Folia Oeconomica Stetinensia
Volume 23 (2023) Issue 1
DOI: 10.2478/foli-2023-0001 | 1–15
ISSN (print): 1730-4237 | ISSN (online): 1898-0198
www.wnus.edu.pl/fos

SOLVING FINITE-HORIZON DISCOUNTED NON-STATIONARY MDPS

El Akraoui Bouchra

Slimane University
Laboratory of Information Processing
and Decision Support, Sultan Moulay, Morocco.
e-mail: mbouchra2@gmail.com
ORCID: 0000-0003-0717-6482

Cherki Daoui

Slimane University
Laboratory of Information Processing
and Decision Support, Sultan Moulay, Morocco.
e-mail:_daouic@yahoo.com
ORCID: 0000-0001-5435-6414

Received 1.0.2023, Revised 21.02.2023, Accepted 26.02.2023

Abstract

Research background: Markov Decision Processes (*MDPs*) are a powerful framework for modeling many real-world problems with finite-horizons that maximize the reward given a sequence of actions. Although many problems such as investment and financial market problems where the value of a reward decreases exponentially with time, require the introduction of interest rates.
Purpose: This study investigates non-stationary finite-horizon *MDPs* with a discount factor to account for fluctuations in rewards over time.
Research methodology: To consider the fluctuations of rewards with time, the authors define new non-stationary finite-horizon *MDPs* with a discount factor. First, the existence of an optimal policy for the proposed finite-horizon discounted *MDPs is proven*. Next, a new Discounted Backward Induction (*DBI*) algorithm is presented to find it. To enhance the value of their proposal, a financial model is used as an example of a finite-horizon discounted *MDP* and an adaptive *DBI* algorithm is used to solve it.
Results: The proposed method calculates the optimal values of the investment to maximize its expected total return with consideration of the time value of money.
Novelty: No existing studies have before examined dynamic finite-horizon problems that account for temporal fluctuations in rewards.

Keywords: Markov Decision Process, Dynamic Programming, Backward Induction algorithm

JEL classification: C02, G11, C44

Introduction

Sequential decision-making is a widespread topic in the area of artificial intelligence. It is the process of deciding on a sequence of actions to take in an uncertain environment based on previous experience to reach specific objectives. Nowadays, Markov Decision Processes provide one of the most powerful formal frameworks for solving decision-making tasks. It has attracted a great deal of interest from researchers as a means of alleviating a wide range of challenges in many complex real-life problems including commercial application (Rimélé *et al.*, 2021), inventory control (Feinberg, 2016), maintenance management (Zhang *et al.*, 2017), health care management (Pavitsos, Kyriakidis, 2009), robot navigation (Ye *et al.*, 2020), Petri Nets (Allamigeon *et al.*, 2021), and physics problems (Bertsimas, Mišić, 2016). *MDPs* can be classified depending on the time horizon if the decisions taken have infinite-horizon and finite-horizon *MDPs*, which differ in analytical properties as well as resolution algorithms. Efficient solvers for *MDPs* had been developed, the most widely used algorithms are Linear Programming (*LP*), Policy Iteration (*PI*), Value Iteration (*VI*), and its variants (Puterman, 1994).

The infinite-horizon *MDPs* are generally desirable when the reward functions and transition probabilities are stationary (time-invariant). For these problems, the total expected discounted reward is the most widely used optimality criteria, whereby the discount factor is used to guarantee that an optimal policy exists with a polynomial-time convergence (Puterman, 1994). Under the total criterion (no discount factor), an optimal stationary policy exists only if there is a dead-end state with an immediate reward of 0 (like death in medical decision models). However, the stationary assumption, on the other hand, is not realistic in many cases, as the transition probability stands for the probability of the disease developing over time.

For several real-world applications, the finite-horizon framework is suitable for situations that terminate at a particular point in time (for example in manufacturing planning over a specific period).

In this context, this work is motivated by the following investment problem introduced in (Spieksma, Nunez-Queija, 2015). Consider an investor who has 10,000 € and must determine the appropriate manner in which to invest it to maximize its expected total return. He/she can choose to invest his/her entire capital in either share of company A or the shares of company B. Putting money into company A will result in a 100% return after one year with a probability of 0.10. On the other hand, with a probability of 0.90, there will be no profit after one year and the investor will recoup his/her investment. Company B has a higher risk profile but offers higher expected returns. With a probability of 0.60, the investment is doubled, while with a probability

of 0.40, the investment is completely lost. If it does not fail, the investor can reinvest his/her money every year (10,000 € each time due to the popularity of the two investments). What is the best investment strategy for the investor, if his/her goal is to maximize the expected profit after five years?

The authors (Spieksma, Nunez-Queija, 2015) modeled this problem as a non-stationary finite-horizon *MDP* and used the classical Backward Induction (*BI*) algorithm as a solver. However, this model does not take into account the fluctuations in the amount invested during each year, which represents an interest rate. It is well known that it is more beneficial to make a profit sooner rather than later. It is therefore known that the time value of the dollar is generally influenced by inflation (price increases) and interest rates that make the value of the dollar in the present greater than it will be in the future. Consequently, in this study, the discounted framework is preferred, we introduce the discount factor α in the objective function. The α-discount factor accounts for the difference that exists between the current value and the future value of the sums invested.

Accordingly, in the first contribution of this paper, we introduce a new definition of a non-stationary finite-horizon discounted *MDPs*, with finite state and action spaces; we prove the existence of an optimal policy, and we propose a new discounted backward induction algorithm to solve it. This algorithm, based on the classical BI algorithm, is easier to implement, alleviates real-time systems issues, and its polynomial convergence bounds are guaranteed (Ye, 2011).

To enhance the value of our proposal, we model the above investment problem as finite-horizon discounted *MDPs* and we present an adaptive *DBI* algorithm to solve it.

The remainder of the paper is organized as follows. In the second section a general introduction to finite-horizon discounted non-Stationary *MDPs* is presented. In the next section, the finite-horizon non-stationary *MDPs* under the discounted criterion are defined; it is also shown that there exists an optimal policy, and a new *DBI* algorithm is proposed to find it. In the fourth section, our algorithm is applied to a financial example. Finally, we conclude, and present our perspectives in possible future work.

1. Finite-horizon non-stationary Markov decision processes

Markov decision processes, also known as stochastic dynamic programming, have been widely studied since their earliest introduction in 1960 (Howard, 1960). They are a well-known mathematical formalism, applied mainly for modeling and solving dynamic decision-making issues in various fields of science and engineering.

The *MDP* framework involves decision epochs, states, actions, rewards, as well as transition probabilities. The selection of an action from a state yields a reward and determines the state in the next decision epoch using a transition probability function.

The goal is to yield the optimal decision policies (action selection) that maximize the expected rewards or minimize costs under each dynamic stochastic environment (Hordijk, Kallenberg, 1984; Sutton, 1988). In this section, we will briefly establish the theory of finite-horizon non-stationary *MDPs* with finite state and action spaces.

1.1. Formalism

An *MDP* is defined by a four tuple (E, A, r, P) where E is a finite set of states where the evolution of the process takes place, when the process occupies state i, a finite set of actions is available and denoted by $A(i)$. We can consider a discrete-time system where actions are taken at different decision epochs $t = 1, 2, \ldots, T$; where T design the finite planning horizon.

Let the random variables X_t and A_t denote respectively the state and action of the system at time t. Once an action a is chosen at time t, an immediate reward r_{ia}^t is accrued and the process moves to a new state j with the transition probability $p_{iaj}^t = P\left(X_{t+1} = j | A_t = a, X_t = i\right)$ (means the probability of the transition from state i to state j by taking action a at time t).

1.2. Strategy and value function

Let His_t denote the set of all histories his_t through to time t, $H_t = \{h_t = (i_1, a_1, ..., i_{t-1}, a_{t-1}, i_t) : (i_n, a_n) \in E \times A$, and $1 \le n \le t-1; i_t \in E\}$. Only the choice rules that apply to the first T decision time points are important, a Policy π is defined by the sequence $\pi = (\pi^1, \pi^2, \ldots, \pi^T)$, where π^t is a decision rule at time point t. It is a function on $H_t \times A := \left\{\left(h_t, a_t\right) : h_t \in H_t,\ a_t \in A\left(i_t\right)\right\}$, which gives the probability of the action to be taken at time t, given the history h_t.

A Markov policy π is one in which π^t is determined only by the present state at time t. A Markov policy with identical decision criteria is known as a stationary policy. Finally, a pure or deterministic policy is a stationary policy with only one nonrandomized decision criterion.

The total expected reward $V_i^T(\pi)$ over the planning horizon T, for every strategy π and any initial state $i \in E$, is denoted by:

$$V_i^T(\pi) = \sum_{t=1}^{T} \sum_{j \in E} \sum_{a \in A(j)} P_\pi\left(X_t = j, Y_t = a | X_1 = i\right) r_{ja}^t \tag{1}$$

where X_t and Y_t are the random variables representing respectively the state and action at time t.

The core problem of finite-horizon non-stationary *MDPs* is determining the best policy for each state. The policy that optimizes the performance metric is referred to as π^* as an optimal policy, i.e., $V_i^T\left(\pi^*\right)=\sup_{\pi} V_i^T\left(\pi\right), i \in E$. Besides, we define the optimal vector value V_i^T: $V_i^T=\sup_{\pi} V_i^T\left(\pi\right), i \in E$.

There are several solution algorithms for finite-horizon *MDPs*. The most widely used method is the backward induction algorithm (Puterman, 1994). We will discuss this further in the next section.

2. Finite-horizon non-stationary MDPs under the discounted criterion

In most sequential problem-solving methods, the reward functions and state transition functions are assumed to be time-invariant, i.e., the underlying *MDPs* are stationary. However, this assumption is violated in a wide range of real-world applications, where non-stationarity arises naturally, as in the case of dynamic pricing (Mao *et al.*, 2018), automated medical care (Asadi *et al.*, 2022), energy management system for a photovoltaic-assisted electric vehicle charging station (Wu *et al.*, 2020), Spot Cloud Services (Peng *et al.*, 2023) and traffic management (Emadi *et al.*, 2022). Similar to stationary *MDPs*, the non-stationary *MDPs* are highly non-trivial due to the general main challenges highlighted by (Dulac-Arnold *et al.*, 2021; White III, White, 1989).

In operations research, existing algorithms finite-horizon non-stationary *MDPs* as dynamic programming namely the *BI* algorithm (Bertsekas, 2012), are usually considered the expected total reward criterion. They do not treat the problems with discounting. Since many projects contain cash flows that are dispersed over several years, it is important to take the time value of money into account because this makes it challenging to estimate the benefit that will be obtained from initiatives. Future financial flows are perceived as being less valuable than those that happen right away due to the time value of money. The further in the future, they occur, the more the discount there must be to represent the greater loss in value as they approach. By discounting each future cash flow proportionately to how far in the future it occurs, we can compare all the future cash flows that result from accepting a project on an even playing field. For all these reasons, the finite-horizon non-stationary discounted *MDPs* with full information about the probability distributions and rewards associated to them are considered in this study.

In this section, we first briefly discuss the *BI* algorithm, then we define the finite-horizon non-stationary *MDPs* under the discounted criterion, we show that there exists an optimal policy. Furthermore, we propose a *DBI* algorithm that computes such an optimal policy.

2.1. Backward induction

The classical backward induction technique is typically used to solve finite-horizon problems (Bellman, 1958), it is an iterative process that starts at the last time period for all possible states and the agent decides the best action for these states. This gives an optimal value for that state in that period. Then, in the next-to-last period for all feasible states, the agent chooses the most suitable action for those states, resulting in the optimal values for the different states in the next period. The process continues backward until the current period is reached period.

The time complexity of the classical *BI* algorithm is $O\left(T\left|A\right|E^2\right)$ for a number of states $E=\left|E\right|$, actions $A=\left|A\right|$, and horizon *T*, see (Larach *et al.*, 2017).

The ameliorated backward induction algorithm described in (El Akraoui *et al.*, 2022) allows reducing the execution time of the classical *BI* algorithm. They were inspired to apply the approach proposed by (Larach *et al.*, 2017). The authors introduced the list of successors of the state-action referred to as $\Gamma_a^+(i)$ for each action a, where $\Gamma_a^+(i)=\left\{j\in E: P_{iaj}^t>0,\ t=1,\ \ldots,\ T\right\}$. This reduced the temporal complexity to $O\left(T\left|\Gamma_a^+\right|E^2\right)$ arithmetic operations, where Γ_a^+ stands for the average number of state-action successors.

2.2. Discounted Backward induction

Considering a finite-horizon discounted *MDP*, the authors show the existence of an optimal policy for the proposed *DBI* Algorithm.

For every strategy π and any initial state $i \in E$; we designate by $V_i^{\alpha,T}(\pi)$ the α-discounted expected reward over the planning horizon *T*, given by equation 2:

$$V_i^{\alpha,T}(\pi)=\sum_{t=1}^{T}\sum_{j\in E}\sum_{a\in A(j)}\alpha^{t-1}P_\pi(X_t=j,Y_t=a\left|X_1=i\right.)\ r_{ja}^t \tag{2}$$

The strategy π^* is an α-optimal policy, if it satisfies, $V_i^{\alpha,T}(\pi^*)=\sup_\pi V_i^{\alpha,T}(\pi)$, $i\in E$.

We define the optimal value vector $\mathrm{V}^{\alpha,T}$ by: $V_i^{\alpha,T}=\sup_\pi V_i^{\alpha,T}(\pi)$, $i\in E$.

For each deterministic decision rule f^t at time point t = 1, 2, …, *T*, we associate the transition matrix P(f^t) and the reward vector r(f^t) defined by equation 3:

$$\left[P\left(f^t\right)\right]_{ij} = \sum_{a\in A(i)} P^t_{iaj}\, f^t(i,a),\ i\in E,\ j\in E, \text{ and } \left[r\left(f^t\right)\right]_i = \sum_{a\in A(i)} r^t_{ia}\, f^t(i,a),\ i\in E \quad (3)$$

The next lemma introduced in (Kallenberg, 1983), allows us to show the correctness of Theorem 1 that shows how to build an α-optimal policy, and explains the procedure followed in the *DBI* algorithm.

Lemma 1: There exists the Markov policy π^* for each starting state i as well as each policy π such as:

$$P_\pi\left(X_t = j, A_t = a \mid X_1 = i\right) = P_{\pi^*}\left(X_t = j, A_t = a \mid X_1 = i\right); \forall (j,a)\in E\times A, \forall t \quad (4)$$

Theorem 1: Allow $v_i^{\alpha,\,T+1} = 0, \forall i\in E$. Let us define a deterministic decision rule f^t as well as a vector $v^{\alpha,t}$ for $t = T, \ldots, 2, 1$ successively as follows:

$$\left\{r\left(f^t\right)\right\}_i + \alpha\left\{P\left(f^t\right)v^{\alpha,t+1}\right\}_i = \max_{a\in A(i)}\left\{r^t_{ia} + \alpha\sum_{j\in E} p^t_{iaj} v^{\alpha,\,t+1}_j\right\}, \forall i\in E \quad (5)$$

and

$$v^{\alpha,t} = r\left(f^t\right) + \alpha\, P\left(f^t\right)v^{\alpha,t+1} \quad (6)$$

Finally, $v^{\alpha,1}$ is the optimal value vector $V^{\alpha,T}$ and $\pi^* = (f^1, f^2, \ldots, f^T)$ is an α-optimal policy.

Proof: Proof is given by induction on the horizon T. Let $\pi = (\pi^1, \pi^2, \ldots, \pi^T)$ be an arbitrary policy. According to equation (2) for $T = 1$ we obtain:

$$V_i^{\alpha,1}(\pi) = \sum_{j\in E}\sum_{a\in A(j)} P_\pi(X_1 = j, Y_1 = a \,|\, X_1 = i) r^1_{ja} =$$
$$= \sum_{a\in A(i)} r^1_{ia}\pi^1_{ia} \le \max_{a\in A(i)} r^1_{ia} = v_i^{\alpha,1} = V_i^{\alpha,1}\left(\pi^*\right), i\in E$$

where $\pi^t_{ia} = P_\pi(Y_t = a \,|\, X_1 = i)$.

For $T = 1, \ldots, t$, let's assume the result is correct. According to Lemma 1, there exists a Markov policy $\bar{\pi}$ in the form $V_i^{\alpha,t+1}(\bar{\pi}) = V_i^{\alpha,t+1}(\pi)$ for $i\in E$, where $\bar{\pi} = \sigma^1, \sigma^2, \ldots, \sigma^{t+1}$. Define the Markov policy $\pi' = \rho^1, \rho^2, \ldots, \rho^t$ by $\rho^n_{ja} = \sigma^{n+1}_{ja}$ for $n = 1, 2, \ldots, t$.

In the planning period $t + 1$, $v^{\alpha,2}$ would be the same as $v^{\alpha,1}$ in the planning period t, based on the induction assumption and the policy π' definition, we obtain that $V_j^{\alpha,t}(\pi') \le v_j^{\alpha,2}$, $j \in E$. Therefore,

$$V_i^{\alpha,t+1}(\pi) = V_i^{\alpha,t+1}(\overline{\pi}) = \sum_{a \in A(i)} \sigma_{ia}^1 \left\{ r_{ia}^1 + \alpha \sum_{j \in E} p_{iaj}^1 V_j^{,t}(\pi') \right\} \le$$

$$\le \sum_{a \in A(i)} \sigma_{ia}^1 \left\{ r_{ia}^1 + \alpha \sum_{j \in E} p_{iaj}^1 v_j^{\alpha,2} \right\} \le \max_{a \in A(i)} \left\{ r_{ia}^1 + \alpha \sum_{j \in E} p_{iaj}^1 v_j^{\alpha,2} \right\} = v_i^{\alpha,1}$$

Furthermore, by using equation (6) for $t = 1$, we have the following:

$$v^{\alpha,1} = r(f^1) + \alpha P(f^1)\, v^{\alpha,2}$$

Replacing $v^{\alpha,2}$ with its formula gives:

$$v^{\alpha,1} = r(f^1) + \alpha P(f^1)\{r(f^2) + \alpha P(f^2)\, v^{\alpha,3}\}$$

And so on, we then obtain the following equation:

$$v^{\alpha,1} = \ldots = \sum_{s=1}^{t+1} \alpha^{k-1} \left\{ P\left(f^1\right) P\left(f^2\right) \ldots P\left(f^{k-1}\right) r\left(f^k\right) \right\} = V^{\alpha,t+1}\left(\pi^*\right)$$

Eventually,

$$V^{\alpha,t+1}(\pi^*) = v^{\alpha,1} \ge V^{\alpha,t+1}(\pi)$$

Then $v^{\alpha,1}$ represents the optimal value vector, and π^* is an α-optimal policy. ♦

Now, we present the discounted Backward Induction algorithm (Algorithm 1) that determines an α-optimal policy for discounted non-stationary *MDPs* over T periods. The *DBI* algorithm is an iterative procedure based on Bellman's principle of optimality, we then calculate the values of the earlier periods starting at the end of the planning horizon. Following that, an optimal policy is discovered after T iterations.

Algorithm1: *DBI* algorithm

DBI (In *MDP*: E, P, A, R, α, T; Out($V^{\alpha*}$, π^*))

1. $t \leftarrow T + 1$;
2. $i \in E$ **Take** $v = 0$;
3. **Repeat**

4. t ← $t - 1$;
5. **For each** $i \in E$

   5.**a.** The deterministic rule f^t is defined as:

$$\left\{r\left(f^t\right)\right\}_i + \alpha\left\{P\left(f^t\right)v^{\alpha,t+1}\right\}_i = \max_{a\in A(i)}\left\{r_{ia}^t + \alpha\sum_{j\in E} p_{iaj}^t\, v_j^{\alpha,\,t+1}\right\} \tag{7}$$

   5.**b.** $$v^{\alpha,t} = r(f^t) + \alpha P(f^t)\, v^{\alpha,t+1} \tag{8}$$

   End For

6. **Return** V^{α^*}, π^*
7. $V^{\alpha^*} = v^{\alpha,1}$ is the optimal value vector, and $\pi^* = (f^1, f^2, ..., f^T)$ stands for an α-optimal policy.

The following section presents first, the modelling of the previous financial example by using discounted finite-horizon *MDPs* which illustrates the decision rules over five years.

3. Experimental results and case study

As mentioned earlier, this paper was motivated by a simple financial example that necessitates 640 combinations (Spieksma, Nunez-Queija, 2015). The number of decision rules can be significantly reduced by using dynamic programming methods. To account for price fluctuations over time, we treat this problem as a finite-horizon discounted *MDP*. The key idea assumes that the value of a dollar today is worth more than a dollar tomorrow especially due to inflation (growth of prices) and interest rates, meaning that the value of the dollar decreases through time. It cannot purchase goods at the current value.

This investment problem is considered as a dynamical system that is observed at times $t = 1, 2, ..., 5$. At each instant t, the state of the process is the potential of capital invested each year t represented by a random variable X_t with values in a set of states $E = \{1, 10{,}000\ €, 20{,}000\ €, 30{,}000\ €, 40{,}000\ €, 50{,}000\ €, 60{,}000\ €\}$, for simplicity, we note $E = \{0, 1, 2, 3, 4, 5, 6\}$. If the process is in state i, an investor chooses an action a from the action set $A(i)$. For $i \in E \setminus \{0, 6\}$, $A(i) = \{A, B\}$, the cases $i = 0$ and $i = 6$ mean that no investment can be done (nothing done), $A(0) = A(6) = \{\theta\}$. The action chosen at time t can be considered as a random variable A_t. Each time the process X_t is in state i at time t, then regardless of the history of the process, the agent gains an immediate reward r_{iaj} and the system moves to another state j with a transition

probability p_{iaj} and the process continues in the same way from state j. If the system is in state $i \in E$ and the action $a \in A(i)$ is taken as the expected reward r_{ia}, would be that:

$$r_{ia} = \sum_{j \in S} p_{iaj} \times r_{iaj} \tag{9}$$

Furthermore, investing 10,000 € in company A yields an immediate return of 1,000 € (0.9 × 0 + 0.1 × 10,000), while investing 10,000 € in company B yields an immediate return of 2,000 € (0.4 × (–10,000) + 0.6 × 10,000). For simplicity, we consider stationary rewards and transition probabilities. Besides, when the process is in state $i \in \{1, 2, 3, 4, 5\}$ and choosing action company A the possible transitions are state i or state $i + 1$, but when action company B is chosen the possible transitions are state $i – 1$ or state $i + 1$.

This *MDP* model can be represented by a graph $G = (E, U)$ where the set of nodes E is the state space, and the set of arcs U is defined by: $U = \{(i, j) \in E \times E : \exists \alpha \in A(i), p_{iaj} > 0\}$ (see Figure 1).

For different actions, the arcs are represented as follows:
→ (Action is company A) and →→ (Action is company B) →→→ (Action is θ). We also note in Figure 1 the probabilities (p_{iaj}) on the arc corresponding to action a.

We can summarize the data of this *MDP* by:

$$E = \{0, 1, 2, 3, 4, 5, 6\}; A(1) = A(2) = A(3) = A(4) = A(5) = \{A, B\}; A(0) = A(6) = \{\theta\};$$
$$\alpha = 0.9; \mathrm{T} = 5$$

– for $i = 0$: $p_{0\theta 0} = 1$; $r_{0\theta} = 0$ €;
– for $i = \{1, 2, 3, 4, 5\}$: $p_{iAi} = 0.9$; $p_{iAi+1} = 0.1$; $r_{iA} = 1{,}000$ €;
$p_{iBi-1} = 0.4$; $p_{iBi+1} = 0.6$; $r_{iB} = 2{,}000$ €;
– for $i = 6$: $p_{6\theta 6} = 1$; $r_{6\theta} = 0$ €.

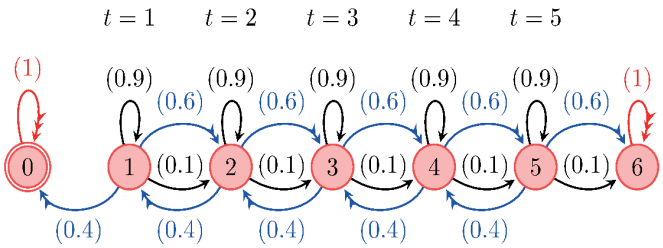


Figure 1. *MDP* graph for the investment problem

Source: own elaboration.

For solving this investment problem, we apply the *DBI* algorithm (algorithm1) with the factor rate $\alpha = 0.9$. The results below show the maximum expected profit and the good investment strategy over five years.

Let us start with $v = 0$ for all $i \in E$ and based on equations (7) and (8) to determine the optimal investment decision at time t and the corresponding value vector. For each state i {0, 1, 2, 3, 4, 5, 6} $i \in \{1, 2, 3, 4, 5, 6\}$ at time t, equation (7) becomes for this problem as below:

$$V_i^{\alpha,t} = \max\left\{\left(r_{iA} + \alpha\left(p_{iAi} \times V_i^{\alpha,t+1} + p_{iAi+1} \times V_{i+1}^{\alpha,t+1}\right)\right);\ \left(r_{iB} + \alpha\left(p_{iBi-1} \times V_{i-1}^{\alpha,t+1} + p_{iBi+1} \times V_{i+1}^{\alpha,t+1}\right)\right)\right\}.$$

Thus, for $i \in \{1, 2, 3, 4, 5\}$ gives:

$$V_i^{\alpha,t} = \max\left\{\left(1000 + 0.9\left(0.9 \times V_i^{\alpha,t+1} + 0.1 \times V_{i+1}^{\alpha,t+1}\right)\right); \left(2000 + 0.9\left(0.4 \times V_{i-1}^{\alpha,t+1} + 0.6 \times V_{i+1}^{\alpha,t+1}\right)\right)\right\}.$$

The proposed algorithms are tested using Intel(R) Core (TM) i7-6500U (2.6 GHz), C++ implementation, and Windows 10 operating system (64bits). The following are the outputs of this process:

T = 5	**T = 4**
i = 0 : $f^5(0) = \theta$; $V_0^{\alpha,5} = 0.00$;	i = 0 : $f^4(0) = \theta$; $V_0^{\alpha,4} = 0.00$;
i = 1 : $f^5(1) = B$; $V_1^{\alpha,5} = 2000.00$;	i = 1 : $f^4(1) = B$; $V_1^{\alpha,4} = 3080.00$;
i = 2 : $f^5(2) = B$; $V_2^{\alpha,5} = 2000.00$;	i = 2 : $f^4(2) = B$; $V_2^{\alpha,4} = 3800.00$;
i = 3 : $f^5(3) = B$; $V_3^{\alpha,5} = 2000.00$;	i = 3 : $f^4(3) = B$; $V_3^{\alpha,4} = 3800.00$;
i = 4 : $f^5(4) = B$; $V_4^{\alpha,5} = 2000.00$;	i = 4 : $f^4(4) = B$; $V_4^{\alpha,4} = 3800.00$;
i = 5 : $f^5(5) = B$; $V_5^{\alpha,5} = 2000.00$;	i = 5 : $f^4(5) = B$; $V_5^{\alpha,4} = 2720.00$;
i = 6 : $f^5(6) = \theta$; $V_6^{\alpha,5} = 0.00$	i = 6 : $f^4(6) = \theta$; $V_6^{\alpha,4} = 0.00$
T = 3	**T = 2**
i = 0 : $f^3(0) = \theta$; $V_0^{\alpha,3} = 0.00$;	i = 0 : $f^2(0) = \theta$; $V_0^{\alpha,2} = 0.00$;
i = 1 : $f^3(1) = B$; $V_1^{\alpha,3} = 4052.00$;	i = 1 : $f^2(1) = B$; $V_1^{\alpha,2} = 4786.83$;
i = 2 : $f^3(2) = B$; $V_2^{\alpha,3} = 5160.80$;	i = 2 : $f^2(2) = B$; $V_2^{\alpha,2} = 6385.52$;
i = 3 : $f^3(3) = B$; $V_3^{\alpha,3} = 5420.00$;	i = 3 : $f^2(3) = B$; $V_3^{\alpha,2} = 6469.76$;
i = 4 : $f^3(4) = B$; $V_4^{\alpha,3} = 4836.80$;	i = 4 : $f^2(4) = B$; $V_4^{\alpha,2} = 5769.92$;
i = 5 : $f^3(5) = B$; $V_5^{\alpha,3} = 3368.00$;	i = 5 : $f^2(5) = B$; $V_5^{\alpha,2} = 3741.25$;
i = 6 : $f^3(6) = \theta$; $V_6^{\alpha,3} = 0.00$	i = 6 : $f^2(6) = \theta$; $V_6^{\alpha,2} = 0.00$

T = 1

i = 0 : $f^1(0) = \theta$; $V_0^{\alpha,1} = 0.00$;

i = 1 : $f^1(1) = A$; $V_1^{\alpha,1} = 5452.03$;

i = 2 : $f^1(2) = B$; $V_2^{\alpha,1} = 7216.93$;

i = 3 : $f^1(3) = B$; $V_3^{\alpha,1} = 7414.54$;

i = 4 : $f^1(4) = B$; $V_4^{\alpha,1} = 6349.39$;

i = 5 : $f^1(5) = B$; $V_5^{\alpha,1} = 4077.17$;

i = 6 : $f^1(6) = \theta$; $V_6^{\alpha,1} = 0.00$

An optimal policy in 5 periods is $\pi^* = (f^1, f^2, f^3, f^4, f^5)$ and the optimal vector value is

$$V^{\alpha*} = (0, 5{,}452.03, 7{,}216.93, 7{,}414.54, 6{,}349.39, 4{,}077.17, 0).$$

The solution to our problem can now be found in the above result. The final decision can be easily deduced: the investor should invest 10,000 € (i.e. $i = 1$ in the output result) in A at the first stage, and then his/her total expected capital is 15,452.03 €. The optimal investment decision at subsequent times depends on the realized capital.

4. Discussion

The approach presented in this study for investigating non-stationary finite-horizon *MDPs* with a discount factor is a promising and novel method for modeling dynamic decision-making problems. The discount factor enables the incorporation of fluctuations in rewards over time, making it useful in scenarios where the value of future rewards is uncertain or variable. The approach has many potential applications in real-world decision-making problems, such as investment decisions, resource allocation, and scheduling in manufacturing processes, operations research, and energy management, among others. In finance, it can help investors determine the optimal investment strategy, considering the fluctuation in market conditions and expected returns over time.

Moreover, the model provides a framework for modeling decision-making problems in situations where there may be uncertainty about the underlying probability distributions of rewards. This is relevant when the reward distribution changes over time due to external factors or changes in the environment.

In terms of computational efficiency, the approach presented in this study provides an efficient and scalable method for solving non-stationary finite-horizon *MDPs* with a discount

factor. This is particularly important in situations where there are large numbers of states or actions, as traditional methods for solving *MDPs* can become computationally infeasible.

Conclusions

The initial objective of the presented work was to identify the best sequence of actions that form an optimal policy to increase the return value of investors in future campaigns considering the temporal fluctuations of the rewards. The Markov decision process is a very useful framework for modeling problems that maximize return by taking a sequence of actions. Although many issues such as investment and financial market problems where the value of a reward decreases exponentially over time.

From a business perspective, a dollar is worth more today than it would be worth tomorrow. In general, the higher the discount, the higher the level of risk associated with an investment and its future cash flows. Discounting is the primary factor used to price a stream of tomorrow's cash flow.

In this paper, we have presented our main contributions. The first one is to define a new non-stationary finite-horizon *MDP* under discounting the criterion dedicated to dealing with real sequential problems with interest rates and a finite number of steps. We also proved the existence of an optimal policy.

The second contribution focused on resenting the proposed *DBI* algorithm, based on the classical *BI* algorithm to compute the optimal value function.

To show the usefulness and power of our proposed framework. We modeled a real-world financial problem as a finite-horizon discounted *MDP* and solved it using an adaptive *DBI* algorithm.

This work, of course, still needs to be expanded and requires more effort and perseverance. Especially, when the state and action spaces become larger. The well-known solvers for *MDPs* that require many mathematical operations and a large memory will not be able to be used. In future work, we will use some techniques of the "divide and conquer" paradigm, to cope with large state and action spaces and in doing so reduce the execution time of the *DBI* algorithm.

References

Allamigeon, X., Boyet, M., Gaubert, S. (2021). Piecewise Affine Dynamical Models of Petri Nets–Application to Emergency Call Centers. *Fundamenta Informaticae*, *183*(3–4), 169–201. DOI: 10.3233/FI-2021-2086.

Asadi, A., Pinkley, S.N., Mes, M. (2022). A Markov decision process approach for managing medical drone deliveries. *Expert Systems With Applications*, *204*, 117490. DOI: 10.1016/j.eswa.2022.117490.

Bellman, R. (1958). Dynamic programming and stochastic control processes. *Information and Control*, *1*(3), 228–239. DOI: 10.1016/S0019-9958(58)80003-0.

Bertsekas, D. (2012). *Dynamic programming and optimal control: Volume I* (vol. 1). Athena scientific.

Bertsimas, D., Mišić, V.V. (2016). Decomposable markov decision processes: A fluid optimization approach. *Operations Research*, *64*(6), 1537–1555. DOI: 10.1287/opre.2016.1531.

Dulac-Arnold, G., Levine, N., Mankowitz, D.J., Li, J., Paduraru, C., Gowal, S., Hester, T. (2021). Challenges of real-world reinforcement learning: Definitions, benchmarks and analysis. *Machine Learning*, *110*(9), 2419–2468. DOI: 10.1007/s10994-021-05961-4.

El Akraoui, B., Daoui, C., Larach, A. (2022). Decomposition Methods for Solving Finite-Horizon Large MDPs. *Journal of Mathematics*, *2022*. DOI: 10.1155/2022/8404716.

Emadi, H., Atkins, E., Rastgoftar, H. (2022). A Finite-State Fixed-Corridor Model for UAS Traffic Management. *ArXiv Preprint ArXiv:2204.05517*.

Feinberg, E.A. (2016). Optimality conditions for inventory control. In *Optimization Challenges in Complex, Networked and Risky Systems* (pp. 14–45). INFORMS. DOI: 10.1287/educ.2016.0145.

Hordijk, A., Kallenberg, L.C.M. (1984). Transient policies in discrete dynamic programming: Linear programming including suboptimality tests and additional constraints. *Mathematical Programming*, *30*(1), 46–70. DOI: 10.1007/BF02591798.

Howard, R.A. (1960). Dynamic programming and markov processes. *MIT Press, Cambridge, MA*. https://books.google.co.ma/books?id=fXJEAAAAIAAJ.

Kallenberg, L.C.M. (1983). Linear programming and finite Markovian control problems, Math. *Centre Tracts*, *148*, 1–245.

Larach, A., Chafik, S., Daoui, C. (2017). Accelerated decomposition techniques for large discounted Markov decision processes. *Journal of Industrial Engineering International*, *13*(4), 417–426. DOI: 10.1007/s40092-017-0197-7.

Mao, W., Zheng, Z., Wu, F., Chen, G. (2018). Online Pricing for Revenue Maximization with Unknown Time Discounting Valuations. *IJCAI*, 440–446. DOI: 10.24963/ijcai.2018/61.

Pavitsos, A., Kyriakidis, E.G. (2009). Markov decision models for the optimal maintenance of a production unit with an upstream buffer. *Computers & Operations Research*, *36*(6), 1993–2006. DOI: 10.1016/j.cor.2008.06.014.

Peng, H., Cheng, Y., Li, X. (2023). Real-Time Pricing Method for Spot Cloud Services with Non-Stationary Excess Capacity. *Sustainability*, *15*(4), 3363.

Puterman, M.L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons Inc. DOI: 10.1002/9780470316887.

Rimélé, A., Grangier, P., Gamache, M., Gendreau, M., Rousseau, L.-M. (2021). E-commerce warehousing: Learning a storage policy. *ArXiv Preprint ArXiv:2101.08828*. DOI: 10.48550/arXiv.2101.08828.

Spieksma, F., Nunez-Queija, R. (2015). Markov Decision Processes. *Adaptation of the Text by R. Nunez-Queija*, 55.

Sutton, R.S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, *3*(1), 9–44. DOI: 10.1007/BF00115009.

White III, C.C., White, D.J. (1989). Markov decision processes. *European Journal of Operational Research*, *39*(1), 1–16. DOI: 10.1016/0377-2217(89)90348-2.

Wu, Y., Zhang, J., Ravey, A., Chrenko, D., Miraoui, A. (2020). Real-time energy management of photovoltaic-assisted electric vehicle charging station by markov decision process. *Journal of Power Sources*, *476*, 228504.

Ye, G., Lin, Q., Juang, T.-H., Liu, H. (2020). Collision-free Navigation of Human-centered Robots via Markov Games. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 11338–11344. DOI: 10.1109/ICRA40945.2020.9196810.

Ye, Y. (2011). The simplex and policy-iteration methods are strongly polynomial for the Markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, *36*(4), 593–603. DOI: 10.1287/moor.1110.0516.

Zhang, Y., Kim, C.-W., Tee, K.F. (2017). Maintenance management of offshore structures using Markov process model with random transition probabilities. *Structure and Infrastructure Engineering*, *13*(8), 1068–1080. DOI: 10.1080/15732479.2016.1236393.

Citation

Bouchra, A., Daoui, C. (2023). Solving Finite-Horizon Discounted Non-Stationary MDPs. *Folia Oeconomica Stetinensia*, *23*(1), 1–15. DOI: 10.2478/foli-2023-0001.