Applied Mathematics and Computation 441 (2023) 127700

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

Time-varying minimum-cost portfolio insurance problem via an adaptive fuzzy-power LVI-PDNN

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

[a] Department of Economics, Division of Mathematics-Informatics and Statistics-Econometrics, National and Kapodistrian University of Athens, Sofokleous 1 Street, Athens 10559, Greece
[b] Laboratory "Hybrid Methods of Modelling and Optimization in Complex Systems", Siberian Federal University, Prosp. Svobodny 79, Krasnoyarsk 660041, Russia
[c] University of Niš, Faculty of Sciences and Mathematics, Višegradska 33, Niš 18000, Serbia
[d] Swansea University, Swansea, UK
[e] School of Management, Shanghai University, China

ARTICLE INFO

Article history:
Received 23 September 2022
Revised 14 October 2022
Accepted 4 November 2022

Keywords:
Neural networks
Fuzzy logic system
Portfolio insurance
Time-varying linear programming
Portfolio optimization

ABSTRACT

It is well known that minimum-cost portfolio insurance (MPI) is an essential investment strategy. This article presents a time-varying version of the original static MPI problem, which is thus more realistic. Then, to solve it efficiently, we propose a powerful recurrent neural network called the linear-variational-inequality primal-dual neural network (LVI-PDNN). By doing so, we overcome the drawbacks of the static approach and propose an online solution. In order to improve the performance of the standard LVI-PDNN model, an adaptive fuzzy-power LVI-PDNN (F-LVI-PDNN) model is also introduced and studied. This model combines the fuzzy control technique with LVI-PDNN. Numerical experiments and computer simulations confirm the F-LVI-PDNN model's superiority over the LVI-PDNN model and show that our approach is a splendid option to accustomed MATLAB procedures.

1. Introduction

In economic decisions, optimization models play a significant role. Popular areas include asset allocation, option pricing, risk management, model calibration etc. Considering that research on metaheuristic algorithms has recently been widely used to address both linear and nonlinear optimization problems [1,2], such optimization models can be approached effectively employing accustomed methods of optimization. For instance, optimal tangency portfolio under cardinality constraint problems are defined in [3,4] as nonlinear programming problems and approached by heuristics. In [5,6], under large data inputs, time-varying Markowitz-based portfolio optimization problems are formulated and studied using nature-inspired optimization algorithms. A nonlinear portfolio optimization problem is presented in [7], for minimizing the cost of insurance along with the portfolio's transaction costs and it is approached using a memetic metaheuristic algorithm, called beetle an-

* Corresponding author.
E-mail addresses: vaskatsikis@econ.uoa.gr (V.N. Katsikis), spirosmourtas@gmail.com (S.D. Mourtas), pecko@pmf.ni.ac.rs (P.S. Stanimirović), shuaili@ieee.org (S. Li), xinweicao@shu.edu.cn (X. Cao).

https://doi.org/10.1016/j.amc.2022.127700

tennae search. In this paper, the time-varying minimum-cost portfolio insurance (TMPI) problem is defined and studied as a time-varying linear programming (TVLP) problem.

Numerous significant theoretical research discoveries, along with hardware and software implementations of an artificial neural network (ANN) have been proposed with the exponential development of artificial intelligence, information technology, and contemporary electronics [8–14]. For example, ANN models have been applied to solar radiation prediction [8], portfolio optimization [10,11], stochastic exchange rates stabilization [12] and feedback control systems stabilization [9]. In general, the main benefits of an ANN contain generalization, noise-tolerance, fault-tolerance, and the capability to anticipate unknown data while reducing processing costs and time [13,14]. Time-varying and static (time-invariant) problems might necessitate alternative techniques due to the fact that usually they behave differently [14]. Keep in mind that the majority of existing techniques and tools are built with the goal of resolving static problems. Many real-time systems (such as real-time monitoring and controlling systems, and high-speed parallel processing systems) are time-varying due to the presence of time-dependent variables and the demand for high performance. As a result, conventional methods cannot be as efficient as dealing with static problems in this case.

Fuzzy logic systems (FLS), on the other hand, have been researched intensively, see for example [15,16]. FLSs have the ability to manage uncertainties, which is why they are used in a variety of fields, such as robotics [17], spacecraft [18], Rössler chaotic dynamical system [19] and automobile [20]. Other well-known FLS applications include nonlinear system control [21] and the study of time-varying systems [22], whereas numerous publications of current research focus on the use of fuzzy control in error-correction neural networks [23,24]. According to the application and characteristics of FLS, this study employs FLS to improve the performance of a recurrent neural network dubbed linear-variational-inequality primal-dual neural network (LVI-PDNN). For instance, a number of time-varying financial portfolio selection problems are proposed and investigated in [10,11,14] and it is found that utilizing ANN models rather than conventional methods reduces the amount of time needed to find the optimal solution. So, conventional methods cannot be as efficient as dealing with static problems in practical scenarios. As a result, an adaptive fuzzy-power LVI-PDNN (F-LVI-PDNN) model is introduced and studied.

In this paper, we study the TMPI problem as a continuous TVLP problem in order to trace the progress of the MPI when changing over time and to provide some type of forecast. More particularly, we use interpolation methods to convert the problem's data in continuous time, and then we apply the proposed models, LVI-PDNN and F-LVI-PDNN, to produce an online solution. Note that, the LVI-PDNN is a well-known approach for tackling a number of static problems and outperforms the approach for handling time-varying problems. By doing so, we overcome the drawbacks of the static approach and propose an online solution that is more realistic to a time-varying financial problem.

The following list summarizes this work's key points:

- A TVLP financial problem, dubbed TMPI, is introduced and investigated.
- A unique LVI-PDNN's design for addressing the TMPI problem is proposed.
- An adaptive F-LVI-PDNN model is introduced and studied.
- Applications of LVI-PDNN and F-LVI-PDNN on actual financial time-series from the real world.
- High-level performing MATLAB function `linprog` is compared with the proposed LVI-PDNN and F-LVI-PDNN models to evaluate their performances.

Since `linprog` is designed for time-invariant linear programming (LP), multiple static LP problems have been approached by the `linprog` to generate a part of the LVI-PDNN's solution on the TVLP problem.

The paper has been organized in the following way. Section 2 presents the TMPI problem, which is a financial TVLP problem, in detail. Portfolio insurance optimization through fuzzy LVI-PDNN is presented in Section 3. Section 4 includes applications on the TMPI problem, employing actual financial time-series from the real world. Section 5 gives concluding remarks.

2. Minimum-cost portfolio insurance

For financial models, reducing portfolio costs is always of major importance. One way to reduce portfolio costs is to minimize the cost of insurance (see [5,25–29]). For instance, an approach based on the notion of Riesz spaces is employed in [25] to solve a time-varying optimization problem for minimizing the insurance costs of the portfolio. In [27], the author solve the problem of optimal expected growth in a random trade time model, taking into account the insurance of the portfolio in a low liquid market, to ensure the optimal constant proportion portfolio insurance approach in a simple form. Inhere, we propose a time-varying equivalent to the relevant static problem that has been discussed and examined in numerous articles, such as [30,31].

The continuous time-varying version of the MPI problem introduced here is an innovative approach that integrates robust processes from neural networks to provide an online solution that is more realistic to a time-varying financial problem. With $C[a,b]$, we indicate the space of real-valued continuous functions specified at $[a,b]$ interval. The initial portfolio is a vector $\phi = [\phi_1, \phi_2, \ldots, \phi_n]^{\mathrm{T}}$ in $\mathbb{R}^n$ and ϕ_i, for each $i = 1, \ldots, n$, denote the investment on asset i. $\mathbb{R}^n$ is referred to as portfolio space. If ϕ is a portfolio that is not zero, then its payoff is given by the formula

$$X^{\mathrm{T}}(t) \cdot \phi,$$

where $X(t) = [x_1(t), x_2(t), \dots, x_n(t)]^{\mathrm{T}}$ is the space of invested assets, $x_i(t) \in C[a, b]$ indicate the i asset's return, for each i, and $t \in [a, b] \subseteq [0, +\infty)$ is the time.

Let us suppose that $p(t) = [p_1(t), p_2(t), \dots, p_n(t)]^{\mathrm{T}}$ is a vector of time-varying prices, where $p_i(t) \in C[a, b]$, $i = 1, 2, \dots, n$ is the insurance price of asset i. Here, we assume that a fixed price along with an asset risk cost make up the costs of insurance for the portfolio. If the price rates related to asset risk are represented by β and the fixed price by δ, the function of the fixed and linear insurance prices is as follows:

$$p_i(t) = \delta + \beta \cdot \mathrm{Var}\left[\frac{M_i(t)}{\max(M_i(t))}\right], \tag{2.1}$$

where $M_i(t) = [x_i(t), x_i(t-1), \dots, x_i(t-c+1)]^{\mathrm{T}}$ with $i = 1, 2, \dots, n$, and $\mathrm{Var}[\cdot]$ signifies the variance. Also, the constant $c \le t-1$, $c \in \mathbb{N}$, implies the time delays. Since $M_i(t) \in \mathbb{R}^c$, it is the variance of its c in number normalized prices that measures the risk of the i asset. Thus, the cost of insurance for any asset is determined by the level of risk it entails, when there is a possibility that the expected return may not match the real return. Because of this, insurance premiums are increasing in pace with inflation.

For a portfolio ϕ and floor price $\lambda \in \mathbb{R}$ the $\max\{X^{\mathrm{T}}(t) \cdot \phi, \lambda\}$, is the insured payoff of the portfolio at floor price λ. As a result, the TVLP formulation of the TMPI problem is the following:

$$\min_{\eta(t) \in \mathbb{R}^n} p^{\mathrm{T}}(t) \cdot \eta(t) \tag{2.2}$$

$$\text{subject to} - X^{\mathrm{T}}(t) \cdot \eta(t) \le \min\{-X^T(t) \cdot \phi, -\lambda\} \tag{2.3}$$

$$0 \le \eta(t) \le \theta(t), \tag{2.4}$$

where $\eta(t) = [\eta_1(t), \eta_2(t), \dots, \eta_n(t)]^{\mathrm{T}} \in \mathbb{R}^n$ is the optimal portfolio and $\theta(t) = X^{\mathrm{T}}(t) \cdot \phi \cdot \left[\frac{1}{x_1(t)}, \dots, \frac{1}{x_n(t)}\right]^{\mathrm{T}} = [\theta_1(t), \theta_2(t), \dots, \theta_n(t)]^{\mathrm{T}}$ with $\theta_i(t) \in C[a, b]$, $i = 1, 2, \dots, n$ is the upper bound of $\eta(t)$. More precisely, $\theta(t)$ is the higher value of any stock that an investor is eligible to hold at time t, when putting all the portfolio ϕ payoff in each one of them.

3. Portfolio insurance optimization through fuzzy LVI-PDNN

Given that LP plays a fundamental role in mathematical optimization, the majority of its facets have been extensively researched during the past few decades. Applications in both research and industry have frequently adopted solutions to LP problems [32–34]. In the past, researchers have often addressed LP problems with no more than two different types of constraints [33]. However, most studies were dedicated to investigating LP problems on the basis of static coefficients, see [35], which implies that approaches intended to deal with this class of LP problems are useless in time-varying contexts.

The neural network technique has been proved to be a powerful real-time computation instrument for over ten years due to the availability of hardware implementation and its parallel distributed computing nature [14,24,36–38]. For example, [36] presents a simple, piecewise-linear and global convergent approach to optimal solutions with dual neural networks.

In [38], the authors introduced a primal-dual neural network with a piecewise-linear dynamic, inline with linear variational inequality. In [37] one can find a LVI-PDNN with the ability to handle both LP and quadratic programming (QP) on the same/unified way. Globally, the LVI-PDNN converges to the optimum solution(s). In [39], LVI-PDNN was applied simultaneously to find real-time solutions to time-varying LP problems, liable to equality, inequality and boundary constraints.

3.1. TMPI through the LVI-PDNN design

To approach the TMPI problem we include the Eqs. (2.2)–(2.4) to the LVI-PDNN from Wu et al. [39]. Consequently, we set the coefficients:

$$H(t) = \begin{bmatrix} 0 & -X(t) \\ X^{\mathrm{T}}(t) & 0 \end{bmatrix}, \quad r(t) = \begin{bmatrix} p(t) \\ \min\{X^{\mathrm{T}}(t) \cdot \phi, -\lambda\} \end{bmatrix},$$

and the primal-dual decision vector $z(t)$, with upper and lower bounds to which it is liable, as below:

$$z(t) = \begin{bmatrix} \eta(t) \\ \upsilon(t) \end{bmatrix} \in \mathbb{R}^{n+1}, \quad \zeta^-(t) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{n+1}, \quad \zeta^+(t) = \begin{bmatrix} \theta(t) \\ 1e100 \end{bmatrix} \in \mathbb{R}^{n+1},$$

where $\upsilon(t) \in \mathbb{R}$ signifies the dual decision variable of (2.3). The TVLP problem of (2.2)–(2.4) can be addressed using the LVI-PDNN dynamical system described below:

$$\dot{z}(t) = \gamma(I + H^{\mathrm{T}}(t))(P_\Omega(z(t) - (H(t)z(t) + r(t))) - z(t)). \tag{3.1}$$

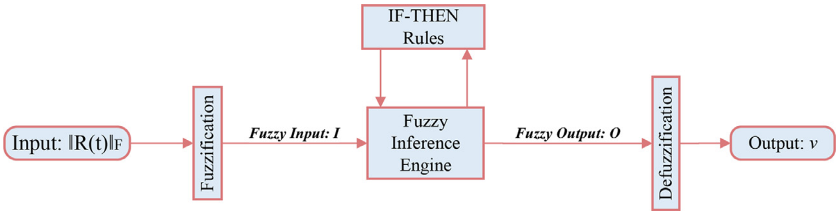


Fig. 1. The FLC structure.

Notice that $\gamma > 0$ is a design parameter, whereas $P_\Omega(\cdot)$ is the projection operator [39], defined as below:

$$P_\Omega(z_i(t)) = \begin{cases} \zeta^-(t), & z_i(t) < \zeta^-(t) \\ z_i(t), & \zeta^-(t) \le z_i(t) \le \zeta^+(t), \ i = 1, 2, \dots, n+1. \\ \zeta^+(t), & z_i(t) > \zeta^+(t) \end{cases} \tag{3.2}$$

It should be highlighted that the solution $z(t)$ of (3.1) can be successfully produced using a MATLAB `ode` solver.

To provide a clearer picture of the real-time LVI-PDNN's convergence, the residual error is given by:

$$R(t) = z(t) - P_\Omega(z(t) - (H(t)z(t) + r(t))). \tag{3.3}$$

According to [35, Theorems 1 and 2], if the Frobenius norm $\|R(t)\|_F \to 0$, the LVI-PDNN converges to the theoretical solution as a consequence that $z(t)$ converge to the theoretical solution $z^*(t)$.

3.2. The adaptive fuzzy-power LVI-PDNN design

Fuzzy logic systems have found extensive use in the control of nonlinear systems due to its potent capacity to approximate complicated systems without the requirement for prior knowledge of system dynamics. The LVI-PDNN model's constant design parameter γ is supplemented by a fuzzy parameter due to the difficulties in precisely describing the size of the error-norm $\|R(t)\|_F$ in the model (3.1). In this approach, the LVI-PDNN model's calculation accuracy and robustness can be increased, and the following fuzzy-power LVI-PDNN (F-LVI-PDNN) dynamics is suggested:

$$\dot{z}(t) = \gamma^\nu (I + H^{\mathrm{T}}(t))(P_\Omega(z(t) - (H(t)z(t) + r(t))) - z(t)), \tag{3.4}$$

where $\gamma > 1$ is the traditional design parameter and ν is the desired fuzzy parameter, acquired using a suitably designed fuzzy logic controller (FLC). Typically, designing an FLC requires three steps [24]. Considering the error of the LVI-PDNN, i.e., $\|R(t)\|_F$, to be the FLC's input, then Fig. 1 illustrates the initiated output ν.

The following is a description of the three steps involved in creating the FLC.

(1) *Fuzzification*: By using the following π-shaped membership function (MF), the fuzzification used in this research converts the input set into the fuzzy input set I and the output set into the fuzzy output set O:

$$\kappa(x) = \begin{cases} 0, & x \le \kappa \\ 2\left(\frac{x-\kappa}{\zeta-\kappa}\right)^2, & \kappa \le x \le \frac{\kappa+\zeta}{2} \\ 1 - 2\left(\frac{x-\zeta}{\zeta-\kappa}\right)^2, & \frac{\kappa+\zeta}{2} \le x \le \zeta \\ 1, & \zeta \le x \le \xi \\ 1 - 2\left(\frac{x-\xi}{\psi-\xi}\right)^2, & \xi \le x \le \frac{\xi+\psi}{2} \\ 2\left(\frac{x-\psi}{\psi-\xi}\right)^2, & \frac{\xi+\psi}{2} \le x \le \psi \\ 0, & \psi \le x, \end{cases} \tag{3.5}$$

where $\kappa, \zeta, \xi, \psi \in \mathbb{R}$ signify constant parameters. Particularly, the MF's feet is determined by the parameters κ, ψ, while its shoulders are determined by ζ, ξ. There are several possibilities available when choosing an MF in general. In order to soften the sharp bounds while maintaining the overall behavior of the $\|R(t)\|_F$ performance ranges, an experimentally determined π-shaped MF was employed to describe the degree of membership of the $\|R(t)\|_F$ values.

The following triangular MF is the function this FLC uses to generate the output set:

$$\rho(x) = \max\left\{\min\left\{\frac{x-\kappa}{\zeta-\kappa}, \frac{\xi-x}{\xi-\zeta}\right\}, 0\right\}, \tag{3.6}$$

where $\kappa, \zeta, \xi \in \mathbb{R}$ signify constant parameters. Particularly, the MF's feet is determined by the parameters κ, ξ, while its peak is determined by ζ.

(2) *Fuzzy inference engine*: The following group of "IF-THEN" rules are fuzzy rules between the fuzzy input set I and the fuzzy output set O:

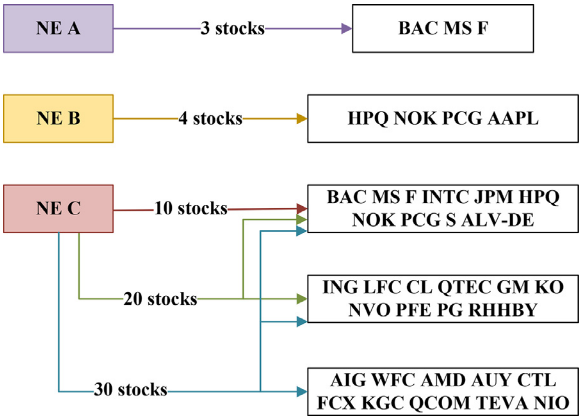


Fig. 2. Market stocks in NEs 4.1–4.3.

- P_1: if $I = L$ then $O = L$,
- P_2: if $I = H$ then $O = H$.

L and H, respectively, are fuzzy set notations that precisely represent low error and high error. Considering $P_l = P_1 \cup P_2$, it is acquired $O = I \circ P_l$, $l = 1, 2$, where $\circ$ signifies the fuzzy transformation symbol. Thereafter, $\alpha_{I \circ R}(\nu) = \alpha_{I \circ P_1} \vee \alpha_{I \circ P_2}$ and $\alpha_{I \circ P_l} = \sup(\alpha_{I_l} \wedge \alpha_{O_l})$ for $l = 1, 2$. It is worth mentioning that $\wedge$ and $\vee$ signify the minimum and maximum values operators, respectively.

(3) *Defuzzification*: To acquire the fuzzy parameter ν, the defuzzification method below, dubbed centroid, is utilized:

$$\nu = \frac{\int_O \nu \, \alpha_{I \circ R}(\nu) \mathrm{d}\nu}{\int_O \alpha_{I \circ R}(\nu) \mathrm{d}\nu}. \tag{3.7}$$

Utilizing the F-LVI-PDNN dynamics (3.4) improves the capacity for continuous learning and preserves knowledge from prior learning during the continual learning process. The key concept is to substitute the primary gain value γ with the power γ^{ν}, which contains the suitably calculated fuzzy exponent ν from the FLC. The fundamental purpose of the suggested FLC is to integrate the prior knowledge, which is based on $\|R(t)\|_F$ values, into the LVI-PDNN dynamics. It is important to note that the FLC controller structure, including the membership functions selected, the amount of fuzzy rules and the defuzzification method selected, was created particularly to address the TMPI problem. The FLC controller structure is therefore heuristic, and different structures are needed for different applications.

4. Numerical experiments

The data inputs for the financial optimization model we work with are time-series. In other words, the data input is in discrete-time. As we are attempting to calculate the online solution of a continuous time-varying problem, these data have to be converted into continuous-time. We accomplish that by transforming the time-series into functions of continuous-time. Particularly, the linear interpolation method is used in this section on five different portfolios. To accomplish this, the custom interpolation function `linots`, taken from [5,28], is employed on $X(t)$ and $p(t)$. It is important to note that [5,28] propose numerous custom interpolation functions of well-known interpolation methods.

The time-series that were utilized in the numerical experiments of this section are shown in Fig. 2. More precisely, Fig. 2 contains the ticker symbol of the stocks that are available in the market. Note that a ticker symbol is a string of letters which is employed to identify stocks, bonds, mutual funds or any other securities traded on the stock exchange. Also, the financial time-series utilized were retrieved from Yahoo Finance, whereas the specific data utilized may also be acquired from https://github.com/SDMourtas/DATA/tree/main/TMPI.

In addition, the time periods in finance may be divided into annually, quarterly, monthly, weekly, daily and combinations of them. But, between two different time periods of the same division their observations may not be equal in number, which is caused by the fact that financial markets can be close, one month has fewer days than another, the year is leap, etc. To solve the omitted observations problem, we employ the parameter ω that splits the observations into time periods for each t within the processes of the proposed models. That is, we use $p(\omega t)$ and $X(\omega t)$ in place of $p(t)$ and $X(t)$. For convenience, we employ the custom function `omega`, which is presented in [5,28]. This function takes as inputs the time period t along with the vector *noep*, which includes the total number of observations for each period, and returns the parameter ω.

The recommended settings for the FLC employed in all numerical examples (NEs) of this section are presented in Table 1, whereas the `ode15s` MATLAB solver is used on (3.1) and (3.4) to produce the solution of the TMPI problem. In the following NEs, the parameters settings are $\gamma = 100$, $\delta = 2$ and $\beta = 1000$. It is important to note that the MATLAB function `linprog` may only address static linear programming (LP) problems, not TVLP problems. As a consequence, multiple static

Table 1
FLC's recommended settings.

Set	MF	Range	Rule	κ	ζ	ξ	ψ	Weight
Input	π-shaped	[0,10]	P_1	-0.1	0	0	0.1	1
			P_2	0	0.01	10	12	1
Output	triangular	[1,7]	P_1	-5	1	2	-	1
			P_2	6	7	13	-	1

LP problems have been approached by the `linprog` to generate a part of the LVI-PDNN's solution on the TVLP problem. Furthermore, using the theoretical solutions produced by the `linprog` as a reference, our attention is solely on comparing the performances of LVI-PDNN and F-LVI-PDNN.

4.1. Numerical example A

In this NE, we deal with a portfolio containing the 3 stocks presented in Fig. 2. Let $X(t) = [x_1(t), x_2(t), x_3(t)]^{\mathrm{T}}$, where $X(t)$ contains the daily close prices of these 3 stocks into $x_1(t)$, $x_2(t)$ and $x_3(t)$, respectively. The time delay parameter has been set to $c = 40$, and we find the minimum-cost insured portfolio $\eta(t)$ for the time period 02/03/2020 to 01/10/2020. Furthermore, we divide our time-series into seven monthly periods by setting $noep = [23, 21, 20, 22, 22, 21, 22]$, and we set the interval of integration [0,7] in the `ode` solver. Note that the vector $noep$ contains the observations number for each month of the time period 02/03/2020 to 01/10/2020. Given a portfolio $\phi = [2, 2, 2]$, a floor $\lambda = 140$, and starting from $z(0) = [\phi, 2]$, we present the results in Fig. 3a–d.

4.2. Numerical example B

A 4 stock portfolio is built in this NE, which contains the 4 stocks presented in Fig. 2. Let $X(t) = [x_1(t), x_2(t), x_3(t), x_4(t)]^{\mathrm{T}}$, where $X(t)$ contains the daily close prices of these 4 stocks into $x_1(t), x_2(t), x_3(t), x_4(t)$, respectively. The time delay parameter has been set to $c = 35$, and we find the minimum-cost insured portfolio $\eta(t)$ for the time period 02/01/2020 to 03/08/2020. Furthermore, we divide our time-series into seven monthly periods by setting $noep = [22, 19, 22, 21, 20, 22, 23]$, and we set the interval of integration [0,7] in the `ode` solver. Note that the vector $noep$ contains the observations number for each month of the time period 02/01/2020 to 03/08/2020. Given a portfolio $\phi = [2, 2, 2, 2]$, a floor $\lambda = 220$, and starting from $z(0) = [\phi, 2]$, the results are presented in Fig. 3e–h.

4.3. Numerical example C

In this NE, we investigate the efficacy of (3.1) and (3.4) in three large portfolios, containing of 10 stocks, 20 stocks and 30 stocks. As a result, the experimental findings prove the reliability of (3.1) and (3.4) approaches and demonstrate that it could be applied to large datasets and real-world scenarios.

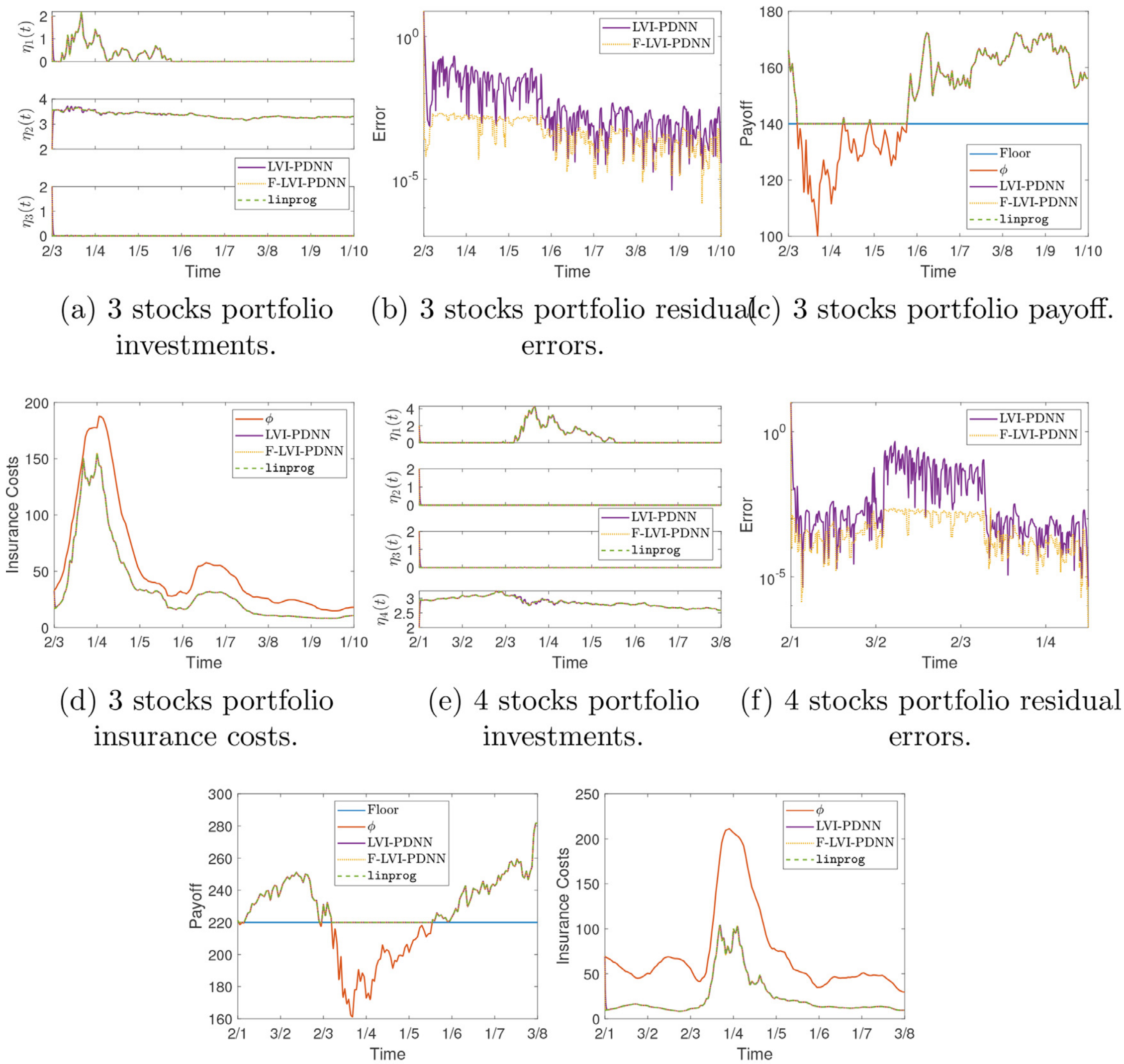
In all portfolio cases of this NE, the time delay parameter has been set to $c = 21$ in the corresponding time-series of Fig. 2, and we find the minimum-cost insured portfolio $\eta(t)$ for the time period 01/05/2019 to 01/10/2019. Furthermore, we divide our time-series into five monthly periods by setting $noep = [23, 20, 22, 22, 21]$, and we set the interval of integration [0,5] in the `ode` solver. Note that the vector $noep$ contains the observations number for each month of the time period 01/05/2019 to 01/10/2019.

Specifically, in the case of 10 stocks portfolio, we set $X(t) = [x_1(t), \dots, x_{10}(t)]^{\mathrm{T}}$, where $X(t)$ contains the daily close prices of market stocks included in the corresponding block of Fig. 2 into $x_1(t), x_2(t), \dots, x_{10}(t)$, respectively. Given a portfolio $\phi = 2\mathrm{ones}(10{,}1)$, a floor $\lambda = 980$, and starting from $z(0) = [\phi, 2]$, the results are presented in Fig. 4a–c. In the case of 20 stocks portfolio, we set $X(t) = [x_1(t), \dots, x_{20}(t)]^{\mathrm{T}}$, where $X(t)$ contains the daily close prices of market stocks included in the corresponding blocks of Fig. 2 into $x_1(t), x_2(t), \dots, x_{20}(t)$, respectively. Given a portfolio $\phi = 2\mathrm{ones}(20{,}1)$, a floor $\lambda = 1980$, and starting from $z(0) = [\phi, 2]$, the results are presented in Fig. 4d–f. In the case of 30 stocks portfolio, we set $X(t) = [x_1(t), \dots, x_{30}(t)]^{\mathrm{T}}$, where $X(t)$ contains the daily close prices of market stocks included in the corresponding blocks of Fig. 2 into $x_1(t), x_2(t), \dots, x_{30}(t)$, respectively. Given a portfolio $\phi = 2\mathrm{ones}(30{,}1)$, a floor $\lambda = 2450$, and starting from $z(0) = [\phi, 2]$, the results are presented in Fig. 4g–i.

4.4. Results discussion

This subsection discusses the findings shown in Figs. 3-4 and compares the performances of (3.1) and (3.4). The optimal portfolios, $\eta(t)$, which contain 3 and 4 stocks, respectively, are shown in Fig. 3a and e. Therein, we notice that the portfolios produced by the `linprog`, (3.1) and (3.4) are identical. Note that `linprog` produces the presumptive theoretical solution.

Figures 3 b, f, 4 a, d and g show the error $\|R(t)\|_F$ of (3.3), produced during the convergence of (3.1) and (3.4) for the portfolios consisting of 3, 4, 10, 20 and 30 stocks, respectively. The noise is expected in these figures since we work with time-series and, considering the parameter's γ small value, the error value is magnificent. However, compared to the



(a) 3 stocks portfolio investments. (b) 3 stocks portfolio residual errors. (c) 3 stocks portfolio payoff.

(d) 3 stocks portfolio insurance costs. (e) 4 stocks portfolio investments. (f) 4 stocks portfolio residual errors.

(g) 4 stocks portfolio payoff. (h) 4 stocks portfolio insurance costs.

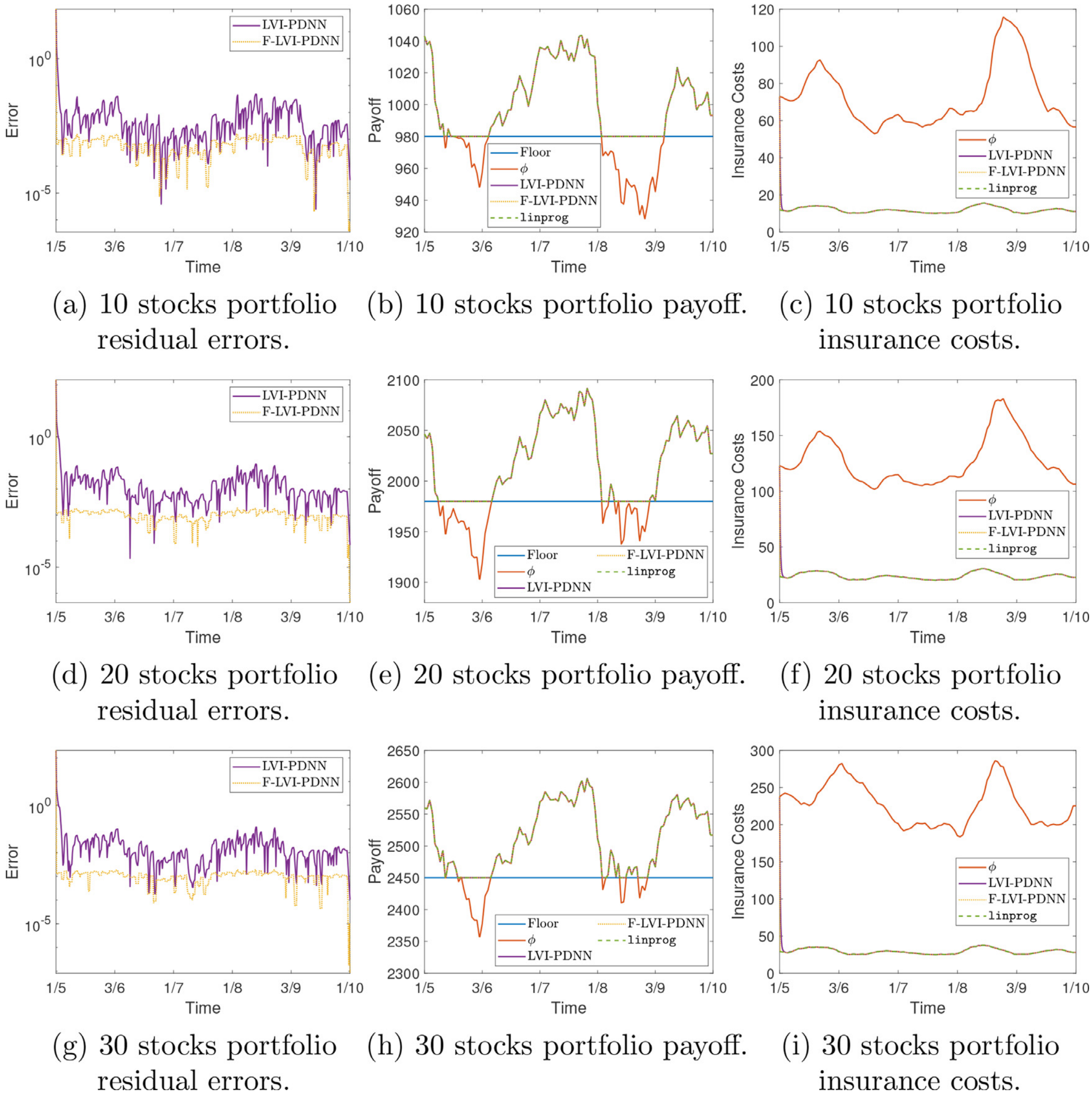
Fig. 3. The convergence, the LVI-PDNN's and F-LVI-PDNN's residual errors, the payoff and the insurance costs for two portfolios containing 3 and 4 stocks, in NE A and B, respectively.

LVI-PDNN, the error produced by the F-LVI-PDNN is far less noisy and converges to zero more quickly. Thereafter, the F-LVI-PDNN outperforms the LVI-PDNN in terms of accuracy.

The floor prices along with the payoffs of the initial portfolios ϕ and the portfolios $\eta(t)$, which contain 3, 4, 10, 20 and 30 stocks, respectively, are shown in Figs. 3c, g, 4 d, e and h. Therein, we notice that the portfolios' payoffs produced by the `linprog`, (3.1) and (3.4) are identical. It is important to note that the payoffs of the portfolios ϕ and $\eta(t)$ are the products of $X^{\mathrm{T}}(t)\cdot\phi$ and $X^{\mathrm{T}}(t)\cdot\eta(t)$, respectively.

Figures 3 d, h, 4 c, f and i present the insurance costs of the initial portfolio ϕ and the portfolios $\eta(t)$, which contain 3, 4, 10, 20 and 30 stocks, respectively. We notice that the insurance costs produced by the `linprog`, (3.1) and (3.4) are the same for the respective portfolios. It is worth noting that the insurance costs of the portfolios ϕ and $\eta(t)$ are the products of $p(t)^{\mathrm{T}}\cdot\phi$ and $p(t)^{\mathrm{T}}\cdot\eta(t)$, respectively.

Comparing the portfolios $\eta(t)$ payoffs in Figs. 3c, g, 4 b, e and h and the portfolios $\eta(t)$ insurance costs in Figs. 3d, h, 4 c, f and i, respectively, we notice that the insurance costs of $\eta(t)$ are rising only in the case where the payoff needs to be kept at the floor. Furthermore, it is obvious that the clear payoff, which is the payoff minus the insurance costs, of portfolio $\eta(t)$ is always greater than the clear payoff of portfolio's ϕ. Moreover, using the parameter ω, which is especially useful when combining different time periods with unequal numbers of observations in each one, is a novel concept. So, by considering the ω parameter, our approach is more realistic. Another important discovery is that, in every NE studied,



(a) 10 stocks portfolio residual errors. (b) 10 stocks portfolio payoff. (c) 10 stocks portfolio insurance costs.

(d) 20 stocks portfolio residual errors. (e) 20 stocks portfolio payoff. (f) 20 stocks portfolio insurance costs.

(g) 30 stocks portfolio residual errors. (h) 30 stocks portfolio payoff. (i) 30 stocks portfolio insurance costs.

Fig. 4. The convergence, the LVI-PDNN's and F-LVI-PDNN's residual errors, the payoff and the insurance costs for portfolios containing 10, 20 and 30 stocks in NE C.

asking for a time-varying portfolio $\eta(t)$, as opposed to a constant portfolio ϕ, results in much cheaper portfolio insurance costs. Comprehensively, the experiments on two small and three large portfolios show that the F-LVI-PDNN outperforms the LVI-PDNN and both performed admirably in solving the TMPI problem.

5. Conclusion

The TMPI problem was presented in this paper, and it was solved using a recurrent neural network dubbed LVI-PDNN. In order to improve the performance of the standard LVI-PDNN model, an adaptive F-LVI-PDNN model was also introduced and studied. A number of NEs demonstrated the competence of the (3.1) and (3.4) approaches in a financial TVLP problem. Our experiments lead us to the conclusion that both the LVI-PDNN and F-LVI-PDNN produce the online solution of the TMPI problem, with the F-LVI-PDNN providing a faster convergence and greater accuracy than the LVI-PDNN. An important discovery is that choosing a time-varying portfolio over a constant portfolio leads in much cheaper portfolio insurance costs. The reliability of the LVI-PDNN and F-LVI-PDNN techniques was proven by experimental findings, which also demonstrated that they could be applied to large datasets and real-world scenarios.

Data Availability

I have a shared code and data link inside the paper.

Acknowledgments

Predrag Stanimirović is supported by Ministry of Education, Science and Technological Development, Republic of Serbia, Contract No. 451-03-68/2020-14/200124.

Predrag Stanimirović is supported by the Science Fund of the Republic of Serbia, #GRANT No 7750185, Quantitative Automata Models: Fundamental Problems and Applications - QUAM.

This work was supported by the Ministry of Science and Higher Education of the Russian Federation (Grant No. 075-15-2022-1121).

References

[1] A.H. Khan, X. Cao, V.N. Katsikis, P. Stanimirovic, I. Brajevic, S. Li, S. Kadry, Y. Nam, Optimal portfolio management for engineering problems using nonconvex cardinality constraint: a computing perspective, IEEE Access (2020), doi:10.1109/access.2020.2982195. 1–1

[2] A.H. Khan, X. Cao, S. Li, V.N. Katsikis, L. Liao, BAS-ADAM: an ADAM based approach to improve the performance of beetle antennae search optimizer, IEEE/CAA J. Autom. Sin. 7 (2) (2020) 461–471.

[3] V.N. Katsikis, S.D. Mourtas, Binary beetle antennae search algorithm for tangency portfolio diversification, J. Model. Optim. 13 (1) (2021) 44–50, doi:10.32732/jmo.2021.13.1.44.

[4] V.N. Katsikis, S.D. Mourtas, Diversification of time-varying tangency portfolio under nonlinear constraints through semi-integer beetle antennae search algorithm, Appl. Math. 1 (1) (2021) 63–73, doi:10.3390/appliedmath1010005.

[5] 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 Oper. Res. Forum 2 (18) (2021), doi:10.1007/s43069-021-00060-5.

[6] S.D. Mourtas, V.N. Katsikis, V-shaped BAS: applications on large portfolios selection problem, Comput. Econ. (2021), doi:10.1007/s10614-021-10184-9.

[7] V.N. Katsikis, S.D. Mourtas, Optimal portfolio insurance under nonlinear transaction costs, J. Model. Optim. 12 (2) (2020) 117–124.

[8] N. Premalatha, A.V. Arasu, Prediction of solar radiation for solar systems by using ANN models with different back propagation algorithms, J. Appl. Res. Technol. 14 (3) (2016) 206–214, doi:10.1016/j.jart.2016.05.001.

[9] S. Mourtas, V. Katsikis, C. Kasimis, Feedback control systems stabilization using a bio-inspired neural network, EAI Endorsed Trans. AI Rob. 1 (1) (2022) 1–13, doi:10.4108/airo.v1i.17.

[10] S.D. Mourtas, V.N. Katsikis, Exploiting the Black-Litterman framework through error-correction neural networks, Neurocomputing 498 (2022) 43–58, doi:10.1016/j.neucom.2022.05.036.

[11] S.D. Mourtas, C. Kasimis, Exploiting mean-variance portfolio optimization problems through zeroing neural networks, Mathematics 10 (3079) (2022) 3079, doi:10.3390/math10173079.

[12] S.D. Mourtas, V.N. Katsikis, E. Drakonakis, S. Kotsios, Stabilization of stochastic exchange rate dynamics under central bank intervention using neuronets, Int. J. Inf. Technol.Decis. Making (2022) 1–29, doi:10.1142/s0219622022500560.

[13] V.N. Katsikis, S.D. Mourtas, P.S. Stanimirović, Y. Zhang, Continuous-time varying complex QR decomposition via zeroing neural dynamics, Neural Process. Lett. (2021), doi:10.1007/s11063-021-10566-y.

[14] V.N. Katsikis, S.D. Mourtas, P.S. Stanimirović, S. Li, X. Cao, Time-varying mean-variance portfolio selection problem solving via LVI-PDNN, Comput. Oper. Res. 138 (2022) 105582, doi:10.1016/j.cor.2021.105582.

[15] A. Wang, L. Liu, J. Qiu, G. Feng, Event-triggered robust adaptive fuzzy control for a class of nonlinear systems, IEEE Trans. Fuzzy Syst. 27 (8) (2019) 1648–1658, doi:10.1109/TFUZZ.2018.2886158.

[16] F. Wang, B. Chen, X. Liu, C. Lin, Finite-time adaptive fuzzy tracking control design for nonlinear systems, IEEE Trans. Fuzzy Syst. 26 (3) (2018) 1207–1216, doi:10.1109/TFUZZ.2017.2717804.

[17] R. Palm, B. Iliev, B. Kadmiry, Recognition of human grasps by time-clustering and fuzzy modeling, Rob. Auton. Syst. 57 (5) (2009) 484–495, doi:10.1016/j.robot.2008.10.012.

[18] S. Feng, H. Wu, Hybrid robust boundary and fuzzy control for disturbance attenuation of nonlinear coupled ode-beam systems with application to a flexible spacecraft, IEEE Trans. Fuzzy Syst. 25 (5) (2017) 1293–1305, doi:10.1109/TFUZZ.2016.2612264.

[19] R.-E. Precup, M.-L. Tomescu, C.A. Dragos, Stabilization of Rössler chaotic dynamical system using fuzzy logic control algorithm, Int. J. Gen. Syst. 43 (5) (2014) 413–433, doi:10.1080/03081079.2014.893299.

[20] H.A. Hagras, A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots, IEEE Trans. Fuzzy Syst. 12 (4) (2004) 524–539, doi:10.1109/tfuzz.2004.832538.

[21] S. Guo, L. Han, Stability and Control of Nonlinear Time-Varying Systems, Springer, Singapore, 2018, pp. 105–130, doi:10.1007/978-981-10-8908-4_5. Ch. Fuzzy observer, fuzzy controller design, and common Hurwitz matrices for a class of uncertain nonlinear systems

[22] B. Wang, J. Cheng, J. Zhan, A sojourn probability approach to fuzzy-model-based reliable control for switched systems with mode-dependent time-varying delays, Nonlinear Anal. Hybrid Syst 26 (2017) 239–253, doi:10.1016/j.nahs.2017.05.006.

[23] J. Dai, P. Tan, X. Yang, L. Xiao, L. Jia, Y. He, A fuzzy adaptive zeroing neural network with superior finite-time convergence for solving time-variant linear matrix equations, Knowl. Based Syst. 242 (2022) 108405, doi:10.1016/j.knosys.2022.108405.

[24] V.N. Katsikis, P.S. Stanimirović, S.D. Mourtas, L. Xiao, D. Karabasević, D. Stanujkić, Zeroing neural network with fuzzy parameter for computing pseudoinverse of arbitrary matrix, IEEE Trans. Fuzzy Syst. 30 (9) (2022) 3426–3435, doi:10.1109/tfuzz.2021.3115969.

[25] 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], Appl. Math. Comput. 349 (2019) 221–244, doi:10.1016/j.amc.2018.12.044.

[26] J. Annaert, M.D. Ceuster, J. Vandenbroucke, Mind the floor: enhance portfolio insurance without borrowing, J. Investing 28 (2019) 39–50, doi:10.3905/joi.2019.1.076.

[27] K. Matsumoto, Portfolio insurance with liquidity risk, Asia-Pacific Financ. Mark. 14 (4) (2008) 363, doi:10.1007/s10690-008-9067-z.

[28] 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), Appl. Math. Comput. 385 (2020) 125453.

[29] V.N. Katsikis, S.D. Mourtas, Computational Management, Vol. 18 of Modeling and Optimization in Science and Technologies, Springer, Cham, 2021, pp. 305–323, doi:10.1007/978-3-030-72929-5_14. Ch. Portfolio insurance and intelligent algorithms

[30] C.D. Aliprantis, D.J. Brown, J. Werner, Minimum-cost portfolio insurance, J. Econ. Dyn. Control 24 (2000) 1703–1719, doi:10.1016/s0165-1889(99)00091-3.

[31] V.N. Katsikis, S.D. Mourtas, ORPIT: a matlab toolbox for option replication and portfolio insurance in incomplete markets, Comput. Econ. (2019) 1, doi:10.1007/s10614-019-09936-5.
[32] Y. Zhang, D. Guo, Linear programming versus quadratic programming in robots' repetitive redundancy resolution: a chattering phenomenon investigation, in: Proc. 4th IEEE Conf. Industrial Electronics and Applications, 2009, pp. 2822–2827, doi:10.1109/ICIEA.2009.5138724.
[33] Y. Zhang, W.E. Leithead, Exploiting Hessian matrix and trust-region algorithm in hyperparameters estimation of Gaussian process, Appl. Math. Comput. 171 (2) (2005) 1264–1281, doi:10.1016/j.amc.2005.01.113.
[34] Y. Zhang, W. Ma, X. Li, H. Tan, K. Chen, MATLAB Simulink modeling and simulation of LVI-based primal-dual neural network for solving linear and quadratic programs, Neurocomputing 72 (7–9) (2009) 1679–1687, doi:10.1016/j.neucom.2008.07.008.
[35] Y. Zhang, On the LVI-based primal-dual neural network for solving online linear and quadratic programming problems, in: Proc. 2005, American Control Conf, vol. 2, 2005, pp. 1351–1356, doi:10.1109/ACC.2005.1470152.
[36] 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 Trans. Neural Netw. 14 (3) (2003) 658–667, doi:10.1109/TNN.2003.810607.
[37] Y. Zhang, S.S. Ge, T.H. Lee, A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators, Part B (Cybernetics) IEEE Trans. Syst.Man Cybern. 34 (5) (2004) 2126–2132, doi:10.1109/TSMCB.2004.830347.
[38] S.S. Ge, Y. Zhang, T.H. Lee, An acceleration-based weighting scheme for minimum-effort inverse kinematics of redundant manipulators, in: Proc. IEEE Int. Symp. Intelligent Control, 2004, pp. 275–280, doi:10.1109/ISIC.2004.1387695.
[39] S. Wu, L. He, M. Zou, J. Li, Y. Zhang, Time-varying linear programming via LVI-PDNN with numerical examples, in: Proc. IEEE 13th Int. Conf. Signal Processing (ICSP), 2016, pp. 326–331, doi:10.1109/ICSP.2016.7877849.