper bound. The proposed algorithm is named ℓ_0-PGD that is comprised of two alternating steps, namely, gradient descent and nonconvex projection. Moreover, the convergence behavior of the ℓ_0-PGD is analyzed. Furthermore, we exploit the momentum technique to enhance its performance, leading to a variant termed ℓ_0-PMGD. Evaluation is performed using three real-world datasets, namely, NASDAQ 100, S&P 500, Russell 1000, and Russell 2000. Our main contributions are summarized as:

(i) *A new optimization model:* The sparse portfolio optimization is formulated as a cardinality-constrained nonconvex optimization problem. The proposed model enables to explicitly control the sparsity level using an upper bound. In addition, the resultant solution meets capital budget, long-only, and sparsity constraints simultaneously.

(ii) *An effective ℓ_0-PGD algorithm:* To handle the resultant optimization problem, we adopt PDG to devise efficient method (dubbed as ℓ_0-PGD). Besides, its convergence is guaranteed. Specifically, the objective function value is convergent, and variable sequences converges to a local minimum without any assumptions.

(iii) *Good performance:* Experimental results based on NASDAQ 100, S&P 500, Russell 1000, and Russell 2000 exhibit the superiority of ℓ_0-PGD and ℓ_0-PMGD over existing approaches in terms of mean return and Sharpe ratio.

The rest of this article is organized as follows. The background of portfolio optimization is presented in Section 2. In Section 3, the PGD-based algorithm and its variant for sparse portfolio optimization are developed. The convergence behavior of the basic version is analyzed in Section 4. In Section 5, numerical results for algorithm evaluation and comparison are reported. Finally, conclusions are drawn in Section 6.

2. Background

2.1. Notation

We use lower-case or upper-case letters to represent scalars, while vectors and matrices are denoted by bold lower-case and upper-case letters, respectively. Given a vector $\boldsymbol{x}$, x_i stands for its ith entry. The transpose operator is denoted as $(\cdot)^{\mathrm{T}}$, and $\boldsymbol{I}$ represents the identity matrix. In addition, $\mathbf{1}$ and $\mathbf{0}$ represent the vector of ones and the vector of zeros, respectively. Other mathematical symbols are defined after their first appearance.

2.2. Portfolio optimization

Consider N risky assets and D trading days. The returns of the risky assets over the trading days form a daily return matrix $\boldsymbol{R} \in \mathbb{R}^{D\times N}$, where each row of the $\boldsymbol{R}$ denotes the return of the N assets in a trading day. From the daily return matrix, the mean return vector $\boldsymbol{u} \in \mathbb{R}^N$ and the covariance matrix $\boldsymbol{K} \in \mathbb{R}^{N\times N}$ can be obtained. It is worth mentioning that $\boldsymbol{K}$ constructed by risky assets is positive definite with $D > N$ [45–47]. When the number of trading days is not enough, $\boldsymbol{K}$ may be positive semi-definite. In such a case, $\boldsymbol{K}$ can be modified to positive definite by adding $\epsilon\boldsymbol{I}$ to the covariance matrix, where ϵ is a small positive number. The classic Markowitz mean–variance model is a constrained quadratic programming problem, given by

$$\begin{aligned} &\min_{\boldsymbol{x}} \ \boldsymbol{x}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{x} - \beta\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x} \\ &\text{s.t. } \boldsymbol{x}^{\mathrm{T}}\mathbf{1} = 1 \text{ and } x_i \geq 0, i \in [1, N], \end{aligned} \tag{1}$$

where $\boldsymbol{x} = [x_1, \ldots, x_N]^{\mathrm{T}}$ is the portfolio weight vector and $\beta > 0$ is called risk parameter to balance the risk $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{x}$ and return $\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x}$. In general, a larger β generates a higher return. When $\beta = 0$, the model becomes the global minimum variance portfolio [20,24], which minimizes the risk only. The first constraint "$\boldsymbol{x}^{\mathrm{T}}\mathbf{1} = 1$" is **capital budget**, while the second one "$x_i \geq 0$" is **long-only**. Note that if the long-only constraint is removed, then short-selling is allowed.

As the solution of (1) is usually not sparse, the resultant portfolio consists of a large number of assets, resulting in difficult management and high transaction costs [6,24]. Therefore, modern portfolio optimization seeks for a sparse portfolio [6,34,48]. To this end, one idea is to leverage the penalty function method, that is, an ℓ_0-norm term is added into the objective function [23], leading to

$$\begin{aligned} &\min_{\boldsymbol{x}} \ \boldsymbol{x}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{x} - \beta\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x} + \gamma\|\boldsymbol{x}\|_0, \\ &\text{s.t. } \boldsymbol{x}^{\mathrm{T}}\mathbf{1} = 1, \ \text{ and } \ x_i \geq 0, i \in [1, N], \end{aligned} \tag{2}$$

where $\|\boldsymbol{x}\|_0$ is the ℓ_0-norm that refers to the number of nonzero entries in $\boldsymbol{x}$, and $\gamma > 0$ is the penalty parameter to control the sparsity level. Due to the NP-hard issue caused by the ℓ_0-norm, the ℓ_p-norm with $0 < p < 1$ is widely considered to replace the ℓ_0-norm [29,33], resulting in

$$\begin{aligned} &\min_{\boldsymbol{x}} \boldsymbol{x}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{x} - \beta\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x} + \gamma\|\boldsymbol{x}\|_p^p \\ &\text{s.t. } \boldsymbol{x}^{\mathrm{T}}\mathbf{1} = 1, \ \text{and } x_i \geq 0, i \in [1, N], \end{aligned} \tag{3}$$

where $\|\boldsymbol{x}\|_p = (\sum_{i=1}^N x_i^p)^{1/p}$ is the ℓ_p-norm. In practice, when managers design the portfolio, they may require a specific number of selected assets. However, for the algorithms based on penalty function to handle (3), they need to tune a regularization parameter to attain the desired sparsity level. That is, managers cannot explicitly determine the number of desired assets. In addition, since above-mentioned methods replace the ℓ_0 norm with its surrogate function, they are biased estimate.

To explicitly control the cardinality of the portfolio, the sparse portfolio design can be formulated as a cardinality-constrained portfolio optimization:

$$\begin{aligned} &\min_{\boldsymbol{x}} \boldsymbol{x}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{x} - \beta\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x}, \\ &\text{s.t. } \boldsymbol{x}^{\mathrm{T}}\mathbf{1} = 1, \ x_i \geq 0, i \in [1, N], \ \text{and } \|\boldsymbol{x}\|_0 \leq s, \end{aligned} \tag{4}$$

where $s > 0$ is to control the sparsity. Herein, the sparsity constraint is characterized using ℓ_0 norm. The associated upper bound s is used to directly control the sparsity level, such that managers are able to explicitly determine the number of the assets selected

for the portfolio. Among the functions for the sparsity characterization, the ℓ_0 norm is the most precise. However, it makes (4) nonsmooth and nonconvex. Bourguignon et al. suggest to recast (4) as the following mixed-integer programming (MIP) [37]:

$$\begin{aligned} \min_{\boldsymbol{x}} \ & \boldsymbol{x}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{x} - \beta\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t. } & \boldsymbol{x}^{\mathrm{T}}\mathbf{1} = 1, \boldsymbol{a}^{\mathrm{T}}\mathbf{1} \le s, \\ & -\tau a_i \le x_i \le \tau a_i, \quad i = 1, \ldots, N, \\ & a_i \in \{0,1\}, \quad i = 1, \ldots, N, \end{aligned} \tag{5}$$

where $\tau > 0$ is a large number that represents an upper bound for the absolute value of all elements in the optimal solution to model (5). In general, solving MIP requires the combination of a continuous optimization algorithm and an exhaustive search, which is time-consuming. To address these difficulties, $a_i \in \{0,1\}$ can be relaxed to $a_i \in [0,1]$, $i = 1, \ldots, N$, leading to the continuous relaxation of MIP [35]. On the other hand, Shi et al. considers tackling (4) without the nonnegative constraint [39]. Although the method [39] can search for a feasible solution, the solution might result in short-selling. Besides, its sequence convergence is not well-explored.

2.3. Projected gradient descent

PGD is a straightforward and effective method for solving constrained optimization problems [49]. Given that the objective function in this work is a constrained optimization, we provide an introduction to PGD prior to its application. Consider a general constrained optimization problem

$$\min_{\boldsymbol{x}} l(\boldsymbol{x}), \ \text{s.t. } \boldsymbol{x} \in Q, \tag{6}$$

where $l(\boldsymbol{x})$ is a differentiable function and Q is a constraint set. The PGD handles (6) via the following iterative procedure:

$$\boldsymbol{z}^{t+1} = \boldsymbol{x}^t - \zeta l(\boldsymbol{x}^t), \tag{7a}$$

$$\boldsymbol{x}^{t+1} = \mathcal{P}_Q(\boldsymbol{z}^{t+1}), \tag{7b}$$

where $\zeta > 0$ is the step-size. That is, the PGD consists of gradient descent (7a) and projection operation (7b), where the projection operation is formulated as

$$\min_{\boldsymbol{x}} \|\boldsymbol{z}^{t+1} - \boldsymbol{x}\|_2^2, \ \text{s.t. } \boldsymbol{x} \in Q. \tag{8}$$

That is, $\boldsymbol{x} \in Q$ is located on the ℓ_2-norm ball of $\boldsymbol{z}^{t+1}$ and thus $\boldsymbol{x} \in Q$ has the shortest Euclidean distance to $\boldsymbol{z}^{t+1}$.

3. Algorithm development

This section introduces two algorithms, namely, ℓ_0-PGD and ℓ_0-PMGD, designed to address the cardinality-constrained portfolio optimization problem.

3.1. ℓ_0-PGD

We first apply the penalty function technique [50,51] to handle the capital budget constraint in (4). Then, the objective function becomes

$$h(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{x} - \beta\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x} + \alpha(\boldsymbol{x}^{\mathrm{T}}\mathbf{1} - 1)^2. \tag{9}$$

Herein, as the magnitudes of elements in $\boldsymbol{K}$ and $\boldsymbol{u}$ are small, $\alpha = 1$ is large enough. Subsequently, we formulate the cardinality-constrained selection as a nonconvex constrained optimization problem:

$$\begin{aligned} \min_{\boldsymbol{x}} \ & h(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{x} - \beta\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x} + \alpha(\boldsymbol{x}^{\mathrm{T}}\mathbf{1} - 1)^2, \\ & \text{s.t. } x_i \ge 0, i \in [1, N], \|\boldsymbol{x}\|_0 \le s. \end{aligned} \tag{10}$$

To address (10), the penalty function method [50,51] can be adopted as the solver. Specifically, the sparsity constraint is converted to a sparsity term in the objective function and then ℓ_0-norm is replaced with its surrogate function, such as ℓ_1 norm [26] and $\ell_{0.5}$ norm [29]. Nevertheless, the sparsity level of solution is determined via tuning the associated penalty parameter, such that the cardinality of the portfolio cannot be explicitly controlled.

To explicitly determine the number of assets, we exploit PGD to directly handle (10), resulting in

$$\boldsymbol{z}^{t+1} = \boldsymbol{x}^t - \zeta \nabla h(\boldsymbol{x}^t), \tag{11a}$$

$$\boldsymbol{x}^{t+1} = \mathcal{P}_C(\boldsymbol{z}^{t+1}), \tag{11b}$$

where $\nabla h(\boldsymbol{x}) = 2\boldsymbol{K}\boldsymbol{x} - \beta\boldsymbol{u} + 2\alpha\mathbf{1}(\mathbf{1}^{\mathrm{T}}\boldsymbol{x} - 1)$ is the gradient of $h(\boldsymbol{x})$, $\zeta > 0$ is the step-size for gradient descent, and $\mathcal{P}_C(\boldsymbol{z})$ is defined as

$$\min_{\boldsymbol{x}} \ \|\boldsymbol{x} - \boldsymbol{z}\|_2^2$$

Algorithm 1 ℓ_0-PGD

Input: $\boldsymbol{K}$, $\boldsymbol{u}$, s, ζ and $T_{\max}$
  Initialize: $\boldsymbol{x} = \boldsymbol{0}$
  for $t = 1, 2, \cdots, T_{\max}$ **do**
    Calculate $\nabla h(\boldsymbol{x}^t) = 2\boldsymbol{K}\boldsymbol{x} - \beta\boldsymbol{u} + 2\alpha\boldsymbol{1}(\boldsymbol{1}^{\mathrm{T}}\boldsymbol{x} - 1)$
    Update $\boldsymbol{z}^{t+1} = \boldsymbol{x}^t - \zeta\nabla f(\boldsymbol{x}^t)$
    Compute $\boldsymbol{x}^{t+1} = \mathcal{P}_C(\boldsymbol{z}^{t+1})$
    Stop if stopping criterion is met.
  end for
Output: $\boldsymbol{x}^{t+1}$

$$\text{s.t. } x_i \geq 0, i \in [1, N], \|\boldsymbol{x}\|_0 \leq s. \tag{12}$$

The derivation of the projection operator will be discussed in the next subsection. To ensure the convergence, in (11a), the step-size ζ should be $\zeta < 1/(2\lambda_{\max})$, where $\lambda_{\max}$ is the maximum eigenvalue of $\boldsymbol{K} + \alpha\boldsymbol{1}\boldsymbol{1}^{\mathrm{T}}$. The convergence of ℓ_0-PGD will be further analyzed in the next section.

The steps of ℓ_0-PGD are summarized in Algorithm 1. One stopping criterion is to reach the maximum iteration number. In addition, the algorithm will terminate when it is approximately convergent, defined as

$$\frac{\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^t\|_2}{\|\boldsymbol{x}^t\|_2} \leq 10^{-6}. \tag{13}$$

3.2. Projection operator: $\mathcal{P}_C(\boldsymbol{z})$

To seek the solution to (12), without loss of generality, we assume that the elements in $\boldsymbol{z}$ are sorted in descending order, such that $z_i \geq z_j$ for $i < j$. In practical computation, we can first rank the components of the vector in descending order and then restore it after projection. Recall that the projection operator is

$$\begin{aligned} &\min_{\boldsymbol{x}} f(\boldsymbol{x}) = \min_{\boldsymbol{x}} \|\boldsymbol{x} - \boldsymbol{z}\|_2^2 = \min_{\boldsymbol{x}} \sum_{i=1}^{N} (x_i - z_i)^2 \\ &\quad \text{s.t. } x_i \geq 0, i \in [1, N], \|\boldsymbol{x}\|_0 \leq s. \end{aligned} \tag{14}$$

To tackle (14), we consider two cases.

- Consider that $z_i \geq 0$ with $i \in [1, N]$. It is easy to know that $x_i = z_i$ is able to minimize $(x_i - z_i)^2$. Given $\|\boldsymbol{x}\|_0 \leq s$, to minimize $f(\boldsymbol{x})$, it requires setting s numbers of x_i to z_i, where the coordinates of all selected x_i compose ϕ_1. Moreover, the remaining $x_i = 0$ with $i \in \phi_0 = \phi - \phi_1$, where $\phi = \{i | i \in [1, N]\}$. Thereby, the cost function would be

  $$f(\boldsymbol{x}) \propto \sum_{i \in \phi_0} (z_i)^2, \tag{15}$$

  which implies that the minimum of $f(\boldsymbol{x})$ can be attained by minimizing $\sum_{i \in \phi_0}(z_i)^2$. It is clear that $N - s$ smallest z_i are able to minimize $\sum_{i \in \phi_0}(z_i)^2$, that is, $(x_i - z_i)^2$ are set to 0 with s largest z_i. Therefore, the minimum of (14) is attained with

  $$x_i = \begin{cases} z_i, & i \in [1, s], \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

- Consider that $z_i \geq 0$ with $i \in [1, p]$ and $z_i < 0$ with $i \in [p + 1, N]$. First, the objective function can be rewritten as

  $$f(\mathbf{x}) \propto \sum_{i=1}^{p} (x_i - z_i)^2 + \sum_{i=p+1}^{N} (x_i^2 + 2x_i|z_i|). \tag{17}$$

  Since x_i's should be greater than or equal to 0, minimizing $f(\mathbf{x})$ requires

  $$x_i = \begin{cases} z_i, & i \in [1, p], \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

Combining (16) and (18), the solution to (14) is

$$x_i = \begin{cases} z_i, & i \in [1, s] \text{ and } z_i \geq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

It is worth mentioning that after projection we need to restore the orders of $\boldsymbol{x}$.

Algorithm 2 ℓ_0-PMGD

Input: $\boldsymbol{K}$, $\boldsymbol{u}$, s, ζ, η, and $T_{\max}$
  Initialize: $\boldsymbol{x} = \boldsymbol{0}$ and $\boldsymbol{g}^0 = \boldsymbol{0}$
  for $t = 1, 2, \cdots, T_{\max}$ **do**
    Calculate $\nabla h(\boldsymbol{x}^t) = 2\boldsymbol{K}\boldsymbol{x} - \beta\boldsymbol{u} + 2\alpha\boldsymbol{1}(\boldsymbol{1}^{\mathrm{T}}\boldsymbol{x} - 1)$
    Compute $\boldsymbol{g}^t = \eta\boldsymbol{g}^{t-1} + (1-\eta)\nabla h(\boldsymbol{x}^t)$
    Update $\boldsymbol{z}^{t+1} = \boldsymbol{x}^t - \zeta\boldsymbol{g}^t$
    Compute $\boldsymbol{x}^{t+1} = \mathcal{P}_C(\boldsymbol{z}^{t+1})$
    Stop if stopping criterion is met.
  end for
Output: $\boldsymbol{x}^{t+1}$

Table 1
Computational complexity comparison.

Methods	ℓ_0-PGD	$\ell_{0.5}$	GSRP	MIP	LR	CVX
Complexity	$\mathcal{O}(N^2 + N\log(N))$	$\mathcal{O}(N^2 + N\log(N))$	$\mathcal{O}(N^3)$	$\mathcal{O}(2^N N^3)$	$\mathcal{O}(N^3)$	$\mathcal{O}(N^3)$

3.3. ℓ_0-norm PGD with momentum

Since the classic gradient descent is a single-step method, where the next iteration depends only on the current point, ℓ_0-PGD may get stuck in flat spots in the search space. Momentum, which accumulates the gradient of the past steps to determine the update direction, is able to avoid gradient oscillations and coast across flat spots [52]. In view of this, we suggest applying the momentum technique in the gradient descent procedure to enhance the performance of ℓ_0-PGD, resulting in

$$\boldsymbol{g}^t = \eta\boldsymbol{g}^{t-1} + (1-\eta)\nabla h(\boldsymbol{x}^t), \tag{20a}$$

$$\boldsymbol{z}^{t+1} = \boldsymbol{x}^t - \zeta\boldsymbol{g}^t, \tag{20b}$$

$$\boldsymbol{x}^{t+1} = \mathcal{P}_C(\boldsymbol{z}^{t+1}), \tag{20c}$$

where $0 < \eta < 1$ is the momentum parameter. A larger η will accommodate more gradients from the past. It is clear that if $\eta = 0$, then it reduces to ℓ_0-PGD. This momentum based variant is termed ℓ_0-PMGD whose steps are summarized in Algorithm 2.

3.4. Computational complexity

For ℓ_0-PGD, the computational complexity of the gradient descent procedure is $\mathcal{O}(N^2)$. In addition, the complexity of the projection operation is $\mathcal{O}(N\log N)$. Thereby, the total computational complexity is $\mathcal{O}(T_{\max}(N^2 + N\log N))$, where $T_{\max}$ is the number of iterations.

The ℓ_0-PMGD has the same projection operation as to the ℓ_0-PGD and thus its projection operation has the complexity of $\mathcal{O}(N\log N)$. For the gradient descent, its complexity is still dominated by $\mathcal{O}(N^2)$. Thereby, the computational complexity of the ℓ_0-PMGD is also $\mathcal{O}(T_{\max}(N^2 + N\log N))$.

The computational complexity of different algorithms are tabulated in Table 1. We see that ℓ_0-PGD and $\ell_{0.5}$ have the same complexity. Besides, their complexity is lower than the other methods using CVX Toolbox.

4. Convergence analysis

This section analyzes the convergence behavior of ℓ_0-PGD. We first prove that the objective function value is convergent. Subsequently, we prove that the variable sequences converges to a local minimum of (10).

The convergence of the objective function value is established in Theorem 1.

Theorem 1. *In ℓ_0-PGD, the objective function value $h(\boldsymbol{x}^t) = (\boldsymbol{x}^t)^{\mathrm{T}}\boldsymbol{K}\boldsymbol{x}^t - \beta\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x}^t + \alpha((\boldsymbol{x}^t)^{\mathrm{T}}\boldsymbol{1} - 1)^2$ satisfies the following properties:*

(i) When $\boldsymbol{K}$ is positive definite, $h(\boldsymbol{x}^t)$ is lower-bounded.
(ii) If $\zeta < 1/(2\lambda_{\max})$, $h(\boldsymbol{x}^t) > h(\boldsymbol{x}^{t+1})$ for $\boldsymbol{x}^t \neq \boldsymbol{x}^{t+1}$, where $\lambda_{\max}$ is the largest eigenvalue of $\boldsymbol{K} + \alpha\boldsymbol{1}\boldsymbol{1}^{\mathrm{T}}$.

Therefore, the convergence of $\{h(\boldsymbol{x}^t)\}_{t=1}^{\infty}$ can be guaranteed with positive definite $\boldsymbol{K}$ and $\zeta < 1/(2\lambda_{\max})$.

Proof. See Appendix A.

Prior to analyzing the variable sequences, we introduce two important lemmas related to the properties of fixed points of ℓ_0-PGD.

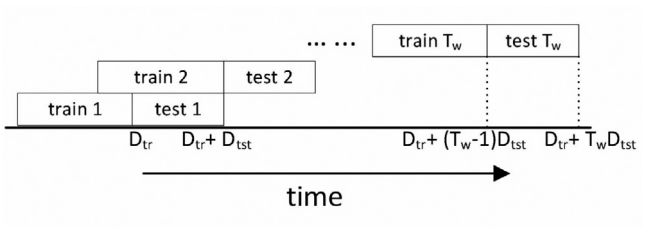

Fig. 1. Portfolio optimization viewed as parameter learning for an adaptive system.

Lemma 1. *Let $\boldsymbol{x}^*$ be a point with $x_i \ast\geq 0$ for $i \in [1, N]$ and $\|\boldsymbol{x}^*\|_0 \leq s$. Define two index sets, namely, ϕ_1 and ϕ_0 such that $\phi_1 = \{i|x_i^* > 0\}$ and $\phi_0 = \{i|x_i^* = 0\}$. A necessary and sufficient condition for $\boldsymbol{x}^*$ to be a fixed point is*

$$-\zeta\Big(2\boldsymbol{K}_i\boldsymbol{x}^*-\beta\mu_i+2\alpha(\mathbf{1}^{\mathrm{T}}\boldsymbol{x}^*-1)\Big)\begin{cases}=0, & \text{if } i\in\phi_1,\\ \leq l^*, & \text{if } i\in\phi_0.\end{cases} \tag{21}$$

where $\boldsymbol{K}_i$ is the ith row of $\boldsymbol{K}$. For the threshold value l^*, there are two cases. For $\|\boldsymbol{x}^*\|_0 = s$, l^* is the minimum of $\{x_i^*\}$ with $i \in \phi_1$. For $\|\boldsymbol{x}^*\|_0 < s$, l^* is set to 0.

Proof. See Appendix B.

It is clear that $2\boldsymbol{K}_i\boldsymbol{x}^*-\beta\mu_i+2\alpha(\mathbf{1}^{\mathrm{T}}\boldsymbol{x}^*-1)$ is the gradient of $h(\boldsymbol{x})$ with respect to (w.r.t.) x_i^*. Thereby, we attain another lemma on the fixed point.

Lemma 2. *Given a fixed point $\boldsymbol{x}^*$, it must satisfy the following properties:*

(i) For $\|\boldsymbol{x}^*\|_0 = s$, $\partial h/\partial x_i^* = 2\boldsymbol{K}_i\boldsymbol{x}^* - \beta\mu_i + 2\alpha(\mathbf{1}^{\mathrm{T}}\boldsymbol{x}^* - 1) = 0$ with $i \in \phi_1$.
(ii) For $\|\boldsymbol{x}^*\|_0 < s$, $\partial h/\partial x_i^* \geq 0$ with $i \in \phi_0$ and $\partial h/\partial x_i^* = 0$ with $i \in \phi_1$.

Proof. The results are easily obtained using Lemma 1.

Based on Lemmas 1 and 2, we derive that a fixed point $\boldsymbol{x}^*$ is a local minimum, given by Theorem 2.

Theorem 2. *For a fixed point $\boldsymbol{x}^*$ of ℓ_0-PGD, $\boldsymbol{x}^*$ is a local minimum of (10).*

Proof. See Appendix C.

We then prove that the sequence $\{\boldsymbol{x}^t\}$ converges to a local minimum.

Theorem 3. *With the initialization of $\|\boldsymbol{x}^1\|_2^2 < +\infty$, the objective function value $h(\boldsymbol{x})$ and variable sequence $\boldsymbol{x}^t$ generated by ℓ_0-PGD have the following properties.*

(i) When $\|\boldsymbol{x}\|_2 \to \infty$, $h(\boldsymbol{x}) \to \infty$.
(ii) $\boldsymbol{x}^t$ is bounded.
(iii) $\forall\ \epsilon > 0$, $\exists\ T_{\max}$ such that $\forall\ t > T_{\max}$, $\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^t\|_2 < \epsilon$.

Proof. See Appendix D.

Based on Theorems 1 and 3, we have the following corollary.

Corollary 1. *$\boldsymbol{x}^t$ converges to a local minimum of (10).*

Proof. See Appendix E.

5. Experiments

In this section, we conduct extensive experiments to evaluate the ℓ_0-PGD and ℓ_0-PMGD. All numerical experiments are performed on a computer with 3.2 GHz CPU and 16 GB memory. Moreover, the version of MATLAB is R2019a.

Table 2
Dataset information.

Dataset	Period	Asset No.	Day	D_{tr}	D_{tst}
NASDAQ 100	01/05/2009–29/01/2016	76	1699	500	60
S&P 500	01/05/2009–29/01/2016	414	1699	500	60
Russell 1000	01/05/2009–29/01/2016	652	1699	800	60
Russell 2000	01/05/2009–29/01/2016	893	1699	800	60

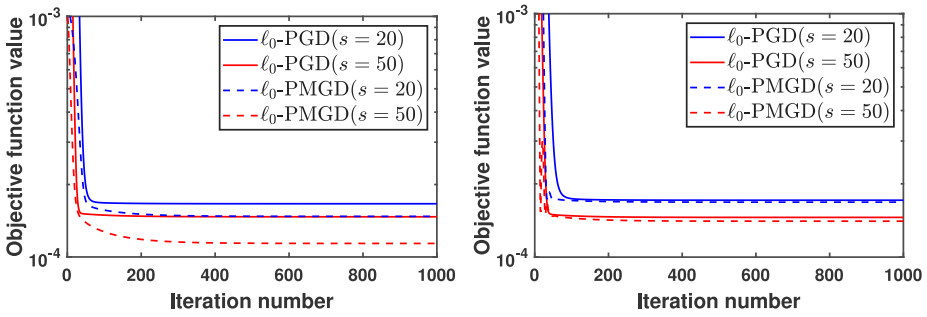

Fig. 2. Convergence behavior of objective function value on Russell 1000 and S&P 500 datasets.

5.1. Experimental settings

We adopt three well-known real-world datasets, namely, NASDAQ 100, S&P 500, and Russell 1000. The datasets are downloaded from Yahoo Finance[1] and contain 1699 trading days from 01-May-2009 to 29-January-2016. Since not all assets have the whole history in this long period, suspended and newly-enlisted assets are excluded following the common practice [42,53]. We utilize a rolling window scheme, shown in Fig. 1, to train and test algorithms [54]. The first window is comprised of $D_{\text{tr}} + D_{\text{tst}}$ days where the first D_{tr} days are used for training, and the remaining D_{tst} days for testing. Then, the window moves forward D_{tst} days to obtain the second one. Note that the second window should have a D_{tr}-day overlap with the first window. Similar to the first window, the second window considers the first D_{tr} days and the following D_{tst} days as the training and test sets, respectively. Furthermore, the window continues to move forward until the whole data are used up. Table 2 tabulates the details of each dataset.

The proposed ℓ_0-PGD and ℓ_0-PMGD are compared with five methods, that is, general sparse risk-parity (GSRP) [23], penalty half thresholding with $\ell_{0.5}$-norm ($\ell_{0.5}$-PHT) [29], mixed-integer programming (MIP) [34], local relaxation method (LR) [40], and a baseline. The baseline is obtained by utilizing the CVX toolbox to solve (1). Note that in CVX, we cannot control the sparsity.

5.2. Performance metric

The prediction performance is evaluated by two metrics, namely, out-of-sample mean return (OSMR) and out-of-sample Sharpe ratio (OSSR). For the ith window, the return

$$\mu_i = \sum_{j=1}^{D_{\text{tst}}} (\boldsymbol{R}_{[i]}\boldsymbol{x}_{[i]})_j, \tag{22}$$

where $\boldsymbol{R}_{[i]}$ and $\boldsymbol{x}_{[i]}$ denote the daily return matrix and portfolio weight vector in the ith window, respectively. The OSMR is computed as

$$\mu = \frac{1}{N_w} \sum_{i=1}^{N_w} \mu_i, \tag{23}$$

where N_w is the number of the windows. It is clear that a large value of μ implies good performance. In addition, the OSSR is defined as the ratio of the return and risk (the variation of the return), given by

$$S = \frac{\mu}{\sigma}, \tag{24}$$

where σ is calculated as:

$$\sigma = \sqrt{\frac{1}{N_w - 1} \sum_{i=1}^{N_w} (\mu_i - \mu)^2}. \tag{25}$$

[1] https://finance.yahoo.com

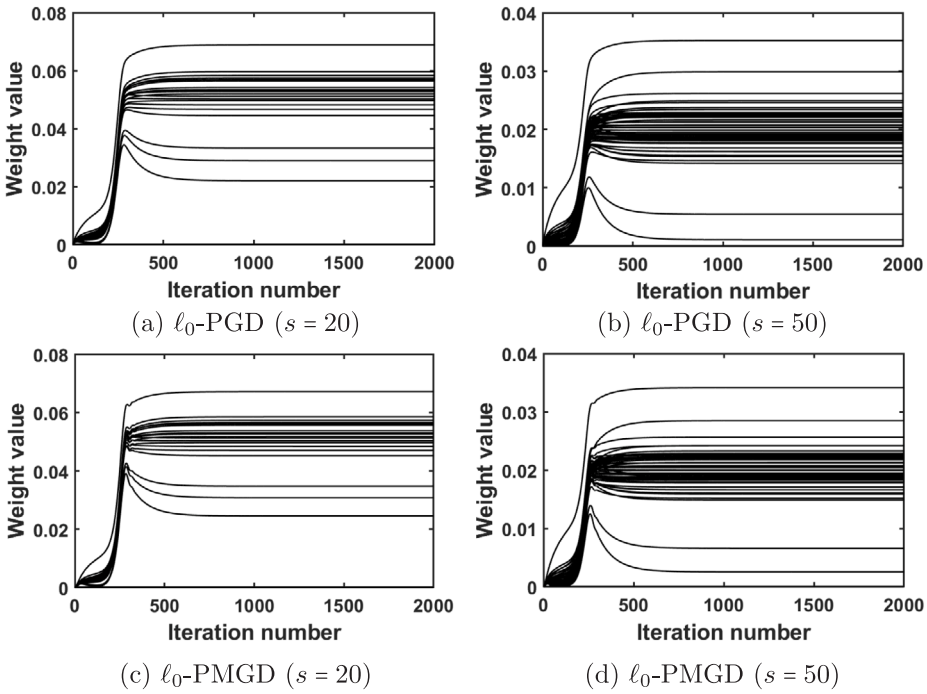

Fig. 3. Dynamics of weight values on Russell 1000 dataset.

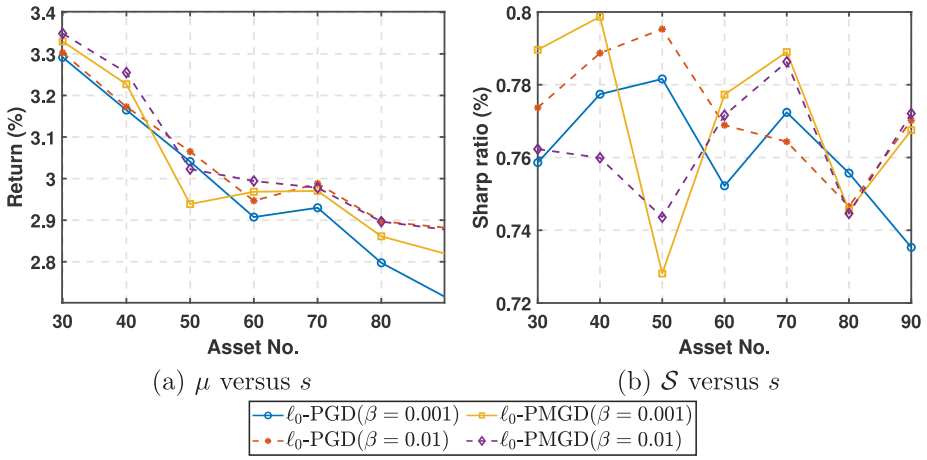

Fig. 4. Influence of β on Russell 1000 dataset.

For a given μ, a large S signifies low risk. Thereby, in finance management, if two portfolios have a similar return, we should choose the one with a higher Sharpe ratio.

5.3. Convergence behavior

This subsection aims at verifying the convergence behavior of ℓ_0-PGD and ℓ_0-PMGD using Russell 1000 and S&P 500 datasets. In Theorem 1, we analyze that the objective function value generated by the ℓ_0-PGD converges. Fig. 2 shows the convergence behavior of the objective function value, where the ℓ_0-PGD and ℓ_0-PMGD adopt two sparsity levels, viz. $s = 20$ and $s = 50$. It is seen that the objective function value decreases and then converges within 400 iterations. In addition, a larger s results in a smaller loss value. Moreover, under the same sparsity, the ℓ_0-PMGD attains a smaller objective function value than the ℓ_0-PGD. Although we have not proved that the objective function value of the ℓ_0-PMGD converges, the empirical results demonstrate that its objective function value is convergent.

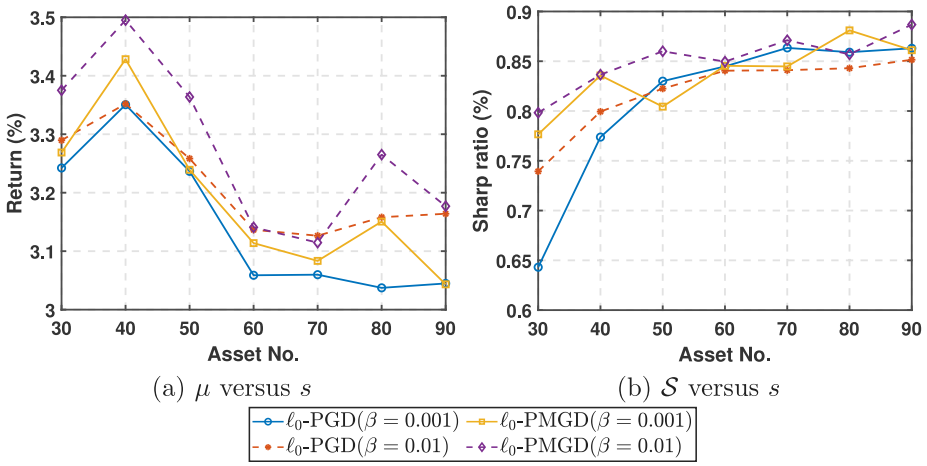

Fig. 5. Influence of β on S&P 500 dataset.

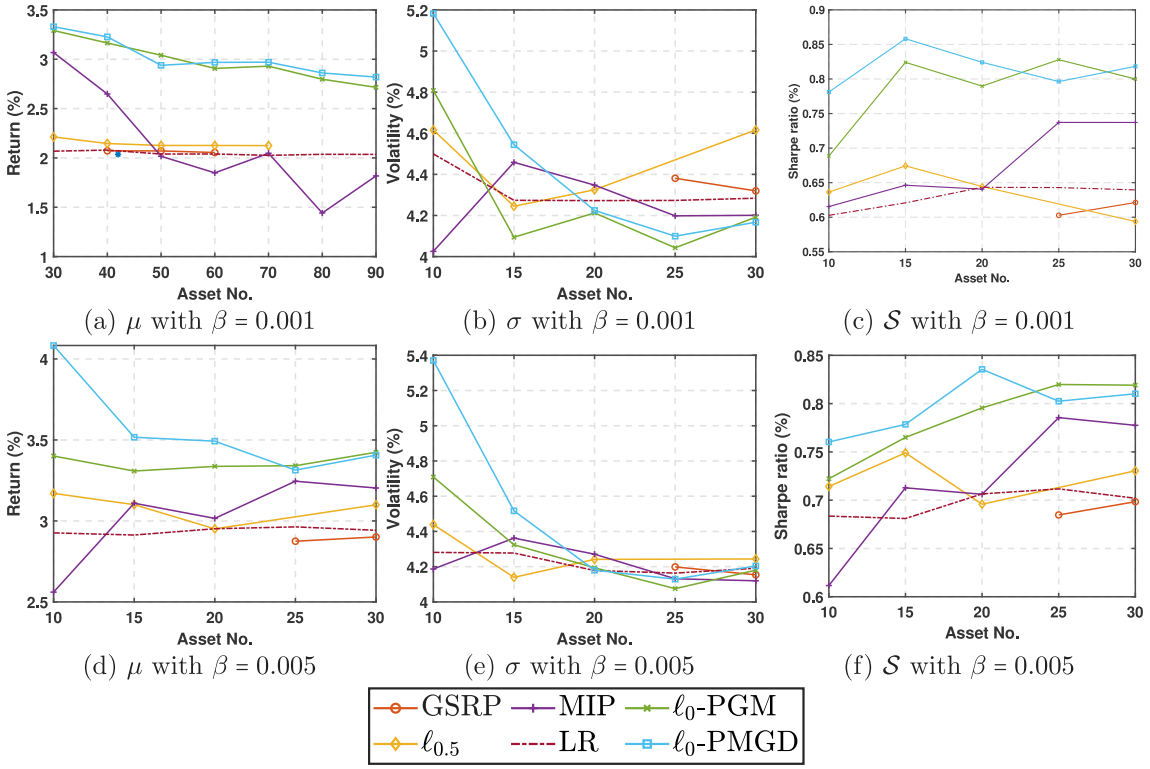

Fig. 6. Performance comparison on NASDAQ 100 dataset.

In Theorem 3, we analyze that the sequence $\{\boldsymbol{x}^t\}_{t=1}^{\infty}$ generated by the ℓ_0-PGD converges. We also verify its convergence behavior by the empirical results, depicted in Figs. 3, where two sparsity levels are considered. It is seen that the estimated weights do not have big changes after 800 iterations. Again, we have not proved that the sequence $\{\boldsymbol{x}^t\}_{t=1}^{\infty}$ of ℓ_0-PMGD converges, but the empirical results demonstrate that the sequence is convergent.

5.4. Influence of risk parameter β

This subsection investigates the performance of the ℓ_0-PGD and ℓ_0-PMGD under different risk parameter values and cardinality levels. Fig. 4 shows the results on Russell 1000 dataset. It is seen that under the same sparsity, a larger β yields a larger value of μ. In addition, given the same sparsity and β, the ℓ_0-PMGD achieves higher return than the ℓ_0-PGD in most of the situations.

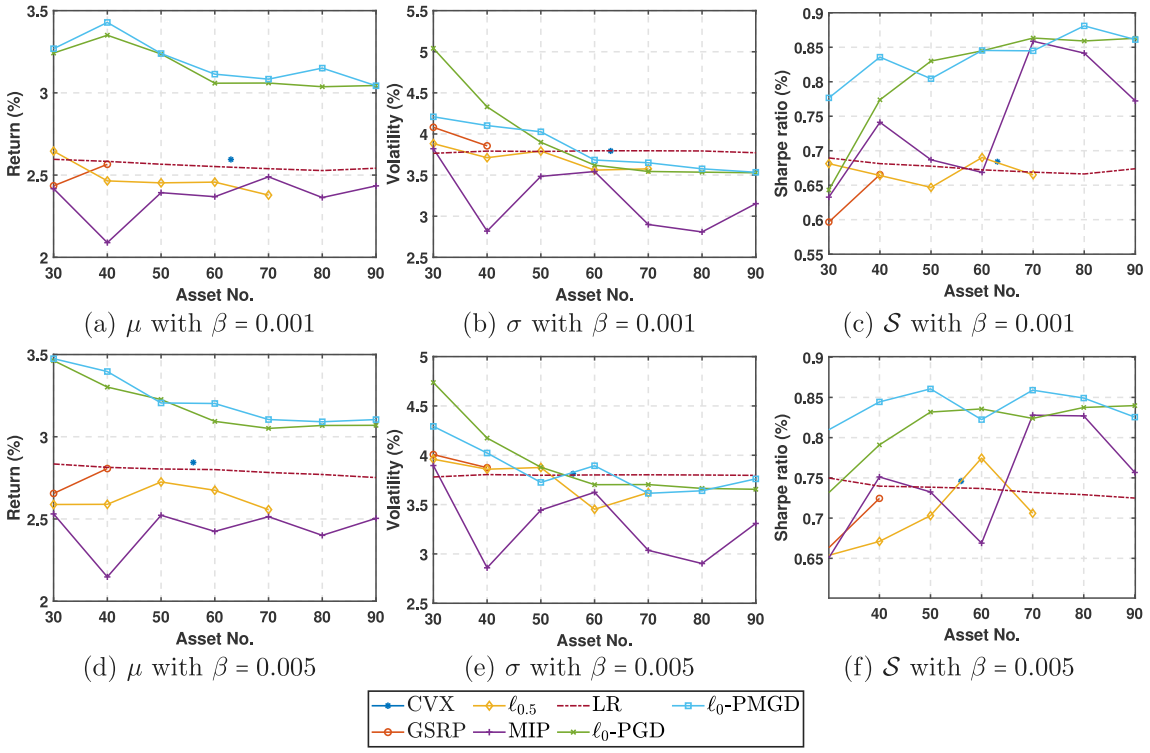

Fig. 7. Performance comparison on S&P 500 dataset.

However, there is no general trend on $\mathcal{S}$ for different values of β. The results on S&P 500 dataset are depicted in Fig. 5. The findings are similar to those of the Russell 1000 dataset. We summarize the effects of β and s as follows:

- A large risk parameter β generally results in a higher return μ under the same sparsity. This can be explained by (10). In (10), a larger β implies that we intend to obtain more return.
- We cannot attain a general trend in $\mathcal{S}$ under different β. However, a common understanding is that a larger return generates a higher risk, i.e., a lower $\mathcal{S}$.
- There is no common trend in $\mathcal{S}$ with different s. However, it should be noted that a large s value leads to high transaction expense.

5.5. Performance comparison

This subsection compares the ℓ_0-PGD and ℓ_0-PMGD with several state-of-the-art approaches using NASDAQ 100, S&P 500, Russell 1000, and Russell 2000 datasets. The results are summarized in Figs. 6 to 9. Note that since CVX solves (1), the sparsity of its solution cannot be adjusted. Thereby, there is one point for CVX in the figures.

- Fig. 6 shows the experimental results by different algorithms on NASDAQ 100 dataset, where two risk parameters are adopted. Figs. 6(a) and 6(c) depict the results with $\beta = 0.001$, while Figs. 6(d) and 6(f) demonstrate the results with $\beta = 0.005$. It is worth mentioning that the results of CVX is not shown since its cardinality level is more than 50.
  It is seen that, given s and β, the ℓ_0-PGD and ℓ_0-PMGD attain larger μ and $\mathcal{S}$ than GSRP, $\ell_{0.5}$-PHT, LR, and MIP. For example, for $s = 20$ and $\beta = 0.001$, the return and Sharpe ratio of the ℓ_0-PGD and ℓ_0-PMGD are greater than 3.3 and 0.77, respectively, as shown in Figs. 8(a) and 8(c). They are much greater than the corresponding values obtained by the competing algorithms. Besides, the volatility of our approaches is higher than the existing methods, indicating that a higher return results in a higher volatility. In addition, the ℓ_0-PMGD is superior to the ℓ_0-PGD in most cases.
  Note that since the GSRP and $\ell_{0.5}$-PHT cannot explicitly control the sparsity, their results cannot cover the whole sparsity range. In other words, they may not attain a portfolio with the desired sparsity in practice.
- The results by different methods on S&P 500 dataset are depicted in Fig. 7. It is seen that the ℓ_0-PGD and ℓ_0-PMGD attain larger returns than their competitors no matter whether the β is large or small. For example, for $s = 40$ and $\beta = 0.001$, the return values of ℓ_0-PGD and ℓ_0-PMGD are greater than 3.3, but those of the competing methods are smaller than 2.6, as shown in Fig. 7(a).

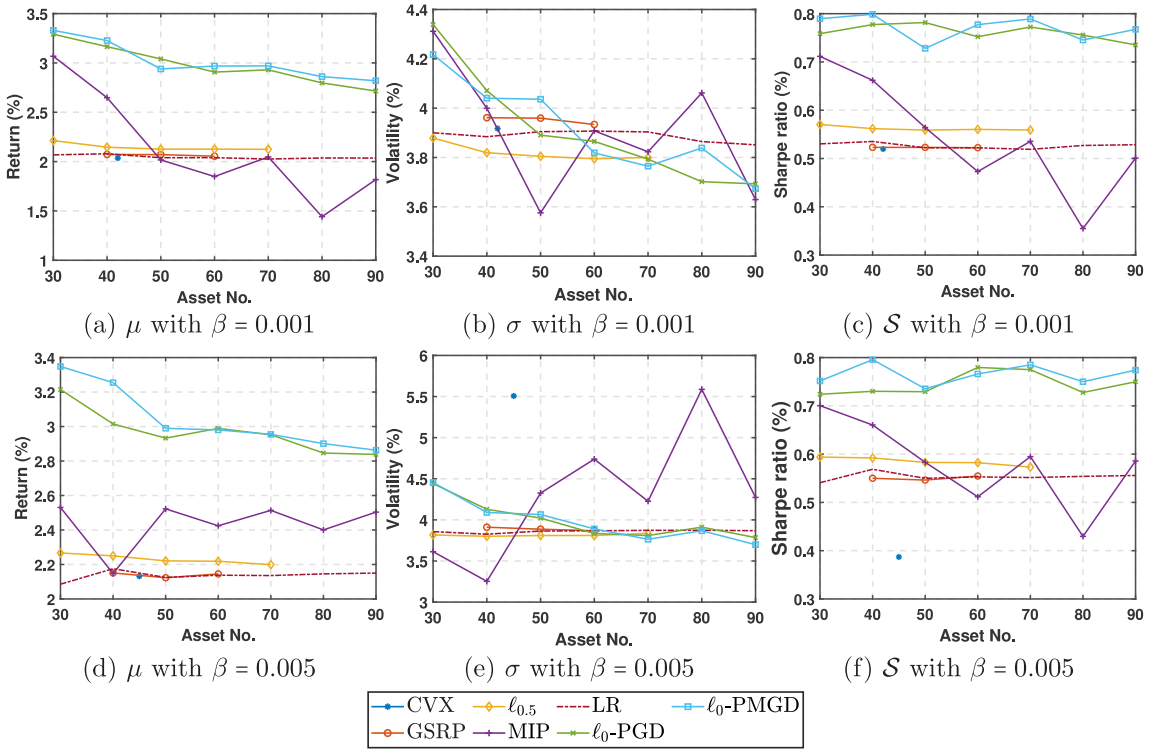

Fig. 8. Performance comparison on Russell 1000 dataset.

For the Sharpe ratio comparison, in general, the performance of ℓ_0-PGD and ℓ_0-PMGD is better than that of the remaining algorithms. There are only a few cases that the latter are better. The first case appears at that $s = 30$ and $\beta = 0.001$, the $\ell_{0.5}$-PHT attains a larger Sharpe ratio than the ℓ_0-PGD, as shown in Fig. 7(c). The second case occurs at $s = 70$ and $\beta = 0.001$, the MIP obtains a slightly larger value of $\mathcal{S}$ than the ℓ_0-PMGD, as shown in Fig. 7(c). Lastly, when $s = 70$ and $\beta = 0.005$, 7(f) shows that the MIP is slightly better. Nevertheless, in all other, the ℓ_0-PGD and ℓ_0-PMGD demonstrate superiority over the CVX, GSRP, $\ell_{0.5}$-PHT, LR, and MIP.

- Figs. 8 and 9 plot the results on Russell 1000 and Russell 2000 datasets. We see that the proposed algorithms are superior to the existing approaches in terms of μ and $\mathcal{S}$ with both large and small β. However, the volatility of the suggested methods is higher than the competing algorithms in most cases.

6. Conclusion

In this work, we have devised two algorithms, namely, ℓ_0-PGD and ℓ_0-PMGD for sparse portfolio design. Different from existing approaches, the ℓ_0-PGD and ℓ_0-PMGD are able to explicitly control the portfolio cardinality via the bounded ℓ_0-norm constraint. Both ℓ_0-PGD and ℓ_0-PMGD exploit the PGD concept to handle the nonconvex constrained optimization problem, resulting in two alternating updates, viz. gradient descent and nonconvex projection. Compared with the ℓ_0-PGD that adopts the standard gradient descent, the ℓ_0-PMGD exploits the momentum technique to remedy the weaknesses of the former. For the ℓ_0-PGD, its objective function value convergence and sequence convergence are guaranteed. Although the convergence analysis of the ℓ_0-PMGD is not provided, empirical results show that its objective function value and sequence converge. Numerical experiments are conducted on three real-world datasets, viz. NASDAQ 100, S&P 500, and Russell 1000, which demonstrate that our proposed algorithms are superior to existing approaches.

CRediT authorship contribution statement

Xiao-Peng Li: Writing – original draft, Software, Methodology, Formal analysis, Data curation. **Zhang-Lei Shi:** Writing – review & editing, Methodology, Data curation. **Chi-Sing Leung:** Writing – review & editing, Supervision, Project administration, Formal analysis, Conceptualization. **Hing Cheung So:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

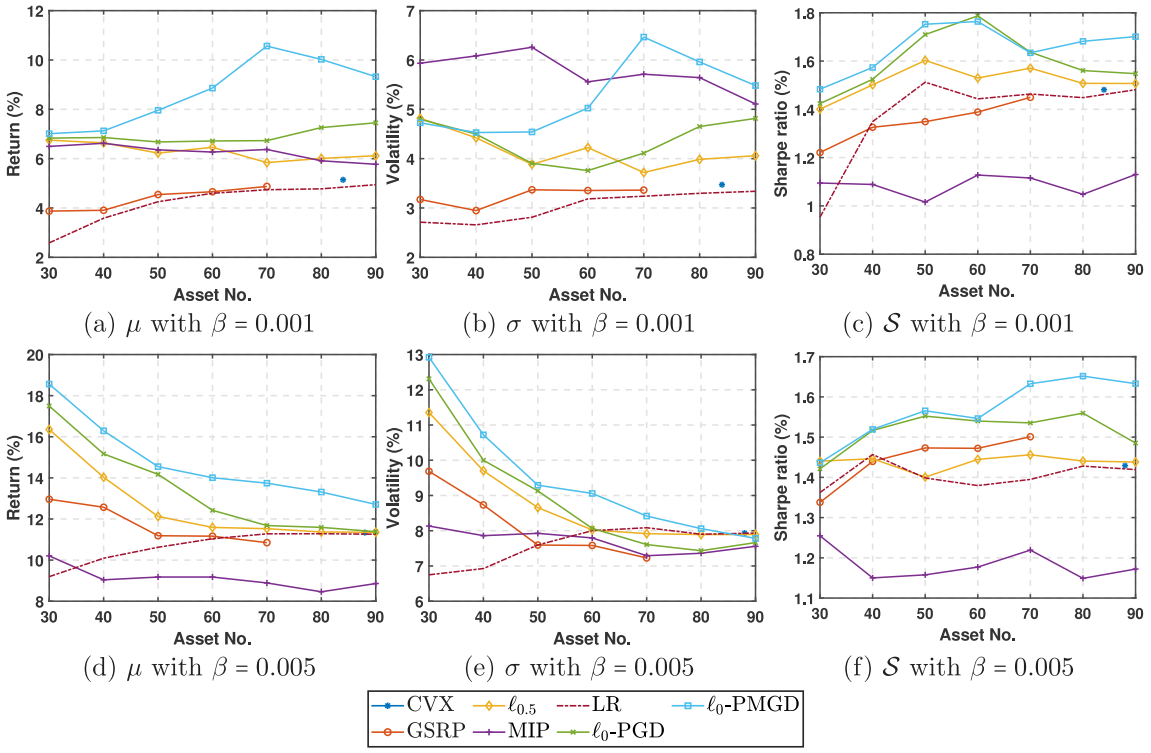

Fig. 9. Performance comparison on Russell 2000 dataset.

Appendix A. Proof of Theorem 1

A.1. Proof of property (i)

The objective function is

$$\begin{aligned} h(\boldsymbol{x}) &= \boldsymbol{x}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{x} - \beta\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x} + \alpha(\boldsymbol{x}^{\mathrm{T}}\mathbf{1} - 1)^2 \\ &= \boldsymbol{x}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{x} - \beta\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x} + \alpha\boldsymbol{x}^{\mathrm{T}}\mathbf{1}\mathbf{1}^{\mathrm{T}}\boldsymbol{x} - 2\alpha\mathbf{1}^{\mathrm{T}}\boldsymbol{x} + \alpha \\ &= \boldsymbol{x}^{\mathrm{T}}(\boldsymbol{K} + \alpha\mathbf{1}\mathbf{1}^{\mathrm{T}})\boldsymbol{x} - (\beta\boldsymbol{u} + 2\alpha\mathbf{1})^{\mathrm{T}}\boldsymbol{x} + \alpha. \end{aligned} \tag{A.1}$$

Since $\boldsymbol{K}$ is symmetric positive definite and $\mathbf{1}\mathbf{1}^{\mathrm{T}}$ is positive semi-definite, $\boldsymbol{K} + \alpha\mathbf{1}\mathbf{1}^{\mathrm{T}}$ is symmetric positive definite. Applying basic algebra, $\boldsymbol{K} + \alpha\mathbf{1}\mathbf{1}^{\mathrm{T}}$ can be decomposed as

$$\boldsymbol{K} + \alpha\mathbf{1}\mathbf{1}^{\mathrm{T}} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}, \tag{A.2}$$

where $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ is a full rank matrix. There are many ways to obtain $\boldsymbol{A}$ from $\boldsymbol{K} + \alpha\mathbf{1}\mathbf{1}^{\mathrm{T}}$, such as eigenvalue decomposition [55] and Cholesky decomposition [56]. Note that different decomposition methods lead to different $\boldsymbol{A}$.

We can then write:

$$h(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} - (\beta\boldsymbol{u} + 2\alpha\mathbf{1})^{\mathrm{T}}\boldsymbol{x} + \alpha. \tag{A.3}$$

Since $\boldsymbol{A}$ has full rank, the inverse of $\boldsymbol{A}^{\mathrm{T}}$ exists. Let $\boldsymbol{b} = \frac{1}{2}(\boldsymbol{A}^{\mathrm{T}})^{-1}(\beta\boldsymbol{u}+2\alpha\mathbf{1})$. Then, we have $(\beta\boldsymbol{u}+2\alpha\mathbf{1}) = 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$, and (A.3) is re-expressed as

$$\begin{aligned} h(\boldsymbol{x}) &= \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} - (\beta\boldsymbol{u} + 2\alpha\mathbf{1})^{\mathrm{T}}\boldsymbol{x} + \alpha \\ &= \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} - 2(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b})^{\mathrm{T}}\boldsymbol{x} + \alpha \\ &= \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} - 2\boldsymbol{b}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} + \|\boldsymbol{b}\|_2^2 - \|\boldsymbol{b}\|_2^2 + \alpha \\ &= \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + c, \end{aligned} \tag{A.4}$$

where $c = \alpha - \|\boldsymbol{b}\|_2^2$. Clearly, $h(\boldsymbol{x})$ is lower bounded. The proof is complete. ■

A.2. Proof of property (ii)

To prove Property (ii), we introduce a surrogate function as an intermediary between contiguous objective function values. The surrogate function is defined as

$$L_s(\boldsymbol{x},\boldsymbol{z}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{x} - \beta\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x} + \alpha(\mathbf{1}^{\mathrm{T}}\boldsymbol{x}-1)^2 + \sum_{i=1}^{N}\iota(x_i) - (\boldsymbol{x}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{x} - 2\boldsymbol{x}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{z} + \boldsymbol{z}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{z}) - \alpha(\mathbf{1}^{\mathrm{T}}\boldsymbol{x} - \mathbf{1}^{\mathrm{T}}\boldsymbol{z})^2 + \frac{1}{2\zeta}\|\boldsymbol{x}-\boldsymbol{z}\|_2^2, \tag{A.5}$$

where $\zeta > 0$, and $\iota(x_i)$ is an indicator function:

$$\iota(x_i) = \begin{cases} \infty, & \text{if } x_i < 0, \\ 0, & \text{if } x_i \geq 0. \end{cases} \tag{A.6}$$

With $\boldsymbol{z} = \boldsymbol{x}^t$, we rewrite the surrogate function as

$$L_s(\boldsymbol{x},\boldsymbol{x}^t) = \frac{1}{2\zeta}\sum_{i=1}^{N}\Big(x_i^2 - 2x_i(x_i^t - \zeta(2\boldsymbol{K}_i^{\mathrm{T}}\boldsymbol{x}^t - \beta u_i + 2\alpha(\mathbf{1}^{\mathrm{T}}\boldsymbol{x}^t - 1))) + 2\zeta\iota(x_i)\Big) + c, \tag{A.7}$$

where $c = \frac{1}{2\zeta}\|\boldsymbol{x}^t\|_2^2 + \alpha - \alpha(\mathbf{1}^{\mathrm{T}}\boldsymbol{x}^t)^2 - (\boldsymbol{x}^t)^{\mathrm{T}}\boldsymbol{K}\boldsymbol{x}^t$ is a constant term w.r.t. $\boldsymbol{x}$, and $\boldsymbol{K}_i$ is the ith row (column) of $\boldsymbol{K}$ ($\boldsymbol{K}$ is symmetric).

As $\iota(x_i) = \infty$ with $x_i < 0$, the minimum of $L_s(\boldsymbol{x},\boldsymbol{x}^t)$ is attained at:

$$x_i^* = \begin{cases} 0, & \text{if } y_i \leq 0, \\ y_i, & \text{if } y_i > 0, \end{cases} \tag{A.8}$$

where $y_i = x_i^t - \zeta(2\boldsymbol{K}_i^{\mathrm{T}}\boldsymbol{x}^t - \beta u_i + 2\alpha(\mathbf{1}^{\mathrm{T}}\boldsymbol{x}^t - 1))$. Furthermore, we know that the loss function value is:

$$L_s(\boldsymbol{x}^*,\boldsymbol{x}^t) \propto -\frac{1}{2\zeta}\sum_{i=1}^{N}(x_i^*)^2. \tag{A.9}$$

It is easy to find that the constrained minimum of the surrogate function (with the number of nonzero elements being less than or equal to s) is achieved by selecting the s largest x_i^*, achieved by the following projection operator

$$\boldsymbol{x}^{t+1} = \mathcal{P}_C(\boldsymbol{x}^t - \zeta(2\boldsymbol{K}\boldsymbol{x}^t - \beta\boldsymbol{u} + 2\alpha(\mathbf{1}^{\mathrm{T}}\boldsymbol{x}^t - 1)\mathbf{1})). \tag{A.10}$$

That is, given $\boldsymbol{x}^t$, the optimal solution to the constrained surrogate function is the same as the result by ℓ_0-PGD.

We now prove that the following expression is positive under $\boldsymbol{x}^{t+1} \neq \boldsymbol{x}^t$.

$$\begin{aligned} &-\Big((\boldsymbol{x}^{t+1})^{\mathrm{T}}\boldsymbol{K}\boldsymbol{x}^{t+1} - 2(\boldsymbol{x}^{t+1})^{\mathrm{T}}\boldsymbol{K}\boldsymbol{x}^t + (\boldsymbol{x}^t)^{\mathrm{T}}\boldsymbol{K}\boldsymbol{x}^t\Big) - \alpha(\mathbf{1}^{\mathrm{T}}\boldsymbol{x}^{t+1} - \mathbf{1}^{\mathrm{T}}\boldsymbol{x}^t)^2 + \frac{1}{2\zeta}\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^t\|_2^2 \\ &= (\boldsymbol{x}^{t+1} - \boldsymbol{x}^t)^{\mathrm{T}}\left(\frac{1}{2\zeta}\boldsymbol{I} - (\boldsymbol{K} + \alpha\mathbf{1}\mathbf{1}^{\mathrm{T}})\right)(\boldsymbol{x}^{t+1} - \boldsymbol{x}^t), \end{aligned} \tag{A.11}$$

When $\zeta < 1/(2\lambda_{\max})$, $\frac{1}{2\zeta}\boldsymbol{I} - (\boldsymbol{K} + \alpha\mathbf{1}\mathbf{1}^{\mathrm{T}})$ is positive definite. Thereby, if $\boldsymbol{x}^{t+1} \neq \boldsymbol{x}^t$, we have

$$(\boldsymbol{x}^{t+1} - \boldsymbol{x}^t)^{\mathrm{T}}\left(\frac{1}{2\zeta}\boldsymbol{I} - (\boldsymbol{K} + \alpha\mathbf{1}\mathbf{1}^{\mathrm{T}})\right)(\boldsymbol{x}^{t+1} - \boldsymbol{x}^t) > 0. \tag{A.12}$$

Since $\boldsymbol{x}^{t+1}$ is the constrained minimizer of the surrogate function $L_s(\boldsymbol{x},\boldsymbol{x}^t)$ (with the number of nonzero elements being less than or equal to s) w.r.t $\boldsymbol{x}$, from (A.12), we attain

$$\begin{aligned} h(\boldsymbol{x}^t) &= L_s(\boldsymbol{x}^t,\boldsymbol{x}^t) \\ &\geq L_s(\boldsymbol{x}^{t+1},\boldsymbol{x}^t) \\ &= h(\boldsymbol{x}^{t+1}) - (\boldsymbol{x}^{t+1})^{\mathrm{T}}\boldsymbol{K}\boldsymbol{x}^{t+1} + 2(\boldsymbol{x}^{t+1})^{\mathrm{T}}\boldsymbol{K}\boldsymbol{x}^t - (\boldsymbol{x}^t)^{\mathrm{T}}\boldsymbol{K}\boldsymbol{x}^t - \alpha(\mathbf{1}^{\mathrm{T}}\boldsymbol{x}^{t+1} - \mathbf{1}^{\mathrm{T}}\boldsymbol{x}^t)^2 + \frac{1}{2\zeta}\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^t\|_2^2 \\ &= h(\boldsymbol{x}^{t+1}) + (\boldsymbol{x}^{t+1} - \boldsymbol{x}^t)^{\mathrm{T}}\left(\frac{1}{2\zeta}\boldsymbol{I} - (\boldsymbol{K} + \alpha\mathbf{1})\right)(\boldsymbol{x}^{t+1} - \boldsymbol{x}^t) + \frac{1}{2\zeta}\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^t\|_2^2 \\ &> h(\boldsymbol{x}^{t+1}) \text{ with } \boldsymbol{x}^t \neq \boldsymbol{x}^{t+1}. \end{aligned} \tag{A.13}$$

Therefore, we have $h(\boldsymbol{x}^t) > h(\boldsymbol{x}^{t+1})$ with $\boldsymbol{x}^t \neq \boldsymbol{x}^{t+1}$. The proof is completed. ■

Appendix B. Proof of Lemma 1

Consider a fixed point $\boldsymbol{x}^*$. It must satisfy the constraints and the following equality:

$$x_i^* = \mathcal{P}_C\Big(x_i^* - \zeta(2\boldsymbol{K}_i\boldsymbol{x}^* - \beta\mu_i + 2\alpha(\mathbf{1}^{\mathrm{T}}\boldsymbol{x}^* - 1))\Big). \tag{B.1}$$

For $i \in \phi_1$, (B.1) holds if and only if $\zeta(2\boldsymbol{K}_i\boldsymbol{x}^* - \beta\mu_i + 2\alpha(\mathbf{1}^{\mathrm{T}}\boldsymbol{x}^* - 1)) = 0$. If $i \in \phi_0$, (B.1) holds if and only if $-\zeta(2\boldsymbol{K}_i\boldsymbol{x}^* - \beta\mu_i + 2\alpha(\mathbf{1}^{\mathrm{T}}\boldsymbol{x}^* - 1)) \leq l^*$. Herein, $l^* > 0$ is the minimum of $\{x_i^*\}$ with $i \in \phi_1$ when $\|\boldsymbol{x}^*\|_0 = s$. For $\|\boldsymbol{x}^*\|_0 < s$, l^* is set to 0. This is because $\|\boldsymbol{x}^*\|_0 < s$ implies that there exists $x_i^* - \zeta(2\boldsymbol{K}_i\boldsymbol{x}^* - \beta\mu_i + 2\alpha(\mathbf{1}^{\mathrm{T}}\boldsymbol{x}^* - 1)) < 0$, resulting in $-\zeta(2\boldsymbol{K}_i\boldsymbol{x}^* - \beta\mu_i + 2\alpha(\mathbf{1}^{\mathrm{T}}\boldsymbol{x}^* - 1)) < -x_i^* \leq 0$ due to $x_i^* \geq 0$.

In conclusion, the compact expression is

$$-\zeta(2\boldsymbol{K}_i\boldsymbol{x}^*-\beta\mu_i+2\alpha(\mathbf{1}^{\mathrm{T}}\boldsymbol{x}^*-1))\begin{cases}=0, & \text{if } i\in\phi_1\\ \le l^*, & \text{if } i\in\phi_0.\end{cases} \tag{B.2}$$

where $l^*>0$ is the minimum of $\{x_i^*\}$ with $i\in\phi_1$ for $\|\boldsymbol{x}^*\|_0=s$. In the case of $\|\boldsymbol{x}^*\|_0<s$, $l^*=0$. The proof is complete. ■

Appendix C. Proof of Theorem 2

It is clear that a fixed point $\boldsymbol{x}^*$ must satisfy the nonnegativity and sparsity constraints, such that $\boldsymbol{x}^*\ge\mathbf{0}$ and $\|\boldsymbol{x}^*\|_0\le s$. We then prove that $h(\boldsymbol{x}^*)\le h(\boldsymbol{x}^*+\boldsymbol{\Delta})$ must hold, where $\boldsymbol{\Delta}$ is a perturbation and meets $\|\boldsymbol{\Delta}\|_2\le\epsilon$ with $\forall\ \epsilon>0$. We consider three cases.

(i) $\|\boldsymbol{x}^*\|_0\le s$ and the support of $\boldsymbol{\Delta}$ is equal to that of $\boldsymbol{x}^*$.
(ii) $\|\boldsymbol{x}^*\|_0=s$ and the support of $\boldsymbol{\Delta}$ is not equal to that of $\boldsymbol{x}^*$.
(iii) $\|\boldsymbol{x}^*\|_0<s$ and the support of $\boldsymbol{\Delta}$ is not equal to that of $\boldsymbol{x}^*$.

Case 1: It is known that $\Delta_i\ne0$ with $i\in\phi_1$ and $\Delta_i=0$ with $i\in\phi_0$. In accordance to Lemma 2, we obtain $\nabla h(\boldsymbol{x}^*)^{\mathrm{T}}\boldsymbol{\Delta}=(\frac{\partial h}{\partial x_1^*},\dots,\frac{\partial h}{\partial x_n^*})^{\mathrm{T}}(\Delta_1,\dots,\Delta_n)=0$. Since $h(\boldsymbol{x})$ is convex and differentiable, we have $h(\boldsymbol{x}^*+\boldsymbol{\Delta})\ge h(\boldsymbol{x}^*)+\nabla h(\boldsymbol{x}^*)^{\mathrm{T}}\boldsymbol{\Delta}=h(\boldsymbol{x}^*)$. Thereby, $\boldsymbol{x}^*$ is a local minimizer of (10).

Case 2: Let Δ_i be a nonzero element where $i\in\phi_0$. To maintain the sparsity constraint, there must be $a_i^*+\Delta_i=0$ with $i\in\phi_1$. Note that we can multiply $\boldsymbol{\Delta}$ with a positive $c_0<1$, resulting in $a_i^*+c_0\Delta_i\ne0$. Thereby, the sparsity constraint cannot be held, denoting that the neighborhood $\boldsymbol{x}^*+c_0\boldsymbol{\Delta}$ of $\boldsymbol{x}^*$ does not meet the sparsity level.

Case 3: To satisfy the nonnegativity constraint, $\Delta_i\ge0$ with $i\in\phi_0$ must be held. Based on Lemma 2, we obtain $\nabla h(\boldsymbol{x}^*)^{\mathrm{T}}\boldsymbol{\Delta}=(\frac{\partial h}{\partial x_1^*},\dots,\frac{\partial h}{\partial x_n^*})^{\mathrm{T}}(\Delta_1,\dots,\Delta_n)\ge0$, resulting in $h(\boldsymbol{x}^*+\boldsymbol{\Delta})\ge h(\boldsymbol{x}^*)+\nabla h(\boldsymbol{x}^*)^{\mathrm{T}}\boldsymbol{\Delta}\ge h(\boldsymbol{x}^*)$. Otherwise, when $\Delta_i<0$ with $i\in\phi_0$, $\boldsymbol{x}^*+\boldsymbol{\Delta}$ does not meet the nonnegativity condition.

In conclusion, $\boldsymbol{x}^*+\boldsymbol{\Delta}$ satisfies the two constraints ($x_i>0$ and $\|\boldsymbol{x}\|_0\le s$) and results in $h(\boldsymbol{x}^*)\le h(\boldsymbol{x}^*+\boldsymbol{\Delta})$. Therefore, $\boldsymbol{x}^*$ is a local minimizer. The proof is complete. ■

Appendix D. Proof of Theorem 3

D.1. Proof of property (i)

From (A.4) in the proof of Theorem 1, as $\boldsymbol{A}$ has full rank, we have $h(\boldsymbol{x})\to+\infty$ as $\|\boldsymbol{x}\|_2\to\infty$. The proof is complete. ■

D.2. Proof of property (ii)

From Theorem 1, we know that $\{h(\boldsymbol{x}^t)\}_{t=1}^{\infty}$ converges and thus $h(\boldsymbol{x}^t)$ is upper and lower bounded. We can prove Property (ii) by contradiction. Suppose $\|\boldsymbol{x}^t\|_2\to+\infty$ results in $h(\boldsymbol{x}^t)\to+\infty$. Based on Property (i), "$\|\boldsymbol{x}^t\|_2\to+\infty$ results in $h(\boldsymbol{x}^t)\to+\infty$" contradicts with Theorem 1. Therefore, $\boldsymbol{x}^t$ is bounded. The proof is complete. ■

D.3. Proof of property (iii)

From (A.12) in Theorem 1, we have

$$(\boldsymbol{x}^{t+1}-\boldsymbol{x}^t)^{\mathrm{T}}\left(\frac{1}{2\zeta}\boldsymbol{I}-(\boldsymbol{K}+\alpha\mathbf{1})\right)(\boldsymbol{x}^{t+1}-\boldsymbol{x}^t)\ge\lambda_{\min}\|\boldsymbol{x}^{t+1}-\boldsymbol{x}^t\|_2^2 \tag{D.1}$$

where $\lambda_{\min}>0$ is the smallest eigenvalue of the positive definite matrix $(\frac{1}{2\zeta}\boldsymbol{I}-(\boldsymbol{K}+\alpha\mathbf{1}\mathbf{1}^{\mathrm{T}}))$. Besides, in accordance to (A.13) and (D.1), we attain

$$\begin{aligned}&h(\boldsymbol{x}^{t+1})+(\boldsymbol{x}^{t+1}-\boldsymbol{x}^t)^T\left(\frac{1}{2\zeta}\boldsymbol{I}-(\boldsymbol{K}+\alpha\mathbf{1})\right)(\boldsymbol{x}^{t+1}-\boldsymbol{x}^t)\le h(\boldsymbol{x}^t)\\ &\Rightarrow h(\boldsymbol{x}^{t+1})+\lambda_{\min}\|\boldsymbol{x}^{t+1}-\boldsymbol{x}^t\|_2^2\le h(\boldsymbol{x}^t).\end{aligned} \tag{D.2}$$

From (D.2), we obtain

$$\lambda_{\min}\|\boldsymbol{x}^{t+1}-\boldsymbol{x}^t\|_2^2\le h(\boldsymbol{x}^t)-h(\boldsymbol{x}^{t+1}). \tag{D.3}$$

By induction on t, we have

$$\lambda_{\min}\sum_{t=1}^{T_{\max}}\|\boldsymbol{x}^{t+1}-\boldsymbol{x}^t\|_2^2\le h(\boldsymbol{x}^1)-h(\boldsymbol{x}^{T_{\max}+1}), \tag{D.4}$$

leading to

$$\lim_{T_{\max}\to\infty}\sum_{t=1}^{T_{\max}}\|\boldsymbol{x}^{t+1}-\boldsymbol{x}^{t}\|_2^2<+\infty. \tag{D.5}$$

Therefore, we attain

$$\lim_{t\to\infty}\|\boldsymbol{x}^{t+1}-\boldsymbol{x}^{t}\|_2^2=0. \tag{D.6}$$

The proof is complete. ■

Appendix E. Proof of Corollary 1

From (D.6) and the boundness of $\{\boldsymbol{x}^t\}_{t=1}^{\infty}$, given $\boldsymbol{x}^1$, $\{\boldsymbol{x}^t\}_{t=1}^{\infty}$ has convergent subsequences and each of them has its limit.

Now, we apply proof by contradiction to analyze that all the limits for a given $\boldsymbol{x}^1$ are the same. If $\{\boldsymbol{x}^t\}_{t=1}^{\infty}$ has at least two different limits, then $\{h(\boldsymbol{x}^t)\}_{t=1}^{\infty}$ has at least two different limits due to $h(\boldsymbol{x}^{t+1})<h(\boldsymbol{x}^t)$ with $\boldsymbol{x}^{t+1}\neq\boldsymbol{x}^t$. This contradicts with the fact that given an initialization of $\|\boldsymbol{x}^1\|_2<+\infty$, $\{h(\boldsymbol{x}^t)\}_{t=1}^{\infty}$ converges. For example, if $\{\boldsymbol{x}^t\}_{t=1}^{\infty}$ has two different limits, namely, $\boldsymbol{x}^*$ and $\boldsymbol{x}^{**}$, then as $t\to\infty$, $h(\boldsymbol{x}^t)$ will alternatively change between $h(\boldsymbol{x}^*)$ and $h(\boldsymbol{x}^{**})$ based on (A.12) and (A.13). This behavior contradicts with Theorem 1 that $\{h(\boldsymbol{x}^t)\}_{t=1}^{\infty}$ is convergent.

To sum up, $\{\boldsymbol{x}^t\}_{t=1}^{\infty}$ has one limit, indicating $\lim_{t\to\infty}\boldsymbol{x}^t=\boldsymbol{x}^*$ where $\boldsymbol{x}^*$ is a fixed point. In Theorem 2, we have proved that the fixed point is a local minimum. Thereby, $\{\boldsymbol{x}^t\}_{t=1}^{\infty}$ converges to a local minimum. The proof is complete. ■

References

[1] L. Zhao, D.P. Palomar, A markowitz portfolio approach to options trading, IEEE Trans. Signal Process. 66 (16) (2018) 4223–4238.
[2] T. Bodnar, S. Dmytriv, Y. Okhrin, N. Parolya, W. Schmid, Statistical inference for the expected utility portfolio in high dimensions, IEEE Trans. Signal Process. 69 (2020) 1–14.
[3] J. Duan, Financial system modeling using deep neural networks (DNNs) for effective risk assessment and prediction, J. Franklin Inst. 356 (8) (2019) 4716–4731.
[4] H. Markowitz, Foundations of portfolio theory, J. Finance 46 (2) (1991) 469–477.
[5] H. Markowitz, Portfolio selection, J. Finance 7 (1) (1952) 77–91.
[6] Y. Lee, M.J. Kim, J.H. Kim, J.R. Jang, W.C. Kim, Sparse and robust portfolio selection via semi-definite relaxation, J. Oper. Res. Soc. 71 (5) (2020) 687–699.
[7] W. Chen, H. Zhang, M.K. Mehlawat, L. Jia, Mean–variance portfolio optimization using machine learning-based stock price prediction, Appl. Soft Comput. 100 (2021) 106943.
[8] M. Sharma, H.S. Shekhawat, Portfolio optimization and return prediction by integrating modified deep belief network and recurrent neural network, Knowl.-Based Syst. 250 (2022) 109024.
[9] Y. Ma, R. Han, W. Wang, Prediction-based portfolio optimization models using deep neural networks, IEEE Access 8 (2020) 115393–115405.
[10] V. Govindan, J. Jayaprakash, C. Park, J.R. Lee, I.N. Cangul, Optimization-based design and control of dynamic systems, Babylon. J. Math. 2023 (2023) 30–35.
[11] V. DeMiguel, L. Garlappi, R. Uppal, Optimal versus naive diversification: How inefficient is the $1/N$ portfolio strategy? Rev. Financ. Stud. 22 (5) (2009) 1915–1953.
[12] G.C. Pflug, M. Pohl, A review on ambiguity in stochastic portfolio optimization, Set-Valued Var. Anal. 26 (4) (2018) 733–757.
[13] R. Zhou, D.P. Palomar, Solving high-order portfolios via successive convex approximation algorithms, IEEE Trans. Signal Process. 69 (2021) 892–904.
[14] J. Brodie, I. Daubechies, C. De Mol, D. Giannone, I. Loris, Sparse and stable Markowitz portfolios, Proc. Natl. Acad. Sci. 106 (30) (2009) 12267–12272.
[15] S. Deshmukh, A. Dubey, Improved covariance matrix estimation with an application in portfolio optimization, IEEE Signal Process. Lett. 27 (2020) 985–989.
[16] J. Liu, D.P. Palomar, Regularized robust estimation of mean and covariance matrix for incomplete data, Signal Process. 165 (2019) 278–291.
[17] D. Goldfarb, G. Iyengar, Robust portfolio selection problems, Math. Oper. Res. 28 (1) (2003) 1–38.
[18] P. Xidonas, R. Steuer, C. Hassapis, Robust portfolio optimization: A categorized bibliographic review, Ann. Oper. Res. 292 (1) (2020) 533–552.
[19] S. Ceria, R.A. Stubbs, Incorporating estimation errors into portfolio selection: Robust portfolio construction, J. Asset Manag. 7 (2) (2006) 109–127.
[20] R. Jagannathan, T. Ma, Risk reduction in large portfolios: Why imposing the wrong constraints helps, J. Finance 58 (4) (2003) 1651–1683.
[21] Z. Dai, F. Wen, Some improved sparse and stable portfolio optimization problems, Finance Res. Lett. 27 (2018) 46–52.
[22] V. DeMiguel, L. Garlappi, F.J. Nogales, R. Uppal, A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms, Manage. Sci. 55 (5) (2009) 798–812.
[23] L. Wu, Y. Feng, D.P. Palomar, General sparse risk parity portfolio design via successive convex optimization, Signal Process. 170 (2020) 107433.
[24] P.J. Kremer, S. Lee, M. Bogdan, S. Paterlini, Sparse portfolio selection via the sorted ℓ_1-norm, J. Bank. Financ. 110 (2020) 105687.
[25] B.K. Natarajan, Sparse approximate solutions to linear systems, SIAM J. Comput. 24 (2) (1995) 227–234.
[26] E.J. Candès, J. Romberg, T. Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, IEEE Trans. Inform. Theory 52 (2) (2006) 489–509.
[27] L. Giarré, F. Argenti, Mixed ℓ_2 and ℓ_1-norm regularization for adaptive detrending with ARMA modeling, J. Franklin Inst. 355 (3) (2018) 1493–1511.
[28] Z. Shi, H. Wang, C.S. Leung, H.C. So, Robust MIMO radar target localization based on Lagrange programming neural networkz, Signal Process. 174 (2020) 107574.
[29] F. Xu, G. Wang, Y. Gao, Nonconvex $L_1/2$ regularization for sparse portfolio selection, Pac. J. Optim. 10 (1) (2014) 163–176.
[30] Y. Teng, L. Yang, B. Yu, X. Song, A penalty PALM method for sparse portfolio selection problems, Optim. Methods Softw. 32 (1) (2017) 126–147.
[31] N. Li, Efficient sparse portfolios based on composite quantile regression for high-dimensional index tracking, J. Stat. Comput. Simul. 90 (8) (2020) 1466–1478.
[32] T. Liu, T.K. Pong, A. Takeda, A successive difference-of-convex approximation method for a class of nonconvex nonsmooth optimization problems, Math. Program. 176 (1) (2019) 339–367.
[33] C. Chen, X. Li, C. Tolman, S. Wang, Y. Ye, Sparse portfolio selection via quasi-norm regularization, 2013, arXiv preprint arXiv:1312.6350.
[34] D. Bertsimas, R. Shioda, Algorithm for cardinality-constrained quadratic optimization, Comput. Optim. Appl. 43 (1) (2009) 1–22.
[35] J. Gao, D. Li, Optimal cardinality constrained portfolio selection, Oper. Res. 61 (3) (2013) 745–761.

[36] D. Bertsimas, R. Cory-Wright, J. Pauphilet, A unified approach to mixed-integer optimization problems with logical constraints, SIAM J. Optim. 31 (3) (2021) 2340–2367.
[37] S. Bourguignon, J. Ninin, H. Carfantan, M. Mongeau, Exact sparse approximation problems via mixed-integer programming: Formulations and computational performance, IEEE Trans. Signal Process. 64 (6) (2015) 1405–1419.
[38] W. Wongrat, A. Younes, A. Elkamel, P.L. Douglas, A. Lohi, Control vector optimization and genetic algorithms for mixed-integer dynamic optimization in the synthesis of rice drying processes, J. Franklin Inst. 348 (7) (2011) 1318–1338.
[39] Z.-L. Shi, X.P. Li, C.-S. Leung, H.C. So, Cardinality constrained portfolio optimization via alternating direction method of multipliers, IEEE Trans. Neural Netw. Learn. Syst. 35 (2) (2022) 2901–2909.
[40] W. Murray, H. Shek, A local relaxation method for the cardinality constrained portfolio optimization problem, Comput. Optim. Appl. 53 (2012) 681–709.
[41] W. Xu, J. Tang, K.F.C. Yiu, J.W. Peng, An efficient global optimal method for cardinality constrained portfolio optimization, INFORMS J. Comput. 36 (2) (2023) 690–704.
[42] M.-F. Leung, J. Wang, Minimax and biobjective portfolio selection based on collaborative neurodynamic optimization, IEEE Trans. Neural Netw. Learn. Syst. 32 (7) (2020) 2825–2836.
[43] M.-F. Leung, J. Wang, Cardinality-constrained portfolio selection based on collaborative neurodynamic optimization, Neural Netw. 145 (2022) 68–79.
[44] M.-F. Leung, J. Wang, H. Che, Cardinality-constrained portfolio selection via two-timescale duplex neurodynamic optimization, Neural Netw. 153 (2022) 399–410.
[45] M.c. Pınar, On robust mean-variance portfolios, Optimization 65 (5) (2016) 1039–1048.
[46] J. Shanken, On the exclusion of assets from tests of the mean variance efficiency of the market portfolio: An extension, J. Finance 41 (2) (1986) 331–337.
[47] L.T. Nielsen, Portfolio selection in the mean-variance model: A note, J. Finance 42 (5) (1987) 1371–1376.
[48] D.X. Shaw, S. Liu, L. Kopman, Lagrangian relaxation procedure for cardinality-constrained portfolio optimization, Optim. Methods Softw. 23 (3) (2008) 411–420.
[49] P.H. Calamai, J.J. Moré, Projected gradient methods for linearly constrained problems, Math. Program. 39 (1) (1987) 93–116.
[50] D.G. Luenberger, Convergence rate of a penalty-function scheme, J. Optim. Theory Appl. 7 (1) (1971) 39–51.
[51] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge Univ. Press., 2004.
[52] N. Qian, On the momentum term in gradient descent learning algorithms, Neural Netw. 12 (1) (1999) 145–151.
[53] G. Guastaroba, M.G. Speranza, Kernel search: An application to the index tracking problem, European J. Oper. Res. 217 (1) (2012) 54–68.
[54] K. Benidis, Y. Feng, D.P. Palomar, Sparse portfolios for high-dimensional financial index tracking, IEEE Trans. Signal Process. 66 (1) (2018) 155–170.
[55] J.N. Franklin, Matrix Theory, Prentice-Hall, Englewood Cliffs, NJ, 1968.
[56] R.A. Horn, C.R. Johnson, Matrix Analysis, Cambridge Univ. Press, New York, NY, USA, 2012.