Engineering Applications of Artificial Intelligence 133 (2024) 108599

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Research paper

Optimal dynamic fixed-mix portfolios based on reinforcement learning with second order stochastic dominance

Giorgio Consigli [a,*], Alvaro A. Gomez [a], Jorge P. Zubelli [a,b]

[a] Department of Mathematics, Khalifa University of Science & Technology, Abu Dhabi, United Arab Emirates
[b] ADIA LAB, Level 26, Al Khatem Tower, Abu Dhabi, United Arab Emirates

ARTICLE INFO

Keywords:
Fixed-mix portfolios
Stochastic dominance
Reinforcement learning
Actor–critic approach
Deep learning
Stochastic gradient

ABSTRACT

We propose a *reinforcement learning* (RL) approach to address a multiperiod optimization problem in which a portfolio manager seeks an optimal constant proportion portfolio strategy by minimizing a tail risk measure consistent with *second order stochastic dominance* (SSD) principles. As a risk measure, we consider in particular the *Interval Conditional Value-at-Risk* (ICVaR) shown to be mathematically related to SSD principles. By including the ICVaR in the reward function of an RL method we show that an *optimal fixed-mix* policy can be derived as solution of short- to medium-term allocation problems through an accurate specification of the learning parameters under general statistical assumptions. The financial optimization problem, thus, carries several novel features and the article details the required steps to accommodate those features within a reinforcement learning architecture. The methodology is tested in- and out-of-sample on market data showing good performance relative to the SP500, adopted as benchmark policy.

0. Introduction

We consider in this contribution a popular portfolio selection model, the so-called *fixed-mix* (FxM) model, that carries a long history in financial practice (Bianchi and Guidolin, 2014). The FxM paradigm, also commonly referred to as *constant proportion portfolio insurance* (CPPI), was established rigorously in finance theory by Black and Perold (1992) under Gaussian assumption on the risky assets' return processes. It was then generalized to the case of discontinuous processes by Cont and Tankov (2009). The rationale of FxM investment strategies is simple, mainly motivated by the stable performance induced by constant proportion portfolios and its simple rationale: buy low and sell high relative to an evolving average market scenario. The 60%–40% equity-bond proportions, for instance, have been advocated over the years as a consistent portfolio composition to attain high performance in the medium-long term (Bender et al., 2010) even outside of classical Gaussian assumptions. The literature on the topic is extensive and rich, see Ziemba and Ziemba (2008), Dempster et al. (2011) to span the early debate on this investment rule. In this introduction, we analyze the key elements of this financial problem and motivate this contribution from a financial and methodological perspective. In the following section, we frame the work in the state-of-the-art and discuss the specific contributions of this article.

From a mathematical perspective the optimization problem associated with an FxM policy was already highlighted by Fleten et al. (2002), Dempster and Leemans (2006) to result in a non-convex optimization problem, making the derivation of the optimal constant proportions hard through numerical methods and jeopardizing the possibility to derive closed-form solutions. In the case of a discrete, scenario-based formulation, however, Dempster et al. (2007), by emphasizing its *near convexity* under relatively general assumptions, proposed a solution method based on a search routine followed by a local convex optimizer. A common way to derive this class of portfolio strategies, traditionally, relied then on policy simulation (Kim et al., 2014; Denault and Simonato, 2017).

As the first motivation of this research, the development of a reinforcement learning approach that would help through the learning process to overcome the potential lack of convexity and the risk of local optima.

Furthermore, from a financial viewpoint, the definition of an optimal dynamic FxM policy has become popular as an allocation criterion when allowing for sufficiently extended investment horizons, typically several years, and rarely associated with some form of risk control or considered in relationship to some benchmark financial strategy. As a result of these limitations, even if effective in the long term, CPPI is known to be exposed to possible losses over short periods. Instead, in this work, the derivation of an optimal FxM strategy is considered over a short-term horizon and jointly with the optimization of a tail

[*] Corresponding author.
E-mail address: giorgio.consigli@ku.ac.ae (G. Consigli).

https://doi.org/10.1016/j.engappai.2024.108599
Received 30 October 2023; Received in revised form 29 March 2024; Accepted 7 May 2024

risk measure. To wit, the *Interval Conditional Value at Risk* (ICVaR). The relationship between ICVaR and *second-order stochastic dominance* (SSD) was established in Liu et al. (2021) and will be recalled below to clarify the adopted modeling approach. We show in the computational results that the proposed decision paradigm, based on a stationary fixed-mix policy and the ICVaR optimization under SSD conditions, leads to an effective short-term risk control of portfolio dynamics and positive risk-adjusted returns over increasing investment horizons, from very short: one month, to medium-term: one and a half years.

The adoption of a reinforcement learning approach has other more general motivations. The optimization problem is formulated below with a continuous action space, to determine optimal portfolio allocations, a discrete model of uncertainty, and a risk-based objective function. Classical methods to tackle this general problem include *stochastic dynamic programming* (SDP) approaches (Infanger, 2008) under Markovian assumptions or *multistage stochastic programming* (MSP) methods (Dupačová et al., 2000; Consigli et al., 2016). In the presence of partial orders among the probability distributions, as in the case of stochastic dominance constraints, see Ogryczak and Ruszczyński (2001), Dentcheva and Ruszczyński (2003), Ruszczynski (2010), both the SDP and the MSP formulations suffer from the so-called *curse of dimensionality* and have proven computationally infeasible (Gomez et al., 2024).

Indeed machine learning techniques and the RL methodology have been recently used to tackle large-scale stochastic optimization problems as shown in Bayraktar and Kara (2023), Jaimungal (2022), Al-Aradi et al. (2018), Han et al. (2017), Huré et al. (2020), Wang et al. (2020), Hambly et al. (2021). The opportunity to accommodate a concave objective function, based on the ICVaR measure, furthermore, is provided by recent advances in so-called *risk-sensitive RL* and convex RL: the optimization of the classical *Conditional Value-at-Risk* introduced by Rockafellar and Uryasev (2002) has been recently considered in those contexts. See Tamar et al. (2014), Chow et al. (2018). We provide further details, here below, in Section 1.

Problems in RL that involve optimizing concave or convex objective functions are now addressed within a novel framework known as convex RL (Mutti et al., 2023; Miryoosefi et al., 2019; Zhang et al., 2020; Geist et al., 2021; Zahavy et al., 2021), where the objective function can take on either concave or convex forms. We frame this contribution in the state-of-the-art in Section 1.

After this introduction, the article evolves from the analysis here next in Section 1 of the state-of-the-art relevant to properly frame and characterize our contribution from a methodological and financial perspective to the definition of the portfolio optimization model in Section 2, where the adopted risk measure and fixed-mix constraints are explained in detail, to Section 3 focusing on the RL methodology and the adopted RL algorithm. In Section 4 we present extended computational evidence to validate the proposed approach in- and out-of-sample, before the conclusion with an indication of future research. The article includes an extended set of results presented in Appendix.

1. State-of-the-art and contribution

The state-of-the-art, specifically in the domain of RL developments for financial optimization and portfolio management problems, is growing rapidly. Yet, it is reasonable to say that as of today, the presented RL approaches are rather problem-specific, and this holds for the methodology presented in this contribution. A step forward to deal with dynamic control problems, came from the introduction of deep reinforcement learning, specifically through the Deep Q Network (DQN) method (Mnih et al., 2015, 2016, 2013).

Under relatively mild assumptions, however an issue of value function overestimation, specifically associated with the DQN algorithm was reported by Hasselt (2010), Hasselt et al. (2016), who proposed the double Q-learning method (DDQN). Unlike the single-network approach of DQN, DDQN utilizes two separate networks. The first, known as the online network, selects actions, while the second, the target network, evaluates these actions. This dual-network strategy effectively tackles the overestimation problem encountered in the DQN framework. Both the DQN and DDQN methods are primarily based on value-based approaches, aiming to determine the optimal action solely from an action-value function. These methods are typically employed when the action space is discrete.

In many reinforcement learning (RL) scenarios, however, the action space is continuous, making discretization impractical. Value-based RL methods often struggle to handle this complexity effectively. In recent years, successful implementations of deep deterministic policy gradient (DDPG) methods have emerged to address continuous action spaces, as proposed by Silver et al. (2014), Lillicrap et al. (2015), Gu et al. (2016), Wang et al. (2020) (2015). DDPG algorithms are grounded in the actor–critic paradigm, where two neural networks interact. The actor network determines actions, while the critic network evaluates the action-value function.

In the work (Zhang et al., 2020), the authors introduce a Variational Policy Gradient (VPG) method, which extends the classical Policy Gradient method (Silver et al., 2014). This method extended the applicability of reinforcement learning (RL) techniques from problems based on simple reward functions to those formulated with more general convex (or concave) objective functions. It anticipated the stream of contributions now associated with the general class of convex RL methods, mentioned in the introduction.

Early applications of ML relied on the adoption of a linear reward function. This, for instance, was the case in a financial context of Deng et al. (2016), in which the authors employ deep direct reinforcement learning to maximize the expected compounded return for a trading system. Specifically in financial optimization problems such assumption is consistent with so-called risk-neutral investors.

In various real-world applications, including those in financial engineering, there are numerous challenges involving a general concave or convex function Q. For instance, in risk-sensitive problems, when the action value Q depends on a risk measure (Dentcheva and Ruszczyński, 2008; Fei et al., 2020).

Most recent novel RL algorithms have been proposed to solve more complex dynamic risk-control problems based on time-consistent formulations and general risk functions (Jaimungal, 2022; Coache and Jaimungal, 2024; Coache et al., 2023; Chow et al., 2018). The dynamic model proposed in Das and Varma (2020) based as a risk measure of the shortfall for a pre-specified investment goal is particularly relevant to the present article. This is formulated as a Markov decision problem (MDP) and solved by backward recursion based on a specific characterization of the value function. We adopt this RL methodology as a benchmark to analyze the properties of the algorithm proposed below.

We have summarized the set of RL methods we consider relevant to assessing our contribution in Table 1. From top to bottom, row-wise we recall a set of key contributions spanning from 2013 to 2023, from the works on DQN to more recent works employing convex RL. For a more comprehensive recent overview of the state-of-the-art in RL, readers are referred to Shakya et al. (2023).

Table 2, instead provides a concise summary of RL-based contributions from 2017 to date, specifically in the finance domain, which preceded our contribution. It is hardly meant to be an exhaustive account of a continuously evolving scientific domain, but surely it conveys the contributions we have considered in our proposal.

Compared with previous applications of RL methodologies in portfolio management, which usually rely on a recursive formula, as in standard RL and RL with dynamic risk measures, our work considers a more general concave utility function which is defined only at terminal stage. This utility function includes a penalty component to enforce stochastic dominance, which is used as a reward function in the optimization model described in Section 2.

We summarize previous RL approaches in finance and portfolio management in Table 2.

Table 1
Summary of recent advancements in deep and convex reinforcement learning methods.

RL method	Advances	Assumptions made	Limitations
• Deep Q Network (DQN)(2013) (Mnih et al., 2015, 2016, 2013)	• This approach employs deep neural networks to learn the action value function	• The objective function takes the form of linear compounded returns. No risk evaluation	• Discrete action space • Poor risk assessment • Overestimation of the value function
• Double Deep Q Network (DDQN) (2015) (Hasselt, 2010; Hasselt et al., 2016)	• Addresses the overestimation of the value function in DQN.	• The objective function takes the form of linear compounded returns. No risk evaluation	• Discrete action space • Poor risk assessment
• Deep Deterministic Policy Gradient (DDPG) (2016) (Silver et al., 2014; Lillicrap et al., 2015; Gu et al., 2016; Wang et al., 2020)	• Addresses the deep RL problem for continuous spaces	• Continuous action space • The objective function takes the form of a linear compounded return	• Can only be applied to compounded linear rewards objective functions
• Variational Policy Gradient (2020) (Zhang et al., 2020)	• Generalizes the policy gradient methodology to concave or convex objective functions	• Continuous action space • Concave or convex objective functions	• Assumes an infinite number of scenarios or trials
• Convex Reinforcement Learning (2023) (Mutti et al., 2023)	• Considers the case of finite scenarios or trials	• Concave (or convex) objective functions and continuous action space	• The method does not address non-concave (non-convex) objective functions

The derivation through reinforcement learning of an optimal dynamic fixed-mix policy, based on a risk measure whose minimization leads to a strategy *stochastically dominating* a benchmark to the second order, represents the key contribution of this research paper from a financial and decision modeling perspective. The stochastic environment is characterized by a *model-free* data-driven return model on which the *training* phase is conducted. From a methodological perspective, we propose an RL approach relying on DDPG and convex RL methodologies.

In summary, the following can thus be claimed as specific contributions of this article to be validated in the following sections:

- The derivation of an optimal constant proportion investment policy by reinforcement learning consistent with *second order stochastic dominance* (SSD) over a benchmark portfolio.
- We show that such optimal fixed-mix, contrary to traditional results, may turn out very effective over a very short-term investment horizon by out-of-sample backtesting.
- The proposed methodology is based on the exploration-exploitation dilemma, the deep deterministic policy gradient, and recent advancements in convex reinforcement learning techniques. In Section 3, we provide detailed insight into why these recent RL techniques are chosen to tackle the fixed-mix portfolio optimization problem introduced in Section 2. The proposed approach shows convergence in computational results, as discussed in Section 4.

2. Portfolio problem formulation

We assume a problem in which a portfolio manager seeks the definition of an optimal constant proportion θ_i invested in asset $i = 1, 2, \dots, I$ defined as a ratio to the current portfolio value. Assuming a finite and discrete time set $t \in \mathcal{T}$, $\mathcal{T} := \{0, 1, 2, \dots, T\}$, then for every t, according to current prices, she/he is expected to rebalance the portfolio to recover that proportion. $\mathcal{T}$ is referred to as the planning horizon of the problem, which terminates in T. Consistently with canonical *non-anticipativity* requirements of the investment policy, every portfolio allocation must occur under residual uncertainty: the first portfolio allocation occurs at $t = 0$ and the last one will then occur at the beginning of the last period. Accordingly, asset returns are random variables at decision times and realized at the end of each stage, until T. The FxM policy rule is determined to maximize the expected terminal value of the portfolio while controlling the excess tail risk relative to a benchmark portfolio, here denoted by y_T. To this purpose we introduce as a risk measure, the interval Conditional Value-at-Risk (ICVaR), defined below as a function of the coefficients α and β and denoted by $\rho_{\alpha,\beta}(.)$, as further clarified in Eq. (4). We look for the solution by reinforcement learning of the following *multistage stochastic optimization* problem:

$$\max_{\theta} \ \mathbb{E}(w(x_T)) + [\rho_{\alpha,\beta}(w(x_T)) - \rho_{\alpha,\beta}(w(y_T))]_- \tag{1a}$$

$$\text{s.t.} \quad w(x_t) = \sum_{i=1}^{m} x_{i,t} + x_{0,t} \qquad t = 1, \dots, T, \tag{1b}$$

$$x_{i,0} = \hat{x}_{i,0} + x^+_{i,0} - x^-_{i,0}, \qquad i = 1, \dots, I, \tag{1c}$$

$$x_{0,0} = \hat{x}_{0,0} + \sum_{i=1}^{I} x^-_{i,0}(1 - c_s) - \sum_{i=1}^{I} x^+_{i,0}(1 + c_b) \tag{1d}$$

$$\frac{x^+_{i,0} - x^-_{i,0} + \hat{x}_{i,0}}{\sum_i x_{i,0}} = \theta_i, \ i = 1, \dots, I, \tag{1e}$$

$$x_{i,t} = x_{i,t-1}(1 + r_{i,t}) + x^+_{i,t} - x^-_{i,t}, \ i = 1, \dots, I, \ t = 1, \dots, T, \tag{1f}$$

$$x_{0,t} = x_{0,t-1}(1 + r_{0,t}) + \sum_{i=1}^{I} x^-_i(1 - c_s) - \sum_{i=1}^{I} x^+_i(1 + c_b) \tag{1g}$$

$$\frac{x^+_{i,t} - x^-_{i,t} + x_{i,t-1}(1 + r_{i,t})}{\sum_i x_{i,t}} = \theta_i, \ i = 1, \dots, I, \ t = 1, \dots, T, \tag{1h}$$

Table 2
Recent advances in portfolio management based on reinforcement learning.

Work	Contribution	Methodology
• Deep direct reinforcement learning for financial signal representation and trading (2016) (Deng et al., 2016)	• The paper introduces an algorithm that utilizes reinforcement learning (RL) for trading strategies. This algorithm is tested in both the stock and commodity futures markets	• The work introduces the Direct RL algorithm to optimize cumulative rewards
• A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem (2017) (Jiang et al., 2017)	• The work presents a dynamic portfolio allocation aimed at maximizing the logarithmic cumulative returns	• The methodology employs the DDPG method with a long short-term memory (LSTM) neural network structure
• A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown (2017) (Almahdi and Yang, 2017)	• The paper addresses a dynamic risk-return portfolio model that includes the Sharpe ratio	• The work utilizes recurrent RL to optimize the respective objective function
• Continuous-time mean–variance portfolio selection (2020) (Wang and Zhou, 2020)	• Solves a dynamic mean–variance portfolio model using RL	• Introduces the RL Exploratory Mean–Variance algorithm and compares it against two other algorithms: maximum likelihood estimation and DDPG
• Dynamic goal-based wealth management using reinforcement learning (2020) (Das and Varma, 2020)	• Solves a dynamic goal-based portfolio problem using an RL approach	• The methodology discretizes the set of optimal actions and uses the deep Q - learning method to solve it.
• The reinforcement learning Kelly strategy (2022) (Jiang et al., 2022)	• Solves a dynamic Kelly portfolio strategy using an RL approach	• This work utilizes entropy-regularization RL to solve the optimization problem
• Reinforcement Learning with Dynamic Convex Risk Measures (2022–2023) (Coache and Jaimungal, 2024; Coache et al., 2023)	• The paper develops an algorithm to solve optimization problems with time-consistent dynamic risk measures. It also presents financial applications within this context.	• The algorithm utilizes DDPG with a recurrent learning methodology to learn dynamic risk measures
• Optimal dynamic fixed-mix portfolios based on reinforcement learning with second order stochastic dominance (2024) (This work)	• A dynamic fixed-mix portfolio strategy with stochastic dominance is solved using an RL approach	• The optimization problem is solved by combining DDPG and Convex reinforcement learning methods

$$x^+_{i,T} = x^-_{i,T} = 0, \qquad i = 1,\dots,I, \tag{1i}$$

$$w(x_t), x_{i,t}, x^+_{i,t}, x^-_{i,t} \geq 0, \qquad i = 1,\dots,I,\ t = 0,1,\dots,T-1. \tag{1j}$$

We denote the portfolio allocation in asset i at time t by $\{x_{i,t}\}_{i=1}^I$, and the rebalancing decisions, in terms of buying and selling decisions, by $\{x^+_{i,t}\}_{i=1}^I$ and $\{x^-_{i,t}\}_{i=1}^I$ respectively, see (1c) and (1f). We assume that no rebalancing decisions are allowed at the end of the planning horizon T, as from (1i). The wealth process $\{w(x_t)\}_{t=1}^T$ in Eq. (1b) is determined by the evolution of the investment portfolio x_t and cash surpluses $x_{0,t}$.

Furthermore, $Y = \{y\}_{t=0}^T$ denotes the benchmark portfolio process determined exogenously and assumed in what follows to reflect the $SP500$ market index. Finally, here next, we let W_T and Y_T denote respectively $w(x_T)$ and $w(y_T)$, the terminal portfolio values.

The fixed-mix strategy is enforced through constraint (1h). Formally, through the parameter $\theta = (\theta_1,\dots,\theta_I) \in \mathbb{R}^I_+$ where $\sum_i \theta_i = 1$, we define an investment policy. Depending on the return of an asset relative to the portfolio, if $x_{i,t} > \theta_i \sum_i x_{i,t}$ a selling decision will be adopted, while if smaller $x_{i,t} < \theta_i \sum_i x_{i,t}$, a buying decision:

$$x_{i,t-1}(1+r_{i,t})+x^+_{i,t}\delta_{\{x_{i,t-1}(1+r_{i,t})<\theta_i \sum_i x_{i,t}\}}-x^-_{i,t}\delta_{\{x_{i,t-1}(1+r_{i,t})>\theta_i \sum_i x_{i,t}\}} = \theta_i \sum_i x_{i,t}. \tag{2}$$

For every asset $i = 1,2,\dots,I$, Eq. (2) represents the constraint associated with the fixed-mix policy in problem (1a). Since the vector θ reflects the portfolio weights, then, if necessary, a normalization step is introduced in the learning process, as further explained below.

At $t = 0$, we also specify an input portfolio $\hat{x}_{i,0}$, if any. The optimal decision at $t = 0$ as in (1c) will be determined by buying or selling according to the fixed-mix strategy (1e).

The objective function can be written as:

$$\mathbb{E}(W_T) + [\rho_{\alpha,\beta}(W_T) - \rho_{\alpha,\beta}(Y_T)]_-, \tag{3}$$

with $[x]_- = min(0,x)$. Eq. (3) specifies the objective function in terms of expected wealth, a performance measure, and a risk measure here determined by the difference between the ICVaR of the managed portfolio and the benchmark: the higher such difference, the better.

The introduction of the penalty term $[\rho_{\alpha,\beta}(W) - \rho_{\alpha,\beta}(Y)]_-$ in the objective function (1a), is based on the relationship between the ICVaR and *second order stochastic dominance* (SSD).

By definition, the ICVaR is the expected value over a shortfall distribution specified on the left of β and it generalizes the *Conditional Value-at-Risk* (CVaR), over the restricted domain $(-\infty, \beta]$:

$$\rho_{\alpha,\beta}(W) = \sup_{\eta \leq \beta} \{\eta - \frac{1}{1-\alpha}\mathbb{E}[\eta - W]_+\}, \ \alpha \in [0,1). \tag{4}$$

For $\beta = VaR_\alpha$ in the return distribution, then $\rho_{\alpha,\beta}$ will agree with the canonical $CVaR_\alpha(W)$ risk measure.

The following results from Liu et al. (2021) establishes the relationship between the SSD partial order and $\rho_{\alpha,\beta}(.)$:

Proposition 1. *The constraint* $F_2(W,\eta) \leq F_2(Y,\eta), \quad \forall \ \eta \leq \beta$, *is equivalent to* $\rho_{\alpha,\beta}(W) \geq \rho_{\alpha,\beta}(Y), \quad \forall \ \alpha \in [0,1)$.

We refer the reader to Liu et al. (2021) for the proof of this result. In Proposition 1 $F_k(W,\eta) = \frac{\mathbb{E}[(\eta-W)_+^{k-1}]}{(k-1)!}$ if $k > 1$ and $F_1(W,\eta) = \mathbb{P}[W \leq \eta]$. Based on the reference point β and $k = \{1,2\}$ and random variables W and Y, it becomes natural to establish a stochastic dominance order $W \succeq_{(k,\beta)} Y$ between the two. The theory on a continuous spanning of partial orders for $k = 1,2, \beta \in \mathbb{R}$ is developed in Liu et al. (2021), following previous contributions by Tsetlin et al. (2015), Müller et al. (2017).

From a decision-theoretic viewpoint, thanks to the ICVaR, we can define, as we do here, a mean-risk trade-off model, that without explicitly introducing a feasibility condition based on SSD order or any multivariate version of it, may enforce the stochastic dominance through time, by relying on the ICVaR (Gomez et al., 2024). On these grounds, we employ a reward function in the RL approach based on those principles.

3. Methodology

We propose a deep reinforcement learning (RL) methodology to solve the dynamic portfolio problem (1). Consider again the objective function (1a): it includes the expected terminal wealth and a penalty function based on the ICVaR associated with the managed portfolio, say X, and the benchmark Y. Let $Q(T, W_T^\theta, \rho_{\alpha,\beta}(W^X, Y)) : \mathcal{T} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a very general concave function of a risk-averse decision maker. To simplify notation, let $Q(W_T^\theta)$ be the such function. The optimization problem (1a) can be written in a very compact way, under an extended set of constraints, as

$$\max_\theta Q(W_T^\theta) \ , \tag{5}$$

where, the maximization is over the set of investment policies θ, to represent the fixed-mix strategy, and Q is a positive expected terminal utility defined as a function of the expected wealth and the penalty at the end of the investment horizon. The terminal wealth W_T^θ is attained as a result of the sequence of non-anticipative fixed-mix allocations x_t^θ and random returns $\hat{r}_t$. Reinforcement learning provides an approach to solving the general problem (5).

Implementing a reinforcement learning approach encounters modeling challenges, particularly when dealing with an objective function as in (5). However, in specific cases where the Q functions follow specific forms, reinforcement learning shows promising performance.

This section is organized in two parts. In the first one, we explain the reinforcement learning methodology adopted to solve problem (5). In the second part, we provide further details on all specific steps employed to solve the fixed-mix problem (1a).

3.1. Deep reinforcement learning

In this section, we utilize a refined approach based on the deterministic policy gradient method (DDPG) and convex reinforcement learning techniques to address Eq. (1a).

The main idea of the algorithm is to utilize the DDPG approach similar to standard reinforcement learning. However, we enhance it by updating the neural network responsible for the action-value function Q (critic network) using convex reinforcement learning principles, given the concave nature of Q.

Let $\mathcal{S}$ the set of all possible states and we denote by $\mathcal{A}$ the set of possible actions. In what follows, specifically in the context of dynamic portfolio optimization, $\mathcal{A}$ is assumed to be continuous. Over a finite and discrete investment horizon $t = 1, \ldots, T$, we focus on a dynamical system evolving from state $S_t \in \mathcal{S}$ to state $S_{t+1} \in \mathcal{S}$ as a result of action $A_t \in \mathcal{A}$. In this context, the definition of a stationary conditional probability function $\mathbb{P}(S_{t+1}|S_t, A_t)$ plays a key role, as it determines the probability distribution of a new state S_{t+1} based on the action A_t taken in state S_t. We assume the transition to the new state to depend only on the current state and action and not on the past: $\mathbb{P}(S_{t+1}|S_t, A_t) = \mathbb{P}(S_{t+1}|S_1, A_1 \cdots S_t, A_t)$, thus satisfying the Markov property. The initial state S_1 (which does not depend on any action) is assumed to have density distribution $\mathbb{P}_1$. The sequence of states and actions is such that once a state S_{t+1} is observed based on the action A_t, the decision maker receives a reward $\hat{r}_t = \hat{r}(S_t, A_t, S_{t+1})$. The reward is given by the wealth evolution over time, i.e, $\hat{r}_t = w(S_t)$ until the horizon T: thanks to rebalancing decisions the terminal wealth at time T will depend on the compounded returns generated over time by the adopted policy.

We consider a set of parameterized policies $\{\pi_\theta\}_{\theta \in \Omega}$, where a policy π_θ is defined as a function between states and actions $\pi_\theta : \mathcal{S} \to \mathcal{A}$, and the parameter θ is assumed to lie in a subset $\Omega \subseteq \mathbb{R}^I$. Under the given Markovian assumptions the parameter θ will thus induce a stationary policy π_θ independent of time. The adoption of the fixed-mix, constant proportion portfolio rule is then fully consistent with the RL rationale in this context. In the methodology, we assume that the set of actions A_t^θ taken by the decision maker (commonly called the agent) are induced by a policy π_θ

$$A_t^\theta = \pi_\theta(S_t). \tag{6}$$

The objective of the decision maker is to find the optimal θ which maximizes the utility $Q(W_T^\theta)$, also referred to as the action-value function and W^θ is computed as the rewarding process over the horizon $t = 1 \cdots T$ which is calculated assuming that the actions A_t^θ are induced by the policy π_θ.

Although the methodology works for general action value function Q, here next we will consider a Q function specified according to the objective in Eq. (1a). The derivation of the optimal policy $\pi_\theta^* = argmax_\theta J(\theta)$ with $J(\theta) = \mathbb{E}_{s_1 \sim \mathbb{P}_1}[Q(W_T^\theta)|S_1 = s_1]$, for given θ_0, is based on a stochastic gradient ascent method with $m = 0, 1, \ldots, M$ possible iterations:

$$\theta_{m+1} = \theta_m + \eta \nabla J(\theta_m). \tag{7}$$

The updating of θ_{m+1} in the iterative scheme requires the definition of the *learning rate* η and the estimate of the gradient $\nabla J(\theta_m)$. The computation of the gradient $\nabla J(\theta_m)$ is a relevant methodological issue in the theory of deterministic gradient policies (Zhang et al., 2020; Silver et al., 2014; Lillicrap et al., 2015), where it is derived as the expected action-value function evaluated at the current reward:

$$\nabla J(\theta) = \mathbb{E}_{s_1 \sim \mathbb{P}_1}[\nabla Q(W^\theta)|S_1 = s_1], \tag{8}$$

The computation of $\nabla Q(W^\theta)$ depends on the random wealth process W and its impact on $Q(W^\theta)$, typically over a high dimensional space. To tackle this numerical issue, the deep *Q-learning theory* proposes a methodology in which a *neural network* (NN) L_ϕ^θ, learns the action value function $Q(W^\theta)$, from information derived from both the states and the policy, π_θ. In this context, ϕ represents the weight parameters of the neural network, which are adjusted during the training process based on the provided training data.

When a generic action-value function Q is adopted as the expected value of a sum of stage rewards, shortly referred to as additive form, then it will satisfy the Bellman equation (Sutton and Barto, 2018), on which grounds an updating of ϕ in the NN L^θ_ϕ based on the minimization of the *Mean Squared Bellman Error* (MSBE). The extension of the action-value function Q from an additive form to more general, possibly concave (or convex), forms and its difficulties is the main subject in the theory of convex reinforcement learning (Zhang et al., 2020; Miryoosefi et al., 2019; Geist et al., 2021; Mutti et al., 2023,?).

In the context of Problem (1) the $Q-$ function is not in linear compounded returns form due to the penalty term associated with the ICVaR. In the case of generic action value functions Q to which the Bellman principle cannot be applied, a default updating of the gradient requires a specific approach. For the fixed-mix problem, we propose an updating rule for ϕ based on the convex reinforcement learning methodology by minimizing the error:

$$\min_\phi \|L^\theta_\phi - (Q(W^\theta) + L^\theta_{\hat{\phi}})/2\|^2_{L^2}, \tag{9}$$

where $\hat{\phi}$ is the previous parameter before updating. This criterion allows the NN L^θ_ϕ to learn by averaging between past knowledge $L^\theta_{\hat{\phi}}$ and new information coming from the environment $Q(W^\theta)$, thus *without forgetting*. The technique is based on the concept of *exploration–exploitation* described in Sutton and Barto (2018), Brandimarte (2021), Hazan et al. (2019), Črepinšek et al. (2013), Wei (2020) whose applicability is pretty general and beyond the specific portfolio problem considered here.

Following Eq. (9) we expect to *learn* the gradient $\nabla Q(W^\theta)$ from ∇L^θ_ϕ through the updating of parameter ϕ. The interaction between the parameters θ and ϕ reflects the *actor–critic* methodology proposed in Goodfellow et al. (2016), in which θ (the actor) and ϕ (the critic) learn dynamically from each other, as follows. Once the parameter ϕ of the NN L^θ_ϕ is updated, the algorithm updates the parameter θ through the gradient descent scheme, and once the parameter θ is updated, the neural network L^θ_ϕ uses this information to update ϕ. In the numerical implementation and through the iterative procedure, the gradient $\nabla_\theta L^\theta_\phi$ is used as a numerical approximation for the gradient $\nabla_\theta Q(R^\theta)$ of the action value function.

The Monte Carlo method is adopted to generate K scenarios $(s^k)^K_{k=1}$ used to train the NN L^θ_ϕ: k is a scenario label for a trajectory, or sample path of the portfolio compound reward process. The set of scenarios is generated following the methodology proposed in Ziemba and Ziemba (2008), Dupačová et al. (2000) as follows. For every scenario k and action following the policy π_θ, we generate recursively the realization s^k_{t+1} of scenario k in state S_{t+1}, by sampling from the transition probability $\mathbb{P}(S_{t+1}|S_t = s^k_t, A_t = \pi_\theta(s^k_t))$ for $t \geq 1$, where the initial state s^k_1 is sampled from the initial distribution $\mathbb{P}_1$. The branching structure of the scenario tree over T stages is denoted by $[n_1, n_2, n_3, \dots, n_T]$, where n_t defines the number of children nodes at the $t-$th stage. Thus gradient (8) is approximated by

$$\nabla J(\theta) \approx \frac{1}{K}\sum_{k=1}^{K} \nabla L^\theta_\phi(s^k). \tag{10}$$

The increasing path of the value function and convergence to 0 of the stochastic gradient, associated with the derivation of an optimal fixed-mix, are used to validate computationally the method, which is then tested on market data. The stopping criterion of the RL algorithm is based on a given tolerance on the gradient-decreasing norm or a maximum number of iterations. Throughout the learning process, it is anticipated that the gradient norm will decrease, while the action-value function will increase. Such behavior would signal an effective learning process. The convergence of the algorithm will also depend on the availability of an adequate amount of training data for the neural networks. Insufficient training data would increase the risk of lack of convergence.

We present evidence in Section 4 and in the Appendix to evaluate the convergence to optimality of the proposed methodology for the fixed-mix problem. We summarize in Algorithm 1 the pseudo-code of the RL methodology.

Algorithm 1: Reinforcement learning.

Input Initial distribution $\mathbb{P}_1$ of state (S_1), no. of iterations M, stages T, no. of trajectories K for exploration, batch size N, learning rate η and tolerance ϵ

```
1. Initialize the database $DB$ (space for storing input data and iterations' outputs)
2. Initialize the actor–critic parameters $\to \theta$ and $\to \phi$
3. For $m = 1 : M$ do for each iteration
    • Generate $K$ sample realizations $s^k_1$ from $\mathbb{P}_1$
    • For $t = 1 : T - 1$ do over the planning horizon
        – For each trajectory $k = 1, 2, ..., K$, along each scenario
        – Compute the action $a^k_t = \pi_\theta(s^k_t)$
        – Generate the next state $s^k_{t+1}$ from $\mathbb{P}(S_{t+1}|s^k_t, a^k_t)$
        – Store the transition points $(s^k_t, a^k_t, s^k_{t+1})$ in the database $DB$
    • end For $t$
    • Select a batch $\mathcal{B}$ of $N$ trajectories from $DB$
    • For each trajectory in the batch, compute $(s^k_t)^T_{t=1}$ and the reward function $\hat{r}(s^k_t, a^k_t, s^k_{t+1})$ for $t = 1, 2 \cdots, T$.
    • Using the sample points $s^k_t$ to approximate the action value function $Q(W^\theta)$
    • Using the critic neural network $L$, compute $y_k = (Q(W^\theta) + L^\theta_\phi(s^k))/2$
    • Update the critic neural network parameter $\phi$ by minimizing
          $\phi \leftarrow \arg\min_\phi \frac{1}{N}\sum_{k\in\mathcal{B}} \|y_k - L^\theta_\phi(s^k)\|^2$
    • Compute the gradient approximation: $\nabla J \approx \frac{1}{N}\sum_{k\in\mathcal{B}} \nabla_\theta L^\theta_\phi(s^k)$
    • Update $\theta$: $\theta \leftarrow \theta + \eta\nabla J$
    • If $\|\nabla J\| \leq \epsilon$ the algorithm stops (stopping criterion)
4. end For $m$
```

Output Optimal parameter θ, optimal policy π_θ and optimal value function $J(\theta)$.

We analyze the computational complexity of this algorithm in Section 3.3. In Section 4, we present in-sample and out-of-sample comparative results of Algorithm 1 versus the RL method developed by Das and Varma (2020) to address a *Goal-based wealth management* (GBWM) problem, with a software which is available in the Financial Toolbox of Matlab (Matlab, 2020).

3.2. RL-based FxM problem specification

Algorithm 1 is applied to solve problem (1a). The following Table 3 helps understand the correspondence between the RL parameters and the specific financial optimization problem.

For $t = 1, 2, \dots, T$ we have:

- The state s_t of the system is determined by the portfolio allocation in $t-1$ and available cash $x_{0,t-1}$ and the realized returns $r_{i,t}$ of the assets.
- The action a_t includes non negative selling and buying decisions $x^-_{i,t}$ and $x^+_{i,t}$.

Table 3
Summary information on the correspondence between fixed-mix portfolio problem formulation and RL parameters.

Parameter	Description
State s_t	$s_t = (x_{0,t-1}, x_{1,t-1}, \dots, x_{I,t-1}, r_{1,t}, \dots, r_{I,t})$, $x_{t-1} = (x_{0,t-1}, x_{1,t-1}, \dots, x_{I,t-1})$ portfolio allocation $r_t = (r_{1,t}, \dots, r_{I,t})$ return vector
Action a_t	$a_t = (x^+_{1,t}, \dots, x^+_{I,t}, x^-_{1,t}, \dots, x^-_{I,t})$ $x^-_{i,t}$ buying of the ith asset at time t $x^+_{i,t}$ selling of the ith asset at time t
Conditional probability $\mathbb{P}$	$\mathcal{N}(\mu, \Sigma)$ Multivariate normal distribution with mean μ and covariance matrix Σ
Policy parameterization θ	$\theta = (\theta^1, \dots, \theta^I) \in \mathbb{R}^n_+$ fixed-mix policy $\theta_i = \frac{x_{i,t}}{\sum_i x_{i,t}}$
Action-value function $Q(W^\theta_T)$	$Q(W^\theta_T) = \mathbb{E}\left[W^\theta_T\right] + [\rho_{\alpha,\beta}(W^\theta_T) - \rho_{\alpha,\beta}(Y_T)]_-$ with W^θ_T the wealth at $t = T$, Y_T benchmark in T $\rho_{\alpha,\beta}$ the ICVaR risk measure.

- The transition between states and the random environment is characterized by the random returns $r_{i,t}$ through

$$x_{i,t} = x_{i,t-1}(1 + r_{i,t}) + x^+_{i,t} - x^-_{i,t} \tag{11}$$

and for the cash:

$$0 \leq x_{0,t-1} + \sum_{i=1}^{n} (1 - c_s)x^-_{i,t} - (1 + c_b)x^+_{i,t}, \tag{12}$$

where c_s and c_b are transaction cost coefficients.
- Through $\theta = (\theta_1, \dots, \theta_I) \in \mathbb{R}^I_+$ we derive a policy $\pi_\theta(s_t)$ in state s_t so that, after rebalancing, $\theta_i = \frac{x_{i,t}}{\sum_i x_{i,t}}$ at the end of every period. Depending on the return of an asset relative to the portfolio, this policy will induce a selling or buying decision in state t according to the fixed-mix. With $\mathbf{r}_t = \{r_{i,t}\}_{i=1}^I$ multivariate normal with mean vector μ and covariance matrix Σ.
- Finally the action-value function $Q(W^\theta_T)$ has been discussed extensively and it does include the reward $\mathbb{E}[W^\theta_T]$ and ICVaR functions $\rho_{\alpha,\beta}(W_T)$ and $\rho_{\alpha,\beta}(Y_T)$ associated with the portfolio value evolution and the adopted benchmark, respectively.

3.3. *Complexity analysis and limitations of Algorithm 1*

We analyze the computational complexity of Algorithm 1. We assume that the states s and policies θ are in Euclidean spaces $\mathbb{R}^I$ and $\mathbb{R}^{\bar{I}}$, respectively, where I corresponds to the number of assets and $\bar{I}$ to the set of possible policies. In Algorithm 1, we perform a maximum of M iterations. Each iteration involves several steps. Firstly, we generate K trajectories over T stages, with a computational complexity of at least $O(I \times T \times K)$. Secondly, we compute the reward r and the action value function $Q(W^\theta)(s^k)$ for a batch of N trajectories s^k, with a complexity of $O(T \times N)$. The next step of the algorithm involves updating the critic neural network ϕ. For the FxM problem, we opt for the Levenberg–Marquardt method for training, thanks to its fast and stable convergence, as reported by Yu and Wilamowski (2018), Wilamowski and Yu (2010). The computational complexity for training the critic network ϕ using the Levenberg–Marquardt algorithm is $O(\text{PNNc}^3 \times N^2)$, where PNNc represents the number of weights and biases in the architecture of the neural network ϕ. This complexity arises from the computation involving the inverse of the perturbation of the Jacobian matrix, accounting for the size of batching points and the weights and biases to be updated in the neural network. The final step in each iteration involves updating the actor neural network θ using the gradient descent method, with a computational cost of $O(\bar{I} \times N)$ (Bottou, 2010; Ruder, 2016). Therefore, for each iteration, the computational complexity sums up to $O(\text{PNNc}^3 \times N^2 + I \times T \times K + (\bar{I} + T) \times N)$. Based on a maximum of M iterations, the minimum computational complexity of Algorithm 1 is at least $O(K \times (\text{PNNc}^3 \times N^2 + I \times T \times K + (\bar{I} + T) \times N))$.

This assessment helps understand the limitations of the Algorithm. The computational complexity is at least cubic in the dimension of parameters involving the neural network ϕ and quadratic in the size of the batching point N, as well as multilinear in the parameters $(I, \bar{I}, T, K)$. Therefore, a high number of layers and neurons, along with a large number of batching points N, will primarily limit the computational performance of the algorithm. Similarly, a large number of stages T, high dimensions of both states I and policies $\bar{I}$, and a large number of trajectories K will also contribute to this limitation.

In Section 4, we select $T = 4$ as number of stages and consider an asset universe of 8 assets, relatively low but sufficiently realistic to address a genuine FxM portfolio problem. Potential limitations may then emerge from employing a large number of layers and neurons in the architecture of the critic neural network ϕ, as well as using a large batching size N and number of trajectories K for training the actor neural network θ.

4. Computational evidence

In this section, we analyze the main implications of the RL method described in Algorithm 1 for portfolio selection under a fixed-mix policy. We consider in the following subsections:

4.1 The definition of the dataset adopted for the validation of the methodology and anticipate the rationale and experimental settings of the analyses conducted in Sections 4.2–4.4. At the beginning of this section we provide in Fig. 1 the architectural design of the critic's NN..

4.2 The evidence emerges from the solution of one instance of a 4-stage dynamic fixed-mix problem with a 1-month horizon and weekly rebalancing.

4.3 A more extended analysis, spanning several problems detailed next, associated with problem instances based on different planning horizons and rebalancing frequency. We consider: the standard reference model with a 1-month short-term horizon with weekly rebalancing with results spanning from January 2021 to June 2023. A 3-month problem with monthly rebalancing and results spanning from January 2019 to June 2023. Then a 9-months and an 18-months sequence of fixed-mix problems with quarterly and semi-annual rebalancing, spanning respectively from January 2017 and from January 2015 to June 2023. The collected evidence aims at verifying the stability of the method under different problem specifications and generate comparative evidence for the RL method considered in the GBWM problem. Further to the core and summary evidence presented in this section, we provide in the Appendix a more extended set of financial and graphical evidence.

4.4 All instances above and related optimal solutions over the given timeframes are considered in the out-of-sample analysis to verify their effectiveness from market data.

In-sample validation aims at verifying the consistency of the collected results in relationship with key modeling assumptions, specifically in our case related to the numerical convergence of the stochastic gradient, the behavior of the problem objective value, and the post-optimality stochastic dominance relationship that can be established between the portfolio value and the benchmark cumulative distributions. The computational evidence will include benchmarking the RL method presented in this paper against the RL methodology developed to solve a dynamic Goal-Based Wealth Management (GBWM) problem studied by Das and Varma (2020). The GBWM method is implemented in Matlab and accessible in the Financial Toolbox of Matlab (Matlab, 2020).

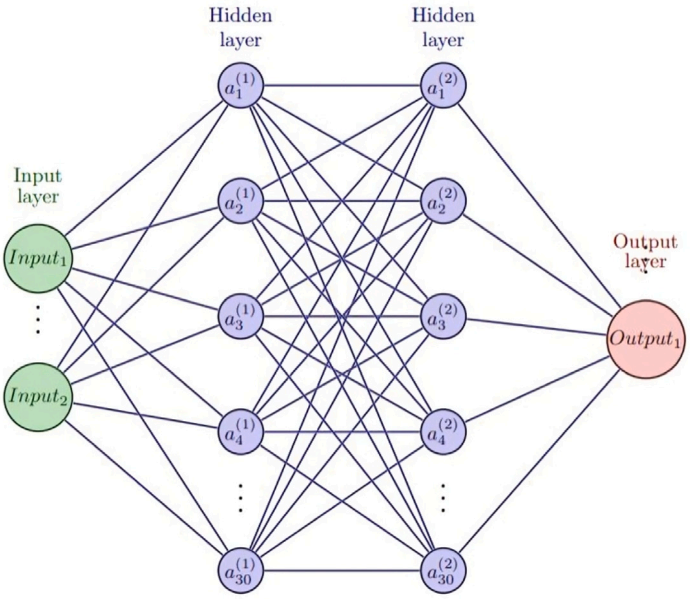


Fig. 1. The architectural design of the critic's neural network. Where $Input_1$ represents the state s while $Input_2$ corresponds to the policy θ. The output $Output_1$ signifies the value function $Q(W_T^\theta)$. And, $a_k^{(i)}$ denotes the kth neuron in the ith hidden layer.

Table 4
Parameters' settings in the RL procedure.

Parameter	Initial value	Comment
$w(s_0)$	1	Initial wealth
β	$\{0.98, 1\}$	Reference point, left portfolio tail $\leq \beta$
α	0.95	Tolerance for the ICVaR function
M	100	Max number of iterations
T	4	Number of stages
K	80 000	Number of trajectories
Tree	[40, 20, 10, 10]	Branching degree
N	2000	Batch size
η	0.2	Learning rate
θ_0	Equally-Weighted	Input fixed-mix
ϵ	10^{-3}	Stopping Criterion

4.1. Numerical setting

The evidence reported in this section relies on a 2-hidden-layer recurrent neural network, with 30 neurons in each layer, which represents the critic network L_ϕ^θ, whose input is state s^k and policy θ and output is the action value function $Q(W_T^\theta)$. The graphical representation of the *neural network* (NN) architecture is provided in Fig. 1. The NN is trained using the Levenberg–Marquardt algorithm (Moré, 2006; Yu and Wilamowski, 2018; Wilamowski and Yu, 2010).

The sample-based estimation of the stochastic gradient relies on a finite difference scheme over the policy θ. The implementation of several executions was done in Matlab Online a service provided in a cloud server. The Algorithm 1 required about 94.40 min to complete.

The parameters used in Algorithm 1 are summarized in Table 4.

Some of the parameters in Table 4 play a relevant role in the learning process and have indeed been subject to extensive calibration. We refer to Algorithm 1 for further insights: M refers to the maximum number of macro-iterations to derive the optimal fixed-mix θ. Then we expect no more than these many iterations for the stochastic gradient to get *sufficiently close* to 0. Every iteration includes the estimation of the transition probabilities based on $K = 80\,000$ scenarios with associated sequential updates of states, actions, and subsequent reward and value function estimation based on a batch of $N = 2000$ trajectories. The definition of $T = 4$ carries both financial and numerical implications. As for the latter, jointly with M and K, we show in Section 4.2 that 4 periods are sufficient to calibrate the RL method and assess its convergence. As for the former, it is worth remarking that FxM models have been considered primarily in medium-long term portfolio problems (Fleten et al., 2002; Dempster et al., 2011): here, without claiming a general result, we show however that such a very short-term planning horizon is sufficient to collect effective performance results. The extent of the investment period and the specific type of FxM problem also constrain the number of assets in the case study.

For the standard problem we rely on weekly data from January 2018 to June 2023 and take the first 3 years, until December 2020 to compute the mean μ and covariance matrix Σ and derive the transition probabilities. These are then used to generate training data as input to compute the optimal fixed-mix policy over the following 4 weeks. In these experiments, transaction costs are not considered. For more extended planning horizons we always use monthly data and determine the input statistics relying on the past 36 months to compute the mean and variance, respectively.

The decision space of the problem is defined by the following assets or investment opportunities, *exchange traded funds* (ETF): (a) four SP500 sub-sectors for Energy (XLE), Finance (XLF), Technology (XLK) and Industry (XLI); (b) the ETF tracking 7–10 year maturity US Treasury bonds (IEF), the ETF for Gold commodity (GLD), to represent an anti-cyclical real asset typically negatively correlated with equity markets and, finally an ETF tracking the performance of the dollar against a set of other currencies (USDU).

The benchmark strategy Y is represented by the $S\&P500$, whose distribution the portfolio manager intends to dominate. To this latter aim, the two sets (a) and (b) include the former those assets that help *replicating* the benchmark and the latter those ETFs that help diversify the portfolio and potentially attain positive outcomes during negative equity market phases. We thus verify whether the proposed methodology may lead to an optimal fixed-mix policy able through the implied ISD-2 conditions to outperform the benchmark.

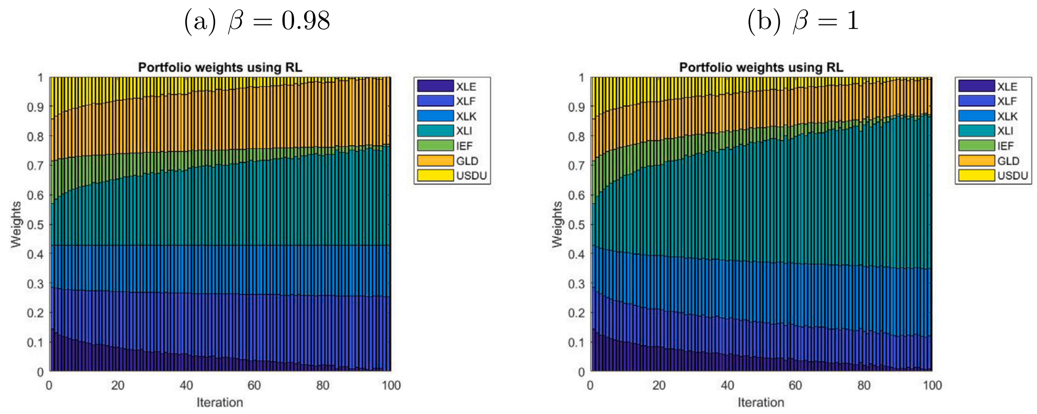
The evidence in Table 5 provides a general assessment of the statistical properties of the assets the portfolio manager may rely upon to outperform the benchmark: notice, in particular, the assets' Sharpe ratios, computed by dividing the mean return to the standard deviations (or asset's volatility), which, consistently with finance theory, shows the good performance of the benchmark relative to many of the assets in the decision space. Table 6 completes this preliminary data analysis

Table 5
Statistics of assets' weekly returns from 07/01/2018 to 30/06/2023.

	SP500	XLE	XLF	XLK	XLI	IEF	GLD	USDU
Mean %	0.07	0.05	0.04	0.13	0.05	−0.01	0.05	0
Max %	12.06	20.08	14.1	12.66	15.92	4.85	6.82	4.48
Min %	−12.39	−24.22	−15.87	−12.09	−15.71	−2.79	−13.27	−5.43
Std %	2.69	5.04	3.51	3.35	3.22	0.91	2	1.01
Skewness	−0.38	−0.47	−0.48	−0.27	−0.25	0.63	−0.84	−0.3
Kurtosis	6.67	6.39	7.75	4.51	8.51	7.19	10.23	8.11
Sharpe ratio %	2.53	1.01	1.1	3.93	1.65	−0.87	2.44	0.44

Table 6
Correlation matrix of weekly return rates of the asset universe in 07/01/2018 to 30/06/2023, weekly data.

	SP500	XLE	XLF	XLK	XLI	IEF	GLD	USDU
SP500	1	0.61	0.86	0.92	0.9	−0.1	0.18	−0.4
XLE	0.61	1	0.72	0.42	0.72	−0.3	0.09	−0.24
XLF	0.86	0.72	1	0.66	0.89	−0.28	0.1	−0.39
XLK	0.92	0.42	0.66	1	0.74	−0.04	0.15	−0.31
XLI	0.9	0.72	0.89	0.74	1	−0.15	0.16	−0.44
IEF	−0.1	−0.3	−0.28	−0.04	−0.15	1	0.39	−0.25
GLD	0.18	0.09	0.1	0.15	0.16	0.39	1	−0.45
USDU	−0.4	−0.24	−0.39	−0.31	−0.44	−0.25	−0.45	1


(a) $\beta = 0.98$

(b) $\beta = 1$

Fig. 2. Plot of the learning policy along the iterations.

by displaying the estimated sample correlations from 07/01/2018 to 30/06/2023.

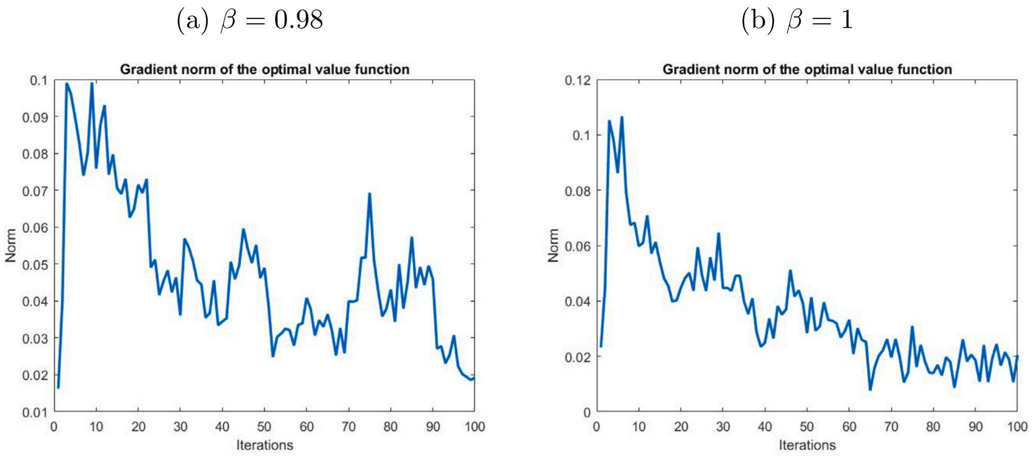
4.2. Method validation: one problem instance

We consider in this section only one instance of an optimal portfolio problem defined at the beginning of December 2022 (from 04/12/2022 until 25/12/2022) to collect qualitative information on the proposed methodology when applied to a single 4-stage problem. For $\beta = \{0.98, 1\}$, we analyze a set of results including the convergence to optimality of the RL methodology for the fixed-mix portfolio and some key statistical evidence. The purpose of this section is primarily to convey on which qualitative evidence we rely to validate the methodology. A more detailed set of evidence is presented in Appendix A. In the following section, the same analyses are applied to sequences of problems to verify both their methodological and financial consistency. In Figs. 2–4, we display, respectively: the evolution of the portfolio composition through the iterations of the learning process, the associated gradient norm and optimal value J over the 100 iterations assumed in this test. For a fixed set of parameters, we see that the norm of the stochastic gradient decreases and the optimal value function increases at every iteration. Evidence suggests the convergence of the learning process toward an optimal fixed-mix policy, as illustrated in Fig. 2.

Fig. 2 shows the evolving fixed-mix policy determined by the learning process at every iteration $m = 1, 2, \dots, 100$. The optimal constant proportion evolves and we consider as optimal fix-mix the one determined at $m = 100$. We see indeed that the gradient's $10^{-}3$ stopping criterion wasn't met, as clear from the evidence in Fig. 3. For $\beta = 1$, the stochastic gradient after the initial increase is declining consistently. Not the same for $\beta = 0.98$, which by definition is deeper in the tail. Taking also the evidence in Fig. 4 into account, we see however, that the gradient instability occurs after the marginal increase of the value function starts declining to 0 very rapidly.

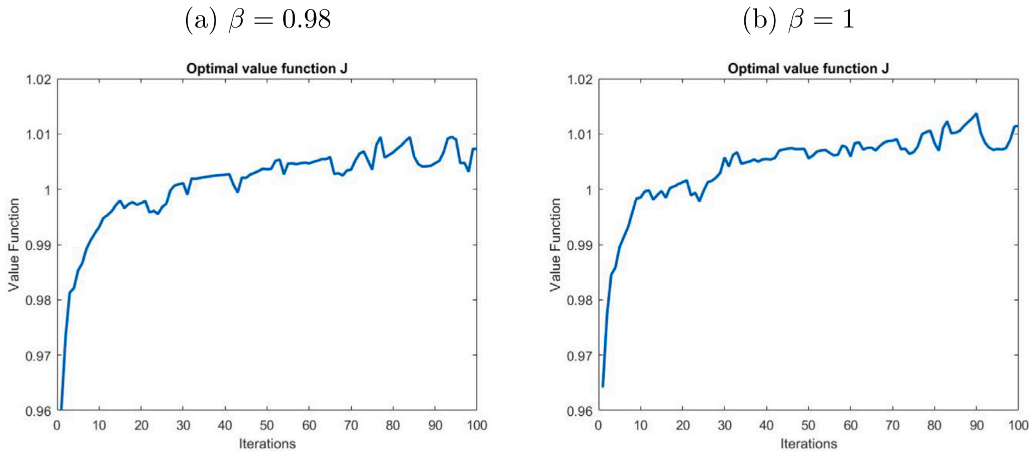
Given the initial investment of \$1, from a financial viewpoint, it is interesting to verify the expected portfolio value at the end of the first period and then in $T = 4$. Those values are averaged over the set of trajectories of the learning process and after the algorithm's termination, we compute the statistics shown in Table 7. The first section of the table shows evidence at the end of the first week W_1, the second at the end of the fourth week W_4, and the last one at the bottom, the average results over the 4 weeks $\bar{W}_4$.

The evidence in Table 7 confirms the improvement of the financial performance (see the expected wealth) and the risk control (volatility and risk-adjusted returns) induced, *ceteris paribus*, by an increase of the number of stages. Furthermore, a higher β is beneficial to attain a more effective control in the tail. For $\beta = 1$ we observe that upon termination of the RL iterations, the portfolio distribution does dominate the benchmark distribution with order ISD-1, stronger than SSD. This evidence is consistent with the proposed ICVaR minimization objective. Appendix A presents the same set of evidence for single instances of FxM problems with 3, 9, and 18 months horizons. We avoid including here such an extended set of graphical results, which are in any case taken into account in Section 4.3 here next, but just observe that the convergence results of the gradient and value functions are


(a) $\beta = 0.98$

(b) $\beta = 1$

Fig. 3. Plot of the gradient norm along the several iterations.


(a) $\beta = 0.98$

(b) $\beta = 1$

Fig. 4. Plot of the value function J along the iterations.

Table 7
In-sample statistics of the RL solution, with a specific focus on the contrast between first and the last-stage fixed-mixes performance. Here a 1-month horizon problem is considered with weekly rebalancing from 04/12/2022 until 25/12/2022.

State	β	0.98	1
First	$\mathrm{E}(W_1)$	1.0015	1.001
	$\sigma(W_1)$	2.757	0.881
	$\mathrm{SR}(W_1)$	0.361	1.137
	$\mathrm{CVaR}_{0.95}(W_1)$	0.9507	0.9524
	ISD-$k.q_\beta(T = 1)$	1.9249	1.9499
Terminal	$\mathrm{E}(W_4)$	1.008	1.0161
	$\sigma(W_4)$	1.286	0.849
	$\mathrm{SR}(W_4)$	0.7838	1.1968
	$\mathrm{CVaR}_{0.95}(W_4)$	0.9746	0.9764
	ISD-$k.q_\beta(T = 4)$	1.9999	1.7080
Average	$\mathrm{E}(\bar{W}_4)$	1.002	1.003
	$\sigma(\bar{W}_4)$	1.644	1.644
	$\mathrm{SR}(\bar{W}_4)$	0.608	0.608
	$\mathrm{CVaR}_{0.95}(\bar{W}_4)$	0.9665	0.9693

confirmed when extending the investment horizon, while the optimal FxM policy becomes less diversified. We come back to this aspect in Section 4.4.

4.3. Method validation over several instances

We extend the analysis to span different investment horizons and rebalancing frequencies. We solve sequentially FxM problems based on:

- 1 month planning horizon, weekly rebalancing, spanning from Jan 2021 to June 2023, resulting in 30 problem instances generated through a rolling windows approach with monthly steps; using training data from January 2018 to June 2023.
- 3 months horizon, monthly rebalancing, spanning from Jan 2019 to June 2023, resulting in 18 problem instances generated through rolling windows with quarterly steps; using training data from January 2006 to June 2023.
- 9 months planning horizon, quarterly rebalancing, spanning from January 2017 to June 2023 for 8 problem instances again based on rolling windows; using training data from January 2000 to June 2023.
- 18 months planning horizon, semi-annual rebalancing, spanning from January 2015 to June 2023 resulting in 5 instances with 1 and half year updates of the rolling windows and training data from January 2000 to June 2023.

Depending on the problem instance, by moving forward the training process with monthly, quarterly, or semiannual steps, we solve the set of 1-, 3-, 9- and 18-month horizon problems and collect a set of evidence. The convergence and the performance of the training process are analyzed in Fig. 5, always based on 100 iterations of the training algorithm with reference points $\beta \in \{0.98, 1\}$. We present additional information on the convergence of the method in Appendix A.3.

Fig. 5 shows from left to right and top to bottom, the average behavior of the gradient norm and optimal value function for every problem type taking all the collected solutions into account over the

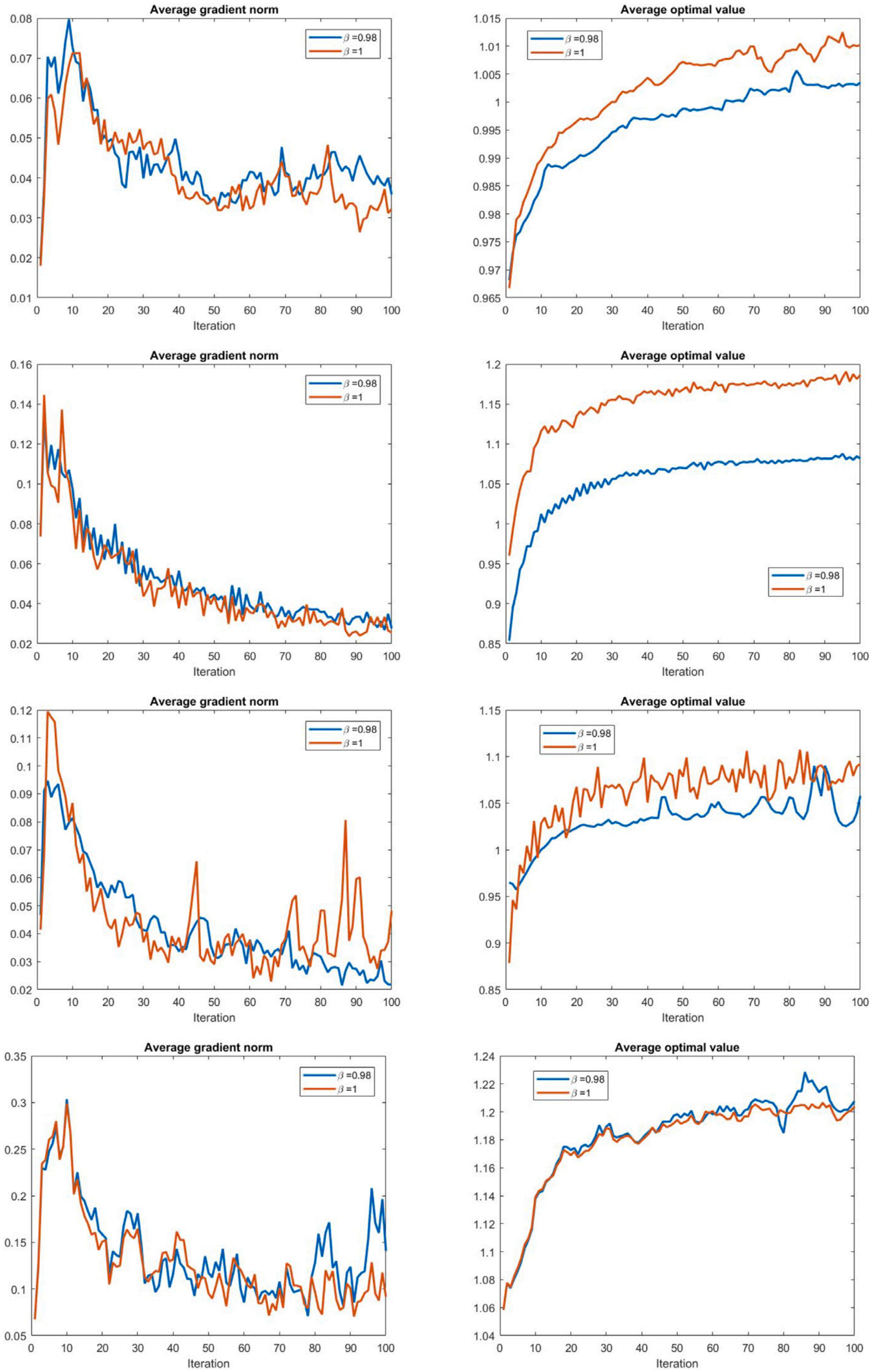


Fig. 5. Gradient norm (left column) and optimal value functions (right column) average behavior over:- The first row: 1 month horizon, (average over 30 instances), Jan 2021 to June 2023;- Second: 3 months horizon (average over 18 instances) from Jan 2019 to June 2023;- Third: 9 months horizon (average over 8 instances) from Jan 2017 to June 2023;- Fourth row: 18 months horizon (average over 5 instances) from Jan 2015 to June 2023, for $\beta \in \{0.98, 1\}$.

common 100 iterations. On the left column for $\beta = 0.98$ and the right for $\beta = 1$. The first pair on the top is collected from the solution of the 1-month horizon problems, then the 3-month problem in the second row, then the 9-month and 18-month problems' solutions at the bottom. We report in the Appendix the underlying evidence generated by each problem solution.

Fig. 5 provides the information we may rely upon to verify the consistency of the results when solving a sequence of FxM problems with different planning horizons. We provide the underlying evidence for every solved problem in Appendix A, while here we focus on average dynamics. Next to Fig. 5, we report on Table 8 associated relevant financial and statistical evidence.

As for the reference 1-month horizon problem, under the settings in Table 4, we observe that taking all 30 instances into account the increasing then decreasing gradient paths and consistently increasing value functions hold almost always with a moderate anomaly after 80 iterations in the case $\beta = 0.98$. When extending the horizon to 3 months, the convergence results remain robust across all runs and the evidence on the value function behavior as we decrease the β is confirmed. As the planning horizon increases the results are less consistent. In the

Table 8
Average comparative in-sample monthly statistics for each planning horizon from 1- to 18-month problems: the ICVaR-based RL method versus the GBWM results.

Planning horizon		$\beta = 0.98$	$\beta = 1$	$GBWM$
$1 - Month$	Mean %	1.0418	1.054	1.295
	Std %	5.604	5.347	5.35
	Skewness	0.166	0.054	0.212
	Kurtosis	2.896	2.909	3.463
	Sharpe ratio %	19.0	19.7	25.0
	ISD-order	1.803	1.697	1.92
$3 - Months$	Mean %	1.053	1.152	1.054
	Std %	2.767	2.164	2.831
	Skewness	1.008	0.851	0.269
	Kurtosis	4.183	3.803	3.303
	Sharpe ratio %	38.06	53.23	42.8
	ISD-order	2	1.944	2.105
$9 - Months$	Mean %	0.3527	0.3640	0.888
	Std %	1.440	1.2213	1.954
	Skewness	0.005	−0.023	0.29
	Kurtosis	2.96	2.925	3.126
	Sharpe ratio %	24.49	29.80	48.32
	ISD-order	1.601	1.61	2.154
$18 - Months$	Mean %	0.2035	0.2008	0.565
	Std %	0.8940	0.8660	0.867
	Skewness	0.149	0.14	0.337
	Kurtosis	2.881	2.891	3.151
	Sharpe ratio %	22.76	23.19	65.8
	ISD-order	1.612	1.582	1.674

Table 9
Average expected time and convergence depending on both the branching tree structure and the sample batching size, denoted as N.

Avg Time	Convergence	Periods	Branching	N	Iter. limit
110’24”	No	5	[40 20 10 10 10]	20 000	0
116’26”	No	5	[40 20 10 10 5]	10 000	4
116’18”	No	5	[40 20 10 10 4]	8000	5
120’29”	No	5	[40 20 10 10 3]	6000	11
129’38”	No	5	[40 20 10 10 2]	4000	25
94’40”	Yes	4	[40 20 10 10]	2000	100
21’13”	Yes	3	[40 20 10]	200	100

following Table 8 we summarize the statistics and financial evidence associated with every set of solutions and compare it with the results produced with the financial toolbox of Matlab, labeled GBWM.

The results in Table 8 are all expressed on a monthly basis and comparable. The statistics are averages over the set of runs for each problem instance.

We summarize here below the main evidence of this in-sample analysis. Different perspectives can be adopted for this purpose. From a modeling viewpoint, one expects the solution to generate relatively stable optimal FxM policies across time, the enforcement of the dominance relationship relative to the benchmark, and a slight discrepancy of the $\beta = 1$ versus $\beta = 0.98$ instances. From a methodological perspective, as discussed above, we expect primarily a stable convergence of the gradient norm and the value functions.

- The evolution of the optimal FxM policies is shown in Section 4.4 and a good level of diversification is shown over time and, supported by the evidence in Appendix A, we confirm a good convergence to diversified portfolios particularly for $T = 1, 3$ months.
- The stochastic dominance for the benchmark distribution is systematically less or equal to 2. Often below. The specification of the reward function is consistent with the enforcement of the mathematical result.
- For $\beta = 1$, the proposed RL model exhibits slightly lower volatility, measured by the standard deviation (Std), compared to RL with $\beta = 0.98$ and the GBWM method. Therefore, using $\beta = 1$ in RL provides an interesting parameter for evaluating the methodology in an out-of-sample analysis
- As for the gradient decreasing pattern and increasing value functions in Fig. 5, we refer to the comments above. As the rebalancing frequency decreases and the investment horizon increases the convergence tends to worsen: when the value function stabilizes and so does the gradient, we also see a well-diversified optimal FxM policy.

We complete this section by sharing the numerical evidence collected through this extensive numerical exercise. As detailed in Table 9, we realized that the computational time to run the RL Algorithm 1 is influenced solely by the variables M, T, K, Branching Tree, N, and ϵ. To ensure convergence and prevent infinite loops, we keep the parameters M and ϵ fixed as stopping criteria. However, the batching size parameters T and K are determined based on the Branching Tree. Notably, when T, K, and N are large, the algorithm encounters infeasibility issues and fails to converge effectively as examined in Section 3.3.

In comparison to our algorithm, the Matlab GBWM algorithm shows much faster convergence in seconds. Surely, we need to attain a substantial complexity reduction of Algorithm 1.

The computational evidence in Table 9 shows the current clear limitations of the algorithm, in the absence of any dimensional reduction technique. Nonetheless, it shouldn’t be underestimated the relevant implications, in the case of 4 stages, of the extremely high set of scenarios, namely 80,000, aimed primarily at spanning the state space of the problem with high accuracy. When reducing the number of stages the convergence is reduced significantly, yet by several minutes. An important related remark is that it is thanks to the rich scenario branching adopted in the RL algorithm that we can assess precisely the evolution of the stochastic dominance against the benchmark distribution. As further commented in the conclusion, a more efficient and fast converging methodology is among the next research tasks.

4.4. Optimal policy back-testing

We now present the evidence relative to the actual market performance that would have been generated by the adoption of the optimal Fx strategies, and output of the RL method. Depending on the investment horizon, from the reference 1-month horizon to 18 months, we verify the ex-post performance of the FxM strategies.

Table 10 provides a comprehensive set of comparative evidence collected over the out-of-sample periods: for each time horizon, we display top to bottom the average monthly return, the standard deviation, the loss associated with the Conditional Value-at-Risk at 95%, the Sharpe ratio (mean return per unit standard dev) the percentage of weeks in which the SP500 has been outperformed, the positive and negative excess returns to the benchmark.

The evidence has primarily a financial and economic rationale when comparing column-wise the results of different strategies depending on the planning horizon and row-wise when assessing their consistency across different horizons. In this same section, we focus below on the representative 1-month horizon on which the RL method has been primarily calibrated. From Table 10 is not easy to derive general and robust evidence, surely the $\beta = 1$ RL solutions do generate consistently across the different horizons very good risk-adjusted performances. We provide more details, below.

We highlight the main results of this out-of-sample analysis.

- The RL methodology outperforms the GBWM, 1/N portfolio, and 60%–40% strategy over 1-month, 9-month, and 18-month planning horizons. This is shown by comparing the Sharpe ratio and metrics like WR, $E(ER)+$, and $E(ER)-$.

Table 10
Average out-of-sample monthly returns over increasing planning horizons. Optimal strategies resulting from the RL model, $1/N$, the 60%–40% equity-bond strategy, and multiperiod goal-based wealth management (GBWM) versus the SP500. The first column is the time horizon in months.

Horizon		RL $\beta = 0.98$	RL $\beta = 1$	$1/N$	$60\% - 40\%$	$GBWM$	SP500
	Mean %	0.53	1.22	0.61	0.47	0.04	0.55
	Std %	9.38	8.47	6.06	5.32	6.66	9.39
	CVaR	−4.92	−4.61	−3.6	−3.13	−3.94	−5.63
$1 - m$	Sharpe ratio %	5.6	14.37	10.13	8.78	0.55	5.87
	WR (%)	45.74	53.49	41.86	43.41	45.74	–
	$E(ER)_+$ (%)	5.25	5.06	4.94	4.94	3.65	0
	$E(ER)_-$ (%)	−4.47	−4.39	−3.45	−3.94	−4.03	0
	Mean %	0.93	0.76	0.81	0.75	1.19	1.17
	Std %	4.63	4.22	3.83	3.38	4.72	5.65
	CVaR	−10.73	−10.04	−8.44	−7.12	−9.76	−12.04
$3 - mm$	Sharpe ratio %	20	17.97	21.27	22.22	25.1	20.63
	WR (%)	46.30	33.33	35.19	35.19	46.30	–
	$E(ER)_+$ (%)	1.24	2.13	1.97	2.36	2.09	0
	$E(ER)_-$ (%)	−1.51	−1.67	−1.61	−1.92	−1.76	0
	Mean %	0.76	0.93	0.73	0.66	0.49	1.05
	Std %	2.95	2.27	2.1	1.82	1.42	3.37
	CVaR	−24.44	−14.66	−11.49	−9.93	−7.56	−21.87
$9 - mm$	Sharpe ratio %	25.8	41.07	34.7	36.37	34.62	31.15
	WR (%)	42.31	42.31	34.62	30.77	38.46	–
	$E(ER)_+$ (%)	1.04	2.05	1.14	1.55	1.83	0
	$E(ER)_-$ (%)	−1.26	−1.71	−1.10	−1.25	−2.05	0
	Mean %	0.75	0.75	0.64	0.57	0.38	0.86
	Std %	1.8	1.61	1.25	1.12	1.08	1.8
	CVaR	−14.81	−8.25	−13.74	−12.41	−9.14	−20.94
$18 - mm$	Sharpe ratio %	41.72	46.89	50.74	50.9	34.76	47.71
	WR (%)	52.94	52.94	29.41	29.41	29.41	–
	$E(ER)_+$ (%)	0.99	0.96	0.61	0.81	1.26	0
	$E(ER)_-$ (%)	−1.34	−1.31	−0.57	−0.74	−1.21	0

- For the 3-month planning horizon, the GBWM outperforms the proposed methodology. The proposed fixed-mix problem achieves second-order, but not first-order stochastic dominance. This suggests that the out-sample performance of the proposed RL methodology is influenced by first-order stochastic dominance. Further studies are needed to determine when the proposed RL algorithm can achieve both second-order and first-order stochastic dominance, which would improve the performance of the proposed fixed-mix problem
- When the time horizon is increased from 1 month to 18 months, the volatility decreases for the RL model, as indicated by the standard deviation (Std). This confirms the assumption made in the in-sample analysis that increasing the time horizon leads to a significant decrease in volatility.

The statistics in Table 10 are computed from the extended set of solutions included in Appendix B, to which we refer for further details. Here next, focusing only on the reference FxM problem with monthly investment horizon, we share the sequence of optimal strategies adopted in the out-of-sample period and comparative dynamics of a 100 USD investment from January 2021 to the end of June 2023.

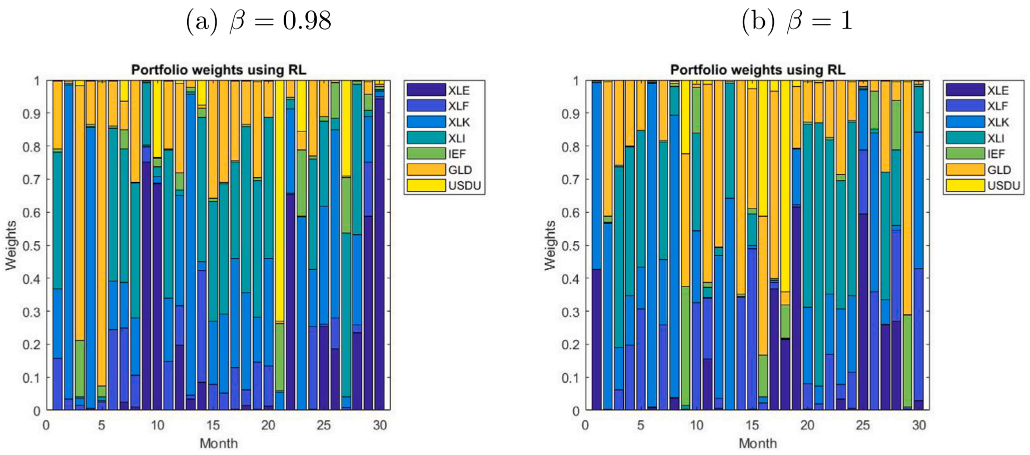
The optimal portfolio allocations over the 30 months are displayed in Fig. 6, while the back-testing wealth process is illustrated in Fig. 7.

The evidence from Fig. 6 is of a sequence of optimal FxM strategies, that without imposing any policy constraints, thus any bounds on the investment proportions, changes over time and preserves very good diversification properties. We present similar evidence for the 3-, 9- and 18-months problems in Appendix B. The sequence of optimal FxM generated by the algorithm varies with the market phase and all show good ex-post performance relative to the $SP500$. We noticed however that in particular, the 18-month RL case problem did not always converge. The optimal FxM policies are benchmarked against the $SP500$, and two popular heuristic rules: the $1/N$ perfectly diversified strategy (take $N = I$ the number of asset classes considered in the problem), and the 60%–40% equity-bond composition based the former on all equity ETF and the latter on the remaining asset classes. We also rely on Matlab's GBWM RL software as a benchmark to evaluate the proposed algorithm.

5. Conclusion and future directions

We conclude by summarizing the key evidence presented in this article, primarily for the developed RL methodology and the solution to the financial optimization problem. We have initially motivated the modeling framework adopted to tackle the FxM problem, relevant in financial practice and that over the years has attracted interest in the area of quantitative finance. In this work, the FxM investment paradigm has been extended to accommodate a tail risk measure with the potential to improve downside risk control but also to bring in the concept of stochastic dominance relative to an exogenous benchmark policy. We propose a solution approach based on a reinforcement learning methodology, tested over several years and under different time frames. The collected evidence allows some concluding remarks:

- The adopted action-value function in Eq. (5) and error minimization in Eq. (9) lead consistently to convergence of the RL algorithm 1 for 1-, 3- and 9-months problems: the evidence is slightly different when considering the $\beta = 1$ and $\beta = 0.98$ cases. The algorithm has difficulties converging in the 18-month problem. As a result, we highlight the dependence of the convergence on the problem characteristics and associated dataset.
- A positive outcome of the learning process requires a careful calibration of the key parameters through extensive tests: in the current version, computational times suffered from the adopted state-space characterization (several thousand scenarios) and call for the adoption of efficient sampling schemes. On the positive side, the convergence to the optimal FxM led consistently to good in-sample and out-of-sample results as shown in Tables 8 and 10.
- Focusing on the original 1-month FxM problem, the evidence is very positive both from methodological and financial perspectives. In this instance, the wealth distribution generated by the optimal control has been shown to stochastically dominate the benchmark distribution with an order often much stronger than SSD, consistently so, over several instances and with very good ex-post performance.


(a) $\beta = 0.98$

(b) $\beta = 1$

Fig. 6. Portfolio allocation based on the RL approach, $\beta = 0.98, 1$.

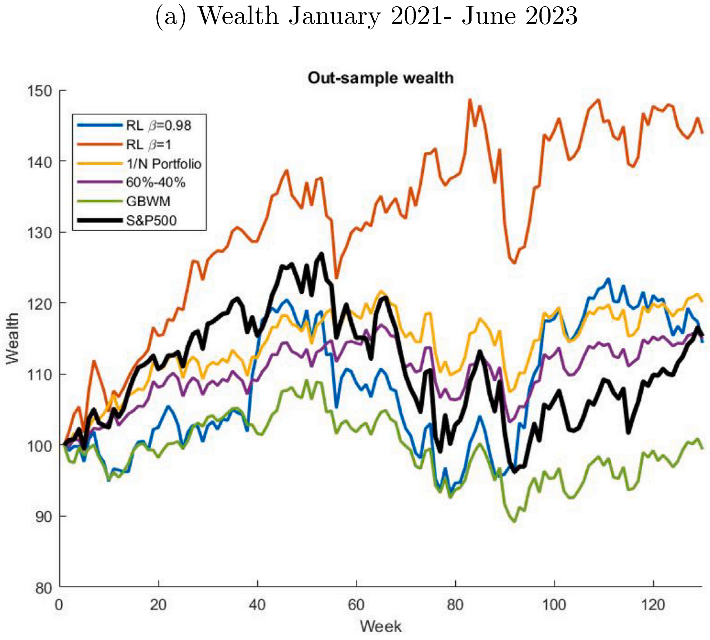

(a) Wealth January 2021- June 2023

Fig. 7. Backtesting results: investing \$100 over the period January 2021–June 2023. Optimal RL fixed-mix portfolios, $1/N$, the 60%–40% equity-bond strategy, and multiperiod goal-based wealth management (GBWM) versus the SP500.

- From a computational viewpoint it must be remarked that when employing for comparative reasons a different RL approach, the GBWM model, accessible within the Matlab financial toolbox, the evidence is significantly improved while from a financial perspective, given the underlying different type of problem, the results are overall comparable in-sample but far less consistent out-of-sample and in general do not accommodate any form of risk control based on stochastic dominance principles.

The above remarks should be considered relying also on the quite extended set of results presented in Appendix A and Appendix B, that contribute to the project.

Future directions

There is quite some space for improvements and future work.

From a financial perspective, very much related to the refinement of the methodology, to further improve performance on similar mean-risk problems, the decision space should be increased as a number of assets and the investment horizon extended preserving the rebalancing frequency. In the context of the proposed ICVaR-based performance enhancement, a dynamic calibration of the reference β for a given α should be evaluated.

From a decision-theoretic standpoint, the research on risk functionals with a theoretical guarantee of first- rather than second-order stochastic dominance is ongoing. The integration of such a result in the RL model specification, as suggested by the computational results, could enhance the out-sample results of the proposed methodology.

Most of the work should go nonetheless on the RL methodology. From a methodological perspective, utilizing machine learning tools and artificial intelligence techniques could be valuable. Improvements in the adopted RL methodology could include using techniques from Gomez et al. (2021, 2023) to compute the actor gradient based on sample points. Implementing dimensional reduction techniques to reduce the algorithm's computational complexity while preserving relevant data features, particularly in the branching tree, is important.

CRediT authorship contribution statement

Giorgio Consigli: Conceptualization, Formal analysis, Methodology, Supervision, Validation, Writing – original draft, Writing – review & editing. **Alvaro A. Gomez:** Conceptualization, Data curation, Methodology, Software, Validation, Writing – original draft. **Jorge P. Zubelli:** Conceptualization, Validation, Writing – review & editing.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

Giorgio Consigli acknowledges the support from Khalifa University of Science and Technology, Grant FSU2022-010 award 000634-00001, project no. 8474000393.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.engappai.2024.108599.

References

Al-Aradi, A., Correia, A., Naiff, D., Jardim, G., Saporito, Y., 2018. Solving nonlinear and high-dimensional partial differential equations via deep learning. arXiv preprint arXiv:1811.08782.

Almahdi, S., Yang, S.Y., 2017. An adaptive portfolio trading system: A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown. Expert Syst. Appl. 87, 267–279.

Bayraktar, E., Kara, A.D., 2023. Approximate Q-learning for controlled diffusion processes and its near optimality. arXiv:2203.07499.

Bender, J., Briand, R., Nielsen, F., Stefek, D., 2010. Portfolio of risk premia: A new approach to diversification. J. Portfolio Manag. 36 (2), 17–25.

Bianchi, D., Guidolin, M., 2014. Can long-run dynamic optimal strategies outperform fixed-mix portfolios? Evidence from multiple data sets. European J. Oper. Res. 236 (1), 160–176.

Black, F., Perold, A., 1992. Theory of constant proportion portfolio insurance. J. Econom. Dynam. Control 16 (3–4), 403–426.

Bottou, L., 2010. Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers. Springer, pp. 177–186.

Brandimarte, P., 2021. From Shortest Paths to Reinforcement Learning: A MATLAB-Based Tutorial on Dynamic Programming. Springer Nature.

Chow, Y., Ghavamzadeh, M., Janson, L., Pavone, M., 2018. Risk-constrained reinforcement learning with percentile risk criteria. J. Mach. Learn. Res. 18 (167), 1–51.

Coache, A., Jaimungal, S., 2024. Reinforcement learning with dynamic convex risk measures. Math. Finance 34 (2), 557–587.

Coache, A., Jaimungal, S., Cartea, Á., 2023. Conditionally elicitable dynamic risk measures for deep reinforcement learning. SIAM J. Financial Math. 14 (4), 1249–1289.

Consigli, G., Kuhn, D., Brandimarte, P. (Eds.), 2016. Optimal Financial Decision Making Under Uncertainty. In: International Series in Operations Research and Management Science, vol. 245, Springer.

Cont, R., Tankov, P., 2009. Constant proportion portfolio insurance in the presence of jumps in asset prices. Math. Finance 19 (3), 379–401.

Črepinšek, M., Liu, S.-H., Mernik, M., 2013. Exploration and exploitation in evolutionary algorithms: A survey. ACM Comput. Surv. 45 (3), 1–33.

Das, S.R., Varma, S., 2020. Dynamic goals-based wealth management using reinforcement learning. J. Invest. Manag. 18 (2), 1–20.

Dempster, M., Evstigneev, I.V., Schenk-Hoppé, K.R., 2011. Growing wealth with fixed-mix strategies. In: The Kelly Capital Growth Investment Criterion: Theory and Practice. World Scientific, pp. 427–455.

Dempster, M., Germano, E., Medova, M., Rietbergen, M., Sandrini, F., Scrowston, M., Zhang, N., 2007. DC pension fund benchmarking with fixed-mix portfolio optimization. Quant. Finance 7 (4), 365–370.

Dempster, M.A., Leemans, V., 2006. An automated FX trading system using adaptive reinforcement learning. Expert Syst. Appl. 30 (3), 543–552.

Denault, M., Simonato, J.-G., 2017. Dynamic portfolio choices by simulation-and-regression: Revisiting the issue of value function vs portfolio weight recursions. Comput. Oper. Res. 79, 174–189.

Deng, Y., Bao, F., Kong, Y., Ren, Z., Dai, Q., 2016. Deep direct reinforcement learning for financial signal representation and trading. IEEE Trans. Neural Netw. Learn. Syst. 28 (3), 653–664.

Dentcheva, D., Ruszczyński, A., 2003. Optimization with stochastic dominance constraints. SIAM J. Optim. 14 (2), 548–566.

Dentcheva, D., Ruszczyński, A., 2008. Duality between coherent risk measures and stochastic dominance constraints in risk-averse optimization. Pac. J. Optim. 4 (3), 433–446.

Dupačová, J., Consigli, G., Wallace, S.W., 2000. Scenarios for multistage stochastic programs. Ann. Oper. Res. 100, 25–53.

Fei, Y., Yang, Z., Chen, Y., Wang, Z., Xie, Q., 2020. Risk-sensitive reinforcement learning: Near-optimal risk-sample tradeoff in regret. Adv. Neural Inf. Process. Syst. 33, 22384–22395.

Fleten, S.-E., Høyland, K., Wallace, S.W., 2002. The performance of stochastic dynamic and fixed mix portfolio models. European J. Oper. Res. 140 (1), 37–49.

Geist, M., Pérolat, J., Lauriere, M., Elie, R., Perrin, S., Bachem, O., Munos, R., Pietquin, O., 2021. Concave utility reinforcement learning: the mean-field game viewpoint. Int. Found. Auton. Agents Multiagent Syst..

Gomez, A., Consigli, G., Liu, J., 2024. Multi-period portfolio selection with interval-based conditional value-at-risk. Ann. Oper. Res. In print, 1–38.

Gomez, A.A., Neto, A.J.S., Zubelli, J.P., 2021. Diffusion representation for asymmetric kernels. Appl. Numer. Math. 166, 208–226.

Gomez, A.A., Neto, A.J.S., Zubelli, J.P., 2023. A diffusion-map-based algorithm for gradient computation on manifolds and applications. IEEE Access 11, 90622–90640.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press, http://www.deeplearningbook.org.

Gu, S., Lillicrap, T., Sutskever, I., Levine, S., 2016. Continuous deep q-learning with model-based acceleration. In: International Conference on Machine Learning. PMLR, pp. 2829–2838.

Hambly, B., Xu, R., Yang, H., 2021. Policy gradient methods for the noisy linear quadratic regulator over a finite horizon. SIAM J. Control Optim. 59 (5), 3359–3391.

Han, J., Jentzen, A., et al., 2017. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. Commun. Math. Statist. 5 (4), 349–380.

Hasselt, H., 2010. Double Q-learning. Adv. Neural Inf. Process. Syst. 23.

Hasselt, H.v., Guez, A., Silver, D., 2016. Deep reinforcement learning with double Q-learning. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. AAAI '16, AAAI Press, pp. 2094–2100.

Hazan, E., Kakade, S., Singh, K., Van Soest, A., 2019. Provably efficient maximum entropy exploration. In: International Conference on Machine Learning. PMLR, pp. 2681–2691.

Huré, C., Pham, H., Warin, X., 2020. Deep backward schemes for high-dimensional nonlinear PDEs. Math. Comp. 89 (324), 1547–1579.

Infanger, G., 2008. Dynamic asset allocation strategies using a stochastic dynamic programming aproach. In: Handbook of Asset and Liability Management. Elsevier, pp. 199–251.

Jaimungal, S., 2022. Reinforcement learning and stochastic optimisation. Finance Stoch. 26, 103–129.

Jiang, R., Saunders, D., Weng, C., 2022. The reinforcement learning Kelly strategy. Quant. Finance 22 (8), 1445–1464.

Jiang, Z., Xu, D., Liang, J., 2017. A deep reinforcement learning framework for the financial portfolio management problem. pp. 0–30, arXiv preprint arXiv:1706.10059.

Kim, J.H., Kim, W.C., Fabozzi, F.J., 2014. Recent developments in robust portfolios with a worst-case approach. J. Optim. Theory Appl. 161, 103–121.

Lillicrap, T., Hunt, J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D., 2015. Continuous control with deep reinforcement learning. CoRR.

Liu, J., Chen, Z., Consigli, G., 2021. Interval-based stochastic dominance: theoretical framework and application to portfolio choices. Ann. Oper. Res. 307 (1–2), 329–361.

Matlab, 2020. Multiperiod goal-based wealth management using reinforcement learning. GBWM [Online; Accessed 25 March 2024].

Miryoosefi, S., Brantley, K., Daume, III, H., Dudik, M., Schapire, R.E., 2019. Reinforcement learning with convex constraints. Adv. Neural Inf. Process. Syst. 32.

Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K., 2016. Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning. PMLR, pp. 1928–1937.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M., 2013. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al., 2015. Human-level control through deep reinforcement learning. Nature 518 (7540), 529–533.

Moré, J.J., 2006. The levenberg-marquardt algorithm: implementation and theory. In: Numerical Analysis: Proceedings of the Biennial Conference. Dundee, June 28–July 1, 1977, Springer, pp. 105–116.

Müller, A., Scarsini, M., Tsetlin, I., Winkler, R.L., 2017. Between first- and second-order stochastic dominance. Manage. Sci. 63 (9), 2933–2947.

Mutti, M., De Santi, R., De Bartolomeis, P., Restelli, M., 2023. Convex reinforcement learning in finite trials. J. Mach. Learn. Res. 24 (250), 1–42.

Ogryczak, W., Ruszczyński, A., 2001. On consistency of stochastic dominance and mean-semideviation models. Math. Program. 89 (2), 217–232.

Rockafellar, R.T., Uryasev, S., 2002. Conditional Value at Risk for general loss distributions. J. Bank. Financ. 26, 1443–1471.

Ruder, S., 2016. An overview of gradient descent optimization algorithms. pp. 1–14, arXiv preprint arXiv:1609.04747.

Ruszczynski, A., 2010. Risk-averse dynamic programming for Markov decision processes. Math. Program. B 125, 235–261.

Shakya, A.K., Pillai, G., Chakrabarty, S., 2023. Reinforcement learning algorithms: A brief survey. Expert Syst. Appl. 120495.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., Riedmiller, M., 2014. Deterministic policy gradient algorithms. In: 31st International Conference on Machine Learning, Vol. 1. ICML 2014.

Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning: An Introduction. MIT Press.

Tamar, A., Glassner, Y., Mannor, S., 2014. Policy gradients beyond expectations: Conditional value-at-risk. arXiv arXiv:1404.3862.

Tsetlin, I., Winkler, R.L., Huang, R.J., Tzeng, L.Y., 2015. Generalized almost stochastic dominance. Oper. Res. 63 (2), 363–377.

Wang, H., Zariphopoulou, T., Zhou, X.Y., 2020. Reinforcement learning in continuous time and space: A stochastic control approach. J. Mach. Learn. Res. 21 (198), 1–34.

Wang, H., Zhou, X.Y., 2020. Continuous-time mean–variance portfolio selection: A reinforcement learning framework. Math. Finance 30 (4), 1273–1308.

Wei, P., 2020. Exploration-exploitation strategies in deep Q-networks applied to route-finding problems. J. Phys. Conf. Ser. 1684 (1), 012073.

Wilamowski, B.M., Yu, H., 2010. Improved computation for Levenberg–Marquardt training. IEEE Trans. Neural Netw. 21 (6), 930–937.

Yu, H., Wilamowski, B.M., 2018. Levenberg–marquardt training. In: Intelligent Systems. CRC Press, pp. 1–12.

Zahavy, T., O'Donoghue, B., Desjardins, G., Singh, S., 2021. Reward is enough for convex mdps. Adv. Neural Inf. Process. Syst. 34, 25746–25759.

Zhang, J., Koppel, A., Bedi, A.S., Szepesvari, C., Wang, M., 2020. Variational policy gradient method for reinforcement learning with general utilities. Adv. Neural Inf. Process. Syst. 33, 4572–4583.

Ziemba, R.E.S., Ziemba, W.T., 2008. Scenarios for Risk Manegement and Global Investment Strategies. Wiley Finance.