A New Approach for Solving a Control Stochastic Problem Driven by a Diffusion Process with Jumps

Oulghazi H.

IMIA Laboratory, A2MSDS Team, Department of Mathematics, Faculty of Sciences and Technics,
Moulay Ismail University of Meknes, P.O. Box 509 Boutalamine, Errachidia 52000, Morocco

(Received 13 August 2025; Revised 28 January 2026; Accepted 31 January 2026)

In this paper, we focus on the numerical solution of high-dimensional stochastic optimal control problems, whose system states are modeled as jump-diffusion processes. Through the maximum principle and deep neural networks, we restate the original control problem as a variational problem, and we introduce specialized algorithms to solve this new formulation. The algorithms and the various architectures employed have been introduced. The mean-variance portfolio selection problem in a financial market consisting of two kinds of assets in a jump-diffusion process setting validates the effectiveness of proposed algorithms.

Keywords: *forward-backward stochastic differential equations with jumps; stochastic maximum principle; deep neural networks; optimal stochastic control; jump diffusions; mean-variance portfolio selection problem.*

2020 MSC: 93E20, 49K45, 60J75, 68T05, 91B28 **DOI:** 10.23939/mmc2026.01.077

1. Introduction

The stochastic control theory is a potent paradigm to model and analyze the decision-making problems under random dynamics. This theory first appeared in the 1960s for linear state dynamics and the quadratic cost function. This problem is also called the linear stochastic regulator. The problem of optimal stochastic control began to be applied in finance and management in the 1970s, notably with the publication of Merton's paper on portfolio allocation [1]. Subsequently, many authors have generalized Merton's model and results; see [2, 3]. The two major powerful ways and crucial methods of solving stochastic optimal control problems are well known as Pontryagin's maximum principle [4] and Bellman's dynamic programming principle [5]. These principles have greatly influenced the development of stochastic control theory and have been extended to solve many complex problems in contemporary science and technology. Another option for solving stochastic control problems is to use numerical approaches such as classical numerical methods, e.g., finite differences, which are commonly used to approximate the partial differential equations (PDEs) associated with control problems. However, these methods suffer from exponentially increasing complexity with respect to the dimensionality of the state space, while the Markov chain approximation method approximates the original controlled process with a controlled Markov chain on a finite state space [6]. Then, Monte Carlo methods, in particular backward Monte Carlo approaches, estimate the solution by simulating many stochastic trajectories. But these classical approaches still suffer from high computational cost and potentially slow convergence. In the past few years, deep learning approaches have been rapidly developed and have shown great success in high-dimensional problems in many application domains [7], and they really pave the way for reducing the curse of dimensionality. In fact, the deep learning method has proven successful in the realm of stochastic optimal control problems, as well as in solving (backward) forward stochastic differential equations (FBSDE) for some highly dimensional problems [8–10]. Due to the favorable results concluded in [8], this type of problem attracted attention, and the research community has proposed many neural network-based algorithms to address such problems, especially the deep learning-based algorithms that can be used from both paradigms: stochastic maximum principle and dynamic programming, such as e.g., [11–15].

https://orcid.org/0009-0007-7518-291X (Oulghazi H.)

A good number of these algorithms are based on a forward-backward stochastic differential equation (FBSDE) obtained through a stochastic characterization of the solution to the Hamilton–Jacobi–Bellman (HJB) equation, taking advantage of links between PDEs and BSDEs given by nonlinear Feynman–Kac type results. However, these techniques become intractable as soon as one also controls the diffusion of the state process, since they do not solve for the second derivatives of the value function, which are necessary to determine optimal control. For overcoming this problem and building a diffusion control formulation, the authors in [11] have developed a deep BSDE algorithm, and the corresponding FBSDE satisfy the stochastic maximum principle (SMP), which they have named the Deep SMP-BSDE.

This work presents an approach using the maximum principle for a stochastic control problem where the dynamics are given by a diffusion process with jumps, which are often more realistic than the non-jump-only background models. The principle of maximum was initially introduced by Pontryagin et al. for the deterministic case [16]. The analogous maximum principle for Itô diffusions was established by [17] and later extended by [18], and others. A sufficient maximum principle has recently been established for jump diffusions by [19]. The rest of this work is organized as follows. In Section 2 we provide some preliminaries about stochastic optimal control problems with jumps and recast our stochastic optimal control problem as a new variational problem. In Section 3 we derive our numerical algorithm for solving this last problem, and we give different neural network architectures. In the final section, we illustrate the numerical results and make a comparison of the results achieved by the proposed algorithms.

2. Preliminaries and problem formulation

2.1. Notations and preliminaries

In this section, we first introduce the necessary notations and formalize the underlying assumptions that govern our framework. We then establish preliminary results that will play a central role in the subsequent analysis. Afterwards, we rigorously formulate the stochastic control problem with jumps along with its associated adjoint processes and finally derive the corresponding stochastic maximum principle.

We consider a complete filtered probability space $(\Omega, \mathbb{F}, \mathcal{F}_t, P)$ for all $t \in [0,T]$, where $T < \infty$. $(\mathcal{F}_t)_{t \geqslant 0}$ is the natural filtration generated by a d-dimensional Brownian motion $(B_t) \in \mathbb{R}^d$ and the Poisson random measure $\widetilde{\mathcal{L}}(dt,dz) \in \mathbb{R}^l$ such that the initial σ-algebra $\mathcal{F}_0$ includes all P-null sets in $\mathcal{F}$, and that the filtration $\{\mathcal{F}_t\}_{t \geqslant 0}$ is right-continuous.

Assume that $\mu(dz)$, the σ finite Levy measure associated with the Poisson random measure $\mathcal{L}(dt,dz)$, satisfies the following conditions:

$$\widetilde{\mathcal{L}}(dt,dz) = \mathcal{L}(dt,dz) - \mu(dz)\,dt \quad \text{and} \quad \int_{\mathbb{R}^n \setminus \{0\}} |z|^2 \mu(dz) < \infty.$$

The second condition signified that the Poisson random measure $\mathcal{L}$ does not have many big jumps; however, this does not imply that they do not occur, but rather that the probability of their occurrence is low. The term $\mu(dz)\,dt$ is referred to as the compensator associated with the Poisson random measure $\widetilde{\mathcal{L}}(dt,dz)$.

We define (L_t) the component that models jumps in terms of the Poisson random measure $\mathcal{L}$ within a Lévy process as follows:

$$\forall\, 0 \leqslant t \leqslant T, \quad L_t := \int_0^t \int_{\mathbb{R}^n \setminus \{0\}} z\, \mathcal{L}(ds,dz). \tag{1}$$

And for all $t > 0$ $\Delta L_t := L_t - L_{t-}$. Also, we define, the Poisson random process N_t in terms of a Poisson random measure $\mathcal{L}$ by

$$N_t := \int_0^t \int_{\mathbb{R}^n \setminus \{0\}} \mathcal{L}(ds,dz).$$

(N_t) has the intensity λt where $\lambda = \int_{\mathbb{R}^n \setminus \{0\}} \mu(dz) < \infty$.

In this work, we are interested in solving a stochastic optimal problem when the state system is described by a diffusion process with jumps. Precisely, we consider the following problem.

Find $u^* \in \mathcal{A}$ that maximizes the performance criterion J such that

$$J(u) = E\left[\int_0^T f(t, x_t, u_t)\, dt + g(x_T)\right],$$

that means finding $u^* \in \mathcal{A}$ that satisfies

$$J(u^*) = \sup_{u \in \mathcal{A}} J(u), \tag{2}$$

where $f\colon [0,T] \times \mathbb{R}^n \times U \to \mathbb{R}$ is a continuous function and $g\colon \mathbb{R}^n \to \mathbb{R}$ is concave and is C^1. Suppose that the state system $X(t)$ is defined in $\mathbb{R}^n$ as a diffusion process with jumps, and it is given by

$$\begin{cases} dX_t = b(t, X_t, u_t)\, dt + c(t, X_t, u_t)\, dB_t + \displaystyle\int_{\mathbb{R}^n \setminus \{0\}} \gamma(t, X_{t^-}, u_{t^-}, z)\, \widetilde{\mathcal{L}}(dt, dz), \\ X(0) = x_0 \in \mathbb{R}^n, \end{cases} \tag{3}$$

where

$$b\colon [0,T] \times \mathbf{R}^n \times U \longmapsto \mathbf{R}^n, \quad c\colon [0,T] \times \mathbf{R}^n \times U \longmapsto \mathbf{R}^{n\times d}, \quad \gamma\colon [0,T] \times \mathbf{R}^n \times U \times \mathbf{R}^n \longmapsto \mathbf{R}^{n\times \ell}$$

are given continuous functions, and

$$E\left[\int_0^T f^-(t, X_t, u_t)\, dt + g^-(X_T)\right] < \infty.$$

The process $u \in U \in \mathbb{R}^n$ is our control. Note that $\mathcal{A}$ is the set of all admissible controls; this set contains each adapted and caglad control u such that equation (3) has a unique strong solution X^u. So, the next step is to give the Hamiltonian function and the associated adjoint system for the above problem; finally, we give the various assumptions for the existence of the solution to different problems. Now, we impose the following assumption.

Hypothesis 1.

1. The functions b, c, and g are locally bounded and Borel-measurable functions.
2. There exists a constant $K \geqslant 0$ and a deterministic positive function φ_z such that $\forall (t, v, v', X, X', z) \in [0,T] \times U^2 \times \mathbb{R}^{2n} \times \mathbb{R}^n \setminus \{0\}$ and for $\Psi = b, c$, the following holds:
   - $|\Psi(t, X, v) - \Psi(t, X', v')| \leqslant K\,(|X - X'| + |v - v'|)$,
   - $|\gamma(t, X, v, z) - \gamma(t, X', v', z)| \leqslant \varphi(z)\,(|X - X'| + |v - v'|)$,
   - $|\Psi(t, X, v)| \leqslant C\,(1 + |X| + |v|)$,
   - $|\gamma(t, X, v, z)| \leqslant \varphi_z\,(1 + |X| + |v|)$.
3. Furthermore, we impose that $\mathcal{A}$ is a compact subset of $\mathbb{R}^n$. In addition, we assume that,

$$\int_{\mathbb{R}^n \setminus \{0\}} \exp\{\varphi(z)\}\, \mu(dz) < \infty.$$

As stated in [20], under assumptions 1, for any admissible control $u(\cdot) \in \mathcal{A}$ and initial state $x_0 \in \mathbb{R}^n$, the system (3) admits a unique strong solution. Furthermore, according to [19], the Hamiltonian function associated with this problem is written as follows:

$$\begin{aligned} H(t, X, u, P, Q, r) &= f(t, X, u) + Pb^T(t, X, u) + \operatorname{tr}\left(c^T(t, X, u)\, Q\right) \\ &\quad + \sum_{j=1}^{l} \sum_{i=1}^{n} \int_{\mathbb{R}\setminus\{0\}} \gamma_{ij}(t, X, u, z_j)\, r_{ij}(t, z_j)\, \mu_j(dz_j), \end{aligned} \tag{4}$$

$H\colon [0,T] \times \mathbb{R}^n \times U \times \mathbb{R}^n \times \mathbb{R}^{n\times d} \times F \to \mathbb{R}$, where F is the set of functions defined in $\mathbb{R}^{n+1}$ with values in $\mathbb{R}^{n\times l}$ provided that the integral of the Hamiltonian H converges. From now on, we assume that H is differentiable from X.

For any admissible pair (X^u, u), we define the corresponding adjoint system as the following backward stochastic differential equation (BSDE):

$$\begin{cases} dP_t = -\nabla_x H(t, X_t, u_t, P_t, Q_t, r_t(\cdot))\, dt + Q_t\, dB(t) + \displaystyle\int_{\mathbb{R}^n\setminus\{0\}} r_t(z)\, \widetilde{\mathcal{L}}(dt, dz), \\ P_T = \nabla_X g(X_T), \end{cases} \tag{5}$$

The unknown processes of this system are $P \in \mathbb{R}^n$, $Q \in \mathbb{R}^{n\times d}$, and $r \in \mathbb{R}^{n\times l}$. Now, we consider the following hypotheses.

Hypothesis 2.

1. For all $(X, v, Y, Z, r), (X', v', Y', Z', r') \in \mathbb{R}^n \times U \times \mathbb{R}^n \times \mathbb{R}^{n\times d} \times F$, there exists a constant $C \geqslant 0$ such that the following holds:
   - $|\nabla_X H(t, X, v, Y, Z, r) - \nabla_x H(t, X', v', Y', Z', r')| + |\nabla g(X) - \nabla g(X')| \leqslant C(1 + |X| + |X'| + |v| + |v'|)(|X - X'| + |v - v'|) + |Y - Y'| + |Z - Z'| + |r - r'|)$,
   - $|\nabla_X H(t, X, v, Y, Z, r)| + |\nabla g(X)| \leqslant C\left(1 + |X|^2 + |v|^2 + |Y| + |Z| + |r|\right)$.
2. The function $r \mapsto \nabla_X H(t, X, v, Y, Z, r)$ is increasing for all $(t, X, v, Y, Z) \in [0, T] \times \mathbb{R}^n \times U \times \mathbb{R}^n \times \mathbb{R}^{n\times d}$ and $0 \leqslant r(t, Z) \leqslant C(1 + |Z|)$ for all $(t, Z) \in [0, T] \times \mathbb{R}^{n\times d}$.

Based on assumption (2) and by [20], the adjoint system (5) admits a unique solution, (P_t, Q_t, R_t) and from the assumptions (1)–(2), the cost function (2) is well defined.

Before giving the maximum stochastic principle for our problem, we assume that the forward system with jumps 3 and its adjoint system 5 satisfied respectively to two conditions

$$\begin{gathered} \mathbb{E}\left[\int_0^T \left(\sigma\sigma^T + \sum_{j=1}^{l} \int_{\mathbb{R}\setminus\{0\}} |\gamma_j(t, X_t, u_t, z_j)|^2 \mu_j(dz_j)\right) dt\right] < \infty \quad \forall u \in \mathcal{A}, \\ \mathbb{E}\left[\int_0^T \left(QQ^T + \sum_{j=1}^{l} \int_{\mathbb{R}\setminus\{0\}} |r_j(t, z_j)|^2 \mu_j(dz_j)\right) dt\right] < \infty. \end{gathered} \tag{6}$$

The integrability of the terms in the first condition guarantees well-defined and controllable system trajectories, while the second condition ensures that the adjoint equation has a stable solution.

Theorem 1 (Ref. [19]). *Let $\hat{u} \in \mathcal{A}$ such that $(\hat{X}_t, \hat{u}_t)$ is the solution of (3) and suppose there exists a solution $(\hat{P}_t, \hat{Q}_t, \hat{r}_t(\cdot))$ of the corresponding adjoint equation (5) satisfying assumptions (6). Moreover, suppose that*

$$H\big(t, \hat{X}_t, \hat{u}_t, \hat{P}_t, \hat{Q}_t, \hat{r}_t(\cdot)\big) = \sup_{v\in U} H\big(t, \hat{X}_t, v, \hat{P}_t, \hat{Q}_t, \hat{r}_t(\cdot)\big)$$

for all $t \in [0, T]$, and suppose that the function

$$(X, v) \longrightarrow H\big(t, X, v, \hat{P}_t, \hat{Q}_t, \hat{r}_t(\cdot)\big) \tag{7}$$

is concave for all $t \in [0, T]$. Then $\hat{u}$ is optimal control of the problem (2).

The proof is based on a lemma 3.6 in [21], which provides a formula for integration by parts; she is also referred to as the Itô formula.

2.2. Problem formulation

It is clear that, according to the theorem 1, any optimal control $\hat{u}$ associated with an optimal trajectory $\hat{x}$ is also the solution of a Hamiltonian system plus a maximization condition of a function H. In our case, the system is in the form of an FBSDE plus a maximization condition,

$$\begin{cases} d\hat{p}_t = -\nabla_x H(t, \hat{x}_t, \hat{u}_t, \hat{p}_t, \hat{q}_t, \hat{r}_t(\cdot))\, dt + \hat{q}_t\, dB_t + \displaystyle\int_{\mathbb{R}^n\setminus\{0\}} \hat{r}_t(z)\, \widetilde{\mathcal{L}}(dt, dz) \\ d\hat{x}_t = b(t, \hat{x}_t, \hat{u}_t)\, dt + c(t, \hat{x}_t, \hat{u}_t)\, dB(t) + \displaystyle\int_{\mathbb{R}^n\setminus\{0\}} \gamma\,(t, \hat{x}_{t^-}, \hat{u}_{t^-}, z)\, \widetilde{\mathcal{L}}(dt, dz), \\ \hat{x}(0) = x_0, \quad \hat{p}_T = \nabla_x g(\hat{x}_T), \\ H(t, \hat{x}_t, \hat{u}_t, \hat{p}_t, \hat{q}_t, \hat{r}_t(\cdot)) = \max\limits_{v\in U} H(t, \hat{x}_t, v, \hat{p}_t, \hat{q}_t, \hat{r}_t(\cdot)) \quad \forall t. \end{cases} \tag{8}$$

Using an extension of the approach discussed in [10] and [22] for solving an FBSDE, we consider the following new control problem, referred to as the associated variational problem, related to the previous problem,

$$\inf_{p_0,q_t,r_t} \mathbb{E}\left[\left|p_T^{p_0,q,r} - \nabla_x g(x_T^{p_0,q,r})\right|^2\right] \tag{9}$$

subject to

$$\begin{cases} x_t^{p_0,q,r} = x_0 + \displaystyle\int_0^t b(s, x_s^{p_0,q,r}, u_s^{p_0,q,r})\, ds + \int_0^t c(s, x_s^{p_0,q,r}, u_s^{p_0,q,r})\, dB_s \\ \qquad + \displaystyle\int_0^t \int_{\mathbb{R}^n\setminus\{0\}} \gamma\left(s, x_{s^-}^{p_0,q,r}, u_{s^-}^{p_0,q,r}, z\right) \widetilde{\mathcal{L}}(ds,dz), \\ p_t^{p_0,q,r} = p_0 - \displaystyle\int_0^t \nabla_x H\big(t, x_s^{p_0,q,r}, u_s^{p_0,q,r}, p_s^{p_0,q,r}, q_s, r_s(\cdot)\big)\, ds + \int_0^t q_s\, dB_s \\ \qquad + \displaystyle\int_0^t \int_{\mathbb{R}^n\setminus\{0\}} r_s(z)\, \widetilde{\mathcal{L}}(ds,dz) \\ u_t^{p_0,q,r} = \arg\max_{v\in U} H\big(t, x_t^{p_0,q,r}, v, p_t^{p_0,q,r}, q_t, r_t(\cdot)\big) \quad \forall t. \end{cases} \tag{10}$$

Now, we give an important proposition, which will play an important role later on.

Proposition 1. Under assumptions (1)–(2) and (6), the variational control problem (9) satisfied

$$\inf_{p_0,q_t,r_t} E\left[\left|p_T^{p_0,q,r} - \nabla_x g(x_T^{p_0,q,r})\right|^2\right] = 0.$$

Moreover, the corresponding quintuple $\big(x_t^{p_0,q,r}, u_t^{p_0,q,r}, p_t^{p_0,q,r}, q_t, r_t(\cdot)\big)$ is the unique solution of FBSDE (8) and $u_t^{p_0,q,r}$ is the unique solution of problem (2).

Proof. If the assumptions (1)–(2) and (6) hold, then by Theorem 1, the Hamiltonian system (8) has a unique solution. So, by regarding $(p_0; q_t; r_t)$ as the control of the variational problem (9)–(10), and observing that the dynamics constraint is satisfied. Since we have $p_T = \nabla_x g(x_T)$ $p - a.s.$ Then

$$\mathbb{E}\left[\left|p_T^{p_0,q,r} - \nabla_x g(x_T^{p_0,q,r})\right|^2\right] = 0,$$

and the corresponding quintuple $(x_t^{p_0,q,r}, u_t^{p_0,q,r}, p_t^{p_0,q,r}, q_t, r_t(\cdot))$ is the unique solution of the Hamiltonian system (8). Indeed, if the assumptions (1)–(2) and (6) hold, the solution is unique. ■

Lemma 1. *Under the assumptions (1)–(2)–(6), and if there exists $(x^*, u^*, p^*, q^*, r^*(\cdot))$ solution of (10) satisfying,*

$$\mathbb{E}\left[|p_T^* - \nabla_x g(x_T^*)|^2\right] = 0.$$

Then $(x^*, u^*, p^*, q^*, r^*(\cdot))$ is the unique solution of (8) and the criterion performance J can be calculated by

$$J(u^*) = E\left[\int_0^T f(t, x_t^*, u_t^*)\, dt + g(x_T^*)\right].$$

Remark 1.

- The previous proposition shows that, under the same assumptions as in the SMP theorem (1), the variational problem (9)–(10) admits a unique solution, while the lemma demonstrates that this solution is also the solution to the original control problem (2).
- It is important to highlight the significance of this transformation. The benefit of the new problem lies in the fact that we are working with a cost function that is quadratic.

3. Numerical algorithms and Neural network architectures

3.1. Numerical algorithm 1

In Section 2, we provided a brief introduction to the Stochastic Maximum Principle (SMP) in the context of our control problem and reformulated it as a new variational problem equivalent to the associated Hamiltonian system. In this section, however, we introduce three algorithms based on deep learning to solve this reformulated variational problem.

As a first step, we discretize the problem 10. Given a positive integer N and a fixed $T > 0$, we consider that the time interval $[0, T]$ is partitioned as follows: $t_0 = 0 < t_1 = t_0 + \Delta t < t_2 = t_0 + 2\Delta t < \ldots < t_N = t_0 + N\Delta t = T$ assuming that the grid is uniform with a constant step size $\Delta t = t_{n+1} - t_n = \frac{T}{N}$ for each $n \in \{0, 1, \ldots, N-1\}$. $\Delta B_{t_n} = B_{t_{n+1}} - B_{t_n}$ denote the increments of the Brownian motion such that $\Delta B_{t_n} \sim N(0, \sqrt{\Delta t_n})$, and $\Delta N_{t_n} = N_{t_{n+1}} - N_{t_n}$ denote the increments of the Poisson random process such that $\Delta N_{t_n} \sim \mathcal{P}(0, \lambda \Delta t_n)$. $(\Delta L_k^n)_{k \in \{1,\ldots,dN_{t_n}\}}$ the k-th jump of the process (L_t) (see relation (1)), which occurs on the time interval $]t_n, t_{n+1}]$. And each element of $(\Delta L_k^n)_{k \in \{1,\ldots,dN_{t_n}\}}$ is sampled from a distribution $\frac{\mu(dz)}{\lambda} 1_{\mathbb{R}^n \setminus \{0\}}$.

Then, the Euler–Maruyama scheme of the problem 10 can be expressed as

$$\begin{cases} \tilde{x}^\pi_{t_{n+1}} = \tilde{x}^\pi_{t_n} + b(t_n, \tilde{x}^\pi_{t_n}, \tilde{u}^\pi_{t_n})\,\Delta t_n + c(t_n, \tilde{x}^\pi_{t_n}, \tilde{u}^\pi_{t_n})\,\Delta B_{t_n} + \sum_{k=1}^{dN_{t_n}} \gamma(t_n, \tilde{x}^\pi_{t_n}, \tilde{u}^\pi_{t_n}, \Delta L_k^n) \\ \qquad - \int_{\mathbb{R}^n \setminus \{0\}} \gamma(t_n, \tilde{x}^\pi_{t_n}, \tilde{u}^\pi_{t_n}, z)\,\mu(dz)\,\Delta t_n, \\ \tilde{p}^\pi_{t_{n+1}} = \tilde{p}^\pi_{t_n} - \nabla_x H(t_n, \tilde{x}^\pi_{t_n}, \tilde{u}^\pi_{t_n}, \tilde{p}^\pi_{t_n}, \tilde{q}^\pi_{t_n}, \tilde{r}^\pi_{t_n})\,\Delta t_n + \tilde{q}^\pi_{t_n}\,\Delta B_{t_n} + \tilde{r}^\pi_{t_n} \sum_{k=1}^{dN_{t_n}} \Delta L_k^n \\ \qquad -\lambda \tilde{r}^\pi_{t_n} \sum_{k=1}^{dN_{t_n}} \mathbb{E}(\Delta L_k^n) \Delta t_n, \\ \tilde{x}^\pi_0 = x_0, \quad \tilde{p}^\pi_0 = \tilde{p}_0, \\ \tilde{u}^\pi_{t_n} = \arg\max_{u \in U} H(t_n, \tilde{x}^\pi_{t_n}, u, \tilde{p}^\pi_{t_n}, \tilde{q}^\pi_{t_n}, \tilde{r}^\pi_{t_n}). \end{cases} \tag{11}$$

Remark 2. The numerical approximation of the two integrals of the system above can be carried using various methods, such as the Monte Carlo simulation method.

For the adjoint process p_{t_n}, we used a discretization method different from the one applied to the process x_{t_n}. This approach allows us to eliminate the second component of the function r_{t_n}, meaning that in this discretization, r_{t_n} only depends on t_n. Consequently, the use of deep learning to approximate the process r_{t_n} becomes more feasible. A similar discretization method was used by Agram et al. in [23].

$\{\tilde{q}^\pi_{t_n}\}_{0 \leqslant n < N}$ and $\{\tilde{r}^\pi_{t_n}\}_{0 \leqslant n < N}$ are considered as two controls of the variational problem (9). These two control are feedback controls of the state $\tilde{x}^\pi_{t_n}$, $\tilde{p}^\pi_{t_n}$ and assume it satisfies, respectively,

$$\begin{aligned} \tilde{q}^\pi_{t_n} &= \mathcal{NN}_1^{\theta_q}(t_n, \tilde{x}^\pi_{t_n}, \tilde{p}^\pi_{t_n}), \\ \tilde{r}^\pi_{t_n} &= \mathcal{NN}_2^{\theta_r}(t_n, \tilde{x}^\pi_{t_n}, \tilde{p}^\pi_{t_n}). \end{aligned}$$

$\mathcal{NN}_1^{\theta_q}$ and $\mathcal{NN}_2^{\theta_q}$ are two feedforward neural networks. In our work, we use the same neural network in all discrete instants. Unlike other works that use a neural network at each time step (see, for example, [10]).

Since both neural networks $\mathcal{NN}_1^{\theta_q}$ and $\mathcal{NN}_2^{\theta_r}$ have the same structure, we only develop, for example, the feedforward neural network $\mathcal{NN}_1^{\theta_q}$. For that, we consider $\mathcal{NN}_1^{\theta_q}$ the neural network that serves to approximate the process $\tilde{q}^\pi_{t_n}$. Denote by s_0 the input dimension and by s_1 the output dimension. The total number of hidden layers is specified by a fixed integer L (not counting input and output layers). We define s to be the number of neurons on each hidden layer and s represent the uniform number of neurons in all hidden layers.

$\mathcal{NN}_1^{\theta_q}$ is defined as a succession of affine transformations and nonlinear activation functions. Specifically, we have:

$$\mathcal{NN}_1^{\theta_q} = \Psi_{L+1} \circ \sigma_a \circ \Psi_L \circ \cdots \circ \sigma_a \circ \Psi_1 \circ \sigma_a \circ \Psi_0,$$

where σ_a is an activation function, for instance ReLU, tanh, sigmoid, ELU, $\ldots$; Ψ_0 is a function from $\mathbb{R}^{s_0}$ to $\mathbb{R}^s$, and for $l = 1$ to L, Ψ_l is a function defined on $\mathbb{R}^s$ to $\mathbb{R}^s$. Ψ_{L+1} is a function from $\mathbb{R}^s$ with values in $\mathbb{R}^{s_1}$. Each affine transformation Ψ_l is expressed in the form

$$\Psi_l(x) = A_l x + \beta_l,$$

where A_l represents a weight matrix and β_l denotes a bias vector.

The $\mathcal{NN}_1^{\theta_q}$ has parameters θ_q, which consist of all the weights and biases associated with the affine functions Ψ, i.e., $\theta_q = (A_l; \beta_l)_{0 \leqslant l \leqslant L+1}$.

The total number of parameters is

$$N_{L,s_0,s,s_1} = (s_0+1)s + L(1+s)s + (s+1)s_1,$$

The neural network $\mathcal{NN}_1^{\theta_q}$ is then trained by searching for the parameters θ_q that minimize the cost function $J(p_0,q,r) = E\left[\left|p_T^{p_0,q,r} - \nabla_x g(x_T^{p_0,q,r})\right|^2\right]$. This can be accomplished using a standard training algorithm, such as stochastic gradient descent or its variants, with the ADAM algorithm [24]. The cost functional $J(p_0,q,r)$ can be approximated by

$$J(p_0,q,r) = \frac{1}{M}\sum_{j=1}^{M}\left|\tilde{p}_T^{\pi,j} - \nabla_x g(\tilde{x}_T^{\pi,j})\right|^2.$$

M denotes the total number of Monte Carlo samples. We treat t, x_n and p_n as inputs; therefore, the dimensions of the input and output layers for $\mathcal{NN}_1^{\theta_q}$ are $2n+1$ and $n\times m$, respectively. Now, we recall the following universal approximation theorem.

Theorem 2 (Ref. [25]). *Assume that the nonconstant function σ_a is bounded and continuous. The neural network $\mathcal{NN}_1$ is then dense in $C(\mathbb{R}^n)$ with $L \geqslant 2$.*

The architecture of Algorithm 1 is as follows.

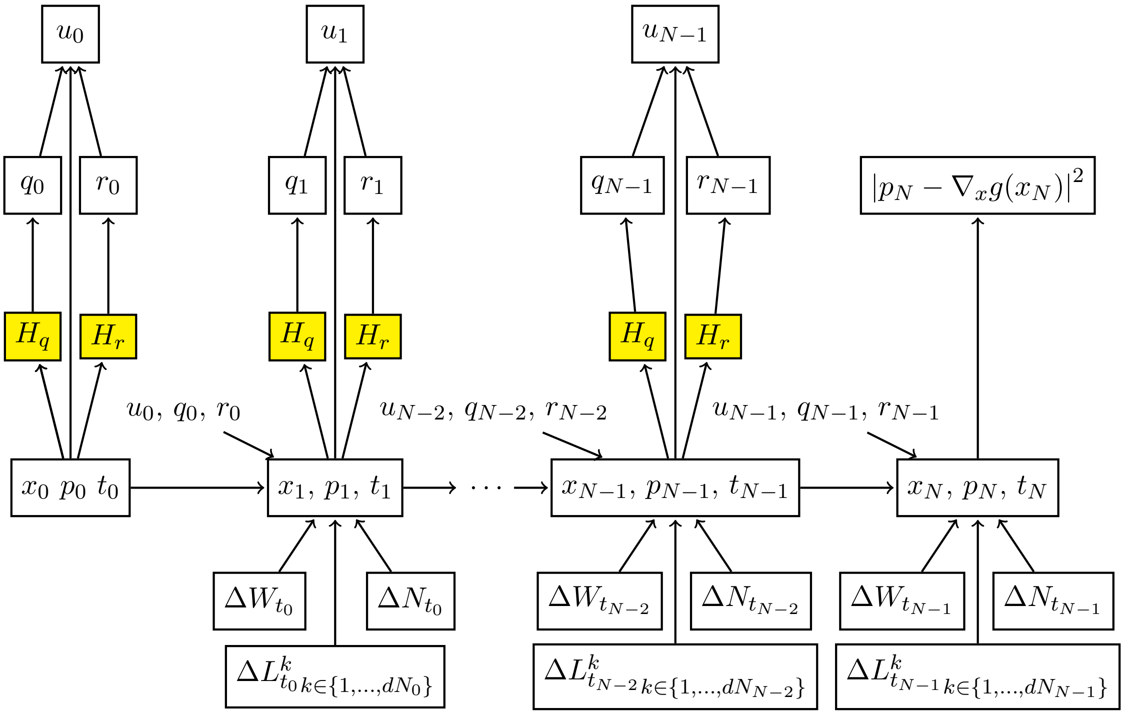

Figure 1. The architecture for Algorithm 1.

$\mathcal{H}_q$ and $\mathcal{H}_r$ represent the hidden layers associated with the two networks $\mathcal{NN}_1^{\theta_q}$ and $\mathcal{NN}_2^{\theta_r}$, respectively. The weights biases of the hidden layers (θ_q,θ_r) and p_0, are trainable parameters. $\tilde{u_n}$ is a function of $(t_n,\tilde{x_n},\tilde{p_n},\tilde{q_n},\tilde{r_n})$. $\tilde{q_n}$ and $\tilde{r_n}$ represent the outputs for $\mathcal{NN}_1^{\theta_q}$ and $\mathcal{NN}_2^{\theta_r}$, respectively, at each time step.

The neural network described above is employed at all time steps. However, the main problem with Algorithm 1 is that solving an extremum problem requires calculating $\tilde{u}_{t_n}^{\pi,l,j} = \arg\max_{u\in U} H\big(t_n,\tilde{x}_{t_n}^{\pi,l,j},u,\tilde{p}_{t_n}^{\pi,l,j},\tilde{q}_{t_n}^{\pi,l,j},\tilde{r}_{t_n}^{\pi,l,j}\big)$ at each time step, which is not straightforward, especially for high-dimensional problems and functions with multiple variables. In high dimensions, an explicit solution for these problems is often unavailable. However, there are several approximation methods that address this challenge, such as the BFGS algorithm [26]. Therefore, using this algorithm demands significant computational effort. Next, we present the pseudocode for Algorithm 1.

Algorithm 1 Algorithm with 2-NNs.

```
Require: Initial parameters $(\theta_q^0; \theta_r^0; \tilde{p}_{t_0}^{0,\pi})$, $x_0$ and $\eta$ a learning rate;
Ensure: $(\tilde{x}_{t_n}^{l,\pi}, \tilde{u}_{t_n}^{l,\pi}, \tilde{p}_{t_n}^{l,\pi}, \tilde{q}_{t_n}^{l,\pi}, \tilde{r}_{t_n}^{l,\pi})$;
for $l = 0, \dots, \mathrm{MaxIter}$
    for $j = 1, \dots, M$
        Generate $\Delta W_j^{t_n}$ based on $\mathcal{N}(0, \Delta t_n)$;
        Generate $\Delta N_j^{t_n}$ based on $P(\lambda \Delta t_n)$;
        Generate $\Delta L_{j,k}^{t_n}$ based on $\frac{\mu(dz)}{\lambda} 1_{\mathbb{R}^n \setminus \{0\}}$ for each $k = 1, \dots, dN_j^{t_n}$ where $\mu(dz) = \exp(-\nu \|z\|) dz$,
        $z \in \mathbb{R}^n \setminus \{0\}$;
        $\tilde{x}_{t_0}^{\pi,l,j} = \tilde{x_0}^{\pi}$, $\tilde{p}_{t_0}^{\pi,l,j} = \tilde{p_0}^{\pi,l}$;
        for $n = 0, \dots, N-1$
            $q_{t_n}^{l,\pi,j} = \mathcal{NN}_1^{\theta_q}(t_n, \tilde{x}_{t_n}^{\pi,l,j}, \tilde{p}_{t_n}^{\pi,l,j}, \theta_q^l)$;
            $r_{t_n}^{l,\pi,j} = \mathcal{NN}_2^{\theta_r}(t_n, \tilde{x}_{t_n}^{\pi,l,j}, \tilde{p}_{t_n}^{\pi,l,j}; \theta_r^l)$;
            $\tilde{u}_{t_n}^{\pi,l,j} = \arg\max_{u \in U} H(t_n, \tilde{x}_{t_n}^{\pi,l,j}, u, \tilde{p}_{t_n}^{\pi,l,j}, \tilde{q}_{t_n}^{\pi,l,j}, \tilde{r}_{t_n}^{\pi,l,j})$;
            $\tilde{x}_{t_{n+1}}^{\pi,l,j} = \tilde{x}_{t_n}^{\pi,l,j} + b(t_n, \tilde{x}_{t_n}^{\pi,l,j}, \tilde{u}_{t_n}^{\pi,l,j}) \Delta t_n + c(t_n, \tilde{x}_{t_n}^{\pi,l,j}, \tilde{u}_{t_n}^{\pi,l,j}) \Delta B_{t_n} + \sum_{k=1}^{dN_{t_n}} \gamma(t_n, \tilde{x}_{t_n}^{\pi,l,j}, \tilde{u}_{t_n}^{\pi,l,j}, \Delta L_k^n)$
            $- \int_{\mathbb{R}^n \setminus \{0\}} \gamma(t_n, \tilde{x}_{t_n}^{\pi,l,j}, \tilde{u}_{t_n}^{\pi,l,j}, z) \mu(dz) \Delta t_n$;
            $\tilde{p}_{t_{n+1}}^{\pi,l,j} = \tilde{p}_{t_n}^{\pi,l,j} - \nabla_x H(t_n, \tilde{x}_{t_n}^{\pi,l,j}, \tilde{u}_{t_n}^{\pi,l,j}, \tilde{p}_{t_n}^{\pi,l,j}, \tilde{q}_{t_n}^{\pi,l,j}, \tilde{r}_{t_n}^{\pi,l,j}) \Delta t_n + \tilde{q}_{t_n}^{\pi,l,j} \Delta B_{t_n} + \tilde{r}_{t_n}^{\pi,l,j} \sum_{k=1}^{dN_{t_n}} \Delta L_k^n$
            $- \lambda \tilde{r}_{t_n}^{\pi,l,j} \sum_{k=1}^{dN_{t_n}} \mathbb{E}(\Delta L_k^n) \Delta t_n$;
loss $= \frac{1}{M} \sum_{j=1}^{M} \left[ \tilde{p}_{t_N}^{\pi,l,j} - \nabla_x g(\tilde{x}_{t_N}^{\pi,l,j}) \right]^2$;
$J(\tilde{u}^{\pi,l}) = \frac{1}{M} \sum_{j=1}^{M} \left[ \frac{1}{M} \sum_{n=0}^{N-1} f(t_n, \tilde{x}_{t_n}^{\pi,l,j}, \tilde{u}_{t_n}^{\pi,l,j}) + h(\tilde{x}_{t_n}^{\pi,l,j}) \right]$;
$(\theta_q^{l+1}, \theta_r^{l+1}, \tilde{p}_0^{\pi,l+1}) = (\theta_q^l, \theta_r^l, \tilde{p}_0^{\pi,l}) - \eta \nabla_{\text{loss}}$;
```

3.2. Numerical algorithm 2

To overcome the difficulty of Algorithm 1 and avoid solving the maximization problem at each time step, we propose a new algorithm based on three neural networks. This algorithm assumes that all coefficients of the Hamiltonian system (8) are continuously differentiable in u. These conditions insure that the control domain $U = \mathbb{R}^n$ and the optimal control $\tilde{u}$ are located on the boundary of the control domain, and then the maximization condition implies

$$H_u(t; \tilde{x}_t; \tilde{u}_t; \tilde{p}_t; \tilde{q}_t) = 0, \quad \forall u, \quad \forall t \in [0, T], \quad \text{a.s.}$$

Therefore, this modification of the constraints leads to a change in the cost function of the new variational problem to be solved. The cost function of the new problem is as follows:

$$\inf_{\tilde{p}_0, \{\tilde{q}_t\}, \{\tilde{r}_t\}, \{\tilde{u}_t\}} \mathbb{E} \left[\left| \tilde{p}_{T_N}^{q,u,r} - \nabla_x g(\tilde{x}_{T_N}^{q,u,r}) \right|^2 + \alpha \int_0^T H_u(t; \tilde{x}_t^{q,u,r}; \tilde{u}_t; \tilde{p}_t^{q,u,r}; \tilde{q}_t, \tilde{r}_t)^2 dt \right], \tag{12}$$

α is a hyperparameter. Using the same approach as in the proof of Proposition 1, it is clear that if the cost function (12) tends to 0, then the $(\tilde{x}_t^{q,u,r}; \tilde{u}_t; \tilde{p}_t^{q,u,r}; \tilde{q}_t, \tilde{r}_t)$ converges to the optimal solution of the problem (8). Compared to Algorithm 1, this time, it is necessary to construct a new neural network $\mathcal{NN}_3^{\theta_u}$ that allows us to simulate $\tilde{u}_{t_n}$. This new network has $(1+2n)$-dim in the input layer and a k-dimensional output layer. For the hidden layers, $\mathcal{NN}_3^{\theta_u}$ has the same structure as $\mathcal{NN}_1^{\theta_q}$ and $\mathcal{NN}_2^{\theta_r}$. These three networks are trained according to the following cost function:

$$\frac{1}{M} \sum_{j=1}^{M} \left[\left(\tilde{p}_{t_N}^{\pi,l,j} - \nabla_x g(\tilde{x}_{t_N}^{\pi,l,j}) \right)^2 + \alpha \frac{T}{N} \sum_{n=0}^{N-1} \left(\nabla_u H(t_n, \tilde{x}_{t_n}^{\pi,l,j}, \tilde{u}_{t_n}^{\pi,l,j}, \tilde{p}_{t_n}^{\pi,l,j}, \tilde{q}_{t_n}^{\pi,l,j}, \tilde{r}_{t_n}^{\pi,l,j}) \right)^2 \right].$$

Thus, the pseudo-code for Algorithm 2 is as follows.

Algorithm 2 Algorithm with 3-NNs.

```
Require: Initial parameters $(\theta_q^0;\theta_r^0,\theta_u^0,\tilde{p}_0^{0,\pi})$, $x_0{}^\pi$, $\eta$ as learning rate and the hyper-parameter $\alpha$;
Ensure: $(\tilde{x}_{t_n}^{l,\pi},\tilde{u}_{t_n}^{l,\pi},\tilde{p}_{t_n}^{l,\pi},\tilde{q}_{t_n}^{l,\pi},\tilde{r}_{t_n}^{l,\pi})$;
for $l=0,\dots,\mathrm{MaxIter}$
    for $j=1,\dots,M$
        Generate $\Delta W_j^{t_n}\sim\mathcal{N}(0,\Delta t_n)$, $\Delta N_j^{t_n}\sim P(\lambda\Delta t_n)$ and $\Delta L_{j,k}^{t_n}\sim\frac{\mu(dz)}{\lambda}1_{\mathbb{R}^n\setminus\{0\}}$ for each $k=1,\dots,dN_j^{t_n}$
        where $\mu(dz)=\exp(-\nu\|z\|)dz$, $z\in\mathbb{R}^n\setminus\{0\}$;
        $\tilde{x}_{t_0}^{\pi,l,j}=\tilde{x_0}^\pi$, $H=0$, $\tilde{p}_{t_0}^{\pi,l,j}=\tilde{p_0}^{\pi,l}$;
        for $n=0,\dots,N-1$
            $q_{t_n}^{l,\pi,j}=\mathcal{NN}_1^{\theta_q}(t_n,\tilde{x}_{t_n}^{\pi,l,j},\tilde{p}_{t_n}^{\pi,l,j};\theta_q^l)$;
            $r_{t_n}^{l,\pi,j}=\mathcal{NN}_2^{\theta_r}(t_n,\tilde{x}_{t_n}^{\pi,l,j},\tilde{p}_{t_n}^{\pi,l,j};\theta_r^l)$;
            $u_{t_n}^{l,\pi,j}=\mathcal{NN}_3^{\theta_u}(t_n,\tilde{x}_{t_n}^{\pi,l,j},\tilde{p}_{t_n}^{\pi,l,j};\theta_u^l)$;
            $\tilde{x}_{t_{n+1}}^{\pi,l,j}=\tilde{x}_{t_n}^{\pi,l,j}+b(t_n,\tilde{x}_{t_n}^{\pi,l,j},\tilde{u}_{t_n}^{\pi,l,j})\,\Delta t_n+c(t_n,\tilde{x}_{t_n}^{\pi,l,j},\tilde{u}_{t_n}^{\pi,l,j})\,\Delta B_{t_n}+\sum_{k=1}^{dN_{t_n}}\gamma(t_n,\tilde{x}_{t_n}^{\pi,l,j},\tilde{u}_{t_n}^{\pi,l,j},\Delta L_k^n)$
            $-\int_{\mathbb{R}^n\setminus\{0\}}\gamma(t_n,\tilde{x}_{t_n}^{\pi,l,j},\tilde{u}_{t_n}^{\pi,l,j},z)\mu(dz)\,\Delta t_n$;
            $\tilde{p}_{t_{n+1}}^{\pi,l,j}=\tilde{p}_{t_n}^{\pi,l,j}-\nabla_x H(t_n,\tilde{x}_{t_n}^{\pi,l,j},\tilde{u}_{t_n}^{\pi,l,j},\tilde{p}_{t_n}^{\pi,l,j},\tilde{q}_{t_n}^{\pi,l,j},\tilde{r}_{t_n}^{\pi,l,j})\,\Delta t_n+\tilde{q}_{t_n}^{\pi,l,j}\Delta B_{t_n}+\tilde{r}_{t_n}^{\pi,l,j}\sum_{k=1}^{dN_{t_n}}\Delta L_k^n$
            $-\lambda\tilde{r}_{t_n}^{\pi,l,j}\sum_{k=1}^{dN_{t_n}}\mathbb{E}(\Delta L_k^n)\Delta t_n$;
            $H=H+\big(\nabla_u H(t_n,\tilde{x}_{t_n}^{\pi,l,j},\tilde{u}_{t_n}^{\pi,l,j},\tilde{p}_{t_n}^{\pi,l,j},\tilde{q}_{t_n}^{\pi,l,j},\tilde{r}_{t_n}^{\pi,l,j})\big)^2$;
loss $=\frac{1}{M}\sum_{j=1}^M\Big[(\tilde{p}_{t_N}^{\pi,l,j}-\nabla_x g(\tilde{x}_{t_N}^{\pi,l,j}))^2+\alpha H\Big]$;
$J(\tilde{u}^{\pi,l})=\frac{1}{M}\sum_{j=1}^M\Big[\frac{T}{N}\sum_{n=0}^{N-1}f(t_n,\tilde{x}_{t_n}^{\pi,l,j},\tilde{u}_{t_n}^{\pi,l,j})+h(\tilde{x}_{t_N}^{\pi,l,j})\Big]$;
$(\theta_q^{l+1};\theta_u^{l+1};\theta_r^{l+1};\tilde{p}_0^{\pi,l+1})=(\theta_q^l;\theta_u^l;\theta_r^l;\tilde{p}_0^{\pi,l})-\eta\nabla_{\mathrm{loss}}$;
```

4. Numerical results

In this section, we will test our algorithms on a mean-variance portfolio selection problem in a financial market with two different assets: one risky asset priced at P_1 and one risk-free asset priced at P_0. Given that the market is self-financing, i.e., a small change in the wealth process x_t is necessarily due to a change in P_0 or P_1. The problem consists of

$$\sup_{v\in\mathcal{A}}\mathbb{E}\left[-\tfrac{1}{2}(x_T-a)^2\right]$$

subject to

$$dx_t=(\alpha_t x_t+(\beta_t-\alpha_t)v_t)\,dt+k_t\,v_t\,dB_t+v_{(t^-)}\int_{\mathbb{R}\setminus\{0\}}\gamma_1(t,z)\,\tilde{\mathcal{L}}(dt,dz),\tag{13}$$

where $x_0\in\mathbb{R}$ is known and $a=\mathbb{E}[x_T]$ the expected wealth at the terminal time T. B_t and $\tilde{\mathcal{L}}(dt,dz)$ represent, respectively, a Brownian motion and a compensated random Poisson measure.

The control $v_t=x_t-P_0\pi_0$ is the amount invested in the risky asset at instant t. Here π_0 represents the number of units invested in the free asset. β_t (resp. α_t) is the expected return of the risky asset (resp. of the risk-free asset). However, the term $k_t v_t dB_t$ models continuous movements in the price of the risky asset due to Brownian motion and the last term $v_{t^-}\int_{\mathbb{R}\setminus\{0\}}\gamma_1(t,z)\tilde{\mathcal{L}}(dt,dz)$ represents the jump component, which models discontinuous movements in the price of the risky asset due to jumps. This problem can be viewed as special case of our initial problem, where

$$f(t,x_t,u_t)=0,\quad g(x)=-\tfrac{1}{2}(x-a)^2,$$
$$b(t,x_t,v_t)=\alpha_t x_t+(\beta_t-\alpha_t)v_t,\quad \sigma(t,x_t,v_t)=k_t v_t,\quad \gamma(t,x_t,v_t,z)=v_{(t^-)}\gamma_1(t,z).$$

We suppose that $t\longleftrightarrow\int_{\mathbb{R}\setminus\{0\}}\gamma_1^2(t,z)\,\mu(dz)$ is locally bounded. v_t is admissible control and we write $v_t\in\mathcal{A}$ if $x_t=x_t^v$ is the solution of (13) and $\mathbb{E}[(x_T^v)^2]<\infty$. The investor aims to determine $v_t\in\mathcal{A}$ that maximizes the objective function

$$J(v_t)=\mathbb{E}\left[-\tfrac{1}{2}\left(x_T-a\right)^2\right].$$

It is well known that this problem, even in the case of a single risky asset, does not admit a closed form analytical solution. A semi-analytical solution exists only under highly restrictive and simplified assumptions. In what follows, we address the problem using both algorithms and compare the numerical results with those obtained from the semi-analytical method provided by by Sulem et al. in [21]. This semi-analytical solution is provided by the following set of formulas,

$$v_t^* = \frac{(\alpha_t - \beta_t)(\varphi_t x_t + \psi_t)}{\varphi_t \Gamma_t},$$

such that

$$\varphi_t = -\exp\left(\int_t^T \left((\alpha_s - \beta_s)^2 \frac{1}{\Gamma_s} - 2\alpha_s\right) ds\right), \quad 0 \leqslant t \leqslant T,$$

$$\psi_t = a \exp\left(\int_t^T \left((\alpha_s - \beta_s)^2 \frac{1}{\Gamma_s} - \alpha_s\right) ds\right), \quad 0 \leqslant t \leqslant T,$$

$$\Gamma_t = k_t^2 + \int_{\mathbb{R}\setminus\{0\}} \gamma_1^2(t,z)\, \mu(dz).$$

As a first step for solving the problem using our approach, we provide the Hamiltonian and the corresponding adjoint process associated with this problem. The Hamiltonian H gets the form

$$H(t,x,v,p,q,r) = \big(\alpha_t x + (\beta_t - \alpha_t)\, v\big)\, p + k_t\, v\, q + v \int_{\mathbb{R}\setminus\{0\}} \gamma_1(t,z)\, r_t(z)\, \mu(dz).$$

And, the adjoint equations is

$$\begin{aligned} dp_t &= -\alpha_t\, p_t\, dt + q_t\, dB_t + \int_{\mathbb{R}\setminus\{0\}} r_t(z)\, \hat{\mathcal{L}}(dt,dz), \quad t < T, \\ p_T &= -(x_T - a). \end{aligned}$$

So, the variational problem associated to the algorithm 1 is

$$\inf_{p_0, q_t, r_t} \mathbb{E}\left[\left|(x_T^{p_0,q,r} - a) + p_T^{p_0,q,r}\right|^2\right] \tag{14}$$

subject to

$$\begin{cases} x_t^{p_0,q,r} = x_0 + \displaystyle\int_0^t \big(\alpha_s x_s^{p_0,q,r} + (\beta_s - \alpha_s) v_s^{p_0,q,r}\big)\, ds + \int_0^t k_s v_s^{p_0,q,r} dB_s + \int_0^t \int_{\mathbb{R}\setminus\{0\}} v_s^{p_0,q,r} \gamma_1(s,z)\, \widetilde{\mathcal{L}}(ds,dz), \\ p_t^{p_0,q,r} = p_0 - \displaystyle\int_0^t \alpha_s\, p_s^{p_0,q,r} ds + \int_0^t q_s\, dB_s + \int_0^t \int_{\mathbb{R}\setminus\{0\}} r_t(z)\, \widetilde{\mathcal{L}}(ds,dz), \\ v_t^{p_0,q,r} = \arg\max_{v\in U} H(t, x_t^{p_0,q,r}, v, p_t^{p_0,q,r}, q_t, r_t(\cdot)) \quad \forall t. \end{cases}$$

The following table shows the optimal value of $J(v^*)$ obtained by Algorithm 1, along with the relative error compared to the semi-analytical method, for different iterations: MaxIter = 1000, 3000, and 5000.

Table 1. The value of $J(v^*)$ obtained by algorithm 1 and by the semi-analytical method

Results	ALg1	Analy Method	ALg1	Analy Method	ALg1	Analy Method
$J(v^*)$	-0.009725	-0.012001	-0.01051	-0.01187	-0.010863	-0.01198
Iterations	1000	—	3000	—	5000	—
Relative Error	2.276×10^{-3}	2.276×10^{-3}	1.36×10^{-3}	1.36×10^{-3}	1.12×10^{-3}	1.12×10^{-3}
Time (s)	2420	—	3100	—	3524	—

To compare the two solutions one obtained using our approach based on Algorithm 1 and the other using the semi-analytical method, we plot the evolution of J_{alg1} and J_{analy} over the iterations (see Figure 2).

We also plot the evolution of the mean of x_T over the iterations to observe its behavior with respect to the target value a (see Figure 3).

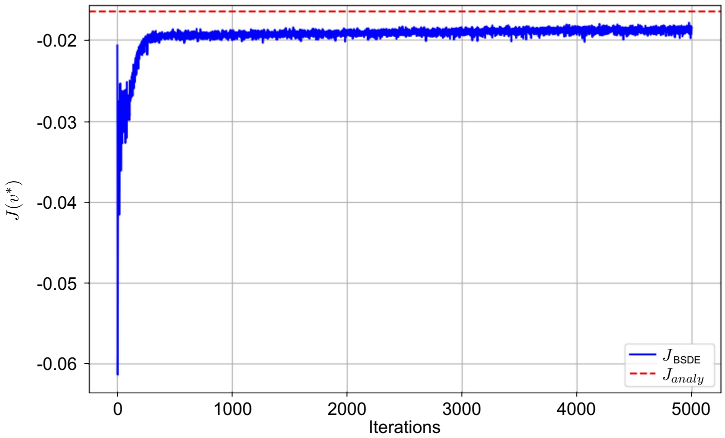

Figure 2. The evolution of J_{alg1} and J_{analy} over the iterations.

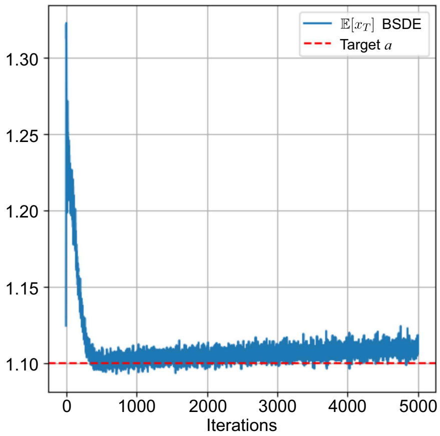

Figure 3. The evolution of the mean of x_T relative to the target a over the iterations.

And for ensuring the convergence of our algorithm, we plotted the loss function against the number of iterations (see Figure 4).

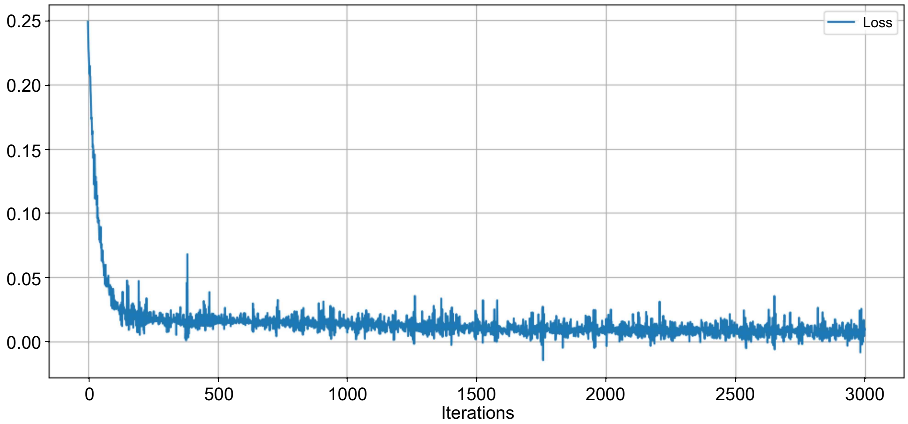

Figure 4. The loss function of Algorithm 1 against the number of iterations.

It is clear that this method is convergent and provides results with an accuracy on the order of 10^{-3}.

Now, we solve this problem using Algorithm 2. This will require 3 neural networks instead of 2, but it allows us to avoid computing the argmax function, which is computationally expensive. However, this also involves a change in the loss function, making it in the form

$$\frac{1}{M}\sum_{j=1}^{M}\left[\left(\tilde{p}_T^{\pi,l,j}+\left(x_T^{\pi,l,j}-a\right)\right)^2+\eta\sum_{n=0}^{N-1}(\nabla_v H)^2\right], \tag{15}$$

where η is a hyperparameter.

The following table presents the values of $J(v^*)$ obtained using Algorithm 2, along with the relative error compared to the semi-analytical method, for different iterations: MaxIter $=1000$, 3000, and 5000.

Table 2. $J(v^*)$ obtained using Algorithm 2, along with the relative error compared to the semi-analytical method for different iterations.

Results	ALg2	Analy Method	ALg2	Analy Method	ALg2	Analy Method
$J(v^*)$	-0.012287	-0.011561	-0.01203	-0.01155	-0.01192	-0.01147
Iterations	1000	—	3000	—	5000	—
Relativ Error	7.26×10^{-4}	7.26×10^{-4}	6.3×10^{-4}	6.3×10^{-4}	4.5×10^{-4}	4.5×10^{-4}
Time (s)	840	—	1260	—	1680	—

To compare the two solutions obtained by our approach based on Algorithm 2 and the semi-analytical method, we plot the evolution of J_{alg2} and J_{analy} over the iterations (see Figure 5).

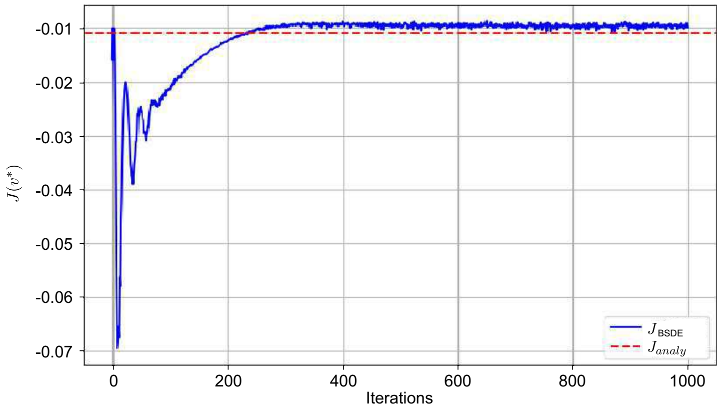

Figure 5. The evolution of J_{alg2} and J_{analy} over the iterations.

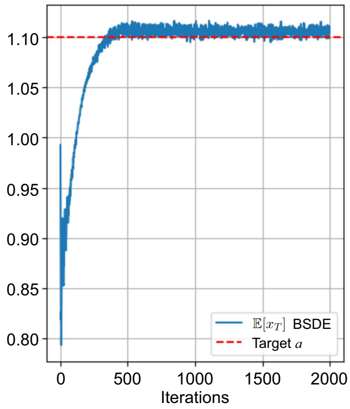

Figure 6. The behavior of $\mathbb{E}[x_T]$ with respect to the target value a over iterations.

We have also ensured the convergence of the second algorithm and plotted the evolution of the mean of x_T over the iterations to observe its behavior with respect to the target value a; see Figures 7 and 6, respectively.

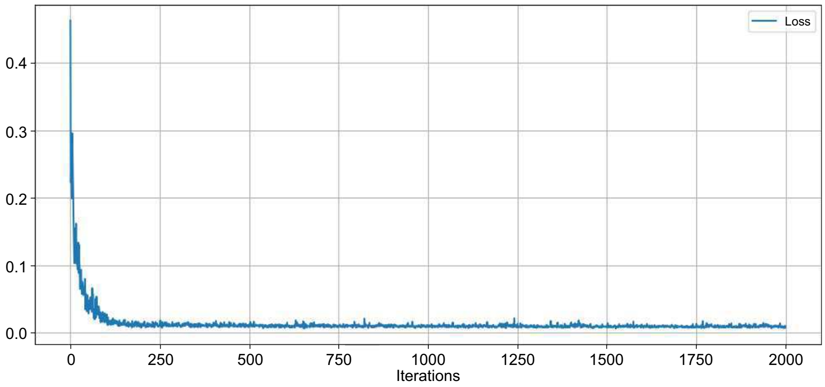

Figure 7. The loss function for Algorithm 2 over iterations.

5. Conclusion

In this work, we have proposed two algorithms to solve a stochastic optimal control problem driven by a diffusion process. We validated our approaches through a financial case study specifically, the mean-variance portfolio selection problem, where the wealth process is modeled by a backward stochastic differential equation (FBSDE) with jumps. This problem does not admit a fully analytical solution, but only a semi-analytical one based on simplifications and approximations.

Our approach relies on deep neural networks combined with the Stochastic Pontryagin Maximum Principle. Numerical results show that our methods converge toward the semi-analytical solution of the problem. For comparison, we observed that Algorithm 2 is more stable and converges faster than Algorithm 1. The relative error for Algorithm 1 is on the order of 10^{-3}, while for Algorithm 2 it reaches 10^{-4}. This improvement is attributed to the fact that Algorithm 1 requires solving an $\arg\max$ optimisation problem, which, in our case, involves an additional auxiliary algorithm (FBGS algorithm).

In conclusion, Algorithm 2 is preferable when the conditions stated in Section 3.2 are met. However, Algorithm 1 remains a valuable alternative, especially for this type of control problem involving jumps and multiple financial assets, where analytical methods are not available.

[1] Merton R. C. Lifetime Portfolio Selection under Uncertainty: The Continuous-Time Case. The Review of Economics and Statistics. **51** (3), 247–257 (1969).

[2] Zariphopoulou T. Consumption–Investment Models with Constraints. SIAM Journal on Control and Optimization. **32** (1), 59–85 (1994).

[3] Oksendal B., Sulem A. Optimal consumption and portfolio with both fixed and proportional transaction costs. SIAM Journal on Control and Optimization. **40** (6), 1765–1790 (2002).

[4] Haussmann U. G. A Stochastic Maximum Principle for Optimal Control of Diffusions. John Wiley & Sons, Inc. (1986).

[5] Bellman R. Dynamic programming and stochastic control processes. Information and Control. **1** (3), 228–239 (1958).

[6] Kushner H. J. Numerical methods for stochastic control problems in continuous time. SIAM Journal on Control and Optimization. **28** (5), 999–1048 (1990).

[7] Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press (2016).

[8] Jiequn H., Weinan E. Deep learning approximation for stochastic control problems. Preprint arXiv:1611.07422 (2016).

[9] Weinan E., Jiequn H., Arnulf J. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. Communications in Mathematics and Statistics. **5** (4), 349–380 (2017).

[10] Ji S., Peng S., Peng Y., Zhang X. Three algorithms for solving high-dimensional fully coupled FBSDEs through deep learning. IEEE Intelligent Systems. **35** (3), 71–84 (2020).

[11] Ji S., Peng S., Peng Y., Zhang X. Solving stochastic optimal control problem via stochastic maximum principle with deep learning method. Journal of Scientific Computing. **93** (1), 30 (2022).

[12] Germain M., Pham H., Warin X. Neural networks-based algorithms for stochastic control and PDEs in finance. Preprint arXiv:2101.08068 (2021).

[13] Carmona R., Laurière M. Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games: II — the finite horizon case. The Annals of Applied Probability. **32** (6), 4065–4105 (2022).

[14] Hu R., Laurière M. Recent developments in machine learning methods for stochastic control and games. Preprint arXiv:2303.10257 (2023).

[15] Andersson K., Andersson A., Oosterlee C. W. Convergence of a robust deep FBSDE method for stochastic control. SIAM Journal on Scientific Computing. **45** (1), A226–A255 (2023).

[16] Boltyanski V. G., Gamkrelidze R. V., Mishchenko E. F., Pontryagin L. S. The maximum principle in the theory of optimal processes of control. IFAC Proceedings Volumes. **1** (1), 464–469 (1960).

[17] Peng S. A general stochastic maximum principle for optimal control problems. SIAM Journal on Control and Optimization. **28** (4), 966–979 (1990).

[18] Tang S., Li X. Necessary conditions for optimal control of stochastic systems with random jumps. SIAM Journal on Control and Optimization. **32** (5), 1447–1475 (1994).

[19] Framstad N. C., Øksendal B., Sulem A. Sufficient stochastic maximum principle for the optimal control of jump diffusions and applications to finance. Journal of Optimization Theory and Applications. **121**, 77–98 (2004).

[20] Meng Q., Dong Y., Shen Y., Tang S. Optimal controls of stochastic differential equations with jumps and random coefficients: stochastic Hamilton–Jacobi–Bellman equations with jumps. Applied Mathematics & Optimization. **87** (1), 3 (2023).

[21] Øksendal B., Sulem A. Applied Stochastic Control of Jump Diffusions. Springer Berlin, Heidelberg (2007).

[22] Han J., Long J. Convergence of the deep BSDE method for coupled FBSDEs. Probability, Uncertainty and Quantitative Risk. **5** (1), 5 (2020).

[23] Agram N., Øksendal B. Deep learning for conditional McKean–Vlasov jump diffusions. Systems & Control Letters. **188**, 105815 (2024).

[24] Kingma D. P., Ba J. Adam: A method for stochastic optimization. Preprint arXiv:1412.6980 (2015).

[25] Hornik K., Stinchcombe M., White H. Multilayer feedforward networks are universal approximators. Neural Networks. **2** (5), 359–366 (1989).

[26] Liu D. C., Nocedal J. On the limited memory BFGS method for large scale optimization. Mathematical Programming. **45** (1), 503–528 (1989).

Новий підхід до розв'язання стохастичної задачі керування, що визначається дифузійним процесом зі стрибками

Ульгазі Г.

Лабораторія IMIA, команда A2MSDS, кафедра математики, факультет наук і техніки, Університет Мулая Ісмаїла в Мекнесі, а/с 509, Буталамін, Ер-Рашидія 52000, Марокко

Стаття зосереджена на чисельному розв'язанні багатовимірних задач оптимального стохастичного керування, стани систем в яких моделюються як процеси дифузії зі стрибками. Використовуючи принцип максимуму та глибокі нейронні мережі, переформульовано вихідну задачу керування як варіаційну задачу та впроваджено спеціалізовані алгоритми для розв'язання цього нового формулювання. Представлено самі алгоритми, а також різні архітектури, що були застосовані. Задача вибору портфеля за критерієм "середнє–дисперсія" на фінансовому ринку, що складається з двох видів активів у межах процесу дифузії зі стрибками, підтверджує ефективність запропонованих алгоритмів.

Ключові слова: *прямо-зворотні стохастичні диференціальні рівняння зі стрибками; стохастичний принцип максимуму; глибокі нейронні мережі; оптимальне стохастичне керування; дифузії зі стрибками; задача вибору портфеля за критерієм "середнє–дисперсія".*