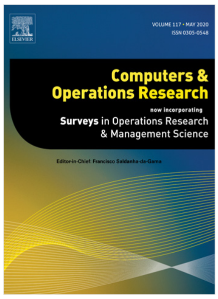


Time-varying mean-variance portfolio selection problem solving via LVI-PDNN

Vasilios N. Katsikis, Spyridon D. Mourtas, Predrag S. Stanimirović, Shuai Li, Xinwei Cao

PII: S0305-0548(21)00309-9
DOI: https://doi.org/10.1016/j.cor.2021.105582
Reference: CAOR 105582

To appear in: *Computers and Operations Research*

Received date : 24 February 2020
Revised date : 22 September 2021
Accepted date : 23 September 2021

Time-Varying Mean-Variance Portfolio Selection Problem Solving via LVI-PDNN

Vasilios N. Katsikis[a,*], Spyridon D. Mourtas[a], Predrag S. Stanimirović[b], Shuai Li[c], Xinwei Cao[d]

[a] *Department of Economics, Division of Mathematics and Informatics, National and Kapodistrian University of Athens, Sofokleous 1 Street, 10559 Athens, Greece*
[b] *University of Niš, Faculty of Sciences and Mathematics, Višegradska 33, 18000 Niš, Serbia*
[c] *Swansea University, Swansea, UK*
[d] *School of Management, Shanghai University, China*

Abstract

It is widely acclaimed that the Markowitz mean-variance portfolio selection is a very important investment strategy. One approach to solving the static mean-variance portfolio selection (MVPS) problem is based on the usage of quadratic programming (QP) methods. In this article, we define and study the time-varying mean-variance portfolio selection (TV-MVPS) problem both in the cases of a fixed target portfolio's expected return and for all possible portfolio's expected returns as a time-varying quadratic programming (TVQP) problem. The TV-MVPS also comprises the properties of a moving average. These properties make the TV-MVPS an even greater analysis tool suitable to evaluate investments and identify trading opportunities across a continuous-time period. Using an originally developed linear-variational-inequality primal-dual neural network (LVI-PDNN), we also provide an online solution to the static QP problem. To the best of our knowledge, this is an innovative approach that incorporates robust neural network techniques to provide an online, thus more realistic, solution to the TV-MVPS problem. In this way, we present an online solution to a time-varying financial problem while eliminating static method limitations. It has been shown that when applied simultaneously to TVQP problems subject to equality, inequality and boundary constraints, the LVI-PDNN approaches the theoretical solution. Our approach is also verified by numerical experiments and computer simulations as an excellent alternative to conventional MATLAB methods.

Keywords: Portfolio selection; time-varying systems; quadratic programming; continuous neural networks.

1. Introduction

Portfolio optimization plays a significant role in financial decisions. Popular fields include insurance costs, risk management, option replication, transaction costs etc. and can be approached efficiently using conventional methods of optimization. For example, in [1], by explicitly integrating a wide range of risk-return portfolio models with return forecasting, transaction costs, and short-sales the authors conclude that the forecasting mechanism more likely yields outperformance when the market is relatively stable. In [2, 3], an optimization problem is defined for minimizing the cost of insurance in portfolios in C[a,b] which constructs the portfolio that replicates the targeted payoff in a subset of states, if the asset span is a lattice-subspace and approached with Riesz spaces theory. In [4], they rebalancing portfolios with transactions costs by extending the standard optimal portfolio theory to an arbitrary number of equally treated assets, a concave utility function, and more broadly stochastic processes. In robotic applications the linear-variational-inequality primal-dual neural network (LVI-PDNN) has been extensively used, see for example [5, 6, 7]. Although several authors have studied various approaches to static portfolio selection problems in conjunction with neural network systems, see for example [8, 9], to the best of our knowledge, this work presents for the first time the time-varying version of the static mean-variance portfolio selection problem (MVPS) that allows the application of the LVI-PDNN to the finance field. This study demonstrates that problems with financial optimization can have an online solution [10, 11], which makes it more realistic. Note that, those problems must be time-varying or converted into a time-varying form first.

The standard approach to solving the static mean-variance portfolio selection (MVPS) problem is based on the usage of quadratic programming (QP) methods. But, we ask the answer to the challenging question: what happens if the MVPS inputs change over time? Because of that, we define and study the time-varying mean-variance portfolio selection (TV-MVPS) problem. The TV-MVPS comprises the properties of a moving average. These properties make the TV-MVPS into an efficient analysis tool suitable to evaluate investments and identify trading opportunities across a continuous-time period. It is known that Zhang neural network (ZNN) can be considered as a predictive dynamics. In [12], the authors claimed that "Static-time and time-varying problems sometimes behave differently. Therefore time-invariant and time-varying problems may require different approaches." In order to achieve the possibility to trace the behavior of the MVPS during the time and introduce a kind of a prediction, we investigate the TV-MVPS problem as

*Corresponding author: Vasilios N. Katsikis

Email addresses: vaskatsikis@econ.uoa.gr (Vasilios N. Katsikis), spirosmourtas@gmail.com (Spyridon D. Mourtas), pecko@pmf.ni.ac.rs (Predrag S. Stanimirović), shuaili@ieee.org (Shuai Li), xinweicao@shu.edu.cn (Xinwei Cao)

a time-varying quadratic programming (TVQP) problem. Also, the ZNN approach is applied as a recognized tool for solving time-varying problems which show better properties compared to a sequence of static problems.

The highlights of this work can be summed up as bellow:
(1) a continuous time-varying quadratic programming (TVQP) financial problem, called TV-MVPS, is introduced and investigated;
(2) a specific LVI-PDNN's structure for approaching the TV-MVPS problem is presented;
(3) the LVI-PDNN's performance on several custom and conventional MATLAB functions for interpolation is investigated;
(4) LVI-PDNN's applications on real-world financial time-series is presented;
(5) a performance comparison between the LVI-PDNN and the `quadprog` MATLAB function is presented.
It is worth noting that the LVI-PDNN's inputs are usually smooth time-varying functions and it has never been used on such noisy data such as financial time-series data.

The paper is organized as follows. Section 2 describes the mean-variance portfolio selection problem and converts the static problem into a time-varying optimization problem. In section 3, the TV-MVPS problem is approached by a linear-variational-inequality based primal-dual neural network (LVI-PDNN). Section 4 contains the proposed algorithmic procedures for data preparation and section 5 contains the numerical examples. The numerical examples use real-world data and examine the efficiency between the LVI-PDNN and the `quadprog` MATLAB function and the efficiency between the proposed algorithmic procedures in different portfolios setup. Finally, the concluding remarks are presented in section 6.

2. Mean-Variance Portfolio Selection Problem

In finance terms, a collection of all stocks or assets held by a public or private institute is known as a portfolio. The portfolio selection problem refers to the optimal distribution of budget on the available stocks such that the expected mean-return is maximized (profit), and the risk is minimized. The factor to measure risk is the variance of the portfolio return, smaller the variance lower will be the risk. This approach was introduced few decades ago by Markowitz's modern portfolio theory [13]. The modern portfolio theory also assumes a perfect market without taxes or transaction costs where short sales are disallowed, but securities are infinitely divisible and can therefore be traded in any (non-negative) fraction.

Over the last decades the Markowitz's modern portfolio theory is studied extensively such as in [14, 15, 16, 17, 18]. For example, the authors in [16] investigate a problem of continuous-time mean-variance portfolio selection with stochastic parameters under a no-bankruptcy limit. In [18], the problem of dynamic portfolio selection is conceived as a Markowitz problem of optimizing mean-variance. They conclude that the single-period Markowitz quadratic programming algorithm can be used with appropriate modifications in the covariance and linear constraint matrices to solve the problem of multi-period asset allocation.

2.1. Definition of the TV-MVPS Financial Problem

Our approach to the mean-variance portfolio selection (MVPS) problem is a time-varying analog of the corresponding static problem defined and studied in a number of papers, such as [13, 14, 15, 16, 17, 18, 19]. The MVPS is a financial optimization problem for assembling a portfolio of assets such that its risk is minimized under a target expected return. As far as we are aware of, our time-varying version of the mean-variance portfolio selection (TV-MVPS) problem is a novel approach that comprises robust techniques from neural networks to provide online, thus more realistic, solution.

The *space of marketed securities* is $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{m\times n}$ where $x_i \in \mathbb{R}^m$ is the security i, $i = 1, 2, \dots, n$, and comprises from the last m observations of its price. In the static MVPS problem the expected return of the marketed space is $r = [r_1, r_2, \dots, r_n] \in \mathbb{R}^n$ where $r_i = \sum_{j=1}^{m} x_i(j)/m \in \mathbb{R}$ is the expected return of the security i, $i = 1, 2, \dots, n$. The expected return of the portfolio is $r_p \in [\min(r), \max(r)] \subseteq \mathbb{R}$ and the variance of the marketed space is $\sigma^2 = \sum_{i=1}^{n}\sum_{i=1}^{n} x_i x_j \sigma_{ij}$ where $\sigma_{ij} = \rho_{ij}\sigma_i\sigma_j$ is the variance and ρ_{ij} is the correlation of i and j securities and σ_i is the variance of i security. That is, $\sigma^2 = X^{\mathrm{T}}CX$ where $C \in \mathbb{R}^{n\times n}$ is the covariance matrix of the marketed space X.

In the TV-MVPS we define the number $\tau \le m-1$, $\tau \in \mathbb{N}$, where τ is a constant number and it denotes the 'number of time periods'. The τ is used for the calculation of the simple moving average. A moving average (*MA*) is a calculation for analyzing data points by creating a series of averages of the complete data set of different subsets. In technical analysis of financial data such as stock prices, returns or volumes of trading the moving average is used as a technical indicator that combines price points of an instrument over a specified time frame divided by the number of data points τ in order to give a single trend line. Hence, a moving average is primarily a lagging indicator and, for that reason, it is one of the most popular tools for technical analysis. The unweighted mean of the previous τ data is called simple moving average (*SMA*). For the observation prices $x_i(t+1), x_i(t+2), \dots, x_i(t+1+\tau)$ of the security i, $i = 1, 2, \dots, n$, the formula of the simple moving average is $SMA_{t+1} = \sum_{j=t+1}^{t+1+\tau} x_i(j)/\tau$. In the case where evaluating consecutive values and a new value, $x_i(t)$, comes into the calculation, the oldest value, $x_i(t+1+\tau)$, drops out. That is, $SMA_t = SMA_{t+1} + (x_i(t) - x_i(t+1+\tau))/\tau$. The chosen period depends on the type of interest movement, for example short, moderate, or long-term. Short-term averages respond quickly to changes in the price of the underlying, while long-term averages are slow to react. Moving average levels can be viewed in financial terms as support in a falling market or resistance in a rising market. In general, there exist several types of moving averages (see [20]). In this paper, we use only one type, the simple moving average (*SMA*). All the rest types of moving averages can be applied to TV-MVPS similarly to *SMA*.

The TV-MVPS comprises from $m-\tau$ in number consecutive values of an *MA* with τ in number observations for each time period. The time $t \in [1, m-\tau]$ denotes the new value that it comes into the calculation of the *MA*. Hence, the expected return of the marketed space is $r(t) = [r_1(t), r_2(t), \dots, r_n(t)] \in \mathbb{R}^n$ where

$$r_i(t) = \sum_{j=t}^{t+\tau} x_i(j)/\tau \in \mathbb{R}$$

is the expected return of the security i, $i = 1, 2, \dots, n$. Obviously, the $r_i(t)$ is an *SMA* and the TV-MVPS problem is built-up on the $r(t)$ for every t. So, the expected return of the portfolio is $r_p(t) \in [\min(r(t)), \max(r(t))] \subseteq \mathbb{R}$, the variance of the marketed space is

$$\sigma^2(t) = \sum_{i=1}^{n}\sum_{i=1}^{n} x_i(t : t+\tau)x_j(t : t+\tau)\sigma_{ij}(t),$$

where $\sigma_{ij}(t) = \rho_{ij}(t)\sigma_i(t)\sigma_j(t)$ is the variance and $\rho_{ij}(t)$ is the correlation of $x_i(t : t+\tau)$ and $x_j(t : t+\tau)$ and $\sigma_i(t)$ is the variance of $x_i(t : t+\tau)$. That is,

$$\sigma^2(t) = X(t : t+\tau, :)^{\mathrm{T}} C(t) X(t : t+\tau, :),$$

where $C(t) = cov(X(t : t+\tau, :)) \in \mathbb{R}^{n\times n}$ is the covariance matrix of the marketed space $X(t : t+\tau, :)$ at time t. The optimal mean-variance portfolio is $\eta(t) = [\eta_1(t), \eta_2(t), \dots, \eta_n(t)]$ where $\eta_i(t)$ is the solution of subsection's 2.1.1 or 2.1.2 optimization problem for the security i, $i = 1, 2, \dots, n$.

The purpose of the number τ is to keep steady the number of observations in the TV-MVPS for each t in $X(t : t+\tau, :)$ while t is moving across the interval $[1, m-\tau]$. Hence, the outcome of the TV-MVPS for each t can be comparable with all the rest outcomes of every other $t \in [1, m-\tau]$ under the same number of observations. Note that, the expected return of the marketed space $r_i(t)$ is a *MA* and it also has the properties of a *MA*. That is, the bigger the τ of the TV-MVPS is the smoother the $r_i(t)$ will be when t is moving across the interval $[1, m-\tau]$, because it filters out the 'noise' from random short-term price fluctuations. Moreover, it affects the optimal mean-variance portfolio $\eta(t)$ in the same way.

We convert the discrete TV-MVPS problem to continuous-time by interpolated the $r(t)$ and the $C(t)$ into continuous functions with any method of preferences. Consequently, $r(t),\ C(t) \in C[0, m-\tau-1]$ where the space $C[0, m-\tau-1]$ is the space of all continuous real functions on the interval $[0, m-\tau-1]$. The optimal mean-variance portfolio is $\eta(t) = [\eta_1(t), \eta_2(t), \dots, \eta_n(t)]$ where $\eta_i(t)$ is the online solution of subsection's 2.1.1 or 2.1.2 optimization problem produced by the LVI-PDNN of section 3.

2.1.1. TV-MVPS with specific expected return target

The *time-varying mean-variance portfolio selection* for a specific target r_p is the solution to the following risk minimization and expected return maximization constrained problem:

$$\begin{aligned}
&\min_{\eta(t)} && \sum_i\sum_j \eta_i(t)\cdot\eta_j(t)\cdot\sigma_{ij}(t) && (1)\\
&\text{subject to} && \sum_i \eta_i(t)\cdot r_i(t) = r_p(t) && (2)\\
& && \sum_i \eta_i(t) = 1 && (3)\\
& && \eta_i(t) \in \mathbb{R}_0^+,\ \forall i, && (4)
\end{aligned}$$

where (1) is the variance $\sigma^2(t)$ of the portfolio $\eta(t)$.

This problem can also be written in the time-varying quadratic programming (TVQP) problem form, by following [21], as follows:

$$\begin{aligned}
&\min_{\eta(t)} && \eta^{\mathrm{T}}(t)\cdot C(t)\cdot\eta(t) && (5)\\
&\text{subject to} && [\mathbf{1}\ \ r(t)]^{\mathrm{T}}\cdot\eta(t) = [1\ \ r_p(t)]^{\mathrm{T}} && (6)\\
& && \mathbf{0} \le \eta(t) \le \mathbf{1}, && (7)
\end{aligned}$$

where $C(t)$ is the covariance matrix of $X(t)$, $\mathbf{0} = [0, 0, \dots, 0] \in \mathbb{R}^n$ denotes the zero vector and $\mathbf{1} = [1, 1, \dots, 1] \in \mathbb{R}^n$ denotes the unit vector.

2.1.2. TV-MVPS with all possible expected returns

In addition, the *time-varying mean-variance portfolio selection* for all possible targets r_p (see [22]) is the solution to the following risk minimization and expected return maximization constrained problem:

$$\begin{aligned}
&\min_{\eta(t)} && \sum_i\sum_j \eta_i(t)\cdot\eta_j(t)\cdot\sigma_{ij}(t) && (8)\\
&\text{subject to} && \sum_i \eta_i(t)\cdot r_i(t) \ge r_p(t) && (9)\\
& && \sum_i \eta_i(t) = 1 && (10)\\
& && \eta_i(t) \in \mathbb{R}_0^+,\ \forall i, && (11)
\end{aligned}$$

where (8) is the variance $\sigma^2(t)$ of the portfolio $\eta(t)$.

By following [21], this problem can also be written in the TVQP problem form as follows:

$$\min_{\eta(t)} \quad \eta^{\mathrm{T}}(t)\cdot C(t)\cdot \eta(t) \tag{12}$$

$$\text{subject to} \quad -r(t)^{\mathrm{T}}\cdot \eta(t) \leq -r_p(t) \tag{13}$$

$$\mathbf{1}^{\mathrm{T}}\cdot \eta(t) = 1 \tag{14}$$

$$\mathbf{0} \leq \eta(t) \leq \mathbf{1}, \tag{15}$$

where $C(t)$ is the covariance matrix of $X(t)$, $\mathbf{0} = [0, 0, \ldots, 0] \in \mathbb{R}^n$ denotes the zero vector and $\mathbf{1} = [1, 1, \ldots, 1] \in \mathbb{R}^n$ denotes the unit vector.

3. Time-Varying Mean-Variance Portfolio Selection Problem via LVI-PDNN

Regarding its fundamental role in mathematical optimization, over the past decades, most aspects of QP have been thoroughly studied. Several methods/algorithms for solving the fundamental static QP problem have been proposed [23]. Such a QP problem has two common general type of solutions. One general type of solution is the numerical algorithms conducted on digital computers and was commonly used to solve static QP problems on a small scale. Nevertheless, in the case of large-scale real-time applications, these numerical algorithms can lead to a decline in performance due to their serial-processing nature [24]. Commonly the less the arithmetic operations are, the less computationally expensive the cube of the Hessian matrix dimension m will be. The other general type of solution is the application of parallel processing which has driven the algorithmic development [25]. Therefore, the comprehensive and thorough research of the recurrent neural network (RNN) has developed and investigated various dynamic and analog solvers. The approximation by neural-dynamic is now considered one of the strong alternatives to QP problems in real-time computing, due to its parallel distributed nature and easiness of hardware implementation [26].

3.1. TV-MVPS problem with specific expected return target via LVI-PDNN

To convert the TV-MVPS problem with specific expected return target into an LVI-PDNN, we need to include the equations (5)-(7) to the coefficients of the LVI-PDNN from [21]. According to the TVQP problem of subsection 2.1.1, if we set

- $G(t) = 2C(t)$
- $x(t) = \eta(t)$
- $D(t) = [\mathbf{1} \;\; r(t)]^{\mathrm{T}}$
- $d(t) = [1 \;\; r_p(t)]^{\mathrm{T}}$
- $B(t) = b(t) = [\;]$
- $g(t) = [\;]$
- $\zeta^-(t) = \mathbf{0}$
- $\zeta^+(t) = \mathbf{1}$,

then the coefficients of the LVI-PDNN can be written as

$$H(t) = \begin{bmatrix} G & -D^{\mathrm{T}}(t) \\ D(t) & 0 \end{bmatrix}, \qquad p(t) = \begin{bmatrix} g(t) \\ -d(t) \end{bmatrix}.$$

Furthermore, the definition of the primal-dual decision vector $y(t)$ can be written as follows, along with the lower and upper boundaries to which it is subject:

$$y(t) = \begin{bmatrix} x(t) \\ \mu(t) \end{bmatrix}, \quad \varsigma^-(t) = \begin{bmatrix} \zeta^-(t) \\ -\varpi 1_v \end{bmatrix}, \quad \varsigma^+(t) = \begin{bmatrix} \zeta^+(t) \\ +\varpi 1_v \end{bmatrix},$$

where

- the constant $\varpi \gg 0$ is the numerical representation and $+\infty$ replacement, large enough for implementation purposes, and the $1v$ vector is the correspondingly dimensioned vector of ones;
- $x(t) \in [\zeta^-(t), \zeta^+(t)]$ clearly denotes the basic parameter decision vector of the primal TVQP (5)-(7);
- $\mu(t) \in \mathbb{R}^l$ is the dual decision variable vector of the equality constraint (6).

3.2. TV-MVPS problem with all possible expected return targets via LVI-PDNN

To convert the TV-MVPS problem with all possible expected return targets into an LVI-PDNN, we need to include the equations (12)-(15) to the coefficients of the LVI-PDNN from [21]. According to the TVQP problem of subsection 2.1.2, if we set

- $G(t) = 2C(t)$
- $x(t) = \eta(t)$
- $D(t) = \mathbf{1}^{\mathrm{T}}$
- $d(t) = 1$
- $B(t) = -r(t)^{\mathrm{T}}$
- $b(t) = -r_p(t)$
- $g(t) = [\;]$
- $\zeta^-(t) = \mathbf{0}$
- $\zeta^+(t) = \mathbf{1}$

then the coefficients of the LVI-PDNN can be written as

$$H(t) = \begin{bmatrix} G(t) & -D^{\mathrm{T}}(t) & B^{\mathrm{T}}(t) \\ D(t) & 0 & 0 \\ -B(t) & 0 & 0 \end{bmatrix}, \qquad p(t) = \begin{bmatrix} g(t) \\ -d(t) \\ b(t) \end{bmatrix}.$$

Furthermore, the definition of the primal-dual decision vector $y(t)$ can be written as follows, along with the lower and upper boundaries to which it is subject:

$$y(t) = \begin{bmatrix} x(t) \\ \mu(t) \\ \varrho(t) \end{bmatrix}, \quad \varsigma^-(t) = \begin{bmatrix} \zeta^-(t) \\ -\varpi 1_v \\ 0 \end{bmatrix}, \quad \varsigma^+(t) = \begin{bmatrix} \zeta^+(t) \\ +\varpi 1_v \\ +\varpi 1_v \end{bmatrix},$$

where

- the constant $\varpi \gg 0$ is the numerical representation and $+\infty$ replacement, large enough for implementation purposes, and the $1v$ vector is the correspondingly dimensioned vector of ones;
- $x(t) \in [\zeta^-(t), \zeta^+(t)]$ clearly denotes the basic parameter decision vector of the primal TVQP (12)-(15);
- $\mu(t) \in \mathbb{R}^l$ is the dual decision variable vector of the equality constraint (14).
- $\varrho(t) \in \mathbb{R}^k$ is the dual decision variable vector of the inequality constraint (13).

3.3. Generalized LVI-PDNN Solution to 3.1 and 3.2 QP problems

The following dynamical system can be used to solve this time-varying QP problem

$$\dot{y}(t) = \gamma(I + H^{\mathrm{T}}(t))(P_{\Omega}(y(t) - (H(t)y(t) + p(t))) - y(t)), \tag{16}$$

where $P_{\Omega}(\cdot)$ is the projection operator (see [21]) and $\gamma > 0$ is known as the design parameter. Within hardware permission, the value of $\gamma > 0$ should be set as the largest, or selected appropriately for simulation or experimental purposes.

While solving static QP problems, beginning with any $y(0) \in \mathbb{R}^{n+l+k}$ initial state, the LVI-PDNN state vector $y(t)$ converges to the equilibrium point y^*, wherein the first n elements are an optimal solution to the TVQP problems (5)-(7) and (12)-(15). Furthermore, the following inequality is true for the static QP's LVI-PDNN solution, [27]:

$$\|y - P_{\Omega}(y - (Hy + p))\|_2^2 \geq \rho \, \|y - y^*\|_2^2 \,, \tag{17}$$

where $\|\cdot\|_2$ corresponds to the vector's two-norm.

To gain a better understanding of LVI-PDNN's real-time convergence, the residual error is defined as

$$e(t) = y(t) - P_{\Omega}(y(t) - (H(t)y(t) + p(t))).$$

Based on the inequality (17), the convergence of the $y(t)$ state vector to the optimal $y^*(t)$ mathematical solution can be reached if $\|e(t)\|_2^2 \to 0$.

3.4. Convergence Analysis

In this subsection, we present, in a formal form, a convergence analysis of the LVI-PDNN model, based on the conceptual framework proposed in [27], by Zhang *et al.* We start with the static general problem, which handles quadratic programming (QP) and linear programming (LP):

$$\min_x \quad x^{\mathrm{T}}Gx/2 + g^{\mathrm{T}}x \tag{18}$$

$$\text{subject to} \quad Dx = d \tag{19}$$

$$Bx \leq b \tag{20}$$

$$\zeta^- \leq x \leq \zeta^+. \tag{21}$$

The proposed primal-dual neural network from [27] could solve online (18)-(21) based on the equivalence of QP/LP, LVI and a system of piecewise linear equations. Then, in our case, equations (6), (7) from [27] can be reformulated as equations (22), (23), respectively, where:

$$y = \begin{bmatrix} x \\ \mu \\ \varrho \end{bmatrix}, \quad \varsigma^- = \begin{bmatrix} \zeta^- \\ -\varpi 1_v \\ 0 \end{bmatrix}, \quad \varsigma^+ = \begin{bmatrix} \zeta^+ \\ +\varpi 1_v \\ +\varpi 1_v \end{bmatrix}. \tag{22}$$

Here, ϖ represents a sufficiently large positive constant (or vector of suitable dimensions). The coefficients in equation (5) are defined as [27]

$$H = \begin{bmatrix} G & -D^{\mathrm{T}} & B^{\mathrm{T}} \\ D & 0 & 0 \\ -B & 0 & 0 \end{bmatrix}, \qquad p = \begin{bmatrix} g \\ -d \\ b \end{bmatrix}. \tag{23}$$

In the following, we will use the same notation as in [27].

Theorem 3.1 ((LP/QP-LVI equivalence) [27], Theorem 1)**.** *It is possible to reformulate the optimization problem (18)-(21) as: find a vector $w^* \in \Omega$ such that $\forall w \in \Omega := \{w|\varsigma^- \leq w \leq \varsigma^+\} \subset \mathbb{R}^{n+l}$,*

$$(w - w^*)^T (Hw^* + p) \geq 0. \tag{24}$$

Theorem 3.2 ((PDNN convergence) [27], Theorem 2)**.** *Starting from arbitrary initial state, the state vector $w(t)$ of the primal-dual neural network (16) converges to the equilibrium w^*, whose first m elements define the optimal solution x^* to the QP model (18)-(21). In fact, the exponential convergence can be reached if there is a constant $\rho > 0$ satisfying $\|w - P_{\Omega}(w - (Hw + p))\|_2^2 \geq \rho \, \|w - w^*\|_2^2$.*

4. Data Preparation

In financial optimization models that we are dealing with, the data inputs are time-series. A time-series is a series of time-indexed data points that means our data input is discrete. Since we are trying to find the online solution to a time-varying optimization problem, we need to convert those data inputs from discrete to continuous-time. We accomplish this by transforming arrays and matrices of time-series to continuous-time functions.

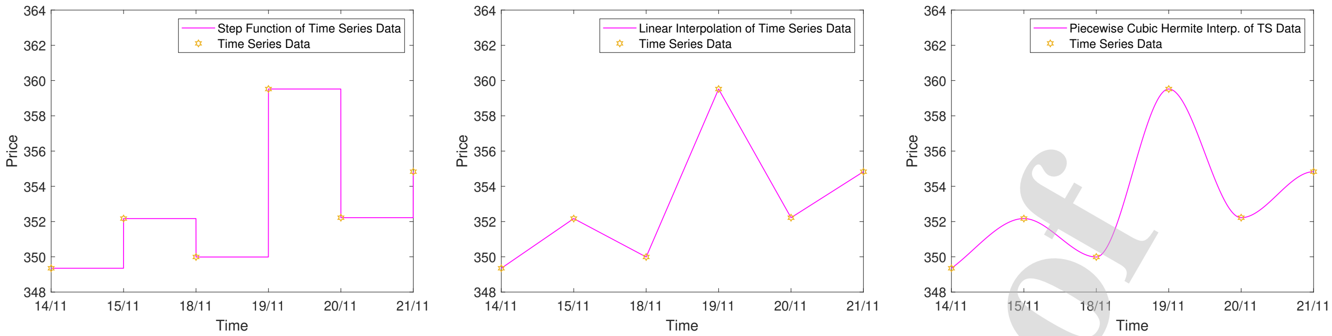
In the TV-MVPS problem, we use the expected return array and the covariance matrix of a portfolio, which comprises of time-series. The following Alg. 1 shows how we construct that array r and matrix C.

Algorithm 1 Algorithm for the data preparation of the portfolio's expected return and covariance.

Input: The marketed space $X = [x_1, x_2, \ldots, x_n]$ which is a matrix of n time series as column vectors of m prices, the moving average's number of time periods $\tau \leq m - 1$, $\tau \in \mathbb{N}$.

```
Set [m, n] =size(X)
Set r =zeros(m − τ, n)
Set C{m − s, 1} = {}
for i = 1 : m − τ do
    Set h =max(X(i : τ + i − 1, :))
    Set C{i, 1} = 100∗cov(X(i : τ + i − 1, :)./h)
    Set r(i, :) =mean(X(i : τ + i − 1, :)./h)
end for
```

Output: The C structure array comprises of the covariance matrices for each time periods of all time-series of the normalized portfolio and the matrix r comprises of the expected return for a number of time periods of each time-series of the normalized portfolio.



(a) Step Function of Time. (b) Linear Interpolation. (c) P.C.Hermite Interpolation.

Figure 1: Data interpolation methods.

Note that we normalize the portfolio's data for each time period in order to have a correct covariance matrix for comparison purposes. Also, without loss of generality, we multiply the covariance matrix C with the number 100, which causes the variance of the portfolio to be in %.

In this paper, three popular interpolation methods are employed that are also offered by *MathWorks*, and we demonstrate how to use them with LVI-PDNN to produce faster results in the case where input data are given in the form of time-series. These interpolation methods are the step function, the linear and the piecewise cubic Hermite (P.C.Hermite). A graphic illustration of these methods is given in Figs. 1a, 1b and 1c, respectively. Note that the data used in Fig. 1 are the daily close prices of *Tesla, Inc. (TSLA)* in the year 2019.

Particularly, for the step function interpolation method, we present the procedure where we convert the time-series arrays and matrices into a piecewise constant function in Alg. 2. In addition, for the linear and the P.C.Hermite interpolation methods the procedures are presented in [28]. Thus, we developed two MATLAB functions, called `sfots` and `sfotss`, for experimental purposes to precisely implement Alg. 2. Furthermore, taken from [28], the MATLAB functions employed for linear interpolation are the `linots` and `linotss`, and for P.C.Hermite interpolation are the `pchinots` and `pchinotss`. Note that the interpolation functions are used on C and r.

In addition, it is possible to split the time periods into daily weekly, monthly, quarterly, annual and their combinations in finance. Yet their results may not be equal in number for two different time periods of the same division, which is due to the fact that financial markets may be close (special days of the calendar), the year may be leap, one month may have fewer days, etc. To solve the problem of missing observations between periods of the same division, we use the parameter ω for each t within the LVI-PDNN, which divides the observations into the time periods. That is, we employ $f_r(\omega t)$ and $f_C(\omega t)$ instead of $f_r(t)$ and $f_C(t)$. The custom function `omega`, introduced in [29], requires as input the time period t and the vector *noep*, which contains the number of observations in each period, and outputs the ω parameter.

Note that most of the custom functions employed in this section are taken from [28, 29] and can be downloaded from `https://github.com/SDMourtas/TV-MVPSTC-CC`. Furthermore, the `ode15s` MATLAB solver is employed on (16) to generate the online solution of the TV-MVPS problem. Lastly, the LVI-PDNN's solutions are checked, for comparison purposes, against the assumed theoretical solutions produced by the `quadprog` MATLAB function.

Algorithm 2 Algorithm for the step function of the expected return and the covariance.

Input: The marketed space $X = [x_1, x_2, \dots, x_n]$, which is a matrix of n time series as column vectors of m prices, and the moving average's number of time periods $\tau \leq m-1$, $\tau \in \mathbb{N}$.

Construct C and r from Alg. 1.
function g = sfots(data,t)
  Set T as the floor price of t
  return $g = data(T+1,:)$
end function
Set $f_r = @(t)\text{sfots}(r,t)$
function h = sfotss(data,t)
  Set T as the floor price of t
  return $h = data\{T+1\}$
end function
Set $f_C = @(t)\text{sfotss}(C,t)$

Output: The conversion of the covariance matrix and the expected return of n time series into time-varying step functions, $f_C(t)$ and $f_r(t)$, respectively.

5. Numerical Examples

In this section, for investigating the performance of LVI-PDNN, three numerical examples under several portfolio setup are presented. The financial time-series used are taken from `https://finance.yahoo.com` and the exact data used can be downloaded from `https://github.com/SDMourtas/DATA/tree/main/TV-MVPS`.

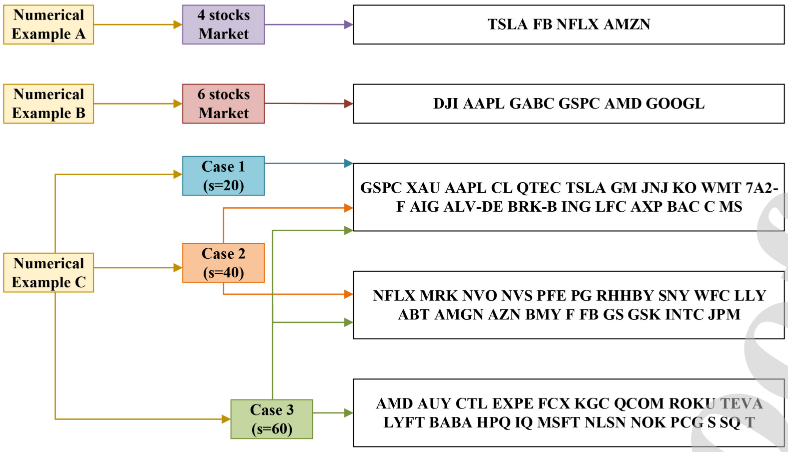


Figure 2: The stocks that were used in each of the three numerical examples.

5.1. Numerical Example A

Fig. 2 includes the ticker symbols of the stocks that we use in our portfolio in this example. Let $X = [x_1, x_2, x_3, x_4]$, where X comprises the daily close prices of the 4 Market stocks of Fig. 2 from 27/9/2018 to 10/12/2019 into x_1, x_2, x_3 and x_4, respectively. For the aforementioned time series, we use the first 50 prices of the observations to calculate the expected return matrix X_r and covariance structure X_C of Alg. 1. Consequently, we set $\tau = 50$. The rest of our data is the period from 10/12/2018 to 10/12/2019 with 253 observations. We divide the remaining data into ten periods of equal number of observations and we construct a ten period time-varying mean-variance portfolio selection. For the ten periods we have $tspan = [0\ 10]$ and, because each time-series comprises from 253 observations, we set $noep(1:10) = 253$ as input in function `omega`. Thus, we get $\omega = 25.3$, constant for all the range of $tspan$. Also, we use linear data interpolation in order to convert X_r and X_C into the functions $f_r(t)$ and $f_C(t)$, respectively.

In this example, we are going to examine two selections of portfolios. In the first selection, we set $r_p = \max(0.87 + 0.004t, \text{mean}(f_r(\omega(t))))$ and we use the LVI-PDNN setup of subsection 3.1. In the second selection, we set $r_p \geq \max(0.87 + 0.004t, \text{mean}(f_r(\omega(t))))$ and we use the LVI-PDNN setup of subsection 3.2. We set $\gamma = 1e10$, $f_r(\omega t)$, $f_C(\omega t)$ and solve the $\dot{y}(t)$ (see (16)) through MATLAB's `ode15s` with $y(0) = \text{rand}(6, 1)$.

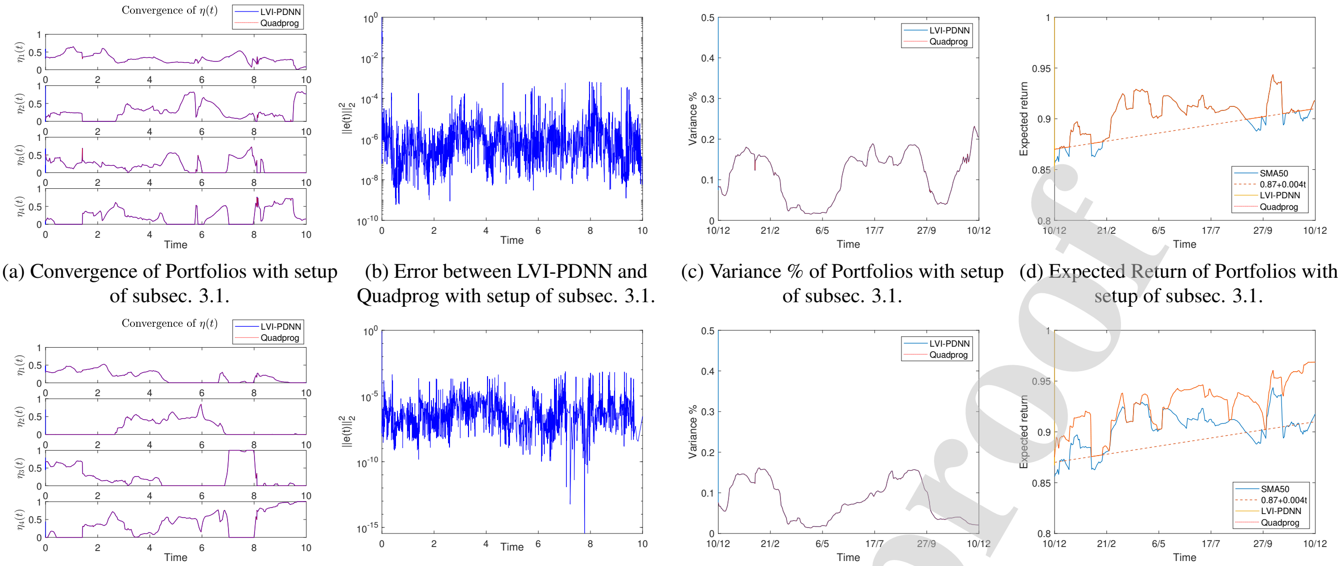
We present the results of the first selection in Figs. 3a-3d and the results of the second in Figs. 3e-3h where:

- Figs. 3a and 3e show the outcome $\eta(t)$ of LVI-PDNN and the outcome of `quadprog` for a specific target expected return and for all expected returns above a specific target, respectively,
- Figs. 3b and 3f show the error $\|e(t)\|_2^2$ between the outcome $\eta(t)$ of LVI-PDNN and the outcome of `quadprog` for a specific target expected return and for all expected returns above a specific target, respectively.
- Figs. 3c and 3g show the variance % of the portfolio $\eta(t)$ compared with the outcome of `quadprog` for a specific target expected return and for all expected returns above a specific target, respectively.
- Figs. 3d and 3h show the expected return of the portfolio $\eta(t)$, which is $\eta(t) f_r(\omega t)$, compared with the outcome of `quadprog`, the simple moving average $SMA50$ of $X(t)$, which is $\text{mean}(f_r(\omega t))$, and the function $0.87 + 0.004t$ for a specific target expected return and for all expected returns above a specific target, respectively.

The results that are depicted in Figs. 3a and 3e show that the LVI-PDNN solves the TV-MVPS problems and produces their online solution, $\eta(t)$. The solutions of the LVI-PDNN is similar to the solution of the MATLAB function `quadprog`, which is the assumed theoretical solution, and the error $\|e(t)\|_2^2$ between them are depicted in Figs. 3d and 3h, respectively. Also, the noise in Figs. 3d and 3h is expected because we are dealing with time-series. The variance of the portfolios $\eta(t)$ is shown in Figs. 3b and 3f and their expected return are shown in Figs. 3c and 3g, respectively. We observe that when we set a specific target expected return the variance of the portfolio is overall greater than if we had set as target all the expected returns above a specific target. Note that, as the value of parameter γ increases, the performance of the LVI-PDNN model improves and approaches the solution of `quadprog` even more. The time consumption of this numerical example is presented in Tab. 1 and shows that the LVI-PDNN method is on average almost two times faster as compared to the `quadprog` function. Overall, the LVI-PDNN worked excellently in solving the two TV-MVPS problems.

5.2. Numerical Example B

Fig. 2 includes the ticker symbols of the stocks that we use in our portfolio. Let $X = [x_1, x_2, x_3, x_4, x_5, x_6]$, where X comprises the daily close prices of the 6 Market stocks of Fig. 2 from 19/3/2013 to 2/1/2020 into $x_1, x_2, \ldots, x_6$, respectively. For the aforementioned time series, we use the first 200 prices of the observations to calculate the expected return matrix X_r and covariance structure X_C of Alg. 1. Consequently, we set $\tau = 200$. The rest of our data is the period from 2/1/2019 to 2/1/2020 with 1511 observations. In particular, the years



(a) Convergence of Portfolios with setup of subsec. 3.1.
(b) Error between LVI-PDNN and Quadprog with setup of subsec. 3.1.
(c) Variance % of Portfolios with setup of subsec. 3.1.
(d) Expected Return of Portfolios with setup of subsec. 3.1.
(e) Convergence of Portfolios with setup of subsec. 3.2.
(f) Error between LVI-PDNN and Quadprog with setup of subsec. 3.2.
(g) Variance % of Portfolios with setup of subsec. 3.2.
(h) Expected Return of Portfolios with setup of subsec. 3.2.

Figure 3: The convergence, the recorded error, the variance % and the expected return for a portfolio consisting of 4 stocks, in numerical example A.

2014, 2015, 2016 have 252 observations each, the years 2017, 2018 have 251 observations each and 2019, 2020 have 253 observation together. So, we have $tspan = [0\ 6]$ and by setting $noep = [252, 252, 252, 251, 251, 253]$ as input in function `omega`, we get

$$\omega = \begin{cases} 252 & , t \in [0, 3) \\ (252 \cdot 3 + 251 \cdot (t-3))/t & , t \in [3, 5) \\ (222 \cdot 3 + 251 \cdot 2 + 253 \cdot (t-5))/t & , t \in [5, 6] \end{cases}$$

where the 1511 observations have been divided in terms of the year which they belong. Also, we use P.C.Hermite data interpolation in order to convert X_r and X_C into the functions $f_r(t)$ and $f_C(t)$, respectively.

In this example, we examine two selections of portfolios. In the first selection, we set $r_p = \max(0.83 + 0.0066t, \text{mean}(f_r(\omega(t))))$ and we use the LVI-PDNN setup of subsection 3.1. In the second selection, we set $r_p = \min(f_r(\omega(t)))$ and we use the LVI-PDNN setup of subsection 3.2. We set $\gamma = 1e10$, $f_r(\omega t)$, $f_C(\omega t)$ and solve the $\dot{y}(t)$ (see (16)) through MATLAB's `ode15s` with $y(0) = \text{rand}(8, 1)$.

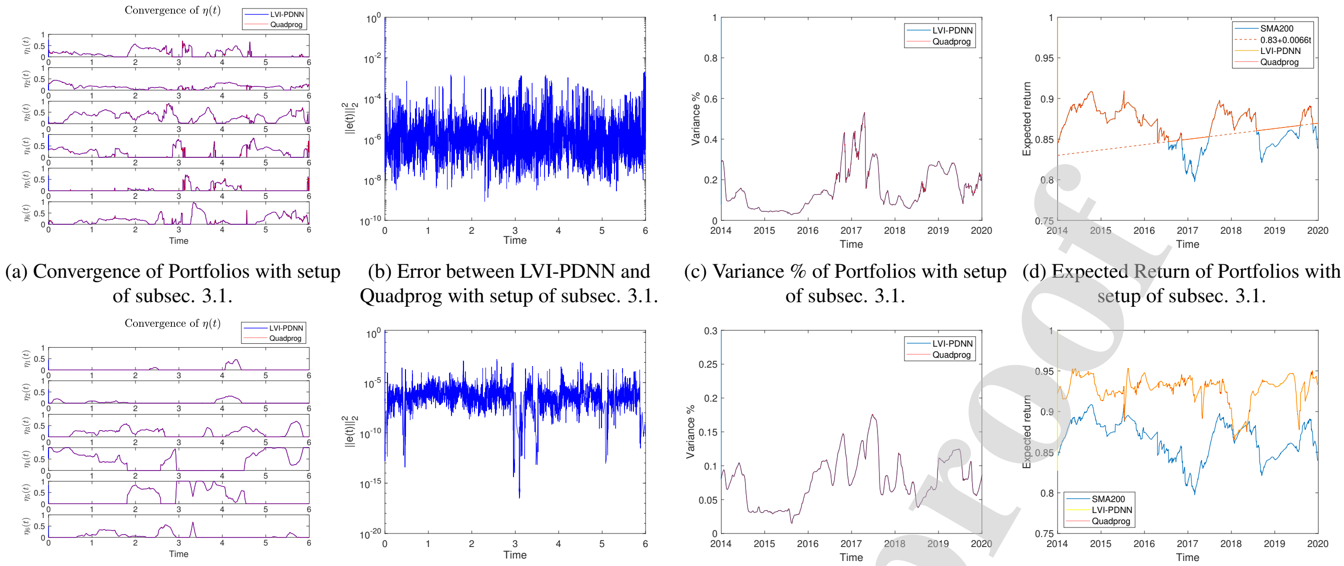
We present the results of the first selection in Figs. 4a-4d and the results of the second in Figs. 4e-4h where:

- Figs. 4a and 4e show the outcome $\eta(t)$ of LVI-PDNN and the outcome of `quadprog` for a specific target expected return and for all expected returns, respectively,
- Figs. 4b and 4f show the error $\|e(t)\|_2^2$ between the outcome $\eta(t)$ of LVI-PDNN and the outcome of `quadprog` for a specific target expected return and for all expected returns, respectively.
- Figs. 4c and 4g show the variance % of the portfolio $\eta(t)$ compared with the outcome of `quadprog` for a specific target expected return and for all expected returns, respectively.
- Figs. 4d and 4h show the expected return of the portfolio $\eta(t)$, which is $\eta(t) f_r(\omega t)$, compared with the outcome of `quadprog` and the simple moving average $SMA200$ of $X(t)$, which is $\text{mean}(f_r(\omega t))$, for a specific target expected return and for all expected returns, respectively. Also, Fig. 4d shows the function $0.83 + 0.0066t$.

The results that are depicted in Figs. 4a and 4e show that the LVI-PDNN solves the TV-MVPS problem and produces their online solution, $\eta(t)$. The solutions of the LVI-PDNN is similar to the solution of the MATLAB function `quadprog`, which is the assumed theoretical solution, and the error $\|e(t)\|_2^2$ between them are depicted in Figs. 4d and 4h, respectively. Also, the noise in Figs. 4d and 4h is expected because we are dealing with time-series. The variance of the portfolios $\eta(t)$ is shown in Figs. 4b and 4f and their expected return are shown in Figs. 4c and 4g, respectively. We observe that when we set a specific target expected return the variance of the portfolio is overall greater than if we had set as target all expected return. Note that, as the value of parameter γ increases, the performance of the LVI-PDNN model improves and approaches the solution of `quadprog` even more. The time consumption of this numerical example is presented in Tab. 1 and shows that the LVI-PDNN method is on average almost two times faster as compared to the `quadprog` function. Overall, the LVI-PDNN worked excellently in solving the two TV-MVPS problems.

5.3. Numerical Example C

This example covers three different portfolio configuration cases with a larger size to prove the reliability of the LVI-PDNN



(a) Convergence of Portfolios with setup of subsec. 3.1.
(b) Error between LVI-PDNN and Quadprog with setup of subsec. 3.1.
(c) Variance % of Portfolios with setup of subsec. 3.1.
(d) Expected Return of Portfolios with setup of subsec. 3.1.
(e) Convergence of Portfolios with setup of subsec. 3.2.
(f) Error between LVI-PDNN and Quadprog with setup of subsec. 3.2.
(g) Variance % of Portfolios with setup of subsec. 3.2.
(h) Expected Return of Portfolios with setup of subsec. 3.2.

Figure 4: The convergence, the recorded error, the variance % and the expected return for a portfolio consisting of 6 stocks, in numerical example B.

method on real-world datasets and demonstrate its efficacy in practical scenarios, even for large data sets. In the ith case, $i = 1, 2, 3$, we consider $X = [x_1, x_2, \dots, x_s]$, where X contains the daily close prices of the s stocks located in Fig. 2 from 2/4/2019 to 1/10/2019 into $x_1, x_2, \dots, x_s$, respectively. For the aforementioned time series, we use the first 20 prices of the observations to calculate the expected return matrix X_r and covariance structure X_C of Alg. 1. Consequently, we set $\tau = 20$. The rest of our data is the period from 1/5/2019 to 1/10/2019 with 107 observations. In particular, May, July, August have 22 observations each, June has 20 observations, September and October have 21 observations together. So, we have $tspan = [0\ 5]$ and by setting $noep = [22, 20, 22, 22, 21]$ as input in the function omega, we get

$$\omega = \begin{cases} 22 & , t \in [0, 1) \\ (22 \cdot 1 + 20 \cdot (t - 1))/t & , t \in [1, 2) \\ (22 \cdot 1 + 20 \cdot 1 + 22 \cdot (t - 2))/t & , t \in [2, 4) \\ (22 \cdot 3 + 20 \cdot 1 + 21 \cdot (t - 4))/t & , t \in [4, 5] \end{cases}$$

where the 107 observations have been divided in terms of the month which they belong. Also, we use the linear data interpolation in order to convert X_r and X_C into the functions $f_r(t)$ and $f_C(t)$, respectively.

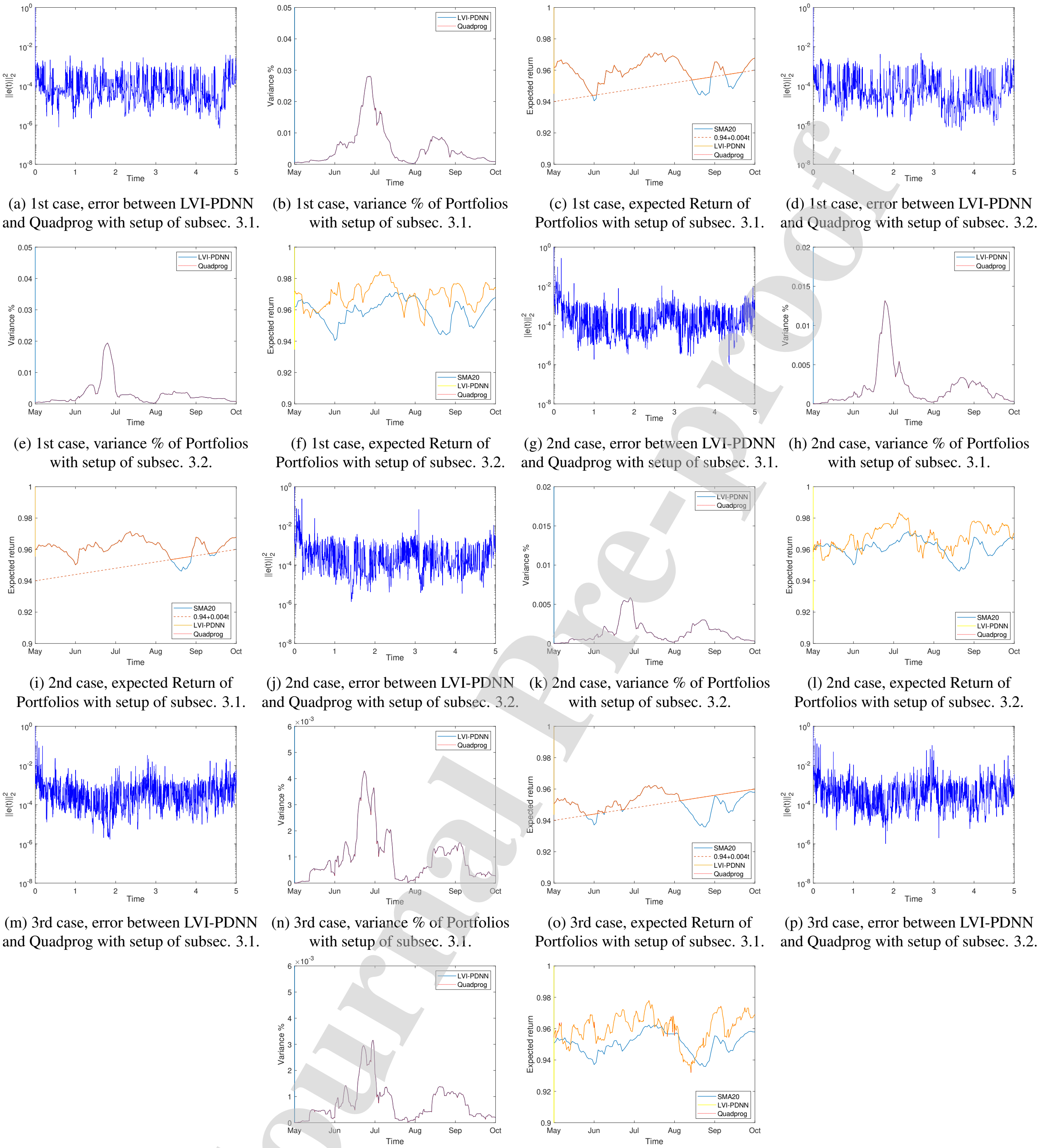
For each case, we examine two selections of portfolios. In the first selection, we set $r_p = \max(0.94 + 0.004t, \text{mean}(f_r(\omega(t))))$ and use the LVI-PDNN setup of subsection 3.1. In the second selection, we set $r_p = \min(f_r(\omega(t)))$ and use the LVI-PDNN setup of subsection 3.2. We set $\gamma = 1e7$, $f_r(\omega t)$, $f_C(\omega t)$ and solve the $\dot{y}(t)$ (see (16)) through MATLAB's `ode15s` with $y(0) = \text{rand}(s + 2, 1)$.

5.3.1. Comparative Results and Discussion

The results from the numerical example 5.3 can be summarized as follows:

- Tab. 1 shows the average execution time of LVI-PDNN and `quadprog` for each portfolio case in numerical example 5.3, by using step function, linear and P.C.Hermite data interpolation,
- for the portfolios consisting of 20 stocks (1st case), Figs. 5a-5c and Figs. 5d-5f show the error $||e(t)||_2^2$, between the outcome $\eta(t)$ of LVI-PDNN and the outcome of `quadprog`, the variance % and the expected return of the portfolio $\eta(t)$, for the selection of a specific target expected return and for the selection of all expected returns TV-MVPS, respectively,
- for the portfolios consisting of 40 stocks (2nd case), Figs. 5g-5i and Figs. 5j-5l show the error $||e(t)||_2^2$, between the outcome $\eta(t)$ of LVI-PDNN and the outcome of `quadprog`, the variance % and the expected return of the portfolio $\eta(t)$, for the selection of a specific target expected return and for the selection of all expected returns TV-MVPS, respectively,
- for the portfolios consisting of 60 stocks (3rd case), Figs. 5m-5o and Figs. 5p-5r show the error $||e(t)||_2^2$, between the outcome $\eta(t)$ of LVI-PDNN and the outcome of `quadprog`, the variance % and the expected return of the portfolio $\eta(t)$, for the selection of a specific target expected return and for the selection of all expected returns TV-MVPS, respectively.

The solution to the LVI-PDNN is similar to the solution of the MATLAB function `quadprog`, which is the assumed theoretical solution, and the error $||e(t)||_2^2$ between them is depicted in Figs. 5a, 5d, 5g, 5j, 5m and 5p. Also, the noise in these Figs. is expected because of the time series in the input. The



(a) 1st case, error between LVI-PDNN and Quadprog with setup of subsec. 3.1.

(b) 1st case, variance % of Portfolios with setup of subsec. 3.1.

(c) 1st case, expected Return of Portfolios with setup of subsec. 3.1.

(d) 1st case, error between LVI-PDNN and Quadprog with setup of subsec. 3.2.

(e) 1st case, variance % of Portfolios with setup of subsec. 3.2.

(f) 1st case, expected Return of Portfolios with setup of subsec. 3.2.

(g) 2nd case, error between LVI-PDNN and Quadprog with setup of subsec. 3.1.

(h) 2nd case, variance % of Portfolios with setup of subsec. 3.1.

(i) 2nd case, expected Return of Portfolios with setup of subsec. 3.1.

(j) 2nd case, error between LVI-PDNN and Quadprog with setup of subsec. 3.2.

(k) 2nd case, variance % of Portfolios with setup of subsec. 3.2.

(l) 2nd case, expected Return of Portfolios with setup of subsec. 3.2.

(m) 3rd case, error between LVI-PDNN and Quadprog with setup of subsec. 3.1.

(n) 3rd case, variance % of Portfolios with setup of subsec. 3.1.

(o) 3rd case, expected Return of Portfolios with setup of subsec. 3.1.

(p) 3rd case, error between LVI-PDNN and Quadprog with setup of subsec. 3.2.

(q) 3rd case, variance % of Portfolios with setup of subsec. 3.2.

(r) 3rd case, expected Return of Portfolios with setup of subsec. 3.2.

Figure 5: The recorded error, the variance % and the expected return for a portfolio consisting of 20, 40 and 60 stocks, in numerical example C.

variance % of the portfolios $\eta(t)$ for a specific target expected return are shown in Figs. 5b, 5h and 5n and the variance % of the portfolios $\eta(t)$ for all expected returns are shown in Figs. 5e, 5k and 5q. The expected return of the portfolios $\eta(t)$, which is $\eta(t)f_r(\omega t)$, compared with the outcome of `quadprog` and the simple moving average $SMA20$ of $X(t)$, which is mean$(f_r(\omega t))$, for a specific target expected return are shown in Figs. 5c, 5i and 5o and the expected return of the portfolios $\eta(t)$ for all expected returns are shown in Figs. 5f, 5l and 5r. Also, the Figs. 5c, 5i and 5o show the function $0.94 + 0.004t$. By considering the ω parameter, which is very helpful in the case where we want to combine different time periods with a different number of observations in each one of them, our approach is more realistic. Another major finding is that, in all the tested cases, the variance of the portfolios for a specific target expected return are significantly higher than the variance of the portfolios for all expected returns. The performance of LVI-PDNN and `quadprog` in numerical example 5.3 is shown in Tab. 1. It is obvious that the LVI-PDNN performance depends on the portfolio dimension and on the interpolation method. When the portfolio comprises from 20 stocks the LVI-PDNN produces faster result than `quadprog` in all cases that we tried. When the portfolio comprises from 40 stocks the LVI-PDNN produces slower result than `quadprog` only in the case of P.C.Hermite interpolation method. When the portfolio comprises from 60 stocks the LVI-PDNN produces faster result than `quadprog` only in the case of linear interpolation and only in the case of LVI-PDNN setup of subsection 3.2. Consequently, we conclude that as the dimension of portfolio rising the performance of LVI-PDNN weakens in comparison with `quadprog` MATLAB function. Overall, the portfolio cases presented in numerical example 5.3 show that the LVI-PDNN worked excellently in solving time-varying mean-variance portfolio selection problems.

5.4. *Time Comparison of LVI-PDNN and Quadprog*

We record the performance of LVI-PDNN with the proposed MATLAB functions in Alg. 2 and [28, 29] against the assumed theoretical solutions produced by the `quadprog` MATLAB function. The performance of LVI-PDNN is presented in Tab. 1. Tab. 1 shows the average execution time of numerical examples 5.1, 5.2 and 5.3 by using step functions, linear interpolation functions and P.C.Hermite interpolation functions. We also monitor the performance of LVI-PDNN with the corresponding MATLAB functions of MathWorks (namely `ts2func`, `interp1`) in Tab. 1. Furthermore, the notation (*) in Tab. 1 denotes that the specific time corresponds to $\gamma = 1e8$ instead of $\gamma = 1e10$. All numerical experiments are performed using the MATLAB R2018b environment on an Intel® Core™ i5-6600K CPU 3.50 GHz, 16 GB RAM, running on Windows 10 64 bit Operating System.

The general conclusion arising from Tab. 1 is that the step function of time series is the least efficient method and that the linear interpolation is the most efficient. In addition, from Tab. 1, we conclude that the proposed MATLAB functions, which manipulate matrices and structures time-series, are the best alternatives in terms of computation time responses, while they produce the same results. In the cases of linear and P.C.Hermite interpolation, only when we apply the proposed custom MATLAB functions, the LVI-PDNN produce faster results than the `quadprog` function.

Table 1: Examples 5.1, 5.2 and 5.3 execution time.

Interpolation Function	**Example A**			
	4 Stocks Portfolio			
	Setup 3.1		Setup 3.2	
	LVI-PDNN	Quadprog	LVI-PDNN	Quadprog
sfots & sfotss	1.6s	5s	1.6s	4.7s
ts2func	2s	5.3s	2s	4.8s
linots & linotss	1.8s	3.6s	8.5s	16s
interp1 (linear)	10.5s	5.9s	46s	22s
pchinots & pchinotss	2.3s	3.9s	9s	13s
interp1 (P.C.Hermite)	27s	9.8s	128s	39.5s
	Example B			
	6 Stocks Portfolio			
	Setup 3.1		Setup 3.2	
	LVI-PDNN	Quadprog	LVI-PDNN	Quadprog
sfots & sfotss	33.6s*	107.7s*	27.3s*	80.9s*
ts2func	55.8s*	118.4s*	41.6s*	85.1s*
linots & linotss	8.9s	17.2s	5.7s	10.6s
interp1 (linear)	86.5s	33.1s	73.5s	28.8s
pchinots & pchinotss	11.7s	18.4s	9.1s	13.4s
interp1 (P.C.Hermite)	398.2s	99s	342.8s	76.5s
	Example C			
	20 Stocks Portfolio			
	Setup 3.1		Setup 3.2	
	LVI-PDNN	Quadprog	LVI-PDNN	Quadprog
sfots & sfotss	6s	15s	4.5s	11.5s
linots & linotss	4.5s	8.5s	3s	6s
pchinots & pchinotss	8.5s	10s	5s	6s
	40 Stocks Portfolio			
	Setup 3.1		Setup 3.2	
	LVI-PDNN	Quadprog	LVI-PDNN	Quadprog
sfots & sfotss	16s	27s	10.5s	20s
linots & linotss	34s	36s	10s	13s
pchinots & pchinotss	75s	37s	35.5s	20.5s
	60 Stocks Portfolio			
	Setup 3.1		Setup 3.2	
	LVI-PDNN	Quadprog	LVI-PDNN	Quadprog
sfots & sfotss	1900s	1000s	1400s	820s
linots & linotss	51s	40s	37s	49s
pchinots & pchinotss	380s	248s	179s	63s

6. Conclusion

This paper introduces the TV-MVPS problem and presents its online solution. We take the LVI-PDNN from [21] to solve the time-varying QP financial problem in real time, subject to equality, inequality and boundary constraints. The efficiency of the LVI-PDNN model in such a time-varying financial QP problem has been demonstrated by a number of numerical examples. Conforming to our numerical simulations, we deduced that with the LVI-PDNN, our approach provides the online solution of a time-varying version of the mean-variance portfolio selection problem. It is also a highly competitive, or even better alternative to the `quadprog` MATLAB function. Nonetheless, as the value of the γ parameter increases, the performance of the LVI-PDNN model improves, and more accurately approaches the predicted theoretical solution. Experimental results show the reliability of the LVI-PDNN method on the real-world datasets in different portfolios setup, and demonstrate its usefulness for normal size data sets in realistic scenarios.

Acknowledgements. Predrag Stanimirović gratefully acknowledges support from the Ministry of Education, Science and Technological Development, Republic of Serbia, Grant No. 174013.

References

[1] J.-R. Yu, W.-J. P. Chiou, W.-Y. Lee, S.-J. Lin, Portfolio models with return forecasting and transaction costs, International Review of Economics & Finance 66 (2020) 118–130. doi:10.1016/j.iref.2019.11.002.

[2] V. N. Katsikis, S. D. Mourtas, A heuristic process on the existence of positive bases with applications to minimum-cost portfolio insurance in C[a, b], Applied Mathematics and Computation 349 (2019) 221–244. doi:10.1016/j.amc.2018.12.044.

[3] V. N. Katsikis, S. D. Mourtas, ORPIT: A Matlab Toolbox for Option Replication and Portfolio Insurance in Incomplete Markets, Computational Economics (2019) 1doi:10.1007/s10614-019-09936-5. URL http://dx.doi.org/10.1007/s10614-019-09936-5

[4] H. Holden, L. Holden, Optimal rebalancing of portfolios with transaction costs, Stochastics An International Journal of Probability and Stochastic Processes 85 (3) (2012) 371–394. doi:10.1080/17442508.2011.651219.

[5] Y. Zhang, J. Wang, Y. Xia, A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits, IEEE Transactions on Neural Networks 14 (3) (2003) 658–667. doi:10.1109/TNN.2003.810607.

[6] Yunong Zhang, Jun Wang, Obstacle avoidance for kinematically redundant manipulators using a dual neural network, Part B (Cybernetics) IEEE Transactions on Systems, Man, and Cybernetics 34 (1) (2004) 752–759. doi:10.1109/TSMCB.2003.811519.

[7] L. Jin, Y. Zhang, S. Li, Y. Zhang, Modified ZNN for time-varying quadratic programming with inherent tolerance to noises and its application to kinematic redundancy resolution of robot manipulators, IEEE Transactions on Industrial Electronics 63 (11) (2016) 6978–6988. doi:10.1109/TIE.2016.2590379.

[8] L. Yu, S. Wang, K. K. Lai, Neural network-based mean–variance–skewness model for portfolio selection, Computers & Operations Research 35 (1) (2008) 34–46.

[9] A. Fernández, S. Gómez, Portfolio selection using neural networks, Computers & Operations Research 34 (4) (2007) 1177–1191.

[10] M. A. Medvedeva, V. N. Katsikis, S. D. Mourtas, T. E. Simos, Randomized time-varying knapsack problems via binary beetle antennae search algorithm: Emphasis on applications in portfolio insurance, Math Meth Appl Sci (2020) 1–11doi:https://doi.org/10.1002/mma.6904.

[11] V. N. Katsikis, S. D. Mourtas, Optimal portfolio insurance under nonlinear transaction costs, Journal of Modeling and Optimization 12 (2) (2020) 117–124.

[12] F. Uhlig, Y. Zhang, Time-varying matrix eigenanalyses via zhang neural networks and look-ahead finite difference equations, Linear Algebra and its Applications. (2019).

[13] H. Markowitz, Portfolio selection, The journal of finance 7 (1) (1952) 77–91.

[14] I. Kulali, Portfolio Optimization Analysis with Markowitz Quadratic Mean-Variance Model, European Journal of Business and Management 8 (7) (2016) 73–79.

[15] Z. Dai, A Closer Look at the Minimum-Variance Portfolio Optimization Model, Mathematical Problems in Engineering 2019 (2019) 1–8.

[16] T. R. Bielecki, H. Jin, S. R. Pliska, X. Y. Zhou, Continuous-time mean-variance portfolio selection with bankruptcy prohibition, Mathematical Finance 15 (2) (2005) 213–244. doi:10.1111/j.0960-1627.2005.00218.x.

[17] G. Cornuejols, R. Tütüncü, Optimization Methods in Finance, Cambridge: Cambridge University Press, 2006. doi:10.1017/CBO9780511753886.

[18] T. Draviam, T. Chellathurai, Generalized Markowitz mean–variance principles for multi–period portfolio–selection problems, Proc. R. Soc. Lond. A. 458 (2002) 2571–2607. doi:10.1098/rspa.2002.0983.

[19] H. M. Markowitz, The general mean-variance portfolio selection problem, Philosophical Transactions of the Royal Society of London 347 (1994) 543–549. doi:10.1098/rsta.1994.0063.

[20] V. Zakamulin, Market Timing with Moving Averages: The Anatomy and Performance of Trading Rules, Springer, 2017.

[21] Y. Zhang, F. Wu, Z. Xiao, Z. Li, B. Cai, Performance analysis of LVI-based PDNN applied to real-time solution of time-varying quadratic programming, in: 2014 International Joint Conference on Neural Networks, IJCNN 2014, Beijing, China, July 6-11, 2014, IEEE, 2014, pp. 3155–3160. doi:10.1109/IJCNN.2014.6889453.

[22] H. M. Markowitz, Portfolio Selection: Efficient Diversification of Investments, Cowles Foundation Monograph: No. 16, Yale University Press, 1959.

[23] S. Boyd, L. Vandenberghe, Convex optimization problems, Cambridge University Press, New York, 2004. doi:10.1017/cbo9780511804441.005.

[24] Y. Zhang, C. Yi, Zhang neural networks and neural-dynamic method, Nova Science Publishers, Inc., 2011.

[25] Z. Zhang, Y. Zhang, Acceleration-Level Cyclic-Motion Generation of Constrained Redundant Robots Tracking Different Paths, Part B (Cybernetics) IEEE Transactions on Systems, Man, and Cybernetics 42 (4) (2012) 1257–1269. doi:10.1109/TSMCB.2012.2189003.

[26] Y. Zhang, Towards Piecewise-Linear Primal Neural Networks for Optimization and Redundant Robotics, in: Proc. Sensing and Control 2006 IEEE Int. Conf. Networking, 2006, pp. 374–379. doi:10.1109/ICNSC.2006.1673175.

[27] Y. Zhang, On the LVI-based primal-dual neural network for solving online linear and quadratic programming problems, in: Proc. 2005, American Control Conf, 2005, pp. 1351–1356 vol. 2. doi:10.1109/ACC.2005.1470152.

[28] V. N. Katsikis, S. D. Mourtas, P. S. Stanimirović, S. Li, X. Cao, Time-Varying Mean-Variance Portfolio Selection under Transaction Costs and Cardinality Constraint Problem via Beetle Antennae Search Algorithm (BAS), SN Operations Research Forum 2 (18) (2021). doi:https://doi.org/10.1007/s43069-021-00060-5.

[29] V. N. Katsikis, S. D. Mourtas, P. S. Stanimirović, S. Li, X. Cao, Time-varying minimum-cost portfolio insurance under transaction costs problem via Beetle Antennae Search algorithm (BAS), Applied Mathematics and Computation 385 (2020) 125453.

- Definition and study of the time-varying mean-variance portfolio selection (TV-MVPS) problem.
- Online solution of the TV-MVPS problem via a Linear-Variational-Inequality Primal-Dual Neural Network (LVI-PDNN).
- The time-varying mean-variance portfolio selection model eliminates the drawbacks of the static strategy, resulting in more practical results.

Vasilios N. Katsikis: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - Original Draft, Writing - Review & Editing

Spyridon D. Mourtas: Methodology, Software, Validation, Formal analysis, Investigation, Writing - Original Draft, Writing - Review & Editing

Predrag S. Stanimirovic: Methodology, Formal analysis, Investigation, Writing - Original Draft, Writing - Review & Editing

Shuai Li: Methodology, Investigation, Writing - Original Draft

Xinwei Cao: Methodology, Investigation, Writing - Original Draft